International Journal of Neural Systems, Vol. 25, No. 4 (2015) 1550012 (19 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0129065715500124

Artificial Neuron–Glia Networks Learning Approach Based on Cooperative Coevolution

Pablo Mesejo
Department of Information Engineering
University of Parma, Parma 43124, Italy

ISIT-UMR 6284 CNRS, University of Auvergne
Clermont-Ferrand 63000, France

Oscar Ibáñez
European Centre for Soft Computing
Mieres 33600, Spain

Departmemt of Computer Science and Artificial Intelligence (DECSAI)
University of Granada, Granada 18071, Spain

Enrique Fernández-Blanco, Francisco Cedrón,
Alejandro Pazos and Ana B. Porto-Pazos*
Department of Information and Communications Technologies
University of A Coruña, A Coruña 15071, Spain
**anuska@udc.es*

Accepted 9 February 2015
Published Online 6 April 2015

Artificial Neuron–Glia Networks (ANGNs) are a novel bio-inspired machine learning approach. They extend classical Artificial Neural Networks (ANNs) by incorporating recent findings and suppositions about the way information is processed by neural and astrocytic networks in the most evolved living organisms. Although ANGNs are not a consolidated method, their performance against the traditional approach, i.e. without artificial astrocytes, was already demonstrated on classification problems. However, the corresponding learning algorithms developed so far strongly depends on a set of glial parameters which are manually tuned for each specific problem. As a consequence, previous experimental tests have to be done in order to determine an adequate set of values, making such manual parameter configuration time-consuming, error-prone, biased and problem dependent. Thus, in this paper, we propose a novel learning approach for ANGNs that fully automates the learning process, and gives the possibility of testing any kind of reasonable parameter configuration for each specific problem. This new learning algorithm, based on coevolutionary genetic algorithms, is able to properly learn all the ANGNs parameters. Its performance is tested on five classification problems achieving significantly better results than ANGN and competitive results with ANN approaches.

Keywords: Artificial neuron–glia networks; artificial neural networks; artificial astrocytes; glial cells; evolutionary algorithms; cooperative coevolutionary genetic algorithm; genetic algorithms; parameter optimization; classification.

1. Introduction

Artificial Neural Networks (ANNs) are composed of a set of very simple processing elements (artificial neurons) and by a certain number of connections between them.[10] They were created to process information in computers like the human brain.[38,44,59] During the last decades, ANNs have been evolving towards more powerful and more biologically realistic models,[27,77] and the research in the neuroscience field has advanced remarkably.

More complex neural circuits have been closely observed[18,36] and, among them, important achievements refer to the glial system.[6,49]

The importance of the functions developed by a type of glial cells called astrocytes leads researchers to think that their participation in the processing of information in the nervous system is much more relevant than previously assumed. It has been demonstrated their key role in the acceleration of the impulse transmission, the establishment of the best transmission routes, or the choice of the elements that constitute a specific circuit.[7,49,50,69] Astrocytes may play more active roles in synaptic transmission that previously thought, and hence, in transferring and processing information in the brain.[5,68] A critical review considering the different schools of thought and scientific trends regarding the role of astrocytes can be found in Ref. 31. Astrocytes might facilitate the processing of information performed by connectionist systems in Artificial Intelligence (AI) and, in that case, it might be useful to integrate into the ANNs other elements that are not artificial neurons (i.e. artificial astrocytes). Accordingly, we have developed artificial models of neuroglial circuits that represent an attempt of studying the effects of the neuroglial activity that have been observed until now at the Neuroscience laboratories, and to give expression to the hypothesis and phenomena of this activity. We analyzed how ANNs should be designed for integrating these artificial astrocytes, resulting in the recently created Artificial Neuron–Glia Networks (ANGNs)[53]: a novel connectionist system still in its very infancy. ANGNs include both artificial neurons and astrocytes for which a biologically inspired learning algorithm was proposed. For the moment, the inclusion of artificial astrocytes has improved the classification accuracy of ANNs in some classification problems.[54] Nevertheless, the behavior of ANGNs strongly depends on a set of glial parameters which are manually tuned. As a consequence, prolonged previous experimental tests, both computational and biological, have to be done in order to determine an adequate set of values, making such manual parameter configuration time consuming and error-prone. It may also introduce a bias in comparing an algorithm with a reference, due to better knowledge of the algorithm under consideration and to possible different time spent tuning each of them. Since the connection weights among neurons have to be considered at the same time, trying all possible parameter combinations is a NP-hard optimization task[11] and, furthermore, their ideal value change with the problem to be tackled. In addition, the strong interrelation between glial parameters and connection weights is expected to draw a noisy and highly multimodal search space. Then, a new learning approach that includes those parameters into the learning process, at the same time that considers the highly complex underlying optimization problem, would be extremely beneficial.

The classic gradient descent search techniques perform efficiently when the problem under consideration satisfies tight constraints. But when the search space is rugged, i.e. the function to be optimized is nonsmooth, discontinuous, multimodal and/or noisy, then stochastic methods have been found to consistently outperform traditional methods.[32,40] Among the stochastic approaches to continuous optimization, Evolutionary Algorithms[a] (EAs) and Swarm Intelligence algorithms, as well as other Metaheuristics, offer a number of attractive features: no requirement for a differentiable or continuous objective function, robust and reliable performance, global search capability, virtually no need of specific information about the problem to solve, easy implementation, and implicit parallelism.[9,64] They have been applied to many different types of difficult problem domains, such as parameter optimization[13] and machine learning.[57] In particular, Evolutionary Neural Networks, the research field dealing with hybridization of EAs and ANNs, present an extensive track of successful approaches.[3,34,35,58,62,76] Within this research field, we would like to highlight coevolutionary learning approaches, which have been successfully applied in the context of ANN automatic design,[14,24,25,30] and ensembles evolution.[23,45,65]

Coevolutionary algorithms (CEAs)[47] provide some interesting features for learning and optimization purposes. In such algorithms, fitness[b] itself

[a] Evolutionary algorithms are bioinspired methods in which a population of abstract representations (called chromosomes, genotype or genome) of candidate solutions (called individuals or phenotypes) to an optimization problem evolves toward better solutions.

[b] The fitness function is defined over the genetic representation and measures the quality of the represented solution.

becomes a measurement of interacting individuals. This assumption ostensibly allows the potential for evolving greater complexity by allowing pieces of a problem to evolve in tandem. Hence, a CEA is an EA (or collection of EAs) in which the fitness of an individual depends on the relationship between that individual and other individuals.[73] Depending on the nature of the interactions among individuals we can distinguish between competitive and cooperative coevolutionary algorithms (CCEAs). In the former, each species competes with the rest,[60] while, in the latter, all the species collaborate to build a solution for the problem.[56]

The originally stated aim of CCEAs was to attack the problem of evolving complicated objects by explicitly breaking them into parts, evolving the parts separately, and then assembling the parts into a working whole. According to Pena-Reyes and Sipper,[48] the cooperative coevolution is recommended when the problem to be resolved has the following characteristics: (i) the search space is complex; (ii) the problem, by definition, can be split; (iii) different types of values and concepts are used; and (iv) there is a strong interdependence between the components of the solution. All these requirements perfectly match with the problem of evolving ANNs connection weights and glial parameters at the same time. The underlying search space is expected to be NP-hard and multimodal, and two different sets of parameters (neurons and astrocytes) have to be learned, with different meaning and range of values. Finally, the performance of the final system strongly depends on the values of these two sets.

Then, in this paper, we propose a novel learning approach for ANGNs with the aim of avoiding the manual tuning of its parameters (problem dependent), to fully automate the learning process and, at the same time, giving the possibility of testing any kind of reasonable parameter configuration. This new learning algorithm, which is based on a coevolutionary approach, is able to properly learn all the ANGNs parameters at the same time, and it supposes another step in what concerns the combination of neural networks, glial cells, and a self-adaptive evolutionary process.

The proposed method is tested on five well-known classification problems obtained from the Machine Learning Repository[8] and the Knowledge Extraction based on Evolutionary Learning data set.[2] The results achieved outperformed the previous ANGN approach, where glial parameters were manually chosen through an extensive preliminary experimentation,[54] and were competitive with the ones obtained by classical ANNs trained using backpropagation.

The structure of the paper is as follows. In Sec. 2 we describe the biological inspiration and fundamentals of the ANGNs. Section 3 is devoted to introduce our new learning approach based on cooperative coevolution. In Sec. 4, we test and compare the new proposal against the previous evolutionary approach over five classification problems. Finally, we present some concluding remarks and future works in Sec. 5.

2. Artificial Neuron–Glia Networks

The details of how the brain performs many information processing tasks are still unknown. Abundant evidence has suggested the existence of bidirectional communication between astrocytes and neurons, and has led to the proposal of a new concept in synaptic physiology, the tripartite synapse, which consists of three functional elements: the presynaptic and postsynaptic elements and the surrounding astrocytes, a subtype of glial cells in the central nervous system.[7,50,69] The communication between these three elements has highly complex characteristics, which seem to reflect more reliably the complexity of the information processing between the elements of the nervous system.

The classical view of astrocytes as simple supportive cells for neurons is being challenged by accumulating evidence that suggest a new vision in which they are active elements involved in the physiology of the nervous system (see Refs. 5, 50 and 68). This new vision is based on the fact that astrocytes are stimulated by synaptically released neurotransmitters, which increase the astrocyte Ca^{2+} levels and stimulate the release of gliotransmitters that regulate synaptic efficacy and plasticity. While this idea has been questioned by some reports that failed to detect astrocyte-mediated neuromodulation[1,21,52,63] (for detailed discussions see Refs. 5, 31 and 68), the abundant reports showing gliotransmitter release and neuronal and synaptic regulation in different brain areas (for a comprehensive list, see Table 1 in Ref. 5) suggest that our understanding of synaptic function, previously thought to exclusively result from signaling between neurons, may

Table 1. Fixed values employed by the ANGN approach (second column) and range of possible values for each gene of the glial chromosomes in the proposed coevolutionary approach (third column). a refers to the percentage of increment, b to the percentage of decrement, k is the number of iterations, μ the number of activations, and N–G alg refers to the neuron–glia algorithm.[4]

Neuron–Glia parameter	ANGN	CCGA approach
Activation-iteration	2–4, 2–6, 3–6 and 3–8	$k \in [3, 9]$ and $\mu \in [2, 8]$ $k \geq \mu$
% of increment/ decrement	25%/ 50%	$a \in [10, 80]$ and $b \in [20, 90]$ $a < b$
Neuron–Glia algorithm	Attenuated effect of astrocyte[4]	$N–G\ alg \in \{1, 6\}$

need to include the bidirectional signaling between neurons and astrocytes. Hence, astrocytes have been postulated to be integral elements involved in the synaptic physiology, therefore contributing to the processing, transfer and storage of information by the nervous system. Accordingly, reciprocal communication between astrocytes and neurons has been hypothesized to be part of the intercellular signaling processes involved in brain function. Consequently, the design of ANGNs endowed with similar properties of astrocyte–neuron communication in biological systems are a logical step to be followed by AI but it is still lacking. We therefore designed ANGNs, that include both artificial neurons and artificial astrocytes,[53,54] and investigated the consequences of the presence of artificial astrocytes by the comparison of the performance of ANNs and ANGNs on some classification problems.

Other researchers have studied the astrocytic functions by means of realistic computational models of neurons and astrocytes. Some of the most representatives are Refs. 37, 70 and 71, where detailed biophysical models of the bidirectional signaling events between astrocytes and neurons are introduced (essentially based on the work by Nadkarni and Jung who elaborated a mathematical model of the synaptic interactions between neurons and astrocytes, taking into account the concept of tripartite synapse). Other related works, which focus on modeling the astrocytes and/or the neuron–astrocyte communication, are the ones by Wallach *et at.*,[72] De Pittà *et al.*,[16,17] Min *et al.*[41] and Mitterauer.[42] They propose mathematical and computational models of astrocytes that are of interest to try to check the effects of this operation in the ANGN efficiency. All the aforementioned proposals are mathematical models implemented with computational simulators that allow the researchers to build realistic models of neurons. Usually, those models are then used to study the brain together with *in vitro* experiments. Therefore they are good references to develop new AI techniques that include the observed behaviors (in our case, they can be useful to inspire new neuron–glia algorithms (N–G alg)).

Finally, there is just a couple of previous works dealing with ANGNs. They both were presented in conferences and have not been developed so far. Ikuta *et al.*[33] simulate the calcium waves generated by the astrocytes to communicate between them. Recently, in Ref. 66 the authors introduce astrocytes in spiking neural networks.

The functioning of our ANGNs follows the steps that were successfully applied in the construction and use of ANNs: design, training, testing and execution. The design of ANGNs is based on feed-forward multilayer architectures which are totally connected, without backpropagation or lateral connections, and oriented towards classification and pattern recognition. The design of the first ANGNs was oriented to solve classification problems by means of simple networks, i.e. multilayer networks, although future research may lead to the design of models in more complex networks. It seemed a logical approach to start the design of these new models from simple ANNs, and to orientate the latest discoveries on astrocytes in information processing towards their use in classification networks. This decision was biologically biased since the control of the reinforcement or weakening of the connections in the brain is related to the adaptation or plasticity of the connections, which lead to the generation of activation ways. This process can therefore improve the classification of the patterns and their recognition by the ANGN.

In order to design the integration of the astrocytes into the ANNs and to elaborate a learning method for the resulting ANGNs that allows us to check whether there is an improvement in these systems, we studied the main existing training methods. We analyzed unsupervised and supervised training methods, and others that use or combine some of their characteristics. After considering the existing approaches for training multilayer ANNs, none of them seemed adequate to be exclusively used.[53] We therefore designed a new hybrid learning method for training ANGNs based on genetic algorithms (GAs).[29] Each individual, randomly generated, represents initial neuronal connections weights of the artificial network. Our method looks for optimal connection weights in two phases. An unsupervised learning phase (where a selected N–G alg is used for modifying weight connections) and a supervised learning phase (where the GA is used to optimize connection weights).[4,54] Both phases where deeply described in Ref. 54 and they are summarized below.

2.1. *Unsupervised learning phase*

In this phase, ANGN connections weight values are modified following rules based on the behavior that astrocytes of the brain are supposed to have, considering the observations made on living beings nervous system.[49] Due to the lack of knowledge regarding the specific characteristics of the modifications that the astrocytes make in the neuronal connections, different N–G alg were implemented trying to mimic biological observations. Accuracy achieved by the network in solving a specific problem depends on the algorithm used, but in all cases the efficiency is higher than the one obtained by an ANN with the same characteristics.[4,53]

Specifically, the ANGNs have one artificial astrocyte per neuron which is responsible for monitoring the activity of that neuron (Fig. 1(a)). This intends to mimic the tripartite synapse in the brain (see Fig. 1(b)) because astrocyte regulation of synaptic transmission may occur at single synapses.[46,49] The astrocyte modifies the weight of the connections it is associated with, depending on the activity of the neuron (Fig. 1(c)). In the brain, the activation of astrocytes is a slow process (i.e. hundreds of milliseconds and seconds), if we compare it with neural activity[49] (i.e. milliseconds). This slowness has led to the presentation to the ANGN of each training pattern during more than one cycle or iteration. If it imitates this slowness, the ANGN will need k cycles or iterations to process each input pattern. We must also consider that the contribution of the astrocytes to the weights of the ANGN connections takes place according to a time factor, given the fact that they act slowly and their responses are nonlinear.[4,54] Although the incorporation of the concept of time could suggest a similar approach to the well-established[22,61,77] Spiking Neural Network models,[26,27] we are trying to model a more complex model in a completely different way as explained below.

We defined two variables in the artificial neuron–glia interaction (μ and k, which are usually introduced together and separated by a hyphen):

(1) Activations μ as the number of times the neuronal connection was required to be active to stimulate the associated astrocyte, i.e. 2-k is more sensitive than 3-k (being $k = 4, 6$ or 8).
(2) Iterations k, as the number of times in which the neuronal connections are possibly active to stimulate the astrocyte (for example, if μ-k corresponds to 3–6, at least three activations of the neuron had to occur during six consecutive iterations to stimulate the associated astrocyte), consequently, the strength is: $\mu - 8 > \mu - 6 > \mu - 4$ (being $\mu = 2$ or 3) because the ability of a neuron to stimulate the associated astrocyte is higher for $k = 8$ than $k = 6$ and $k = 4$.

Therefore, the behavior of our artificial astrocyte is determined by the parameters $k \in \mathbb{N}\backslash\{0\}, \mu \in [1, k]$ and $a, b \in [0, 1]$ (see Fig. 1(c)). Each instance or input pattern that is used for training, validating or testing the artificial network is processed k times (iterations). The artificial astrocyte registers the activity of the neuron during the k iterations, applying a function: $u : \mathbb{R} \to \mathbb{Z}$ over the output of the neuron $y_j(t)$, where u indicates if the neuron has fired ($u(x) = 1$) or not ($u(x) = -1$).

$$u(x) = \begin{cases} -1 & \text{if } x \leq 0, \\ 1 & \text{if } x > 0. \end{cases} \tag{1}$$

In our model, the activation of a neuron is determined by the following activation functions: hyperbolic tangent in all layers but in the output layer where the threshold function is applied (with threshold equal to 0.5 and an expected binary output (1 if the function input value is larger than 0.5, and 0 otherwise). Hence the artificial astrocyte has a register of the neuron's activity with a temporal window of k iterations. Observing this activity, the astrocyte will modify the weight of its associated neuronal connections (w_i) when the counter of the activity of the neurons reaches the value μ. Figure 1(c) shows

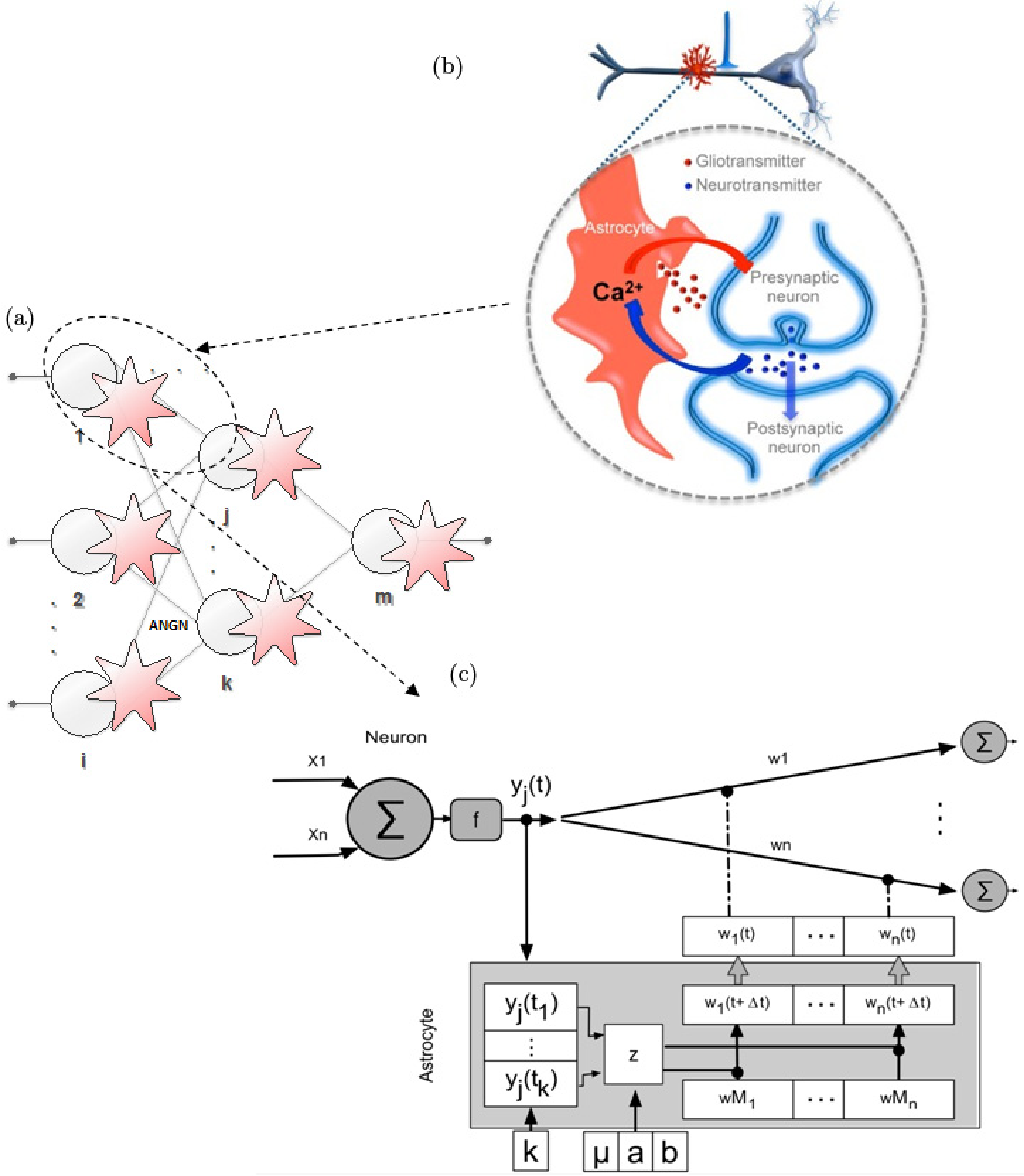

Fig. 1. (a) ANGN structure, where artificial neurons and artificial astrocytes are represented by circles and stars, respectively. (b) Scheme of tripartite synapse in the brain. (c) Regulation of artificial neuron by artificial astrocyte.

how the input neuronal connections are modified. An astrocyte might modify output neuronal connections or both. In this work, the artificial astrocyte modify only output neuronal connections: $w_i(t+\triangle t) = w_i(t) + \triangle w_i(t)$, where $\triangle w_i(t) = |w_i(t)|z(t)$ and $z : \mathbb{N}\backslash\{0\} \rightarrow \mathbb{R}$ is a function defined as

$$z(t) = \begin{cases} a & \text{if } r_j(t) = \mu, \\ -b & \text{if } r_j(t) = -\mu, \end{cases} \tag{2}$$

being $r_j : \mathbb{N}\backslash\{0\} \rightarrow [-\mu, \mu]$ the function that returns the number of times a neuron has fired. If the neuron was active μ times, the weights of the connections, depending on its sign, will be increased/decreased by a percentage a, while they will be decreased/increased by a percentage b if the neuron remained inactive during those μ iterations.[4] These changes take into account the fact that the algorithm employed in this paper does not respect the limit of the connection weights, i.e. they do not have to be kept between −1 and 1 when they are affected by the astrocitic action.

Taking into account the above mentioned, six algorithms were implemented so far. They were different in two aspects: the specific implementation they make of the r_j function, and whether or not they respect the weight limit when the neuronal connection is being modified. As shown in Table 1, the algorithm used is the so-called "Attenuated Effect of Astrocyte". The implementations of different N–G alg (r_j) are explained in Ref. 4 but a summary of the most important ones can be found below:

- *Consecutive activations, weight limited.* The astrocyte respects the weight limit of the connections (wM_i): $w_i(t + \triangle t) = \min\{w_i(t) + \triangle w_{ji}(t), wM_i\}$. This algorithm contemplates only consecutive neuron activations; if the neuron reaches the activity or inactivity level that makes the astrocyte act, the activity counter is restarted. The neuronal activity level, following these restrictions, is given by the function:
$$r_j(t) = \begin{cases} u(y_j(t)) + r_j(t-1) & \text{if (condition 1),} \\ u(y_j(t)) & \text{in other case,} \end{cases} \tag{3}$$
where (condition 1) is true if $t > 0$, $u(y_j(t)) = u(y_j(t-1))$ and $r_j(t-1) \in (-\mu, \mu)$.
- *Nonconsecutive activations, weight limited.* The astrocyte respects the weight limit of the connections: $w_i(t+\triangle t) = \min\{w_i(t)+\triangle w_{ji}(t), wM_i\}$. In this algorithm, the neuron activations need not be consecutive. If the neuron reaches the activity or inactivity level that makes the astrocyte act, the activity counter is restarted. The neuron activity level, following these restrictions, is given by the following function:
$$r_j(t) = \begin{cases} u(y_j(t)) + r_j(t-1) & \text{if (condition 2),} \\ u(y_j(t)) & \text{in other case,} \end{cases} \tag{4}$$
where (condition 2) is true if $t > 0$ and $r_j(t-1) \in (-\mu, \mu)$. Having the activity of the neuron not required to be consecutive gives rise to this result: if an astrocyte increments the weight of a connection of a neuron, it indicates that the neuron fired μ iterations more than it remained inactive. If an astrocyte decrements the weight of a connection to a neuron, it indicates that the neuron fired μ iterations less than it remained inactive.
- *Consecutive activations, weight unlimited.* The behavior of this algorithm is the same as the previous one, except that in this case the astrocyte will not respect the limit weight of the connections; hence they can reach any value: $w_i(t+\triangle t) = w_i(t) + \triangle w_{ji}(t)$.
- *Nonconsecutive activations, weight unlimited.* The behavior of this algorithm is the same as the previous one, except that in this case the astrocyte will not respect the limit weight of the connections; hence they can reach any value: $w_i(t+\triangle t) = w_i(t) + \triangle w_{ji}(t)$.
- *Attenuated effect of astrocyte.* This is the algorithm used in this work. The astrocyte will not respect the limit weight of the connections ($w_i(t+\triangle t) = w_i(t) + \triangle w_{ji}(t)$) and the activity of the neuron need not be consecutive:
$$r_j(t) = \begin{cases} u(y_j(t)) + r_j(t-1) & \text{if (condition 2),} \\ r_j(t-1) & \text{if (condition 3),} \\ u(y_j(t)) & \text{in other case,} \end{cases} \tag{5}$$
where (condition 3) is true if $t > 0$ and $r_j(t-1) \in \{-\mu, \mu\}$.

The major difference with the previous algorithms stems from the management of the activity counter of the neuron: when the neuron reaches the activity level $\{-\mu, \mu\}$ that makes the astrocyte modify its neuronal connections, the activity counter is not set to zero (it retains the value). This behavior has a noticeable consequence in the modification of the connections weight: when the point at which an astrocyte modifies the weight of the connections is reached in a given iteration and the neuron fires again in the next iteration, the astrocyte will increase the connections weight of the neuron again. The behavior when the neuron remains inactive is similar, with the outcome being the weight is decreased. In the previous algorithms, having the activation counter be set to zero, the counter needed to reach the value $\{-\mu, \mu\}$ again for the astrocyte to act (thus a minimum of μ iterations of neuronal activity/inactivity are required). This behavior implies an extra reinforcement on those neurons that fire the most, it also makes the astrocytic effect last longer, and disappear only gradually over time.

In summary, present neuron–glia networks had an artificial astrocyte for each neuron. Each astrocyte only responds to the activity of the associated neuron and modulates the connections of that neuron with neurons of the next (adjacent) layer. The parameters defining the neuron–astrocyte interactions are: iterations (k), activations (μ), percentage increment (a) and decrement (b) of the connection weights, and the N–G alg. Their values were chosen according to the results obtained in the experiments conducted by Araque *et al.*[6,49–51] Since the particular mode of action of astrocytes in brain synapses is still unknown, the set up of parameter values was manually tuned after numerous experimental tests (see Table 1).

For simplicity, spatial spread of the astrocyte signal to other neurons or communication between astrocytes were not considered. Several mechanisms and physiological consequences of astrocyte–neuron communication occur. Under what conditions one specific modulatory effect takes place in a particular neural network remains unknown. For simplicity and as a first approximation to a complex problem, present work focused in modeling astrocyte-induced synaptic potentiation to investigate whether artificial astrocytes improve ANN performance. The development of future models of astrocyte–neuron interaction that incorporate the richness of biological interactions are required to test whether they provide similar, or even better, improvements of neural network performance.

2.2. *Supervised learning phase*

In the second learning phase, supervised learning, the connection weights are optimized by means of GAs, according to their Mean Square Error (MSE), which was stored in the first phase.

It is important to remark that, like in the brain where astrocytes are always present, artificial counterparts act on both the validation and test phases, and not only during the training. Therefore, astrocytes are active in the operation/execution phase of the network. They are much more than mere temporal scales since they control and monitor the neuronal activity, and they decide how, what and when the ANN is going to learn.

So far, the inclusion of artificial astrocytes in ANNs has demonstrated to improve the classification accuracy of ANNs.[53,54] The reason why these satisfactory results were achieved with the ANGN is attributed to what was expected when the hybrid method was designed. The individuals, once evaluated, are presented to the GA arranged by their MSE. The changes made to the weights during the unsupervised learning phase cause this resulting order to be different from a potential order obtained in case this phase did not exist. This is so because their MSE varies according to this new method. The number of generations necessary for reaching zero or a minimal error is in many cases smaller than in the case of the ANN which are only trained with GA. This is motivated by the fact that the process of searching for the global minimum carried out by the GA adds to the local adjustment process implemented thanks to this nonsupervised stage.

In addition, we built ANNs with the same additional neurons and layers as astrocytes were incorporated into ANGNs. The objective was twofold, ruling out that ANGN performance improvement is due to incorporate more processing elements, and showing that an artificial astrocyte is different from a neuron. The results indicated that ANN performance did not correlate with the number of neurons or the architecture, suggesting that differences in ANN and NGN performances cannot be accounted for an increased number of elements, but they are specifically due to astrocytes.[54]

Further conclusions are[54]: (i) the relative efficacy of ANGNs versus ANNs increases as the complexity of the network increases, which agrees with the gradual increase of the glia proportion observed in the phylogeny as the nervous system complexity increases; (ii) the relative network performance improvement by artificial astrocytes depends on the problem tested; and (iii) ANGNs performance improvement depends on intrinsic properties of astrocytes, which supports the notion that neuron–glia interaction in biological synapses represents a fine tuned communication.

3. A CCEA for ANGNs Optimization

As already mentioned in Sec. 1, the optimization of connection weights among neurons and the optimization of the neuron–glia parameters at the same time is a NP-hard optimization task and, furthermore, their ideal values change with the problem tackled. CCEAs present really good characteristics to tackle the problem of simultaneously evolving all the ANGN parameters. Therefore, the main contribution of this paper is the introduction, design and validation of a novel learning approach that uses CCEAs for the optimization of all ANGN parameters. We aim to increase the knowledge on how neurons and astrocytes interact when they process information, both in the brain and in connectionist systems. In the following subsections specific details about this novel learning approach are given together with the motivation of the design decisions taken so far.

3.1. *Codification*

As explained before, in the previous approach (ANGN) the values of the neuron–glia parameters (see first column of Table 1) were fixed based on biological observations and the good performance achieved.[54] In the case of the proposed approach, we have implemented a CCEA with two populations to optimize the ANGN weights and the neuron–glia parameters, respectively. The individuals of the former are codified using a variable-length real-valued chromosome whose size corresponds to the number of connections weights among artificial neurons. The latter deals with a mixed coding scheme to represent the five parameters that model the neuron–astrocyte interactions (Table 1), that is:

- Two integer-coded genes to determine the number of iterations (k) and activations (μ). Their values are restricted to the following premise, $k \geq \mu$. In addition, the bounds of the variables are set to $3 \leq k \leq 9$, and $2 \leq \mu \leq 8$. These values were experimentally determined in Ref. 55 after concluding that, above eight iterations the computational time is too high and below three there is not significant contribution of the astrocytes.
- Two integer-coded genes to determine the increment (a) and decrement (b) percentage to be applied over the ANN weights. Based on biological and physiological observations[49] we establish the following premise $a < b$ with the following bounds [10%, 80%] and [20%, 90%] for a and b, respectively. In contrast, previous works[4,53,54] fixed those values to 25% and 50% due to the inability to automatically learn them. We only try multiples of 10 as feasible values for the percentages of increment and decrement, since these values appear to be sufficient to emulate the different types of possible reinforcement strategies in the brain.
- One integer-coded gene specifying the type of (N–G alg) to be applied. Each of the six existing algorithms[4] is codified with a number in the interval $[1, 6]$. The simultaneous learning of the algorithm and the rest of the parameters is done without the need of further restrictions. This is possible because the six algorithms use the same set of parameters (a, b, k and μ).

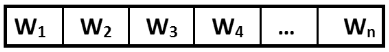

Fig. 2. Chromosome 1: Connection weights.

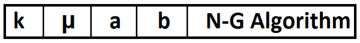

Fig. 3. Chromosome 2: Neuron–glia parameters.

The codification of the chromosomes of both populations is summarized in Figs. 2 and 3. As in previous works, all the astrocytes share the same parameters. Namely, the neuron–glia parameters encoded in chromosome 2 are common for all the astrocytes in the network.

3.2. *Interaction mechanisms*

A single individual in an EA (population-based approaches) represents itself a solution to the problem at hand. However, no matters the specific coevolutionary method (either cooperative or competitive), they manage two or more populations whose individuals need to collaborate/interact to construct a solution for the problem. This implies that it is not possible to evaluate how good is an individual without its interaction with the individuals of other populations. Thus, the question of how these collaborators or competitors are determined may be among the most important design decisions for the successful application of CEAs.[74] Figure 4 shows the general interaction scheme of CCEAs. The most obvious (and computationally expensive) method to evaluate an individual in a coevolutionary setting is to make it interact with all possible collaborators or competitors from the other populations, i.e. perform all possible combinations of that individual with each individual in the other populations. Another extreme

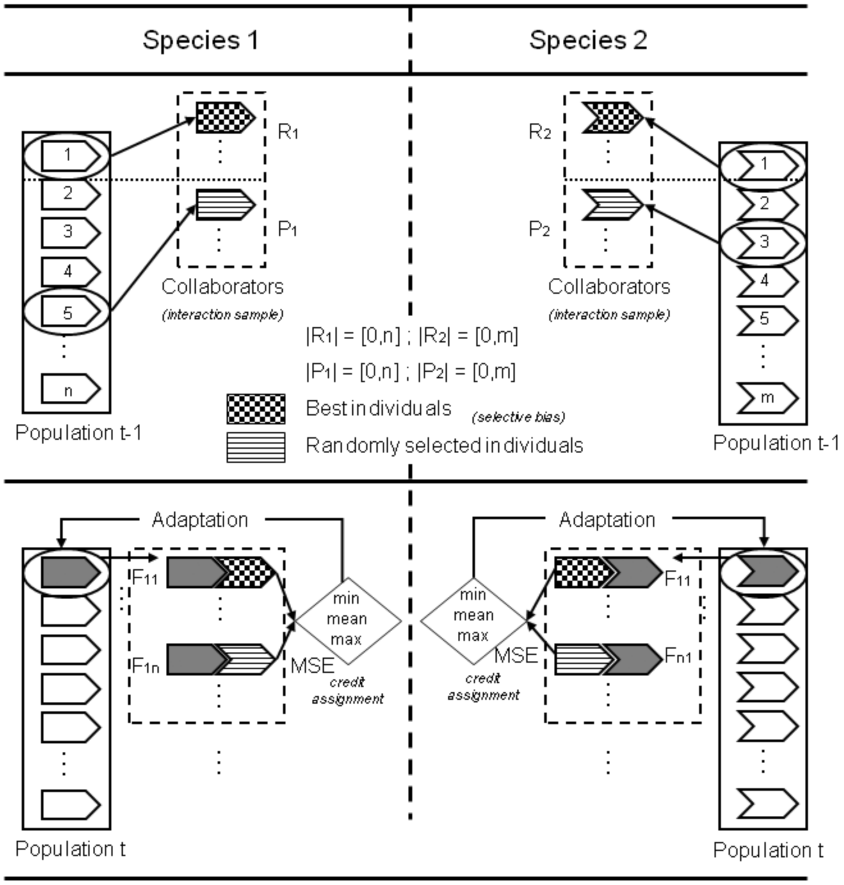

Fig. 4. General interaction scheme of CCEAs.

alternative is to make an individual be only involved in a single interaction, i.e. combine that individual with one single individual selected among all the individuals in the other populations. Such choice leaves open the obvious question of how to pick the collaborator or competitor. Of course, between these two extremes there is a whole set of possibilities involving some subset of interactions. Again, collaborators/competitors for such interactions may be chosen in a variety of ways ranging from uniformly random, to fitness biased. In total, there are mainly three attributes to be specified to properly describe the interaction mechanism among populations in CEAs, suggesting a wide range of possible strategies[73]:

- Interaction sample size: The number of collaborators from each population employed to evaluate a particular individual of the remaining population.
- Interaction selective bias: The method for selecting collaborators/competitors form other populations. For example, an individual could be selected either randomly or based on the fitness value.
- Interaction credit assignment: The method for computing the individual evaluation (fitness) from its multiple interactions with other individuals. The most common methods are the minimum, the maximum and the mean of the fitness values of all the selected interactions.

We decided to use previously successful interaction selective bias[12,74] where collaborators are selected combining (one or more) best individuals (from now on R set) with (zero, one or more) random ones (from now on P set). In addition, we tested other selection mechanisms that could bias the search process in a more deterministic way: tournament and diversity. The first one is based on the tournament selection operator frequently used in GAs[29] which is able to control the selection pressure with different window sizes. The second is inspired by the diverse subset of solutions included in the reference

Table 2. Different interaction sample sizes initially considered. R refers to best individuals while P does for random (or diverse) ones.

R-best	P-random
1	0
1	1
1	2
2	2
3	3
4	4

set of the Scatter Search[28] metaheuristic, aiming to include the most diverse solutions according to a specific distance function.

In the case of the credit assignment mechanism, the minimum, the maximum and the mean of the fitness values of all the selected interactions were considered. Finally, concerning the interaction sample size, we tested all the combinations listed in Table 2.

3.3. *Fitness function*

In order to evaluate the performance of any individual in both populations we need, at least, one individual from the other population. The aim is the minimization of the MSE in the following equation (6):

$$\mathrm{MSE}_{ij} = \frac{1}{N}\sum_{l=1}^{N}(F_{ij}(x^l) - y^l)^2, \tag{6}$$

where F_{ij} is the composition of the codified subcomponents in the chromosomes i and j of the species 1 and 2, respectively, i.e. F_{ij} represents an ANGN whose connection weights are those codified in the genes of chromosome i of species 1 (Fig. 2) and whose neuron–glia parameters are those codified in the genes of chromosome j of species 2 (Fig. 3). $F_{ij}(x^l)$ represents the output obtained once the pattern l is presented to the ANGN described by F_{ij}; y^l is the output expected (ground truth) for the same input pattern; and N corresponds to the number of examples in the training set.

With this new evolutionary approach, every time the evaluation of an individual has to be done, either from the population of weights or parameters, this is combined with a number, previously established, of individuals of the other species (see Fig. 4).

Each individual is evaluated according to a fitness function defined for the particular species as follows: $f1(i) = g_{\mathrm{CA}}\{\mathrm{MSE}_{ij}$ with $j \in R2 \cup P2\}$ and $f2(j) = g_{\mathrm{CA}}\{\mathrm{MSE}_{ij}$ with $i \in R1 \cup P1\}$, being i and j the individuals of the species 1 and 2, respectively; $R1$ and $R2$ are two sets of individuals with the higher fitness function value in the previous generation of species 1 and 2, respectively; and $P1$ and $P2$ are two sets of individuals selected randomly from the previous population of the species 1 and 2, respectively. Finally, g_{CA} is the method of credit assignment used.

Thus, each population evolves separately as in a traditional GA through several generations applying the genetic operators' crossover and mutation but needs to cooperate with the individuals of the counterpart population to be able to evaluate their own individuals, i.e. coevolution. Population 1 evolves ANGN connection weights, but for evaluating a given individual i (fitness function $f1(i)$) it needs a set of glial parameters (individuals of Population 2) to build a complete ANGN that can be tested through the fitness function MSE_{ij} (Eq. (6)). Similarly, Population 2 evolves ANGN glial parameters, but for evaluating a given individual j (fitness function $f2(j)$) it needs a set of connection weights (individuals of Population 1) to build a complete ANGN that can be tested through the fitness function MSE_{ij} (Eq. (6)). In other words, each population evolves according to the pipeline of a conventional GA but, in this case, every individual from one population needs individuals from the other one to compute its own fitness value. Afterwards, once the genetic operators have been applied to the individuals and their fitness values have been calculated, the replacement strategy operates normally and independently over each population to generate the next generation (in this case, the worst individuals according to their fitness are substituted by the newly created ones).

3.4. *EA design*

The whole algorithm is a hybrid-based iterative procedure (including unsupervised and supervised phases) that will continue until a minimal error or a maximum number of iterations ("generations") are reached. During the first iteration, the synaptic weights and neuron–glia parameters are randomly initialized and, henceforth, the astrocytes using the established neuron–glia parameters modify those initial weights depending on the neuronal activity. Once astrocytes have finished their work during such a generation, the CCEA simultaneously optimizes both sets (synaptic weights and neuron–glia parameters). The proposed CCEA co-evolves two different populations by means of two different GAs that cooperate during the calculation of the fitness function. One GA for the population that codifies the neuron–glia parameters and another one for the population that codifies the ANGN synaptic weights (w). The CCEA selects the suitable neuron–glia parameters in a way that artificial astrocytes can act and, after that, with the changes that astrocytes perform on the weights (w'), the artificial network produces a certain fitness value. That fitness value (MSE) is used by the CCEA to perform the optimization of the initial ANGN weights (w). Importantly, the changes made to the weights by the astrocytes (w') are not incorporated in the population as new individuals but they cause that the number of generations, needed by the CCEA to achieve zero or minimal error, is usually smaller than in the case of not having artificial astrocytes.

In the first GA, SBX crossover[19] and random mutation have been implemented. Both genetic operators are constrained to the range of possible values for the neuron–glia parameters as previously depicted in Table 1. In the second GA, we have employed random mutation and 1-point crossover. The reason to use that simple crossover was the reduced number of genes (5) and their integer nature.

In both populations the initial set of individuals is randomly generated considering the aforementioned problem restrictions. A steady-state reproduction[67] is employed, and elitism is considered in order to keep unchanged the best individual along the evolution process. In contrast to other optimization techniques, it is important to remark how simple is to codify and operate with the problem restrictions within an evolutionary framework.

4. Experiments

4.1. *Experimental setup*

With the purpose of evaluating the performance of the novel learning approach proposed in this work, the following methods will be compared under the same conditions: (a) ANGN trained by means of the hybrid method detailed in Sec. 2, with the neuron–glia parameters empirically established; and (b) ANGN trained with the proposed coevolutionary approach which includes both connection weights and neuron–glia parameters. From now on we will refer to them as ANGN and ANGN-CCGA, respectively.

In the two cases, the same kind of network was employed: fully connected feedforward ANNs with one hidden layer, without lateral connections and backpropagation.

In the case of ANGN-CCGA we carried out a preliminary experimentation including all possible combinations of the CCGA interaction parameters (see Sec. 3.2): sample size, selective bias and credit assignment. The combination of best and random individuals resulted to be the best selective bias. None of the proposed selection mechanisms outperformed it so we decided not to include them in the final learning approach. The best results were obtained by the combination 1 (best) – 2 (random). Thus, these sample size values were the interactions employed in the comparison against the ANGN.

Table 3 summarizes all the genetic operators and the corresponding parameter values for the learning algorithms under comparison. The operators and parameters employed by the ANGN were already established in the original ANGN proposal.[53] In the case of the coevolutionary approach, the table distinguishes between the two different GAs optimizing ANN weights and glia parameters.

The two approaches were challenged to solve five classification problems, obtained from the University

Table 3. Genetic parameters of the two approaches analyzed.

Genetic Params	ANGN	ANGN-CCGA	
	GA weights	GA weights	GA glia
Crossover op.	SBX ($\mu = 0.3$)	SBX ($\mu = 0.3$)	1-point
Crossover rate	90%	90%	90%
Mutation op.	Random	Random	Random
Mutation rate	10%	10%	10%
Population size	150	150	30
Elitism	YES	YES	YES
Selection	Montecarlo	Tournament	Tournament
Substitution	Darwinian	Darwinian	Darwinian
Sample size and selective bias	—	1 best – 2 random	
Credit assign.	—	minimum	

of California Irvine Machine Learning Repository[8] and the Knowledge Extraction based on Evolutionary Learning data set.[2] They were selected among a large number of possible data sets based on their different characteristics and complexities as follows:

- Breast Cancer (BC) problem: The presence of cancer from nine properties in 699 patients have to be predicted (i.e. nine inputs; a binary output);
- Ionosphere (IS) problem: The goal is to define "good" or "bad" radar signals according to the state of the IS by analyzing 34 characteristics of 351 radar signals (i.e. 34 inputs; a binary output);
- Mushroom (MU) problem: The task is to classify "edible" or "no edible" mushrooms according to 22 characteristics of 5644 mushrooms (i.e. 22 inputs; a binary output);
- PIMA (PI) problem: The task is to classify "diabetes negative" or "diabetes positive" patients according to 8 attributes of 768 instances (i.e. 8 inputs; a binary output);
- SONAR (SO) problem: The task is to detect a signal as "acuatic bomb" or "rock" according to 60 attributes of 208 instances (i.e. 60 inputs; a binary output).

The network architecture (see Table 4) was established according to the results obtained in previous works.[53,54] Notice that we are not looking for the optimal architecture but for a common one that allow us to compare the two methods under the same conditions. The stopping times in Table 4 were chosen such that they were the same for all models to compare and they were large enough to allow the convergence of all models. Note that this time is the worst time needed to finish any execution of the EA, i.e. any of the 10 independent runs with each of the 10 populations used by the EA. A larger time in MU can be explained by the number of input patterns: more than 10 times more in MU than the average of all other problems. In Table 5, we show the mean validation times where the best configuration was selected, in order to avoid overfitting. As one can easily see, the times of CCGA are much larger. The reasons for that are twofold. First, ANGN runs just one previously selected combination of parameters, while ANGN-CCGA has the capability to try thousands of combinations in only one run (6 N-G alg × 35 possible act-iter combinations ×36 possible decr-incr percentages = 7560 different configurations to try on each dataset). Second, the selected combination of parameters in ANGN was obtained after a tedious and very time-consuming preliminary study, not quantified in Tables 4 and 5. Importantly, the mean validation time in ANGN-CCGA corresponds to the configuration with the best validation value found so far, which does not rule out the possibility of finding in much less time a sufficiently good configuration of parameters.

Table 4. Network architecture and training time for each specific problem.

	One hidden layer	Stop time (min)
BC	9–7–1	667
IS	34–9–1	1750
MU	22–7–1	15,000
PI	8–4–1	500
SO	60–6–1	1917

Table 5. Mean validation time (in min).

Problem	ANGN				ANGN-CCGA
	2–4	2–6	3–6	3–8	
BC	1.03	2.25	1.17	4.41	128
IS	0.11	1.16	5.87	3.28	368
MU	4.96	9.99	8.07	15.15	6239
PI	0.93	0.28	0.15	0.26	119
SO	3.97	6.69	9.37	7.14	287

To secure independent results, 5x2cv cross-validation was implemented. Thus, 10 different datasets were obtained where each of them include: 25% training patterns, 25% validation patterns and 50% test patterns. It is important to emphasize the difficulty involved in this patterns division: 25% training and 50% test. Obviously, it has been the same configuration for all methods, but this is still to "make it difficult" to the algorithms, which could explain that the results obtained are maybe not as good as those obtained in the same datasets by other methods. The steady training, validation and test accuracies were measured after a time period that was previously established for each problem based on the time it takes for the different models to converge.

This time was the same for the two methods under comparison (see Table 4). The selected model to be tested corresponds to the best configuration obtained along the validation phase. Due to the

Table 6. Comparative study of the classification accuracy (training and validation values) of ANGN (with four different configurations) and ANGN-CCGA. The values depicted in the table refer to average performance (100 different runs), standard deviation and the existence of statistically significant ($^{*}P < 0.05$, $^{**}P < 0.01$ and $^{***}P < 0.001$).

	ANGN				ANGN-CCGA
	2–4	2–6	3–6	3–8	
			Training		
BC	$91.94 \pm 7.67^{***}$	$91.17 \pm 7.52^{***}$	$91.49 \pm 7.56^{***}$	$90.98 \pm 7.58^{***}$	94.20 ± 9.77
IS	$83.38 \pm 11.06^{***}$	$87.94 \pm 9.02^{***}$	$84.90 \pm 11.99^{***}$	$85.51 \pm 11.11^{***}$	92.11 ± 5.06
MU	98.22 ± 2.00	98.71 ± 1.29	98.87 ± 1.04	98.61 ± 2.29	98.55 ± 1.25
PI	$72.03 \pm 5.13^{***}$	$72.37 \pm 5.57^{***}$	$71.51 \pm 5.48^{***}$	$72.52 \pm 6.07^{***}$	79.06 ± 4.14
SO	$86.91 \pm 10.42^{***}$	$88.23 \pm 9.85^{***}$	$86.42 \pm 10.49^{***}$	$86.88 \pm 10.37^{***}$	93.12 ± 4.90
			Validation		
BC	$85.48 \pm 11.48^{***}$	$84.52 \pm 11.41^{**}$	$84.87 \pm 11.26^{***}$	$84.40 \pm 11.60^{***}$	93.50 ± 6.77
IS	$78.26 \pm 4.11^{***}$	$76.61 \pm 5.30^{***}$	$75.62 \pm 5.46^{***}$	$76.18 \pm 5.09^{***}$	90.57 ± 3.39
MU	$97.30 \pm 2.26^{***}$	97.86 ± 1.37	98.12 ± 1.27	97.73 ± 2.30	98.30 ± 1.26
PI	$67.34 \pm 2.86^{***}$	$67.74 \pm 2.40^{***}$	$67.01 \pm 2.64^{***}$	$67.20 \pm 2.92^{***}$	71.38 ± 3.22
SO	$65.29 \pm 6.14^{***}$	$66.71 \pm 6.70^{***}$	$66.08 \pm 7.01^{***}$	$66.82 \pm 6.11^{***}$	73.64 ± 4.18

Table 7. Comparative study of the classification accuracy (test values) of ANGN (with four different configurations) and ANGN-CCGA. The values depicted in the table refer to average performance (100 different runs), standard deviation and the existence of statistically significant ($^{*}P < 0.05$, $^{**}P < 0.01$ and $^{***}P < 0.001$).

	ANGN				ANGN-CCGA
	2–4	2–6	3–6	3–8	
BC	$83.59 \pm 16.73^{***}$	$82.27 \pm 14.95^{***}$	$83.28 \pm 16.57^{***}$	$81.63 \pm 14.30^{***}$	87.43 ± 12.94
IS	$75.71 \pm 4.69^{***}$	$73.74 \pm 6.50^{***}$	$73.97 \pm 5.81^{***}$	$73.65 \pm 5.69^{***}$	78.21 ± 7.43
MU	$97.26 \pm 2.12^{***}$	$97.72 \pm 1.39^{*}$	97.95 ± 1.25	97.65 ± 2.26	97.99 ± 1.88
PI	$63.46 \pm 3.88^{***}$	$63.42 \pm 4.23^{***}$	$63.38 \pm 4.16^{***}$	$63.67 \pm 4.14^{***}$	67.81 ± 5.16
SO	$65.29 \pm 8.93^{***}$	$65.51 \pm 7.85^{***}$	$65.61 \pm 7.55^{***}$	$64.41 \pm 7.81^{***}$	71.44 ± 7.12

stochastic nature of the two methods compared we run each of them 10 times using 10 different initial genetic populations, 100 runs in total due to the cross-validation set-up. Wilcoxon signed rank test[75] was used for corroborating statistical significance because the requirements for applying parametric statistical tests were not met.[20]

The two learning approaches were implemented in C programming language. All the experiments were run on the SVG supercomputer located in the Supercomputing Center of Galicia (CESGA).[c]

4.2. *Results and discussion*

Tables 6 and 7 show mean and standard deviation for the 100 runs (10 different data sets × 10 different initial genetic populations) of ANGN and ANGN-CCGA in training, validation and test. As expected, training performance is always better than validation. Also, the accuracy results in validation are slightly better than the ones obtained during the test phase. The best results, in terms of absolute numbers, were obtained by ANGN-CCGA in all cases but in MU during training.

If we focus only on the test results (see Table 7), our proposal achieves a higher classification accuracy compared with the mean values of the four different ANGN configuration tested (2–4, 2–6, 3–6, 3–8). Notice that for the latter comparison one single run of the ANGN-CCGA proposal is compared against four different runs of the ANGN approach which implies a total training time four times higher. In addition, Table 7 shows the test accuracy including

[c] http://www.cesga.es/en/.

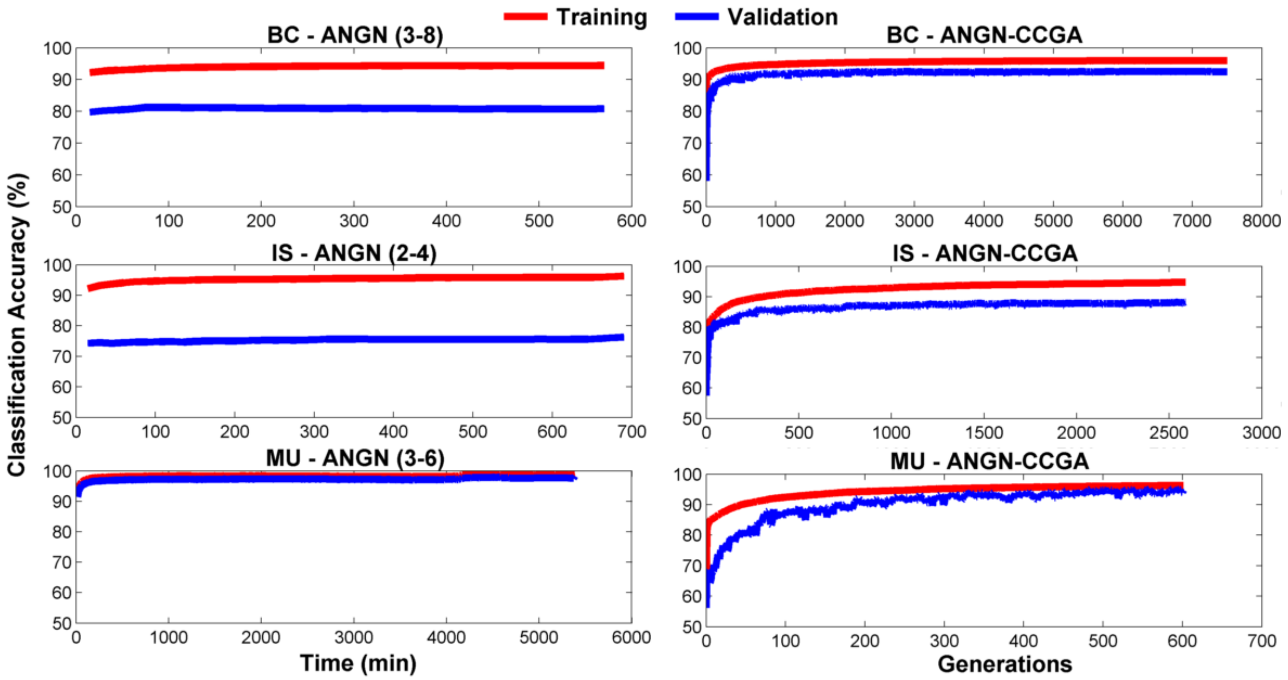

Fig. 5. Evolution in time of the classification accuracy for three cases in training and validation. First column represents results obtained by ANGN, with the combination used within brackets. Second column shows the results obtained by the ANGN-CCGA learning approach.

the degree of significance (p-values) according to the Wilcoxon test ($^{*}P < 0.05$, $^{**}P < 0.01$ and $^{***}P < 0.001$). For the given set of problems, ANGN-CCGA demonstrated its superiority: in 18 out of 20 comparisons the coevolutionary approach outperforms its counterpart. In 17 cases there are large statistically significant differences ($***$), while in another one the statistical significance corresponds to a p-value smaller than 0.05. In the remaining two cases, there are no statistically significant differences between the approaches. It is important to notice that the only case when there were no statistically significant differences (MU with 3–6 and 3–8) corresponds with the best classification results obtained so far. In other words, the margin of improvement is extremely low to detect differences, even if these, in absolute numerical terms, exist: again ANGN-CCGA obtained the best results with an accuracy of 97.99%.

Notice that this performance study is aimed to analyze the possible contribution of the new learning approach to the ANGN model. The low percentage of training patterns (25%) has affected without any doubt the classification accuracy of both ANGN and ANGN-CCGA models. However, it was not our intention to compare these models against the state-of-the-art classification models.

Figure 5 shows the evolution in time of the classification accuracy during training and validation in three representative cases. The first column of graphs represents ANGN (with the activation/iteration combination within brackets), while the second one shows the results obtained by the ANGN-CCGA. In the latter case, each generation takes a completely different amount of time depending on the set of parameters evaluated during that particular generation (for instance, a combination activation/iteration 2–3 takes much less time than a combination 8–9). Therefore, the x-axis in ANGN-CCGA refers to generations instead of minutes. One of the most interesting aspects of Fig. 5 is that it allows us to conclude that ANGN-CCGA has a much lower tendency to overtrain, as can be seen in BC and IS where the difference in accuracy of ANGN in training and validation is much larger. This phenomenon is not present in ANGN-CCGA. Also, it is important to notice that the example related to MU is an exception, since it corresponds to 1 of the 2 cases, out of 20, where the ANGN-CCGA did not

Table 8. Comparative study of the classification accuracy of ANN (trained using backpropagation) and ANGN-CCGA. The values depicted in the table refer to average performance (100 different runs), standard deviation and the existence of statistically significant ($^{*}P < 0.05$, $^{**}P < 0.01$ and $^{***}P < 0.001$). The asterisks are located over the technique whose results are statistically significantly worse.

	TRAINING		VALIDATION		TEST	
	ANN	ANGN-CCGA	ANN	ANGN-CCGA	ANN	ANGN-CCGA
BC	93.68 ± 14.55	94.20 ± 9.77	$87.03 \pm 15.47^{*}$	93.50 ± 6.77	$84.68 \pm 19.28^{*}$	87.43 ± 12.94
IS	93.73 ± 14.00	$92.11 \pm 5.06^{***}$	$85.19 \pm 13.19^{**}$	90.57 ± 3.39	77.43 ± 14.61	78.21 ± 7.43
MU	$95.05 \pm 14.05^{***}$	98.55 ± 1.25	$95.04 \pm 13.91^{***}$	98.30 ± 1.26	$94.78 \pm 14.05^{***}$	97.99 ± 1.88
PI	77.89 ± 7.91	79.06 ± 4.14	72.95 ± 4.34	$71.38 \pm 3.22^{***}$	71.26 ± 5.99	$67.81 \pm 5.16^{***}$
SO	85.40 ± 15.87	93.12 ± 4.90	$68.81 \pm 8.92^{***}$	73.64 ± 4.18	$65.66 \pm 10.65^{***}$	71.44 ± 7.12

obtain statistically significant better results than its counterpart. The second row (IS problem) displays a case where ANGN-CCGA is clearly better in validation but not in training in comparison to ANGN, thus showing a major generalization ability.

In order to have some reference about the performance of ANGN-CCGA in comparison to more classical approaches, Table 8 compares the performance of ANGN-CCGA with a classical ANN trained using Levenberg–Marquardt backpropagation (with the MATLAB default parameters). Nevertheless, it is fundamental to highlight that the main intention of this research is not to compare the performances of ANGN and other machine learning techniques, but to analyze the influence of artificial astrocytes within the ANN to improve the current information processing systems. We know that the architectures and different components we are employing are not always optimal, and we simply try to compare both approaches under equal conditions (by building the exact same networks including astrocytes). To do so, we keep the same initial weights, the same seeds of random number generation, the same sets of patterns, the same general structure (fully connected feedforward ANNs with one hidden layer without lateral connections), the same transference functions and architectures (see Table 4). The results obtained showed that ANGN-CCGA offers statistically significant better results in three problems (being also better in another one but with no statistically significant differences), while it is significantly worse in one. The ANN standard deviations are much larger than the ones of ANGN-CCGA: this indicates a more robust behavior, as ANGN-CCGA is able to produce more consistent results over different trials; the ANN performance is much more dependent on its random initialization. Given the results obtained with the five datasets tested, the overall performance of ANGN-CCGA can be considered competitive with the one offered by a traditional ANN.

5. Conclusions and Future Work

A novel learning approach for ANGNs based on CCGA has been presented. In contrast to existing learning methods our proposal does not need a manual tuning of the glial parameters. This new method is a step forward in the development of bio-inspired systems: not only combines artificial neurons and astrocytes, but adaptively evolves the system parameters thanks to a coevolutionary approach to solve classification problems. It has been demonstrated that the CCGA approach favors artificial astrocytes action in ANGNs. In particular, given an unbiased and rigorous experimental setup, our novel approach yielded significantly better results than its counterpart in 18 out of 20 experiments performed over five different classification problems. More problems were discarded due to the computational time required and the conclusive results obtained.

There is still an open field of research in order to mimic closer the neuro–glia system with the objectives of increasing the understanding of the biological systems and, at the same time, to provide the machine learning community with a competitive novel approach. In particular, as the biological model suggests, the artificial model could be benefited by the inclusion of one or more artificial astrocytes modifying the activity of each artificial neuron. Indeed, it has been already studied that the most advanced species present a higher amount of astrocytes cells and also a higher percentage of this type of cells against neurons. Other research directions include the implementation of more recent evolutionary methods such as Differential Evolution[15] or to test alternative approaches to the coevolution of several populations with the use of niching methods.[39] Finally, a major research direction within this project would consist on showing the appropriateness and effectiveness of this learning algorithm when solving other tasks different from classification and considering other neuron–astrocytes interactions.

Acknowledgments

We would like to thank Dr. Araque and IJNS reviewers for their valuable comments. This work is supported by the Spanish Ministerio de Ciencia e Innovación including EDRF fundings (TIN2009-07707, RD07/0067/005), the Spanish Ministerio de Industria, Turismo y Comercio (Ref. TSI-020110-2009-53), the Iberoamerican program CYTED (Red Ibero-NBIC 209RT0366) and by the Galician Council (REGICC 2009/58, 10SIN105004PR, 10MDS014CT). Pablo Mesejo is funded by the European Commission (Marie Curie ITN MIBISOC, FP7 PEOPLE-ITN-2008, GA No. 238819). Dr. Ibáñez's work has been supported by Spanish MINECO Juan de la Cierva Fellowship JCI-2012-15359. All the experiments were supported by the computing resources at the Supercomputation Center of Galicia (CESGA), Spain.

References

1. C. Agulhon, T. A. Fiacco and K. D. McCarthy, Hippocampal short- and long-term plasticity are not modulated by astrocyte Ca^{2+} signaling, *Science* **327**(5970) (2010) 1250–1254.
2. J. Alcalá-fdez, L. Sánchez, S. García, M. J. D. Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández and F. Herrera, KEEL: A software tool to assess evolutionary algorithms to data mining problems. *Soft Computing* **13**(3) (2009) 307–318, doi: 10.1007/s00500-008-0323-y.
3. L. M. Almeida and T. B. Ludermir, A multi-objective memetic and hybrid methodology for optimizing the parameters and performance of artificial neural networks, *Neurocomputing* **73**(7–9) (2010) 1438–1450.
4. A. Alvarellos, A. Pazos and A. B. Porto, Computational models of neuron-astrocyte interactions lead to improved efficacy in the performance of neural networks, *Comput. Math. Methods Med.* **2012** (2012).
5. A. Araque, G. Carmignoto, P. G. Haydon, S. H. Oliet, R. Robitaille and A. Volterra, Gliotransmitters travel in time and space, *Neuron* **81**(4) (2014) 728–739.
6. A. Araque, G. Carmignoto and P. G. Haydon, Dynamic signaling between astrocytes and neurons, *Annu. Rev. Physiol.* **63**(1) (2001) 795–813.
7. A. Araque, V. Parpura, R. P. Sanzgiri and P. G. Haydon, Tripartite synapses: Glia, the unacknowledged partner, *Trends Neurosci.* **22**(5) (1999) 208–215.
8. M. Lichman, UCI machine learning repository. (http://archive.ics.uci.edu.ml). Irvine, CA: University of California, School of Information and Computer Science.
9. T. Bäck, D. B. Fogel and Z. Michalewicz, *Handbook of Evolutionary Computation* (IOP Publishing Ltd and Oxford University Press, 1997).
10. C. M. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, Inc., 1995).
11. A. Blum and R. L. Rivest, Training a 3-node neural network is np-complete, *Neural Netw.* **5**(1) (1992) 117–127.
12. L. Bull, On coevolutionary genetic algorithms, *Soft Comput.* **5**(3) (2001) 201–207.
13. B. Campomanes-Álvarez, O. Cordón and S. Damas, Evolutionary multi-objective optimization for mesh simplification of 3D open models, *Integr. Comput. Aided Eng.* **20**(4) (2013) 375–390.
14. R. Chandra, M. R. Frean and M. Zhang, An encoding scheme for cooperative coevolutionary feedforward neural networks, *Australasian Conf. Artificial Intelligence*, Lecture Notes in Computer Science, Vol. 6464 (Springer, 2010), pp. 253–262.
15. S. Das and P. Suganthan, Differential evolution: A survey of the state-of-the-art, *IEEE Trans. Evol. Comput.* **15**(1) (2011) 4–31.
16. M. De Pittà, V. Volman, H. Berry and E. Ben Jacob, A tale of two stories: Astrocyte regulation of synaptic depression and facilitation, *PLoS Comput. Biol.* **7**(12) (2011) e1002.
17. M. De Pittà, V. Volman, H. Berry, V. Parpura, A. Volterra and E. Ben-Jacob, Computational quest for understanding the role of astrocyte signaling in synaptic transmission and plasticity, *Front. Comput. Neurosci.* **6**(98) (2012) 1–25.
18. D. F. de Sevilla, M. Fuenzalida, A. Porto and W. Buño, Selective shunting of the NMDA EPSP component by the slow after hyperpolarization in rat CA1 pyramidal neurons, *J. Neurophysiol.* **97**(5) (2007) 3242–3255.
19. K. Deb and R. B. Agrawal, Simulated binary crossover for continuous search space, *Complex Syst.* **9** (1995) 115–148.
20. J. Derrac, S. García, D. Molina and F. Herrera, A practical tutorial on the use of nonparametric

statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm Evol. Comput.* **1**(1) (2011) 3–18.

21. T. A. Fiacco, C. Agulhon, S. R. Taves, J. Petravicz, K. B. Casper, X. Dong, J. Chen and K. D. McCarthy, Selective stimulation of astrocyte calcium *in situ* does not affect neuronal excitatory synaptic activity, *Neuron* **54**(4) (2007) 611–626.
22. J. Friedrich, R. Urbancziky and W. Senn, Code-specific learning rules improve action selection by populations of spiking neurons, *Int. J. Neural Syst.* **24**(5) (2014) 1450002.
23. N. García-Pedrajas, C. Hervas-Martinez and D. Ortiz-Boyer, Cooperative coevolution of artificial neural network ensembles for pattern classification, *IEEE Trans. Evol. Comput.* **9**(3) (2005) 271–302.
24. N. García-Pedrajas and D. Ortiz-Boyer, A cooperative constructive method for neural networks for pattern recognition, *Pattern Recogn.* **40**(1) (2007) 80–98.
25. N. García-Pedrajas, D. Ortiz-Boyer and C. Hervás-Martínez, Cooperative coevolution of generalized multi-layer perceptrons, *Neurocomputing* **56** (2004) 257–283.
26. S. Ghosh-Dastidar and H. Adeli, A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection, *Int. J. Neural Syst.* **22**(10) (2009) 1419–1431.
27. S. Ghosh-Dastidar and H. Adeli, Spiking neural networks, *Int. J. Neural Syst.* **19**(4) (2009) 295–308.
28. F. Glover, M. Laguna and R. Marti, Scatter search, in *Advances in Evolutionary Computing*, eds. A. Ghosh and S. Tsutsui, *Natural Computing Series* (Springer Berlin Heidelberg, 2003), pp. 519–537.
29. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley, 1989).
30. F. Gomez, J. Schmidhuber and R. Miikkulainen, Accelerated neural evolution through cooperatively coevolved synapses, *J. Mach. Learn. Res.* **9** (2008) 937–965.
31. N. B. Hamilton and D. Attwell, Do astrocytes really exocytose neurotransmitters? *Nat. Rev. Neurosci.* **11**(4) (2010) 227–238.
32. H. H. Hoos and T. Stützle, *Stochastic Local Search Foundations and Applications* (Elsevier, 2004).
33. C. Ikuta, Y. Uwate and Y. Nishio, Investigation of Multi-layer perceptron with pulse glial chain based on individual inactivity period, *Int. Joint Conf. Neural Networks* (*IJCNN'14*) (2014), pp. 1638–1643.
34. M. M. Islam and X. Yao, Evolving artificial neural network ensembles, in *Computational Intelligence*: *A Compendium*, Vol. 115 (Springer, 2008), pp. 851–880.
35. T. Kathirvalavakumar and S. Jeyaseeli Subavathi, Neighborhood based modified backpropagation algorithm using adaptive learning parameters for training feedforward neural networks, *Neurocomputing* **72**(16–18) (2009) 3915–3921.
36. D. Le Ray, D. F. de Sevilla, A. Porto, M. Fuenzalida and W. Buño, Heterosynaptic metaplastic regulation of synaptic efficacy in ca1 pyramidal neurons of rat hippocampus, *Hippocampus* **14**(8) (2004) 1011–25.
37. M.-L. Linne, R. Havela, A. Saudargien and L. McDaid, Modeling astrocyte-neuron interactions in a tripartite synapse, *BMC Neurosci.* **15**(1) (2014) 98.
38. W. S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biophys.* **5** (1943) 115–133.
39. O. J. Mengshoel and D. E. Goldberg, The crowding approach to niching in genetic algorithms, *Evol. Comput.* **16**(3) (2008) 315–354.
40. P. Mesejo, R. Ugolotti, F. Di Cunto, M. Giacobini and S. Cagnoni, Automatic hippocampus localization in histological images using differential evolution-based deformable models, *Pattern Recogn. Lett.* **34**(3) (2013) 299–307.
41. R. Min, M. Santello and T. Nevian, The computational power of astrocyte mediated synaptic plasticity, *Front. Comput. Neurosci.* **6**(93) (2012) 1–15.
42. B. Mitterauer, Qualitative information processing in tripartite synapses: A hypothetical model, *Cogn. Comput.* **4**(2) (2012) 181–194.
43. S. Nadkarni and P. Jung, Modeling synaptic transmission of the tripartite synapse, *Phys. Biol.* **4**(1) (2007) 1–9.
44. J. V. Neumann, *The Computer and the Brain* (Yale University Press, 1958).
45. M. H. Nguyen, H. A. Abbass and R. I. McKay, A novel mixture of experts model based on cooperative coevolution, *Neurocomputing* **70**(1–3) (2006) 155–163.
46. A. Panatier, J. Vallee, M. Haber, K. K. Murai, J. C. Lacaille and R. Robitaille, Astrocytes are endogenous regulators of basal transmission at central synapses, *Cell* **146**(5) (2011) 785–798.
47. J. Paredis, Coevolutionary computation, *Artif. Life* **2**(4) (1995) 355–375.
48. C. Pena-Reyes and M. Sipper, Fuzzy coco: A cooperative-coevolutionary approach to fuzzy modeling, *IEEE Trans. Fuzzy Syst.* **9**(5) (2001) 727–737.
49. G. Perea and A. Araque, Astrocytes potentiate transmitter release at single hippocampal synapses, *Science* **317**(5841) (2007) 1083–1086.
50. G. Perea, M. Navarrete and A. Araque, Tripartite synapses: Astrocytes process and control synaptic information, *Trends Neurosci.* **32** (2009) 421–431.
51. G. Perea and A. Araque, Communication between astrocytes and neurons: A complex language, *J. Physiol. Paris* **96**(34) (2002) 199–207.
52. J. Petravicz, T. A. Fiacco and K. D. McCarthy, Loss of IP3 receptor-dependent Ca^{2+} increases in hippocampal astrocytes does not affect baseline CA1

pyramidal neuron synaptic activity, *J. Neurosci.* **28**(19) (2008) 4967–4973.

53. A. Porto, A. Araque, J. R. Rabuñal, J. Dorado and A. Pazos, A new hybrid evolutionary mechanism based on unsupervised learning for connectionist systems, *Neurocomputing* **70**(16–18) (2007) 2799–2808.
54. A. Porto, N. Veiguela, P. Mesejo, M. Navarrete, A. Alvarellos, O. Ibáñez, A. Pazos and A. Araque, Artificial astrocytes improve neural network performance, *PloS One* **6**(4) (2011) 1–8.
55. A. Porto, A. Pazos and A. Araque, Artificial neural networks based on brain circuits behavior and genetic algorithms, *IWANN*, Lecture Notes in Computer Science Vol. 3512 (Springer, 2005), pp. 99–106.
56. M. A. Potter and K. A. De Jong, Cooperative coevolution: An architecture for evolving coadapted subcomponents, *Evol. Comput.* **8**(1) (2000) 1–29.
57. O. Reyes, C. Morell and S. Ventura, Evolutionary feature weighting to improve the performance of multi-label lazy algorithms, *Integr. Comput. Aided Eng.* **21**(4) (2014) 339–354.
58. D. Rivero, J. Dorado, J. Rabuñal and A. Pazos, Generation and simplification of artificial neural networks by means of genetic programming, *Neurocomputing* **73**(16–18) (2010) 3200–3223.
59. F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, *Psychol. Rev.* **65**(6) (1958) 386–408.
60. C. D. Rosin and R. K. Belew, New methods for competitive coevolution, *Evol. Comput.* **5**(1) (1997) 1–29.
61. J. Rosselló, V. Canals, A. Oliver and A. Morro, Studying the role of synchronized and chaotic spiking neural ensembles in neural information processing, *Int. J. Neural Syst.* **24**(5) (2014) 1430003.
62. N. Siddique and H. Adeli, *Computational Intelligence*: *Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing* (John Wiley & Sons, 2013).
63. W. Sun, E. McConnell, J. F. Pare, Q. Xu, M. Chen, W. Peng, D. Lovatt, X. Han, Y. Smith and M. Nedergaard, Glutamate-dependent neuroglial calcium signaling differs between young and adult brain, *Science* **339**(6116) (2013) 197–200.
64. E. Talbi, *Metaheuristics*: *From Design to Implementation* (Wiley, 2009).
65. J. Tian, M. Li, F. Chen and J. Kou, Coevolutionary learning of neural network ensemble for complex classification tasks, *Pattern Recogn.* **45**(4) (2012) 1373–1385.
66. G. Valenza, L. Tedesco, A. Lanata, D. De Rossi and E. Scilingo, Novel spiking neuron-astrocyte networks based on nonlinear transistor-like models of tripartite synapses, *Engineering in Medicine and Biology Society* (*EMBC'13*) (2013), pp. 6559–6562.
67. F. Vavak and T. C. Fogarty, Comparison of steady state and generational genetic algorithms for use in nonstationary environments, *IEEE Int. Conf. Evolutionary Computation* (*CEC*) (1996), pp. 192–195.
68. A. Volterra, N. Liaudet and I. Savtchouk, Astrocyte Ca^{2+} signalling: An unexpected complexity, *Nat. Rev. Neurosci.* **15**(5) (2014) 327–335.
69. A. Volterra, P. J. Magistretti and P. G. Haydon, *The Tripartite Synapse*: *Glia in Synaptic Transmission* (Oxford University Press Inc., 2002).
70. J. Wade, L. McDaid, J. Harkin, V. Crunelli and S. Kelso, Self-repair in a bidirectionally coupled astrocyte-neuron (AN) system based on retrograde signaling, *Front. Comput. Neurosci.* **6** (2012) 76.
71. J. J. Wade, L. J. McDaid, J. Harkin, V. Crunelli and J. A. Kelso, Bidirectional coupling between astrocytes and neurons mediates learning and dynamic coordination in the brain: A multiple modeling approach, *PLoS One* **6**(12) (2011) e29445.
72. G. Wallach, J. Lallouette, N. Herzog, M. De Pitta, E. Ben Jacob, H. Berry and Y. Hanein, Glutamate mediated astrocytic filtering of neuronal activity, *PLoS Comput. Biol.* **10**(12) (2014) e1003964.
73. R. P. Wiegand, An analysis of cooperative coevolutionary algorithms, Ph.D. thesis, George Mason University, Fairfax, Virginia (2003).
74. R. P. Wiegand, W. Liles and K. De Jong, An empirical analysis of collaboration methods in cooperative coevolutionary algorithms, in *Proc Genetic and Evolutionary Computation Conf.* (2001), pp. 1235–1242.
75. F. Wilcoxon, Individual comparisons by ranking methods, *Biometr. Bull.* **1**(6) (1945) 80–83.
76. X. Yao, Evolving artificial neural networks, *Proc. IEEE* **87** (1999) 1423–1447.
77. G. Zhang, H. Rong, F. Neri and M. Perez-Jimenez, An optimization spiking neural P system for approximately solving combinatorial optimization problems, *Int. J. Neural Syst.* **24**(5) (2014) 1–16.