



Adaptive memetic method of multi-objective genetic evolutionary algorithm for backpropagation neural network

Ashraf Osman Ibrahim^{1,2,3} · Siti Mariyam Shamsuddin⁴ · Ajith Abraham⁵ · Sultan Noman Qasem^{6,7}

Received: 12 December 2017 / Accepted: 31 December 2018 / Published online: 21 January 2019
© Springer-Verlag London Ltd., part of Springer Nature 2019

Abstract

In recent years, multi-objective evolutionary optimization algorithms have shown success in different areas of research. Due to their efficiency and power, many researchers have concentrated on adapting evolutionary algorithms to generate Pareto solutions. This paper proposes a new memetic adaptive multi-objective evolutionary algorithm that is based on a three-term backpropagation network (MAMOT). This algorithm is an automatic search method for optimizing the parameters and performance of neural networks, and it relies on the use of the adaptive non-dominated sorting genetic algorithm-II integrated with the backpropagation algorithm, being used as a local search method. The presented MAMOT employs a self-adaptive mechanism toward improving the performance of the proposed algorithm and a local optimizer improving all the individuals in a population in order to obtain better accuracy and connection weights. In addition, it selects an appropriate number of hidden nodes simultaneously. The proposed method was applied to 11 datasets representing pattern classification problems, including two-class, multi-class and complex data reflecting real problems. Experiments were performed, and the results indicated that the proposed method is viable in pattern classification tasks compared to a multi-objective genetic algorithm based on a three-term backpropagation network (MOGAT) and some of the methods mentioned in the literature. The statistical analysis results of the *t* test and Wilcoxon signed-ranks test also show that the performance of the proposed method is significantly better than MOGAT.

Keywords Self-adaptive · Artificial neural networks · Three-term backpropagation · Multi-objective evolutionary · Non-dominated sorting genetic algorithm-II

1 Introduction

The artificial neural network (ANN) is a machine learning method; this method has an architecture that uses mathematical models. The growth in ANNs and their achievements in the previous research show that they are a reliable solution for many computational applications and models in different application areas [1], especially when

addressing very large datasets that have many dimensions [2]. Despite their success with different problems, ANNs cannot reach optimum performance in many nonlinear problems [3], due to the problem of choosing appropriate values for the initial value of the connection weights, structure of the networks (number of hidden nodes), training error and convergence of the learning algorithms. Although the choice of parameters is a very important

✉ Ashraf Osman Ibrahim
a.osman@aou.edu.sd

¹ Faculty of Computer Science and Information Technology, Alzaaiem Alazhari University, 13311 Khartoum North, Sudan

² Arab Open University, Khartoum, Sudan

³ Faculty of computer science, Future University, Khartoum, Sudan

⁴ UTM Big Data Centre, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

⁵ Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, Auburn, Washington, USA

⁶ Computer Science Department, College of Computer and Information Sciences, Al Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia

⁷ Computer Science Department, Faculty of Applied Sciences, Taiz University, Taiz, Yemen

aspect of ANNs, this task is not easy because one parameter can affect the network performance and the adjustment of all of the parameters depends on the user experience. Thus far, the difficulty in determining optimal network parameters is still a major challenge that is faced by users of ANNs. In other words, there is a question of which parameter should be optimized to make the best use of the ANNs and the optimum value of that parameter. Therefore, the optimization of the connection weights, training process and structure of the network has become more attractive in the past few years.

Because ANNs suffer from these problems, evolutionary algorithms (EAs) are used to solve the above problems, evolve the ANNs efficiently and improve network performance. Moreover, they can choose the best connection weights and also reduce the number of hidden nodes with an effective structure for the network size and with positive effects on the network performance [4, 5]. Recent studies have proposed exploiting EA techniques to overcome the above problems [4–9]. Most of these studies utilize EAs for evolving ANNs to gain simple and accurate ANNs. More importantly, the integration of EAs and ANNs is still under research; combining the advantages of each can yield a more efficient method. One of the most successful applications of EAs is their use for evolving ANNs, as shown in [10]. Due to modern applications in many fields in which there are many incompatible objectives, as an alternative to addressing a single optimal solution, a set of optimal solutions called the Pareto optimal set exists for problems, such as multi-objective optimization problem (MOP) [11]. The corresponding objective functions, of which non-dominated solutions are in the Pareto optimal set, are called a Pareto front.

The multi-objective evolutionary algorithms (MOEAs) research area is one of the most active areas in the field of evolutionary computation [12]. Therefore, MOEAs are used to produce and optimize ANN parameters with the optimization of two conflicting objectives, namely the minimization of the ANNs' structural complexity and the maximization of the network's capacity. These types of algorithms are applied to improve the generalization ability, from the training data to the network unseen data. Moreover, MOEAs are suitable to produce and design appropriate and accurate ANNs from the simultaneous optimization of two or more conflicting objectives. Hence, due to their ability to improve the structural performance, recently, MOEAs have been applied successfully to optimize the network structure and connection weights [13–16]. Furthermore, in a single run, they can find multiple solutions [17–20]. However, a considerable number of studies in the literature were applied using these techniques. As an example, multi-objective genetic algorithm optimization was used by [21, 22] to train a feedforward

neural network. Similarly, there are hybrid methods that use ANNs with evolutionary Pareto-based algorithms, and this type of research is known as multi-objective evolutionary artificial neural networks (MOEANNs) [23]. Another method based on the generalized multilayer perceptron (MLP) improved the performance of the evolutionary model [24]. A major study [25] proposed a hybrid multi-objective genetic algorithm (MOGA), which is based on the Strength Pareto Evolutionary Algorithm 2 (SPEA2) and non-dominated sorting genetic algorithm-II (NSGA-II) algorithms to optimize the training and topology of the recurrent neural network (RNN) simultaneously. Recently, [15] applied a non-dominated sorting genetic algorithm-II as a MOGA to train the neural network and optimize its weights and biases with respect to the maximum accuracy and minimum dimension.

On the other hand, memetic algorithms (MAs) have been developed over the past few years. Recently, several studies in the literature have used ANNs, MOEAs and local optimizers to speed up convergence [17, 26, 27]. In addition, Abbass [28] concludes that his proposed memetic Pareto artificial neural network (MPANN), which is based on a Pareto optimal solution, has better generalization and lower computational cost. Almeida and Ludermir [29] proposed a multi-objective memetic and hybrid approach for optimizing the parameters and performance of the ANNs, by using a combination of evolutionary strategies, genetic algorithms and particle swarm optimization. Another study proposed a memetic multi-objective evolutionary neural network algorithm to automatically design ANN models with sigmoid basis units for multi-classification problems [30]. Likewise, [18] is considered to be a memetic Pareto evolutionary approach that is based on the NSGA2 evolutionary artificial neural network algorithm to optimize two conflicting main objectives: a high correct classification rate and a high classification rate for each class. Qasem and Shamsuddin [31] presented a new memetic multi-objective evolutionary algorithm for the design of radial basis function networks (RBFNs) for classification tasks. Recent work in [32] introduced a multi-objective evolutionary learning algorithm using an improved version of the NSGA2 algorithm hybridized with a local search algorithm for training ANNs with generalized radial basis functions.

Modern studies on self-adaptive properties for multi-objective optimization algorithms in the literature indicate that the self-adaptive method can improve performance. Abbass [33] presented a self-adaptive Pareto differential evolution algorithm for multi-objective optimization problems (SPDE) that self-adapts the crossover and mutation rates. In conducted experiments, the SPDE algorithm outperforms the other evolutionary multi-objective optimizations. A multi-objective self-adaptive

differential evolution algorithm with objective-wise learning strategies (OW-MOSaDE) is introduced to solve numerical optimization problems with multiple conflicting objectives [34]. Another mechanism used self-adaptive features for MOEAs, which suggested that the dynamic adjustment of the distribution index of the simulated binary crossover (SBX) operator has been shown [35]. Lately, [36] introduced a study about adaptive memetic computing applied in multi-objective optimization that yielded better results in optimization performance. The results showed the strengths of the proposed technique and proved the efficiency of the proposed adaptive memetic technique.

The memetic adaptive techniques that were applied in multi-objective optimization in a different application benefited from two techniques that improved the process and also led to better accuracy in the final solutions. When using an adaptive and local search technique, the adaptive method can cause dynamic behavior to adjust to the distribution index of the SBX crossover at each generation in the genetic process. This arrangement led the algorithm to produce much better results than the original or fixed SBX crossover. On the other hand, the local search technique includes speeding up the convergence and increasing the quality of the Pareto optimal solutions. It has been observed that both of the mentioned techniques during the evolutionary process can improve the MOEA's performance by exploiting and optimizing the balance between the exploration and exploitation during the various stages of the evolutionary search.

Motivated by this observation, in this study, we present a new memetic adaptive multi-objective evolutionary algorithm that is based on a three-term backpropagation (TBP) network (MAMOT) to optimize the parameters of the TBP network and improve the network accuracy. The adaptive non-dominated sorting genetic algorithm (ANSGA-II) is utilized to optimize three objectives (parameters) simultaneously, namely the number of hidden nodes in the hidden layer, the norm of connection weights and the error of the network, to solve a pattern classification problem. For performance metrics, we used some of the performance metrics that are used for classification problems [37, 38], namely the accuracy, sensitivity, specificity and mean squared error (MSE).

Although EAs have several advantages, these algorithms are slow to converge, which is a major setback [39], and there is difficulty in tuning the final solutions in the search space [40]. To overcome these setbacks, a global search algorithm combined with a local search technique (a memetic process) offers a better speed of convergence for the evolutionary approach and better accuracy of the final solutions. This approach has yielded very promising results in other complex problem solving. At the same time, the flexibility of the crossover operator brought about a

dynamic nature to the proposed method and has been the motivation for this study. However, previous studies indicate that the memetic adaptive methods applied to multi-objective evolutionary algorithms have achieved success in diverse applications. At the same time, no measure has been taken for using this method in the literature, which optimizes and automatically designs the ANNs. Therefore, it can be argued that such an action is a novel approach in this research area.

The novelty of the proposed method came from using an adaptive method with local search technique for designing an artificial neural network. The adaptive method can cause dynamic behavior to adjust to the distribution index of the SBX crossover at each generation in the genetic process. This arrangement led the algorithm to produce much better results than the original or fixed SBX crossover. On the other hand, the local search technique includes speeding up the convergence and increasing the quality of the Pareto optimal solutions. The goal of this proposed method is to generate an automatic design of the ANN structure and to reduce the error rate of the TBP network achieving better performance as well as a better architecture in terms of the hidden nodes.

The remainder of this study is organized as follows: Sect. 2 introduces the materials and methods used in this study. The proposed method and flowchart of the algorithm are illustrated in Sect. 3. The experimental results, datasets, experimental setup, results, discussion and statistical testing are given in Sect. 4. Finally, Sect. 5 concludes the study.

2 Materials and methods

In this section, we highlight the main methods that are used in this paper. The hybrid method is to train the three-term backpropagation neural network, which is dependent on self-adaptive simulated binary crossover method of multi-objective evolutionary algorithm combined with local search technique which can significantly speed up the multi-objective evolutionary algorithm performance.

2.1 Multi-objective evolutionary artificial neural network

The use of evolutionary approaches for ANN training, known as evolutionary artificial neural networks (EANNs), has been a key research area for the past few years [17]. Researchers have developed methods and techniques to find better approaches to evolve ANNs, attempting to find a simple architecture and accurate ANNs with good generalization capabilities. Moreover, there are many advantages of evolutionary approaches for ANN training, with the main advantages being the ability to escape a local optimum, robustness and ability to adapt in a changing environment.

Research into EANNs has usually taken one of three approaches: first, evolving the weights of the network; second, evolving the network architecture; and last, evolving both the weight and architecture simultaneously [10]. The preliminary work of [17] has succeeded in designing networks that have a good generalization capability. However, finding a good ANN architecture has also been discussed in the ANN research literature. The main advantages of the evolutionary approach to ANN training are the ability to escape a local optimum, robustness and the ability to adapt to a changing environment [4, 13, 41]. Multi-objective evolutionary algorithms (MOEAs) represent a population-based search approach, and hence, in a single run, many Pareto optimal sets (solutions) can be obtained, and these are attractive when using this type of algorithm. The current research focuses on the application of multi-objective evolutionary algorithms to solve multi-objective optimization problems in different fields [32, 42–44].

2.2 Parameter optimization

To evaluate the three-term backpropagation network performance of the proposed method, three objective functions were used in this study, as follows:

1. The performance of the network (accuracy) is based on the mean squared error (MSE) on the training set. This performance involves the first objective function and is given below:

$$f_1 = \frac{1}{N} \sum_{j=1}^N (t_j - o_j)^2 \tag{1}$$

where o_j is the network output value of the output unit, t_j is the target value of the output, and N is the number of samples.

2. The complexity of the network is based on the number of hidden nodes in the hidden layer of the TBP network and is a second objective function. This function is computed as follows:

$$f_2 = \sum_{h=1}^H \rho_h \tag{2}$$

where ρ is the dimension of H and H is the maximum number of hidden nodes in the network. $\rho_h \in \rho$ is a binary value {0,1} used to refer to the hidden node with respect to whether it exists in the network or not. Turning a hidden unit ON or OFF works like a switch; this mechanism is involved in determining the maximum number of hidden nodes in the TBP network.

3. The complexity of the TBP network is based on the weights of the network, which is based on the notion of regularization and represents the smoothness of the

model. This function is the last objective of this study (f_3) and is given as:

$$f_3 = \{||\omega||\} \tag{3}$$

where ω is a matrix of weights in the network.

In this study, three fitness functions were analyzed to optimize the performance of the network (f_1), to minimize the structure of the network (hidden nodes) (f_2) and (f_3) to minimize the connections (weights) of the TBP network.

2.3 Three-term backpropagation algorithm

The three-term backpropagation network (TBP) is a type of ANN and has been proposed by Zweiri et al. [45] to speed up the weight adjustment process by increasing the convergence rate of the algorithm and reducing the learning stalls. The TBP network modifies the architecture and procedure of the standard backpropagation (BP) algorithm by adding an extra term to increase the BP learning speed [46]. The neurons in all of the layers are connected with connection links. A weight is associated with each connection link and is multiplied with the signal that exits within each neuron in the network (from input to hidden and from hidden to output layer), see Fig. 1.

In TBP network, in addition to the learning rate and momentum parameters, a third parameter, called the proportional factor (PF), is introduced. This presentation of PF has proven to be successful in improving the convergence rate of the algorithm and speeding up the weight adjustment process.

$$net_j = \sum_{i=1}^M W_{ij}O_i + \theta_j \tag{4}$$

$$O_j = \frac{1}{1 + e^{-net_j}} \tag{5}$$

where net_j is the summation of the weighted inputs added to the bias, W_{ij} is the weighted value between input layer i and hidden layer j , O_i is the output from the input layer i at

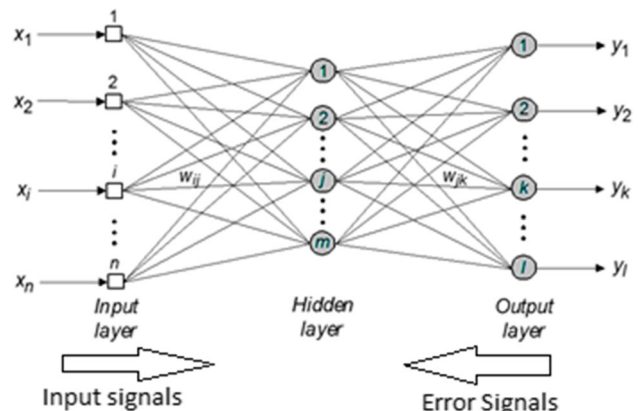


Fig. 1 ANN architecture

the same time that it is the input to the hidden layer j , and θ_j is the bias associated with each connection link between the two respective layers. Equation (5) shows the calculation of O_j , which is the output of the activation function at the hidden layer j .

$$E = \frac{1}{L} \sum_{k=1}^L (t_k - o_k)^2 \tag{6}$$

where E is the error function of the network mean squared error, t_k is the target output at output layer k , and the network has L output neurons.

Consider W as network weights vector, k as iteration number of the weight vector, and $\Delta W(k) = W(k + 1) - W(k)$. The weight adjustment in Eq. (7) is modified to include the proposed third parameter, which is proportional to the difference between the desired and calculated output in Eq. (8). Thus, we can say that Eq. (7) presented the two-term backpropagation, while Eq. (8) shows the three-term backpropagation.

$$\Delta W(k) = \alpha(-\nabla E(W(k))) + \beta\Delta W(k - 1) \tag{7}$$

$$\Delta W(k) = \alpha(-\nabla E(W(k))) + \beta\Delta W(k - 1) + \gamma e(W(k)) \tag{8}$$

where α and β are the learning rate and momentum, respectively; γ is the proportional factor; and $e(W(k))$ represents the difference between the output and the target at each iteration.

2.4 Adaptive NSGA-II

The genetic algorithm (GA) is based on simulating the biological evolution of the search space in the search process automatically, and it is a parallel global search method [47]. The non-dominated sorting genetic algorithm-II (NSGA-II) is proposed in [48] because of its good performance in global searching. The NSGA-II method proposes a new method and a new arithmetic operator by improving the first version of the NSGA [43]: the fast non-dominated sorting approach and the crowded comparison operator. Thus far, many studies regarding optimization and design have been performed [4, 32, 49, 50]. All of these studies prove that the genetic algorithm and its upgraded derivatives are feasible for optimal design. Recently, many studies have proven that the adaptive multi-objective optimizations are valuable and able to achieve better results in a variety of applications [35, 51, 52]. Using the self-adaptive crossover operator can dynamically adapt the solution and can create children solutions in an appropriate way from the parents. The update process for the distribution index can be increased or decreased for the next generation depending on how the child outperforms (has a better fitness value than) the

parents. These processes comprise the NSGA-II Adaptive algorithm, which is called ANSGA-II. This methodology guarantees improving the solution.

The main idea for the crowding distance is to find the Euclidian distance between every individual front, using the objectives in the dimensional hyperspace. Individuals at the boundary with infinite distance are always selected. The crowding distance is assigned once the non-dominated sort is completed. Having picked individuals based on crowding distance ranking, individual populations are assigned to the crowding distance value. Thus, reversing the labels by assigning front wise and comparing the crowding distance between two individuals is meaningless, see Fig. 2.

2.5 Self-adaptive simulated binary crossover

Self-adaptation techniques are based on a population’s diversity. Whereas the adaptation of the operator ensures a good convergence speed, the degree of diversity determines the convergence reliability. Generally, the relationship between the parent and offspring population is controlled with a self-adaptation technique. A self-adaptive simulated binary crossover (SA-SBX) was proposed by [51] to adjust the distribution index of the simulated binary crossover (SBX) operator dynamically at each generation in NSGA-II. Compared to the SBX, several studies have reported that the SA-SBX produces better solutions when applied to both single- and multi-objective optimization problems [35, 51]. The important factor in the SBX crossover is finding the appropriate value of the distribution index (η_c) because it has an effect on the convergence speed and local/global optimum solution, and thus, the self-adaptive SBX adaptively updates the distribution index to solve these problems. Moreover, self-adaptive simulated binary crossover (SA-SBX) at each generation in the NSGA-II procedure can dynamically adjust the distribution index η_c of the crossover operator adaptively. As is well known, the crossover operator in genetic algorithms produces children by recombining the information from the parents. If the child is better than the parent, then the child is extended

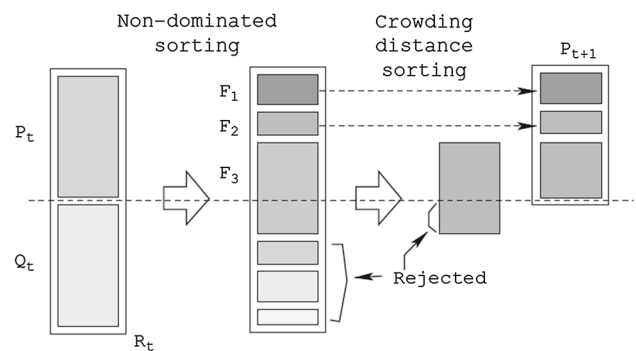


Fig. 2 Standard NSGA-II technique

further in the hope of creating a better solution, while the opposite occurs if a worse solution is created.

The process of calculating offspring solutions $x_i^{(1,t+1)}$ and $x_i^{(2,t+1)}$ from the parent solutions $x_i^{(1,t)}$ and $x_i^{(2,t)}$ appear in Eq. (9). In addition, the spread factor β_i is defined as the ratio of the absolute difference in the offspring values to that of the parents and described in Eq. (9) as well:

$$\beta_i = \left| \frac{x_i^{(2,t+1)} - x_i^{(1,t+1)}}{x_i^{(2,t)} - x_i^{(1,t)}} \right| \quad (9)$$

A random number, u_i , is created and ranges between 0 and 1, which establishes a probability distribution function. The probability distribution in Eq. (10) is graphically shown in Fig. 3 for $\eta_c = 2$ and 5, which is used for making offspring from two parent solutions ($x_i^{(1,t)} = 2.0$ and $x_i^{(2,t)} = 5.0$). In expression (10), η_c is any nonnegative real number. From Fig. 3, it can be seen that a large value for η_c yields a higher probability for creating (near parent) solutions and, consequently, for providing a pathway for a focused search. Moreover, a small value of η_c permits distant solutions to be chosen as offspring, which permits diverse searches. Please see [51] for more details about this technique.

$$P(\beta_i) = \begin{cases} 0.5(\eta_c + 1)\beta_i^{\eta_c}, & \text{if } \beta_i \leq 1; \\ 0.5(\eta_c + 1)\frac{1}{\beta_i^{\eta_c+2}}, & \text{otherwise.} \end{cases} \quad (10)$$

2.6 Local search algorithm

A local search algorithm is a meta-heuristic approach that is used to solve hard optimization tasks. In a local search

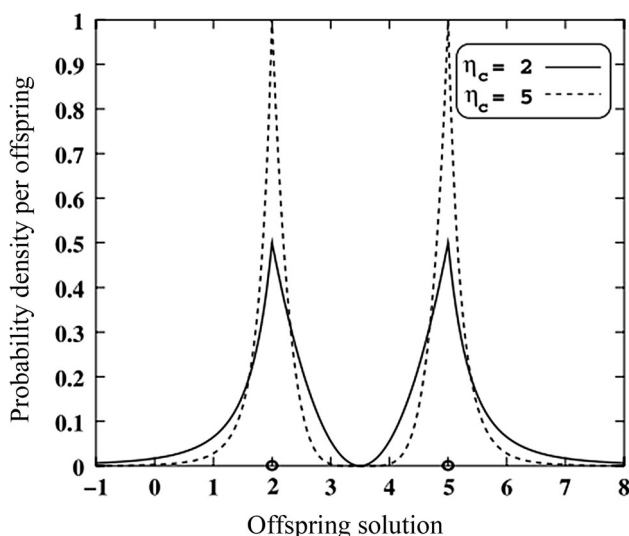


Fig. 3 Probability density function for creating offspring solutions with the SBX operator [51]

method, the algorithm moves through the search space and searches for a solution from a number of solutions by applying local changes; this process is continued until one of the solutions is considered to be optimal or after the expiration of a specified amount of time.

Local search algorithms are widely used for several problems in different areas, but they have received more attention in computer science and engineering, especially in artificial intelligence applications. It is known that the local methods can find a local optimum when searching in a small area of space. Therefore, with a combination of EA and local search algorithms, EAs perform a global search within the space of solutions and use ANNs to locate solutions near the global optimum and to apply a local search method to quickly and efficiently find the best solution. This type of hybrid algorithm is known as a memetic algorithm (MA). MAs can provide not only the best speed of convergence to the evolutionary approach but also the best accuracy for the final solutions [53].

There are several studies [14, 28] that use MOEAs along with ANN local optimizers to adjust the weights. This approach is called lifetime learning, and it consists of updating each individual with respect to the approximation error. The main problem with this type of algorithm is the computational cost. Some studies used a local search algorithm after the crossover and mutation operations for all of the individuals in the population in each iteration [17, 28, 54]. As is well known, the BP algorithm is a learning heuristic for supervised learning in ANNs. Therefore, in this study, we used a classical BP algorithm as a local search method.

3 The proposed memetic adaptive multi-objective genetic evolutionary algorithm

This section introduces a memetic adaptive multi-objective genetic evolutionary algorithm (MAMOGEA) that is basically adapted from the non-dominated sorting genetic algorithm-II (NSGA-II) [48] and modifies the crossover operator to self-adaptive crossover, hybridized with BP as a local search algorithm to optimize TBP network parameters being implemented for solving pattern classification problems. The network architecture and accuracy are evolved simultaneously, with each individual being a fully specified TBP network. In this study, MAMOGEA based on the TBP network has been proposed to determine the best parameters, performance and corresponding architecture of the TBP network, which we call MAMOT.

In addition, self-adaptive simulated binary crossover (SA-SBX) [51] gives the proposed method the property of having a dynamic nature to the distribution index; it can automatically update the crossover operator, providing the

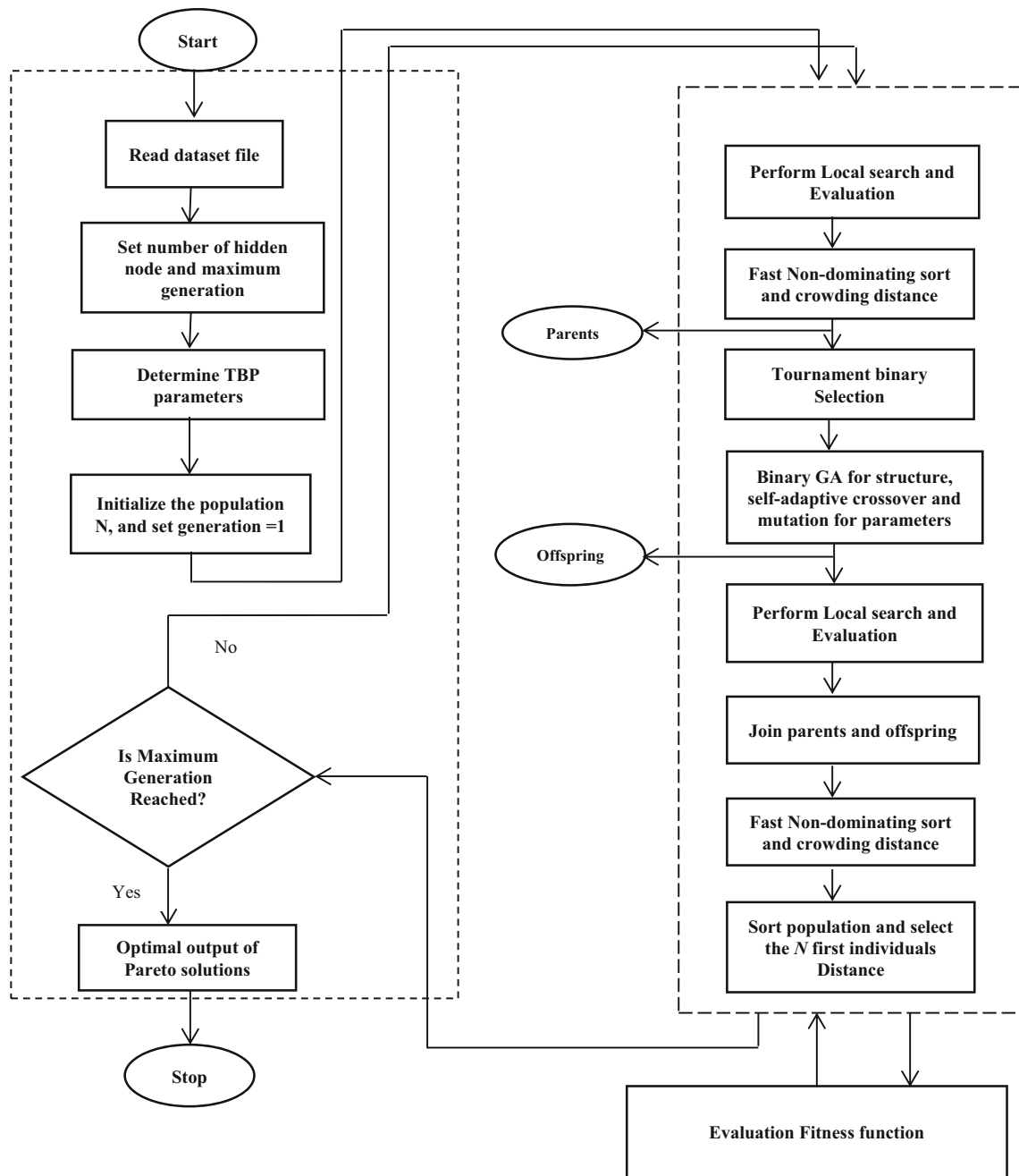


Fig. 4 Flowchart of the proposed MAMOT

ability to create child solutions in the true direction from the parents. If the child solutions that are produced by this process are worse than the parent solutions, then it can provide a move to obtain a modified child to move very near to the parents' results. This process can optimize the balance between exploration and exploitation during the various stages of the evolutionary search. The ANSGA-II is hybridized with the BP algorithm as a local search algorithm to enhance all of the individuals in the population to improve the classification accuracy. At the same

time, the above scenario helps the proposed method to produce good final solutions. These solutions represent three objectives and the following analysis: (1) optimize the performance of the network (f_1); (2) minimize the network complexity based on the number of hidden nodes in the hidden layer of the TBP network (f_2); and (3) minimize the connection weights of the TBP network (f_3), which is based on the notion of regularization and represents the smoothness of the model. To measure the network complexity, the study uses both of the objectives, f_2 and f_3 .

The attempt to minimize f_2 leads to a lower number of hidden nodes in the hidden layer of the TBP network, while an attempt to minimize f_3 leads to a lower matrix of weights. However, to assist in the TBP network design, GA and MOEA are combined as a rank-density-based GA to perform the fitness evaluation and mating selection schemes. Similarly, the MAMOT begins by collecting, normalizing and reading the dataset. The result determines the dataset. Then, the number of hidden nodes and the maximum number of iterations are set. Additionally, the individual dimension is determined. Furthermore, it

generates and initializes a population of the TBP network, and during the experiment, the initialization is set before each TBP training generation.

Every individual is evaluated for every iteration based on the objective functions. After the maximum iterations are reached, the proposed method stops and outputs a set of non-dominated TBP networks. Figure 4 shows the flowchart of the MAMOA based on the TBP network. Furthermore, the proposed method is given in the following steps:

1. Generate a random population $P(g)$, $g = 0$, of size N , and g is the number of the current generation.
2. Evaluate the individuals $P(g)$ based on three objectives (error, connection weights and hidden nodes).
3. Utilization of fast non-dominated sorting for obtaining a list F with the fronts of the population $P(g)$.
4. Assign to each individual a rank value that is equal to its dominance level and a crowding distance value.
5. While the stopping condition is not met:
 - 5.1. Use a binary tournament for selecting N individuals of $P(g)$ on the basis of the individual rank and the crowding distance value.
 - 5.2. Execute single point crossover and bitwise mutation for the binary encoding of ρ (network structure/hidden nodes) and the Self-Adaptive simulated binary crossover (SA-SBX) operator and polynomial mutation for the real encoding of ω (parameters of the network) for individuals of $P(g)$ that were selected to generate a new offspring population $Q(g)$ of size N .
 - 5.3. Perform a local search algorithm with each individual and evaluate each member of the population $Q(g)$ based on its accuracy and complexity.
 - 5.4. $f = f + 1$ and $R(g) = P(g) \cup Q(g)$.
 - 5.5. $F =$ fast-non-dominated-sort ($R(g)$).
 - 5.6. While the size of the population $P(g+1) < N$.
 - 5.6.1 Calculate s , the crowding-distance for the front F^f
 - 5.6.2 $P(g+1) = P(g+1) \cup F^f$
 - 5.6.3 $f = f + 1$ // increment the number of the Pareto front.
 - 5.7. end while
 - 5.8. Sort population $P(g+1)$ according to the rank and crowding value, and select the first N individuals. The new population $P(g+1)$ of size N is now completed.
 - 5.9. $g = g + 1$.
6. end while.

Pseudo-code of proposed memetic adaptive multi-objective genetic evolutionary algorithm

Table 1 A summary of the methods used in the literature

References	Description	Notes	Method
Abbass [26]	Memetic Pareto artificial neural network (MPANN)	Based on a Pareto optimal solution, MPANN has better generalization and good results	Memetic
Abbass [31]	A self-adaptive Pareto differential evolution algorithm for multi-objective optimization problems (SPDE), that self-adapts the crossover and mutation rates	In the experiments that were conducted, the SPDE algorithm outperforms the other evolutionary multi-objective optimizations	Self-adaptive
Huang et al. [32]	Multi-objective self-adaptive differential evolution algorithm with objective-wise learning strategies (OW-MOSaDE)	To solve numerical optimization problems with multiple conflicting objectives	Multi-objective self-adaptive
Pettersson et al. [19]	Multi-objective genetic algorithm optimization to train a feed forward neural network	Using noisy data from an industrial iron blast furnace	Multi-objective
Almeida and Ludermir [27]	Multi-objective memetic and hybrid of evolutionary strategies, genetic algorithms and particle swarm optimization	By using for optimizing the parameters and performance of the ANNs	Multi-objective + memetic
Cruz-Ramírez et al. [28]	Memetic multi-objective evolutionary neural network algorithm to automatically design ANN models with sigmoid basis units	Used for multi-classification problems	Memetic + multi-objective
Fernandez Caballero et al. [16]	Memetic Pareto evolutionary approach that is based on the NSGA2 evolutionary artificial neural network algorithm	Use to optimize two conflicting main objectives: a high correct classification rate and a high and for classification rate for each class	Memetic
Delgado et al. [23]	Hybrid multi-objective genetic algorithm (MOGA), which is based on the Strength Pareto Evolutionary Algorithm 2 (SPEA2) and non-dominated sorting genetic algorithm-II (NSGA-II) algorithms to optimize the training and topology of the recurrent neural network (RNN) simultaneously	Applied for time-series (CATS) benchmark	Hybrid multi-objective
Ak et al. [14]	A non-dominated sorting genetic algorithm-II to train the artificial neural network. They optimize its weights and biases with respect to the maximum accuracy and minimum dimension	Used to estimation of prediction intervals of scale deposition rate in oil gas equipment	Multi-objective
Qasem and Shamsuddin [29]	Memetic multi-objective evolutionary algorithm (Elitist Pareto Differential Evolution algorithm) for the design of radial basis function networks (RBFNs)	Used for classification tasks	Memetic + multi-objective
Cruz-Ramírez et al. [30]	A multi-objective evolutionary learning algorithm using NSGA2 algorithm hybridized with a local search algorithm for training ANNs with generalized radial basis functions	Used for donor–recipient decision system in liver transplants	Multi-objective

Pseudo-code of proposed memetic adaptive multi-objective genetic evolutionary algorithm.

The method starts by generating a random population $P(g)$, $g = 0$, of size N . Evaluation of the individuals $P(g)$ based on three objectives was mentioned in the section on parameter optimization. Then, the population is sorted according to the non-domination aspect, and the solution ranks are provided based on the non-domination levels and a crowding distance value. The procedure is described as follows: First, the usual binary tournament selection and the SA-SBX crossover and mutation are used for binary encoding, and also, the mutation and SA-SBX crossover operators are used for real encoding to create an offspring population $Q(g)$ of size N . Second, apply the BP local

method to each individual of the offspring population $Q(g)$ and evaluate the individuals of the population $Q(g)$ based on their accuracy and complexity. Because elitism is introduced by comparing the current population with the previously found best non-dominated solutions, the procedure is presented in each generation as a combined population $R(g) = P(g) \cup Q(g)$ that is formed, with the size of the $R(g)$ population being equal to $2N$. Afterward, according to the non-domination criteria, the population $R(g)$ is sorted, and the best solutions in the population are those that belong to the best non-dominated set F^1 . If the size of F^1 is smaller than N , then all of the members of the set F^1 are definitely chosen for the new population $P(g+1)$. The remaining members of the population $P(g+1)$ are

chosen from subsequent non-dominated fronts in their order of ranking. Thus, solutions from the set F^2 are chosen next, followed by solutions from the set F^3 , and so on. This procedure is continued until no more sets can be accommodated. Then, the new population $P(g+1)$ is sorted according to the rank and crowding values, and the first N individuals are selected. Finally, we use a binary tournament on $P(g+1)$ to obtain N individuals (Table 1).

4 Experiments

4.1 Datasets

For the experimental design, we consider 11 real-world datasets for classification tasks. The datasets include two-class, multi-class and complex real problem pattern classification tasks. The breast cancer, diabetes, heart, hepatitis and liver datasets represent two-class datasets, while the iris, lung cancer, QAC, segment, wine and yeast data represent multi-class datasets. All of the datasets are obtained from the UCI machine learning repository [55], except for the Qualitative Analytical Chemistry (QAC) dataset, which is sometimes called BTX, and it is considered to also be a complex problem. More information can be found about QAC in [56]. Table 2 shows the number of features, classes and instances for all of the datasets. For each dataset, 75% of the dataset is used for the training set, and the remaining 25% is used for the testing set. In addition, all of the dataset values are normalized in the range [0, 1].

4.2 Experimental setup

The experiments are conducted to test the efficiency of the proposed method for all of the datasets. The proposed

method is evaluated by using the tenfold cross-validation technique. In tenfold cross-validation, the dataset is split into ten equally sized subsets. Nine of those subsets are used as the training dataset, and the one remaining subset is used as the testing dataset. This train and test process is repeated in such a way that all of the subsets are used as a test dataset. The results of MAMOT for each dataset are compared to MOGAT. The number of input and output nodes is dependent on the dataset, and it is different from one dataset to another. The maximum number of hidden nodes is set to 10 [17, 23, 54]. The maximum number of neural network training iterations is set to 1000 [2, 46] for all of the datasets. For the local search algorithm, the learning rate is set to 0.01 and the number of iterations is set to 5 [17, 54]. Table 3 presents the other various parameter settings. From Table 3, the “ N ” refers to the dimension of the individual. Moreover, there are some parameters of the TBP network that must be specified by the user. In the MAMOT, we assign a distribution index η_c value in the initial population using $\eta_c = 2$. Afterward, this value is updated depending on the creation of a better child or a worse child than both parents.

4.3 Results and discussion

This section presents the results of MAMOGEA and MOGA applied to the TBP network (called MAMOT and MOGAT, respectively). MAMOT is the proposed method in this paper, it is a new memetic adaptive multi-objective evolutionary algorithm based on a three-term backpropagation network. It used self-adaptive NSGA-II combined with the local search method to optimizing the parameters and performance of neural networks. On the other hand, MOGAT is proposed in [14] and implemented in this for comparison to MAMOT. It used non-dominated sorting genetic algorithm-II based on TBP network. Both

Table 2 Summary of datasets used in the experiments

Dataset	Number of features	Number of classes	Number of instances
Breast cancer	9	2	683
Diabetes	8	2	768
Heart	13	2	297
Hepatitis	19	2	155
Iris	4	3	150
Liver	6	2	345
Lung cancer	56	3	32
QAC	3	7	63
Segment	19	7	2310
Wine	13	3	178
Yeast	8	10	1484

Table 3 Parameter settings for the proposed algorithms

Algorithm	Value
<i>Local search algorithm</i>	
Learning rate	0.01
Number of iterations	5
<i>TBP network</i>	
Maximum number of hidden nodes	10
Learning rate	0.7
Momentum	0.7
Proportional Factor	0.9
<i>MAMOT</i>	
Maximum number of iterations	1000
Population size	100
Probability of crossover (P_c)	0.9
Probability of mutation (P_m)	1/N
Distribution indices for crossover (η_c)	20
Distribution indices for mutation (η_m)	20

MAMOT and MOGAT are based on TBP network. The main differences between them are the MAMOT used self-adaptive method to improve the performance of the algorithm and a local search technique to improve all of the individuals in a population.

The results of these algorithms are Pareto optimal solutions to improving generalization on unseen data. The training set is used to train the TBP network to obtain the Pareto optimal solutions, while the testing set is used to test the generalization performance of the Pareto TBP network. The result for each dataset focused on the analysis of the hidden nodes, network error and accuracy, and the results are analyzed based on the convergence to the Pareto front with their classification performance. In Tables 4, 5, 6 and 7, the best results are the highlighted bold entries.

The hybrid of the local search algorithm with evolutionary algorithms is a good choice in the problems studied because the hybrid algorithms achieve the best performance in all of the datasets. In addition, self-adaptive crossover helped the algorithms in MAMOT to obtain better solutions than MOGAT in most cases. Therefore, the crossover operator benefited from the adaptive process. In fact, the algorithm MAMOT has obtained the best performing networks in all of the problems. Moreover, the size of the networks obtained by this algorithm is, in general, smaller than MOGAT because of the selective pressure produced as a result of the objectives in Eqs. (2) and (3) working together. To evaluate the classification capabilities of the proposed method, the performance of the MAMOT and MOGAT in the average sensitivity, specificity and accuracy was performed with the results shown in Tables 6 and 7 in addition to Figs. 8, 9 and 10.

Table 4 Statistical evaluation of the testing errors for MAMOT and MOGAT methods

Dataset	MAMOT Testing error	MOGAT Testing error
<i>Breast cancer</i>		
Mean	0.0192	0.0241
SD	0.0064	0.0082
<i>Diabetes</i>		
Mean	0.1642	0.1719
SD	0.0134	0.0157
<i>Heart</i>		
Mean	0.1077	0.1219
SD	0.0288	0.0255
<i>Hepatitis</i>		
Mean	0.1166	0.1311
SD	0.0249	0.0251
<i>Iris</i>		
Mean	0.1087	0.1196
SD	0.0099	0.0114
<i>Liver</i>		
Mean	0.2128	0.2212
SD	0.0288	0.0128
<i>Lung cancer</i>		
Mean	0.1674	0.1987
SD	0.0191	0.0291
<i>QAC</i>		
Mean	0.1181	0.1181
SD	0.0057	0.0066
<i>Segment</i>		
Mean	0.0963	0.1309
SD	0.0187	0.0144
<i>Wine</i>		
Mean	0.1459	0.1563
SD	0.0337	0.0242
<i>Yeast</i>		
Mean	0.0805	0.0816
SD	0.0094	0.0088

Tables 8 and 9 show the robust tests for normality and a paired sample test, respectively. Robust tests for normality used to check the normality assumption and a paired sample t test are used after the normality test to compare the proposed methods (MAMOT with the MOGAT). This test is to ensure that there is no statistically significant difference in the means between the accuracy obtained by the proposed methods.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FP}}\% \tag{11}$$

Table 5 Average numbers of the hidden nodes in TBP network obtained by MAMOT and MOGAT methods

Dataset	Network complexity (Hidden nodes)	
	MAMOT	MOGAT
<i>Breast cancer</i>		
Mean	3.80	4.70
SD	1.23	1.64
<i>Diabetes</i>		
Mean	4.40	5.60
SD	2.01	2.72
<i>Heart</i>		
Mean	3.60	4.60
SD	1.17	1.26
<i>Hepatitis</i>		
Mean	4.60	5.20
SD	2.11	2.53
<i>Iris</i>		
Mean	4.70	5.20
SD	2.41	2.82
<i>Liver</i>		
Mean	4.70	4.60
SD	1.70	1.70
<i>Lung cancer</i>		
Mean	2.00	2.00
SD	0.00	0.00
<i>QAC</i>		
Mean	3.40	4.10
SD	1.17	1.20
<i>Segment</i>		
Mean	4.10	4.80
SD	0.99	1.14
<i>Wine</i>		
Mean	4.10	4.90
SD	1.37	2.18
<i>Yeast</i>		
Mean	3.4	3.50
SD	0.84	0.85

Table 6 Statistical evaluations for the testing accuracy of the MAMOT and MOGAT methods

Dataset	MAMOT testing accuracy	MOGAT testing accuracy
<i>Breast cancer</i>		
Mean	97.94	96.69
SD	0.75	1.01
<i>Diabetes</i>		
Mean	76.29	73.98
SD	3.05	3.46
<i>Heart</i>		
Mean	85.16	82.78
SD	4.76	3.81
<i>Hepatitis</i>		
Mean	84.01	80.82
SD	5.01	3.73
<i>Iris</i>		
Mean	85.13	79.26
SD	2.76	5.80
<i>Liver</i>		
Mean	64.16	61.35
SD	5.62	4.40
<i>Lung cancer</i>		
Mean	73.96	69.26
SD	5.03	9.17
<i>QAC</i>		
Mean	85.95	85.78
SD	0.75	0.20
<i>Segment</i>		
Mean	87.20	85.71
SD	1.93	0.00
<i>Wine</i>		
Mean	78.84	73.74
SD	6.55	1.34
<i>Yeast</i>		
Mean	90.03	90.01
SD	0.06	0.03

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \% \quad (12)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \% \quad (13)$$

where TP = true positives, FN = false negatives, TN = true negatives, and FP = false positives.

The performance measures used in this study for the classification of the datasets are the sensitivity, specificity and accuracy. Sensitivity is the measure of the classifier according to its ability to identify the correct positive samples, and it depends on the number of true positives and

false negatives, as shown in Eq. (11). Specificity is a measure of the classifier's ability to predict correctly the negative samples; it depends on the number of the true negatives and false positives, as shown in Eq. (12). Additionally, accuracy is a measure of the classifier's ability to produce a level of accurate diagnosis; Eq. (13) shows the accuracy formula.

The results in Table 4 demonstrate the performance of the proposed method (training and testing error) for all eleven datasets used. The average of the results values and the sample standard deviations determine how far each value in the results varies from the average value of the result and the maximum value and minimum value, which

Table 7 Sensitivity and specificity for the testing sets for MAMOT and MOGAT methods

Dataset	MAMOT testing		MOGAT testing	
	Sensitivity	Specificity	Sensitivity	Specificity
<i>Breast cancer</i>				
Mean	97.07	97.80	96.67	97.30
SD	2.63	1.54	2.64	1.77
<i>Diabetes</i>				
Mean	50.43	90.10	48.43	89.20
SD	9.58	3.75	10.92	4.34
<i>Heart</i>				
Mean	80.25	89.73	78.85	88.13
SD	8.03	7.82	7.94	8.04
<i>Hepatitis</i>				
Mean	35.34	95.83	30.83	94.23
SD	23.34	5.83	25.47	6.84
<i>Iris</i>				
Mean	55.36	98.21	52.67	97.00
SD	14.79	3.59	15.54	5.76
<i>Liver</i>				
Mean	26.38	91.20	21.38	90.50
SD	20.26	9.25	19.95	9.85
<i>Lung cancer</i>				
Mean	42.03	92.78	55.00	71.11
SD	17.89	9.46	27.27	14.30
<i>QAC</i>				
Mean	0.00	99.76	0.00	100.00
SD	0.00	0.75	0.00	0.00
<i>Segment</i>				
Mean	1.67	99.18	0.00	100.00
SD	5.27	2.58	0.00	0.00
<i>Wine</i>				
Mean	52.96	96.88	45.96	96.15
SD	19.57	5.91	22.12	6.00
<i>Yeast</i>				
Mean	0.00	100.00	0.00	100.00
SD	0.00	0.00	0.00	0.00

appear in Table 4 as the mean, SD, max and min, respectively. The average of the mean squared error (MSE) of the proposed methods for all of the datasets is presented in this Table. The error rates for all of the results are shown in the same Table, as obtained by MAMOT and MOGAT. The results show the generalization error of the proposed methods. From Table 4, we can observe that in all of the datasets on the mean rows, MAMOT gives promising results regarding the performance (testing error) compared to MOGAT. Additionally, MAMOT produced the smallest error on all of the datasets. Furthermore, the testing errors

that are shown in the same Table are the average of the errors obtained in a single run of the MAMOT and MOGAT applied to the TBP network.

Moreover, Fig. 5 shows the comparison of all of the errors obtained in the training and testing set, respectively, using MAMOT and MOGAT. From the same Figure, the Y-axis plots the MSE, while the X-axis plots the datasets used in this study. We can see that the error rates are reasonable and small in all of the datasets, especially in the breast cancer dataset, which has the lowest amount of error, followed by the yeast data.

Regarding complexity, Table 5 presents the results of the complexity or the average number of hidden nodes in the TBP network structure. Of all of the datasets, the MAMOT achieved a better network structure with the lowest complexity and lower average results for the hidden nodes than MOGAT. In addition, from Table 5, we can observe very clearly that the average number of hidden nodes in the structure of the TBP network in all of the datasets is not more than 4.7 when using MAMOT. Precisely, two datasets obtained 4.7, which are the iris and the liver datasets, while the average number of hidden nodes of the MOGAT is not more than 5.60, which was achieved by the diabetes dataset. On the other hand, the minimum average number of hidden nodes when using the lung cancer dataset is 2.00 in both algorithms. If there are more hidden nodes in the network, we can learn a training set more quickly, but it might not generalize well on a testing set. Therefore, Table 5 and Fig. 6 show that MAMOT has the capability to design simple TBP networks with the lowest number of hidden nodes.

In terms of the classification accuracy rate for all of the datasets, the accuracy rates demonstrate very good results in general, especially in the breast cancer and yeast dataset. As shown in Table 6, the two mentioned datasets obtained 97.94 and 90.03%, respectively, in MAMOT, while using MOGAT they obtained an accuracy rate on the same two datasets of 96.69 and 90.01%, respectively. Table 6 shows the highest testing accuracy highlighted with bold font for all of the datasets. In general, the best results for classification accuracy in the testing sets are obtained through the MAMOT approach for all datasets. Figure 7 clearly shows the average percentage of accuracy obtained in the testing accuracy for all datasets.

Table 7 shows the statistical results for the sensitivity, specificity and classification accuracy of the proposed methods on the training set and testing set for all datasets. In terms of the sensitivity, MAMOT produced the best results on the training and testing set for all of the datasets and on average as well, except for the lung cancer dataset, which had better results using MOGAT for the training and testing set, although the breast cancer dataset had the best values for sensitivity among all of the datasets, obtaining

Table 8 Results of the normality tests

Test	Accuracy				Complexity			
	MAMOT		MOGAT		MAMOT		MOGAT	
	Statistic	<i>p</i> value	Statistic	<i>p</i> value	Statistic	<i>p</i> value	Statistic	<i>p</i> value
RJB	$X^2 = 0.90$	0.64	$X^2 = 0.12$	0.94	$X^2 = 3.87$	0.14	$X^2 = 19.74$	< .005
JB	$X^2 = 0.36$	0.84	$X^2 = 0.18$	0.91	$X^2 = 2.76$	0.25	$X^2 = 5.03$	0.08
SJ	Standardized SJ = 1.77	0.09	Standardized SJ = 0.60	0.29	Standardized SJ = 1.323	0.14	Standardized SJ = 3.31	0.02
SK	$\chi^2 = 1.69$	0.43	$\chi^2 = 0.21$	0.90	$\chi^2 = 5.98$	0.05	$\chi^2 = 8.31$	0.02
SW	$W = 0.99$	0.76	$W = 0.99$	0.99	$W = 0.88$	0.09	$W = 0.84$	0.03
AD	$A = 0.39$	0.30	$A = 0.14$	0.96	$A = 0.47$	0.20	$A = 0.73$	0.04

Table 9 Results of paired *t* test and Wilcoxon’s signed-ranks test

Algorithm	Measure	Mean	SD	Mean positive rank	Mean negative rank	Test Statistic	<i>p</i> value <i>t</i> test	<i>p</i> value signed rank
MOGAT versus MAMOT	Accuracy	- 2.66182	1.93539	-	-	4.561	0.001	-
	Complexity	-	-	5.94	1.5	- 2.654	-	0.008

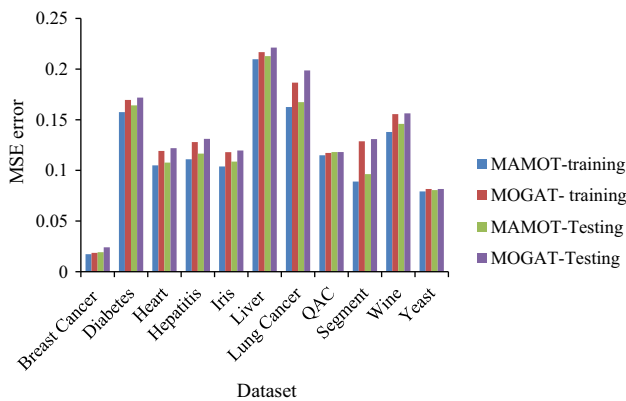


Fig. 5 Comparison of training and testing average error results obtained by MAMOT and MOGAT

98.87% for the training set and 97.07% for testing set. The yeast dataset achieved the lowest sensitivity value (which was 0.00%) for both the training and testing set. Some of the datasets, such as QAC, segment and yeast, achieved very low values in the sensitivity and did not exceed 6.50%. We infer that the improvement in the sensitivity is very difficult in these datasets because there are difficult classification problems in these datasets and because the datasets are extremely unbalanced. Thus, this difficulty leads to lower sensitivity in these datasets. Moreover, Fig. 8 shows the comparison of MAMOT and MOGAT

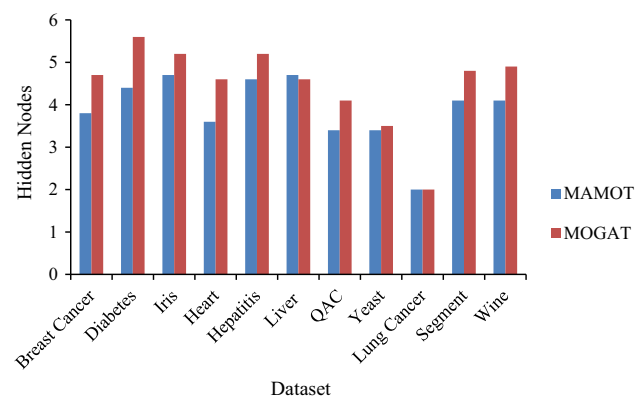


Fig. 6 Comparison of MAMOT and MOGAT for the average number of hidden nodes for all of the datasets

with respect to the sensitivity results obtained in the testing set.

With regard to the specificity in Table 7, MAMOT provided the best results on the training and testing sets for all of the datasets and on average as well, except for two datasets, which are the QAC and Segment datasets. The results reported in Table 7 and the histogram in Fig. 9 show that MAMOT and MOGAT produced the same specificity results only for the yeast dataset. MAMOT has better results in training and testing, especially in the iris and yeast datasets, which obtained 98.18% in training and 98.21% in the testing set for the iris data, while the yeast

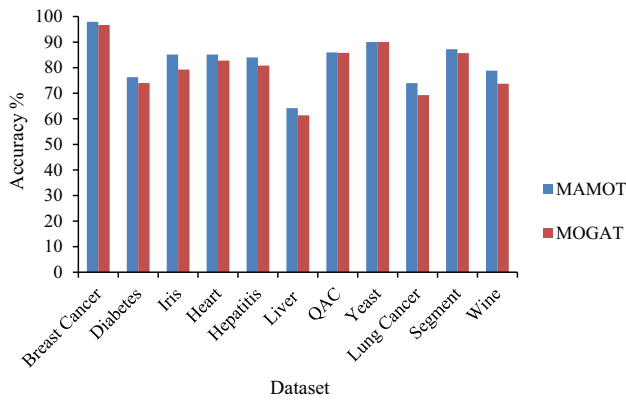


Fig. 7 Comparison of MAMOT and MOGAT with respect to the accuracy results obtained on the testing set

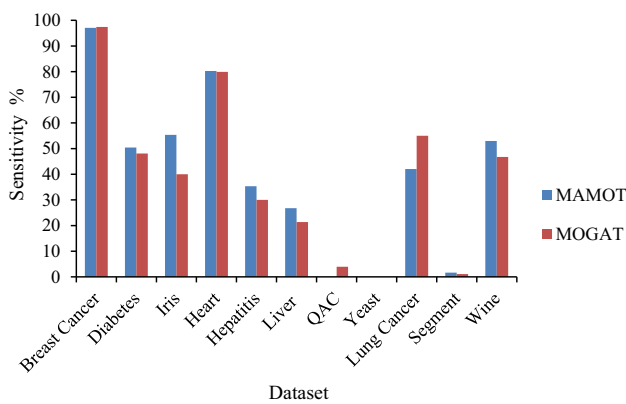


Fig. 8 Comparison of MAMOT and MOGAT with respect to the sensitivity results obtained on the testing set

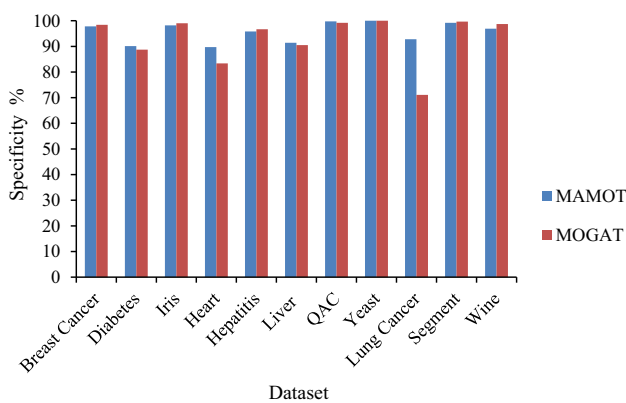


Fig. 9 Comparison of MAMOT and MOGAT with respect to the specificity results obtained on the testing set

data achieved 100.00% in both the training and testing set. MOGAT achieved successful results, which means 100.00% specificity in the QAC, segment and yeast datasets. In general, it can be clearly seen from the data in Table 7 that the specificity results for all of the datasets obtained high specificity rates.

Based on the evaluation viewpoint that was utilized in this study, it can be concluded that the MAMOT is more suitable to be employed as a classifier, whereas the proposed method shows the best performance to some extent. However, the optimal parameters are very important for ensuring the accuracy of the ANNs. Hence, the MAMOT has facilitated the searching process for the optimal parameters of the TBP network and is thus able to produce more accurate results. The results of this study also demonstrated that the use of the adaptive method in estimating the parameters of the TBP network was able to improve the accuracy of the proposed method in all of the datasets used.

4.4 Statistical test

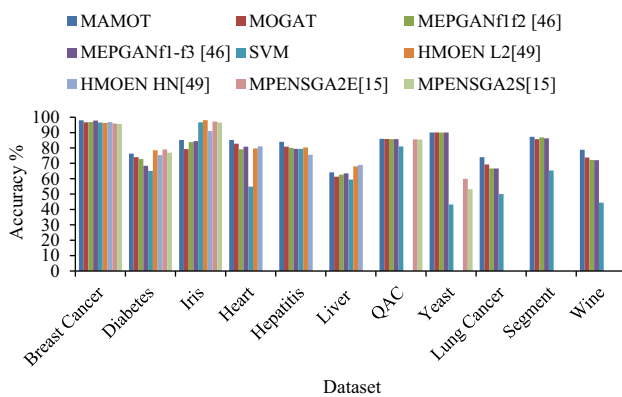
To compare two or more classifiers on multiple datasets in this study, we used statistical tests to determine whether the algorithms are significantly different or not. Several known statistical tests are examined, and their suitability was studied to determine the significance of the proposed methods based on the complexity of the TBP network and its accuracy. To test the difference between two classifiers' results over various datasets, a paired t test was used, which determines whether the average difference in their performance over the datasets is significantly different from zero. On the other hand, the Wilcoxon signed-ranks test was also used to detect significant differences between the behaviors of the algorithms' pair.

Before we used such statistical tests, we performed some statistical analysis so as to check the normality assumption. The used tests were Robust Jarque–Bera test (RJB), the Jarque–Bera (JB) test, the SJ test (SJ), the popular Shapiro–Wilk test (SW) and the Anderson–Darling test (AD); these tests are robust tests for normality with reference to the study [59] and Skewness Kurtosis test (SK). We used such robust tests for normality to investigate whether the different values for accuracy and complexity are normally distributed or not. Table 8 shows the statistical tests results and proved that the accuracy results of MAMOT and MOGAT follow the normal distribution assumption. Furthermore, the results proved that the complexity results of MAMOT are normally distributed, while the MOGAT method violates the normality assumption. Based on our results of the robust tests for normality, we used a paired sample *t* test to test difference of the accuracy for the MAMOT and MOGAT. On the other hand, we used Wilcoxon signed-ranks test for the difference of the complexity for MAMOT and MOGAT.

The MAMOT and MOGAT were tested using the *t* test and Wilcoxon signed-ranks test for testing the model accuracy difference and complexity, respectively. Let us first construct the null hypothesis to test the significance of

Table 10 Classification accuracies of the proposed method and some of the other methods in the literature

Dataset	Method/References								
	MAMOT	MOGAT	MEPGAN f_1f_2 [54]	MEPGAN f_1f_3 [54]	SVM [54]	HMOEN L2 [57]	HMOEN HN [57]	MPENSGA2E [18]	MPENSGA2S [18]
Breast cancer	97.94	96.69	96.78	97.80	96.49	96.26	96.82	95.87	95.60
Diabetes	76.28	73.98	72.78	68.35	65.10	78.48	75.36	78.99	76.96
Iris	85.13	79.26	83.78	84.44	96.67	98.00	91.03	97.18	96.50
Heart	85.16	82.78	79.07	80.79	54.88	79.69	81.06	–	–
Hepatitis	84.01	80.82	80.04	79.38	79.36	80.30	75.51	–	–
Liver	64.16	61.35	62.63	63.50	59.42	68.00	68.94	–	–
QAC	85.95	85.78	85.71	85.71	80.95	–	–	85.56	85.40
Yeast	90.03	90.01	90.00	90.01	43.26	–	–	59.91	53.21
Lung cancer	73.96	69.26	66.67	66.67	50.00	–	–	–	–
Segment	87.20	85.71	86.90	86.22	65.37	–	–	–	–
Wine	78.84	73.74	72.18	72.04	44.38	–	–	–	–

**Fig. 10** Performance comparisons of the proposed and existing methods on the testing set for all of the datasets

MAMOT in relation to MOGAT in accuracy. The null hypothesis is that there is no difference between the average accuracy of MAMOT versus MOGAT. The results of the applied paired t test are shown in Table 9. The p value of the t test is less than $\alpha = 0.05$ significance level, which implies the rejection of the null hypothesis. Therefore, there were significant differences; furthermore, MAMOT was significantly better than MOGAT. On the other hand, the p value resulted from the Wilcoxon signed rank of testing the differences of the algorithms complexity also was lower than α , which implied the rejection of the null hypothesis. Thus, there were significant differences. As we can see below, the results from Table 9 show that there were statistically significant differences between MAMOT and MOGAT.

4.5 Comparisons with other studies

The performance of the proposed methods can be compared with MOGAT and other memetic and multi-objective genetic algorithm-based ANN algorithms found in the literature (which used the same datasets) and some baseline methods, such as (HMOEN L2 and HMOEN HN [57], MPENSGA2E and MPENSGA2S [18], MEPGANf1f2 and MEPGANf1f3 [54] and also SVM [54]). Table 10 and Fig. 10 show a summary of the comparison results. Our proposed method, MAMOT, is the best of all of the methods reported in Table 10 on all of the datasets, except for diabetes, iris and liver, in which MPENSGA2E [18] is better than our algorithm in diabetes, while [57] there are two methods, HMOEN L2 and HMOEN HN, which are better than our algorithm for the iris and liver datasets, respectively. Additionally, on the breast cancer data, MAMOT achieved a better accuracy value than the other methods.

5 Conclusions

This paper introduces a new memetic adaptive multi-objective evolutionary algorithm that is based on the TBP network, MAMOT, for optimizing the TBP network parameters by using an adaptive non-dominated sorting genetic algorithm, ANSGA-II. The memetic process introduces the BP algorithm as a local optimizer hybrid with ANSGA-II, which was used to enhance all of the individuals in the population. On the other hand, the self-

adaptive SBX crossover was used to improve the performance of the ANSGA-II. The new memetic adaptive multi-objective evolutionary algorithm simultaneously optimizes the neural network parameters, specifically the connection weights, error rate and optimal structure in terms of the number of nodes in the hidden layer. This MAMOT-based algorithm not only helps to improve the classification accuracy, but also automatically designs and reduces the network structure during the classification phase of the neural classifier. To assess the performance of the MAMOT, experiments were conducted with 11 different dataset types for classification problems, 10 datasets obtained from UCI repository and another complex environment problem from qualitative analytical chemistry. The experimental results obtained show that the proposed method (MAMOT) was able to obtain a TBP network with better classification accuracy and a simpler network structure on the classification tasks compared with the other algorithms found in the literature. Based on an evaluation and statistical tests that were conducted, it can be concluded that the proposed MAMOT is suitable to be employed as a classifier for classification problems. As a future work, we plan to integrate a preferential local search with adaptive weights, as proposed in [58], to improve the performance of the algorithm and obtain better results. Our future work also will include the investigation of the proposed method and the effectiveness with larger datasets. In addition, we are planning to check the performance of the proposed method in other types of artificial neural networks.

Acknowledgements The authors would like to thank the anonymous reviewers and the editor for their useful advice and constructive comments.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- Salari N, Shohaimi S, Najafi F, Nallappan M, Karishnarajah I (2014) A novel hybrid classification model of genetic algorithms, modified k -nearest neighbor and developed backpropagation neural network. *PLoS One* 9:e112987
- Prieto A, Bellas F, Duro RJ, Lopez-Peña F (2007) Auto adjustable ANN-based classification system for optimal high dimensional data analysis. *Computational and ambient intelligence*. Springer, Berlin, pp 588–596
- Garro BA, Sossa H, Vázquez RA (2011) Evolving neural networks: a comparison between differential evolution and particle swarm optimization. Springer, Berlin, pp 447–454
- Qasem SN, Shamsuddin SM, Zain AM (2011) Multi-objective hybrid evolutionary algorithms for radial basis function neural network design. *Knowl Based Syst* 27:475–497
- Sagar G, Chalam SV, Singh MK (2011) Evolutionary algorithm for optimal connection weights in artificial neural networks. *Int J Eng (IJE)* 5:333
- Yu Q, Peng J (2011) Music category based on adaptive mutation particle swarm optimization BP neural network. In: Wu Y (ed) *Advances in computer, communication, control and automation*, vol 121. Lecture notes in electrical engineering. Berlin, Heidelberg, pp 657–663
- Abraham A, Nath B (2000) Optimal design of neural nets using hybrid algorithms. In: *PRICAI 2000 topics in artificial intelligence*, Springer, Berlin, pp 510–520
- Zhang R, Tao J (2017) Data-driven modeling using improved multi-objective optimization based neural network for coke furnace system. *IEEE Trans Ind Electron* 64:3147–3155
- Ahmad F, Isa NAM, Hussain Z, Osman MK, Sulaiman SN (2015) A GA-based feature selection and parameter optimization of an ANN in diagnosing breast cancer. *Pattern Anal Appl* 18:861–870
- Yao X (1999) Evolving artificial neural networks. *Proc IEEE* 87:1423–1447
- Hernández C, Schütze O, Sun J-Q (2017) Global multi-objective optimization by means of cell mapping techniques. In: *EVOLVE—a bridge between probability, set oriented numerics and evolutionary computation VII*, Springer, Cham, pp 25–56
- Zhou A, Qu B-Y, Li H, Zhao S-Z, Suganthan PN et al (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm Evolut Comput* 1:32–49
- Cruz-Ramírez M, Hervás-Martínez C, Jurado-Expósito M, López-Granados F (2012) A multi-objective neural network based method for cover crop identification from remote sensed data. *Expert Syst Appl* 39:10038–10048
- Ibrahim AO, Shamsuddin SM, Ahmad NB, Qasem SN (2013) Three-term backpropagation network based on elitist multiobjective genetic algorithm for medical diseases diagnosis classification. *Life Sci J* 10(4):1815–1822
- Ak R, Li Y, Vitelli V, Zio E, Droguett EL et al (2013) NSGA-II-trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment. *Exp Syst Appl* 40:1205–1212
- Fernández JC, Cruz-Ramírez M, Hervás-Martínez C (2018) Sensitivity versus accuracy in ensemble models of artificial neural networks from multi-objective evolutionary algorithms. *Neural Comput Appl* 30(1):289–305
- Abbass HA (2003) Speeding up backpropagation using multi-objective evolutionary algorithms. *Neural Comput* 15:2705–2726
- Fernandez Caballero JC, Martínez FJ, Hervás C, Gutiérrez PA (2010) Sensitivity versus accuracy in multiclass problems using memetic Pareto evolutionary neural networks. *IEEE Trans Neural Netw* 21:750–770
- Silva VV, Fleming PJ, Sugimoto J, Yokoyama R (2008) Multiobjective optimization using variable complexity modelling for control system design. *Appl Soft Comput* 8:392–401
- Gorzalczany MB, Rudziński F (2017) Interpretable and accurate medical data classification—a multi-objective genetic-fuzzy optimization approach. *Exp Syst Appl* 71:26–39
- Pettersson F, Chakraborti N, Saxén H (2007) A genetic algorithms based multi-objective neural net applied to noisy blast furnace data. *Appl Soft Comput* 7:387–397
- Jin Y, Sendhoff B, Körner E (2005) Evolutionary multi-objective optimization for simultaneous generation of signal-type and symbol-type representations. Springer, Berlin, pp 752–766
- Jin Y, Sendhoff B (2008) Pareto-based multiobjective machine learning: an overview and case studies. *IEEE Trans Syst Man Cybern Part C Appl Rev* 38:397–415
- García-Pedrajas N, Ortiz-Boyer D, Hervás-Martínez C (2004) Cooperative coevolution of generalized multi-layer perceptrons. *Neurocomputing* 56:257–283

25. Delgado M, Cuéllar MP, Pegalajar MC (2008) Multiobjective hybrid optimization and training of recurrent neural networks. *IEEE Trans Syst Man Cybern Part B Cybern* 38:381–403
26. Jin Y, Sendhoff B, Körner E (2006) Simultaneous generation of accurate and interpretable neural network classifiers. *Multi-objective machine learning*. Springer, Berlin, pp 291–312
27. Wiegand S, Igel C, Handmann U (2004) Evolutionary multi-objective optimisation of neural networks for face detection. *Int J Comput Intell Appl* 4:237–253
28. Abbass HA (2002) An evolutionary artificial neural networks approach for breast cancer diagnosis. *Artif Intell Med* 25:265–281
29. Almeida LM, Ludermir TB (2010) A multi-objective memetic and hybrid methodology for optimizing the parameters and performance of artificial neural networks. *Neurocomputing* 73:1438–1450
30. Cruz-Ramírez M, Sánchez-Monedero J, Fernández-Navarro F, Fernández J, Hervás-Martínez C (2010) Memetic pareto differential evolutionary artificial neural networks to determine growth multi-classes in predictive microbiology. *Evol Intel* 3:187–199
31. Qasem SN, Shamsuddin SM (2011) Memetic elitist pareto differential evolution algorithm based radial basis function networks for classification problems. *Appl Soft Comput* 11:5565–5581
32. Cruz-Ramírez M, Hervás-Martínez C, Fernández JC, Briceño J, de la Mata M (2012) Multi-objective evolutionary algorithm for donor-recipient decision system in liver transplants. *Eur J Oper Res* 222(2):317–327
33. Abbass HA (2002) The self-adaptive pareto differential evolution algorithm. In: *Proceedings of the 2002 congress on evolutionary computation*. pp 831–836
34. Huang VL, Zhao SZ, Mallipeddi R, Suganthan PN (2009) Multi-objective optimization using self-adaptive differential evolution algorithm. In: *IEEE congress on evolutionary computation* pp 190–194
35. Zeng F, Low MYH, Decraene J, Zhou S, Cai W (2010) Self-adaptive mechanism for multi-objective evolutionary algorithms. In *Proceedings of the international multiconference of engineers and computer scientists*, pp 7–12
36. Shim VA, Tan KC, Tang H (2014) Adaptive memetic computing for evolutionary multiobjective optimization. *IEEE Trans Cybern* 45(4):610–621
37. Caruana R, Niculescu-Mizil A (2004) Data mining in metric space: an empirical analysis of supervised learning performance criteria. In: *ACM*. pp 69–78
38. Fawcett T (2006) An introduction to ROC analysis. *Pattern Recognit Lett* 27:861–874
39. Thangaraj R, Pant M, Abraham A, Badr Y (2009) Hybrid evolutionary algorithm for solving global optimization problems. *Hybrid artificial intelligence systems*. Springer, Berlin, pp 310–318
40. Krasnogor N, Smith J (2005) A tutorial for competent memetic algorithms: model, taxonomy, and design issues. *IEEE Trans Evolut Comput* 9:474–488
41. Fernández J, Hervás C, Martínez F, Gutiérrez P, Cruz M (2009) Memetic Pareto differential evolution for designing artificial neural networks in multiclassification problems using cross-entropy versus sensitivity. In: *Hybrid artificial intelligence systems*. pp 433–441
42. Schaffer JD (1985) Multiple objective optimization with vector evaluated genetic algorithms. *L. Erlbaum Associates Inc, Mahwah*, pp 93–100
43. Srinivas N, Deb K (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol Comput* 2:221–248
44. Fonseca CM, Fleming PJ (1998) Multiobjective optimization and multiple constraint handling with evolutionary algorithms. II. Application example. *IEEE Trans Syst Man Cybern Part A Syst Hum* 28:38–47
45. Zweiri Y, Whidborne J, Seneviratne L (2003) A three-term backpropagation algorithm. *Neurocomputing* 50:305–318
46. Zweiri YH (2007) Optimization of a three-term backpropagation algorithm used for neural network learning. *Int J Comput Intell* 3:322–327
47. Deb K (1999) An introduction to genetic algorithms. *Indian Acad Sci* 24:293–315
48. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evolut Comput* 6:182–197
49. Ak R, Li Y, Vitelli V, Zio E, Drogue EL et al (2012) NSGA-II-trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment. *Exp Syst Appl* 40(4):1205–1212
50. Ramesh S, Kannan S, Baskar S (2012) Application of modified NSGA-II algorithm to multi-objective reactive power planning. *Appl Soft Comput* 12(2):741–753
51. Deb K, Sindhya K, Okabe T. Self-adaptive simulated binary crossover for real-parameter optimization; 2007. In: *Proceedings of the genetic and evolutionary computation conference (GECCO-2007)*, UCL London. pp 1187–1194
52. Zhang C, Ren M, Zhang B (2013) A self-adaptive multi-objective genetic algorithm for the QoS based routing and wavelength allocation problem in WDM network. *Opt Int J Light Electron Opt* 124(20):4571–4575
53. Lara A, Sanchez G, Coello Coello CA, Schutze O (2010) HCS: a new local search strategy for memetic multiobjective evolutionary algorithms. *IEEE Trans Evolut Comput* 14:112–132
54. Qasem SN, Shamsuddin SM, Hashi SZM, Darus M, Al-Shammari E (2013) Memetic multiobjective particle swarm optimization-based radial basis function network for classification problems. *Inf Sci* 239:165–190
55. Asuncion A, Newman DJ (2007) UCI machine learning repository. [http://www.icsuciedu/\\$\sim\\$mllearn/{MLR}epositoryhtml](http://www.icsuciedu/\simmllearn/{MLR}epositoryhtml). Accessed 18 July 2017
56. Hervás C, Silva M, Gutiérrez PA, Serrano A (2008) Multilogistic regression by evolutionary neural network as a classification tool to discriminate highly overlapping signals: qualitative investigation of volatile organic compounds in polluted waters by using headspace-mass spectrometric analysis. *Chemom Intell Lab Syst* 92(2):179–185
57. Goh C-K, Teoh E-J, Tan KC (2008) Hybrid multiobjective evolutionary design for artificial neural networks. *IEEE Trans Neural Netw* 19:1531–1548
58. Bhuvana J, Aravindan C (2016) Memetic algorithm with Preferential Local Search using adaptive weights for multi-objective optimization problems. *Soft Comput* 20(4):1365–1388
59. Stehlík M, Štěpánek L, Thulin M (2014) On robust testing for normality in chemometrics. *Chemom Intell Lab Syst* 130:98–108

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval, sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com