Applying Fuzzy CoCo to Breast Cancer Diagnosis

Carlos Andrés Peña-Reyes
Logic Systems Laboratory
Swiss Federal Institute of Technology Lausanne
CH-1015 Lausanne, Switzerland
Carlos.Pena@epfl.ch

Moshe Sipper
Logic Systems Laboratory
Swiss Federal Institute of Technology Lausanne
CH-1015 Lausanne, Switzerland
Moshe.Sipper@epfl.ch

Abstract- Coevolutionary algorithms have received increased attention in the past few years within the domain of evolutionary computation. In this paper, we combine the search power of coevolutionary computation with the expressive power of fuzzy systems, introducing a novel algorithm, *Fuzzy CoCo:* **Fuzzy Cooperative Coevolution. We demonstrate the efficacy of Fuzzy CoCo by applying it to a hard, real-world problem—breast cancer diagnosis—obtaining the best results to date while expending less computational effort than formerly.**

1 Introduction

In recent years the natural phenomenon of coevolution—the simultaneous, coupled evolution of two or more species—has been explored by evolutionary-computation practitioners, who introduced the notion of *coevolutionary algorithms*. It has been shown that, for certain problem domains, coevolution produces better solutions while incurring a lower computational cost.

We explore herein the application of coevolution to the design of fuzzy systems, introducing *Fuzzy CoCo:* Fuzzy Cooperative Coevolution. We demonstrate the efficacy of Fuzzy CoCo by applying it to a hard, real-world problem: breast cancer diagnosis.

This paper is organized as follows: in the next section we present evolutionary fuzzy modeling, namely, the use of evolutionary-computation techniques in the design of fuzzy systems. Section 3 provides a brief overview of cooperative coevolution and summarizes the (relatively few) works in this novel domain. Section 4 presents Fuzzy CoCo, our cooperative coevolutionary approach to fuzzy modeling. In Section 5 we describe how Fuzzy CoCo is applied to the evolution of breast cancer diagnostic systems, followed by a presentation of our results in Section 6. Finally, we present concluding remarks in Section 7.

2 Evolutionary Fuzzy Modeling

Fuzzy logic is a computational paradigm that provides a mathematical tool for representing and manipulating information in a way that resembles human communication and reasoning processes [22]. It is based on the assumption that, in contrast to Boolean logic, a statement can be *partially* true (or false), and composed of imprecise concepts. For example, the expression "I live near Geneva," where the fuzzy value "near" applied to the fuzzy variable "distance", in addition to being imprecise is subject to interpretation. A *fuzzy variable* (also called a *linguistic variable*; see Figure 1) is characterized by its name tag, a set of *fuzzy values* (also known as *linguistic values* or *labels*), and the membership functions of these labels; these latter assign a membership value $\mu_{label}(u)$ to a given real value $u \in \Re$, within some predefined range (known as the universe of discourse).

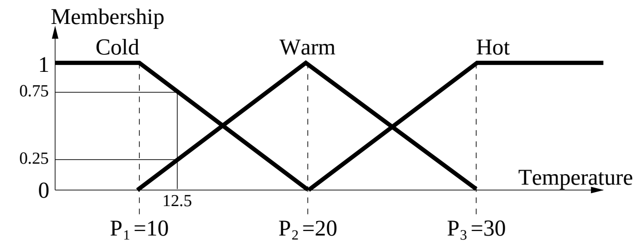

Figure 1 Example of a fuzzy variable, *Temperature* has three possible fuzzy values, labeled **Cold**, **Warm**, and **Hot**, and orthogonal membership functions, plotted above as degree of membership versus input values. The values P_i define the membership functions. The orthogonality condition means that the sum of all membership functions at any point is one. In the figure, an example value 12.5 is assigned the membership values $\mu_{Cold}(12.5) = 0.75$, $\mu_{Warm}(12.5) = 0.25$, and $\mu_{Hot}(12.5) = 0$ (as can be seen the sum $\mu_{Cold}(12.5) + \mu_{Warm}(12.5) + \mu_{Hot}(12.5) = 1$).

A *fuzzy inference system* is a rule-based system that uses fuzzy logic, rather than Boolean logic, to reason about data [22]. Its basic structure includes four main components, as depicted in Figure 2 [13]: (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 decision-making process is performed by the inference engine using the rules contained in the rule base. These fuzzy rules define the connection between input and output fuzzy variables. A fuzzy rule has the form:

if $antecedent$ **then** $consequent$,

where $antecedent$ is a fuzzy-logic expression, composed of one or more simple fuzzy expressions connected by fuzzy operators, and $consequent$ is an expression that assigns fuzzy

Table 1 Parameter classification of fuzzy inference systems [13].

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

values to the output variables. The inference engine evaluates all the rules in the rule base and combines the weighted consequents of all relevant rules into a single fuzzy set using the *aggregation* operation.

Fuzzy modeling is the task of identifying the parameters of a fuzzy inference system so that a desired behavior is attained [21]. 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. 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.

The parameters of fuzzy inference systems can be classified into four categories (Table 1) [13]: logical, structural, connective, and operational. Generally speaking, this order also represents their relative influence on performance, from most influential (logical) to least influential (operational). In fuzzy modeling, 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.

Fuzzy modeling can be considered as an optimization process where part or all of the parameters of a fuzzy system constitute the search space. 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).

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. Works investigating the application of evolutionary techniques in the domain of fuzzy modeling had first appeared about a decade ago [4, 5]. 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, 2].

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-parameters 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].

Knowledge tuning (operational parameters). The evolutionary algorithm is used to tune the knowledge contained in the fuzzy system by finding membership function values.

Behavior learning (connective parameters). In this approach, one supposes that extant knowledge is sufficient in order to define the membership functions. The evolutionary algorithm is used to find either the rule consequents, or an adequate subset of rules to be included in the rule base.

Structure learning (structural parameters). In this approach, 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. 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 [8]. A more recent method has been proposed specifically for fuzzy modeling: the iterative rule learning approach [3]. These three approaches are presented 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.

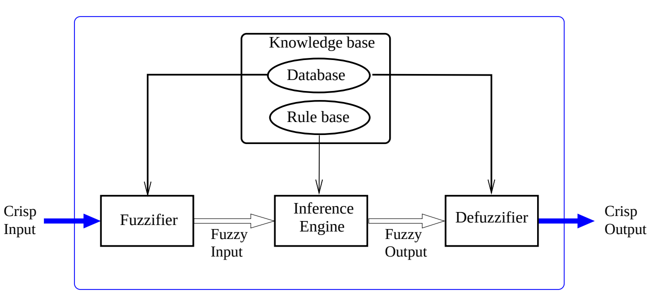

Figure 2 Basic structure of a fuzzy inference system.

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 with similar antecedents), 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 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 [10]. 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). In a competitive coevolutionary algorithm the fitness of an individual is based on direct competition with individuals of other species, which in turn evolve separately in their own populations. Increased fitness of one of the species implies a diminution in the fitness of the other species. This evolutionary pressure tends to produce new strategies in the populations involved so as to maintain their chances of survival. This "arms race" ideally increases the capabilities of each species until they reach an optimum. For further details about competitive coevolution, the reader is referred to [17].

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 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) components-ordering drastically affects fitness. Cooperative coevolution addresses effectively these issues, consequently widening the range of applications of evolutionary computation.

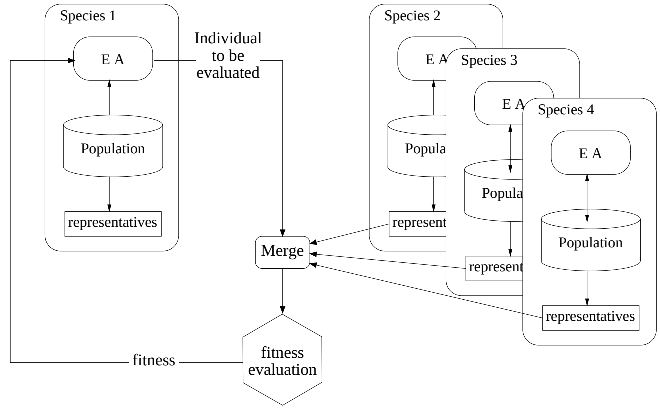

Figure 3 Potter's cooperative coevolutionary system. The figure shows the evolutionary process from the perspective of Species 1. The individual being evaluated is combined with one or more *representatives* of the other species so as to construct several solutions which are tested on the problem. The individual's fitness depends on the quality of these solutions.

Paredis [10] applied cooperative coevolution to problems which involved finding simultaneously the values of a solution and their adequate order. In his approach, a population of solutions coevolves alongside a population of permutations performed on the genotypes of the solutions. Moriarty [9] used a cooperative coevolutionary approach to evolve neural networks. Each individual in one species corresponds to a single hidden neuron of a neural network and its connections with the input and output layers. This population coevolves alongside a second one whose individuals encode sets of hidden neurons (i.e., individuals from the first population) forming a neural network. Potter [16] 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 3 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.

Results presented by Potter show that his approach addresses adequately issues like problem decomposition and interdependencies between subcomponents [16]. The cooperative coevolutionary approach performs as good as, and sometimes better than, single-population evolutionary algorithms. Finally, cooperative coevolution usually requires less computation than single-population evolution as the populations involved are smaller and the convergence measured as number of generations is faster.

4 Fuzzy CoCo: A Cooperative Coevolutionary Approach to Fuzzy Modeling

Fuzzy CoCo is a Cooperative Coevolutionary approach to fuzzy modeling where two coevolving species are defined: database (membership functions) and rule base. This approach is based primarily on the framework defined by Potter [16] (Section 3).

A fuzzy modeling process has usually to deal 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 structural parameters whose definition is best suited for human skills). Thus, fuzzy modeling can be thought of as two separate but intertwined searching 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 in Section 3 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 represented by continuous, real values, while rules are represented by discrete, symbolic values. (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 cooperative 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 $Temperature$ shown in Figure 1, 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 one of the linguistic labels of fuzzy variable v is used by the rule. For example, a valid rule could contain the expression

if … **and** ($Temperature$ **is** $Warm$) **and** … **then** …

which includes the membership function $Warm$ whose defining parameters are contained in the first species.

The two evolutionary algorithms used to control the evolution of the two populations are instances of a simple genetic algorithm. Figure 4 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.

```
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 GA
```

Figure 4 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 5.

We introduced elitism to avoid the divergent behavior of Fuzzy CoCo, observed in our preliminary trial runs. Non-elitist versions of Fuzzy CoCo often tend to 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 populations which renders difficult the preservation of good solutions while exploring the search space. The introduction of simple elitism produces an undesirable effect on the Fuzzy CoCo 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 is necessary to increase the mutation probability P_m by an order of magnitude (Table 3) so as to improve the exploration capabilities of the algorithm.

A more detailed view of the fitness evaluation process is

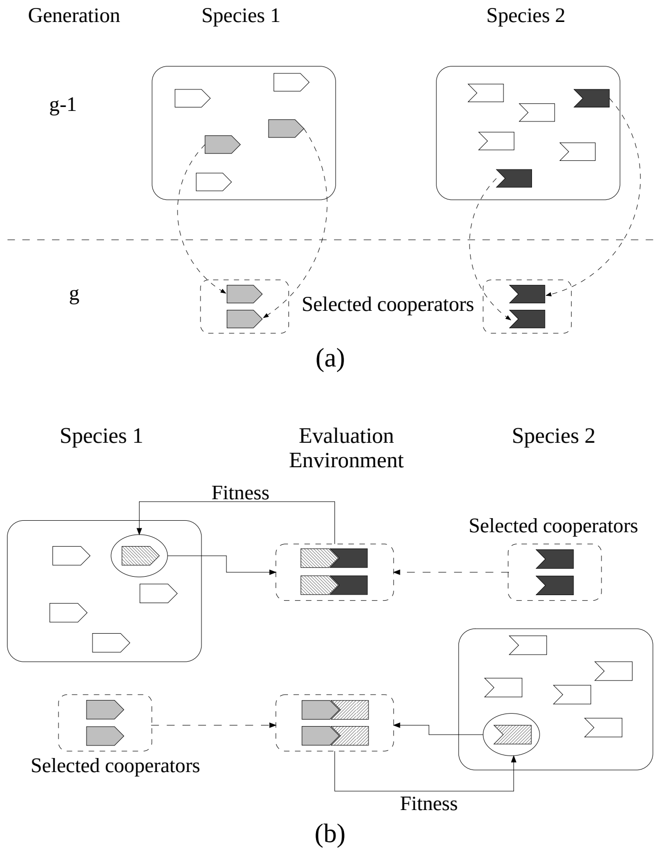

Figure 5 Fitness evaluation in Fuzzy CoCo. (a) Several individuals from generation $g-1$ of each species are selected according to their fitness to be the representatives of their species during generation g; these representatives are called "cooperators." (b) During the evaluation stage of generation g (after selection, crossover, and mutation—see Figure 4), individuals are combined with the selected cooperators 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.

depicted in Figure 5. 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).

Representatives, called here *cooperators*, are selected both, fitness-proportionally and randomly from the last generation since they have already been assigned a fitness value (see Figure 4). In Fuzzy CoCo, N_{cf} cooperators are probabilistically 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.

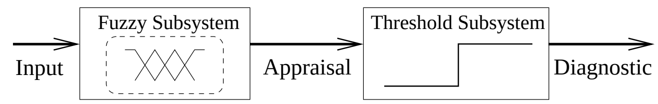

Figure 6 Proposed diagnosis system. Note that the fuzzy subsystem displayed to the left is in fact the entire fuzzy inference system of Figure 2.

5 Applying Fuzzy CoCo to Breast Cancer Diagnosis

The Wisconsin Breast Cancer Diagnosis (WBCD) problem involves classifying a presented case as to whether it is benign or malignant. It admits a relatively high number of variables and consequently a large search space. The WBCD database [7] consists of nine visually assessed characteristics obtained from fine needle aspirates[1] of breast masses, each of which is ultimately represented as an integer value between 1 and 10. The measured variables are as follows: (1) Clump Thickness (v_1); (2) Uniformity of Cell Size (v_2); (3) Uniformity of Cell Shape (v_3); (4) Marginal Adhesion (v_4); (5) Single Epithelial Cell Size (v_5); (6) Bare Nuclei (v_6); (7) Bland Chromatin (v_7); (8) Normal Nucleoli (v_8); and (9) Mitosis (v_9).

The diagnostics in the WBCD database were furnished by specialists in the field. The database itself consists of 683 cases, with each entry representing the classification for a certain ensemble of measured values:

$case$	v_1	v_2	v_3	$\ldots$	v_9	$diagnostic$
1	5	1	1	$\ldots$	1	$Benign$
2	5	4	4	$\ldots$	1	$Benign$
:	:	:	:		:	:
683	4	8	8	$\ldots$	1	$Malignant$

There are several studies based on this database. Among them, researchers having interpretability of the diagnostic as a prior objective, have applied the method of extracting Boolean rules from neural networks [18–20]. Our own work on the evolution of fuzzy rules for the WBCD problem showed that it is possible to obtain diagnostic systems exhibiting high performance, coupled with interpretability and a confidence measure [11–13]. In our previous work we used a simple genetic algorithm rather than Fuzzy CoCo.

The solution scheme we propose for the WBCD problem is depicted in Figure 6. It consists of a fuzzy system and a threshold unit. The fuzzy system computes a continuous appraisal value of the malignancy of a case, based on the input values. The threshold unit then outputs a *benign* or *malignant* diagnostic according to the fuzzy system's output.

Our previous knowledge about the WBCD problem represents valuable information to be used for our choice of fuzzy parameters. When defining our setup we took into consideration the following three results concerning the composition of potential high-performance systems: (1) small number of rules; (2) small number of variables; and (3) monotonicity of the input variables [13]. Some fuzzy models forgo inter-

[1] Fine needle aspiration is an outpatient procedure that involves using a small-gauge needle to extract fluid directly from a breast mass [6].

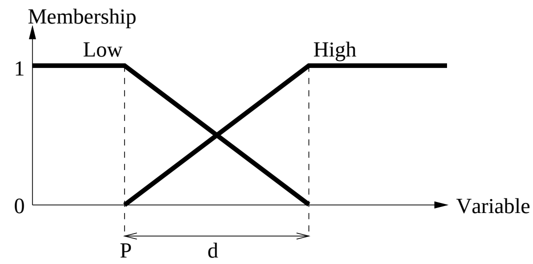

Figure 7 Fuzzy variables for the WBCD problem. All the variables have two labels: **Low** and **High**, and orthogonal membership functions. *P* and *d* define the start point and the length of membership-function edges, respectively.

pretability in the interest of improved performance. Where medical diagnosis is concerned [14], interpretability—also called linguistic integrity—is the major advantage of fuzzy systems. This motivated us to take into account the following five semantic criteria, defining constraints on the fuzzy parameters [13]: (1) distinguishability; (2) justifiable number of elements; (3) coverage; (4) normalization; and (5) orthogonality.

Referring to Table 1, and taking into account the above criteria, we delineate below the fuzzy system setup:

- Logical parameters: singleton-type fuzzy systems; min-max fuzzy operators; orthogonal, trapezoidal input membership functions; weighted-average defuzzification.
- Structural parameters: two input membership functions (*Low* and *High*); two output singletons (*benign* and *malignant*); a user-configurable number of rules. The relevant variables are one of Fuzzy CoCo's objectives.
- 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 which plays the role of an `else` condition. 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 2 and 4, for *benign* and *malignant*, respectively.

Fuzzy CoCo is thus used to search for four parameters: input membership function values, relevant input variables, and antecedents and consequents of rules. These search goals are more ambitious than those defined in our previous work [11–13] as the consequents of rules are added to the search space. The genomes of the two species are constructed as follows:

- Species 1: Membership functions. There are nine variables ($v_1 - v_9$), each with two parameters *P* and *d*, defining the start point and the length of the membership-function edges, respectively (Figure 7).
- Species 2: Rules. The i-th rule has the form:
  if (v_1 **is** A_1^i) **and** ... **and** (v_9 **is** A_9^i) **then** ($output$ **is** C^i),

  A_j^i can take on the values: 1 (*Low*), 2 (*High*), or 0 or 3 (*Other*). C^i can take on the values: 1 (*Benign*) or 2 (*Malignant*). Relevant variables are searched for implicitly by letting the algorithm choose non-existent membership functions as valid antecedents; in such a case the respective variable is considered irrelevant.

Table 2 delineates the parameters encoding for both species' genomes, which together describe an entire fuzzy system. Note that in our previous work both membership functions and values were encoded in the same genome, i.e., there was only one species.

Table 2 Genome encoding of parameters for both species. Genome length for membership functions is 54 bits. Genome length for rules is $19 \times N_r + 1$, where N_r denotes the number of rules.

Parameter	Values	Bits	Qty	Total bits
Species 1: Membership functions				
P	[1-8]	3	9	27
d	[1-8]	3	9	27
	Total			54
Species 2: Rules				
A	[0-3]	2	$9 \times N_r$	$18 \times N_r$
C	[1,2]	1	$N_r + 1$	$N_r + 1$
	Total			$19 \times N_r + 1$

To evolve the fuzzy inference system, we applied a Fuzzy CoCo algorithm with the same evolutionary parameters for both species. Values and ranges of values used for these parameters were defined according to preliminary tests performed on benchmark problems (mostly function-optimization problems found in Potter [16]). Table 3 delineates these values. The algorithm terminates when the maximum number of generations, G_{max} is reached (we set $G_{max} = 1000 + 100 \times N_r$, 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 (in our experiments we used as threshold value 10^{-4}). Note that mutation rates are relatively higher than with a simple genetic algorithm, which is typical with coevolutionary algorithms [16]. This is due in part to the small population sizes and to elitism.

Table 3 Fuzzy CoCo set-up for the WBCD problem.

Parameter	Values
Population size N_p	40
Maximum generations G_{max}	$1000 + 100N_r$
Crossover probability P_c	1
Mutation probability P_m	{0.1,0.2,0.3}
Elitism rate E_r	{0.4,0.5,0.6}
"Fit" cooperators N_{cf}	1
Random cooperators N_{cr}	{1,2,3}

Our fitness function combines two criteria: 1) F_c—classification performance, computed as the percentage of cases correctly classified and 2) F_v—the maximum number of variables in the longest rule. The fitness function is given by $F = F_c - \alpha F_v$, where $\alpha = 0.0015$. F_c, the percentage of correctly diagnosed cases, is the most important measure of performance. F_v measures the linguistic integrity (interpretability), penalizing systems with a large number of variables in their rules. The value α was calculated to allow F_v to occasion a fitness difference only among systems exhibiting the same classification performance.

We stated earlier that cooperative coevolution reduces the computational cost of the search process. In order to measure this cost we calculate the maximum number of fuzzy-system evaluations performed by a single run of Fuzzy CoCo. Each generation, the N_p individuals of each population are evaluated $(N_{cf} + N_{cr})$ times. The total number of fuzzy-system evaluations is then $2 \times G_{max} \times N_p \times (N_{cf} + N_{cr})$. This value goes from 352×10^3 evaluations for a one-rule system search, up to 480×10^3 evaluations for a five-rule system. The number of fuzzy-system evaluations required by our single-population approach was 500×10^3 for a one-rule system and 900×10^3 for a five-rule system [13].

6 Results

A total of 193 evolutionary runs were performed, all of which found systems whose classification performance exceeds 95.1%. In particular, considering the best individual per run (i.e., the evolved system with the highest classification success rate), 167 runs led to a fuzzy system whose performance exceeds 97.5%, and of these, 11 runs found systems whose performance exceeds 98.5%.

Table 4 compares our best systems with the top systems obtained in our own previous work, which were the best reported to date [13] (also, see [13], for several references concerning breast-cancer diagnostic systems). The evolved fuzzy systems described in this paper can be seen to surpass those obtained by our previous approach in terms of performance, while still containing simple, interpretable rules. As shown in Table 4, we obtained higher-performance systems for all five rule-base sizes, i.e., from one-rule systems all the way up to five-rule systems.

Table 4 Comparison of the best systems evolved by Fuzzy CoCo with the top systems obtained in our previous work with a single-population approach [13]. Shown below are the classification performance values of the top systems obtained by these approaches, along with the number of variables of the longest rule in parentheses. Results are divided into five classes, in accordance with the number of rules-per-system, going from one-rule systems to five-rule ones.

Rules per system	Single population GA [13]	Fuzzy CoCo	
	best	average	best
1	97.07% (4)	97.12% (3)	97.36% (4)
2	97.36% (4)	97.66% (4.1)	98.54% (5)
3	97.80% (6)	97.91% (4.4)	98.54% (4)
4	97.80% (-)	98.00% (4.7)	98.54% (3)
5	97.51% (-)	97.98% (5.2)	98.68% (5)

We next describe two of our top-performance systems, which serve to exemplify the solutions found by Fuzzy CoCo. The first system, delineated in Figure 8, presents the highest classification performance evolved to date. It consists of five rules (note that the `else` condition is not counted as an active rule) with the longest one including 5 variables. This system obtains an overall classification rate (i.e., over the entire database) of 98.68%.

In addition to the above five-rule system, evolution found systems with 2, 3, and 4 rules exhibiting the second-best classification performance (Table 4). Among these three systems, we consider as best the system with the smallest number of conditions (i.e., the total number of variables in the rules). Figure 9 presents one such system. This two-rule system, containing a total of 9 conditions, obtains an overall classification rate of 98.54%. Its longest rule has 5 variables.

Database

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
P	5	1	2	3	6	2	3	5	1
d	8	6	2	1	4	7	3	8	2

Rule base

Rule 1 **if** (v_2 **is** *Low*) **and** (v_3 **is** *Low*) **and** (v_7 **is** *Low*) **then** ($output$ **is** *benign*)

Rule 2 **if** (v_1 **is** *Low*) **and** (v_4 **is** *Low*) **and** (v_6 **is** *Low*) **and** (v_8 **is** *Low*) **and** (v_9 **is** *Low*) **then** ($output$ **is** *benign*)

Rule 3 **if** (v_1 **is** *High*) **and** (v_5 **is** *Low*) **and** (v_6 **is** *High*) **and** ($output$ **is** *malignant*)

Rule 4 **if** (v_1 **is** *High*) **and** (v_4 **is** *Low*) **and** (v_6 **is** *High*) **and** (v_8 **is** *Low*) **then** ($output$ **is** *malignant*)

Rule 5 **if** (v_1 **is** *High*) **and** (v_4 **is** *High*) **and** (v_6 **is** *High*) **and** (v_8 **is** *High*) **and** (v_9 **is** *High*) **then** ($output$ **is** *malignant*)

Default **else** ($output$ **is** *malignant*)

Figure 8 The best evolved, fuzzy diagnostic system with five rules. It exhibits an overall classification rate of 98.68%, and its longest rule includes 5 variables.

Database

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
P	3		2	3		3	1	8	2
d	8		2	1		5	5	1	1

Rule base

Rule 1 **if** (v_1 **is** *Low*) **and** (v_3 **is** *Low*) **and** (v_7 **is** *Low*) **and** (v_8 **is** *Low*) **then** ($output$ **is** *benign*)

Rule 2 **if** (v_1 **is** *Low*) **and** (v_4 **is** *Low*) **and** (v_6 **is** *Low*) **and** (v_8 **is** *Low*) **and** (v_9 **is** *Low*) **then** ($output$ **is** *benign*)

Default **else** ($output$ **is** *malignant*)

Figure 9 The best evolved, fuzzy diagnostic system with two rules. It exhibits an overall classification rate of 98.54%, and a maximum of 5 variables in the longest rule.

7 Concluding remarks

We presented Fuzzy CoCo, a new, cooperative coevolutionary approach to fuzzy modeling. We applied Fuzzy CoCo to the

Wisconsin breast cancer diagnosis problem comparing it with the non-coevolutionary approach we applied to the same problem [13]. Our coevolved systems attained a higher classification performance (note that our previous work's performance was the best shown to date), and required less computation to obtain the diagnostic systems than the single-population approach.

These promising results have incited us to engage in further investigation of this approach. We are currently pursuing two avenues of research: (1) application of Fuzzy CoCo to more complex diagnosis problems; and (2) improving and expanding upon the methodology presented herein (e.g., by adapting the set-up according to the particularities of each species). Our underlying goal is to provide an approach for automatically producing high-performance, interpretable fuzzy systems for real-world diagnosis problems.

Bibliography

[1] J. T. Alander. An indexed bibliography of genetic algorithms with fuzzy logic. In Pedrycz [15], pages 299–318.

[2] 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.

[3] 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.

[4] C. L. Karr. Genetic algorithms for fuzzy controllers. *AI Expert*, 6(2):26–33, February 1991.

[5] 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.

[6] O. L. Mangasarian, W. N. Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. Mathematical Programming Technical Report 94-10, University of Wisconsin, 1994.

[7] C. J. Merz and P. M. Murphy. UCI repository of machine learning databases. http://www.ics.uci.edu/˜mlearn/MLRepository.html, 1996.

[8] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Heidelberg, third edition, 1996.

[9] D. E. Moriarty. *Symbiotic Evolution of Neural Networks in Sequential Decision Tasks*. PhD thesis, The University of Texas at Austin, 1997.

[10] J. Paredis. Coevolutionary computation. *Artificial Life*, 2:355–375, 1995.

[11] C. A. Peña-Reyes and M. Sipper. Evolving fuzzy rules for breast cancer diagnosis. In *Proceedings of 1998 International Symposium on Nonlinear Theory and Applications (NOLTA'98)*, volume 2, pages 369–372, Lausanne, 1998. Presses Polytechniques et Universitaires Romandes.

[12] C. A. Peña-Reyes and M. Sipper. Designing breast cancer diagnostic systems via a hybrid fuzzy-genetic methodology. In *1999 IEEE International Fuzzy Systems Conference Proceedings*, volume 1, pages 135–139. IEEE Neural Network Council, 1999.

[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. Evolutionary computation in medicine: An overview. *Artificial Intelligence in Medicine*, 2000. To appear.

[15] W. Pedrycz, editor. *Fuzzy Evolutionary Computation*. Kluwer Academic Publishers, 1997.

[16] M. A. Potter. *The Design and Analysis of a Computational Model of Cooperative Coevolution*. PhD thesis, George Mason University, 1997.

[17] C. D. Rosin and R. K. Belew. New methods for competitive coevolution. *Evolutionary Computation*, 5(1):1–29, 1997.

[18] R. Setiono. Extracting rules from pruned neural networks for breast cancer diagnosis. *Artificial Intelligence in Medicine*, pages 37–51, 1996.

[19] R. Setiono and H. Liu. Symbolic representation of neural networks. *IEEE Computer*, 29(3):71–77, March 1996.

[20] I. Taha and J. Ghosh. Evaluation and ordering of rules extracted from feedforward networks. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 221–226, 1997.

[21] R. R. Yager and D. P. Filev. *Essentials of Fuzzy Modeling and Control*. John Wiley & Sons, Inc., 1994.

[22] R. R. Yager and L. A. Zadeh. *Fuzzy Sets, Neural Networks, and Soft Computing*. Van Nostrand Reinhold, New York, 1994.