[Array 14 (2022) 100173](https://doi.org/10.1016/j.array.2022.100173)

|  |  |  |
| --- | --- | --- |
|  | Contents lists available at [ScienceDirect](http://www.elsevier.com/locate/array) |  |
| Array |
| journal homepage: [www.elsevier.com/locate/array](http://www.elsevier.com/locate/array) |
|  | | |

Optimising Multilayer Perceptron weights and biases through a Cellular Genetic Algorithm for medical data classification   
Matías Gabriel Rojasa,∗, Ana Carolina Oliveraa,b, Pablo Javier Vidala,b   
a *Instituto Universitario para las Tecnologías de la Información y las Comunicaciones, Consejo Nacional de Investigaciones Científicas y Técnicas, Universidad Nacional de Cuyo, Padre Jorge Contreras 1300, Mendoza, M5502JMA, Mendoza, Argentina*   
b *Facultad de Ingeniería, Universidad Nacional de Cuyo, Centro Universitario, Mendoza, M5502JMA, Mendoza, Argentina*

|  |  |  |
| --- | --- | --- |
| A R T I C L E | I N F O | A B S T R A C T |
| *Keywords:*  Multilayer Perceptron  Training methods  Cellular Genetic Algorithm Metaheuristics  Medical data classification | | In recent years, technology in medicine has shown a significant advance due to artificial intelligence becoming a framework to make accurate medical diagnoses. Models like Multilayer Perceptrons (MLPs) can detect implicit patterns in data, allowing identifying patients conditions that cannot be seen easily. MLPs consist of biased neurons arranged in layers, connected by weighted connections. Their effectiveness depends on finding the optimal weights and biases that reduce the classification error, which is usually done by using the Back Propagation algorithm (BP). But BP has several disadvantages that could provoke the MLP not to learn. Metaheuristics are alternatives to BP that reach high-quality solutions without using many computational resources. In this work, the Cellular Genetic Algorithm (CGA) with a specially designed crossover operator called Damped Crossover (DX), is proposed to optimise weights and biases of the MLP to classify medical data. When compared against state-of-the-art algorithms, the CGA configured with DX obtained the minimal Mean Square Error value in three out of the five considered medical datasets and was the quickest algorithm with four datasets, showing a better balance between time consumed and optimisation performance. Additionally, it is competitive in enhancing classification quality, reaching the best accuracy with two datasets and the second-best accuracy with two of the remaining. |

**1. Introduction**

Nowadays, it is impossible to imagine advances in medicine without talking about artificial intelligence. The incredible amount of data from different sources, such as medical images, data from clinical examinations, sensors and many others, outperforms by far the human capacity to process and analyse them [1]. For example, an average radiologist technician analyses about 215,000 radiography in about 40 years, while an artificial intelligence method processes that amount in about an hour [2].

Artificial Neural Networks (ANNs) undoubtedly are one of the arti-ficial intelligence methods that more contributions to the medical field have reported [3]. Examples of applications of ANNs to medicine are: diagnosis of diseases [4,5], prediction of treatments behaviour [6–8] and preventive medicine [9,10].

Multilayer Perceptron (MLP) [11] is a kind of ANN in which calculus units, called neurons, are organised in three types of layers. Each neuron is connected to all the neurons of the following layer, and data flow from the first to the last layer. Connections between neurons have a weight representing the strength of the linkage. In addition, both

hidden and output neurons have a bias that acts as a threshold of activation of the neuron.

What became attractive from the MLP was its ability to be a universal classifier that adapts to different distributions, features and complexities of data [12]. This quality is highly desirable in the medical field, considering that medical data can have noise, be imbalanced in the distribution of the classes and can have errors of registration [13]. The MLP effectiveness depends on its learning process, which iden-tifies the weights and biases values that minimise the classification error of training samples. The Back Propagation Algorithm (BP) is the standard way to make the MLP learn. However, BP has several weaknesses that can lead to a divergence in the MLP learning process, such as, a tendency to get stuck in local optima or dependency on initial values of hyperparameters [14,15].

In recent years, metaheuristics have gained attention as alterna-tives to the BP method. They are iterative algorithms able to find high-quality solutions in a reasonable time. One of their highlighted characteristics is that they can be applied to different kinds of prob-lems without needing specific knowledge [16,17], which makes them

∗ Corresponding au[thor.](mailto:mrojas@mendoza-conicet.gob.ar)

*E-mail addresses:* [mrojas@mendoza-conicet.gob.ar](mailto:mrojas@mendoza-conicet.gob.ar) (M.G. Rojas), [acolivera@conicet.gov.ar](mailto:acolivera@conicet.gov.ar) (A.C. Olivera), [pjvidal@conicet.gov.ar](mailto:pjvidal@conicet.gov.ar) (P.J. Vidal).

<https://doi.org/10.1016/j.array.2022.100173>  
[Received 24 January 2022; Received in revis](https://doi.org/10.1016/j.array.2022.100173)ed form 22 April 2022; Accepted 23 April 2022   
Available online 30 April 2022   
[2590-0056](http://creativecommons.org/licenses/by-nc-nd/4.0/)/© 2022 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

*M.G. Rojas et al.*  *Array 14 (2022) 100173*

|  |  |  |
| --- | --- | --- |
| **Nomenclature**   |  |  | | --- | --- | | *𝛽𝑘*  *𝜂*  **𝐁**  **𝐋**  **𝐖**  *𝜔𝐻 𝑖,𝑗*  *𝜔𝑂 𝑗,*1  *𝐵𝐼𝑖*  *𝑙𝑏*  *𝑚*  *𝑛*  *𝑂𝑓*1*𝑖*  *𝑂𝑓*2*𝑖*  *𝑝*1  *𝑝*2  *𝑟*  *𝑆*  *𝑠𝑜𝑙𝑡*+1 *𝑖*  *𝑠𝑜𝑙𝑡 𝑖*  *𝑆𝑢𝑚𝑂* 1  *𝑆𝑢𝑚𝐻 𝑗*  *𝑇*  *𝑡*  *𝑢𝑏*  *𝑋𝑖*  *𝑌*  *𝑦𝐻 𝑗*  *𝑦𝑂* 1  *𝑦𝑙*  ABC  ALO  ANN  AX  BAT  BBO  BOA | Bias for a neuron *𝑘* of the hidden or output layer.  Distribution index.  Set of biases of the whole MLP.  Set of training samples of a given dataset. Set of weights of the whole MLP.  A weight from the input neuron *𝑖* to the hidden neuron *𝑗* in the MLP.  A weight from the hidden neuron *𝑗* to the output neuron in the MLP.  Element *𝑖* of the best solution in the population.  Lower variables boundary.  number of hidden neurons.  Number of input neurons.  Element *𝑖* of the first offspring of the crossover operator.  Element *𝑖* of the second offspring of the crossover operator.  First parent for the crossover operator.  Second parent for the crossover operator. Random number.  A candidate solution found by an algorithm. Value of the position *𝑖* of the solution *𝑠𝑜𝑙* at evaluation *𝑡* + 1.  Value of the position *𝑖* of the solution *𝑠𝑜𝑙* at evaluation *𝑡*.  Result of the summation operation for the output neuron.  Result of the summation operation for the hidden neuron *𝑗*.  number of fitness evaluations to Total  perform.  Number of performed fitness evaluations. Upper variables boundary.  An input to the MLP.  Binary output of the MLP.  Output of the hidden neuron *𝑗*.  Output of the unique output neuron.  Expected output for a sample *𝑙* of the training set.  Artificial Bee Colony.  Ant Lion Optimiser.  Artificial Neural Network.  Adjusted Crossover.  Bat Algorithm.  Biogeography-Based Optimisation Algorithm.  Butterfly Optimisation Algorithm. | |

suitable for solving complex optimisation problems such as the learning process of the MLP [18,19]. Studies have demonstrated that meta-heuristics can perform well in training MLP models, even when a large number of weights and biases must be optimised [20,21].

Cellular Genetic Algorithm (CGA) [22] is a metaheuristic based on canonical Genetic Algorithm (GA), which works with a decentralised population where genetic operators act over a small overlapped sub-population per time. These features improve the performance of the

2

*M.G. Rojas et al.*  *Array 14 (2022) 100173*

3. Different configurations of the DX are evaluated against well-known genetic operators for the CGA. Five benchmark medical datasets were considered for experiments.

4. The CGA is compared deeply against state-of-the-art algorithms previously utilised for optimising a MLP. Comparisons are made observing how much each one improves the classification quality of the MLP.

This paper is organised as follows. Section 2 introduces the MLP, gives a notion about the traditional ways of training, defines the problem formally and presents the related works in literature. Section 3 describes the CGA, the representation of the solution and the DX. Section 4 is about the experiments configuration, the datasets used for tests, the state-of-the-art algorithms and genetic operators used in experiments and the metrics to evaluate the quality of classification. Results and their analysis are shown in Section 5. Finally, conclusions and future work are presented in Section 6.

**2. Multilayer perceptron**

Multilayer Perceptron (MLP) is an Artificial Neural Network (ANN) belonging to the feed-forward neural network family. The MLP has a set of processing units called neurons that transform data to get an expected output [27].

Internally, neurons are organised by three well-differentiated layers. The first layer contains the input neurons, which receives the input data and redirect them to the following layer. The number of input neurons is usually the same as the number of features. The second layer, called the hidden layer, contains neurons that map the data using mathematical functions. An MLP can be configured with one or more hidden layers, according to the complexity of the problem. Finally, the output layer receives the data transformed by the hidden layer and returns a result. The amount of neurons in the output layer depends on the codification of the expected result.

MLP is hierarchical and fully connected, meaning that neurons of one layer interact with all the neurons of the following layer by weighted connections, e.g., each input neuron is connected to all the neurons in the hidden layer.

Weights (*𝜔*) of each connection indicates how strong is the con-nection between two given neurons. In addition, hidden and output neurons have an element called bias (*𝛽*), which is a threshold to adjust the prediction by conditioning the neuron output. Depending on the *𝛽* value, the response of a neuron will be excitatory (positive) or inhibitory (negative) [28,29]. The learning process of the MLP consist of finding the optimal set of weights and biases.

An structure of an MLP with *𝑛* input neurons, one hidden layer with *𝑚* neurons and one output neuron is presented in Fig. 1. Weights for connections from input to hidden layer are represented by *𝜔𝐻 𝑖,𝑗*and weights for connections from hidden to output layer are showed as

*𝜔𝑂 𝑗,*1, with *𝑖* = {1*,* … *, 𝑛*} and *𝑗* = {1*,* … *, 𝑚*}. Biases appear as *𝛽𝑘* with *𝑘* = {1*,* … *, 𝑚* + 1}.

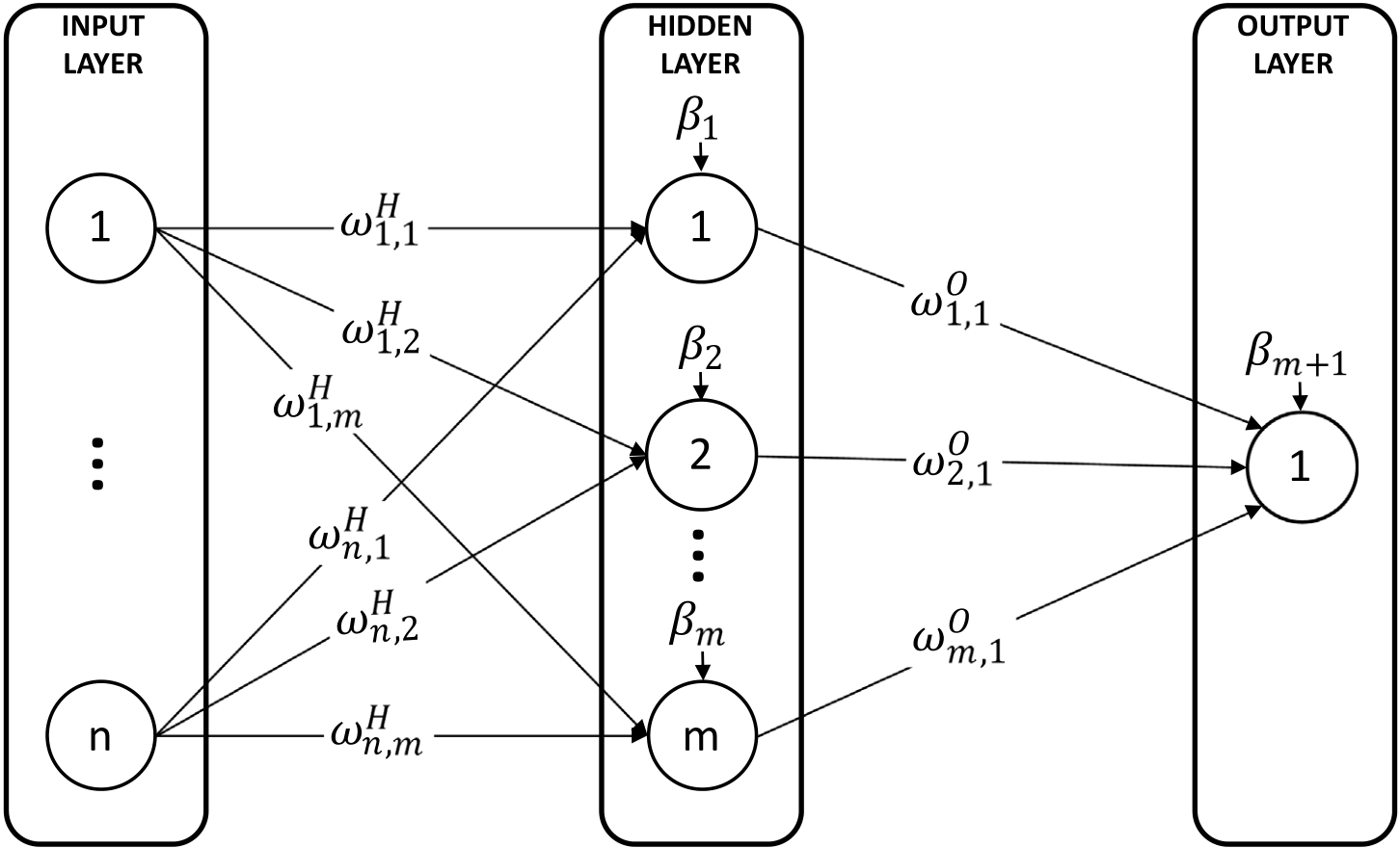
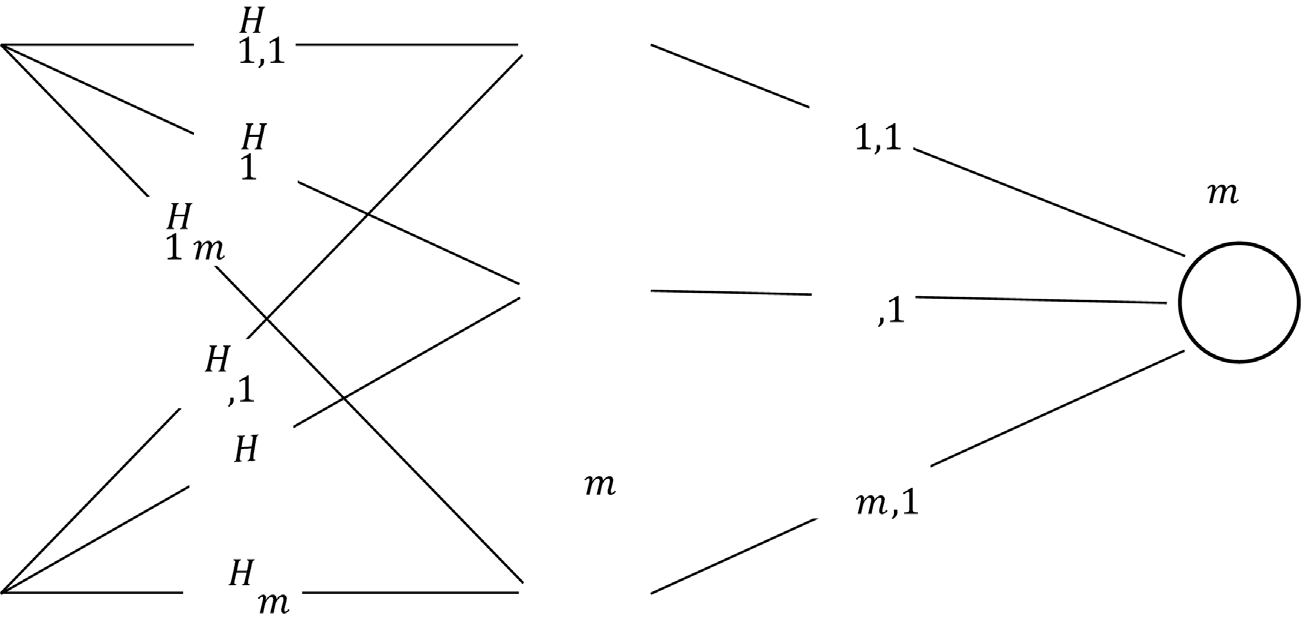
Hidden neurons transform data by performing two operations: Sum-mation and Activation. The former is the sum of the product between the outputs of neurons from the previous layer and the weights of the connections, added to the correspondent bias. Eq. (1) is used to apply summation on a given neuron *𝑗* of the hidden layer.

*𝑆𝑢𝑚𝐻 𝑗*=∑*𝜔𝐻 𝑖,𝑗*× *𝑋𝑖*+ *𝛽𝑗*  (1)

where *𝜔𝐻 𝑖,𝑗*is the weight of the connection between an input neuron *𝑖* of

the input layer and the neuron *𝑗* of the hidden layer. *𝑋𝑖* is the output of neuron *𝑖* that feeds the neuron *𝑗*, and *𝛽𝑗* is the bias of the neuron *𝑗*. The activation operation applies a mathematical function to map the result of the summation operation. This function is known as the

3



*M.G. Rojas et al.*  *Array 14 (2022) 100173*

**Fig. 1.** Example of structure of a Multilayer Perceptron (MLP).

One of the most conventional training methods for MLP is Back Propagation (BP) [31]. It starts setting random values for weights and biases. Classified samples (referred to as training set) are presented to the MLP to get an output value. Then, the error between the obtained and the desired values is calculated and propagated backwards to correct weights and biases. These steps are repeated until an acceptable error is reached [29].

The basic BP algorithm uses a first-order gradient descent for the optimisation of the MLP. Other existent approaches for optimising MLP are conjugate gradient [32] that is based on a second-order minimisation method, Quasi-Newton Method [33], Gauss–Newton [34] or Levenberg–Marquardt [35] that is based on the approximation by least-squares [15].

Although conventional approaches have shown to be effective in most of the problems they were applied to, there were situations in which they stuck in the same error value of MLP during extended peri-ods or even stuck in local optima. Furthermore, their success strongly depends on the initial values of weights, the values of momentum and the learning rate, which can provoke divergence if they are not right defined [14]. Finally, conventional methods put aside biases, focusing just on the values of the weights [15,36].

Considering the disadvantages of traditional methods, in this paper, an MLP optimiser based on the Cellular Genetic Algorithm is proposed to determine the optimal combination of weights and biases values to improve the classification quality.

*2.3. Related works*

In the latest years, it has been demonstrated that metaheuristics are able to be applied for training MLPs, reaching even better results than traditional mathematical methods [37]. This motivated different authors to train MLPs by metaheuristics for different problems in real life, getting featured results.

The work by Kaveh et al. [38] uses the Biogeography-Based Op-timisation algorithm (BBO) to train an MLP that classifies sonar data into three different classes: noises, reverberation, and clutter. A novel mutation operator is introduced in this work to enhance the exploration capability of the BBO. Results have demonstrated that the proposal of new operators can positively impact the behaviour of the algorithm, increasing the resulting classification performance. Qiao et al. [39] also worked with sonar data, proposing a modified Whale Optimisation

4

*M.G. Rojas et al.*  *Array 14 (2022) 100173*

is proposed. Another bio-inspired algorithm in the literature is the pro-posal by Das et al. [44] in that a Velocity Enhanced Whale Optimisation Algorithm (VEWOA) trains an ANN to classify data related to breast cancer, cervical cancer, and lung cancer. The VEWOA raises that each whale has to have a velocity, calculated as in PSO, where positions of the best and the previous positions of particles are considered. Results were compared against different machine learning approaches and the canonical WOA, making it difficult to observe whether this approach enhances the performance of metaheuristics specifically designed for MLP training.

Kumar et al. [45] propose an Inertia Motivated Gray Wolf Optimisa-tion Algorithm (IMGWO) that trains an MLP to classify data concerning breast cancer, heart disease, hepatitis and Parkinson’s disease. The IMGWO introduces a new method of calculating the balance between exploration and exploitation using a non-linear function. It also pro-poses to use velocity concepts similar to the PSO. The number of eval-uations required for the IMGWO to converge is significantly higher than the one used in the literature, indicating that those changes could have a negative effect on the convergence capacity of the algorithm. The IMGWO performed better than the canonical version of GA, PSO, and GWO. Despite this, comparisons didn’t make with the metaheuristics prepared to train MLPs, resulting in an unfair comparison.

Sharifi et al. [46] compared the GA and the PSO in defining the initial weights and biases of an MLP for detecting thyroid func-tional disease. After the initialisation phase, the MLP is trained by the Levenberg–Marquardt method. Results demonstrated that GA and PSO could contribute to accurate diagnoses, being the GA better than the PSO at improving the classification quality.

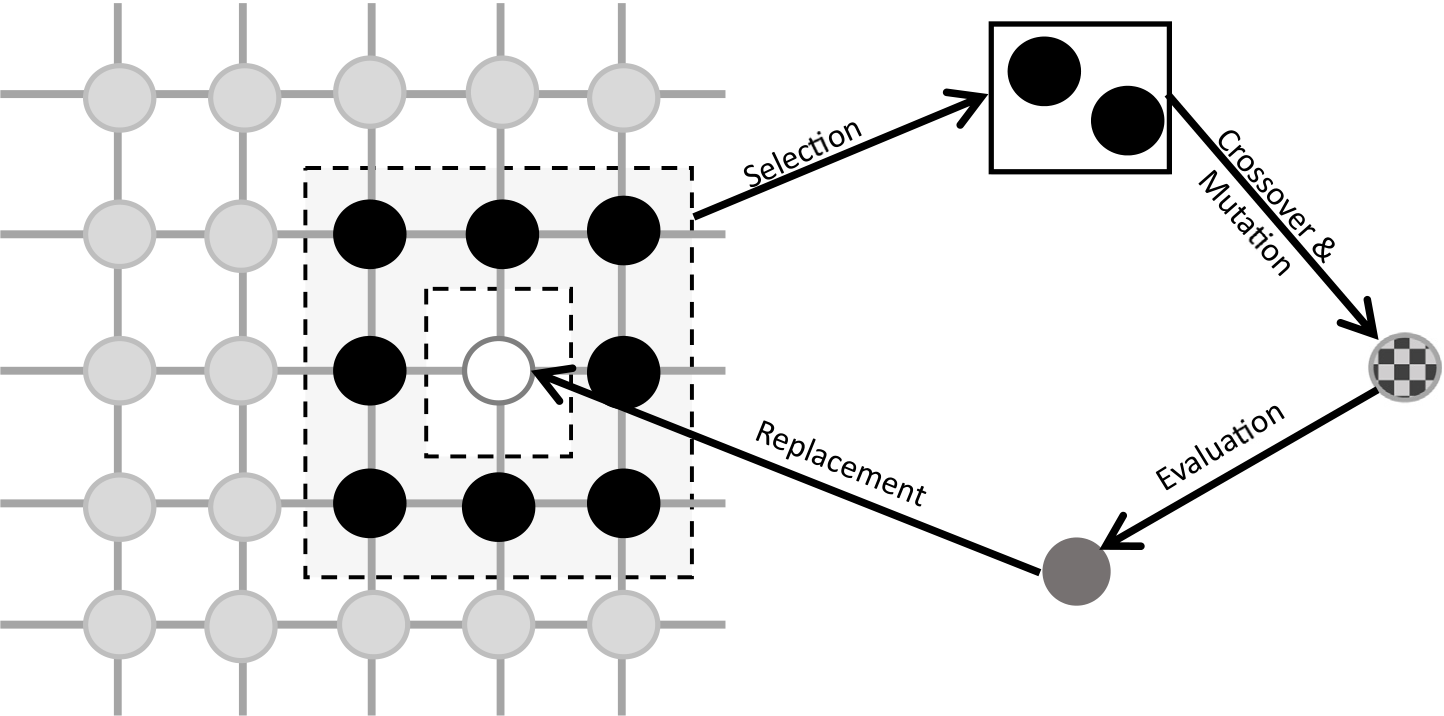
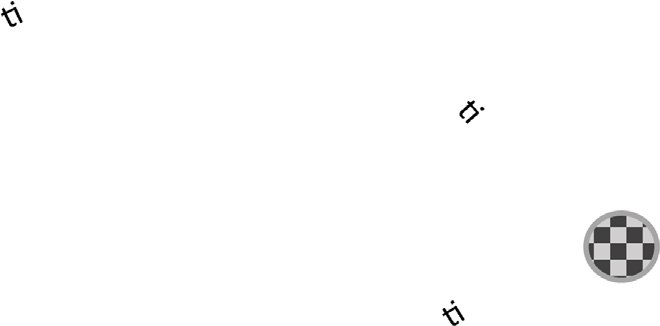
Salman et al. [47] compare GA, PSO and Fireworks Algorithms (FWA) at optimising weights and biases of an MLP for the classification of five benchmark medical datasets. Metaheuristics were tried over different MLP architectures, which contributed to understanding how metaheuristics performance could be affected by ANN architectural decisions and different parameters configurations. Results show that architectural changes did not significantly affect the performance of the algorithms. Nonetheless, increasing the iterations performed by each metaheuristic reported improvements in classification quality. Results inherently imply that metaheuristics perform well even when many weights or biases have to be optimised.

Bhattacharjee in [48] proposes five different hybridisations between GA and PSO to train an MLP that classifies human glioma from molec-ular brain neoplasia data. This paper provides an interesting point of view about how PSO and GA can be combined and which combination provides the best results. This work established that hybridisations between PSO and GA can report good results due to the synergetic effect generated.

In [49] authors compare eleven recently-proposed metaheuristics for training an ANN to classify fifteen different medical datasets. Al-gorithms included in the experiments were the Artificial Bee Colony (ABC), the Ant Lion Optimiser (ALO), the BBO, the Equilibrium Opti-miser (EO), the MFO, the Marine Predators Algorithm (MPA), the PSO, the Sine–Cosine Algorithm (SCA), the Salp Swarm Algorithm (SSA), the Trigonometric Mutation Differential Evolution (TDE), the WOA, a hybrid SSA with PSO, a hybrid SSA with SCA and the deterministic method for training ANN Levenberg–Marquardt. Evaluations focused on seven different classification quality metrics. Metaheuristics have proven to be highly effective in training ANNs. However, the evalu-ation has not considered the BP, overlooking one of the widely used options for training an ANN. The authors highlighted the EO among the metaheuristics, which obtained better values when considering all the classification metrics. Besides, the parameters of the EO were selected by observing which configuration provides the best results. The other algorithms were configured by adaptive approaches or setups used in literature, suggesting that comparisons could have been unfair.

Orozco et al. [50] applied multi-objective CGA to optimise an MLP. The multi-objective approach is focused on optimising the architecture

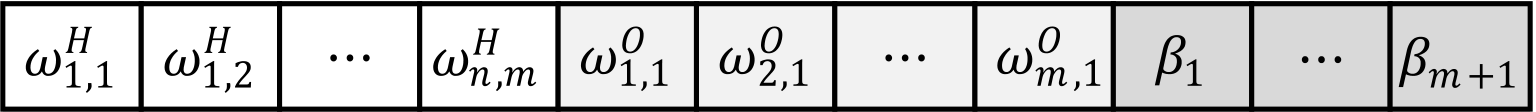
5



*M.G. Rojas et al.*  *Array 14 (2022) 100173*



**Fig. 2.** Type of neighbourhood C9 and application of genetics operators in an CGA.



**Fig. 3.** Example of vector representation of an MLP structure.

• **Finalisation stage:** Involves lines 16 and 17. When the evolution-ary stage has reached the stop condition, the best solution found is returned as a final solution.

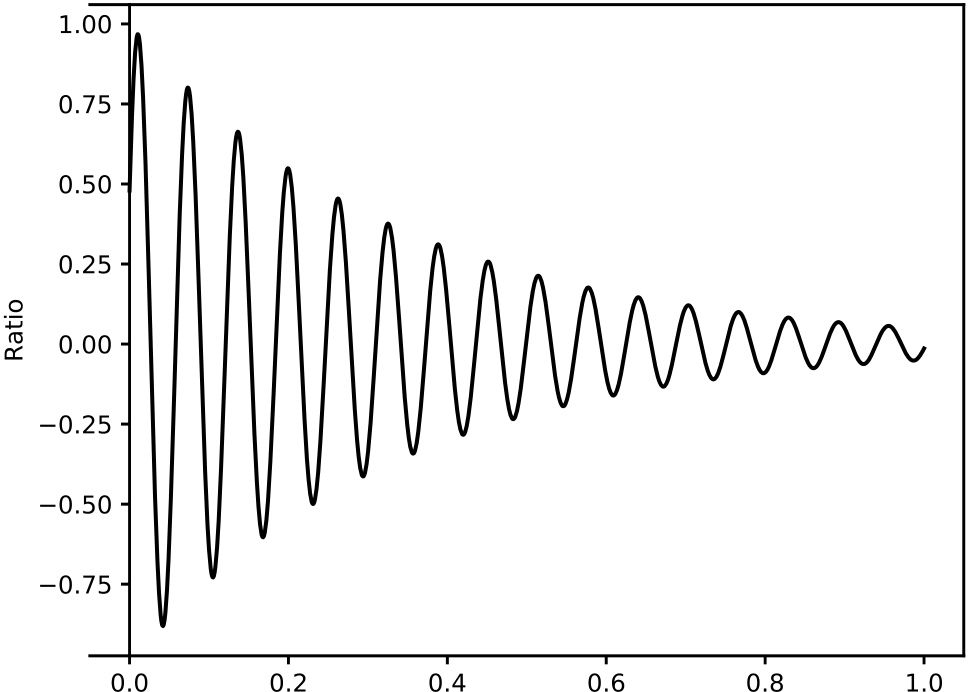
|  |
| --- |
| **Algorithm 1** Pseudo-code of the Cellular Genetic Algorithm (CGA)  1: **function** CellularGA(popSize,crossoverRate, mutationRate, maxE-  valuations)  2: *𝑝𝑜𝑝𝑢𝑙𝑎𝑡𝑖𝑜𝑛* ← *𝑖𝑛𝑖𝑡𝑖𝑎𝑙𝑖𝑠𝑒*(*𝑝𝑜𝑝𝑆𝑖𝑧𝑒*)  3: *𝑒𝑣𝑎𝑙𝑢𝑎𝑡𝑒*(*𝑝𝑜𝑝𝑢𝑙𝑎𝑡𝑖𝑜𝑛*)  4: *𝑡* ← *𝑝𝑜𝑝𝑆𝑖𝑧𝑒* *⊳* Evaluations counter  5: **while** *𝑡 < 𝑚𝑎𝑥𝐸𝑣𝑎𝑙𝑢𝑎𝑡𝑖𝑜𝑛𝑠* **do**  6: **for all** *𝑖𝑛𝑑𝑖𝑣𝑖𝑑𝑢𝑎𝑙* ∈ *𝑝𝑜𝑝𝑢𝑙𝑎𝑡𝑖𝑜𝑛* **do**  7: *𝑛𝑒𝑖𝑔ℎ𝑏𝑜𝑢𝑟𝑠* ← *𝑔𝑒𝑡𝑁𝑒𝑖𝑔ℎ𝑏𝑜𝑢𝑟𝑠*(*𝑖𝑛𝑑𝑖𝑣𝑖𝑑𝑢𝑎𝑙, 𝑝𝑜𝑝𝑢𝑙𝑎𝑡𝑖𝑜𝑛*)  8: *𝑝𝑎𝑟𝑒𝑛𝑡𝑠* ← *𝑠𝑒𝑙𝑒𝑐𝑡𝑖𝑜𝑛*(*𝑛𝑒𝑖𝑔ℎ𝑏𝑜𝑢𝑟𝑠*)  9: *𝑜𝑓𝑓𝑠𝑝𝑟𝑖𝑛𝑔* ← *𝑐𝑟𝑜𝑠𝑠𝑜𝑣𝑒𝑟*(*𝑝𝑎𝑟𝑒𝑛𝑡𝑠, 𝑐𝑟𝑜𝑠𝑠𝑜𝑣𝑒𝑟𝑅𝑎𝑡𝑒*)  10: *𝑜𝑓𝑓𝑠𝑝𝑟𝑖𝑛𝑔* ← *𝑚𝑢𝑡𝑎𝑡𝑖𝑜𝑛*(*𝑜𝑓𝑓𝑠𝑝𝑟𝑖𝑛𝑔, 𝑚𝑢𝑡𝑎𝑡𝑖𝑜𝑛𝑅𝑎𝑡𝑒*)  11: *𝑒𝑣𝑎𝑙𝑢𝑎𝑡𝑒*(*𝑜𝑓𝑓𝑠𝑝𝑟𝑖𝑛𝑔*)  12: *𝑟𝑒𝑝𝑙𝑎𝑐𝑒*(*𝑖𝑛𝑑𝑖𝑣𝑖𝑑𝑢𝑎𝑙, 𝑜𝑓𝑓𝑠𝑝𝑟𝑖𝑛𝑔*)  13: *𝑡* ← *𝑡* + 2 *⊳* Increased by the number of offsprings  14: **end for**  15: **end while**  16: *𝑏𝑒𝑠𝑡𝐼𝑛𝑑𝑖𝑣𝑖𝑑𝑢𝑎𝑙* ← *𝑔𝑒𝑡𝐵𝑒𝑠𝑡𝐼𝑛𝑑𝑖𝑣𝑖𝑑𝑢𝑎𝑙*(*𝑝𝑜𝑝𝑢𝑙𝑎𝑡𝑖𝑜𝑛*)  17: **return** *𝑏𝑒𝑠𝑡𝐼𝑛𝑑𝑖𝑣𝑖𝑑𝑢𝑎𝑙*  18: **end function** |

*3.1. Individual representation*

In literature, three kinds of individuals representation are used for MLP weights and biases optimisation: vector, matrix or binary [14]. The vector is used in this work. It represents each individual as a real array with a dimension equal to the total number of weights and biases in an MLP. Each element of the array corresponds to either a weight or a bias value. An example of this kind of representation, based on Fig. 1, is presented in Fig. 3.

6

*M.G. Rojas et al.*  *Array 14 (2022) 100173*





**Fig. 4.** Behaviour of damped harmonic oscillation function, presented in Eq. (11).

*𝑟𝑎𝑡𝑖𝑜* is determined by Eq. (11). It applies the damped harmonic oscillation function to determine how big the influence of parents and the best individual will be. This function provides a desirable behaviour because it applies more variable changes at the beginning of the evolutionary process but is more precise at the end, doing minimal changes to the influence of the best individual and the parents.

*𝑟𝑎𝑡𝑖𝑜* = *𝐴* × *𝑒*(−*𝐶𝑡*)× sin (*𝑃* + 99*.*75 × *𝑡*) (11)

where *𝑡* is the number of evaluations performed at the moment of the execution of the DX. *𝐴* is the initial amplitude of the function set to 1.0. *𝐶* is the variable that controls how fast *𝐴* decreases. The bigger is *𝐶*, the faster *𝐴* will decrease. *𝑃* is the phase of the function. The values of *𝐶* and *𝑃* used are 3 and 0.5, respectively. The Eq. (11) configured with the mentioned values of *𝐴*, *𝐶* and *𝑃* behaves like it is shown in Fig. 4, where a value of 0.0 in the *𝑥* axis represents the initial step of the evolutionary process, and a value of 1.0 is when the process is completed. The values of *𝐴*, *𝐶* and *𝑃* were selected to make the curve have a sustained decrease, thereby allowing precise variations at final evaluations but avoiding performing insignificant movements.

Both the *𝑟𝑎𝑡𝑖𝑜* and the *𝑑𝑖𝑓𝑓𝑖* functions aim to improve the exploita-tion of the DX by focusing on reaching the best individual and reducing the changes when the end of the evolutionary process is near. At the same time, *𝑑𝑖𝑓𝑓* and *𝑟𝑎𝑡𝑖𝑜* act as direction indicators as they make the new element increase or decrease the *𝑖*th element of the parents according to the difference with the *𝐵𝐼*.

**4. Experiments configuration**

This section presents the experimental setup. It provides informa-tion related to the evaluations to be performed, considered algorithms, standard configurations of the MLP and characteristics of the used datasets.

Comparisons were performed against different variations of the CGA, introduced in Section 4.1, and state-of-the-art algorithms. Consid-ered algorithms and their configurations are presented in Section 4.2. The MLP uses the sigmoid function as the activation function. The fitness function for the learning process performed by metaheuristics is the MSE, also known as cost function in the ANN scope [43,53]. The structure of the MLP is always the same. The number of input neurons (*𝑛*) matches the number of features of the dataset. The MLP is configured with a unique hidden layer where the number of hidden neurons (*𝑚*) is determined by Eq. (12), following the rule established in [14].

*𝑚* = 2 × *𝑛* + 1 (12)

7

*M.G. Rojas et al.*  *Array 14 (2022) 100173*

|  |
| --- |
| where *𝑝*1*𝑖* is the element *𝑖* of the first parent, *𝑝*2*𝑖* is the element *𝑖* of the second parent and *𝛽𝑞* is an ordinate obtained by Eq. (17).  *𝛽𝑞* = Being *𝑟* a random number between 0 and 1, and *𝜂* a distribution index fixed to 20 as considered in [60].⎧⎪⎨⎪⎩[ (2 × *𝑟*) 2×(1−*𝑟*)  1  ]  *𝜂*+1  1  *𝜂*+1  1  if *𝑟* ≤ 0*.*5  *𝑂𝑡ℎ𝑒𝑟𝑤𝑖𝑠𝑒*   (17) |

CGA variations are produced using four different mutation operators that vary in the strategy to diversify the population. The mutation operators are:

|  |  |  |  |
| --- | --- | --- | --- |
| • **Non-Uniform**  **Mutation**  **(NUM):**  It was first proposed by  Michalewicz et al. [61]. The objective of this operator is to avoid  generating new elements randomly. The mutated value *𝑠𝑜𝑙𝑡*+1 *𝑖*  of  the *𝑖*th element of a solution in the evaluation *𝑡* is generated  by Eq. (18) according to a mutation probability.  *𝑠𝑜𝑙𝑡*+1 *𝑖*  = { *𝑠𝑜𝑙𝑡*  *𝑠𝑜𝑙𝑡*   *𝑖*+ ▵ (*𝑡, 𝑢𝑏* − *𝑠𝑜𝑙𝑡*  *𝑖*+ ▵ (*𝑡, 𝑠𝑜𝑙𝑡 𝑖*− *𝑙𝑏*)   *𝑖*) if *𝑟* ≤ 0*.*5  *𝑂𝑡ℎ𝑒𝑟𝑤𝑖𝑠𝑒*   (18)  With *𝑙𝑏* and *𝑢𝑏* as the lower and the upper bounds of the element  *𝑠𝑜𝑙𝑡 𝑖*, a random number *𝑟* ∈ [0*,* 1] and the function ▵ (*𝑡, 𝑑*)  calculated by Eq. (19).  ▵ (*𝑡, 𝑑*) = *𝑑* ×  *𝑇* is the maximum number of evaluations, and *𝑏* establishes the ( 1 − *𝑟*(1− *𝑡 𝑇*)*𝑏*) (19)  dependency on the evaluations number, set to 0.5. As the value  of *𝑏* is bigger, more disturbance is applied by the operator as  evaluations are performed. Thus, small values of *𝑏* mean that  more precise movements will be made.  • **Uniform Mutation (UM):** In this case, the element *𝑠𝑜𝑙𝑡 𝑖*is substi-  tuted by a mutated value *𝑠𝑜𝑙𝑡*+1 *𝑖*  obtained by Eq. (20). | | | |
| *𝑠𝑜𝑙𝑡*+1 *𝑖* | = *𝑠𝑜𝑙𝑡 𝑖*+ (*𝑟* − 0*.*5) × *𝑢* | | (20) |
| With a random number *𝑟* ∈ [0*,* 1] and the disturbance level of the operator *𝑢* set to 0.5. If the value generated is out of the range delimited by the lower and the upper bounds, the new value will be one of the bounds.  • **Polynomial Mutation (PM):** This mutation operator was pro- posed by Deb and Agrawal [62]. A polynomial probability dis- tribution is used in this proposal to mutate each element of the *𝑖*), taking into account the lower (*𝑙𝑏*) and upper (*𝑢𝑏*) individual (*𝑠𝑜𝑙𝑡*   bounds. Eq. (21) is used to obtain the new value *𝑠𝑜𝑙𝑡*+1 *𝑖*  for the *𝑖*th element of the solution to mutate.  *𝑠𝑜𝑙𝑡*+1 *𝑖*  = {*𝑠𝑜𝑙𝑡 𝑖*+ *𝛾𝑅* × (*𝑢𝑏* − *𝑠𝑜𝑙𝑡*  *𝑖*+ *𝛾𝐿* × (*𝑠𝑜𝑙𝑡 𝑖*− *𝑙𝑏*) *𝑖*) if *𝑟* ≤ 0*.*5  *𝑂𝑡ℎ𝑒𝑟𝑤𝑖𝑠𝑒*  (21)  where *𝑟* is a random number belonging to the range [0*,* 1] and the functions *𝛾𝐿*(*𝑑*) and *𝛾𝑅*(*𝑑*) are calculated by Eqs. (22) and (23) respectively. | | | |
| *𝛾𝐿*(*𝑟*) = (2 × *𝑑*) | | 1  1+*𝜂* | (22) |
| *𝛾𝑅*(*𝑟*) = 1 −(2 × (1 − *𝑑*)) With *𝜂* fixed to 5, which is a user-defined parameter that controls 1+*𝜂*  (23) 1  the perturbation applied to elements of the individuals.  • **Random Mutation (RM):** This operator is one of the simplest ways to mutate real-coded individuals. It mutates the value of an element *𝑠𝑜𝑙𝑡 𝑖*by generating a completely new value between the lower bound (*𝑙𝑏*) and the upper bound (*𝑢𝑏*) of the element, considering a random number *𝑟* in the range [0*,* 1]. For this, it uses Eq. (24). | | | |
| *𝑠𝑜𝑙𝑡*+1 *𝑖* | = *𝑙𝑏* + (*𝑢𝑏* − *𝑙𝑏*) × *𝑟* | | (24) |

8

*M.G. Rojas et al.*  *Array 14 (2022) 100173*

|  |  |  |
| --- | --- | --- |
| **Table 1**  Algorithms parameters configurations. | |  |
| Algorithm | Parameter | Value |
| CGA (This paper) | Population size  Neighbourhood  Crossover probability Mutation probability Selection operator | 10 × 10  C9  0.9  0.01  Binary tournament |
| BAT | Population size  Loudness  Pulse rate  Frequency minimum Frequency maximum | 50  0.5  0.5  0  1 |
| CS | Number of nests  Discovery rate | 50  0.25 |
| DE | Population size  Crossover probability Differential weight | 50  0.9  0.5 |
| GA | Population size  Crossover operator Mutation operator Selection operator | 50  SBX (Prob.: 0.9)  UM (Prob.: 0.01) Binary tournament |
| GWO | Population size *̂𝑎* | 50  Decrease linearly from 2 to 0 |
| MFO | Population size  *𝑏* | 50  1  [−1*,* 1] |
| *𝑡* | |
| MVO | Population size  Min. wormhole existence prob. Max. wormhole existence prob. | 50  0.2  1.0 |
| PSO | Number of particles Inertia weight  Cognitive component | 50  0.721  1.193 |

Table 1 presents a summary of the configuration for each considered algorithm. All the CGA versions use the same parameters configuration. Parameters utilised by state-of-the-art algorithms are established based on previous works oriented to adjust weights and biases of the MLP with metaheuristics [30,55,70,71].

*4.3. Datasets*

Evaluations of the MLP optimised by metaheuristics were performed using five different medical datasets obtained from the UCI machine learning repository.3Each dataset is described in the following para-graphs and a summary is provided in Table 2.

• **Breast:** This dataset is composed of 699 instances, where each one corresponds to a patient submitted to surgery. Eight vari-ables were measured, and the instances are distinguished between benign or malignant cases [72,73].

• **Diabetes:** It is composed of 768 instances which are classified as positive or negative diabetes cases. Data were collected from a population of Pima-Indian women at least 21 years old living near Phoenix, Arizona, USA [74].

• **Liver:** Instances come from blood tests performed over 345 male patients with apparent liver disorders by excessive alco-hol consumption. Instances are split into positives and negative classes [75].

• **Parkinson:** This dataset was obtained from voice analysis per-formed over thirty-one patients. Specialists took almost 6 record-ings for each individual. Recordings have 22 different metrics, and each one is classified as Parkinson’s disease or normal [76].

3 <https://archive.ics.uci.edu/>.

9

*M.G. Rojas et al.*  *Array 14 (2022) 100173*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table 3**  Mean and standard deviation of the fitness quality indicator. | | | | | |  |
| Algorithm | Breast | | Diabetes | Liver | Parkinsons | Vertebral |
| BAT  CS  DE  GA  GWO  MFO  MVO  PSO  AX+NUM AX+PM  AX+RM  AX+UM  SBX+NUM SBX+PM  SBX+RM  SBX+UM  DX+NUM DX+PM  DX+RM  DX+UM | | 0*.*0360*.*002  0*.*0530*.*004  0*.*0480*.*002  0*.*0360*.*002  0*.*0350*.*002  0*.*0350*.*002  0*.*0300*.*002  0*.*0300*.*001  0*.*0360*.*002  0*.*0360*.*002  0*.*0360*.*002  0*.*0380*.*002  0*.*0370*.*002  0*.*0370*.*001  0*.*0370*.*001  0*.*0370*.*001  0*.*0290*.*002  *0.0280.001* 0*.*0300*.*002  **𝟎***.***𝟎𝟐𝟖𝟎***.***𝟎𝟎𝟏** | 0*.*1520*.*002  0*.*1710*.*005  0*.*1640*.*004  0*.*1530*.*001  0*.*1530*.*001  0*.*1530*.*002  0*.*1470*.*002  0*.*1480*.*001  0*.*1550*.*002  0*.*1550*.*002  0*.*1560*.*002  0*.*1570*.*003  0*.*1560*.*001  0*.*1560*.*001  0*.*1560*.*001  0*.*1550*.*001  0*.*1460*.*002  *0.146*0.001 0*.*1470*.*001  **𝟎***.***𝟏𝟒𝟓𝟎***.***𝟎𝟎𝟏** | 0*.*1960*.*006  0*.*2110*.*004  0*.*2080*.*004  0*.*1990*.*002  0*.*2000*.*003  0*.*1950*.*004  **𝟎***.***𝟏𝟕𝟔𝟎***.***𝟎𝟎𝟑**  0*.*1860*.*002  0*.*2020*.*002  0*.*2020*.*002  0*.*2040*.*002  0*.*2070*.*004  0*.*2050*.*003  0*.*2050*.*002  0*.*2060*.*002  0*.*2060*.*002  0*.*1820*.*002  0*.*1820*.*002  0*.*1840*.*002  *0.1810.002* | 0*.*0940*.*007  0*.*2200*.*027  0*.*1820*.*018  0*.*1040*.*006  0*.*0920*.*004  0*.*1070*.*010  0*.*0770*.*007  *0.0760.005* 0*.*1070*.*007  0*.*1080*.*007  0*.*1090*.*007  0*.*1150*.*013  0*.*1260*.*008  0*.*1270*.*009  0*.*1260*.*009  0*.*1170*.*008  0*.*0870*.*006  0*.*0870*.*006  0*.*0920*.*005  **𝟎***.***𝟎𝟔𝟔𝟎***.***𝟎𝟎𝟒** | 0*.*1320*.*004 0*.*1480*.*006 0*.*1440*.*003 0*.*1330*.*002 0*.*1340*.*002 0*.*1310*.*002 **𝟎***.***𝟏𝟏𝟖𝟎***.***𝟎𝟎𝟑** 0*.*1250*.*002 0*.*1370*.*003 0*.*1370*.*002 0*.*1380*.*003 0*.*1410*.*004 0*.*1390*.*002 0*.*1390*.*002 0*.*1400*.*002 0*.*1390*.*002 0*.*1220*.*002 0*.*1220*.*002 0*.*1230*.*002 *0.1210.001* |

*5.1. Fitness and time analysis*

This section begins with an analysis of the numerical performance of metaheuristics. Then, a comparison of the time consumed in seconds to reach the stop criteria is presented. Tests are made by applying the algorithms to the five considered benchmarks datasets during the 30 independent runs.

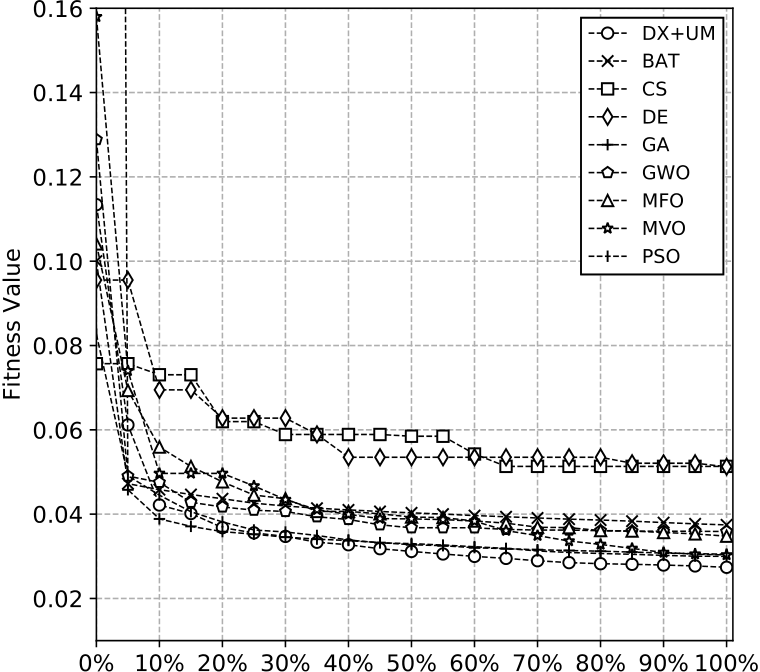
Table 3 reports the mean fitness and the standard deviation reached by the evaluated algorithms. The first column contains the considered algorithms, and the results obtained are presented in columns two to six. Results show that the CGA configured with DX crossover and the UM mutation (DX+UM) overcomes the other metaheuristics in three out of the five datasets (Breast, Diabetes and Parkinsons). A possible explanation for these results could be that the UM does not apply a significant disturbance over the genes, contributing to finding better near solutions and not diverting the seek. Furthermore, UM is not influenced by the number of evaluations done (already considered by the DX), which helps when the algorithm is stuck. DX+UM also reached the second-best solution for liver and vertebral datasets, falling behind the MVO, which got the best average of fitness values in both instances. These results appear to be related to how the neural network learns, which can be affected when a small number of attributes is used for training, as in liver and vertebral datasets. Results could be even worse so if those attributes are noisy. Beyond that, the DX+UM kept very near the best results obtained by the MVO, meaning that it achieves minimal values of MSE.

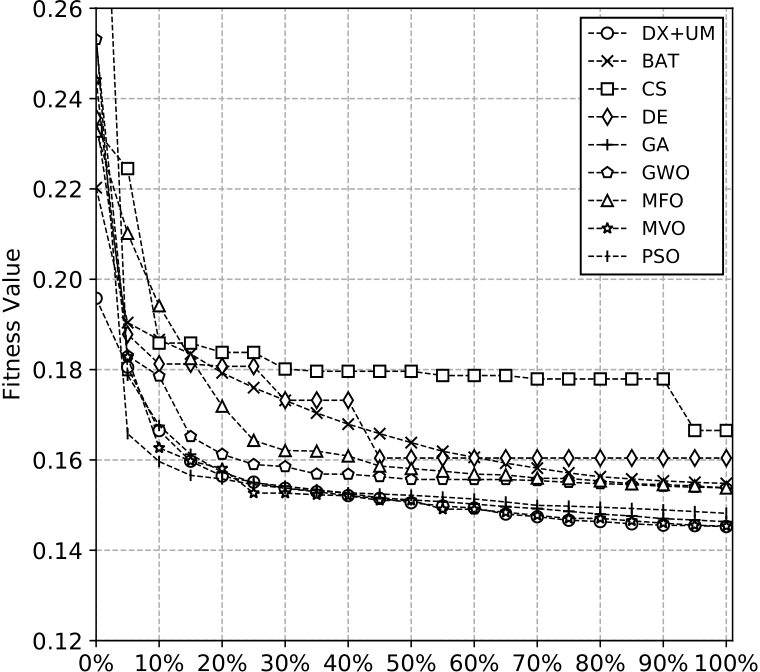
In general, DX variations have shown similar behaviour, suggesting that the DX characteristics of taking into account the best current solution and making more minor changes as the population evolves can increase in a significant way the exploration and exploitation capability of the CGA.

Table 4 shows the results of the statistical analysis obtained by applying the Wilcoxon rank-sum test over the fitness value obtained by each algorithm. The table compares the DX variations of the CGA (located in columns) against the other metaheuristics (placed in rows). Each table cell contains a set of five symbols, representing the result of the comparison using each of the datasets, namely Breast Cancer, Diabetes, Liver, Parkinson and Vertebral. A leftward triangle (*⊲*) means that the row algorithm gets statistically better values than the column algorithm. An upward triangle (▵) means that the column algorithm gets better values than the row metaheuristic. If no significant differ-ences are found, the place is completed with a dash (–). For example, the first upward triangle in the table means that with the dataset Breast,

10

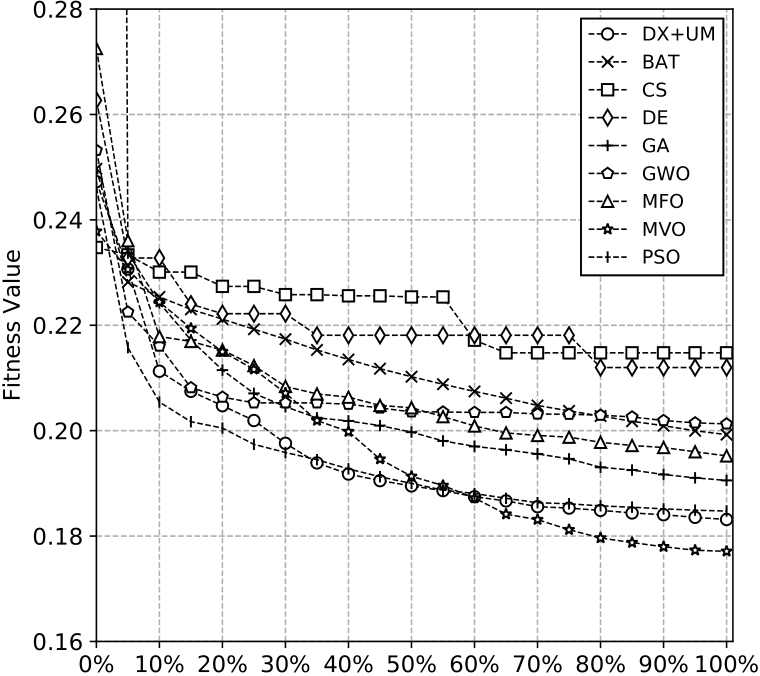
*M.G. Rojas et al.*  *Array 14 (2022) 100173*

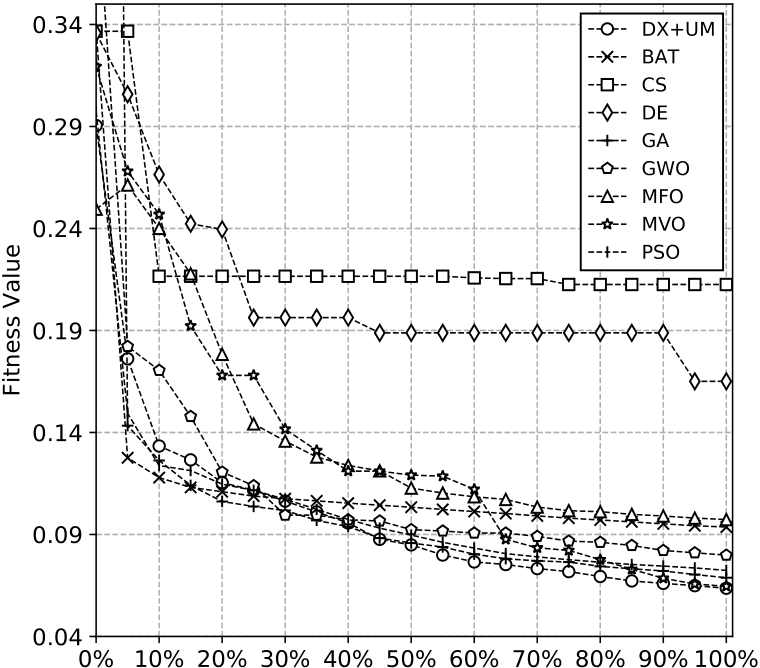




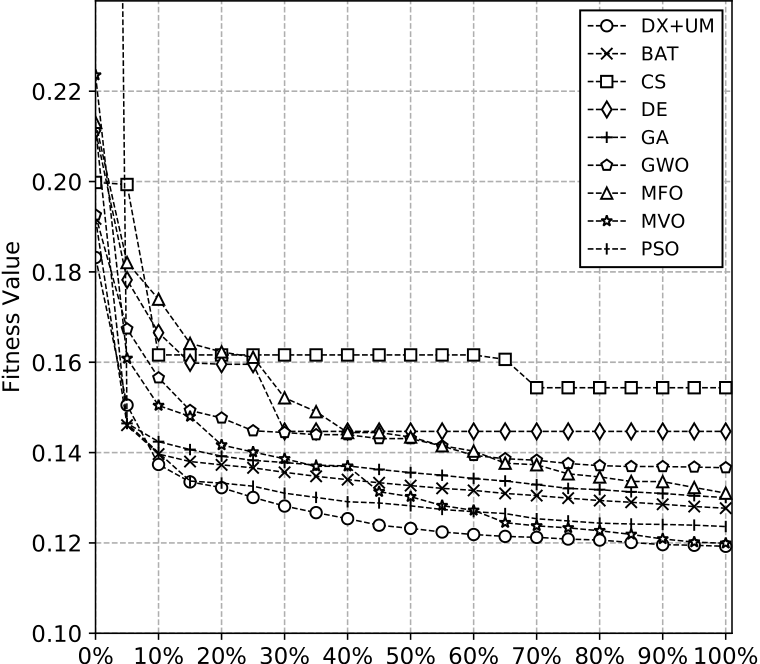
 











**Fig. 5.** Convergence curves of fitness value (MSE) of the five datasets.

*5.2. Classification metrics analysis*

This section provides a point of view about the performance of the MLP configured by the metaheuristics. First of all, the accuracy reached by each MLP is informed. Next, Sensitivity (Sn) and Specificity (Sp) are analysed to observe if the classification model has been balanced

11

*M.G. Rojas et al.*  *Array 14 (2022) 100173*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 5**  Mean and standard deviation of the time quality indicator. | | | | | | | |
| Algorithm | Breast | | | Diabetes | Liver | Parkinsons | Vertebral |
| BAT  CS  DE  GA  GWO  MFO  MVO  PSO  AX+NUM AX+PM  AX+RM  AX+UM  SBX+NUM SBX+PM  SBX+RM  SBX+UM  DX+NUM DX+PM  DX+RM  DX+UM | | 670*.*0853*.*469  671*.*2013*.*404  679*.*6063*.*443  675*.*9244*.*141  858*.*6347*.*135  694*.*2845*.*010  688*.*6382*.*860  684*.*3853*.*963  664*.*88824*.*317  665*.*01823*.*633  669*.*45522*.*040  667*.*60621*.*882  666*.*91023*.*772  672*.*32721*.*499  668*.*75820*.*317  664*.*90923*.*683  **𝟒𝟏𝟎***.***𝟑𝟑𝟔𝟑𝟒***.***𝟎𝟒𝟓**  *425.737 20.098* 427*.*16819*.*789  426*.*24624*.*491 | | 731*.*0708*.*384  733*.*28010*.*611  741*.*7857*.*983  732*.*09610*.*269  922*.*1998*.*422  755*.*43611*.*762  751*.*5527*.*176  739*.*88011*.*074  728*.*18129*.*326  731*.*38026*.*913  728*.*37328*.*904  726*.*21525*.*157  729*.*47824*.*990  728*.*70629*.*179  726*.*25926*.*142  725*.*25326*.*871  **𝟒𝟑𝟓***.***𝟒𝟕𝟖𝟒𝟗***.***𝟑𝟕𝟐**  *454.12825.014* 456*.*10227*.*252  461*.*05335*.*099 | 326*.*2351*.*927  324*.*9891*.*397  332*.*5021*.*428  329*.*2436*.*266  442*.*4382*.*194  339*.*7925*.*065  337*.*8465*.*612  332*.*4986*.*024  330*.*5785*.*546  324*.*70010*.*418 327*.*5958*.*761  329*.*5816*.*725  330*.*4256*.*960  330*.*3326*.*679  330*.*4066*.*855  330*.*3016*.*435  **𝟐𝟏𝟕***.***𝟓𝟒𝟒𝟖***.***𝟖𝟕𝟗**  *219.9896.717* 220*.*7005*.*058  221*.*6834*.*913 | *273.507 5.402* **𝟐𝟓𝟑***.***𝟏𝟔𝟖𝟑***.***𝟔𝟑𝟑**  339*.*1174*.*780  333*.*2895*.*370  1503*.*95962*.*336 389*.*8377*.*562  426*.*7264*.*205  392*.*3564*.*975  311*.*9291*.*758  312*.*5431*.*231  310*.*7511*.*378  311*.*3411*.*342  319*.*0252*.*191  318*.*3611*.*847  317*.*4621*.*934  318*.*3812*.*309  621*.*92888*.*362 674*.*76367*.*467 673*.*41964*.*273 676*.*34064*.*095 | 296*.*5532*.*318  289*.*59410*.*815 297*.*4329*.*723  299*.*9224*.*780  409*.*9006*.*772  311*.*2313*.*031  306*.*3357*.*593  302*.*3035*.*761  297*.*8356*.*342  297*.*1187*.*528  297*.*4606*.*379  295*.*2098*.*767  299*.*6255*.*178  299*.*1045*.*860  299*.*7605*.*351  298*.*6486*.*704  **𝟏𝟗𝟗***.***𝟑𝟖𝟗𝟏𝟓***.***𝟐𝟏𝟕** *203.6918.301* 204*.*1958*.*599  204*.*6558*.*437 |
| **Table 6**  Mean and standard deviation of accuracy metric reached by each algorithm. | | | | | | | |
| Algorithm | Breast | | | Diabetes | Liver | Parkinsons | Vertebral |
| BAT  CS  DE  GA  GWO  MFO  MVO  PSO  BP  AX+NUM AX+PM  AX+RM  AX+UM  SBX+NUM SBX+PM  SBX+RM  SBX+UM  DX+NUM DX+PM  DX+RM  DX+UM | | | 0*.*9760*.*005  0*.*9570*.*011  0*.*9620*.*006  0*.*9730*.*005  0*.*9780*.*004  0*.*9760*.*006  0*.*9730*.*007  0*.*9780*.*005  0*.*9530*.*011  0*.*9740*.*005  0*.*9750*.*007  0*.*9750*.*005  0*.*9710*.*005  0*.*9710*.*006  0*.*9730*.*005  0*.*9750*.*007  0*.*9720*.*006  *0.9810.004 0.9810.005* 0*.*9800*.*006  **𝟎***.***𝟗𝟖𝟐𝟎***.***𝟎𝟎𝟓** | 0*.*7540*.*007  0*.*7240*.*030  0*.*7380*.*021  0*.*7510*.*007  0*.*7510*.*005  0*.*7520*.*009  0*.*7570*.*009  0*.*7530*.*008  0*.*7160*.*026  0*.*7470*.*010  0*.*7500*.*010  0*.*7470*.*012  0*.*7440*.*008  0*.*7470*.*011  0*.*7480*.*013  0*.*7510*.*011  0*.*7490*.*010  0*.*7540*.*007  *0.757 0.011* 0*.*7540*.*011  **𝟎***.***𝟕𝟓𝟖𝟎***.***𝟎𝟏𝟎** | 0*.*7510*.*018  0*.*6910*.*030  0*.*7270*.*028  *0.7580.016* **𝟎***.***𝟕𝟔𝟎𝟎***.***𝟎𝟏𝟗**  0*.*7570*.*014  0*.*7280*.*017  0*.*7530*.*018  0*.*6470*.*043  0*.*7440*.*019  0*.*7530*.*021  0*.*7400*.*020  0*.*7310*.*021  0*.*7430*.*019  0*.*7390*.*019  0*.*7400*.*020  0*.*7430*.*016  0*.*7480*.*019  0*.*7510*.*015  0*.*7520*.*014  0*.*7490*.*018 | 0*.*8610*.*017  0*.*6990*.*060  0*.*7540*.*060  0*.*8480*.*036  **𝟎***.***𝟖𝟕𝟔𝟎***.***𝟎𝟐𝟏**  0*.*8420*.*037  0*.*8540*.*032  0*.*8670*.*026  0*.*8610*.*059  0*.*8320*.*036  0*.*8320*.*027  0*.*8460*.*032  0*.*8240*.*043  0*.*8160*.*038  0*.*8050*.*049  0*.*8020*.*034  0*.*8400*.*026  *0.8700.021* 0*.*8630*.*025  0*.*8600*.*021  *0.8700.025* | *0.8750.013* 0*.*8070*.*033  0*.*8380*.*031  0*.*8730*.*014  **𝟎***.***𝟖𝟕𝟔𝟎***.***𝟎𝟏𝟎**  0*.*8710*.*014  0*.*8690*.*014  0*.*8730*.*011  0*.*8180*.*036  0*.*8640*.*017  0*.*8720*.*020  0*.*8680*.*019  0*.*8470*.*031  0*.*8570*.*026  0*.*8690*.*023  0*.*8610*.*020  0*.*8680*.*020  0*.*8730*.*009  0*.*8730*.*013  *0.8750.015 0.8750.016* |

a diagnosis) or a high specificity value (e.g. when mislabelling a sample as positive is detrimental) [78].

Table 6 shows the average of the accuracy values and the standard deviation obtained by the MLP configured with weights and biases generated by metaheuristics. Results were calculated using the test subset of every dataset.

Metaheuristics have overcome the mean accuracy of the Back Prop-agation algorithm (BP) in all the considered datasets. Focusing on the DX variations, the DX+UM has stood out in two out of the five datasets (Breast and Diabetes) and has had the second-best mean accuracy in two other datasets (Parkinson and Vertebral). The GWO has emerged as the best solution with Liver, Parkinsons and Vertebral datasets. Because the algorithms that achieved the best fitness value did not reach the best accuracy with these datasets, it is evident that there is no relationship between fitness value and accuracy. As can be seen, all the metaheuristics have obtained very similar results. In particular, CGA with DX crossover has proven to achieve competitive accuracy results with all the datasets.

Table 7 displays the mean of the Sn and the Sp for the DX variations and the algorithms of the state-of-the-art. The best results are marked with bold font, and the second-best is marked with italic font.

12

*M.G. Rojas et al.*  *Array 14 (2022) 100173*

process. DX operator has demonstrated to improve the exploration and exploitation of the search space of CGA even more.

Experiments use five well-known benchmark medical datasets. Com-parisons are against state-of-the-art algorithms and CGA versions con-figured with well-known genetic operators. Two aspects were eval-uated, the convergence capability of the algorithms and the quality of classification achieved by the MLP optimised by the evaluated metaheuristics.

In general, DX variations have demonstrated that they can rapidly converge to optimal points of the search space. Regarding MSE values, the CGA+DX was the best in three out of the five considered datasets and the second-best in the remaining two. The DX operator combined with the UM mutation achieved better results than the other algorithms and reached minimal fitness values. Considering the times consumed by each algorithm, the DX variations were the best in four out of five datasets, overwhelming even to the CGA with other crossover operators. It is reliable proof that the DX operator performs its task efficiently, making the CGA work quicker.

The DX+UM produced better accuracy results in two datasets and was the second-best in the other two datasets. The other DX variations showed to be very near to the results of the DX+UM and the state-of-the-art algorithms. These results suggest that the optimisation process depends on the fitness function definition, because minimal values of MSE do not necessarily imply better accuracy values. Despite that, these results confirm that DX variations can have a featured performance in optimising weights and biases of the MLP, being able to improve the classification.

Besides, metrics of classification quality showed that solutions CGA variations with DX crossover get competitive results of specificity and sensitivity, deriving in a level of learning and generalisation com-parable to other approaches. Specifically, DX variations highlight in the Breast dataset by reaching the second-best result. With the other datasets, the performance was very similar to the state-of-the-art algo-rithms.

To conclude, results demonstrate that the CGA can be a robust and reliable tool for identifying the optimal weights and biases of the MLP for classifying medical data.

For future work, extending the CGA application to optimise the parameters and structure of neural networks is proposed. Furthermore, it is desirable to study alternatives to MSE function as fitness functions to obtain a better relationship between the fitness function of the metaheuristic and the accuracy reached by the MLP.

**CRediT authorship contribution statement**

**Matías Gabriel Rojas:** Conceptualization, Methodology, Software, Investigation, Formal analysis, Writing – original draft, Writing – re-view & editing. **Ana Carolina Olivera:** Conceptualization, Method-ology, Validation, Investigation, Formal analysis, Writing – original draft, Writing – review & editing. **Pablo Javier Vidal:** Conceptualiza-tion, Methodology, Validation, Investigation, Formal analysis, Writing- original draft, Writing – review & editing.

**Declaration of competing interest**

The authors declare that they have no known competing finan-cial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Acknowledgements**

This work was supported in part by Universidad Nacional de Cuyo, Argentina (Project No. 06/B081-B). Authors thank Consejo Nacional de Investigaciones Científicas *𝑦* Técnicas, Argentina.

13

*M.G. Rojas et al.*  *Array 14 (2022) 100173*

[24] Alba E, Dorronsoro B. Continuous optimization. In: Cellular genetic algorithms.

Boston, MA: Springer US; 2008, p. 167–74. <http://dx.doi.org/10.1007/978-0-387-77610-1_12>, Ch. 12.

[25] [Dorronsoro B, A](http://dx.doi.org/10.1007/978-0-387-77610-1_12)lba E. A simple cellular genetic algorithm for continuous opti-mization. In: 2006 IEEE international conference on evolutionary computation. 2006, p. 2838–44. <http://dx.doi.org/10.1109/CEC.2006.1688665>.

[26] Tinós R. Artificial [neural network based crossover for evolution](http://dx.doi.org/10.1109/CEC.2006.1688665)ary algorithms.

Appl Soft Comput 2020;95:106512. [http://dx.doi.org/10.1016/J.ASOC.2020. 106512](http://dx.doi.org/10.1016/J.ASOC.2020.106512).

[27] Mirjalili S. Evolutionary multi-layer perceptron. In: Evolutionary algorithms and neural networks: Theory and applications. Cham: Springer International Publishing; 2019, p. 87–104. <http://dx.doi.org/10.1007/978-3-319-93025-1_7>, Ch. 7.

[28] Principe JC, Lefebvre C, Fancourt CL. Dataflow learning in coupled lattices: An application to artificial neural networks. In: Handbook of global optimization: Volume 2. Boston, MA: Springer US; 2002, p. 363–86. [http://dx.doi.org/10. 1007/978-1-4757-5362-2\_10](http://dx.doi.org/10.1007/978-1-4757-5362-2_10), Ch. 10.

[29] [Krogh A. What are artificial n](http://dx.doi.org/10.1007/978-1-4757-5362-2_10)eural networks? Nature Biotechnol 2008;26(2):195– 7. <http://dx.doi.org/10.1038/nbt1386>.

[30] He[idari](http://dx.doi.org/10.1038/nbt1386)  [AA,](http://dx.doi.org/10.1038/nbt1386)  [Faris](http://dx.doi.org/10.1038/nbt1386)  [H,](http://dx.doi.org/10.1038/nbt1386)  [Aljarah](http://dx.doi.org/10.1038/nbt1386)  [I,](http://dx.doi.org/10.1038/nbt1386)  Mirjalili S. An efficient hybrid multi- layer perceptron neural network with grasshopper optimization. Soft Comput 2018;23(17):7941–58. <http://dx.doi.org/10.1007/s00500-018-3424-2>.

[31] Werbos PJ. Generalizat[ion of backpropagation with application to a re](http://dx.doi.org/10.1007/s00500-018-3424-2)current gas market model. Neural Netw 1988;1(4):339–56. <http://dx.doi.org/10.1016/0893-6080(88)90007-X>.

[32] [Hestenes MR, Sti](http://dx.doi.org/10.1016/0893-6080(88)90007-X)[efel E, et al. Methods of conjugate gradients for solving linear](http://refhub.elsevier.com/S2590-0056(22)00033-9/sb32)  [systems. J Res Natl Bur Stand 1952;49(6):409–36.](http://refhub.elsevier.com/S2590-0056(22)00033-9/sb32)

[33] [Chen O-C, Sheu BJ. Optimization schemes for neu](http://refhub.elsevier.com/S2590-0056(22)00033-9/sb32)ral network training. In: Pro-ceedings of 1994 IEEE international conference on neural networks (ICNN’94), Vol. 2. 1994, p. 817–22. <http://dx.doi.org/10.1109/ICNN.1994.374284>.

[34] Bertsekas DP. Nonlinear [programming. J Oper Res Soc 1997;48(3):3](http://dx.doi.org/10.1109/ICNN.1994.374284)34. [http:](http://dx.doi.org/10.1057/palgrave.jors.2600425)  [//dx.doi.org/10.1057/palgrave.jors.2600425](http://dx.doi.org/10.1057/palgrave.jors.2600425).

[35] [Marquardt DW. An algorithm for least-squa](http://dx.doi.org/10.1057/palgrave.jors.2600425)res estimation of nonlinear parame-ters. J Soc Ind Appl Math 1963;11(2):431–41, URL [http://www.jstor.org/stable/ 2098941](http://www.jstor.org/stable/2098941).

[36] Devikanniga D, Vetrivel K, Badrinath N. Review of meta-heuristic optimiza-tion based artificial neural networks and its applications. J Phys Conf Ser 2019;1362:012074. <http://dx.doi.org/10.1088/1742-6596/1362/1/012074>.

[37] Hemeida AM, Hassa[n SA, Mohamed A-AA, Alkhalaf S, Mahmoud MM, Senj](http://dx.doi.org/10.1088/1742-6596/1362/1/012074)yu T, El-Din AB. Nature-inspired algorithms for feed-forward neural network classifiers: A survey of one decade of research. Ain Shams Eng J 2020;11(3):659–75. <http://dx.doi.org/10.1016/j.asej.2020.01.007>.

[38] [Kaveh M, Khishe M, Mosavi MR. Design and](http://dx.doi.org/10.1016/j.asej.2020.01.007) implementation of a neighborhood search biogeography-based optimization trainer for classifying sonar dataset using multi-layer perceptron neural network. Analog Integr Circuits Signal Process 2018;100(2):405–28. <http://dx.doi.org/10.1007/s10470-018-1366-3>.

[39] Qiao W, Khishe M, Ravakha[h S. Underwater targets classification using l](http://dx.doi.org/10.1007/s10470-018-1366-3)ocal wavelet acoustic pattern and Multi-Layer Perceptron neural network optimized by modified Whale Optimization Algorithm. Ocean Eng 2021;219:108415. [http: //dx.doi.org/10.1016/j.oceaneng.2020.108415](http://dx.doi.org/10.1016/j.oceaneng.2020.108415).

[40] [Jalali SMJ, Hedjam R, Khosravi A, Heidar](http://dx.doi.org/10.1016/j.oceaneng.2020.108415)i AA, Mirjalili S, Nahavandi S.

Autonomous robot navigation using moth-flame-based neuroevolution. In: Evo-lutionary machine learning techniques: Algorithms and applications. Singapore: Springer Singapore; 2020, p. 67–83. <http://dx.doi.org/10.1007/978-981-32-9990-0_5>, Ch. 5.

[41] [Mansouri](http://dx.doi.org/10.1007/978-981-32-9990-0_5) A, Majidi B, Shamisa A. Metaheuristic neural networks for anomaly recognition in industrial sensor networks with packet latency and jitter for smart infrastructures. Int J Comput Appl 2018;43(3):257–66. [http://dx.doi.org/ 10.1080/1206212x.2018.1533613](http://dx.doi.org/10.1080/1206212x.2018.1533613).

[42] [Aladejare AE, Onifade M, Lawal A](http://dx.doi.org/10.1080/1206212x.2018.1533613)I. Application of metaheuristic based artificial neural network and multilinear regression for the prediction of higher heating values of fuels. Int J Coal Prep Util 2020;1–22. [http://dx.doi.org/10.1080/ 19392699.2020.1768080](http://dx.doi.org/10.1080/19392699.2020.1768080).

[43] Jalali SMJ, Ahmadian S, Kebria PM, Khosravi A, Lim CP, Nahavandi S. Evolv-ing artificial neural networks using butterfly optimization algorithm for data classification. In: Gedeon T, Wong KW, Lee M, editors. Neural information processing. Cham: Springer International Publishing; 2019, p. 596–607. [http: //dx.doi.org/10.1007/978-3-030-36708-4\_49](http://dx.doi.org/10.1007/978-3-030-36708-4_49).

[44] [Das S, Mishra S, Senapati MR. New appro](http://dx.doi.org/10.1007/978-3-030-36708-4_49)aches in metaheuristic to classify medical data using artificial neural network. Arab J Sci Eng 2019;45(4):2459–71. <http://dx.doi.org/10.1007/s13369-019-04026-y>.

[45] [Kumar N, Kumar D. An improved grey wolf](http://dx.doi.org/10.1007/s13369-019-04026-y) optimization-based learning of artificial neural network for medical data classification. J Inf Commun Technol 2021;20(Number 2):213–48. <http://dx.doi.org/10.32890/jict2021.20.2.4>.

[46] Sharifi A, Alizadeh K. Comp[arison of the particle swarm optimization](http://dx.doi.org/10.32890/jict2021.20.2.4) with the genetic algorithms as a training for multilayer perceptron technique to diagnose thyroid functional disease. Shiraz E-Med J 2020;22(1). [http://dx.doi.org/10. 5812/semj.100351](http://dx.doi.org/10.5812/semj.100351).

[47] [Salman I, Ucan O](http://dx.doi.org/10.5812/semj.100351), Bayat O, Shaker K. Impact of metaheuristic iteration on artificial neural network structure in medical data. Processes 2018;6(5):57. [http: //dx.doi.org/10.3390/pr6050057](http://dx.doi.org/10.3390/pr6050057).

14

*M.G. Rojas et al.*  *Array 14 (2022) 100173*

[73] Wolberg WH, Mangasarian OL. Multisurface method of pattern separa-

tion for medical diagnosis applied to breast cytology. Proc Natl Acad Sci

1990;87(23):9193–6. <http://dx.doi.org/10.1073/pnas.87.23.9193>.

[74] Smith JW, Everhart J[E, Dickson WC, Knowler WC, Johannes RS.](http://dx.doi.org/10.1073/pnas.87.23.9193) Using the ADAP

learning algorithm to forecast the onset of diabetes mellitus. In: Proceedings of

the annual symposium on computer application in medical care. 1988, p. 261–5.

PMC2245318[pmcid].

[75] McDermott J, Forsyth RS. Diagnosing a disorder in a classification benchmark.

Pattern Recognit Lett 2016;73:41–3. [http://dx.doi.org/10.1016/j.patrec.2016.01.](http://dx.doi.org/10.1016/j.patrec.2016.01.004)

[004](http://dx.doi.org/10.1016/j.patrec.2016.01.004).

15