Neural Computing and Applications (2024) 36:10475–10499
https://doi.org/10.1007/s00521-024-09566-5

ORIGINAL ARTICLE

SHuffled Ant Lion Optimization approach with an exponentially weighted random walk strategy

Pinar G. Durgut[1] · Mirac Bugse Tozak[1] · M. Tamer Ayvaz[1]

Received: 31 July 2023 / Accepted: 24 January 2024 / Published online: 18 March 2024

Abstract

Ant Lion Optimization (ALO) method is one of the population-based nature-inspired optimization algorithms which mimics the hunting strategy of antlions. ALO is successfully employed for solving many complicated optimization problems. However, it is reported in the literature that the original ALO has some limitations such as the requirement of high number of iterations and possibility of trapping to local optimum solutions, especially for complex or large-scale problems. For this purpose, the SHuffled Ant Lion Optimization (SHALO) approach is proposed by conducting two improvements in the original ALO. Performance of the proposed SHALO approach is evaluated by solving some unconstrained and constrained problems for different conditions. Furthermore, the identified results are statistically compared with the ones obtained by using the original ALO, two improved ALOs which are the self-adaptive ALO (saALO) and the exponentially weighted ALO (EALO), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO) approaches. Identified results indicated that the proposed SHALO approach significantly improves the solution accuracy with a mean success rate of 76% in terms of finding the global or near-global optimum solutions and provides better results than ALO (22%), saALO (25%), EALO (14%), GA (28%), and PSO (49%) approaches for the same conditions.

Keywords Ant Lion Optimization (ALO) · SHALO · Boundary shrinking procedure · Shuffling

1 Introduction

Reaching the global optimum solution is quite a difficult task for many engineering optimization problems. For this purpose, various deterministic and meta-heuristic optimization methods are developed in the literature. Deterministic methods mostly use the gradient and sometimes the hessian information to find a search direction. Although these methods are enormously efficient for finding the global optimum solutions for convex optimization problems, they are highly dependent on the initial solutions for the non-convex and multimodal optimization problems. Furthermore, these methods may be inadequate for solving the optimization problems where the objective functions cannot be differentiated. For this reason, meta-heuristic methods are generally preferred to solve these kinds of optimization problems [1, 2].

Meta-heuristic optimization methods are the approaches which mimic the processes observed in natural phenomena by considering some stochastic components [3]. In the literature, these methods are classified under two categories: local search-based optimization methods and population-based optimization methods [4]. In local search-based optimization methods, the search process is started with a single solution and the objective function value is continuously improved by means of the considered heuristic processes. On the contrary, in the population-based optimization methods, the same tasks are conducted by considering a population rather than a single solution [5–7]. Note that the local search-based meta-heuristic optimization methods have some advantages including fast convergence, strong exploitation capability. However, use of the population-based methods is usually preferred for

✉ M. Tamer Ayvaz
tayvaz@pamukkale.edu.tr

Pinar G. Durgut
pgdurgut@gmail.com

Mirac Bugse Tozak
mbtozak@hotmail.com

[1] Department of Civil Engineering, Pamukkale University, Denizli, Türkiye

solving the complex problems since performance of the local-search-based methods is lower than the population-based methods especially during the global exploration process [6]. In the literature, there are different population-based metaheuristic optimization methods such as Genetic Algorithm (GA) [8], Ant Colony Optimization (ACO) [9], Particle Swarm Optimization (PSO) [10], Differential Evolution (DE) [11]. There exist a huge number of studies which employ these meta-heuristic optimization methods for solving different optimization problems in different disciplines of science.

Ant Lion Optimization (ALO) method [7] is one of the population-based nature-inspired optimization algorithms which mimics the hunting strategy of antlions. The ALO has been used in different studies and applications in the literature such as feature selection [12–15], data clustering [16–18], machine learning [19–22], and various engineering optimization problems [23–29]. Although ALO is successfully employed for solving many complicated problems, there are some reported limitations such as the requirement of high number of iterations, possibility of trapping to local optimum solutions, and the premature convergence especially for the complex or large-scale problems [30–32]. To overcome these limitations, different improved versions of ALO are proposed in the literature. Yao and Wang [33] proposed a Dynamic Adaptive Ant Lion Optimization (DAALO) approach by including the Levy flights to random walk process in ALO. The results of the DAALO approach are compared with GA, PSO, artificial bee colony (ABC), and original ALO. Dinkar and Deep [30] improved ALO by replacing the uniform distribution with Laplace distribution during generation of the random numbers. Furthermore, they considered an Opposition-Based (OB) Learning model for exploring the better candidate solutions. The performance of their proposed OB-L-ALO approach is evaluated with some benchmark and engineering design problems. Rajan et al. [34] improved ALO by replacing the elitism process of the original ALO with the weighted elitism concept by including a weight parameter to update ant positions. Their modified ALO approach (MALO) is used for solving the optimal reactive power dispatch problem. Wu et al. [35] improved the original ALO (IALO) by mapping the initial positions of ants and antlions to chaos space. They evaluated the performance of their IALO approach by solving some benchmark problems and by comparing the identified results with PSO, bat algorithm (BA), and ALO. Kilic and Yuzgec [36] improved the original ALO by modifying the process of returning the ants to the search space and used tournament selection instead of roulette wheel selection to solve the problem of parallel machine scheduling. Yang et al. [37] introduced an escape strategy and adaptive convergence criterion for improving the original ALO. Their proposed approach is used for estimating the support vector machine (SVM) parameters. Their results indicated that the proposed approach provided better results than ALO-, GA-, and PSO-based approaches. Toz [38] proposed a new boundary shrinking procedure for ALO which is based on the inverse of the incomplete Gamma function. This improved ALO (IALO) approach handles the problem of sudden boundary changes due to the stationary points in boundary shrinking procedure of the original ALO. Results of the proposed IALO approach indicated that this modification improved the performance of the original ALO during solution of the image clustering problem. Guo et al. [31] proposed a new search pattern consisting of a spiral complex path for the random walk process in the original ALO method. Their proposed approach aims to accelerate the convergence speed and evaluate its performance by solving some benchmark problems. Kilic et al. [39] applied the same modifications of Kilic and Yuzgec [36] for training an Adaptive Neuro-Fuzzy Inference System (ANFIS) model parameters. The performance of their Improved ALO approach with Tournament selection (IALOT) is also evaluated by solving some engineering optimization problems, and the obtained results are compared with DE, PSO, ABC, simulated annealing (SA), and touring ant colony optimization (TACO). Liu et al. [40] proposed a method called self-adaptive ALO (saALO) by using the trigonometric Sine function in random walk process to accelerate the convergence speed. Chaitanya et al. [41] modified the original ALO (MALO) by including a new tuning parameter in the elitism process. They used the proposed approach to solve the optimal reactive power dispatch problem and compared the results with original ALO, moth-flame optimization technique (MFO), PSO, and tabu search (TS). Singh and Singh [42] proposed an improved ALO (IALO) by shrinking of the radius of ant's random walks to avoid over-exploitation process. Their approach also aims to replace the averaging process with the blend crossover (BLX) operation for improving the exploration and exploitation capabilities. Yao et al. [43] proposed virtual force-directed ant lion optimization algorithm (VF-IALO) by including adaptive boundary shrinkage factor, dynamic weight coefficient for updating ant positions and improving selection of elite antlions. They used VF-IALO for coverage control of wireless sensor networks. El Bakrawy et al. [44] modified the original ALO (MALO) by including a new parameter which depends on the step length of each ant for updating their positions. Their proposed approach MALO is evaluated with some benchmark functions, and the identified results are compared with the original ALO. Chen et al. [32] proposed an exponential-weighted ant lion optimization algorithm (EALO) to increase the diversity of the random walks of the ants. The performance of EALO is evaluated by solving some benchmark problems, and the obtained results are compared with original ALO, saALO, PSO, and grey wolf optimization (GWO). Yan et al. [45] proposed a new approach which mutually integrates the original ALO with Levy flight and golden Sine algorithm (LGSALO). The main objective of their proposed LGSALO approach is to avoid trapping to local optimum solutions and increase the convergence accuracy. The performance of LGSALO is evaluated with benchmark problems and the obtained results compared with PSO and original ALO. Lin and Ouyang [46] implemented a anti-collision factor in the seagull optimization to the random walk of ants around antlion phase for improving the search capability of ALO. Moreover, they proposed an elimination probability for eliminating the last ants sorted by fitness and updating their position. Shen and Liu [47] are proposed an improved ALO algorithm in four phases: initialization, updating ant position, random walk around ant lion, and elitism. The performance of their proposed approach is evaluated with the layout optimization problem of branch pipelines. Lu et al. [48] introduced the chaotic strategy in the initial stage of each iteration and used the Gaussian mutation after the ant lion iteration to improve the performance of algorithms. They evaluated the performance of the algorithm using benchmark problems and then combined it with SVM to solve feature selection problem.

As can be seen from the aforementioned studies, improvements and modifications are mostly conducted in the random walk around antlions process of the original ALO method. The reason of these modifications is associated with the importance of this process for increasing the diversity of the solutions. In this context, the SHuffled Ant Lion Optimization (SHALO) approach is proposed in this study. In SHALO, the original ALO is modified by conducting two improvements. The first improvement deals with the boundary shrinking procedure of the random walk around antlions process. For this purpose, a new exponentially weighted approach is proposed. The second improvement is associated with the shuffling process which aims to increase the diversity of the random walk vector of the original ALO. The performance of the proposed SHALO approach is evaluated by solving four unconstrained and two constrained benchmark problems, and two constrained engineering design problems. All the problems are solved 100 times for different random number seeds, and the identified results are statistically compared with the ones obtained by using the original ALO, saALO [40], EALO [32], GA [8], and PSO [10]-based optimization approaches. The results of this analysis indicated that the proposed SHALO approach significantly improves the solution accuracy and can find the global or near-global optimum solutions with high success rates.

2 Ant Lion Optimization (ALO) method

ALO is a population-based meta-heuristic optimization method which is first developed by Mirjalili [7] depending on the hunting strategy of antlions in nature. Note that antlions belong to the family of Neuroptera (web-winged insects) which are carnivorous in both larvae and adult phases. In larvae phase, the antlions dig a cone-shaped sand pit and hide under the sand at the bottom of the pit for hunting insects, especially the ants (Fig. 1). When the antlion realizes that a prey (e.g., ant) walks inside pit, it starts throwing sand particles to the ant for sliding toward the bottom of the pit. After hunting and consuming the prey, the antlion repairs the shape of the pit to prepare for the next hunt. Depending on this philosophy, solution of an optimization problem by using ALO can be mathematically formulated as follows:

Let T is the maximum number of iterations$(t = 1, 2, 3, \cdots, T)$; $\boldsymbol{x}^t$ and $\widetilde{\boldsymbol{x}}^t$ be the population matrices of ants and antlions, respectively; n be the number of ants or antlions (candidate solutions) in the population; d be the number of decision variables of the problem; x_{ij}^t and $\widetilde{x}_{ij}^t$ are the elements of $\boldsymbol{x}^t$ and $\widetilde{\boldsymbol{x}}^t$ matrices at i^{th} candidate solution, j^{th} decision variable, and t^{th} iteration $(i = 1, 2, 3, \cdots, n\,;\; j = \quad 1, 2, 3, \cdots, d\,;\; t = 1, 2, 3, \cdots, T)$, respectively. Under these definitions, the population matrices of ants and antlions at t^{th} iteration can be defined as follows:

$$\boldsymbol{x}^t = \begin{bmatrix} x_{11}^t & x_{12}^t & \ldots & x_{1d}^t \\ x_{21}^t & x_{22}^t & \ldots & x_{2d}^t \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1}^t & x_{n2}^t & \ldots & x_{nd}^t \end{bmatrix} \tag{1}$$

$$\widetilde{\boldsymbol{x}}^t = \begin{bmatrix} \widetilde{x}_{11}^t & \widetilde{x}_{12}^t & \ldots & \widetilde{x}_{1d}^t \\ \widetilde{x}_{21}^t & \widetilde{x}_{22}^t & \ldots & \widetilde{x}_{2d}^t \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{x}_{n1}^t & \widetilde{x}_{n2}^t & \ldots & \widetilde{x}_{nd}^t \end{bmatrix} \tag{2}$$

The elements of $\boldsymbol{x}^t$ and $\widetilde{\boldsymbol{x}}^t$ are calculated randomly at the start of optimization $(t = 0)$ by considering the lower and upper bounds of the decision variable as follows:

$$x_{ij}^{t=0} = x_{\min,j}^{t=0} + \text{rand}(0,1) \cdot \left(x_{\max,j}^{t=0} - x_{\min,j}^{t=0} \right) \tag{3}$$

where $x_{\min,j}^{t=0}$ and $x_{\max,j}^{t=0}$ are the lower and upper bounds of the j^{th} decision variable at initial stage of the optimization, and $\text{rand}(0,1)$ is a uniform random number in the range of $(0,1)$. After initial generation of $\boldsymbol{x}^{t=0}$ and $\widetilde{\boldsymbol{x}}^{t=0}$ randomly, the corresponding fitness values are calculated as follows:

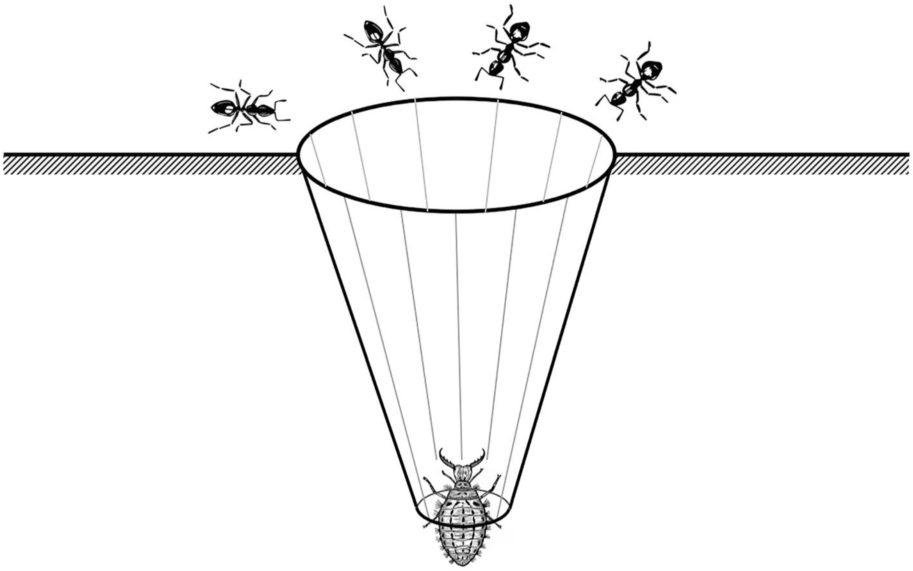

Fig. 1 Hunting strategy of antlions in the constructed cone-shaped pit

$$\mathcal{F}^t = \begin{bmatrix} f\left(\begin{bmatrix} x_{11}^t & x_{12}^t & \cdots & x_{1d}^t \end{bmatrix}\right) \\ f\left(\begin{bmatrix} x_{21}^t & x_{22}^t & \cdots & x_{2d}^t \end{bmatrix}\right) \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ f\left(\begin{bmatrix} x_{n1}^t & x_{n2}^t & \cdots & x_{nd}^t \end{bmatrix}\right) \end{bmatrix} \tag{4}$$

$$\widetilde{\mathcal{F}}^t = \begin{bmatrix} f\left(\begin{bmatrix} \widetilde{x}_{11}^t & \widetilde{x}_{12}^t & \cdots & \widetilde{x}_{1d}^t \end{bmatrix}\right) \\ f\left(\begin{bmatrix} \widetilde{x}_{21}^t & \widetilde{x}_{22}^t & \cdots & \widetilde{x}_{2d}^t \end{bmatrix}\right) \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ f\left(\begin{bmatrix} \widetilde{x}_{n1}^t & \widetilde{x}_{n2}^t & \cdots & \widetilde{x}_{nd}^t \end{bmatrix}\right) \end{bmatrix} \tag{5}$$

where $\mathcal{F}^t$ and $\widetilde{\mathcal{F}}^t$ represent the fitness vector of the ants and antlions at t^{th} iteration, respectively. After this step, a roulette wheel (RW) approach is employed to mimic the hunting behavior of antlions. This is an important stage in ALO since the selection of antlions is conducted based on their calculated fitness values. By applying the roulette wheel approach, the fitter antlions get higher chance to catch the ants for consuming [7]. After selection of the antlions by means of the roulette wheel approach, the sliding behavior of ants toward the bottom of the pit is simulated based on a boundary shrinking procedure in which the lower and upper bounds of the variables are adaptively decreased as follows:

$$x_{\min,j}^t = \frac{x_{\min,j}^t}{I} \tag{6}$$

$$x_{\max,j}^t = \frac{x_{\max,j}^t}{I} \tag{7}$$

$$I = 10^{\omega}\frac{t}{T} \tag{8}$$

As can be seen from Eqs. (6) and (7), the search space of the ants is decreased by dividing the corresponding lower and upper bounds to parameter I given in Eq. (8). Note that the value of I is dynamically calculated depending on the ratio of t/T and ω. The value of ω is used for adjusting the accuracy of exploitation and calculated by dividing the iteration domain into several segments as follows:

$$\omega = \begin{cases} 1 & \text{if} \quad t > 0 \\ 2 & \text{if} \quad t > 0.10T \\ 3 & \text{if} \quad t > 0.50T \\ 4 & \text{if} \quad t > 0.75T \\ 5 & \text{if} \quad t > 0.90T \\ 6 & \text{if} \quad t > 0.95T \end{cases} \tag{9}$$

To simulate the entrapment process of the ants inside the cone-shaped pit, the lower and upper bounds of the ants are updated by using the position of the antlions as follows:

$$x_{j,\min}^t = \begin{cases} \widetilde{x}_{ij,\text{RW}}^t + x_{j,\min}^t & \text{if } \operatorname{rand}(0,1) < 0.5 \\ \widetilde{x}_{ij,\text{RW}}^t - x_{j,\min}^t & \text{otherwise} \end{cases} \tag{10}$$

$$x_{j,\max}^t = \begin{cases} \widetilde{x}_{ij,\text{RW}}^t + x_{j,\max}^t & \text{if } \operatorname{rand}(0,1) \geq 0.5 \\ \widetilde{x}_{ij,\text{RW}}^t - x_{j,\max}^t & \text{otherwise} \end{cases} \tag{11}$$

where $\widetilde{x}_{ij,\text{RW}}^t$ is the position of the selected i^{th} antlion at t^{th} iteration by means of the RW approach. After this step, the random walk process of the ants to find a forage is modeled based on the following equation:

$$\boldsymbol{\mathcal{X}}_j = \left[0, \mathcal{X}_j^1, \mathcal{X}_j^2, \cdots, \mathcal{X}_j^t, \cdots, \mathcal{X}_j^T\right] \tag{12}$$

$$\mathcal{X}_j^t = \mathcal{C}(2 \cdot \ r(t) - 1) \tag{13}$$

$$r(t) = \begin{cases} 1 \ \text{ if } \mathrm{rand}(0,1) > 0.5 \\ \quad 0 \ \text{ otherwise} \end{cases} \tag{14}$$

where $\boldsymbol{\mathcal{X}}_j$ is the random walk vector for j^{th} decision variable $(j = 1, 2, 3, \cdots, d)$; $\mathcal{X}_j^t$ is the t^{th} element of the $\boldsymbol{\mathcal{X}}_j$ vector; $r(t)$ is a stochastic step function; $\mathcal{C}(\circ)$ is a function which calculates the cumulative sum of the given input. Note that the elements of the random walk vector are normalized to ensure satisfying the specified lower and upper bounds of the decision variables based on the following min–max normalization equation:

$$\mathcal{X}_j^t = \frac{\left(\mathcal{X}_j^t - a_j\right) \cdot \left(x_{j,\max}^t - x_{j,\min}^t\right)}{b_j - a_j} + x_{j,\min}^t \tag{15}$$

where $a_j = \min\left(\boldsymbol{\mathcal{X}}_j\right)$ and $b_j = \max\left(\boldsymbol{\mathcal{X}}_j\right)$. During solution of the optimization problem, keeping the elite solution is an important step in any phase of the search process. Thus, the best solution should be stored to guarantee the improved convergence in every iteration. Note that in ALO, new positions of the ants are calculated by using the random walk around the selected antlion by roulette wheel approach and the elite antlion in terms of the output of Eq. (15) as follows:

$$x_{ij}^t = \frac{\mathcal{R}a_j^t + \mathcal{R}e_j^t}{2} \tag{16}$$

where $\mathcal{R}a_j^t$ and $\mathcal{R}e_j^t$ are the random walk around the selected (RW approach) and the elite antlions for j^{th} decision variable at t^{th} iteration, respectively. When the ants get trapped at the bottom of the pit, the antlion updates its position to prepare the trap for a new prey. According to the provided source code of ALO [7], this process is described based on the following steps: (*i*) determine fitness vector for the recently calculated ant positions; (*ii*) append the fitness vector of ants to the bottom of the fitness vector of antlions (double fitness vector); (*iii*) append the position matrix of ants to the bottom of the position matrix of antlions (double position matrix); (*iv*) sort double fitness vector in ascending order and determine the corresponding sort index values; (*v*) re-order the double position matrix based on the sort index values; (*vi*) assign new positions and fitness values of antlions from the first half of the sorted double position matrix and double fitness vector, respectively. It should be noted that the computational scheme described above is iterated until reaching the maximum number of iterations in the original ALO.

3 The proposed SHuffled Ant Lion Optimization algorithm (SHALO)

As indicated previously, the ALO method is applied for solving many different optimization problems in the literature due to its interesting philosophy and simple computational structure. However, ALO has some limitations in the random walk around antlions process. Therefore, a significant number of the improved ALO approaches in the literature focus on this issue and try to improve the solution accuracy by modifying this process [30–45]. Depending on this issue, the SHALO approach is proposed in this study for increasing the solution accuracy of ALO approach. Note that the computational structure of the proposed SHALO approach also focuses on modifying the random walk around ant lions process with two improvements in the current ALO structure. The first improvement is related to the boundary shrinking procedure of ALO approach which is an important and essential part since the original procedure depends on a step-by-step decreasing procedure for shrinking the boundary. This procedure is controlled with the parameter ω in Eq. (9) which is used to adjust level of exploitation. Note that the value of ω in Eq. (9) is suddenly increased at some stationary points and this change results with a suddenly increased value of I as given in Eq. (8). Although this procedure can be useful for conducting the boundary reduction around the solutions, it can reduce the random search capability of the algorithm due to a sudden increase of the value of I at stationary points [38]. Therefore, a new exponentially weighted boundary shrinking approach is proposed to solve this problem as follows:

$$I = \begin{cases} 1 \ \text{ if } \ k = 0 \\ \widetilde{I} \ \text{ if } \ k = 1, 2, 3, 4, 5 \end{cases} \tag{17}$$

$$\widetilde{I} = \left(I_k^{\min} + I_k^{\max}\right) - I_k^{\max} \cdot \left\{ e^{\left[\mathcal{L}_k \cdot \left(t - t_k^{\min}\right)\right]} + \lambda \cdot \mathrm{rand}(0,1) \cdot |\mathcal{L}_k| \cdot \left(t - t_k^{\min}\right)\right\} \tag{18}$$

$$\mathcal{L}_k = \frac{\ln\left(\frac{I_k^{\min}}{I_k^{\max}}\right)}{t_k^{\max} - t_k^{\min}} \tag{19}$$

where $\widetilde{I}$ is the value of the proposed exponentially weighted function, $I_k^{\min}$ and $I_k^{\max}$ are the minimum and maximum I values for k^{th} region in Fig. 2, $t_k^{\min}$ and $t_k^{\max}$ are the minimum and maximum t values for k^{th} region in Fig. 2, $\mathcal{L}_k$ is a function whose value is calculated logarithmically based on Eq. (19) for k^{th} region in Fig. 2, $|\circ|$ is the absolute value, and λ is the parameter controlling the magnitude of the random deviations from the value of $\widetilde{I}$.

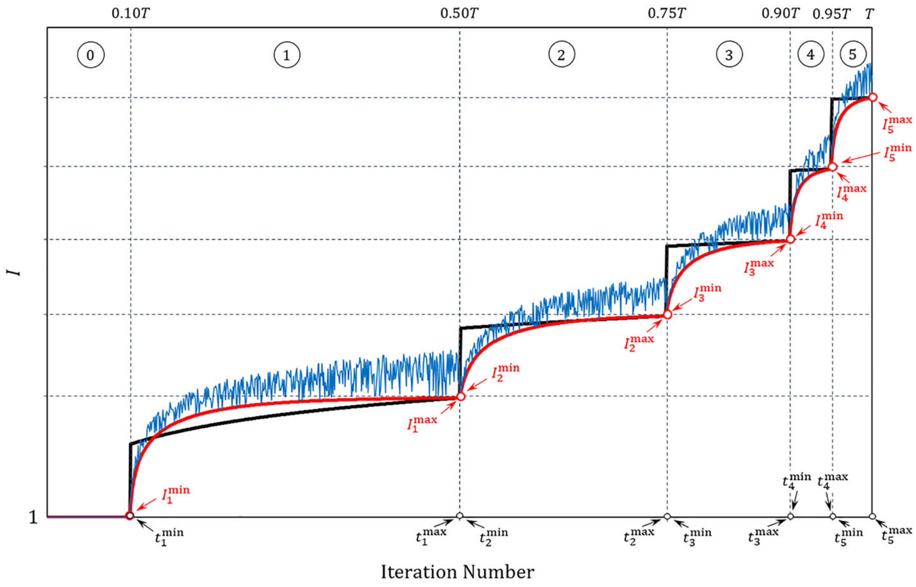

Fig. 2 Variation of I versus t for ALO (black lines) and SHALO (red ($\lambda = 0$) and blue ($\lambda = 1$) lines) approaches (color figure online)

The detailed representation of the parameters used in Eqs. (18) and (19) can be seen in Fig. 2.

As can be seen from Fig. 2, the iteration domain of the optimization problem is partitioned into 6 regions ($k = 0, 1, 2, \cdots, 5$). According to the original ALO formulation, the value of I is suddenly increased at starting points of the 1st to 5th regions. On the other hand, the value of I linearly increases inside of each region depending on the ratio of (t/T). Figure 2 shows the variation of I for each region in the original ALO method with the black lines. Note that the minimum and maximum values of I for each region are taken as $I_k^{\min}$ and $I_k^{\max}$ in SHALO. Similarly, for each region, the minimum and maximum values of t are considered as $t_k^{\min}$ and $t_k^{\max}$ in Eqs. (18) and (19). Note that Fig. 2 also includes the variation of the proposed exponentially weighted function of $\widetilde{I}$. As can be seen, for $\lambda = 0$ (red line), the proposed exponentially weighted function does not have any random deviations in terms of the calculated $\widetilde{I}$ values. This means that the proposed function behaves just as the one given in the original ALO method. The key difference between the proposed function and the one given in the original ALO is that the proposed function does not have any stationary points where the value of I suddenly increases. Instead, the function value exponentially increases between the same minimum and maximum values for each region. This kind of use increases the random search capability of the approach rather than the original ALO method. Note that the random search capability of the proposed exponential function can also be increased by perturbing the calculated $\widetilde{I}$ values with random deviations. This perturbation is also given in Fig. 2 for $\lambda = 1$ (blue line). As can be seen, although trend of the calculated $\widetilde{I}$ values remains the same, their values have strong local oscillations depending on the considered value of λ. Therefore, the proper selection of the value of λ is very important since it adjusts the magnitude of random deviations from the calculated value of $\widetilde{I}$.

The second improvement of the SHALO approach is related to the modification of the min–max normalization process given in Eq. (15) in original ALO. As indicated previously, the generated random walk vector cannot be directly used for updating the position of ants. Therefore, values in this vector are normalized by using Eq. (15) in the original ALO by considering the minimum and maximum values of the random walk vector. In the proposed SHALO approach, the associated min–max normalization is modified as follows:

$$\mathcal{X}_j^t = \frac{\left(\mathcal{X}_j^t - \widetilde{a}_j\right)\cdot\left(x_{j,\max}^t - x_{j,\min}^t\right)}{\widetilde{b}_j - \widetilde{a}_j} + x_{j,\min}^t \tag{18}$$

$$\widetilde{a}_j = \mathrm{mean}\left(\boldsymbol{\mathcal{X}}_j\right) - \min\left(\boldsymbol{\mathcal{X}}_j\right) \tag{19}$$

$$\widetilde{b}_j = \max\left(\boldsymbol{\mathcal{X}}_j\right) - \mathrm{mean}\left(\boldsymbol{\mathcal{X}}_j\right) \tag{20}$$

where $\mathrm{mean}\left(\boldsymbol{\mathcal{X}}_j\right)$ is the arithmetic mean of the random walk vector for j^{th} variable. As can be seen from Eq. (20), the values of a_j and b_j in Eq. (15) are replaced with $\widetilde{a}_j$ and $\widetilde{b}_j$ by including the mean of the random walk vector as an additional variable. This inclusion significantly increases the diversity of random walk vector since it shuffles the content of the random walk vector. During this process, the two cases given in Fig. 3 are observed.

As can be seen from Fig. 3, the cases where the $\mathrm{mean}\left(\boldsymbol{\mathcal{X}}_j\right)$ values are lower and greater than $\mathrm{median}\left(\boldsymbol{\mathcal{X}}_j\right)$ are named as Case 1 and 2, respectively. Although these two cases are not important in the original ALO, they are

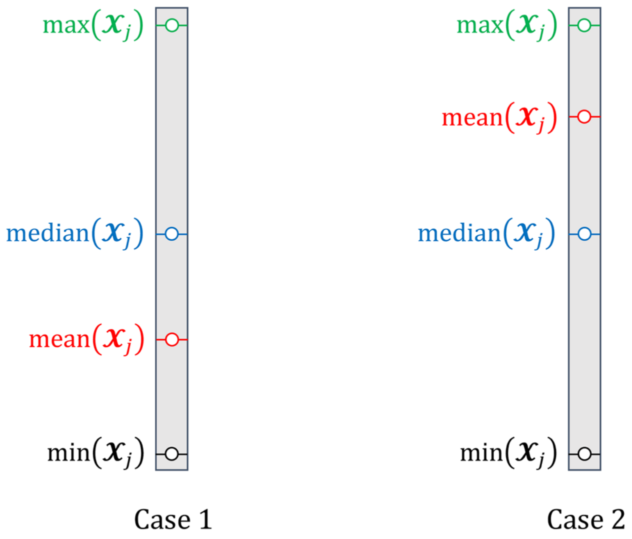

Fig. 3 The two cases observed during the shuffling process

important in SHALO for understanding the nature of the shuffling process. As an example, Fig. 4a–c compares the random walks for both ALO and SHALO approaches at 10th, 100th, and 800th iterations in terms of trends and statistics. Note that both approaches in Fig. 4 are executed by considering the same random number seeds to eliminate the influence of random numbers. As can be seen from Fig. 4a, the calculated random walks are in the same trend although the corresponding values in SHALO are much greater than the ones obtained in the original ALO. The random walks in Fig. 4a represent the behavior of Case 2 for both ALO and SHALO approaches where the calculated mean is greater than the median. On the other hand, in Fig. 4b, the calculated random walks also have greater values than those obtained in ALO. However, the corresponding values of SHALO are all obtained as upside down compared to the ones in the original ALO. The reason for this change is associated with the conditions given in Fig. 3 such a way that ALO represents the behavior of Case 2, whereas the SHALO is Case 1. Finally, in Fig. 4c, the same outcome is also observed where random walks in SHALO are all obtained as upside down compared to the original ALO due to their differences in terms of the cases in Fig. 3. Note that comparison of Fig. 4a–c also indicates the decreased values of the random walks in SHALO. As an example, Fig. 5 shows the variation of random walks through iterations for both ALO and SHALO approaches. As can be seen, the random walk vector of SHALO gets significantly greater values than those calculated values for ALO in early iterations. However, these big differences decrease in later iterations and values of the random walk vector approach to the optimum values. It is concluded from these results that the proposed SHALO approach significantly increases the diversity of random walk vector, especially in early iterations by increasing and shuffling the random walks.

4 Comparison with other approaches

As indicated previously, there are various modified versions of ALO in the literature for improving the quality of random walks around antlion process. Since SHALO also aims to improve this process, it is essential and required task to compare its performance with these approaches. In this context, the identified results of SHALO are compared with the original ALO for the same initial populations, optimization parameter values, and random number seeds. Similarly, all the identified results are compared with two improved versions of ALO which are the self-adaptive ALO (saALO) [40] and the Exponential-weighted ALO (EALO) [32] approaches. Both saALO and EALO approaches aim to improve the boundary shrinking procedure by modifying the variable I. These modifications are conducted by introducing the sinusoidal and exponential terms to saALO and EALO approaches, respectively, as follows:

$$\text{saALO}: \ I = 10^{\omega}\frac{t}{T}\left(0.5 + \sin\left(\frac{\pi t}{2T}\right)\cdot \text{rand}(0,1)\right) \tag{23}$$

$$\text{EALO}: \ I = 10^{\omega}\frac{t}{T}\left(\exp\left(\frac{t}{T}\right)\cdot \text{rand}(0,1)\cdot \eta\right) \tag{24}$$

As can be seen from Eqs. (23) and (24), both saALO and EALO approaches use the same structure with Eq. (8) such a way that the value of I linearly increases with the ratio of t/T together with a dynamic change with the parameter of ω. The key difference is that in saALO, an additional term is introduced to Eq. (8) to include the sinusoidally changed random perturbations. Similarly, in EALO, this random perturbation is included to Eq. (8) by means of an exponential term together with a weight of η which controls the magnitude of random perturbations. Note that the performance of the proposed SHALO approach is also evaluated by solving the same problem via GA and PSO optimization approaches by using the same random number seeds with the other approaches.

5 Numerical applications

The applicability of the proposed SHALO approach is evaluated by solving four unconstrained and four constrained benchmark problems. The last two of the constrained problems are related with the solution of well-known engineering design problems. These problems are also solved with the original ALO, saALO, EALO, GA, and PSO approaches. All the approaches are executed by

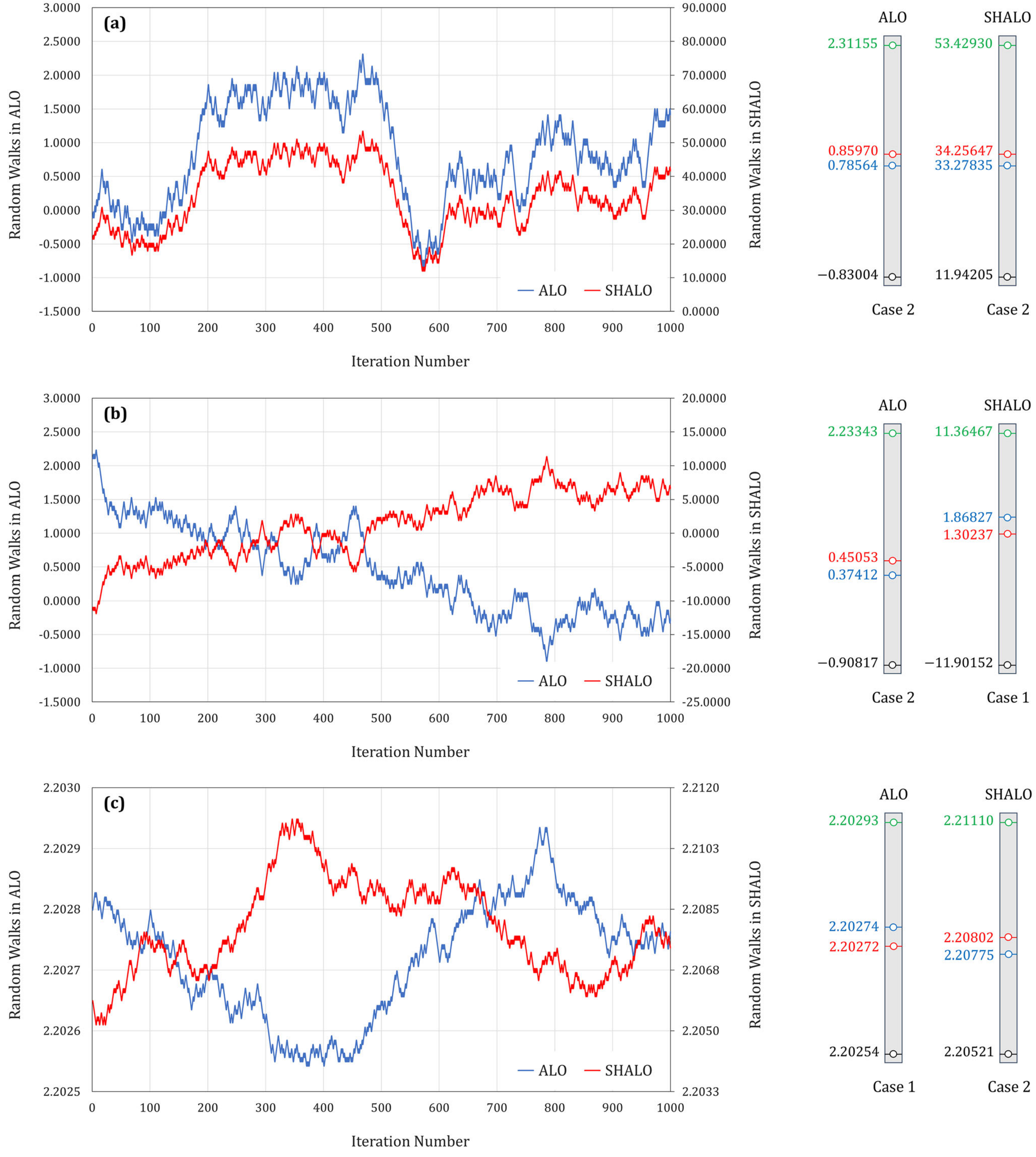

Fig. 4 Comparison of the trends statistics of random walks for both ALO and SHALO approaches for **a**: 10th iteration; **b**: 100th iteration; **c**: 800th iteration (the figures on the right side represent the statistics of the random walks in terms of the minimum (black), maximum (green), mean (red), and median (blue) metrics (color figure online))

considering the number of candidate solutions (population) as 50 and the maximum iteration (generation) number of 1000. To make an exact comparison for all the approaches, a data set including 100 different uniform random number seed values is randomly generated for statistically evaluating the model results. For this purpose, commonly used statistical measures (minimum, maximum, mean, median, and standard deviation) are used together with a measure of

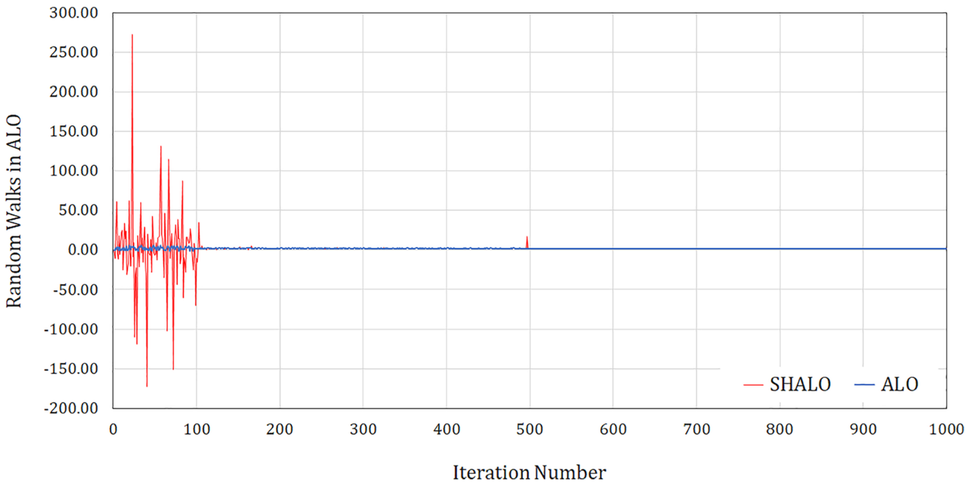

Fig. 5 Variation of random walks through optimization iterations for both ALO and SHALO approaches (color figure online)

Success Rate (SR), given in Eq. (25), which is calculated by dividing the number of successful solutions (SS) to the number of total solutions (TS) [49]. A model execution is assumed to be successful depending on the mathematical definition given in Eqs. (26).

$$SR(\%) = \frac{\sum SS}{TS} \cdot 100 \tag{25}$$

$$SS = \begin{cases} \begin{bmatrix} 1 & \text{if } |f(x) - f(x^*)| \le \varepsilon \\ 0 & \text{otherwise} \end{bmatrix} & \text{if } \; f(x^*) = 0 \\ \begin{bmatrix} 1 & \text{if } \dfrac{|f(x) - f(x^*)|}{|f(x^*)|} \le \varepsilon \\ 0 & \text{otherwise} \end{bmatrix} & \text{if } \; f(x^*) \neq 0 \end{cases} \tag{26}$$

where $f(x^*)$ is the global optimum solution of the optimization problem, and ε is a threshold value. As can be seen from Eq. (26), the value of SS equals to 1 if absolute or absolute relative error values are lower than the given ε value depending on the value of the global optimum solution. Note that the measure of SR is used to evaluate the model results in terms of the calculated objective function values. Evaluation of model results in terms of the relative closeness to the optimum solution is conducted based on the accuracy metric (AC) as follows [36, 39]:

$$AC\,(\%) = \left(1 - \frac{\left|x'_k - x^*_k\right|}{\left|x^{t=0}_{\max,k} - x^{t=0}_{\min,k}\right|}\right) \cdot 100 \tag{27}$$

where x'_k and x^*_k represent the calculated and the optimum solution of the k^{th} decision variable ($k = 1, 2, 3, \cdots, d$). Note that performance of the proposed SHALO approach is evaluated for 12 different λ values ($\lambda \in [0, 1, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000]$) where the model is executed 100 times for each λ values by considering the generated random number seed data set. The reason for this evaluation is to assess the applicability of the proposed boundary shrinking procedure for different magnitudes of random perturbations.

5.1 Unconstrained optimization applications

5.1.1 Michalewicz's test function

Michalewicz's test function is a nonlinear multimodal test function which contains d local optimum [50]. The mathematical description of the function together with the lower and upper bounds of the decision variables is given in Eq. (28).

$$\min f(x) = -\sum_{k=1}^{d} \sin(x_k)\left[\sin\left(\frac{k.x_k^2}{\pi}\right)\right]^{2\mu} \tag{28}$$

$$\text{s.t. } 0 \le x_k \le \pi \; ; \; k = 1, 2, 3, \cdots, d \tag{28a}$$

where the parameter μ describes the "steepness" of the valleys or sides and is assumed to be 10 for this solution. This function has a global optimum solution at

$$x^* = [2.202906 \; 1.570796 \; 1.284992 \; 1.923058 \; 1.720470]$$

with a function value of $f(x^*) = -4.688$ for $d = 5$. Figure 6 shows three-dimensional representation of this function for $d = 2$. As can be seen, the function has many local optimum solutions around global optimum that is why it may be classified as a difficult optimization test problem. As indicated previously, this problem is solved for different λ values in SHALO; each of them consists of 100 model executions. For these model executions, the calculated SR values are given in Fig. 7.

As can be seen from Fig. 7, the maximum value of SR is obtained for the solution with $\lambda = 20$. For this solution, the results of the corresponding 100 model executions are compared with the ones obtained by using ALO, saALO, EALO, GA, and PSO approaches, respectively. As

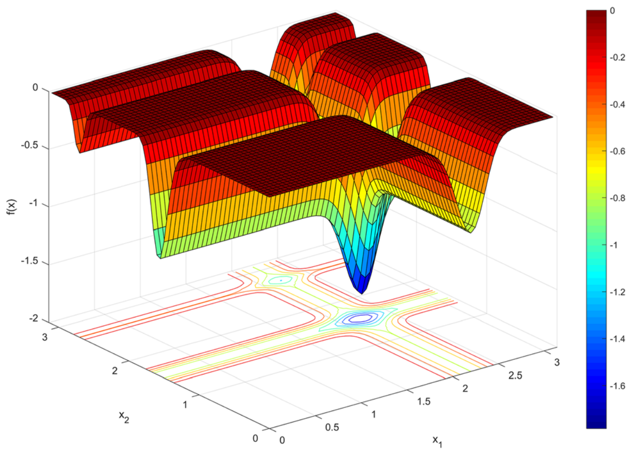

Fig. 6 Michalewicz's test function

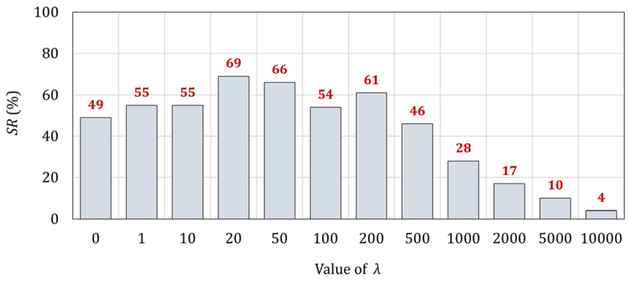

Fig. 7 Variation of *SR* (%) for different λ values for Michalewicz's test function

indicated previously, all these approaches are also executed 100 times for the same random number seed data set. Table 1 compares the identified results of these model executions.

It is seen that each approach determined the global optimum solution (− 4.688) at least one time for the conducted 100 model executions. When the obtained results are evaluated, the associated *SR* values of ALO, saALO, and EALO approaches are obtained as 3%, 4%, and 3%, respectively. However, the corresponding *SR* value is obtained as 48% and 32% for GA and PSO, respectively. For this example, the proposed SHALO approach is resulted with a *SR* value of 69% which is better than the other approaches. This outcome can also be seen from the mean and median values where the calculated values are very close to the global optimum solution in SHALO. For the calculated *AC* values, while the original ALO has the mean accuracy of 90%, all the other approaches have the mean accuracy 97% and over. These results indicate that there are significant local optimum solutions around the global optimum. The convergence plots for ALO, saALO, EALO, GA, PSO, and SHALO approaches are compared in Fig. 8. Note that these plots are composed by taking the arithmetic mean of the 100 model executions for each optimization approach. Results of these plots indicate that the proposed SHALO approach converges to the better mean objective function values than the other approaches when considering overall 100 model executions. For the best solutions in Table 1, the final decision variables and objective function values are compared in Table 2. As can be seen, all the identified values are the same since all the approaches converged to the global optimum solution at least once.

5.1.2 Rastrigin's test function

Rastrigin's test function is also a nonlinear and multimodal test function containing the large number of local optimum solutions [51]. The mathematical form of the function together with the considered lower and upper bounds of the decision variables is given in Eq. (29).

$$\mathrm{Min} f(x) = 10d + \sum_{k=1}^{d}\left[x_k^2 + 10\cos(2\pi x_k)\right] \tag{29}$$

$$\text{s.t.} \quad -5.12 \le x_k \le 5.12 \quad k = 1, 2, \cdots, d \tag{29a}$$

This function has a global optimum solution at $x_k^* = 0$ $(k = 1, 2, \cdots, d)$ with a function value of $f(x^*) = 0$ for any given value of d. In this study, the problem is solved by considering $d = 5$. Figure 9 demonstrates three-dimensional representation of this function for $d = 2$. As can be seen, the function has many local optimums, and therefore, it may also be classified as a difficult optimization test problem. By applying the same solution scheme, this function is also solved 100 times for different λ values.

Table 1 Evaluation of the 100 model executions for Michalewicz's test function for different optimization approaches

Measure	ALO	saALO	EALO	GA	PSO	SHALO
SR(%)	3	4	3	48	32	69
AC(%)	90	97	97	99	99	99
Best	− 4.688	− 4.688	− 4.688	− 4.688	− 4.688	− 4.688
Worst	− 3.151	− 3.151	− 2.948	− 4.333	− 4.059	− 4.496
Mean	− 4.119	− 4.122	− 4.040	− 4.616	− 4.596	− 4.668
Median	− 4.251	− 4.251	− 3.923	− 4.646	− 4.646	− 4.688
Std. deviation	0.435	0.436	0.464	0.087	0.100	0.040

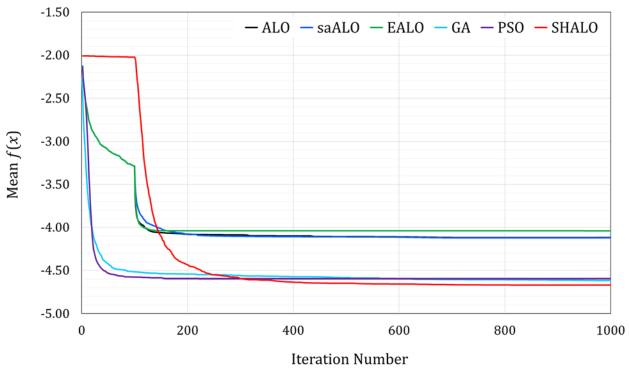

Fig. 8 Comparison of the convergence histories of different approaches for the Michalewicz's test function

Figure 10 shows the final *SR* values for these model executions with different λ values.

As can be seen from Fig. 10, the maximum *SR* value is obtained as 95% which is obtained for the solution with $\lambda = 100$. For this solution, the results of the corresponding 100 model executions are compared with the ones obtained by using ALO, saALO, EALO, GA, and PSO approaches, respectively. The identified results of these model executions are compared in Table 3. As can be seen, the original ALO is not successful in determining the global optimum solution for any of the 100 model executions. Similarly, the saALO and EALO approaches obtained the global optimum solution only once. For the same conditions, GA and PSO approaches determined the global optimum solution 46 and 36 times, respectively. On the other hand, the proposed SHALO approach determined the global optimum solution 95 times out of 100 model executions. This outcome can also be seen from the calculated mean and median values which are close to the global optimum solution in SHALO although there are some differences in the other approaches. When the results of *AC* measure are evaluated, ALO, saALO, and EALO approaches resulted with the same mean *AC* value of 93%. On the other hand, GA, PSO, and SHALO approaches resulted with the mean *AC* values of 99%, 98%, and 100%, respectively.

The convergence plots of each approach for the mean objective function values of 100 model executions are compared in Fig. 11. Results of these plots demonstrate

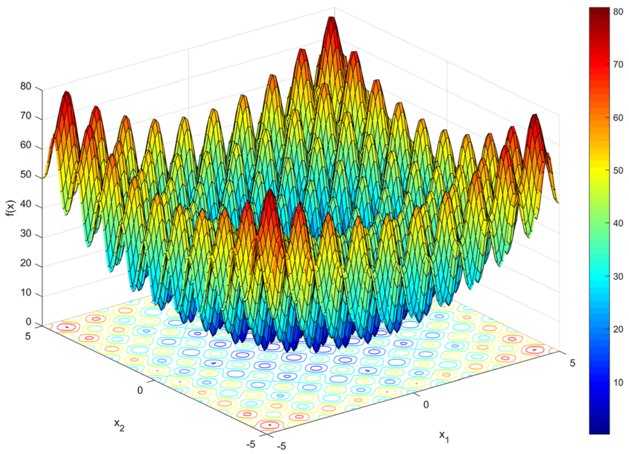

Fig. 9 Rastrigin's test function

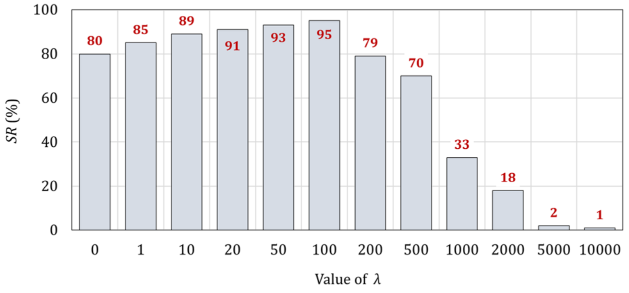

Fig. 10 Variation of *SR* (%) for different λ values for Rastrigin's test function

Table 3 Evaluation of the 100 model executions for Rastrigin's test function for different optimization approaches

Measure	ALO	saALO	EALO	GA	PSO	SHALO
SR(%)	0	1	1	46	36	95
AC(%)	93	93	93	99	98	100
Best	0.995	0.000	0.000	0.000	0.000	0.000
Worst	14.924	13.929	14.924	3.980	5.970	0.995
Mean	5.253	4.686	4.457	0.746	1.094	0.050
Median	4.975	3.980	3.980	0.995	0.995	0.000
Std. deviation	3.224	2.922	2.953	0.825	1.165	0.217

Table 2 The final decision variables and objective function values for Michalewicz's test function for the best solutions in Table 1

Variables and function values	ALO	saALO	EALO	GA	PSO	SHALO
x_1	2.203	2.203	2.203	2.203	2.203	2.203
x_2	1.571	1.571	1.571	1.571	1.571	1.571
x_3	1.285	1.285	1.285	1.285	1.285	1.285
x_4	1.923	1.923	1.923	1.923	1.923	1.923
x_5	1.720	1.720	1.720	1.720	1.720	1.720
$f(x)$	− 4.688	− 4.688	− 4.688	− 4.688	− 4.688	− 4.688

that the proposed SHALO approach converges to the minimum mean objective function value compared to the other approaches used. For the best solutions in Table 3, the final decision variables and objective function values are compared in Table 4. As can be seen, all the identified variables are very close to 0 which is the global optimum solution for this problem.

5.1.3 Trefethen's test function

Trefethen's test function [52] is an unconstrained nonlinear multimodal test function. This function includes two decision variables ($d = 2$) and combines exponential, trigonometric, and arithmetic expressions in a single function. The mathematical form of the function together with the search domain is given in Eq. (30).

$$\begin{aligned}\mathrm{Min}f(x) = e^{\sin(50x_1)} + \sin(60e^{x_2}) + \sin(70sin(x_1))\\ + \sin(80\sin(x_2)) - \sin(10(x_1 + x_2))\\ + \frac{1}{4}\left({x_1}^2 + {x_2}^2\right)\end{aligned} \tag{30}$$

$$\text{s.t.} \quad -6.5 \leq x_1 \leq 6.5 \tag{30a}$$

$$-4.5 \leq x_2 \leq 4.5 \tag{30b}$$

The global optimum solution of this function is observed at $x^* = [-0.0244031\ 0.2106124]$ with a function value of $f(x^*) = -3.307$. Three-dimensional representation of the Trefethen's test function is given in Fig. 12. As can be seen, the function has many local oscillations which behave as the local optimum solutions. This example is also solved 100 times for different λ values. Results of the proposed SHALO approach in terms of the calculated SR values are given in Fig. 13 for different λ values.

As can be seen from Fig. 13, the solution with $\lambda = 1$ is selected as the best since the calculated SR value of that solution is maximum. For this solution, the results of the corresponding 100 model executions are compared with the results which are obtained by using ALO, saALO, EALO, GA, and PSO approaches, respectively. Similarly, all the approaches are also executed 100 times for the same random number seed data set. A comparison of the obtained results of all model executions is given in Table 5.

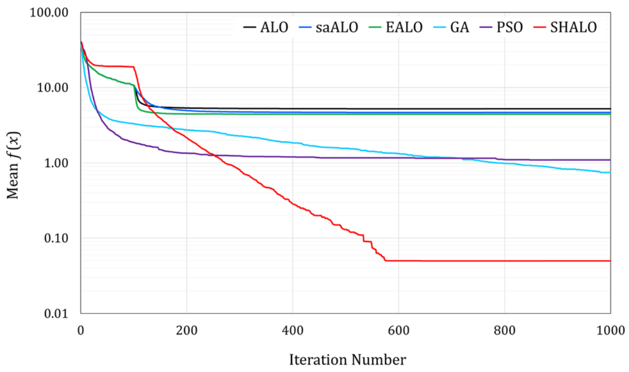

Fig. 11 Comparison of the convergence histories of different approaches for the Rastrigin's test function

As can be seen, all the approaches determined the global optimum solution of − 3.307 for the conducted 100 model executions. Results of the calculated SR values for ALO, saALO, EALO, GA, and PSO approaches are obtained as 47%, 56%, 20%, 36%, and 63%, respectively. For this function, the corresponding SR value in SHALO is obtained as 66% which is the best compared to the other approaches. The calculated mean AC values are obtained as 98% for EALO and 99% for the other approaches. Note that the Trefethen's test function has many local optimums around global optimum solution and this result can be clearly seen from Table 5 that the EALO-based optimization approach is resulted with a SR value of 20% although the calculated mean AC value of the same approach is 98%. Figure 14 shows the convergence plots of all approaches in terms of the mean objective function values. As can be seen, the proposed SHALO approach again converges to the minimum mean objective function value than the other approaches. For the best solutions given in Table 5, the final decision variables and objective function values are compared in Table 6. As can be seen, all the identified values are the same since all the approaches converged to the global optimum solution at least once.

5.1.4 De Villiers–Glasser 1 function

De Villiers–Glasser 1 test function is a continuous nonlinear test function which has a differentiable, non-separable, and non-scalable multimodal mathematical solution space [53]. The mathematical form of the function together with the search space is given in Eq. (31).

$$\mathrm{Min}f(x) = \sum_{i=1}^{24}\left[x_1 x_2^{t_i}\sin[x_3 t_i + x_4] - y_i\right]^2 \tag{31}$$

$$t_i = 0.1(i - 1) \tag{31a}$$

$$y_i = 60.137 \cdot 1.371^{t_i} \cdot \sin(3.112t_i + 1.761) \tag{31b}$$

$$\text{s.t. } 1 \leq x_k \leq 100 \quad k = 1, 2, \cdots, d \tag{31c}$$

This function has a global optimum solution at $x^* = [60.137\ 1.371\ 3.112\ 1.761]$ with a function value of $f(x^*) = 0$ for $d = 4$. Similarly, this problem is also solved for different λ values in SHALO; each of them consists of 100 model executions. Figure 15 shows the final SR values for these model executions.

As can be seen from Fig. 15, the maximum value of SR is obtained for the solution with $\lambda = 2000$. After selecting the best λ value, the results of the corresponding 100 model

Table 4 The final decision variables and objective function values for Rastrigin's test function for the best solutions in Table 3

Variables and function values	ALO	saALO	EALO	GA	PSO	SHALO
x_1	– 9.95E–01	3.68E–08	2.40E–08	1.47E–09	2.95E–09	– 2.53E–08
x_2	– 3.87E–08	2.06E–07	– 4.08E–08	2.36E–09	– 5.25E–10	2.37E–08
x_3	2.52E–07	– 1.66E–07	2.68E–08	– 3.53E–09	– 3.28E–09	– 1.69E–08
x_4	3.15–07	– 1.39E–07	2.89E–08	4.58E–10	2.29E–09	9.31E–09
x_5	– 3.42E–07	– 4.28E–08	– 1.99E–08	3.05E–09	– 1.79E–09	2.83E–08
$f(x)$	9.95E–01	1.83E–11	8.24E–13	0.00E + 00	0.00E + 00	4.62E–13

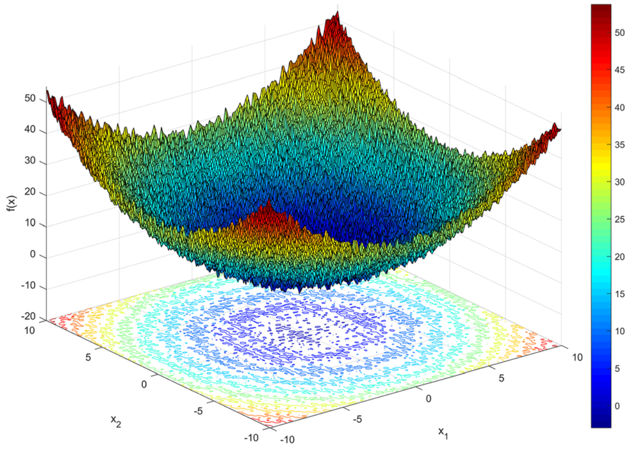

Fig. 12 Trefethen's test function

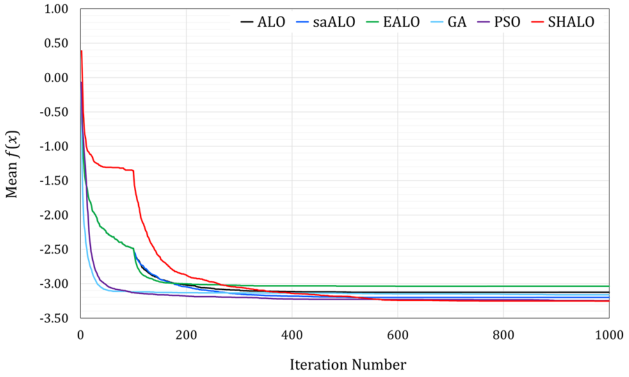

Fig. 14 Comparison of the convergence histories of different approaches for the Trefethen's test function

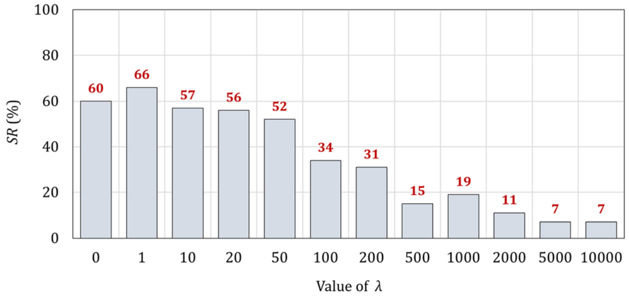

Fig. 13 Variation of *SR* (%) for different λ values for Trefethen's test function

executions are compared with the ones those obtained by using ALO, saALO, EALO, GA, and PSO approaches, respectively (Table 7). As can be seen from Table 7, the original ALO, saALO, EALO, and GA could not reach the global optimum in any of the 100 model executions. However, the problem is successively solved for 60 and 77 times out of 100 model executions by PSO and SHALO approaches, respectively. The calculated mean *AC* values are all obtained over 65% for all the approaches. Note that this function also has many local optimums around global optimum solution. For instance, while the mean value of *AC* is obtained as 68% for ALO, the corresponding value of *SR* is 0% for the same approach.

Table 5 Evaluation of the 100 model executions for Trefethen's test function for different optimization approaches

Measure	ALO	saALO	EALO	GA	PSO	SHALO
$SR(\%)$	47	56	20	36	63	66
$AC(\%)$	99	99	98	99	99	99
Best	– 3.307	– 3.307	– 3.307	– 3.307	– 3.307	– 3.307
Worst	– 2.643	– 2.643	– 2.428	– 2.729	– 3.063	– 2.761
Mean	– 3.126	– 3.201	– 3.039	– 3.163	– 3.248	– 3.254
Median	– 3.176	– 3.307	– 3.063	– 3.208	– 3.307	– 3.307
Std. deviation	0.221	0.159	0.215	0.146	0.088	0.090

Table 6 The final decision variables and objective function values for Trefethen's test function for the best solutions in Table 5

Variables and function values	ALO	saALO	EALO	GA	PSO	SHALO
x_1	− 0.024	− 0.024	− 0.024	− 0.024	− 0.024	− 0.024
x_2	0.211	0.211	0.211	0.211	0.211	0.211
$f(x)$	− 3.307	− 3.307	− 3.307	− 3.307	− 3.307	− 3.307

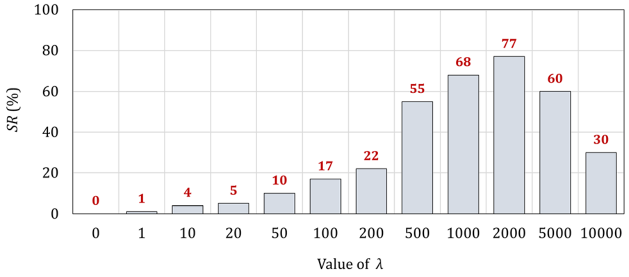

Fig. 15 Variation of *SR* (%) for different λ values for De Villiers–Glasser 1 test function

The convergence plots for ALO, saALO, EALO, GA, PSO, and the proposed SHALO approaches are compared in Fig. 16 for the mean objective function values of 100 model executions. Results of these plots demonstrate that the proposed SHALO approach again converges to the minimum mean objective function value compared to the other optimization approaches. For the best solutions in Table 7, the final decision variables and objective function values are given in Table 8. As can be seen, only PSO and SHALO approaches determined the global or near-global optimum solution and the other approaches could not determine it.

5.2 Constrained optimization applications

In this section, the performance of the proposed SHALO approach is evaluated by solving two constrained benchmark problems and two well-known engineering design problems. For these problems, general form of the constrained optimization model can be mathematically described as follows:

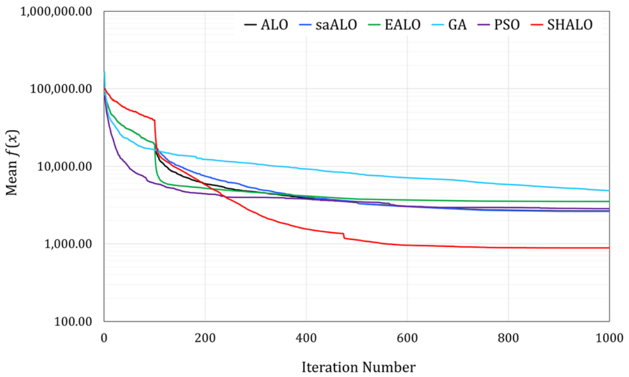

Fig. 16 Comparison of the convergence histories of different approaches for the De Villiers–Glasser 1 test function

$$\min f(x) \tag{32}$$

$$\text{s.t.} \quad g_i(x) \leq 0 \quad i = 1, 2, 3, \cdots, m \tag{32a}$$

$$h_j(x) = 0 \quad j = 1, 2, 3, \cdots, n \tag{32b}$$

where m is the number of inequality constraints; $g_i(x)$ is the i^{th} inequality constraint; n is the number of equality constraints; and $h_j(x)$ is the j^{th} equality constraint. Since ALO is a meta-heuristic optimization method, it can only be used for solving the unconstrained optimization problems in its original form. Therefore, solution of the constrained problems is conducted by converting them to the unconstrained problems by means of the penalty function approach as follows:

$$\min f'(x) = f(x) + \sum_{i=1}^{m} \alpha_i \cdot P_i^g + \sum_{j=1}^{n} \beta_j \cdot P_j^h \tag{33}$$

Table 7 Evaluation of the 100 model executions for De Villiers–Glasser 1 test function for different optimization approaches

Measure	ALO	saALO	EALO	GA	PSO	SHALO
$SR(\%)$	0	0	0	0	60	77
$AC(\%)$	68	69	68	58	65	74
Best	0.990	1.210	2.034	0.050	0.000	0.000
Worst	11,786.676	12,355.975	23,644.427	50,179.865	35,863.062	41,453.299
Mean	2649.741	2619.385	3528.969	4851.426	2845.219	886.577
Median	1713.426	1693.898	1374.254	138.214	0.000	0.000
Std. deviation	2840.333	3032.582	4974.243	10,817.759	7014.526	5822.534

Table 8 The final decision variables and objective function values for De Villiers–Glasser 1 test function for the best solutions in Table 7

Variables and function values	ALO	saALO	EALO	GA	PSO	SHALO
x_1	60.625	60.676	59.441	60.141	60.137	60.138
x_2	1.365	1.364	1.380	1.371	1.371	1.371
x_3	65.942	65.942	3.115	3.111	3.112	3.112
x_4	52.029	64.595	8.041	96.010	1.761	1.761
$f(x)$	0.990	1.210	2.034	0.050	0.000	0.000

$$P_i^g = \begin{cases} 0 & \text{if } g_i(x) \le 0 \\ [g_i(x)]^2 & \text{otherwise} \end{cases} \tag{34}$$

$$P_j^h = \begin{cases} 0 & \text{if } h_j(x) = 0 \\ [h_j(x)]^2 & \text{otherwise} \end{cases} \tag{35}$$

where $f'(x)$ is the transformed objective function; P_i^g and P_j^h are the penalty functions for inequality and equality constraints, respectively; α_i and β_j are the corresponding penalty coefficients for inequality and equality constraints, respectively. As can be seen from Eqs. (33) to (35), the given constrained problem is transformed to the unconstrained optimization problem with inclusion of penalty functions. These functions get a value of zero when all the constraints are satisfied, whereas they get a penalty value depending on constraint violation when the constraints are not satisfied. These penalty functions are integrated to the objective function by using the penalty coefficients of α_i and β_j which are used to adjust the magnitude of penalty terms. Note that selection of α_i and β_j values are the mostly problem dependent. Therefore, values of them are selected by conducting some trials before execution of the optimization model. Depending on these previous trials, values of penalty coefficients are taken as 1000 for all the problems.

5.2.1 Floudas and Pardalos's function

Floudas and Pardalos's function is a nonlinear constrained optimization test function which includes 13 decision variables and 9 inequality constraints [54]. The mathematical form of the objective and constraint functions together with the lower and upper bounds of the decision variables is given in Eq. (36).

$$\text{Min} f(x) = 5\sum_{k=1}^{4} x_k - 5\sum_{k=1}^{4} x_k^2 - \sum_{k=5}^{13} x_k \tag{36}$$

$$\text{s.t. } g_1(x) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0 \tag{36a}$$

$$g_2(x) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \le 0 \tag{36b}$$

$$g_3(x) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0 \tag{36c}$$

$$g_4(x) = -8x_1 + x_{10} \le 0 \tag{36d}$$

$$g_5(x) = -8x_2 + x_{11} \le 0 \tag{36e}$$

$$g_6(x) = -8x_3 + x_{12} \le 0 \tag{36f}$$

$$g_7(x) = -2x_4 - x_5 + x_{10} \le 0 \tag{36g}$$

$$g_8(x) = -2x_6 - x_7 + x_{11} \le 0 \tag{36h}$$

$$g_9(x) = -2x_8 - x_9 + x_{12} \le 0 \tag{36j}$$

$$0 \le x_k \le 1, \ k = 1, 2, \cdots, 9 \tag{36k}$$

$$0 \le x_k \le 100, \ k = 10, 11, 12 \tag{36l}$$

$$0 \le x_k \le 1, \ k = 13 \tag{36m}$$

Floudas and Pardalos's function has a global optimum solution at $x^* = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 3\ 3\ 3\ 1]$ with a function value of $f(x^*) = -15$. As conducted previously, this problem is also solved for different λ values in SHALO for 100 model executions. Figure 17 shows the variation of *SR* for different λ values.

As can be seen from Fig. 17, the maximum *SR* value is obtained as 62 for the solution with $\lambda = 100$. Therefore, this solution is selected to compare the performance of the proposed SHALO approach with the ALO, saALO, EALO, GA, and PSO approaches. Similarly, all the approaches are executed 100 times for the same random number seed data set with SHALO. Table 9 compares the identified results of 100 model executions using different approaches.

As can be seen from Table 9, the original ALO, saALO, and EALO approaches could not find the global optimum solution for any of the 100 model executions. On the other

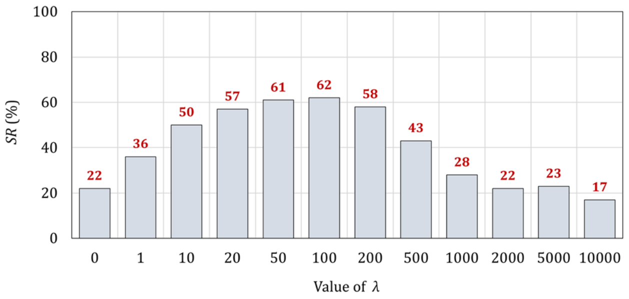

Fig. 17 Variation of *SR* (%) for different λ values for Floudas and Pardalos's test function

Table 9 Evaluation of the 100 model executions for Floudas and Pardalos's test function for different optimization approaches

Measure	ALO	saALO	EALO	GA	PSO	SHALO
$SR(\%)$	0	0	0	8	10	62
$AC(\%)$	58	59	58	90	88	97
Best	− 10.655	− 9.536	− 10.774	− 15.000	− 15.000	− 15.000
Worst	− 2.536	− 2.510	− 2.507	− 6.000	− 6.000	− 11.420
Mean	− 5.051	− 5.077	− 4.878	− 10.598	− 10.357	− 14.260
Median	− 4.922	− 4.867	− 4.698	− 11.413	− 9.000	− 14.970
Std. deviation	1.454	1.319	1.415	2.613	2.470	1.105

hand, GA, PSO, and the proposed SHALO approaches found the global optimum solution 8, 10, and 62 times, respectively. These results indicate that the proposed SHALO approach provides the best identification results in terms of the calculated *SR* measure. For the *AC* measure, while the original ALO and EALO approaches resulted with the mean *AC* value of 58%, this measure is obtained for the saALO approach as 59%. However, the GA, PSO, and the proposed SHALO approach provided the mean *AC* value of 90%, 88%, and 97%, respectively. When the results are evaluated statistically, the mean and the median values of the objective function are very close to the global optimum solution in SHALO compared to the other approaches in Table 9. The convergence plots of each approach are compared in Fig. 18 in terms of the mean objective function values of 100 model executions. It is seen from Fig. 18 that the proposed SHALO approach converges to the minimum mean objective function value compared to the other approaches.

For the best solutions in Table 9, the final decision variables, objective, and constraint function values for each approach are given in Table 10. As can be seen, GA, PSO, and the proposed SHALO approach determined the optimum decision variable values, while the original ALO, saALO, and EALO approaches could not fully determine them. When the results are evaluated in terms of the constraint violations, all the constraint functions are satisfied without any violation by each approach.

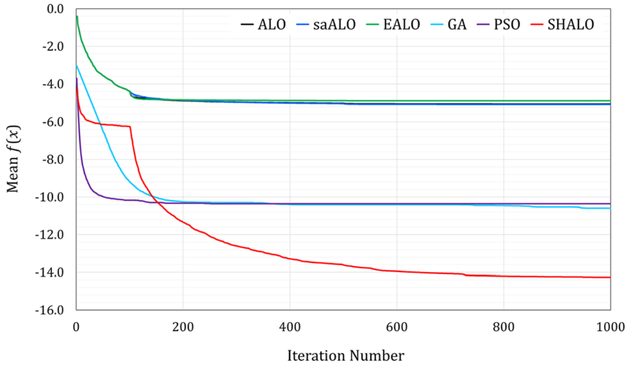

Fig. 18 Comparison of the convergence histories of different approaches for the Floudas and Pardalos's test function

5.2.2 Rastrigin's constrained function

Rastrigin's constrained function, proposed by Poole and Allen [49], is a modified version of Rastrigin's unconstrained test function. It includes a parameter controlling the number of global optimums, the Heaviside function, and a constrain function for transforming the original Rastrigin's function to a constrained one. The mathematical form of the objective and constraint functions together with the lower and upper bounds of the decision variables is given in Eq. (37).

$$\mathrm{Min} f(x) = \sum_{k=1}^{d} \left\{ 10(1+\cos(2\pi\mathcal{K}_k x_k)) + 2\mathcal{K}_k (x_k - 1)^2 H(x_k - 1) \right\} \tag{37}$$

$$\text{s.t. } g(x) = \sum_{k=1}^{d} 20\cos(4\pi\mathcal{K}_k x_k) \le 0 \tag{37a}$$

$$H(y) = \begin{cases} 1 & \text{if } \; y > 0 \\ 0 & \text{otherwise} \end{cases} \tag{37b}$$

$$0 \le x_k \le 2, \quad k = 1, 2, 3, \cdots, d \tag{37c}$$

where $H(y)$ is the Heaviside function; and $\mathcal{K}_k \in \boldsymbol{K}$ is a parameter which controls the number of global optimums for each search direction. Note that this modified problem has $2^d \prod_{k=1}^{d} \mathcal{K}_k$ global optimum solutions having the same objective function value of $f = 10d - 5d\sqrt{2}$ [49]. For $d = 2$, Fig. 19 shows 2-dimensional representation of the Rastrigin's constrained function. As can be seen, the function has 24 different locations where global optimum exists. For this case, the problem is solved for $d = 5$ decision variables and the corresponding $\mathcal{K}_k$ values for each variable are selected from the set of $\boldsymbol{K} = [1\ 1\ 1\ 1\ 2]$.

As conducted previously, the problem is solved 100 times for each λ values. Figure 20 shows the variation of the calculated *SR* measures for different λ values. Among the obtained results, the solution of $\lambda = 1$ is selected as the best solution since it has the highest *SR* value. After this

Table 10 The final decision variables, objective, and constraint function values for Floudas and Pardalos's test function for the best solutions in Table 9

Variables and function values	ALO	saALO	EALO	GA	PSO	SHALO
x_1	0.575	0.529	0.566	1.000	1.000	1.000
x_2	0.794	0.852	0.798	1.000	1.000	1.000
x_3	1.000	0.998	0.998	1.000	1.000	1.000
x_4	0.994	0.928	1.000	1.000	1.000	1.000
x_5	0.784	0.794	0.802	1.000	1.000	1.000
x_6	0.992	0.865	1.000	1.000	1.000	1.000
x_7	0.996	0.998	1.000	1.000	1.000	1.000
x_8	0.838	0.742	0.817	1.000	1.000	1.000
x_9	0.491	0.450	0.529	1.000	1.000	1.000
x_{10}	2.772	2.649	2.802	3.000	3.000	3.000
x_{11}	2.980	2.728	2.999	3.000	3.000	3.000
x_{12}	2.166	1.935	2.163	3.000	3.000	3.000
x_{13}	0.707	0.596	0.708	1.000	1.000	1.000
$g_1(x)$	− 1.510	− 1.860	− 1.470	0.000	0.000	0.000
$g_2(x)$	− 1.913	− 2.361	− 1.907	0.000	0.000	0.000
$g_3(x)$	− 1.266	− 1.636	− 1.247	0.000	0.000	0.000
$g_4(x)$	− 1.825	− 1.587	− 1.730	− 5.000	− 5.000	− 5.000
$g_5(x)$	− 3.374	− 4.092	− 3.384	− 5.000	− 5.000	− 5.000
$g_6(x)$	− 5.832	− 6.049	− 5.819	− 5.000	− 5.000	− 5.000
$g_7(x)$	0.000	0.000	0.000	0.000	0.000	0.000
$g_8(x)$	0.000	0.000	0.000	0.000	0.000	0.000
$g_9(x)$	0.000	0.000	0.000	0.000	0.000	0.000
$f(x)$	− 10.655	− 9.536	− 10.774	− 15.000	− 15.000	− 15.000

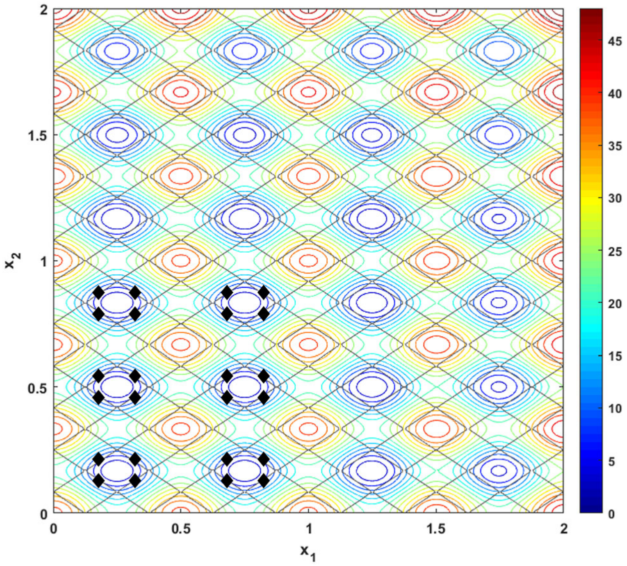

Fig. 19 Two-dimensional representation of the Rastrigin's constrained function (the black diamonds represent the locations of the global optimum solution)

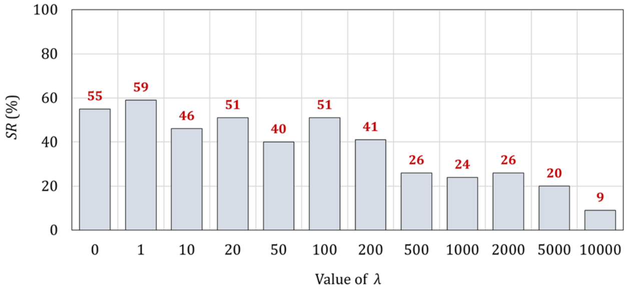

Fig. 20 Variation of *SR* (%) for different λ values for Rastrigin's constrained test function

process, the solution corresponding to the selected λ value compared with the original ALO, saALO, EALO, GA, and PSO approaches. Similarly, each approach is executed 100 times with the same random number seed data set for comparing the identified results. The results of this comparison are given in Table 11.

As can be seen from Table 11, the calculated *SR* values of the original ALO, saALO, and EALO approaches are obtained as 30%, 37%, and 30%, respectively. For the same conditions, while the GA found the global optimum solution with a *SR* value of 43%, this value is obtained as 58% and 59% for PSO and the proposed SHALO approaches, respectively. As indicated previously, the number of global optimum solutions is given as $2^d \prod_{k=1}^{d} \mathcal{K}_k$. By considering $d = 5$ decision variables and $\mathcal{K}_k \in [1\ 1\ 1\ 1\ 2]$,

Table 11 Evaluation of the 100 model executions for Rastrigin's constrained test function for different optimization approaches

Measure	ALO	saALO	EALO	GA	PSO	SHALO
SR(%)	30	37	30	43	58	59
Best	14.645	14.645	14.645	14.650	14.645	14.645
Worst	17.573	17.572	17.580	16.036	16.113	16.248
Mean	15.408	15.330	15.442	15.027	15.007	15.002
Median	15.273	15.203	15.347	14.969	14.927	14.924
Std. deviation	0.608	0.619	0.640	0.263	0.318	0.286

the number of global optimum solutions having the same objective function value (14.645) is calculated as 64. Therefore, the measure of *AC* is not calculated for this example since there is no single optimum solution. The convergence plots for each approach in terms of the mean objective function values are compared in Fig. 21.

It can be seen from Fig. 21 that the proposed SHALO approach converges to the minimum mean objective function values than the other approaches. For the best solutions in Table 11, the final decision variables, objective, and constraint function values are compared in Table 12. As can be seen, different decision variables are obtained by each approach although their corresponding objective function values are the same. This is an expected outcome since the modified problem has 64 different global optimum solutions.

5.2.3 Pressure vessel design problem

The pressure vessel design problem is a popular constrained engineering design problem which is suggested by Kannan and Kramer [55]. The purpose of the problem is to minimize the total cost including the material, forming, and welding costs. A general schematic view of the considered pressure vessel is given in Fig. 22. There are four design variables in this problem which are T_s (shell thickness), T_h (head thickness), R (inner radius), and L (length of the cylindrical section). These variables are denoted as x_1 to x_4, respectively. Note that among these design variables, T_s and T_h get discrete values which consist of the integer multipliers of 0.0625 inch, whereas R and L are continuous variables. The mathematical form of the design problem is given as follows:

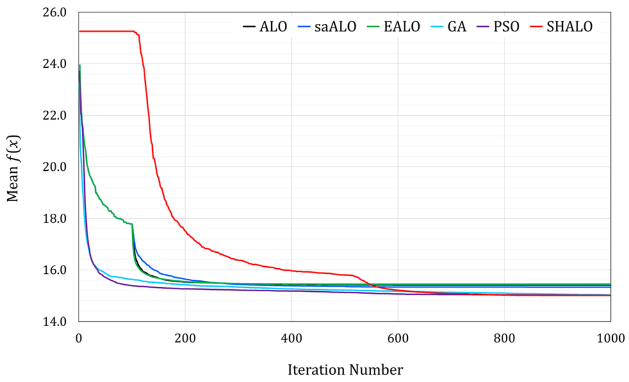

Fig. 21 Comparison of the convergence histories of different approaches for the Rastrigin's constrained test function

$$\min f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{38}$$

$$\text{s.t. } g_1(x) = -x_1 + 0.0193x_3 \leq 0 \tag{38a}$$

$$g_2(x) = -x_2 + 0.00954x_3 \leq 0 \tag{38b}$$

$$g_3(x) = -\pi x_3^2x_4 - 1.3333\pi x_3^3 + 1296000 \leq 0 \tag{38c}$$

$$g_4(x) = x_4 - 240 \leq 0 \tag{38d}$$

$$0.0625 \leq x_1, x_2 \leq 99 \times 0.0625 \tag{38e}$$

$$10 \leq x_3, x_4 \leq 200 \tag{38f}$$

Note that Yang et al. [56] proved that this function has a near-global optimum solution at $x^* = [0.8125\ 0.4375\ 42.0984\ 176.6366]$ with a function value of $f(x^*) = 6059.7143$. This engineering design problem is also executed 100 times for each λ values for the proposed SHALO approach. Figure 23 compares the final calculated *SR* values for these model executions. As can be seen, the maximum *SR* value is obtained as 95% and this result is observed for the solution with $\lambda = 1$. This solution is again compared with the original ALO, saALO, EALO, GA, and PSO approaches which are also executed 100 times with the same random number seed data set. The identified results of each approach are compared in Table 13.

As can be seen from Table 13, the *SR* values of the ALO, saALO, EALO approaches are obtained as 84%, 88%, and 52%, respectively. On the other hand, the same measure is obtained in GA, PSO, and the proposed SHALO approach as 36%, 84%, and 95%, respectively. The calculated mean *AC* measures are obtained over 87% for each approach. The convergence plots of each approach for the mean objective function values are compared in Fig. 24.

As can be seen from Fig. 24, the proposed SHALO approach converges to a lower mean objective function value than the other approaches. For the best solutions in Table 13, Table 14 compares the final decision variables,

Table 12 The final decision variables, objective, and constraint function values for Rastrigin's constrained test function for the best solutions in Table 11

Variables and function values	ALO	saALO	EALO	GA	PSO	SHALO
x_1	0.625	0.626	0.625	0.378	0.375	0.375
x_2	0.625	0.625	0.375	0.624	0.375	0.375
x_3	0.375	0.625	0.625	0.623	0.375	0.375
x_4	0.625	0.625	0.375	0.373	0.625	0.625
x_5	0.187	0.188	0.187	0.685	0.313	0.687
$g(x)$	0.000	0.000	0.000	0.000	0.000	0.000
$f(x)$	14.645	14.645	14.645	14.650	14.645	14.645

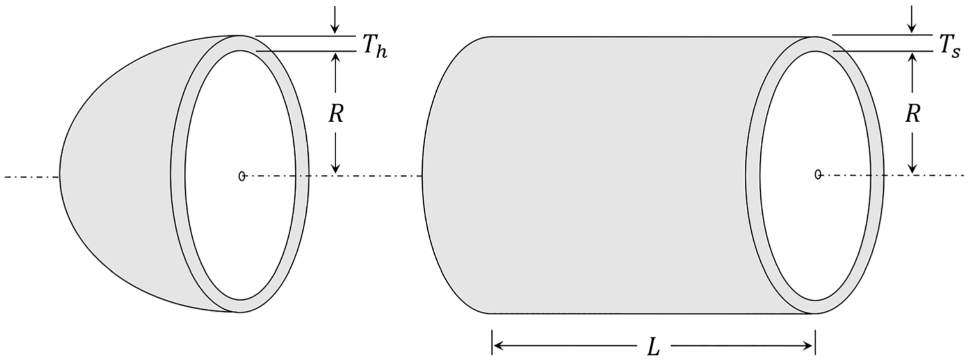

Fig. 22 The pressure vessel design problem

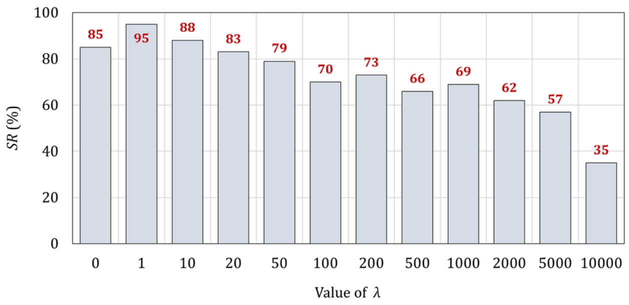

Fig. 23 Variation of SR (%) for different λ values for pressure vessel design problem

objective, and constraint function values for each approach. As can be seen, all the identified design variables are close to each other, and the constraints are satisfied without any violation for all the approaches.

5.2.4 Trapezoidal channel design problem

Design of the trapezoidal composite channels is an important problem in water resources engineering. The reason for using these channels is to transmit the water over long distances for various purposes including irrigation, flood control, or water supply, etc. Therefore, a small amount of reduction on their dimensions corresponds to the significant savings in the project budget. In this context, this problem is considered as an important engineering design problem and examined by several researchers in the literature [57–60]. Figure 25 shows the schematic view of the considered trapezoidal channel cross section. The design variables of this problem are the bed width (b), the flow depth (y), and the slopes of the left and right faces of the channel (z_1 and z_2). These variables are denoted as x_1 to x_4, respectively.

Table 13 Evaluation of the 100 model executions for pressure vessel design problem for different optimization approaches

Measure	ALO	saALO	EALO	GA	PSO	SHALO
$SR(\%)$	84	88	52	36	84	95
$AC(\%)$	91	92	91	87	93	95
Best	6059.714	6059.714	6059.714	6059.769	6059.714	6059.714
Worst	7332.842	7332.842	10,841.412	7551.971	7353.978	6820.422
Mean	6307.377	6288.707	6687.230	6685.965	6277.594	6174.185
Median	6090.527	6090.527	6410.087	6566.208	6090.526	6090.528
Std. deviation	326.329	317.147	796.002	457.941	338.994	198.342

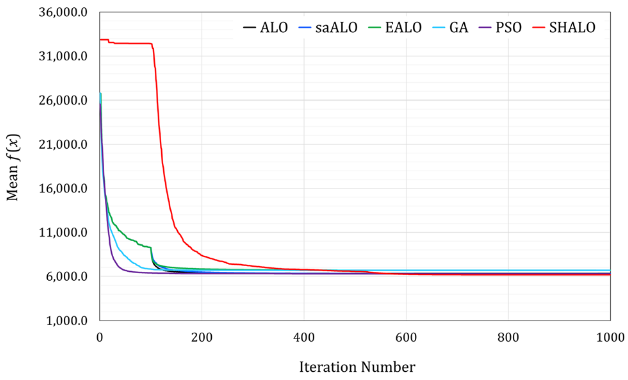

Fig. 24 Comparison of the convergence histories of different approaches for the pressure vessel design problem

where T_t is the top width of the cross section; T_w is the top width of flow section, f is the freeboard, n_1,, n_2, and n_3 are the Manning's surface roughness coefficient values for the left and right faces, and the bed of the channel, respectively. The objective of the channel design problem is to minimize the cost function which can be written in mathematical form as follows:

$$\min f(x) = c_1 A_t + c_2 P_1 + c_3 P_2 + c_4 P_3 \tag{39}$$

$$\text{s.t. } g(x) = \left| \frac{Qn_e}{\sqrt{S_0}} - \frac{A_w^{5/3}}{P_w^{2/3}} \right| - \varepsilon \le 0 \tag{39a}$$

where c_1 is the cost of excavation per unit cross-sectional area for a unit length of the channel; c_2, c_3, and c_4 are the costs of the lining per unit length of the perimeter of left-side face, right-side face (including the freeboard), and channel bed, respectively; A_t is the total cross-sectional area of the channel; P_1, P_2, and P_3 are perimeters of left and right side faces (including the freeboard), and channel bed, respectively; ε is the allowable threshold value; Q is the flow rate in the channel; S_0 is the bed slope; A_w is the wetted flow area; P_w is the wetted perimeter; and n_e is the equivalent Manning's surface roughness coefficient. The mathematical forms of these parameters are given in Eqs. (39b)–(39l) [60].

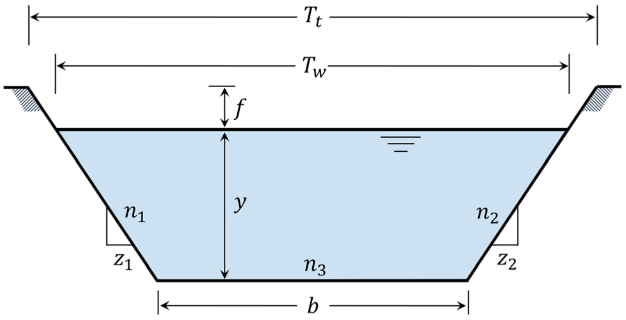

Fig. 25 Composite channel design problem: trapezoidal cross section

$$A_w = by + (z_1 + z_2)\frac{y^2}{2} \tag{39b}$$

$$P_w = \left\{ \left[\left(z_1^2 + 1\right)^{1/2} + \left(z_2^2 + 1\right)^{1/2} \right] y + b \right\} \tag{39c}$$

$$T_w = b + (z_1 + z_2)y \tag{39d}$$

$$n_e = \left[\frac{\left(\sqrt{z_1^2 + 1} \cdot n_1^{3/2} + \sqrt{z_2^2 + 1} \cdot n_2^{3/2} \right) y + b \cdot n_3^{3/2}}{P_w} \right]^{2/3} \tag{39e}$$

$$A_t = b(y + f) + (z_1 + z_2)\frac{(y + f)^2}{2} \tag{39f}$$

$$P_t = \left\{ \left[\left(z_1^2 + 1\right)^{1/2} + \left(z_2^2 + 1\right)^{1/2} \right] (y + f) + b \right\} \tag{39g}$$

$$P_1 = \sqrt{\left(z_1^2 + 1\right)}(y + f) \tag{39h}$$

$$P_2 = \sqrt{\left(z_2^2 + 1\right)}(y + f) \tag{39k}$$

$$P_3 = b \tag{39l}$$

Similarly, this constrained design problem is also executed 100 times for each different λ values by considering the previously generated random number seeds. During

Table 14 The final decision variables, objective, and constraint function values for pressure vessel design problem for the best solutions in Table 13

Variables and function values	ALO	saALO	EALO	GA	PSO	SHALO
x_1	0.813	0.813	0.813	0.813	0.813	0.813
x_2	0.438	0.438	0.438	0.438	0.438	0.438
x_3	42.098	42.098	42.098	42.098	42.098	42.098
x_4	176.637	176.637	176.637	176.642	176.637	176.637
$g_1(x)$	0.000	0.000	– 0.033	0.000	0.000	0.000
$g_2(x)$	– 0.036	– 0.036	– 0.021	– 0.036	– 0.036	– 0.036
$g_3(x)$	– 0.011	– 0.012	– 0.001	0.000	0.000	– 0.007
$g_4(x)$	– 63.363	– 63.363	– 81.500	– 63.358	– 63.363	-63.363
$f(x)$	6059.714	6059.714	6059.714	6059.769	6059.714	6059.714

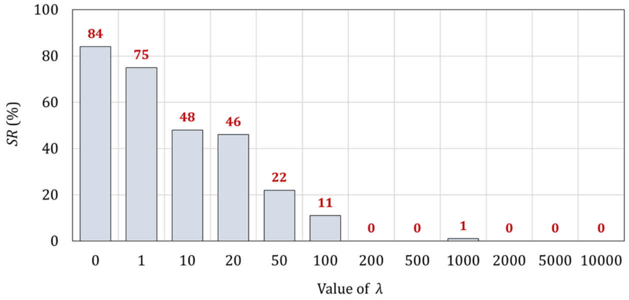

Fig. 26 Variation of *SR* (%) for different λ values for composite channel design problem

these solutions, the constants which are required to solve the channel design problem are taken as: $Q = 100, f = 0.5$, $S_0 = 0.0016$, $n_1 = 0.018$, $n_2 = 0.020$, $n_3 = 0.015$, $c_1 = 0.60$, $c_2 = 0.25$, $c_3 = 0.20$, $c_4 = 0.30$, and $\varepsilon = 0.001$. Figure 26 compares the determined *SR* values for these model executions. As can be seen, the best solution is obtained for $\lambda = 0$ which gives a *SR* value of 84%. By considering this solution, the performance of SHALO is also compared by solving the same problem with the original ALO, saALO, EALO, GA, and PSO approaches for the same conditions. Table 15 compares the statistical results of these model executions for each approach.

As can be seen from Table 15, the *SR* values of the original ALO and saALO are obtained as 13%, while the same measure in EALO is obtained as 3%. On the other hand, the GA, PSO, and the proposed SHALO approaches solved the problem with the *SR* values of 4%, 46%, and 84%, respectively. This outcome can also be seen from the statistical results in Table 15 that the obtained standard deviation value in SHALO is lower than the ones obtained by using the other approaches. For the *AC* values, the original ALO, saALO, EALO, and GA approaches resulted with the mean *AC* values of 79%, 80%, 75%, 84%, respectively. On the other hand, the same measure is obtained for PSO and SHALO as 93% and 96%, respectively. The convergence plots for ALO, saALO, EALO, GA, PSO, and SHALO approaches are compared in Fig. 27 for the mean objective function values of the 100 model executions. It can be seen from Fig. 27 that the proposed SHALO approach converges to a lower mean objective function value than the other approaches. For the best solutions in Table 15, the final decision variables, objective, and constraint function values are given in Table 16. As can be seen, although there are some differences in the identified design variables, they do not produce significant variations in the calculated objective function values. Furthermore, all the solutions are obtained without any constraint violations.

6 Conclusions

In this study, the SHuffled Ant Lion Optimization (SHALO) approach is proposed. The main contribution of this approach is to improve the random walk around antlions process of ALO by means of two new modifications. As the first modification, a new exponentially weighted approach is proposed for the boundary shrinking procedure to increase the random search capability of the algorithm. The second modification deals with increasing the diversity of the random walk vector by means of the proposed shuffling approach. The performance of the proposed SHALO approach is evaluated by using four unconstrained and two constrained benchmark problems, and two constrained engineering design problems. These problems are solved 100 times for performance evaluation of the proposed approach in terms of different random number seed values. Furthermore, the performance of the proposed boundary shrinking procedure is evaluated for 12 different cases in terms of different magnitudes of random perturbations. All the identified results are also compared with the ones which are obtained by using the original ALO, two modified versions of ALO which are the saALO and EALO, GA, and PSO approaches. Although the overall results of the proposed SHALO approach are quite effective, the following critical issues need to be taken into account when solving an optimization problem:

The main critical point of ALO-based approaches is the necessity of completing the given number of iterations to obtain a solution. This is important since the boundary shrinking procedure in random walk around antlions

Table 15 Evaluation of the 100 model executions for composite channel design problem for different optimization approaches

Measure	ALO	saALO	EALO	GA	PSO	SHALO
SR(%)	13	13	3	4	46	84
AC(%)	79	80	75	84	93	96
Best	22.961	22.959	22.984	22.968	22.958	22.958
Worst	29.379	28.697	29.666	27.702	24.400	23.993
Mean	24.652	24.511	25.111	23.975	23.202	23.041
Median	24.221	24.117	24.852	23.854	23.080	22.968
Std. deviation	1.494	1.370	1.518	0.717	0.290	0.182

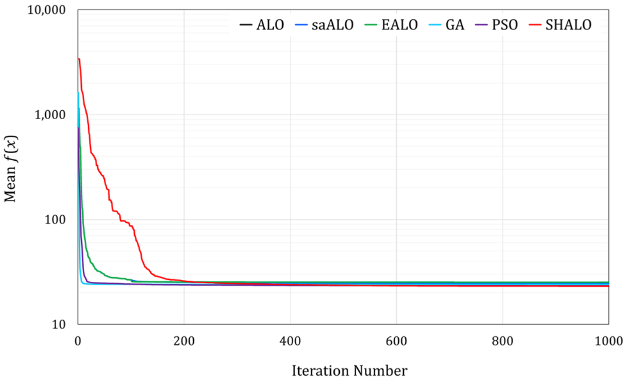

Fig. 27 Comparison of the convergence histories of different approaches for the composite channel design problem

process requires of specifying the dynamic change points (e.g., $0.10T$, $0.50T$, $0.75T$, $0.90T$, and $0.95T$ given in Fig. 2) for the iteration domain. Therefore, the optimization process requires to proceed until satisfying the maximum number of iterations. The proposed SHALO approach also uses this solution strategy and requires solving the problem until reaching the maximum iteration number without considering any other termination criterion. This solution strategy may increase the required CPU times, especially for the real-world engineering optimization problems.

One of the important contributions of this study is the newly proposed exponentially weighted boundary shrinking approach. Note that in original ALO, the boundary shrinking procedure is conducted by suddenly changing the value of I at some stationary points (Fig. 2). These stationary points may decrease the performance of the random search feature of the algorithm. This issue is eliminated by means of the proposed exponentially weighted boundary shrinking approach such a way that the value of I is continuously and exponentially increased between given minimum and maximum points for each region in Fig. 2. Note that these minimum and maximum I values are used as the same with the original ALO to make an exact comparison of the approaches. It may be possible to achieve better results for different minimum and maximum I values for each region in Fig. 2. However, this issue is beyond the scope of this work and can be examined in different studies.

The other issue of the proposed exponentially weighted boundary shrinking procedure is to control the magnitudes of the random perturbations. For this purpose, a weight parameter of λ is used such a way that big values of this parameter increase the random perturbation in the calculated value of I. Since there is no any systematic way of determining the λ values, the performance of the proposed SHALO approach should be evaluated for different values of λ as conducted in this study.

Another contribution of the proposed SHALO approach is to shuffle the elements of the random walk vector. This process is conducted by modifying the min–max normalization equation as given between Eqs. (20) and (22). This modification significantly increases the diversity of the random walk vector and improves the local optima problem by altering the magnitudes and trends of random walks as indicated in Fig. 4. Note that these changes are particularly significant in early iterations (while shuffling in early iterations, the proposed SHALO approach may return NaN values for some variable values and this problem is handled by randomly generating those variable values in the feasible search space). In this context, with the proposed approach, it is possible to explore the search space more comprehensively at the beginning of the optimization.

The performance of the proposed SHALO approach is compared with the original ALO, saALO, and EALO approaches by solving each problem for the same random number seed values. Therefore, all the problems are solved by considering the same initial populations and sequence of the random numbers. This outcome can be clearly seen from the convergence plots such that the original ALO, saALO, and EALO approaches start the search process from the same function value since the initial population of them are the same. Although the same initial populations are used, the proposed SHALO approach starts the search process from different function values. This result is associated with the shuffled random walk vector which influences the function values at the start of the search process.

Table 16 The final decision variables, objective, and constraint function values for composite channel design problem for the best solutions in Table 15

Variables and function values	ALO	saALO	EALO	GA	PSO	SHALO
x_1	5.942	5.787	6.039	6.020	5.854	5.839
x_2	4.041	4.049	4.026	3.906	4.048	4.047
x_3	0.258	0.289	0.297	0.286	0.263	0.262
x_4	0.209	0.244	0.138	0.262	0.239	0.248
$g(x)$	0.000	0.000	0.000	0.000	0.000	0.000
$f(x)$	22.961	22.958	22.959	22.984	22.968	22.958

The performance of the SHALO approach is also evaluated by solving the same problems on GA and PSO approaches. During these solutions, built-in GA and PSO optimization modules of the MATLAB platform are used. Since the solutions in ALO, saALO, EALO, and SHALO approaches are conducted by considering the population number of 50 and the maximum number of iterations of 1000, the same values are also used for both GA and PSO to make an exact comparison in terms of the total number of function evaluations. However, different values of these parameters may provide better results for GA and PSO which is out of scope of this work. Furthermore, although the same random number seeds are also used, both GA and PSO approaches start the search process from different initial points since the solution sequence of the MATLAB's built-in modules are different.

The reason for solving all the problems 100 times is to evaluate the performance of the SHALO approach for different random number seeds. This is an important analysis to evaluate the robustness of the approach. In this context, the outcome of these 100 model executions is evaluated by using different statistical metrics (minimum, maximum, mean, median, and standard deviation). Furthermore, all the solutions are evaluated in terms of the success rate (SR) and accuracy (AC) metrics. These model evaluations indicate that the proposed SHALO approach statistically provides better results than all the other approaches. The same outcome is also observed for the other measures that the mean SR values of the solved problems are obtained as 22% for ALO, 25% for saALO, 14% for EALO, 28% for GA, and 49% for PSO. On the other hand, this mean SR value is obtained as 76%for the proposed SHALO which is significantly better than the other approaches. Similarly, the mean AC values for each solved example are obtained as: 83% for ALO and EALO, 84% for saALO, 88% for GA, and 91% for PSO. For the same measure, the proposed SHALO approach is resulted with a mean AC value of 91%. These results indicate that use of the SHALO optimization approach can increase the rate of reaching the global optimum solutions for the problems solved in this work.

Acknowledgements The first author would like to thank the 100/2000 Program of the Turkish Council of Higher Education Institution (YÖK) and BİDEB 2211-A PhD Student Scholarship Program of The Scientific and Technological Research Council of Turkey (TÜBİTAK).

Author contribution PGD and MBT helped in software and writing—review & editing. MTA contributed to methodology, software, writing—review and editing.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). This study is supported by The Scientific Research Projects Division (PAÜBAP) at Pamukkale University, Turkey, under the project number 2022FEBE011.

Availability of data and materials Data are available on request from the authors.

Declarations

Conflict of interest No potential conflict of interest was reported by the authors.

Ethical approval This article does not contain any studies with human participants performed by any of the authors.

References

1. Ayvaz MT, Kayhan AH, Ceylan H, Gurarslan G (2009) Hybridizing the harmony search algorithm with a spreadsheet 'Solver' for solving continuous engineering optimization problems. Eng Optim 41:1119–1144. https://doi.org/10.1080/03052150902926835
2. Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194:3902–3933. https://doi.org/10.1016/J.CMA.2004.09.007
3. Boussaïd I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. Inf Sci 237:82–117. https://doi.org/10.1016/J.INS.2013.02.041
4. Abualigah L, Diabat A (2020) A comprehensive survey of the Grasshopper optimization algorithm: results, variants, and applications. Neural Comput Appl 32:15533–15556. https://doi.org/10.1007/S00521-020-04789-8/TABLES/4
5. Abualigah L, Shehab M, Alshinwan M et al (2021) Ant lion optimizer: a comprehensive survey of its variants and applications. Arch Comput Methods Eng 28:1397–1416. https://doi.org/10.1007/s11831-020-09420-6
6. Abualigah L (2020) Multi-verse optimizer algorithm: a comprehensive survey of its results, variants, and applications. Neural Comput Appl 32:12381–12401. https://doi.org/10.1007/s00521-020-04839-1
7. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98. https://doi.org/10.1016/J.ADVENGSOFT.2015.01.010
8. Holland JH (1975) Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor, MI
9. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1:28–39. https://doi.org/10.1109/MCI.2006.329691

10. Kennedy J, Eberhart R (1995) New optimizer using particle swarm theory. Proc Int Symp Micro Mach Hum Sci. https://doi.org/10.1109/MHS.1995.494215
11. Storn R, Price K (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359. https://doi.org/10.1023/A:1008202821328
12. Emary E, Zawbaa HM, Hassanien AE (2016) Binary ant lion approaches for feature selection. Neurocomputing 213:54–65. https://doi.org/10.1016/J.NEUCOM.2016.03.101
13. Wang M, Wu C, Wang L et al (2019) A feature selection approach for hyperspectral image based on modified ant lion optimizer. Knowl-Based Syst 168:39–48. https://doi.org/10.1016/J.KNOSYS.2018.12.031
14. Zawbaa HM, Emary E, Grosan C (2016) Feature selection via chaotic antlion optimization. PLoS ONE 11:e0150652
15. Emary E, Zawbaa HM (2019) Feature selection via Lèvy Antlion optimization. Pattern Anal Appl 22:857–876. https://doi.org/10.1007/S10044-018-0695-2/TABLES/10
16. Majhi SK, Biswal S (2018) Optimal cluster analysis using hybrid K-Means and ant lion optimizer. Karbala Int J Mod Sci 4:347–360. https://doi.org/10.1016/J.KIJOMS.2018.09.001
17. Chen J, Qi X, Chen L et al (2020) Quantum-inspired ant lion optimized hybrid k-means for cluster analysis and intrusion detection. Knowledge-Based Syst 203:106167. https://doi.org/10.1016/J.KNOSYS.2020.106167
18. Yogarajan G, Revathi T (2018) Improved cluster based data gathering using ant lion optimization in wireless sensor networks. Wirel Pers Commun 98:2711–2731. https://doi.org/10.1007/S11277-017-4996-3/FIGURES/14
19. Yamany W, Tharwat A, Hassanin MF, et al. (2016) A new multilayer perceptrons trainer based on ant lion optimization algorithm. Proc - 2015 4th Int Conf Inf Sci Ind Appl ISI 2015 40–45. https://doi.org/10.1109/ISI.2015.9
20. Banadkooki FB, Ehteram M, Ahmed AN et al (2020) Suspended sediment load prediction using artificial neural network and ant lion optimization algorithm. Environ Sci Pollut Res 27:38094–38116. https://doi.org/10.1007/S11356-020-09876-W/FIGURES/13
21. Kose U (2018) An ant-lion optimizer-trained artificial neural network system for chaotic electroencephalogram (EEG) prediction. Appl Sci 1613(8):1613. https://doi.org/10.3390/APP8091613
22. Singh D, Singh B, Kaur M (2020) Simultaneous feature weighting and parameter determination of neural networks using ant lion optimization for the classification of breast cancer. Biocybern Biomed Eng 40:337–351. https://doi.org/10.1016/J.BBE.2019.12.004
23. Ali ES, Abd Elazim SM, Abdelaziz AY (2017) Ant Lion Optimization Algorithm for optimal location and sizing of renewable distributed generations. Renew Energy 101:1311–1324. https://doi.org/10.1016/J.RENENE.2016.09.023
24. Ali ES, Abd Elazim SM, Abdelaziz AY (2016) Ant lion optimization algorithm for renewable distributed generations. Energy 116:445–458. https://doi.org/10.1016/J.ENERGY.2016.09.104
25. Dubey HM, Pandit M, Panigrahi BK (2016) Ant lion optimization for short-term wind integrated hydrothermal power generation scheduling. Int J Electr Power Energy Syst 83:158–174. https://doi.org/10.1016/J.IJEPES.2016.03.057
26. Chandrasekaran GK, KumarChandrasekaran PV (2021) Test scheduling of system-on-chip using dragonfly and ant lion optimization algorithms. J Intell Fuzzy Syst 40:4905–4917. https://doi.org/10.3233/JIFS-201691
27. Samui P, Yesilyurt SN, Dalkilic HY et al (2022) Comparison of different optimized machine learning algorithms for daily river flow forecasting. Earth Sci Inform. https://doi.org/10.1007/s12145-022-00896-3
28. Mahendru NM, Mehta S (2015) Optimal load dispatch using ant lion optimization. Int J Eng Res Appl 5:10–19
29. Saxena P, Kothari A (2016) Ant Lion Optimization algorithm to control side lobe level and null depths in linear antenna arrays. AEU - Int J Electron Commun 70:1339–1349. https://doi.org/10.1016/J.AEUE.2016.07.008
30. Dinkar SK, Deep K (2017) Opposition based laplacian ant lion optimizer. J Comput Sci 23:71–90. https://doi.org/10.1016/J.JOCS.2017.10.007
31. Guo M, Wang J-S, Zhu L et al (2020) Improved ant lion optimizer based on spiral complex path searching patterns. IEEE Access 8:22094–22126. https://doi.org/10.1109/ACCESS.2020.2968943
32. Chen S-C, Huang W-C, Hsueh M-H et al (2022) A novel exponential-weighted method of the antlion optimization algorithm for improving the convergence rate. Processes 10:1413. https://doi.org/10.3390/PR10071413
33. Yao P, Wang H (2017) Dynamic adaptive ant lion optimizer applied to route planning for unmanned aerial vehicle. Soft Comput 21:5475–5488. https://doi.org/10.1007/S00500-016-2138-6/FIGURES/10
34. Rajan A, Jeevan K, Malakar T (2017) Weighted elitism based ant lion optimizer to solve optimum VAr planning problem. Appl Soft Comput 55:352–370. https://doi.org/10.1016/J.ASOC.2017.02.010
35. Wu Z, Yu D, Kang X (2017) Parameter identification of photovoltaic cell model based on improved ant lion optimizer. Energy Convers Manag 151:107–115. https://doi.org/10.1016/J.ENCONMAN.2017.08.088
36. Kılıç H, Yüzgeç U (2019) Improved antlion optimization algorithm via tournament selection and its application to parallel machine scheduling. Comput Ind Eng 132:166–186. https://doi.org/10.1016/J.CIE.2019.04.029
37. Yang D, Miao J, Zhang F et al (2019) Bearing fault diagnosis using a support vector machine optimized by an ımproved ant lion optimizer. Shock Vib. https://doi.org/10.1155/2019/9303676
38. Toz M (2019) An improved form of the ant lion optimization algorithm for image clustering problems. Turkish J Electr Eng Comput Sci 27:1445–1460. https://doi.org/10.3906/elk-1703-240
39. Kilic H, Yuzgec U, Karakuzu C (2020) A novel improved antlion optimizer algorithm and its comparative performance. Neural Comput Appl 32:3803–3824. https://doi.org/10.1007/S00521-018-3871-9/TABLES/8
40. Liu J, Huo Y, Li Y (2020) Preferred strategy based self-adaptive ant lion optimization algorithm. Pattern Recognit Artif Intell 33:121–132
41. Chaitanya SNVSK, Rao BV, Bakkiyaraj RA (2021) Solution of an optimal reactive power dispatch problem: an application of modified ant lion optimizer. In: Proceedings of 2021 31st Australasian universities power engineering conference, AUPEC 2021. Institute of Electrical and Electronics Engineers Inc.
42. Singh D, Singh B (2021) Effective and efficient classification of gastrointestinal lesions: combining data preprocessing, feature weighting, and improved ant lion optimization. J Ambient Intell Humaniz Comput 12:8683–8698. https://doi.org/10.1007/S12652-020-02629-0/TABLES/8
43. Yao Y, Li Y, Xie D et al (2021) Coverage enhancement strategy for WSNs based on virtual force-directed ant lion optimization algorithm. IEEE Sens J 21:19611–19622. https://doi.org/10.1109/JSEN.2021.3091619
44. El Bakrawy LM, Akif Cifci M, Kausar S et al (2022) A modified ant lion optimization method and its application for instance reduction problem in balanced and imbalanced data. Axioms 11:95. https://doi.org/10.3390/AXIOMS11030095

45. Yan R, Lin Y, Yu N, Wu Y (2023) A low-carbon economic dispatch model for electricity market with wind power based on improved ant-lion optimisation algorithm. CAAI Trans Intell Technol 8:29–39. https://doi.org/10.1049/CIT2.12138
46. Li Z, Ouyang J (2023) Optimal design method for sub-array beamforming based on an improved ant lion algorithm. J Phys Conf Ser 2437:012106. https://doi.org/10.1088/1742-6596/2437/1/012106
47. Shen H, Liu S (2023) Optimization of branch pipe routing considering tee constraint ant lion. IEEE Access 120270–120280. https://doi.org/10.1109/ACCESS.2023.3324853
48. Lu H, Huang L, Xie Y et al (2023) Prediction of fractional flow reserve with enhanced ant lion optimized support vector machine. Heliyon. https://doi.org/10.1016/j.heliyon.2023.e18832
49. Poole DJ, Allen CB (2019) Constrained niching using differential evolution. Swarm Evol Comput 44:74–100. https://doi.org/10.1016/J.SWEVO.2018.11.004
50. Michalewicz Z (1992) Genetic Algorithms + Data Structures = Evolution Programs. SpringerVerlag Berlin
51. Rastrigin LA (1974) Systems of Extreme Control. Nauka
52. Adorio EP, Dilman UP (2005) MVF-Multivariate Test Function Library İn C For Unconstrained Global Optimization Methods
53. de Villiers N, Glasser D (1981) A continuation method for nonlinear regression. SIAM J Numer Anal 18:1139–1154. https://doi.org/10.1137/0718079
54. Floudas CA, Pardalos PM (1990) A collection of test problems for constrained global optimization algorithms. Springer, Berlin, Heidelberg
55. Kannan BK, Kramer SN (1994) An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 116:405–411
56. Yang XS, Huyck C, Karamanoglu M, Khan N (2013) True global optimality of the pressure vessel design problem: a benchmark for bio-inspired optimisation algorithms. Int J Bio-Inspired Comput 5:329–335. https://doi.org/10.1504/IJBIC.2013.058910
57. Kaveh A, Talatahari S (2010) An improved ant colony optimization for constrained engineering design problems. Eng Comput 27:155–182. https://doi.org/10.1108/02644401011008577/FULL/PDF
58. Das A (2000) Optimal channel cross section with composite roughness. J Irrig Drain Eng 126:68–72. https://doi.org/10.1061/(ASCE)0733-9437(2000)126:1(68)
59. Jain A, Bhattacharjya RK, Sanaga S (2004) Optimal design of composite channels using genetic algorithm. J Irrig Drain Eng 130:286–295. https://doi.org/10.1061/(ASCE)0733-9437(2004)130:4(286)
60. Subramanya K (2015) Flow in open channels (ISBN: 9353166292). McGraw-Hill, India

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