

Fuzzy CoCo: Balancing Accuracy and Interpretability of Fuzzy Models by Means of Coevolution.

Carlos-Andrés Peña-Reyes¹ and Moshe Sipper²

¹ Logic Systems Laboratory, Swiss Federal Institute of Technology in Lausanne, CH-1015 Lausanne, Switzerland, Email: Carlos.Pena@epfl.ch

² Department of Computer Science, Ben-Gurion University, Beer-Sheva 84105, Israel, Email: sipper@cs.bgu.ac.il

Abstract. In this chapter we present Fuzzy CoCo, a fuzzy modeling technique based on cooperative coevolution, conceived to provide high numeric precision (accuracy) while incurring as little a loss of linguistic descriptive power (interpretability) as possible. The search for interpretability is represented by several constraints taken into account when designing the evolutionary algorithm, which induce the drive for accuracy. Interpretability-oriented fuzzy modeling must conduct two separate but intertwined search processes: (1) the search for membership functions, and (2) the search for rules. Towards this end, Fuzzy CoCo employs two coevolving species: database (membership functions) and rule base. Coevolution allows to overcome limitations presented by single-population evolutionary algorithms when confronted with fuzzy modeling, including stagnation, convergence to local optima, and computational costliness. We demonstrate the efficacy of Fuzzy CoCo by applying it to a hard, real-world problem—prediction of breast-cancer malignancy—obtaining excellent results.

1 Introduction

The earliest fuzzy systems were constructed using knowledge provided by human experts, and were thus linguistically correct. However, the difficulty of applying such an approach for high-dimension, ill-known, or data-intensive models, led to the coming of new data-driven fuzzy modeling techniques.

These techniques initially concentrated on solving a parameter-optimization problem based on the numeric performance of the systems. Unfortunately, they paid little attention to linguistic aspects, thus neglecting one of the most important advantages offered by fuzzy systems. Recently, as fuzzy modeling techniques have concentrated more on linguistic issues, the difficulty of improving system interpretability without losing (numeric) performance has become evident. This accuracy-interpretability trade-off is currently one of the most active research lines in fuzzy modeling.

In this chapter we present Fuzzy CoCo, a fuzzy modeling technique based on cooperative coevolution, conceived to provide high accuracy while incurring as little a loss of interpretability as possible. With Fuzzy CoCo the user

is in control of the balance between accuracy and interpretability thanks to the method’s configurability.

This chapter is organized as follows: In the next section we provide an overview of evolutionary computation and evolutionary fuzzy modeling. Then, Section 3 discusses some aspects related to interpretability requirements. Section 4 presents Fuzzy CoCo, our cooperative coevolutionary approach to fuzzy modeling. Section 5 then describes a sample application of Fuzzy CoCo to a hard problem: breast-cancer assessment by mammography interpretation. The results obtained are presented in Section 6. Finally, we present concluding remarks in Section 7.

2 Background

2.1 Evolutionary computation

The domain of evolutionary computation involves the study of the foundations and the applications of computational techniques based on the principles of natural evolution. Evolution in nature is responsible for the “design” of all organisms on earth, and for the strategies they use to interact with each other. Evolutionary algorithms employ this powerful design philosophy to find solutions to hard problems.

Generally speaking, evolutionary techniques can be viewed either as search methods, or as optimization techniques. As written by Michalewicz [10]:

Any abstract task to be accomplished can be thought of as solving a problem, which, in turn, can be perceived as a search through a space of potential solutions. Since usually we are after ‘the best’ solution, we can view this task as an optimization process.

Three basic mechanisms drive natural evolution: *reproduction*, *mutation*, and *selection*. The first two act on the *chromosomes* containing the genetic information of the *individual* (the *genotype*), rather than on the individual itself (the *phenotype*) while selection acts on the phenotype. Reproduction is the process whereby new individuals are introduced into a *population*. During sexual reproduction, *recombination* (or *crossover*) occurs, transmitting to the offspring chromosomes that are a melange of both parents’ genetic information. Mutation introduces small changes into the inherited chromosomes; it often results from copying errors during reproduction. Selection, acting on the phenotype, is a process guided by the Darwinian principle of survival of the fittest. The fittest individuals are those best adapted to their environment, which thus survive and reproduce.

Evolutionary computation makes use of a metaphor of natural evolution, according to which a problem plays the role of an environment wherein lives a population of individuals, each representing a possible solution to the problem. The degree of adaptation of each individual (i.e., candidate solution)

to its environment is expressed by an adequacy measure known as the *fitness function*. The phenotype of each individual, i.e., the candidate solution itself, is generally encoded in some manner into its *genome* (genotype). Evolutionary algorithms potentially produce progressively better solutions to the problem. This is possible, thanks to the constant introduction of new “genetic” material into the population, by applying so-called genetic operators which are the computational equivalents of natural evolutionary mechanisms.

The archetypal evolutionary algorithm proceeds as follows: an initial population of individuals, $P(0)$, is generated at random or heuristically. Every evolutionary step t , known as a *generation*, the individuals in the current population, $P(t)$, are *decoded* and *evaluated* according to some predefined quality criterion, referred to as the fitness, or fitness function. Then, a subset of individuals, $P'(t)$ —known as the *mating pool*—is selected to reproduce, with selection of individuals done according to their fitness. Thus, high-fitness (“good”) individuals stand a better chance of “reproducing,” while low-fitness ones are more likely to disappear.

Selection alone cannot introduce any new individuals into the population, i.e., it cannot find new points in the search space. These points are generated by altering the selected population $P'(t)$ via the application of crossover and mutation, so as to produce a new population, $P''(t)$. Crossover tends to enable the evolutionary process to move toward “promising” regions of the search space. Mutation is introduced to prevent premature convergence to local optima, by randomly sampling new points in the search space. Finally, the new individuals $P''(t)$ are introduced into the next-generation population, $P(t+1)$; usually $P''(t)$ simply becomes $P(t+1)$. The termination condition may be specified as some fixed, maximal number of generations or as the attainment of an acceptable fitness level. Figure 1 presents the structure of a generic evolutionary algorithm in pseudo-code format.

```

begin EA
  t:=0
  Initialize population  $P(t)$ 
  while not done do
    Evaluate  $P(t)$ 
     $P'(t) := \text{Select}[P(t)]$ 
     $P''(t) := \text{ApplyGeneticOperators}[P'(t)]$ 
     $P(t+1) := \text{Introduce}[P''(t), P(t)]$ 
    t:=t+1
  end while
end EA

```

Fig. 1. Pseudo-code of a standard evolutionary algorithm.

Because they combine elements of directed and stochastic search, evolutionary techniques exhibit a number of advantages over other search methods. First, they usually require less knowledge and fewer assumptions about the

characteristics of the search space. Second, they can more easily avoid getting stuck in local optima. Finally, they strike a good balance between *exploitation* of the best solutions, and *exploration* of the search space. The strength of evolutionary algorithms derives from their population-based search, and from the use of the genetic mechanisms described above. The existence of a population of candidate solutions entails a parallel search, with the selection mechanism directing the search to the most promising regions, the crossover operator encouraging the exchange of information between these search-space regions, and the mutation operator enabling the exploration of new directions.

The application of an evolutionary algorithm involves a number of important considerations. The first decision to take when applying such an algorithm is how to encode candidate solutions within the genome. The representation must allow for the encoding of all possible solutions while being sufficiently simple to be searched in a reasonable amount of time. Next, an appropriate fitness function must be defined for evaluating the individuals. The (usually scalar) fitness value must reflect the criteria to be optimized and their relative importance. Representation and fitness are thus clearly problem-dependent, in contrast to selection, crossover, and mutation, which seem *prima facie* more problem-independent. Practice has shown, however, that while standard genetic operators can be used, one often needs to tailor these to the problem as well.

2.2 Evolutionary fuzzy modeling

Fuzzy modeling is the task of identifying the parameters of a fuzzy inference system so that a desired behavior is attained [23]. With the *direct* approach a fuzzy model is constructed using knowledge from a human expert. This task becomes difficult when the available knowledge is incomplete or when the problem space is very large, thus motivating the use of *automatic* approaches to fuzzy modeling. Selection of relevant variables and adequate rules is critical for obtaining a good system. One of the major problems in fuzzy modeling is the *curse of dimensionality*, meaning that the computation requirements grow exponentially with the number of variables.

A *fuzzy inference system* is a rule-based system that uses fuzzy logic to reason about data [24]. Its basic structure consists of four main components, as depicted in Figure 2: (1) a fuzzifier, which translates crisp (real-valued) inputs into fuzzy values; (2) an inference engine that applies a fuzzy reasoning mechanism to obtain a fuzzy output; (3) a defuzzifier, which translates this latter output into a crisp value; and (4) a knowledge base, which contains both an ensemble of fuzzy rules, known as the rule base, and an ensemble of membership functions known as the database.

The parameters of fuzzy inference systems can be classified into four categories (Table 1) [13, 14]: logical, structural, connective, and operational.

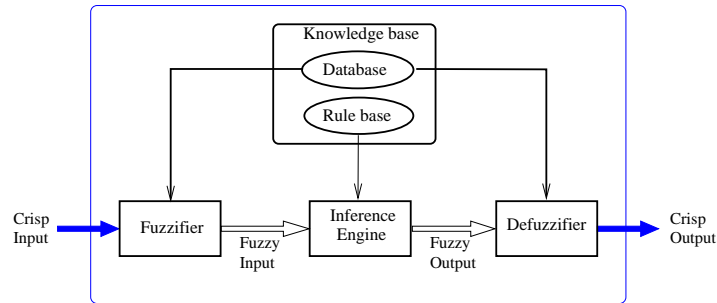


Fig. 2. Basic structure of a fuzzy inference system.

Table 1. Parameter classification of fuzzy inference systems.

Class	Parameters
Logical	Reasoning mechanism
	Fuzzy operators
	Membership function types
	Defuzzification method
Structural	Relevant variables
	Number of membership functions
	Number of rules
Connective	Antecedents of rules
	Consequents of rules
	Rule weights
Operational	Membership-function values

Logical parameters are usually predefined by the designer based on experience and on problem characteristics. Structural, connective, and operational parameters may be either predefined, or obtained by synthesis or search methodologies. Generally, the search space, and thus the computational effort, grows exponentially with the number of parameters. Therefore, one can either invest more resources in the chosen search methodology, or infuse more *a priori*, expert knowledge into the system (thereby effectively reducing the search space). The aforementioned trade-off between accuracy and interpretability is usually expressed as a set of constraints on the parameter values, thus complexifying the search process.

Evolutionary algorithms are used to search large, and often complex, search spaces. They have proven worthwhile on numerous diverse problems, able to find near-optimal solutions given an adequate performance (fitness) measure. Fuzzy modeling can be considered as an optimization process where part or all of the parameters of a fuzzy system constitute the search space. Works investigating the application of evolutionary techniques in the domain of fuzzy modeling had first appeared about a decade ago [8,9]. These focused mainly on the tuning of fuzzy inference systems involved in control tasks (e.g., cart-pole balancing, liquid-level system, and spacecraft rendezvous operation). Evolutionary fuzzy modeling has since been applied to an ever-

growing number of domains, branching into areas as diverse as chemistry, medicine, telecommunications, biology, and geophysics. For a detailed bibliography on evolutionary fuzzy modeling up to 1996, the reader is referred to [1, 3].

Depending on several criteria—including the available *a priori* knowledge about the system, the size of the parameter set, and the availability and completeness of input-output data—artificial evolution can be applied in different stages of the fuzzy-parameter search. Three of the four categories of fuzzy parameters in Table 1 can be used to define targets for evolutionary fuzzy modeling: structural parameters, connective parameters, and operational parameters [13, 14]. As noted before, logical parameters are usually predefined by the designer based on experience.

Knowledge tuning (operational parameters). The evolutionary algorithm is used to tune the knowledge contained in the fuzzy system by finding membership-function values. An initial fuzzy system is defined by an expert. Then, the membership-function values are encoded in a genome, and an evolutionary algorithm is used to find systems with high performance. Evolution often overcomes the local-minima problem present in gradient descent-based methods. One of the major shortcomings of knowledge tuning is its dependency on the initial setting of the knowledge base.

Behavior learning (connective parameters). In this approach, one assumes that expert knowledge is sufficient in order to define the membership functions; this determines, in fact, the maximum number of rules [23]. The genetic algorithm is used to find either the rule consequents, or an adequate subset of rules to be included in the rule base.

As the membership functions are fixed and predefined, this approach lacks the flexibility to modify substantially the system behavior. Furthermore, as the number of variables and membership functions increases, the curse of dimensionality becomes more pronounced and the interpretability of the system decreases rapidly.

Structure learning (structural parameters). In many cases, the available information about the system is composed almost exclusively of input-output data, and specific knowledge about the system structure is scant. In such a case, evolution has to deal with the simultaneous design of rules, membership functions, and structural parameters. Some methods use a fixed-length genome encoding a fixed number of fuzzy rules along with the membership-function values. In this case the designer defines structural constraints according to the available knowledge of the problem characteristics. Other methods use variable-length genomes to allow evolution to discover the optimal size of the rule base.

Both behavior and structure learning can be viewed as rule-base learning processes with different levels of complexity. They can thus be assimilated within other methods from machine learning, taking advantage of experience gained in this latter domain. In the evolutionary-algorithm community

there are two major approaches for evolving such rule systems: the Michigan approach and the Pittsburgh approach [10]. A more recent method has been proposed specifically for fuzzy modeling: the iterative rule learning approach [6]. These three approaches are briefly described below.

The Michigan approach. Each individual represents a single rule. The fuzzy inference system is represented by the entire population. Since several rules participate in the inference process, the rules are in constant competition for the best action to be proposed, and cooperate to form an efficient fuzzy system. The cooperative-competitive nature of this approach renders difficult the decision of which rules are ultimately responsible for good system behavior. It necessitates an effective credit-assignment policy to ascribe fitness values to individual rules.

The Pittsburgh approach. Here, the evolutionary algorithm maintains a population of candidate fuzzy systems, each individual representing an entire fuzzy system. Selection and genetic operators are used to produce new generations of fuzzy systems. Since evaluation is applied to the entire system, the credit-assignment problem is eschewed. This approach allows to include additional optimization criteria in the fitness function, thus affording the implementation of multi-objective optimization. The main shortcoming of this approach is its computational cost, since a population of full-fledged fuzzy systems has to be evaluated each generation.

The iterative rule learning approach. As in the Michigan approach, each individual encodes a single rule. An evolutionary algorithm is used to find a single rule, thus providing a partial solution. The evolutionary algorithm is used iteratively for the discovery of new rules, until an appropriate rule base is built. To prevent the process from finding redundant rules (i.e., rules covering the same input subspace), a penalization scheme is applied each time a new rule is added. This approach combines the speed of the Michigan approach with the simplicity of fitness evaluation of the Pittsburgh approach. However, as with other incremental rule-base construction methods, it can lead to a non-optimal partitioning of the antecedent space.

3 Interpretability Considerations

As mentioned before, the fuzzy-modeling process has to deal with an important trade-off between the *accuracy* and the *interpretability* of the model. The model is expected to provide high numeric precision while incurring as little a loss of linguistic descriptive power as possible. Currently, there exist no well-established definitions for interpretability of fuzzy systems, mainly due to the subjective nature of such a concept. However, some works have attempted to define objective criteria that facilitate the automatic modeling of interpretable fuzzy systems [5, 20].

The fuzzy system of Figure 2 processes information in three stages: the input interface (fuzzifier), the processing stage (inference engine), and the

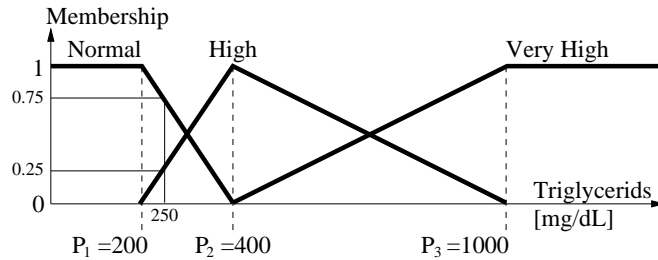


Fig. 3. Example of a fuzzy variable: *Triglycerides* has three possible fuzzy values, labeled **Normal**, **High**, and **Very High**, plotted above as degree of membership versus input value. The values P_i , setting the trapezoid and triangle apices, define the membership functions. In the figure, an example input value 250 mg/dL is assigned the membership values $\mu_{Normal}(250) = 0.75$, $\mu_{High}(250) = 0.25$, and $\mu_{VeryHigh}(250) = 0$. Note that $\mu_{Normal}(250) + \mu_{High}(250) + \mu_{VeryHigh}(250) = 1$.

output interface (defuzzifier). The interface deals with linguistic variables and their corresponding labels. These linguistic variables define the *semantics* of the system. The inference process is performed using fuzzy rules that define the connection between input and output fuzzy variables. These fuzzy rules define the *syntax* of the fuzzy system. Fuzzy modelers must thus take into account both semantic and syntactic criteria to obtain interpretable systems. Below, we present some criteria that represent conditions driving fuzzy modeling toward human-interpretable systems together with strategies to satisfy them.

3.1 Semantic criteria

The notion of “linguistic variable” formally requires associating a meaning to each fuzzy label [25]. Hence, each membership function should represent a linguistic term with a clear semantic meaning. For example, in Figure 3, the fuzzy variable *Triglycerides* has three meaningful labels: **Normal**, **High**, and **Very High**. The following semantic criteria describe a set of properties that the membership functions of a fuzzy variable should possess in order to facilitate the task of assigning linguistic terms [13, 14, 16]. The focus is on the meaning of the ensemble of labels instead of the absolute meaning of each term in isolation.

- *Distinguishability.* Each linguistic label should have semantic meaning and the fuzzy set should clearly define a range in the universe of discourse of the variable. In the example of Figure 3, to describe variable *Triglycerides* we used three meaningful labels: **Normal**, **High**, and **Very High**. Their membership functions are defined using parameters P_1 , P_2 , and P_3 .
- *Justifiable number of elements.* The number of membership functions of a variable should be compatible with the number of conceptual entities

a human being can handle. This number should not exceed the limit of 7 ± 2 distinct terms.

- *Coverage.* Any element from the universe of discourse should belong to at least one of the fuzzy sets. That is, its membership value must be different than zero for at least one of the linguistic labels. More generally, a minimum level of coverage ϵ may be defined, giving rise to the concept of *strong coverage*. Referring to Figure 3, we see that any value along the x-axis belongs to at least one fuzzy set; no value lies outside the range of all sets.
- *Normalization.* Since all labels have semantic meaning, then, for each label, at least one element of the universe of discourse should have a membership value equal to one. In Figure 3, we observe that all three sets **Normal**, **High**, and **Very High** have elements with membership value equal to 1.
- *Complementarity.* For each element of the universe of discourse, the sum of all its membership values should be equal to one (as in the example in Figure 3). This guarantees uniform distribution of meaning among the elements.

3.2 Syntactic criteria

A fuzzy rule relates one or more input-variable conditions, called antecedents, to their corresponding output fuzzy conclusions, called consequents. The example rule presented in Figure 4, associates the conditions **High** and **Middle** of the input variables *Triglycerides* and *Age*, respectively, with the conclusion **Moderate** of the output variable *Cardiac risk*. The linguistic adequacy of a fuzzy rule base lies on the interpretability of each rule as well as on that of the whole set of rules. The following syntactic criteria define some conditions which—if satisfied by the rule base—reinforce the interpretability of a fuzzy system [5].

- *Completeness.* For any possible input vector, at least one rule should be fired to prevent the fuzzy system from getting blocked.
- *Rule-base simplicity.* The set of rules must be as small as possible. If, however, the rule base is still large, rendering hard a global understanding of the system, the number of rules that fire simultaneously for any input vector must remain low in order to furnish a simple local view of the behavior.
- *Single-rule readability.* The number of conditions implied in the antecedent of a rule should be compatible with the aforementioned number of conceptual entities a human being can handle (i.e., $\leq 7 \pm 2$).
- *Consistency.* If two or more rules are simultaneously fired, their consequents should not be contradictory, i.e., they should be semantically close.

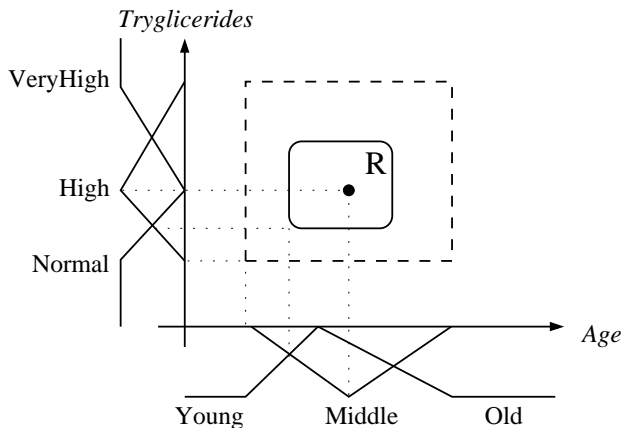


Fig. 4. Example of a fuzzy rule and its firing range. The rule **if** *Tryglicerides* **is** *High* **and** *Age* **is** *Middle* **then** *Cardiac risk* **is** *Moderate*, marked as **R**, is (partially) fired by input values into the dashed-line rectangle (i.e. $\mu(R) > 0$). The solid-line rectangle denotes the region where $\mu(R) > 0.5$.

3.3 Strategies to satisfy semantic and syntactic criteria

The criteria presented above, intended to assess interpretability of a fuzzy system, define a number of restrictions on the definition of fuzzy parameters. Semantic criteria limit the choice of membership functions, while syntactic criteria bind the fuzzy rule base. We present below some strategies to apply these restrictions when defining a fuzzy model.

Linguistic labels shared by all rules. A number of fuzzy sets is defined for each variable, which are interpreted as linguistic labels and shared by all the rules [5]. In other words, each variable has a unique semantic definition. This results in a grid partition of the input space as illustrated in Figure 5. To satisfy the completeness criterion, it is normally used a fully defined rule base, meaning that it contains all the possible rules, as in the example shown in Figure 6a. Label sharing by itself facilitates but does not guarantee the semantic integrity of each variable. More conditions are necessary.

Normal, orthogonal membership functions. The membership functions of two successive labels must be complementary (i.e., their sum must be equal to one) in their overlapping region, whatever form they have [4, 20]. Moreover, in such regions each label must ascend from zero to unity membership values [13, 14]. The variables presented in Figures 3 and 5, satisfy these requirements.

Don't-care conditions. A fully defined rule base, as that shown in Figure 6a, becomes impractical for high-dimension systems. The number of rules in a fully defined rule base exponentially increases as the number of input variables increases (e.g., a system with five variables, each with three labels, would contain $3^5 = 243$ rules). Moreover, given that each rule antecedent contains a condition for each variable, the rules might be too lengthy to be understandable, and too specific to describe general circumstances.

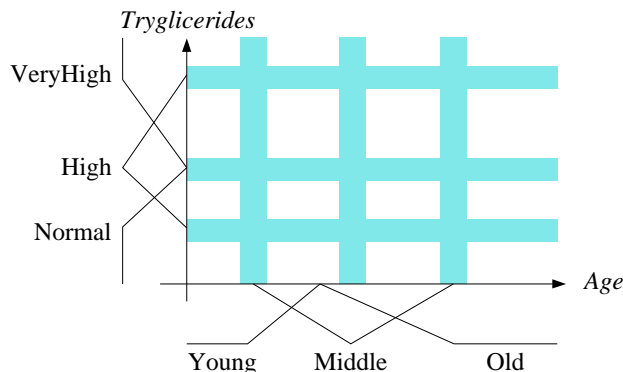


Fig. 5. Grid partition of the input space. In this example, two semantically correct input variables, each with three labels, divide the input space into a grid of nine regions.

To tackle these two problems some authors use “*don’t-care*” as a valid input label [7, 13, 14]. Variables in a given rule that are marked with a *don’t-care* label are considered as irrelevant. For example, in the rule base shown in Figure 6b the rule R_A :

if Triglycerides is *don’t-care* and Age is Old then Cardiac risk is Moderate,
replaces three rules (i.e., R_3 , R_6 , and R_9 in Figure 6a) and is interpreted as:

if Age is Old then Cardiac risk is Moderate.

Although *don’t-care* labels allow a reduction of rule-base size, their main advantage is the improvement of rule readability.

Default rule. In many cases, the behavior of a system exhibits only a few regions of interest, which can be described by a small number of rules (e.g., R_5 , R_A , and R_B in Figure 6b). To describe the rest of the input space, a simple default action, provided by the default rule, would be enough [22]. The example in Figure 6c shows that the default rule, named R_0 , covers the space of rules R_1 , R_2 , and R_4 . By definition, a default condition is true when all other rule conditions are false. The activation degree of the default rule, $\mu(R_0)$, is thus given by $\mu(R_0) = 1 - \max(\mu(R_i))$, where $\mu(R_i)$ is the activation degree of the i -th rule.

4 Fuzzy CoCo: A Cooperative Coevolutionary Approach to Fuzzy Modeling

As mentioned earlier, the accuracy-interpretability trade-off fuzzy modelers face implies the assumption of constraints acting on the parameter values, mainly on the membership-function shapes.

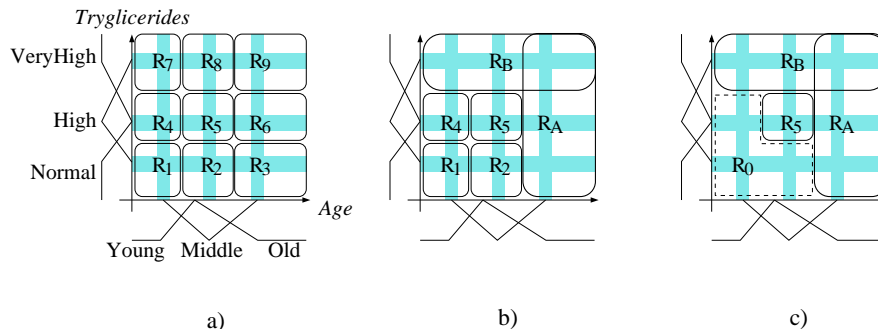


Fig. 6. Strategies to define the rule base. a) Fully defined rule base: the system contains all nine possible rules of the form **if** *Tryglicerides* **is** *label* **and** *Age* **is** *label* **then** *...*. b) *don't-care* labels: two rules, R_A and R_B , containing *don't-care* labels cover almost half of the input space. Variables marked with a *don't-care* label are considered as irrelevant for the rule in question. R_A is thus interpreted as **if** *Age* **is** *Old* **then** *...*, and R_B as **if** *Tryglicerides* **is** *VeryHigh* **then** *...*. c) Default rule, called here R_0 , defines a default action to be performed when none of the so-called active rules— R_5 , R_A , and R_B —apply. By definition, the activation of the (fuzzy) default rule is $\mu(R_0) = 1 - \max(\mu(R_i))$, with $i = \{1, 2, 3, \dots\}$. The rectangles denote the region where $\mu(R_i) > 0.5$.

In this section we present Fuzzy CoCo, a cooperative coevolutionary approach, capable of obtaining high-performance, interpretable systems. We have conceived Fuzzy CoCo to allow a high degree of freedom in the type of fuzzy systems it can design in order to allow the user to manage the trade-off between performance and interpretability. The next subsection briefly explains cooperative coevolution, after which Section 4.2 presents Fuzzy CoCo.

4.1 Cooperative coevolution

Coevolution refers to the simultaneous evolution of two or more species with coupled fitness. Such coupled evolution favors the discovery of complex solutions whenever complex solutions are required [12]. Simplistically speaking, one can say that coevolving species can either compete (e.g., to obtain exclusivity on a limited resource) or cooperate (e.g., to gain access to some hard-to-attain resource). Cooperative (also called symbiotic) coevolutionary algorithms involve a number of independently evolving species which together form complex structures, well-suited to solve a problem. The fitness of an individual depends on its ability to collaborate with individuals from other species. In this way, the evolutionary pressure stemming from the difficulty of the problem favors the development of cooperative strategies and individuals. Single-population evolutionary algorithms often perform poorly—manifesting stagnation, convergence to local optima, and computational costliness—when confronted with problems presenting one or more of the following features: (1) the sought-after solution is complex, (2) the

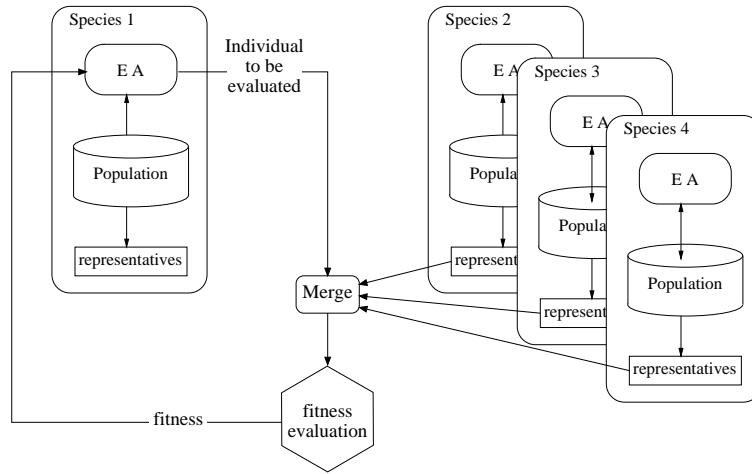


Fig. 7. Potter’s cooperative coevolutionary system. The figure shows the evolutionary process from the perspective of Species 1. The individual being evaluated is combined with one or more *representatives* of the other species so as to construct several solutions which are tested on the problem. The individual’s fitness depends on the quality of these solutions.

problem or its solution is clearly decomposable, (3) the genome encodes different types of values, (4) strong interdependencies among the components of the solution, (5) component-ordering drastically affects fitness. Cooperative coevolution effectively addresses these issues, consequently widening the range of applications of evolutionary computation. Potter [18, 19] developed a model in which a number of populations explore different decompositions of the problem. Below we detail this framework as it forms the basis of our own approach.

In Potter’s system, each species represents a subcomponent of a potential solution. Complete solutions are obtained by assembling *representative* members of each of the species (populations). The fitness of each individual depends on the quality of (some of) the complete solutions it participated in, thus measuring how well it cooperates to solve the problem. The evolution of each species is controlled by a separate, independent evolutionary algorithm. Figure 7 shows the general architecture of Potter’s cooperative coevolutionary framework, and the way each evolutionary algorithm computes the fitness of its individuals by combining them with selected representatives from the other species. A greedy strategy for the choice of representatives of a species is to use one or more of the fittest individuals from the last generation.

4.2 The coevolutionary algorithm

Fuzzy CoCo is a cooperative coevolutionary approach to fuzzy modeling wherein two coevolving species are defined: database (membership functions)

and rule base [13,14]. This approach is based primarily on the framework defined by Potter [18,19].

A fuzzy modeling process usually deals with the simultaneous search for operational and connective parameters (Table 1). These parameters provide an almost complete definition of the linguistic knowledge describing the behavior of a system, and the values mapping this symbolic description into a real-valued world (a complete definition also requires logical and structural parameters whose definition is best suited for human skills). Thus, fuzzy modeling can be thought of as two separate but intertwined search processes: (1) the search for the membership functions (i.e., operational parameters) that define the fuzzy variables, and (2) the search for the rules (i.e., connective parameters) used to perform the inference.

Fuzzy modeling presents several features discussed earlier which justify the application of a cooperative-coevolutionary approach: (1) The required solutions can be very complex, since fuzzy systems with a few dozen variables may call for hundreds of parameters to be defined. (2) The proposed solution—a fuzzy inference system—can be decomposed into two distinct components: rules and membership functions. (3) Membership functions are continuous and real-valued, while rules are discrete and symbolic. (4) These two components are interdependent because the membership functions defined by the first group of values are indexed by the second group (rules).

Consequently, in Fuzzy CoCo, the fuzzy modeling problem is solved by two coevolving, cooperating species. Individuals of the first species encode values which define completely all the membership functions for all the variables of the system. For example, with respect to the variable *Triglycerides* shown in Figure 3, this problem is equivalent to finding the values of P_1 , P_2 , and P_3 .

Individuals of the second species define a set of rules of the form:

if (v_1 **is** A_1) **and** ... **and** (v_n **is** A_n) **then** (*output is* C),

where the term A_v indicates which of the linguistic labels of fuzzy variable v is used by the rule. For example, a valid rule could contain the expression:

if ... **and** (*Triglycerides is High*) **and** ... **then** ...

which includes the membership function *High* whose defining parameters are contained in the first species (population).

The two evolutionary algorithms used to control the evolution of the two populations are instances of a simple genetic algorithm [21]. Figure 8 presents the Fuzzy CoCo algorithm in pseudo-code format. The genetic algorithms apply fitness-proportionate selection to choose the mating pool, and apply an elitist strategy with an elitism rate Er to allow some of the best individuals to survive into the next generation. Standard crossover and mutation operators are applied with probabilities P_c and P_m , respectively.

We introduced elitism to avoid the divergent behavior of Fuzzy CoCo, observed in preliminary trial runs. Non-elitist versions of Fuzzy CoCo tended to

```

begin Fuzzy CoCo
  g:=0
  for each species S
    Initialize populations  $P_S(0)$ 
    Evaluate population  $P_S(0)$ 
  end for
  while not done do
    for each species S
      g:=g+1
       $E_S(g)$  = elite-select  $P_S(g-1)$ 
       $P'_S(g)$  = select  $P_S(g-1)$ 
       $P''_S(g)$  = crossover  $P'_S(g)$ 
       $P'''_S(g)$  = mutate  $P''_S(g)$ 
       $P_S(g)$  =  $P'''_S(g)$  +  $E_S(g)$ 
      Evaluate population  $P_S(g)$ 
    end for
  end while
end Fuzzy CoCo

```

Fig. 8. Pseudo-code of Fuzzy CoCo. Two species coevolve in Fuzzy CoCo: membership functions and rules. The elitism strategy extracts E_S individuals to be reinserted into the population after evolutionary operators have been applied (i.e., selection, crossover, and mutation). Selection results in a reduced population $P'_S(g)$ (usually, the size of $P'_S(g)$ is $\|P'_S\| = \|P_S\| - \|E_S\|$). The line “Evaluate population $P_S(g)$ ” is elaborated in Figure 9.

lose the genetic information of good individuals found during evolution, consequently producing populations with mediocre individuals scattered throughout the search space. This is probably due to the relatively small size of the population which renders difficult the preservation of good solutions while exploring the search space. The introduction of simple elitism produces an undesirable effect on Fuzzy CoCo’s performance: populations converge prematurely even with reduced values of the elitism rate E_r . To offset this effect without losing the advantages of elitism, it was necessary to increase the mutation probability P_m by an order of magnitude so as to improve the exploration capabilities of the algorithm. (Increased mutation rates were also reported by Potter [18, 19] in his coevolutionary experiments.)

A more detailed view of the fitness evaluation process is depicted in Figure 9. An individual undergoing fitness evaluation establishes cooperations with one or more representatives of the other species, i.e., it is combined with individuals from the other species to construct fuzzy systems. The fitness value assigned to the individual depends on the performance of the fuzzy systems it participated in (specifically, either the average or the maximal value).

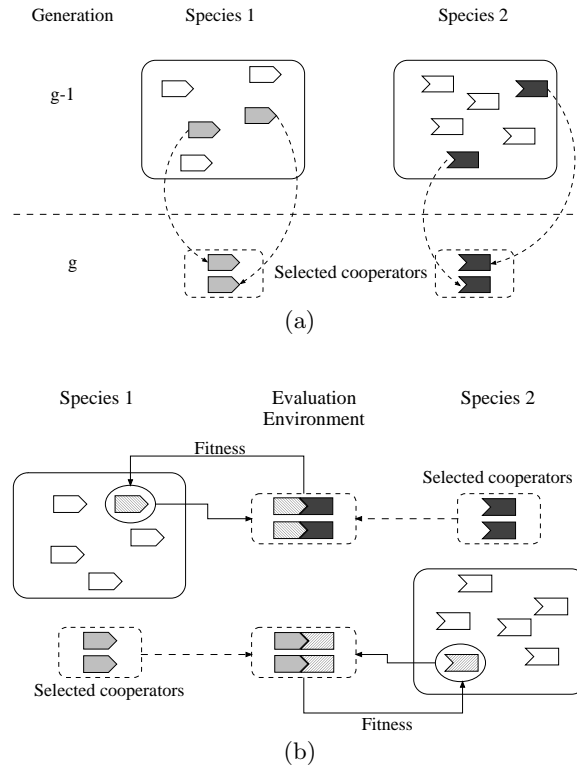


Fig. 9. Fitness evaluation in Fuzzy CoCo. (a) Several individuals from generation $g - 1$ of each species are selected according to their fitness to be the representatives of their species during generation g ; these representatives are called “cooperators.” (b) During the evaluation stage of generation g (after selection, crossover, and mutation—see Figure 8), individuals are combined with the selected cooperators of the other species to construct fuzzy systems of the other species to construct fuzzy systems. These systems are then evaluated on the problem domain and serve as a basis for assigning the final fitness to the individual being evaluated.

Representatives, called here *cooperators*, are selected both fitness-proportionally and randomly from the previous generation since they have already been assigned a fitness value (see Figure 8). In Fuzzy CoCo, N_{cf} cooperators are selected according to their fitness, usually the fittest individuals, thus favoring the exploitation of known good solutions. The other N_{cr} cooperators are selected randomly from the population to represent the diversity of the species, maintaining in this way exploration of the search space.

Table 2. Variables corresponding to a patient’s clinical data.

v_1	Age	[28-82] years	
v_2	Menstrual history	1	Premenopausal
		2	Postmenopausal
v_3	Family history	1	None
		2	Second familiar
		3	First familiar
		4	Contralateral
		5	Homolateral

5 Application Example: The Catalonia Online Breast-Cancer Risk Assessor

Mammography remains the principal technique for detecting breast cancer. Its undoubtable value in reducing mortality notwithstanding, mammography’s positive predictive value (PPV) is low: only between 15 and 35% of mammographic-detected lesions are cancerous [11, 17]. The remaining 65 to 85% of biopsies, besides being costly and time-consuming, cause understandable stress on women facing the doubt of cancer. A computer-based tool that assists radiologists during mammographic interpretation would contribute to increasing the PPV of biopsy recommendations.

5.1 The database

The *Catalonia Mammography Database*, which is the object of our study, was collected at the Duran y Reynals hospital in Barcelona. It consists of 15 input attributes and a diagnostic result indicating whether or not a carcinoma was detected after a biopsy. The 15 input attributes include three clinical characteristics (Table 2) and two groups of six radiologic features, according to the type of lesion found in the mammography: mass or microcalcifications (Table 3).

A radiologist fills out a reading form for each mammography, assigning values for the clinical characteristics and for one of the groups of radiologic features. Then, the radiologist interprets the case using a five-point scale: (1) benign; (2) probably benign; (3) indeterminate; (4) probably malignant; (5) malignant. According to this interpretation a decision is made on whether to practice a biopsy on the patient or not. The Catalonia database contains data corresponding to 227 cases. Each case is examined by three different readers—for a total of 681 readings—but only diverging readings are kept. The actual number of readings in the database is 516, among which 187 are positive (malignant) cases and 329 are negative (benign) cases.

Table 3. Variables corresponding to radiologic features. There are two groups of variables used to describe the mammographic existence of microcalcifications and masses (left and right columns respectively).

Microcalcifications		Mass	
v_4	Disposition	v_{10}	Morphology
1	Round	1	Oval
2	Indefinite	2	Round
3	Triangular or Trapezoidal	3	Lobulated
4	Linear or Ramified	4	Polilobulated
		5	Irregular
v_5	Other signs of group form	v_{11}	Margins
1	None	1	Well delimited
2	Major axis in direction of nipple	2	Partially well delimited
3	Undulating contour	3	Poorly delimited
4	Both previous	4	Spiculated
v_6	Maximum diameter of group [3-120] mm	v_{12}	Density greater than parenchyma
v_7	Number	1	Not
1	<10	2	Yes
2	10 to 30	v_{13}	Focal distortion
3	>30	1	Not
		2	Yes
v_8	Morphology	v_{14}	Focal asymmetry
1	Ring shaped	1	Not
2	Regular sharp-pointed	2	Yes
3	Too small to determine	v_{15}	Maximum diameter [5-80] mm
4	Irregular sharp-pointed		
5	Vermicular, ramified		
v_9	Size irregularity		
1	Very regular		
2	Sparingly regular		
3	Very irregular		

5.2 Proposed system

The solution scheme we propose is depicted in Figure 10. It consists of a reading form, a fuzzy subsystem, and a threshold unit. Based on the 15 input attributes collected with the reading form, the fuzzy system computes a continuous appraisal value of the malignancy of a case. The threshold unit then outputs a biopsy recommendation according to the fuzzy system's output. The threshold value used in this system is 3, which corresponds to the "indeterminate" diagnostic. Fuzzy CoCo is applied to design the fuzzy system in charge of appraising malignancy.

5.3 Fuzzy-parameter setup

We used prior knowledge about the Catalonia database to guide our choice of fuzzy parameters. In addition, we took into account the interpretability

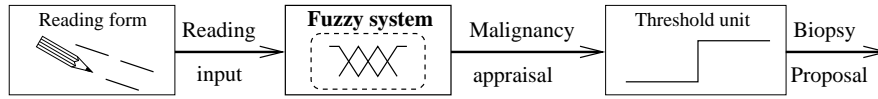


Fig. 10. Proposed system. Note that the fuzzy system displayed in the middle is in fact the entire fuzzy inference system of Figure 2.

criteria presented in Section 3 to define constraints on the fuzzy parameters. Referring to Table 1, we delineate below the fuzzy system’s set-up:

- Logical parameters: singleton-type fuzzy systems; min-max fuzzy operators; orthogonal, trapezoidal input membership functions (see Figure 11); weighted-average defuzzification.
- Structural parameters: two input membership functions (*Low* and *High*; see Figure 11); two output singletons (*benign* and *malignant*); a user-configurable number of rules. The relevant variables are one of Fuzzy CoCo’s evolutionary objectives.

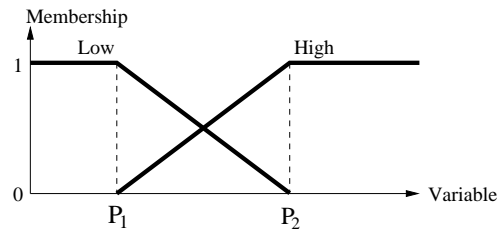


Fig. 11. Input fuzzy variables. Each fuzzy variable has two possible fuzzy values labeled **Low** and **High**, and orthogonal membership functions, plotted above as degree of membership versus input value. P_1 and P_2 define the membership-function apices.

- Connective parameters: the antecedents and the consequent of the rules are searched by Fuzzy CoCo. The algorithm also searches for the consequent of the default rule. All rules have unitary weight.
- Operational parameters: the input membership-function values are to be found by Fuzzy CoCo. For the output singletons we used the values 1 and 5, for *benign* and *malignant*, respectively.

5.4 Genome encodings

Fuzzy CoCo thus searches for four parameters: input membership-function values, relevant input variables, and antecedents and consequents of rules. To encode these parameters into both species’ genomes, which together describe an entire fuzzy system, it is necessary to take into account the heterogeneity of the input variables as explained below.

Table 4. Genome encoding of parameters for membership-function species. Genome length is 106 bits.

Variable type	Qty	Parameters	Bits	Total bits
Continuous	3	2	7	42
Discrete	8	2	4	64
Total Genome Length				106

- Species 1: Membership functions. The fifteen input variables ($v_1 - v_{15}$) present three different types of values: continuous (v_1, v_6 , and v_{15}), discrete ($v_3 - v_5$ and $v_7 - v_{11}$), and binary (v_2 and $v_{12} - v_{14}$). It is not necessary to encode membership functions for binary variables as they can only take on two values. The membership-function genome encodes the remaining 11 variables—eight continuous and three discrete—each with two parameters P_1 and P_2 , defining the membership-function apices (Figure 11). Table 4 delineates the parameters encoding the membership-function genome.
- Species 2: Rules. The i -th rule has the form:
if (v_1 **is** A_1^i) **and** \dots **and** (v_{15} **is** A_{15}^i) **then** (*output is* C^i),
where A_j^i can take on the values: 1 (*Low*), 2 (*High*), or 0 or 3 (*don't-care*). C^i can take on the values: 1 (*benign*) or 2 (*malignant*). However, as mentioned before, each database case presents three clinical characteristics and six radiologic features according to the type of lesion found: mass or microcalcifications (note that only a few special cases contain data for both groups). To take advantage of this fact, the rule-base genome encodes, for each rule, 11 parameters: the three antecedents of the clinical-data variables, the six antecedents of one radiological-feature group, an extra bit to indicate whether the rule applies for mass or microcalcifications, and the rule consequent. Furthermore, the genome contains an additional parameter corresponding to the consequent of the default rule. Relevant variables are searched for implicitly by allowing the algorithm to choose non-existent membership functions as valid antecedents ($A_j^i = 0$ or $A_j^i = 3$); in such case the respective variable is considered irrelevant, and removed from the rule. Table 5 delineates the parameters encoding the rules genome.

Table 5. Genome encoding of parameters for rules species. Genome length is $20 \times N_r + 1$ bits, where N_r denotes the number of rules.

Parameters	Qty	Bits	Total bits
Clinical antecedents	$3 \times N_r$	2	$6 \times N_r$
Radiologic antecedents	$6 \times N_r$	2	$12 \times N_r$
Rule-type selector	N_r	1	N_r
Consequents	$N_r + 1$	1	$N_r + 1$
Total Genome Length			$20 \times N_r + 1$

5.5 Evolutionary parameters

Table 6 delineates values and ranges of values of the evolutionary parameters. The algorithm terminates when the maximum number of generations, G_{max} , is reached (we set $G_{max} = 700 + 200 \times N_r$, i.e., dependent on the number of rules used in the run), or when the increase in fitness of the best individual over five successive generations falls below a certain threshold (10^{-4} in our experiments). Note that mutation rates are relatively higher than those used with a simple genetic algorithm.

Table 6. Fuzzy CoCo set-up.

Parameter	Values
Population size N_p	90
Maximum generations G_{max}	$700 + 200N_r$
Crossover probability P_c	1
Mutation probability P_m	{0.005, 0.01}
Elitism rate E_r	{0.1, 0.2}
“Fit” cooperators N_{cf}	1
Random cooperators N_{cr}	1

5.6 Fitness function

As the main function of the proposed system is the assessment of a medical diagnosis, our fitness definition takes into account medical diagnostic criteria. The most commonly employed measures of the validity of diagnostic procedures are the sensitivity and specificity, the likelihood ratios, the predictive values, and the overall classification (accuracy) [2]. Table 7 provides expressions for four of these measures which are important for evaluating the performance of our systems. Three of them are used in the fitness function, the last one is used in Section 6 to support the analysis of the results. Besides these criteria, the fitness function provides extra selective pressure based on two syntactic criteria: simplicity and readability (see Section 3).

Our fitness function combines the following five criteria: 1) F_{sens} : sensitivity, or true-positive ratio, computed as the percentage of positive cases correctly classified; 2) F_{spec} : specificity, or true-negative ratio, computed as the percentage of negative cases correctly classified (note that there is usually an important trade-off between sensitivity and specificity which renders difficult the satisfaction of both criteria); 3) F_{acc} : classification performance, computed as the percentage of cases correctly classified; 4) F_r : rule-base size fitness, computed as the percentage of unused rules (i.e., the number of rules that are never fired and can thus be removed altogether from the system); and 5) F_v : rule-length fitness, computed as the average number of *don't-care* antecedents (i.e., unused variables) per rule. This order also represents their relative importance in the final fitness function, from most important (F_{sens}) to least important (F_r and F_v).

Table 7. Diagnostic performance measures. The values used to compute the expressions are: True positive (TP): the number of positive cases correctly detected, true negative (TN): the number of negative cases correctly detected, false positive (FP): the number of negative cases diagnosed as positive, and false negative (FN): the number of positive cases diagnosed as negative.

Sensitivity	$\frac{TP}{TP + FN}$
Specificity	$\frac{TN}{TN + FP}$
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Positive predictive value (PPV)	$\frac{TP}{TP + FP}$

The fitness function is computed in three steps—basic fitness, accuracy reinforcement, and size reduction—as explained below:

1. Basic fitness. Based on sensitivity and specificity, it is given by

$$F_1 = \frac{F_{sens} + \alpha F_{spec}}{1 + \alpha},$$

where the weight factor $\alpha = 0.3$ reflects the greater importance of sensitivity.

2. Accuracy reinforcement. Given by

$$F_2 = \frac{F_1 + \beta F'_{acc}}{1 + \beta},$$

where $\beta = 0.01$. $F'_{acc} = F_{acc}$ when $F_{acc} > 0.7$; $F'_{acc} = 0$ elsewhere. This step slightly reinforces the fitness of high-accuracy systems.

3. Size reduction. Based on the size of the fuzzy system, it is given by

$$F = \frac{F_2 + \gamma F_{size}}{1 + 2\gamma},$$

where $\gamma = 0.01$. $F_{size} = (F_r + F_v)$ if $F_{acc} > 0.7$ and $F_{sens} > 0.98$; $F_{size} = 0$ elsewhere. This step rewards top systems exhibiting a concise rule set, thus directing evolution toward more interpretable systems.

6 Results

This section describes the results obtained when applying the methodology described in Section 5. We first delineate the success statistics relating to the

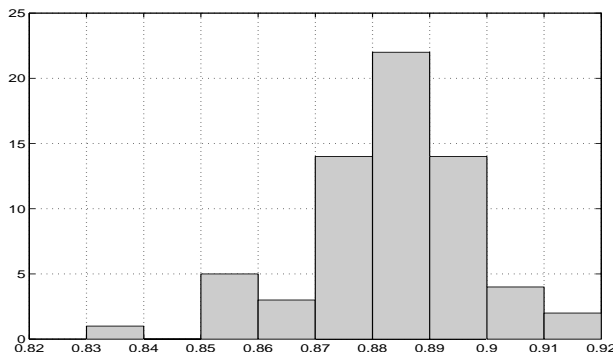


Fig. 12. Summary of results of 65 evolutionary runs. The histogram depicts the number of systems exhibiting a given fitness value at the end of the evolutionary run. The fitness considered is that of the best individual of the run.

evolutionary algorithm. Then, we present the diagnostic performance of two selected evolved fuzzy systems that exemplify our approach.

A total of 65 evolutionary runs were performed, all of which found systems whose fitness exceeds 0.83. In particular, considering the best individual per run (i.e., the evolved system with the highest fitness value), 42 runs led to a fuzzy system whose fitness exceeds 0.88, and of these, 6 runs found systems whose fitness exceeds 0.9; these results are summarized in Figure 12.

Table 8 shows the results of the best systems obtained. The maximum number of rules per system was fixed at the outset to be between ten and twenty-five.

Table 8. Results of the best systems evolved. Results are divided into four classes, in accordance with the maximum number of rules-per-system, going from 10-rule systems to 25-rule ones. Shown below are the fitness values of the top systems as well as the average fitness per class, along with the number of rules which effectively used by the system (R_{eff}) and the average number of variables per rule (V_r).

Maximum number of rules	Best individual			Average per class		
	Fitness	R_{eff}	V_r	Fitness	R_{eff}	V_r
10	0.8910	9	2.22	0.8754	9.17	2.52
15	0.8978	12	2.50	0.8786	12.03	2.62
20	0.9109	17	2.41	0.8934	14.15	2.59
25	0.9154	17	2.70	0.8947	15.78	2.76

As mentioned before, our fitness function includes two syntactic criteria to favor the evolution of good diagnostic systems exhibiting interpretable rule bases (see Section 3). Concerning the simplicity of the rule base, rules that are encoded in a genotype but that never fire are removed from the phenotype (the final system), rendering it more interpretable. Moreover, to improve readability, the rules are allowed (and encouraged) to contain *don't-*

care conditions. The relatively low values of R_{eff} and V_r in Table 8 confirm the reinforced interpretability of the evolved fuzzy systems.

Table 9 shows the diagnostic performance measures of two selected evolved systems. The first system, which is the top one over all 65 Fuzzy CoCo runs, is a 17-rule system exhibiting a sensitivity of 99.47% (i.e., it detects all but one of the positive cases), and a specificity of 68.69% (i.e., 226 of the 329 negative cases are correctly detected as benign). The second system is the best found when searching for ten-rule systems. The sensitivity and the specificity of this 9-rule system are, respectively, 98.40% and 64.13%. As mentioned in Section 5, the usual positive predictive value (PPV) of mammography ranges between 15 and 35%. As shown in Table 9, Fuzzy CoCo increases this value beyond 60%—64.36% for the 17-rule system—while still exhibiting a very high sensitivity.

Table 9. Diagnostic performance of two selected evolved systems. Shown below are the sensitivity, the specificity, the accuracy, and the positive predictive value (PPV) of two selected evolved systems. In parentheses are the values, expressed in number of cases, leading to such performance measures. The 17-rule system is the top system. The 9-rule system is the best found when searching for ten-rule systems.

	17-rule	9-rule
Sensitivity	99.47% (186/187)	98.40% (184/187)
Specificity	68.69% (226/329)	64.13% (211/329)
Accuracy	79.84% (412/516)	76.55% (395/516)
PPV	64.36% (186/289)	60.93% (184/302)

7 Concluding Remarks

We presented Fuzzy CoCo, a fuzzy modeling technique based on cooperative coevolution, along with an application to breast-cancer diagnosis. In fuzzy modeling, the interpretability-accuracy trade-off is of crucial import, imposing several conditions on the input and output membership functions as well as on the rule definition. In evolutionary fuzzy modeling these conditions are translated both into restrictions on the choice of fuzzy parameters and into criteria included in the fitness function. We designed Fuzzy CoCo to be highly configurable, thus facilitating the management of the interpretability-accuracy trade-off.

Applying Fuzzy CoCo to breast-cancer diagnosis we concentrated on increasing the interpretability of solutions by the mechanisms proposed earlier, obtaining excellent results. We note, however, that the consistency of the entire rule base and its compatibility with the specific domain knowledge can only be assessed by further interaction with medical experts (radiologists, oncologists).

In the future we wish to test some novel ideas that could improve Fuzzy CoCo: 1) Coevolution of $N_r + 1$ species, one species for each of the N_r rules in addition to the membership-function species. 2) Coexistence of several Fuzzy CoCo instances (each one set to evolve systems with a different number of rules), permitting migration of individuals among them so as to increase the exploration and the diversity of the search process. 3) Apply the strategy of rising and death of species proposed by Potter and DeJong [19] in order to evolve systems with variable numbers of rules and membership functions.

References

1. J. T. Alander. An indexed bibliography of genetic algorithms with fuzzy logic. In Pedrycz [15], pages 299–318.
2. H. Brenner. Measures of differential diagnostic value of diagnostic procedures. *Journal of Clinical Epidemiology*, 49(12):1435–1439, December 1996.
3. O. Cordón, F. Herrera, and M. Lozano. On the combination of fuzzy logic and evolutionary computation: A short review and bibliography. In Pedrycz [15], pages 33–56.
4. J. Espinosa and J. Vandewalle. Constructing fuzzy models with linguistic integrity from numerical data-AFRELI algorithm. *IEEE Transactions on Fuzzy Systems*, 8(5):591, October 2000.
5. S. Guillaume. Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Transactions on Fuzzy Systems*, 9(3):426–443, June 2001.
6. F. Herrera, M. Lozano, and J. L. Verdegay. Generating fuzzy rules from examples using genetic algorithms. In B. Bouchon-Meunier, R. R. Yager, and L. A. Zadeh, editors, *Fuzzy Logic and Soft Computing*, pages 11–20. World Scientific, 1995.
7. H. Ishibushi, T. Nakashima, and T. Murata. Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics*, 29(5):601–618, October 1999.
8. C. L. Karr. Genetic algorithms for fuzzy controllers. *AI Expert*, 6(2):26–33, February 1991.
9. C. L. Karr, L. M. Freeman, and D. L. Meredith. Improved fuzzy process control of spacecraft terminal rendezvous using a genetic algorithm. In G. Rodriguez, editor, *Proceedings of Intelligent Control and Adaptive Systems Conference*, volume 1196, pages 274–288. SPIE, February 1990.
10. Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Heidelberg, third edition, 1996.
11. S. G. Orel, N. Kay, C. Reynolds, and D. C. Sullivan. Bi-rads categorization as a predictor of malignancy. *Radiology*, 211(3):845–880, June 1999.
12. J. Paredis. Coevolutionary computation. *Artificial Life*, 2:355–375, 1995.
13. C. A. Peña-Reyes and M. Sipper. A fuzzy-genetic approach to breast cancer diagnosis. *Artificial Intelligence in Medicine*, 17(2):131–155, October 1999.
14. C. A. Peña-Reyes and M. Sipper. Fuzzy CoCo: A cooperative-coevolutionary approach to fuzzy modeling. *IEEE Transactions on Fuzzy Systems*, 9(5):727–737, October 2001.

15. W. Pedrycz, editor. *Fuzzy Evolutionary Computation*. Kluwer Academic Publishers, 1997.
16. W. Pedrycz and J. Valente de Oliveira. Optimization of fuzzy models. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 26(4):627–636, August 1996.
17. L. Porta, R. Villa, E. Andia, and E. Valderrama. Infraclinic breast carcinoma: Application of neural networks techniques for the indication of radioguided biopsias. In J. Mira, R. Moreno-Díaz, and J. Cabestany, editors, *Biological and Artificial Computation: From Neuroscience to Technology*, volume 1240 of *Lecture Notes in Computer Science*, pages 978–985. Springer, 1997.
18. M. A. Potter. *The Design and Analysis of a Computational Model of Cooperative Coevolution*. PhD thesis, George Mason University, 1997.
19. M. A. Potter and K. A. DeJong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29, spring 2000.
20. J. Valente de Oliveira. Semantic constraint for membership function optimization. *IEEE Transactions on Systems, Man, and Cybernetics. Part A: Systems and Humans*, 29(1):128–138, January 1999.
21. M. D. Vose. *The Simple Genetic Algorithm*. MIT Press, Cambridge, MA, August 1999.
22. P. Vuorimaa. Fuzzy self-organizing map. *Fuzzy Sets and Systems*, 66:223–231, 1994.
23. R. R. Yager and D. P. Filev. *Essentials of Fuzzy Modeling and Control*. John Wiley & Sons., New York, 1994.
24. R. R. Yager and L. A. Zadeh. *Fuzzy Sets, Neural Networks, and Soft Computing*. Van Nostrand Reinhold, New York, 1994.
25. L. A. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning. *Information Science*, Part I, 8:199-249; Part II, 8:301-357; Part III, 9:43-80., 1975.