

Neural Networks Modeling Based on Recent Global Optimization Techniques



Anwar Jarndal , Sadeque Hamdan , Sanaa Muhaureq ,
and Maamar Bettayeb 

1 Introduction

Global optimization techniques can be classified into evolutionary and swarm intelligence algorithms. The evolutionary algorithms such as genetic algorithm (GA) are stochastic optimizations inspired by nature evolution and heredity mechanisms [1, 2]. Swarm intelligence algorithms such as PSO, ant colony optimization and artificial bee colony are inspired by collective intelligence, which appears through the interaction of animals and social insects [3–6]. These optimization techniques are used in wide range of engineering problems. Recently, some works started to combine global optimization techniques with artificial neural networks (ANNs). The key element of the ANNs is the neurons (processing units), which are connected together into a network and arranged in layers to perform the required computing. Each neuron within the network takes one or more inputs and produces an output. At each neuron, all inputs are scaled by weights to modify their strengths and processed by proper base function to produce an output, which will be considered as an input for the next neuron (in the next layer). During training, the ANN processes the inputs and compares its resulting outputs with the desired outputs to calculate the errors, which are then propagated back through the system to adjust the weights for best fitting. This typically used back-propagation technique works well for simple model of lower number of weights (variables) [7]. However, for larger scale ANN model of

A. Jarndal (✉) · S. Muhaureq · M. Bettayeb
Electrical Engineering Department, University of Sharjah, Sharjah, United Arab Emirates
e-mail: ajarndal@sharjah.ac.ae

S. Hamdan
Sustainable Engineering Asset Management (SEAM) Research Group, University of Sharjah,
Sharjah, United Arab Emirates

M. Bettayeb
Center of Excellence in Intelligent Engineering Systems (CEIES), King Abdulaziz University,
Jeddah, Saudi Arabia

larger number of weights, the final solution will have higher dependency on the initial guess and may be trapped in a local minimum. In this case, the user needs more effort to re-initiate the training process many times to get the best fitting, which is not practical for some application such as automatic controlling. For that reason, training the ANN using global optimization methods could provide an efficient alternative and avoids drawbacks of local minimization methods. Various optimization techniques in the literature have been used to optimize ANN weight. For instance Whitley et al. [8] discussed using GA for optimizing neural network weights and architectures in the form of connectivity patterns. Yamazaki et al. [9] used simulated annealing to optimize neural network architectures and weights which resulted in good generalization performance and low complexity. Karaboga et al. [10] used artificial bee colony optimization to optimize neural network weights and overcome local optimal solutions. Mirjalili et al. [11] implemented a hybrid technique of PSO and gravitational search algorithm. On the other hand, some research has been conducted to compare the back-propagation for training neural networks with other global optimization techniques. For example, Gudise and Venayagamoorthy [12] showed that ANN weight convergence is faster using PSO, in learning feed-forward ANN, with respect to back-propagation. Mohaghegi et al. [13] compared PSO with back-propagation in optimizing neural network weights. The comparison revealed that PSO is efficient, requires less effort and is more efficient in finding optimal weights compared with back-propagation.

To the best knowledge of the authors, the research on application of global optimization for ANN is still limited and more extensive researches are needed. Also, many recently developed swarm intelligence optimization techniques have not been used, except for grey wolf optimization (GWO). Mirjalili [14] used GWO for training multi-layer perception and found that it outperforms some other global optimization techniques. As a result, this paper presents combined techniques of ANN modeling with different recent global optimizations and investigates their performance (in terms of the efficiency and effectiveness) in solving practical modeling problems.

The contribution of this paper can be summarized as follows:

1. This work investigates the performance of recent global optimization techniques such as dragonfly algorithm (DA), grasshopper optimization algorithm (GOA), whale optimization algorithm (WOA) and GWO in training the ANNs.
2. The developed techniques are demonstrated by modeling the GaN transistor and compared their performance with the measured data.
3. This paper compares the performance of the widely used optimization technique GA with the recently developed swarm optimization techniques in terms of speed and accuracy.

Figure 1 illustrates the proposed methodology used in this work. In the first phase, measured data are used as inputs for the ANN model, which consists of two hidden layers (see details in Sect. 2). In the second phase, the ANN model weights are optimized using five different optimization techniques. In the last phase, the output of each combined ANN model is compared with the actual data. Finally, the results of all considered optimization techniques are compared and discussed. The organization

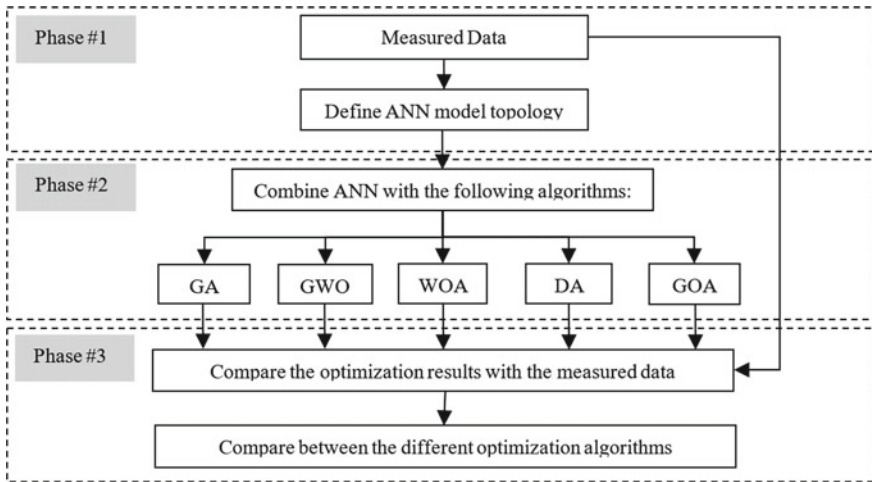


Fig. 1 Methodological framework

of this paper can be summarized as follows: Sect. 2 describes the ANN model used in this work. Section 3 describes the ANN-GA model and Sect. 4 describes ANN-GWO model. Sections 5, 6, 7 provide details on ANN-WOA, ANN-DA and ANN-GOA models. Section 8 describes the case study of modeling the GaN transistor. Section 9 provides analyses, results, discussion and insights. The last section provides a conclusion.

2 ANN Model and Experiment Design

ANN is used to capture the relationship between inputs and outputs. In Fig. 2, we show a simple ANN model consisting of two hidden layers, two inputs (X_1, X_2) and one output (Y). The output can be calculated using the following equation:

$$Y = \sum_{i=1}^3 W_i \tan h \left(W_{i1} + \sum_{j=2}^4 W_{ij} \tanh(W_{1j}X_1 + W_{2j}X_2 + W_{3j}) \right). \quad (1)$$

In Eq. (1) the input weights are W_{1j} , W_{2j} and W_{3j} , the intermediate weights are expressed by W_{ij} and the output weight is expressed by W_i . The activation function used in this model is $\tanh(\cdot)$. In this paper, we optimize the weights of the ANN model using five global optimization techniques. The five global optimization techniques are to be tested over five numbers of solutions (50, 100, 200, 300 and 500) and over six different maximum numbers of iterations (50, 100, 200, 400, 600 and 800). Each configuration (number of solution and number of iteration) is to be run 30 times and

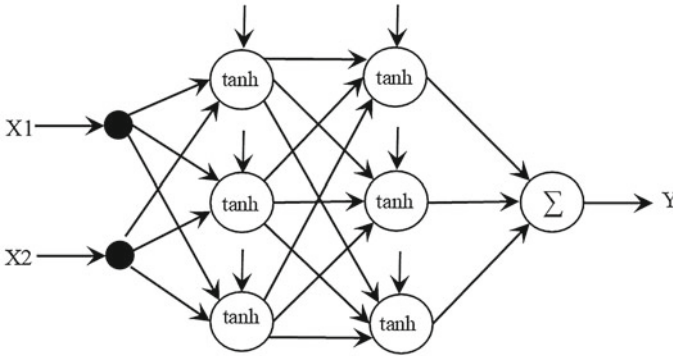


Fig. 2 A simple illustrative ANN model

the average performance is measured. Moreover, all the techniques will be fed with the same initial population.

3 Training ANN Using Genetic Algorithm Optimization

GA is inspired by the evolutionary biology, and it mimics the natural evolution processes such as reproduction, mutation, selection and crossover [1]. GA is classified as an evolutionary technique and was found to provide optimal or near-optimal solutions with good quality. Moreover, GA has been used to find optimal values for the weight of the ANN model in Fig. 2. The optimization approach starts by generating a set of random population (24 weights) within the range from -1 to 1. Next, the fitness of all solutions is evaluated, which is the total error among the measured and simulated values in Eq. (2). The objective is to minimize the difference between the measured data and the simulated results obtained from Eq. (1) in ANN-GA model.

$$\text{Error} = \frac{1}{M} \sum_{i=1}^M (Y_{\text{measured}} - Y_{\text{simulated}})^2 \quad (2)$$

where M is the total number of the measured data. Y_{measured} and $Y_{\text{simulated}}$ represent the measured and the simulated data, respectively. After that, the parents are selected based on their fitness. Next, the parents are recombined using the crossover operator. Then, the offspring is mutated randomly with low probably. It is worth to note that the crossover and mutation ensure the exploration in GA algorithm and avoid getting stuck in local optima. Furthermore, the new offspring fitness is evaluated and the next generation population is selected. The solution with best fitness (least error) is selected and returned as the optimal weights for the ANN-GA model.

4 Training ANN Using Grey Wolf Optimization

GWO algorithm is inspired by grey wolves, and it mimics their strict leadership hierarchy and hunting mechanisms [15]. The grey wolves' hierarchy contains of four types: alpha, beta, delta and omega. The alpha wolf is the leader and responsible for the decision making in the pack. In the following types (beta, delta and omega) the dominance decreases in each level. The grey wolves hunting behavior for preys and selecting the best prey (optimal solution) among these multiple preys can be described by: tracking, encircling and attack the prey [15]. In order to mathematically model the GWO algorithm the alpha wolf (α) is considered to be the optimal solution, the beta (β) and delta (δ) wolves are considered second and third best solutions [15]. The remaining candidate solutions are assumed to be the omega wolves (ω). First the mathematical model inserts the measured ANN weights. Next the grey wolves' population positions are initialized (24 weights) and the corresponding coefficients are calculated using the equations in [15]. The fitness of grey wolves is calculated, and alpha, beta and delta wolves are selected. Subsequently, GWO starts optimizing positions of packs as described in [15]. It is worth mentioning that the adaptive values of GWO coefficients allow a good balance between explorations, exploitation and avoiding stagnation in local solutions [15]. Note that in the mathematical model the alpha, beta and delta positions are saved and all search agents update their positions accordingly [15]. It can be observed that the search agents' new position will be directed by the alpha, beta and delta wolves with some randomness as a result of the coefficients. In other words, the alpha, beta and delta wolves estimate the preys' position and the search agents update their positions around the prey. Furthermore, after obtaining the packs' new positions, the fitness is calculated and the new positions for alpha, beta and delta are updated. Finally, alpha is returned as the optimal solution for the ANN model.

5 Training ANN Using Humpback Whales Optimization

WOA is inspired by the social behavior of humpback whales. The WOA mathematical model mimics the bubble-net hunting strategy to find optimal solutions for optimization problems. The bubble-net hunting method is basically when the humpback whale circulates around the prey (optimal solution) going up in a helix-shaped path while releasing bubbles along the path. The humpback whale will get closer and closer to the prey while circulating and then reaches the optimal solution [16]. The mathematical model for encircling the prey for WOA is similar to the GWO algorithm. However, there are differences because the WOA depends on the bubble-net feeding method [16]. Similar to other models, initially, the measured data are inserted and then population is initialized. To mathematically model the bubble-net attacking method of the humpback whales, two approaches are used: Shrinking encircling mechanism and spiral updating position. The humpback whales use the shrinking encircling and

spiral updating position approaches simultaneously. To model this behavior, there is a probability of 50% the whales will choose either of two approaches to update the position and reach optimal solution.

6 Training ANN Using Dragonfly Optimization (ANN-DA)

Another swarm intelligence optimization technique was developed by Mirjalili [3]. The DA imitates the social behavior of dragonflies in avoiding enemies and searching for food sources. Dragonflies have mainly two types of movement, static and dynamic movements, which correspond to exploitation and exploration needed for global optimization. Static movement is used for hunting and represents the exploitation phase of the optimization. On the other hand, dynamic movement is when dragonflies move in bigger swarms to migrate in one direction for long distances, this represents the exploration phase. This optimization technique is similar to the conventional particles swarm optimization, however; instead of being based on the velocity of PSO, DA updates the position of the individuals by using the step vector. The DA-ANN model starts by inserting the measured data. Next, the dragonflies' positions and step vectors are initialized. Then the fitness values for all dragonflies' positions are evaluated using Eq. (2). It is worth mentioning that the food source represents the best (optimal) value and enemy represents the worst value, and it is updated in each iteration. After that, DA updates social factors such as separation, alignment, cohesion, attraction to food and distraction from the enemy as described in [3]. In addition to the five factors that control the updated position of each individual, the step vector takes into account the inertia weight factor ω of the current position. Finally, check the boundaries and modify the new position if the boundaries are violated. Continue until stopping criterion is met and return best optimal solution (food source) [3].

7 Training ANN Using Grasshopper Optimization Algorithm

GOA is inspired by the behavior of grasshopper swarms in nature. Saremi et al. [4] developed a global optimization algorithm by modeling grasshoppers in searching for food and mimicking their exploration and exploitation movements. GOA, as all the other swarm techniques, is based on a set of individuals with random position, where the position of each agent is updated until it converges to an optimal solution. GOA updates the position of each search agents taking into account social interaction, gravity force and wind advection. However, this mathematical model cannot be used directly to solve optimization problems. The main reason is because grasshoppers quickly reach their comfort zone and don't converge to optimal solution. Therefore,

a modified version of the position equation is proposed in [5]. It is assumed there is no gravity force and that the wind direction is always toward target. First, measured data for ANN-GOA is inserted. Next, the grasshopper population is initialized (24 weights) and parameters. Then Eq. (2) is used to evaluate the fitness of each search agent and best search agent is selected. Subsequently, the GOA starts its iteration with updating the coefficient that shrinks the comfort zone, attraction zone and repulsion zone. After that each search agent normalizes the distance between grasshoppers [4]. Then a check is performed on all search agents if the new position goes beyond boundaries. After that all solutions are evaluated again, and the best optimal solution is updated until the maximum iteration is reached. It is worth mentioning that GOA updates the position of search agents based on the current position, global-best position and position of all other grasshoppers in population. This is different from the well-known PSO where the position is updated using current position, self-best position and global-best position.

8 ANN Modeling of GaN Transistor (Case Study)

ANN can be used to solve a wide range of modeling problems. One of these problems is transistor modeling, which is needed for circuit design purposes, especially, for the new technology such as Gallium Nitride High Electron Mobility Transistor (GaN HEMT). The current-voltage characteristics of this power transistor affected mainly by power dissipation induced self-heating which results in current collapsing in higher power dissipation operating conditions (Fig. 3a) [17]. This regenerative process, in addition to the strong nonlinear behavior of the current with respect to

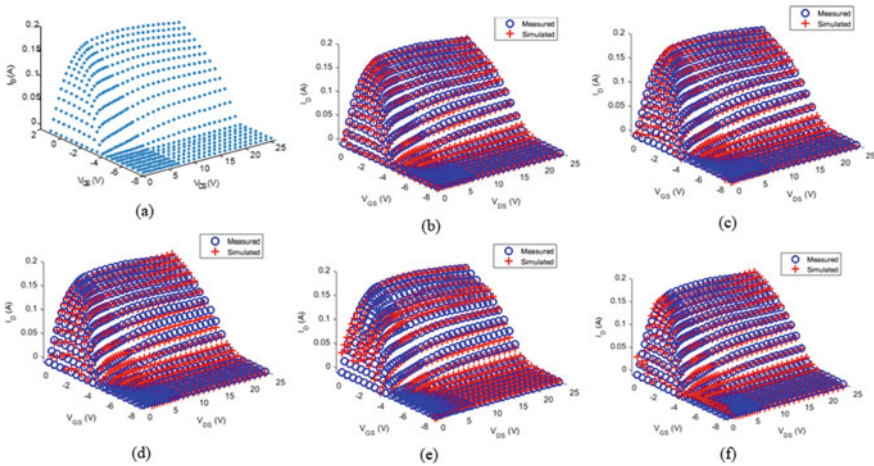


Fig. 3 a Measured current–voltage characteristics of GaN HEMT. b ANN-GA best fitting. c ANN-GWO best fitting. d ANN-WOA best fitting. e ANN-DA best fitting. f ANN-GOA best fitting

applied drain and gate voltages (V_{GS} and V_{DS}), put more effort on the implemented model. Analytical modeling is one of the commonly used techniques to simulate this device. However, this approach needs higher effort to adopt an efficient closed formula and also to find the corresponding fitting parameters [18]. ANN modeling can provide an optimal solution with lower effort and higher accuracy [19]. Of course, to obtain an accurate simulation for the nonlinear behavior of the current, a higher ANN order model with a higher number of weights is required. In this case, the modeling process will be reduced to an optimization problem and global optimization techniques should be used. The model, illustrated in Fig. 2, has been used efficiently to simulate the current–voltage characteristics of the GaN HEMT. The model inputs X_1 and X_2 represent the gate and drain voltages, V_{GS} and V_{DS} , respectively. The output Y represents the drain current I_D . In Fig. 3(b–f) we can observe the obtained results for the best fitting for ANN-GA, ANN-GWO, ANN-WOA, ANN-DA and ANN-GOA. It can be noticed the ANN-GA and ANN-GWO give the best fitting performance.

9 Results and Discussion

GaN transistor modeling data, shown in Fig. 3, was used in the ANN model combined with five global optimizations, one technique at a time. The analysis was performed by changing the number of iterations and number of individuals. Since all the optimization techniques mentioned in this paper share the same starting process, which consists in imitating a set of individuals that represent the 24 weights of the ANNs model, the same population set was used to test all the techniques. Thus, before starting any optimization technique, the population was generated and stored then used as initial guess for model optimization. This was done in order to guarantee equal conditions for all optimization techniques during the comparison. All optimization techniques were used with their default parameters. Each ANNs optimization model was tested with five different numbers of solutions (50, 100, 200, 300 and 500). While changing the number of iterations (50, 100, 200, 400, 600 and 800). The performance of each model was evaluated with respect to two criteria: efficiency in terms of time and effectiveness in terms of solution's quality (error value). Figure 4 shows the optimization value (error) of different ANN-optimization models. It can be seen that ANN-GWO outperforms all the other optimization techniques under small number of iterations and solutions (see Fig. 4, solutions = 50, iteration 50 and 100). It is noted that ANN-GA consistently had a stable robust performance that didn't fluctuate while increasing the number of solutions or iterations unlike the ANN-DA and ANN-GOA.

It was noticed that the best fitting using ANN-GA was obtained for 500 solutions and 800 iterations with an error equal to 0.02022 (see Fig. 4). The best result for ANN-GWO was obtained for 300 solutions and 800 iterations with an error equal to 0.026 (see Fig. 4). Moreover, for ANN-WOA, ANN-DA and ANN-GOA, the best fitting was obtained for 100, 100 and 100 solutions with 800, 400 and 400 iterations, respectively. The obtained errors, 0.04992, 0.169 and 0.03375, respectively. To sum

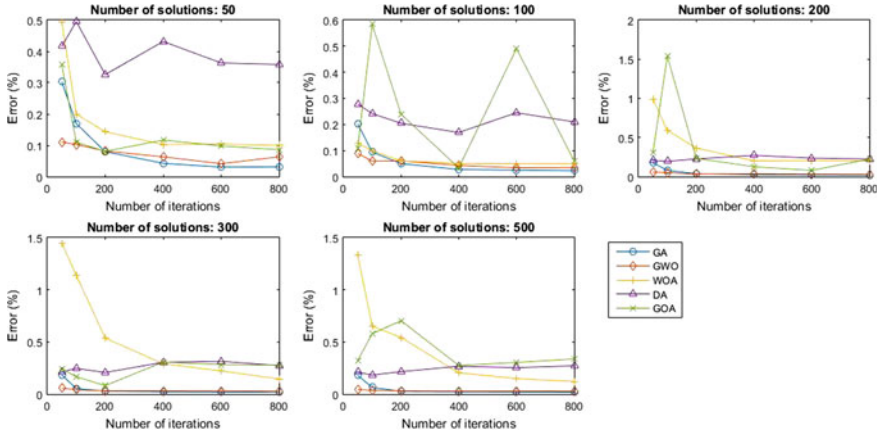


Fig. 4 Average error values of different ANN-optimization models as the number of iterations increases for different number of solutions

up, ANN-GA resulted to be the most accurate fitting; however, ANN-GWO gave a very competitive result under reasonable number of iterations, solution and time. Both ANN-GA, ANN-GWO and ANN-WOA are efficient and fast enough with a minimum computation time of 3.42 s for 50 solutions and 50 iterations. Furthermore, 471.27 s for 500 solutions and 800 iterations. On the other hand, ANN-DA and ANN-GOA have the highest computation time range (5.77–3314 s for ANN-DA and 20.38–27070 s for ANN-GOA) (Fig. 5). It is worth mentioning that all the ANNs optimization models have been run on a computer equipped with an Intel(R) Core (TM) i5-4590, CPU @ 3.3 GHz, 8.00 GB RAM and Windows 7 64-bit operating system.

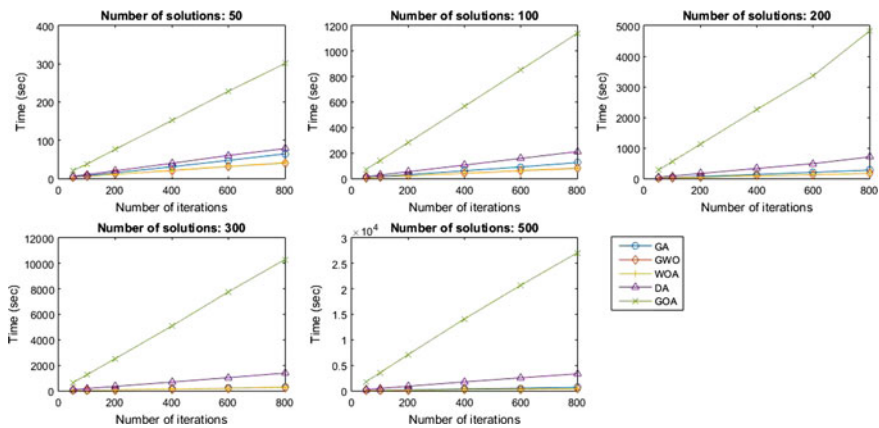


Fig. 5 Average CPU time of different ANN-optimization models as the number of iterations increases for different number of solutions

10 Conclusion

In this paper, five different global optimization techniques were used to optimize the weights of an ANN model. The performance and computational time of these techniques were compared. It has been found that ANN-GA and ANN-GWO show competitive performance in most of the considered cases. Furthermore, ANN-GA shows robustness and improved performance with increasing number of iterations and solutions. The ANN-GWO obtained competitive results with high speed of convergence. In general, ANN-GA provided the most accurate fitting (lowest error value) compared to others, but the convergence speed was less than that of ANN-GWO. On the other hand, ANN-DA and ANN-GOA are considered extremely slow with respect to the other techniques and didn't achieve best fitting values. As a further step, it is suggested to study the enhanced version of the developed global optimization algorithms and to improve the algorithms to account for parallel computing in order to enhance the speed of the algorithms. In addition, studying the effect of the parameters of each algorithm in feeding neural networks might be of interest as well.

Acknowledgements The authors gratefully acknowledge the support of University of Sharjah (Sharjah, United Arab Emirates).

References

1. W. Zhong, J. Liu, M. Xue, L. Jiao, A multiagent genetic algorithm for global numerical optimization, *IEEE Trans. Syst. Man. Cybern. B. Cybern.* **34**(2), 1128–1141 (2004)
2. S. Hamdan, A. Jarndal, A two stage green supplier selection and order allocation using AHP and multi-objective genetic algorithm optimization, in *2017 7th International Conference on Modeling, Simulation, and Applied Optimization, ICMSAO* (2017). <https://doi.org/10.1109/icmsao.2017.7934843>
3. S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **27**(4), 1053–1073 (2016). <https://doi.org/10.1007/s00521-015-1920-1>
4. S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application. *Adv. Eng. Softw.* **105**, 30–47 (2017). <https://doi.org/10.1016/j.advengsoft.2017.01.004>
5. N. Nawayseh, A. Jarndal, S. Hamdan, Optimizing the parameters of a biodynamic responses to vibration model using particle swarm and genetic algorithms, in *2017 7th International Conference on Modeling, Simulation, and Applied Optimization, ICMSAO* (2017). <https://doi.org/10.1109/icmsao.2017.7934851>
6. A. Jarndal, S. Hamdan, Forecasting of peak electricity demand using ANNGA and ANN-PSO approaches, in *2017 7th International Conference on Modeling, Simulation, and Applied Optimization, ICMSAO* (2017). <https://doi.org/10.1109/icmsao.2017.7934842>
7. J F. Kolen, J.B. Pollack, Back propagation is sensitive to initial conditions, in *Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems*, pp. 860–867 (1990)
8. D. Whitley, T. Starkweather, C. Bogart, Genetic algorithms and neural networks: optimizing connections and connectivity. *Parallel Comput.* **14**(3), 347–361 (1990). [https://doi.org/10.1016/0167-8191\(90\)90086-O](https://doi.org/10.1016/0167-8191(90)90086-O)

9. A. Yamazaki, M.C.P. de Souto, T.B. Ludermir, Optimization of neural network weights and architectures for odor recognition using simulated annealing, in *Proc. 2002 Int. Jt. Conf. Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, vol. 1, no. 3, pp. 547–552, (2002). <https://doi.org/10.1109/ijcnn.2002.1005531>
10. D. Karaboga, B. Akay, C. Ozturk, Artificial Bee Colony (ABC) optimization algorithm for training feed-forward neural networks, in *Proceedings of the 4th International Conference on Modeling Decisions for Artificial Intelligence*, pp. 318–329 (2007). https://doi.org/10.1007/978-3-540-73729-2_30
11. S. Mirjalili, S.Z. Mohd Hashim, H. Moradian Sardroudi, Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Appl. Math. Comput.* **218**(22), 11125–11137 (2012). <https://doi.org/10.1016/j.amc.2012.04.069>
12. V.G. Gudise, G.K. Venayagamoorthy, Comparison of particle swarm optimization and back-propagation as training algorithms for neural networks, in *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*, pp. 110–117 (2003)
13. S. Mohaghegi, Y. Del-Valle, G.K. Venayagamoorthy, A comparison of PSO and backpropagation for training RBF neural networks for identification of a power system with STATCOM, in *Swarm Intelligence Symposium, SIS 2005*, pp. 381–384, (2005). <https://doi.org/10.1109/SIS.2005.1501646>
14. S. Mirjalili, How effective is the Grey Wolf optimizer in training multi-layer perceptrons. *Appl. Intell.* **43**(1), 150–161 (2015). <https://doi.org/10.1007/s10489-014-0645-7>
15. S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014). <https://doi.org/10.1016/j.advengsoft.2013.12.007>
16. S. Mirjalili, A. Lewis, The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67 (2016). <https://doi.org/10.1016/j.advengsoft.2016.01.008>
17. I. Saidi, Y. Cordier, M. Chmielowska, H. Mejri, H. Maaref, Thermal effects in AlGaIn/GaN/Si high electron mobility transistors. *Solid-State Electron.* **61**(1), 1–6 (2011). <https://doi.org/10.1016/j.sse.2011.02.008>
18. A. Jarndal, F.M. Ghannouchi, Improved modeling of GaN HEMTs for predicting thermal and trapping-induced-kink effects. *Solid State Electron.* **123**, 19–25 (2016). <https://doi.org/10.1016/j.sse.2016.05.015>
19. A. Jarndal, Genetic algorithm-based neural-network modeling approach applied to AlGaIn/GaN devices. *Int. J. RF Microw. Comput. Eng.* **23**(2), 149–156 (2013). <https://doi.org/10.1002/mmce.20660>