A Model-Free Approach for Online Optimization of Nonlinear Systems

Ameer Hamza Khan, Shuai Li, Bin Xu, and Xinwei Cao

***Abstract*—This paper proposes a strategy to search optimal control parameters of a complex nonlinear system using a metaheuristic optimization algorithm in a computationally efficient manner. The proposed algorithm, called BAS-swarm (Beetle Antennae Search-swarm), is a gradient-free optimizer based on the BAS algorithm, inspired by mimicking the food foraging behavior of beetles. BAS-swarm takes advantage of the fact that the antennae of insects are not single sensory organs. The antennae contain an array of tiny fiber. Antennae fiber enables the insects to have a microscopic insight into the environment when moving toward the source of food smell. BAS-swarm uses this insight to improve the performance of BAS by approximating the gradient direction at each iteration with the help of a swarm of antenna fiber. Since the proposed algorithm approximates gradient by mimicking the behavior of beetle antenna fiber located at random locations, it does not require the numerical computation of the actual gradient, making it very efficient for optimization of nonlinear non-convex systems. We verified the accuracy and efficiency of the proposed algorithm by training single-layer neural networks with nonlinear activation function and compared its performance with Particle Swarm Optimizer (PSO), a well-studied extremum seeking algorithm, and the original BAS algorithm. The experiment shows that the proposed algorithm provides several-fold improvement and faster convergence as compared to other metaheuristic algorithms.**

***Index Terms*—Metaheuristic Optimization, Nature-inspired Algorithm, Swarm Intelligence**

I. Introduction

Most of the real world systems are nonlinear and require optimization of non-convex multimodal function for their operation [1], [2], [3], [4]. With the rise of Machine Learning (ML), the need to develop a numerically efficient optimization method with fast convergence have gained huge attention from researchers [5], [6]. Most common optimum searching algorithm use gradients of the objective function to search for the optimum value [7], [8], [9], [10]. For complicated functions, the calculation of gradient is numerically expensive [11], [12]. Furthermore, the calculation of gradient requires the objective function to be continuous and differentiable [13]. Such conditions does not hold true for some practical systems [14], [15], [16].

Natural phenomenon act as a great source of inspiration for the development of metaheuristic algorithms, e.g. biological evolution, animal lifestyle and chemical reactions. Several nature-inspired metaheuristic algorithms have been proposed in literature [17], [18]. The process of biological evolution and natural selection of living organisms have given rise to a complete class of Evolutionary Algorithms (EAs) [19] including the Genetic algorithm (GA) [20], [21], [22], [23]. Kennedy et al. [24] proposed Particle Swarm Optimizer (PSO), which is inspired by the swarming behavior of insects and birds. Fig. 1 shows how metaheuristic algorithm can be used for optimizing performance of a nonlinear system. Metaheuristic algorithms have found several real-world applications in recent years; they are being applied for robotics [25], [26], manipulator dynamics control [27], [5], and optimizing the dynamical performance of soft robotic systems [28].

Among the proposed metaheuristic algorithms BAS [29], inspired by food forging behavior of beetles, is of particular interest because beetles do not usually work in a swarm to search for food. An individual beetle is fully capable of searching for food by just using a pair of antennae and a strong sense of smell. Original BAS algorithm is based on the fact that the smell differential at two antennae will help the beetle in choosing the correct direction, i.e. direction of maximum smell. However, the original BAS took a rather simplistic approach in term of the sensory capability of the beetle antennae. It assumes that each antenna act as a single smell sensor. We improve the performance of BAS by taking advantage of a well known biological fact; antennae of insects are not single sensory organs [30]. Antennae have several tiny fiber which act as separate smell sensor and give fine grain sensory capability to the insects. By feeling the difference in intensity of smell at different antennae fiber, the beetle can make a better choice for the search direction as compared to bidirectional search outlined by original BAS.

The remaining paper is orgainized as follow; Section II lays down the details of BAS-swarm, Section III describe the experimental methodology, Section IV present the experimental results, and Section V concludes the paper.

II. BAS-swarm Algorithm

The biological reasoning behind the BAS-swarm algorithm has already been described in the Introduction. Now we will describe the technical details of the algorithms.

A. Beetle behavior Modeling

We will first model the simple food foraging behavior of the beetle as outlined in the original BAS ([31]). Consider the following unconstrained optimization problem:

$$\max_{\mathbf{x}} \ f(\mathbf{x}), \tag{1}$$

A.H. Khan is with Department of Computing, Hong Kong Polytechnic University, Hong Kong (email: ameer.h.khan@connect.polyu.hk).

S. Li is with Swansea University, Swansea, UK (email: shuaili@ieee.org).

B. Xu is with Northwestern Polytechnical University, Xi'an 710072, China (email: smileface.binxu@gmail.com).

X. Cao is with School of Management, Shanghai University, Shanghai 201900, China (email: xinweicao@shu.edu.cn).

S. Li is the corresponding authors.

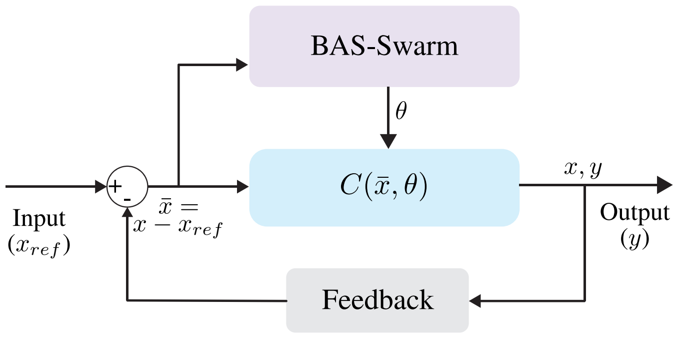


Fig. 1: Schematic diagram showing the role of BAS-swarm to optimize the parameters of a nonlinear control system.

where $\mathbf{x} \in \mathbb{R}^n$ and $f(\mathbf{x})$ is the n-dimensional objective function. For the beetle analogy, this essentially translates to a beetle searching food in an n-dimensional, where $f(x)$ describe the smell distribution in the space. The objective of the beetle is to search the location of food source, i.e. maximum smell. Suppose at time instant t the beetle is located at location $\mathbf{x}_t$. In order to get an estimate for the direction of the food source, the beetle uses its two antennae to measure smell intensity at its right and left antennae. The smell intensity in these two opposite directions can be modeled using a normally distributed normalized random vector $\overrightarrow{\mathbf{b}} \in \mathbb{R}^n$:

$$\mathbf{x}^l = \mathbf{x} + d_t \overrightarrow{\mathbf{b}},$$
$$\mathbf{x}^r = \mathbf{x} - d_t \overrightarrow{\mathbf{b}}, \tag{2}$$

where d^t is the length of the beetle antenna at time instant t, $\mathbf{x}^l$ and $\mathbf{x}^r$ are the endpoints of the two antennae. We made the antennae length d dependent on the time instant t to make the algorithm adaptive for improved searching performance. d^t is the diameter of the search probed by beetle to decide an optimal direction. It is a usual practice to keep search region large in the beginning iterations of optimization and gradually decrease its size with time; this process is called annealing [32]. Therefore in our work, we defined d_t as follow:

$$d_t = \frac{d_0}{(1+\eta t)^\gamma}, \tag{3}$$

where d_0 is the initial antennae length, η and γ are the factors controlling the decay rate. We found the following rule provides a good initial guess for d_0,

$$d_0 \propto \sqrt{n} \tag{4}$$

where n is the dimensionality of the search space, this rule makes sure that for high dimensional spaces, the initial antennae length is also significant since the volume of the search region increase with dimensionality.

After the two direction $\mathbf{x}_l$ and $\mathbf{x}_r$ are calculated. The next step is to sense smell intensity i.e. evaluate objective function value at those points. The differenc in objective function values $f(\mathbf{x}_l)$ and $f(\mathbf{x}_r)$ defines the direction of increasing smell. Based on this fact, we can write an updated value $\mathbf{x}_{new}$ as:

$$\mathbf{x}_{new} = \mathbf{x}_t + \delta_t \text{sign}(f(\mathbf{x}_l) - f(\mathbf{x}_r))\overrightarrow{\mathbf{b}}, \tag{5}$$

where δ_t is called the step size and denote the euclidean distance between $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$. Note that $\mathbf{x}_{new}$ is the candidate for the update value not the actual update rule, the update rule $\mathbf{x}_{t+1}$ is defined later. The $\text{sign}(f(\mathbf{x}_r) - f(\mathbf{x}_l))$ factor make sure that if $f(\mathbf{x}_r) > f(\mathbf{x}_l)$, then step direction is $+\overrightarrow{\mathbf{b}}$, otherwise it is $-\overrightarrow{\mathbf{b}}$. The step length δ is also a function of time instant t. In our work, we defined it as proportional to the antennae length as follow:

$$\delta_t = c d_t, \tag{6}$$

where c is a constant factor representing the ratio of step length to the antennae length.

To define the update rule for $\mathbf{x}_{t+1}$, we also keep track of the best solution $\mathbf{x}_{best}$ and the corresponding objective function value f_{best} obtained thus far i.e. $f(\mathbf{x}_{best}) = f_{best}$. We only update the value of $\mathbf{x}_{t+1}$ if there is any improvement as compared to f_{best} i.e.,

$$\mathbf{x}_{t+1} = \begin{cases} \mathbf{x}_{new}, & \text{if } f(\mathbf{x}_{new}) > f_{best} \\ \mathbf{x}_t, & \text{otherwise.} \end{cases} \tag{7}$$

B. Modelling Antennae Fiber

As already explained in Section I, the antenna of beetle have small fiber, which provide micro-grain smell sensing capability. Therefore instead of generating just one random direction $\mathbf{b}$ is shown above, the beetle can sense smell at a swarm of points and use the smell measurement to estimate smell gradient, i.e., the direction of maximum smell change. This ability can be exploited in beetle search algorithm to enhance its convergence speed further. To model it mathematically, consider a set $\mathcal{B}$ of m normally distributed normalized random vector i.e. $\mathcal{B} = \{\overrightarrow{\mathbf{b}}_1, \overrightarrow{\mathbf{b}}_2, ..., \overrightarrow{\mathbf{b}}_m\} \subset \mathbb{R}^n$ corresponding to m antennae fiber. If the current location of the beetle is $\mathbf{x}_t$ than we can describe the location of each antennae fiber as follow:

$$\begin{aligned} \mathbf{x}^{\mathbf{b}_1} &= \mathbf{x}_t + d_t \overrightarrow{\mathbf{b}}_1, \\ \mathbf{x}^{\mathbf{b}_2} &= \mathbf{x}_t + d_t \overrightarrow{\mathbf{b}}_2, \\ \vdots \;&\vdots\; \vdots \\ \mathbf{x}^{\mathbf{b}_m} &= \mathbf{x}_t + d_t \overrightarrow{\mathbf{b}}_m, \end{aligned} \tag{8}$$

where d_t is the antennae length and controlled by same annealing rule as described in (3). We denote the set of these atennae fiber location as $\mathcal{X} = \{\mathbf{x}^{\mathbf{b}_1}, \mathbf{x}^{\mathbf{b}_2}, ..., \mathbf{x}^{\mathbf{b}_m}\}$

After generating a swarm of m antennae fiber, we create a set $\mathcal{F}$ of the objective values at points of set $\mathcal{X}$ i.e. $\mathcal{F} = \{f(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\} = \{f(\mathbf{x}^{\mathbf{b}_1}), f(\mathbf{x}^{\mathbf{b}_2}), ..., f(\mathbf{x}^{\mathbf{b}_m})\}$. In order to get an estimate for the gradient direction, our strategy is to find a vector from the point $\mathbf{x}_l \subset \mathcal{X}$ corrosponding to the lowest value in set $\mathcal{F}$, to the point $\mathbf{x}_h \subset \mathcal{X}$ corrosponding to the heighest value in the set $\mathcal{F}$. To make this strategy more robust, we choose $k(< m)$ lowest and heighest points instead of a single points. To denote it mathematically, we first create the following sets

$$\mathcal{F}_l = \{f : f \subset \mathcal{F} \wedge f \in \min_k \mathcal{F}\},$$
$$\mathcal{F}_h = \{f : f \subset \mathcal{F} \wedge f \in \max_k \mathcal{F}\}, \tag{9}$$

where notation $\min_k$ and $\max_k$ represnent choosing k minimum and maximum value from the set corrospondingly. These

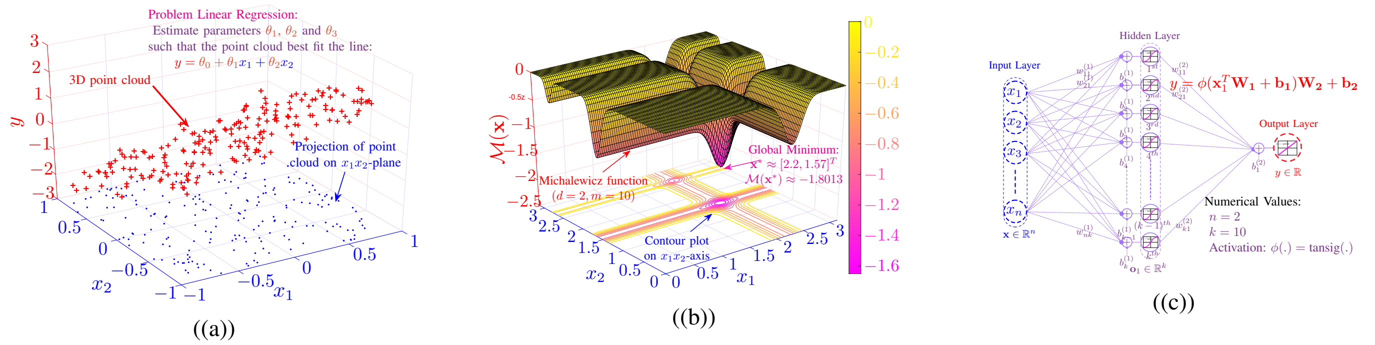


Fig. 2: Validations problems. (a) 3D point cloud for the linear regression and the neural network training problem. (b) Michalewicz function referred in Section III-B. (c) Neural network architecture used in validation problem of Section III-C.

values are selected by first evaluating the objective function at m antennae fiber locations and sorting them in a descending order. The value of k controls the robustness of the gradient estimation. The higher the value of k, the accurate the value of the estimated gradient to the true gradient; however, it will reduce the BAS algorithm's stochastic nature. The algorithm is more likely to get trapped in a local minimum. On the other hand, using a smaller k will make the algorithm more stochastic. It is more likely to reach a global solution but at the cost of a highly variant convergence profile (i.e., large oscillations and overshooting near the optimal point).

The sorted list is then used to select k elements with the highest and lowest values. Then we create subsets of $\mathcal{X}$ corresponding to values in $\mathcal{F}_l$ and $\mathcal{F}_h$ as follow:

$$\mathcal{X}_l = \{\mathbf{x} : \mathbf{x} \subset \mathcal{X} \wedge f(\mathbf{x}) \in \mathcal{F}_l\},$$
$$\mathcal{X}_h = \{\mathbf{x} : \mathbf{x} \subset \mathcal{X} \wedge f(\mathbf{x}) \in \mathcal{F}_h\}. \quad (10)$$

In simple terms, $\mathcal{X}_l$ and $\mathcal{X}_h$ are subsets of antenna fiber locations $\mathcal{X}$, which generate k lowest and highest values for the objective function f. We calculate the centroid of sets $\mathcal{X}_l$ and $\mathcal{X}_h$ to get the mean location of antenna fiber corresponding to the minimum and the maximum smell as follow:

$$\mathbf{x}_l = \sum_{\mathbf{x} \in \mathcal{X}_l} \mathbf{x}/k$$
$$\mathbf{x}_h = \sum_{\mathbf{x} \in \mathcal{X}_h} \mathbf{x}/k. \quad (11)$$

To summarize, the vectors $\mathbf{x}_l \in \mathbb{R}^n$ and $\mathbf{x}_h \in \mathbb{R}^n$ calculated in (11) corresponds to points inside a swarm of antennae fiber at which the value of the objective function is minimum and maximum respectively. Once we obtained these points, we calculate the approximate gradient as follow:

$$\widetilde{\nabla} = \mathbf{x}_h - \mathbf{x}_l, \quad (12)$$

where $\widetilde{\nabla}$ represent the approximate gradient. We can now use this approximated gradient to write an update value of $\mathbf{x}_{new}$ as follow:

$$\mathbf{x}_{new} = \mathbf{x}_t + \delta_t \text{sign}(f(\mathbf{x}_h) - f(\mathbf{x}_l))\widetilde{\nabla}, \quad (13)$$

the significance of δ_t and explanation for using sign(.) is already provided in (5). Similar to 7, we keep track of best solution $\mathbf{x}_{best}$ and corresponding objective function value f_{best} and write the following update rule for $\mathbf{x}_{t+1}$ as follow:

$$\mathbf{x}_{t+1} = \begin{cases} \mathbf{x}_{new}, & \text{if } f(\mathbf{x}_{new}) > f_{best}, \\ \mathbf{x}_t, & \text{otherwise.} \end{cases} \quad (14)$$

III. Validation Methodology

In this section, we will present the validation methodology to verify the performance, efficacy, and efficiency of the proposed BAS-swarm algorithm. We choose a set of three problems to present the validation results in this paper.

A. Problem: Linear Regression

Linear regression is a convex optimization problem and simplest of all the three validation since it only has a single minimum. The linear regression problem can be defined as follow: given an array of n variables $\{x_1, x_2, ..., x_n\}$, estimate the parameter $\{\theta_0, \theta_1, \theta_2, ..., \theta_n\}$ which best fit the following equation:

$$y = \theta_n x_n + \theta_{n-1} x_{n-1} + ... + \theta_1 x_1 + \theta_0 = \sum_{i=0}^{n} \theta_i x_i, \quad (15)$$

where $x_0 = 1$. Lets denote an array of variables $\mathbf{x} = [x_0, x_1, x_2, ..., x_n]^T \in \mathbb{R}^{n+1}$ and an array of cofficient parameters $\boldsymbol{\theta} = [\theta_0, \theta_1, \theta_2, ..., \theta_n] \in \mathbb{R}^{n+1}$, the above equation can be simplified as:

$$y = \boldsymbol{\theta}^T \mathbf{x}. \quad (16)$$

If we are given with a set of m sample values of variable $\mathbf{x}$, i.e. $\{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_{m+1}\}$, and corrosponding values of variable y i.e. $\{y_1, y_2, ...y_m\}$ we can define a vector of residual as follow:

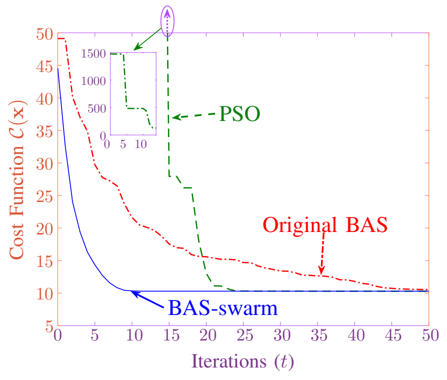
$$\mathcal{R}(\boldsymbol{\theta}) = [y_1 - \boldsymbol{\theta}^T\mathbf{x}_1,\ y_2 - \boldsymbol{\theta}^T\mathbf{x}_2,\ ...,\ y_m - \boldsymbol{\theta}^T\mathbf{x}_m]^T, \quad (17)$$

and define the following squared residuals cost function:

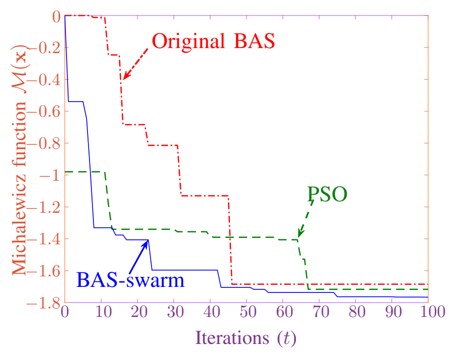
$$\mathcal{C}(\boldsymbol{\theta}) = ||\mathcal{R}(\boldsymbol{\theta})||_2, \quad (18)$$
$$= \sum_{i=0}^{m} (y_i - \boldsymbol{\theta}^T\mathbf{x}_i)^2. \quad (19)$$

This fundamentally transforms the linear regression problem into the following least square optimization problem:

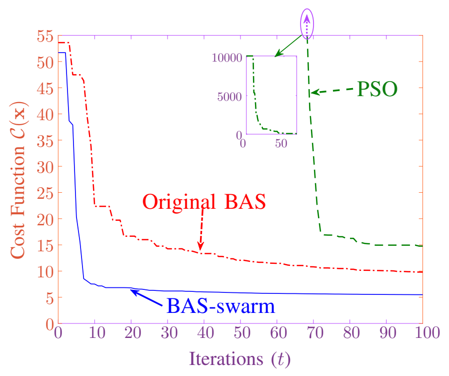
$$\boldsymbol{\theta}^* = \min_{\boldsymbol{\theta}} \ \mathcal{C}(\boldsymbol{\theta}) \ \equiv \ \max_{\boldsymbol{\theta}} -\mathcal{C}(\boldsymbol{\theta}). \quad (20)$$



((a))



((b))



((c))

Fig. 3: Numerical results. Convergence of BAS, BAS-swarn abd PSO for (a) linear regression problem, (b) Michalewicz function, and (c) Neural network training.

We use $-\mathcal{C}(\boldsymbol{\theta})$ as the objective function for our BAS-swarm algorithm.

For the numerical validation, we choose $n = 2$, $m = 300$ i.e. 300 samples points. The resultant data is shown in Fig. 2(a). The solution to this problem requires estimation of three parameters i.e. $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3] \in \mathbb{R}^3$ In Section IV we will present and discuss the numerical results for this problem.

B. Problem: Michalewicz function

Michalewicz function ([33]) is a nonlinear multimodal function with an infinite number of local minimum and unique global minimum. The function is excellent in testing the abilities of optimization algorithm to avoid local minimums. Therefore, it is used as a testbench function to evaluate the performance of the proposed optimization algorithm. Michalewicz function is defined as follow:

$$\mathcal{M}(\mathbf{x}) = -\sum_{i=1}^{d} \sin(x_i) \sin\left(\frac{ix_i^2}{pi}\right)^{2m}, \tag{21}$$

where d is the dimensionality of the function i.e. $\mathbf{x} = [x_1, x_2, ..., x_d] \in \mathbb{R}^d$. The value of m controls the steepness of the valleys of the function. It has been shown in literature that $\mathcal{M}(\mathbf{x})$ have a total of $d!$ local minima in the range $x_i \in [0, \pi]$. The plot of Michalewicz function is shown in 2(b). Since it is a highly non-convex function with several local minima, it provides a good validation of the performance of any optimization algorithm. In its actual form Michalewicz function is solved as a minimization problem, therefore we formulate the following maximization problem based on (21):

$$\mathbf{x}^* = \min_{\mathbf{x}} \ \mathcal{M}(\mathbf{x}) \ \equiv \ \max_{\mathbf{x}} \ -\mathcal{M}(\mathbf{x}). \tag{22}$$

For numerical validation, we choose $d = 2$ and $m = 10$. $d = 2$ was choosen for ease of visualization. The results are presented in Section IV.

C. Problem: Training Neural Network

The third problem involved the training of a single layer neural network. We choose this as a validation problem because most of the metaheuristic algorithm show poor performance when it comes to the training of a machine learning model. The machine learning models usually have a large number of trainable parameters and highly nonlinear transfer functions, which make it difficult to find an optimal solution by just using metaheuristic algorithms. On the other hand, the gradient-based methods have been known for their superior performance for the training of a machine learning model, e.g. neural networks ([34]). As already explained in Section II that our algorithm integrate the swarm behaviour of metaheuristic algorithms with the gradient estimation $\widetilde{\nabla}$ using (12) at each iteration. Therefore it tends to show better performance as compared to other metaheuristic algorithms in the training of the machine learning models.

IV. Results & Discussion

In this section we will present experimental results by solving the three validation problem presented in Sections III-A, III-B, and III-C. We compared the performance of the proposed algorithm with PSO and original BAS to comprehensively evaluate the efficiency of the proposed algorithm. We also present the statistical results to verify the robustness and repeatability of the proposed algorithm.

A. Problem: Linear Regression (Solution)

Among the three validation problems, this one simplest in term objective function. It is a convex optimization problem with a single minimum. The objective function given in (20) is essentially a second order polynomial in term of the estimation parameters θ_0, θ_1 and θ_2. We optimized the objective function using BAS-swarm, original BAS and PSO algorithm. For the BAS-swarm and original BAS, we used equal values for parameters d_0, η, γ and c. For BAS-swarm we additionally used $m = 20$ and $k = 5$. The significance of these parameters is given in Section II-B. For PSO we used the default parameter in MATLAB's toolbox implementation. The performance of the three algorithms is shown in Fig. 3(a). It can be seen that all three optimization algorithms are eventually able to reach the optimal solution. However, the speed at which they converge varies widely. Our proposed BAS-swarm took just around ten iterations to reach the optimum. On the other hand, the PSO tool around 25 iterations (2.5x slower) and original BAS tool around 50 iterations (i.e., 5x slower). The rate of convergence clearly shows the superiority of the proposed BAS-swarm as compared to the other metaheuristic algorithms. This fast rate of convergence can be attributed to the estimation of gradient which is absent from other optimization algorithms. Based on the obtained results we can say that the following equation of a plane is the best fit to the 3D point cloud given in Fig. 3(a):

$$y = 0.52 + 1.92x_1 + 5.00x_2 \tag{23}$$

B. Problem: Michalewicz function (Solution)

Searching the minima of Michalewicz function is a harder optimization problem as compared linear regression discussed above. It is a non-convex multiple modal function with multiple local minima. It is a good benchmark to test the global convergence of an optimization algorithm. For the Michalewicz function in (21), it is known in literature that for used $d = 2$ and $m = 10$, the function have a global minima at $\mathbf{x}^* \approx [2.20,\ 1.57]$ with an optimal value of $\mathcal{M}(\mathbf{x}^*) \approx -1.8013$. Similar to validation problem 1, we used BAS-swarm, PSO and original BAS for searching the optimal solution. The performance of the three algorithms is shown in Fig. 3(b). It can be seen that BAS-swarm has again had the fastest convergence rate as compared to the other algorithms. Additionally, the optimal point reached by the BAS-swarm is the same is the global optimum shown in Fig. 2(b). This shows that BAS-swarm can effectively reach global optimum even for a highly multimodal objective function. In comparison original BAS converges relatively slow and the final point is a bit far from the global optimum. For PSO, it can be seen that it converges to a local minimum very far from the global minimum shown in Fig. 2(b). This experiment again confirms the fast global convergence of the proposed BAS-swarm algorithm.

C. Problem: Neural Network Training (Solution)

Neural network training is the most complicated among all the three validation problem for the reasons explained in Section III-C. We used the point cloud shown in Fig. 2(a) to train the neural network. Although we used the same point cloud for linear regression problem but neural network are excellent in modeling nonlinear features of a system and therefore expected to produce better accuracy. The neural network architecture used in this paper is shown in Fig. 2(c). It has ten nonlinear neurons in the hidden layer. We solved the neural network training problem using all the three algorithms.

The performance of the optimization algorithms for training the neural network is shown in Fig. 3(c). It can be seen that the performance shown by BAS-swarm is far superior as compared to original BAS and PSO. The BAS converge to the optimum value in just ten iterations. Whereas original BAS and PSO show poor behavior while searching for the minimum value. The BAS-swarm is able to reach a minimum value of around 5 i.e. $\mathcal{C}(\mathbf{W}_1^*, \mathbf{W}_2^*, \mathbf{b}_1^*, \mathbf{b}_2^*) = 5$. In the case of the BAS the minimum value is around ten while for PSO the minimum value is around 15. It is also worth noting that although the linear regression and the neural network are modeling the same point cloud using linear regression and neural network respectively, the final cost for BAS-swarm in case of the neural network is much lower.

V. Conclusion

In this paper, we presented a nature-inspired metaheuristic algorithm, BAS-swarm, inspired by the original BAS algorithm. The algorithm makes use of the fact that beetles have excellent food searching ability because of their antennae and tiny antenna fiber. We provided comprehensive experimental results using three different optimization problems, including training a hidden-layer neural network, to validate the performance of our proposed algorithm. We solved all three validation problems using original BAS, PSO, and BAS-swarm. The experimental results show that on average BAS-swarm takes 500% fewer iterations as compared to original BAS, and 250% fewer iterations as compared to PSO. Additionally, by efficiently training a neural network using BAS-swarm, this paper demonstrates the potential application of metaheuristic algorithms in training the machine learning models. A potential future research direction is introducing an adaptive step-size adjustment mechanism (e.g., ADAM algorithm) to speed up the convergence. Another development is to implement the algorithm on a real-world system and evaluate its performance.

References

[1] S. Kalra and M. Nabi, "Tpwl simulation of large nonlinear circuits using subspace angle-based adaptive sampling," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 3, pp. 575–579, 2019.

[2] L. Liu and J. Huang, "Adaptive robust stabilization of output feedback systems with application to chua's circuit," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 9, pp. 926–930, 2006.

[3] C. W. Wu, "Synchronization in arrays of coupled nonlinear systems with delay and nonreciprocal time-varying coupling," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 52, no. 5, pp. 282–286, 2005.

[4] Y. Cheng, C. Li, S. Li, and Z. Li, "Motion planning of redundant manipulator with variable joint velocity limit based on beetle antennae search algorithm," *IEEE Access*, vol. 8, pp. 138788–138799, 2020.

[5] A. H. Khan, S. Li, and X. Luo, "Obstacle avoidance and tracking control of redundant robotic manipulator: An rnn based metaheuristic approach," *IEEE Transactions on Industrial Informatics*, 2019.

[6] M. Claesen and B. De Moor, "Hyperparameter search in machine learning," *arXiv preprint arXiv:1502.02127*, 2015.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[8] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[9] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, Springer, 2010.

[10] Z. Li, W. Zuo, and S. Li, "Zeroing dynamics method for motion control of industrial upper-limb exoskeleton system with minimal potential energy modulation," *Measurement*, vol. 163, p. 107964, 2020.

[11] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Advances in neural information processing systems*, pp. 315–323, 2013.

[12] Z. Li and S. Li, "A sparse optimization based control method for manipulator with simultaneous potential energy minimization," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2020.

[13] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, "Gradient descent only converges to minimizers," in *Conference on Learning Theory*, pp. 1246–1257, 2016.

[14] D. Chen, S. Li, F.-J. Lin, and Q. Wu, "New super-twisting zeroing neural-dynamics model for tracking control of parallel robots: A finite-time and robust solution," *IEEE transactions on cybernetics*, vol. 50, no. 6, pp. 2651–2660, 2019.

[15] Z. Li, C. Li, S. Li, and X. Cao, “A fault-tolerant method for motion planning of industrial redundant manipulator,” *IEEE transactions on industrial informatics*, vol. 16, no. 12, pp. 7469–7478, 2019.

[16] A. H. Khan, X. Cao, V. N. Katsikis, P. Stanimirović, I. Brajević, S. Li, S. Kadry, and Y. Nam, “Optimal portfolio management for engineering problems using nonconvex cardinality constraint: A computing perspective,” *IEEE Access*, vol. 8, pp. 57437–57450, 2020.

[17] I. Fister Jr, X.-S. Yang, I. Fister, J. Brest, and D. Fister, “A brief review of nature-inspired algorithms for optimization,” *arXiv preprint arXiv:1307.4186*, 2013.

[18] J. Krause, J. Cordeiro, R. S. Parpinelli, and H. S. Lopes, “A survey of swarm algorithms applied to discrete optimization problems,” in *Swarm Intelligence and Bio-Inspired Computation*, pp. 169–191, Elsevier, 2013.

[19] P. J. Angeline, G. M. Saunders, and J. B. Pollack, “An evolutionary algorithm that constructs recurrent neural networks,” *IEEE transactions on Neural Networks*, vol. 5, no. 1, pp. 54–65, 1994.

[20] D. E. Golberg, “Genetic algorithms in search, optimization, and machine learning,” *Addion wesley*, vol. 1989, no. 102, p. 36, 1989.

[21] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune, “Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning,” *arXiv preprint arXiv:1712.06567*, 2017.

[22] A. T. Khan and S. Li, “Human guided cooperative robotic agents in smart home using beetle antennae search,” *SCIENCE CHINA Information Sciences*.

[23] A. R. Khan, A. T. Khan, M. Salik, and S. Bakhsh, “An optimally configured hp-gru model using hyperband for the control of wall following robot,” *International Journal of Robotics and Control Systems*, vol. 1, no. 1, pp. 66–74, 2021.

[24] J. Kennedy and R. Eberhart, “C. 1995. particle swarm optimization,” in *IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ*, pp. 1942–1948, 2001.

[25] A. T. Khan, S. Li, and X. Cao, “Control framework for cooperative robots in smart home using bio-inspired neural network,” *Measurement*, vol. 167, p. 108253, 2021.

[26] D. Chen, S. Li, and Q. Wu, “A novel supertwisting zeroing neural network with application to mobile robot manipulators,” *IEEE transactions on neural networks and learning systems*, 2020.

[27] D. Chen and Y. Zhang, “Robust zeroing neural-dynamics and its time-varying disturbances suppression model applied to mobile robot manipulators,” *IEEE transactions on neural networks and learning systems*, vol. 29, no. 9, pp. 4385–4397, 2017.

[28] A. H. Khan, Z. Shao, S. Li, Q. Wang, and N. Guan, “Which is the best pid variant for pneumatic soft robots? an experimental study,” *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 1, p. 1, 2019.

[29] A. H. Khan, X. Cao, S. Li, and C. Luo, “Using social behavior of beetles to establish a computational model for operational management,” *IEEE Transactions on Computational Social Systems*, vol. 7, no. 2, pp. 492–502, 2020.

[30] M. C. Gopfert, H. Briegel, and D. Robert, “Mosquito hearing: sound-induced antennal vibrations in male and female aedes aegypti,” *Journal of Experimental Biology*, vol. 202, no. 20, pp. 2727–2738, 1999.

[31] X. Jiang and S. Li, “Bas: beetle antennae search algorithm for optimization problems,” *arXiv preprint arXiv:1710.10724*, 2017.

[32] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[33] A. P. Engelbrecht, “Fitness function evaluations: A fair stopping condition?,” in *2014 IEEE Symposium on Swarm Intelligence*, pp. 1–8, IEEE, 2014.

[34] O. Nelles, *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer Science & Business Media, 2013.