International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Eagle Strategy Based on Modified Barnacles Mating Optimization and Differential Evolution Algorithms for Solving Transient Heat Conduction Problems

Mert Sinan Turgut*[1], Oguz Emrah Turgut[2]

Submitted: 05/05/2021 Accepted : 13/09/2021

Abstract: *Solving time-dependent heat conduction problems through a conventional solution procedure of iterative root-finding method may sometimes cause difficulties in obtaining accurate temperature distribution across the heat transfer medium. Analytical root-finding methods require good initial estimates for finding exact solutions, however locating these promising regions is some kind of a black-box process. One possible answer to this problem is to convert the root-finding equation into an optimization problem, which eliminates the exhaustive process of determining the correct initial guess. This study proposes an Eagle Strategy optimization framework based on modified mutation equations of Barnacles Mating Optimizer and Differential Evolution algorithm for solving one-dimensional transient heat conduction problems. A test suite of forty optimization benchmark problems have been solved by the proposed algorithm and the respective solution outcomes have been compared with those found by the reputed literature optimizers. Finally, a case study associated with a transient heat conduction problem have been solved. Results show that Eagle strategy can provide efficient and feasible results for various types of solution domains.*

Keywords: *Barnacles Mating Optimizer, Constrained optimization, Eagle Strategy, Heat conduction, Hybrid algorithms*

1. Introduction

Nature-inspired algorithms have been an active research area for decades not only for researchers developing dexterous algorithm-based solution strategies but also for practitioners and engineers working on finding efficient methods for solving real-world design problems. Swam-Intelligence and population-based methods consist of a huge part of the nature-inspired algorithms, whose main inspiration is drawn from natural events or behaviors. Firefly Algorithm [1], Cuckoo Search [2], Particle Swarm Optimization [3] and some recently proposed optimizers [4-6] are prominent examples of these methods. They have significant and deliberate advantages over conventional deterministic methods, therefore their feasible applications reach almost all fields ranging from electric load dispatch problems [7] to PEM fuel cell parameter estimation cases [8]. Swarm-based algorithms use some kind of nondeterministic stochastic methods, which are built upon mathematical models simulating random-walks. A proper balance between the effects of deterministic components and stochastic random walks surprisingly evolves into a successful metaheuristic algorithm as there are plenty of instances evident in the literature [9-11].

Metaheuristic algorithms are known for their efficiency and effectiveness in solving challenging optimization algorithms. Most metaheuristics can probe the search space to finding the optimum solution to the problem however, obtaining the global optimum point may not be guaranteed due to the stochastic nature of the algorithms. They do not require extensive problem information to be solved and effectively search large domains of trial solutions with relatively minimum computational effort. Success in finding the optimal solution depends on the structural concept on which the natural paradigm is imitated. For instance, Particle Swarm Optimizer mimics the flocking behaviors of birds and fishes obeying a simple mathematical formulation describing the spatial movements of the particles to iteratively improve the candidate trial solutions until the termination criterion is satisfied. Differential evolution utilizes algorithm-specific genetic operators to create a mutated solution by manipulating the current solution. Then, a sample candidate vector with the best fitness value obtained after successive iterations is selected as the global answer to the problem. Harmony search [12] is another metaheuristic inspired by a musician who aims to find perfect harmony in his or her composition. A physics-based metaheuristic called Charged System Search [13] imitates the governing rules of the Columb and Newtonian laws of physics. Biogeography based Optimization applies the mathematical foundations of biogeographical concept to form an iterative solution procedure devised to solve optimization problems [14].

Metaheuristics can be considered as a high-level problem-solving strategy responsible for selecting sub-level heuristics that can maintain sufficiently possible answers to a related optimization problem, particularly for cases wherein domain information is insufficient or computation capacity is limited [15]. According to the No Free Lunch theorem [16], there is a clear statement that the performance of any two algorithms is equivalent when their solution accuracies are averaged to all optimization problems. That

[1] *Department of Mechanical Engineering, Faculty of Engineering, Ege University, Bornova, Izmir, Turkey, ORCID ID : 0000-0002-5739-2119*
[2] *Department of Industrial Engineering, Faculty of Engineering and Architecture, Izmir Bakircay University, Menemen, Izmir, Turkey ORCID ID : 0000-0003-3556-8889*
** Corresponding Author Email: sinanturgut@me.com*

is, an optimization algorithm may converge to the optimal solution of a problem but may fail in another case. To conquer these drawback, different researchers propose different metaheuristics showing some specific features and capabilities to provide feasible candidates those having the ability to improve the previous solutions. Furthermore, an algorithm should avoid quick convergence raised from the entrapment of the local optimum points. A successful algorithm should also diversify the candidate solutions as much as possible to explore the unvisited paths on the search domain. Recent researches on algorithm development focus on these two important algorithm-specific features those need to be deeply analyzed and investigated. The literature on metaheuristics has been crowded with persistent claims of novelty and superiority over the precedent methods. Although there have been developed high-quality researches featuring a promising future for the upcoming studies, most of the raised claims have been unproven and unverified due to the lack of conceptual elaboration and insufficient numerical experiments made on the proposed algorithms [17].

A successful metaheuristic should combine the favorable merits of exploration and exploitation, those of which perform in a harmony to obtain accurate solutions. As mentioned above, an efficient algorithm should be able to probe the most important search regions where the global solution may reside in. An algorithm should also be capable to jump out of the local optimum points which eliminate stagnation in the search process and avoid premature convergence. These two capabilities respectively show the exploration and exploitation performance of a metaheuristic optimizer. A good synergy between them possibly increases the solution accuracy, however, there is no credible theoretical knowledge or framework in literature as to how to create a plausible balance between these two terms [18]. Despite lack of information, empirical knowledge and extensive observations and evaluations on literature studies suggest that the convergence speed of an optimization algorithm increases with too much emphasis on exploitation while too much exploration reducing the convergence rate.

Most of the literature considers the hybridization of different algorithms to maintain a dexterous interplay between the exploration and exploitation phases of an algorithm. Hybrid algorithms are constructed in such a way that each constituent method maximize their advantages while eliminating algorithm-specific deficiencies. Singh et al. [19] hybridized the Salp Swarm Algorithm (SSA) with Sine-Cosine Algorithm to enhance the search capacity of those optimizers. In the concept of this hybridization, SSA is performed as a global explorer while SCA is used as a local exploiter. Numerical results of the engineering design problems obtained from the proposed hybrid reveal that global convergence is greatly boosted up through this created synergy. A hybrid algorithm composed of Firefly Algorithm and Particle Swarm Optimization was proposed by Aydilek [20]. The main aim behind this hybridization was to benefit from the strong points of these two algorithms to get rid of their intrinsic algorithmic deficiencies. Son et al. [21] proposed using hybrid adaptive Differential Evolution and Jaya algorithm for extracting unknown parameters of Bouc-Wen hysteresis model. A reputed mutation scheme of DE/rand/1 is concurrently operated with manipulation equations of the Jaya algorithm to equalize the workload between intensification and diversification. Emperor Penguin Optimization is hybridized with Salp Swarm Algorithm to enhance the exploration capacity of both methods. A multi-population solution strategy was proposed in which valuable information is exchanged in each iteration to reach the global answer to the problem [22]. Because of the incompetencies of the Biogeography-based algorithm in dealing with a complex real-world optimization problem, this algorithm was hybridized with Shuffled Frog Leaping Algorithm. Two novel contributions are proposed, including a new migration operator and an improved mutation scheme is included to ameliorate the probing mechanism of the hybrid method [23].

This study aims to take advantage of the merits of Eagle Strategy (ES), which was developed by Yang and Deb [24] to be utilized as a practical framework for dealing with nonlinear optimization problems. Eagle Strategy will be investigated in detail by combining modified mutation schemes of Differential Evolution [25] and Barnacles Mating Optimization [26]. Within this framework, updated manipulation equations of Differential Evolution will perform as local search agents responsible for exploiting the promising regions while Levy flight [2] and Lozi chaotic map [27] enriched modified mutation scheme of Barnacles Mating Optimizer dealing with the exploration of unvisited paths of the search domain. Eagle strategy in terms of optimization framework has been applied to many literature optimizers including Differential Evolution [28,29], Particle Swarm Opimization [30,31], and Jaya Algorithm [32] to enhance probing efficiency of the algorithm through the iteratively exchanged population information between the hybridized algorithms. Several novelties are also evident in this study. One is that this is the first application of Barnacles Mating Optimizer on a hybrid optimization algorithm. Another contribution to the literature is to create a synergy between these optimizers to form a two-stage based solution procedure, which has not been accomplished in the previous studies yet in any type of hybridization method. Optimization performance of Eagle strategy-based hybridization is verified on forty benchmark functions consisting of unimodal and multimodal test problems. Finally, a transient heat conduction problem will be solved by the proposed Eagle Strategy based algorithm to eliminate the deficiencies raised from the traditional root-finding methods, many of which may sometimes lack from misleading initial estimates to find the exact roots of the nonlinear equations.

2. Eagle Strategy

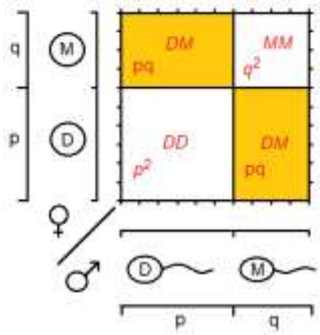
Eagle Strategy is an optimization framework, not an algorithm, aimed to combine at least two different optimization algorithms. It is a nature-inspired solution procedure mimicking the two-stage hunting strategy of eagles. In nature, eagles experience a two-stage foraging strategy including a roaming stage and a chasing stage. In the roaming stage, an eagle searches a large domain of space to observe prey. Once the prey is observed, the chasing stage takes place and the eagle swiftly shifts into chasing action to catch the sighted prey as soon as possible. These consecutive actions can be conceptualized into a two-stage optimization framework in which the first stage dealing with the exploration of the whole search domain while the latter stages are devoted to fine-tuning the promising solution obtained in the preceding actions. The main idea behind this kind of hybridization is to maintain a compromise set of actions to balance between the global and local search mechanisms. One can easily apply any type of optimization algorithm in different stages of the successive iterations. Therefore, it will be more advantageous to benefit from the merits of each algorithm at different phases within the proposed framework. In the original study in which Eagle Strategy was firstly coined, it was proposed that global search is maintained employing the random walks governed by Levy flights [33], then the promising solutions are updated by local search algorithm of Firefly Optimization. However, efficient local optimizers of simplex search and hill-climbing methods can be alternatively applied in the latter stages to polish the so-far-best-obtained solution in the course of iterations. The optimization success of these local search methods highly depends on the quality of the initial estimates. Consecutive stages of the algorithm switch on and off according to the quality of the iterative candidate solutions. The success of the Eagle Strategy based optimization framework relies on the dexterity of the global and local search mechanisms, respectively responsible for diversifying the sample solutions as much as possible and exploiting the fertile regions in the search domain. The global explorer in the framework should produce enough randomness to effectively diversify the trial solutions and reach the unvisited paths in the search domain. This relatively slow processing search mechanism accelerates as iterations proceed and the solution gradually converges. Moreover, the local optimizer should efficiently avoid the local optimum points within the minimum number of function evaluations to speed up convergence to the global optimum solution.

In this study, it is considered to take advantage of the optimization capabilities of the modified and amended mutation schemes of Barnacles-Mating Optimizer (BMO) and Differential Evolution (DE) algorithms. In the first stage, the required randomization is maintained by Levy flight and chaotic random numbers enriched probing equation of BMO to scatter the population individuals across the search domain. In the second stage, modified DE variants of DE/rand/1 and DE/best/1 are ensembled together to form an effective local optimizer, whose mutation equations are amended by the chaotic random nuımbers generated by Lozi map and onezero() function which produces one or zero and applies this number to a population member. The next section provides the essentials of Barnacles-Mating Optimizer and Differential Evolution algorithms and their mutual hybridization to construct an efficient optimization framework based on the foundations of Eagle Strategy.

2.1. Barnacles-Mating Optimizer

Inspired by the characteristic mating process of barnacles, this method simulates the Hardy-Weinberg [34] principles to generate versatile off-springs in the concept of BMO. Barnacles are hermaphroditic species, which explains that they have both male and female reproductions. A typical feature that distinguishes this living being from the other types is that they are known for their relatively large penises compared to their body size, which is seven or eight times larger to cope with their sedentary life-style mostly taking place nearby the rocks and corals [35]. A barnacle searches for a partner within the reach of its penis to copulate and secretes a sperm into the mantle cavity of its partner. For barnacles living in solitude, sperm-cast mating takes place where the released sperm floating the water is captured by the partner to fertilize its eggs. BMO algorithm is built upon these mating behavior to solve challenging optimization problems. Off-spring generation is occurred obeying the main principles of Hardy-Weinberg. Consider two alleles *D* and *M* standing for Dad and Mum with selection frequencies respectively $f(D)=p$ and $f(M)=q$. Under these circumstances, the expected genotype frequencies under normal mating conditions can be expressed as follows: genotype frequency for *DD* homozygotes is $f(DD)=p^2$, genotype frequency for *MM* homozygotes is $f(MM)=q^2$, genotype frequency for *DM* homozygotes is $f(DM)=2pq$. Schematical representation of forming genotypes for the next off-spring generation is illustrated in Figure 1. Rectangular areas that represent the genotype frequencies in Figure 1 show that the sum of each frequency is $p^2+2pq+q^2=1$ so that $p+q=1$. This equation explains the governing mechanism behind off-spring generation, which is mainly based on the genotype frequencies *p* and *q* of parents.

Fig. 1. Schematic view of Punnet square



The algorithm is initialized with populating the candidate barnacles' vectors in the form of the below-given matrix representation

$$X = \begin{bmatrix} x_1^1 & \cdots & x_1^D \\ \vdots & \ddots & \vdots \\ x_N^1 & \cdots & x_N^D \end{bmatrix} \tag{1}$$

Where *X* is the barnacles' population, *N* represents the size of the population and *D* is the number of decision variables of the considered optimization problem. After the initialization process, all population individuals are evaluated and sorted based on their corresponding fitness value where the best individual stays on the top while the worst one is located at the bottom. This algorithm aims to give a balance to intersification and diversification phases phases relying on the procedure visually explained in Figure 2. Barnacles population composed of ten individuals is represented in the matrix form in Figure 2. In the BMO algorithm concept, the best solution is located at the top of the population as was mentioned before. Consider that the penis size of the barnacles is seven times larger than its body length, which means that penis length (*pl*) is equal to seven. This implies that barnacle #1 can mate with barnacles #2 to barnacles#7 in the current iteration. If barnacle#7 opts to select barnacles #9 for mating, then it will exceed the predefined limit set by the penis length. So sperm-cast mating is occurred rather than normal mating to generate an offspring population. Below given equations explain how a simple selection mechanism works to produce candidate solutions

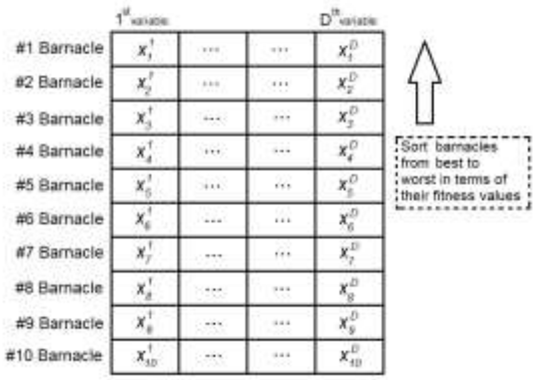


Fig. 2. BMO mating mechanism

$$D_{barnacle} = randperm(X) \tag{2}$$
$$M_{barnacle} = randperm(X) \tag{3}$$

Where *randperm*() shuffles the members of the barnacle's population (X) to create Dad ($D_{barnacle}$) and Mum ($M_{barnacle}$) to be mated to generate off-springs. Generating trial solutions in this algorithm is a bit different than the other evolutionary algorithms in the literature. Since there is not defined specific formula or set of equations to be used for mutation, the algorithm relies on genotype frequencies of the barnacles to generate offsprings. The following equations are responsible for producing candidate barnacle off-springs

$$x_i^{D_{new}} = p \cdot D_{barnacle}^{D} + q \cdot M_{barnacle}^{D} \quad if \;\; k \leq pl \tag{4}$$

$$x_i^{D_{new}} = M_{barnacle}^{D} \times rand(0,1) \qquad if \;\; k > pl \tag{5}$$

Where p is a uniform random number defined in the range [0,1] and q=1-p; $D_{barnacle}$ and $M_{barnacle}$ respectively stand for the vector symbolizing dad and mum of the generated off-spring; k is the absolute value of the difference between $D_{barnacle}$ and $M_{barnacle}$ vectors expressed by the following equation:

$$k = |D_{barnacle} - M_{barnacle}| \tag{6}$$

The inheritance behavior of the parents is controlled by the numerical values of the genotype frequencies. For example, let p = 0.65 which implies that newly generated off-spring respectively inherits %65 and %35 of characteristic features of dad's and mum's. BMO initializes the barnacles' population randomly within the prescribed search boundaries and sorts the population individuals according to their corresponding fitness values in such a way that the best barnacles staying on top while the worst ones are located at the bottom. Equations (4) and (5) shuffle the barnacles' population by changing their respective row elements to derive a new mutated dad and mum vectors. New off-springs are produced based on Eq. (6) and Eq. (7). Exploration and exploitation which are two main features of BMO controlling the solution diversity and efficiency are maintained by these two equations. Eq. (6) enables the algorithm to diversify the search space by using the merits of Gaussian distributed random numbers and mutation vectors of parents. Eq. (7) exploits the promising regions where the best barnacles in the population reside in. This equation governs the local search mechanism guided by the penis length (*pl*) parameter which decides the number of neighboring populations to be mated for the current iteration.

2.2. Differential Evolution

Differential Evolution (DE) is a population based evolutionary algorithm. This algorithm draws significant interest from the research community because of its outstanding performance in solving real-world complex optimization problems from different domains. DE was firstly conceptualized by the technical report completed by Storn [36] then followed by the base study [25] in which all different aspects of the proposed algorithm were discussed. Some of the past literature approaches [37,38] reached a consensus that DE is considered to be a powerful and competitive member of population-based metaheuristics. DE applies three different operators to obtain the global solution of the optimization problem it deals with. These are mutation, crossover, and selection mechanisms that playing important role in sampling trial solutions. At each iteration, these three operators are executed independently. After initializing population members within the predefined upper and lower bounds, iterative solutions are obtained by commencing the mutation phase in which mutant solutions are produced by using mutation operators. Then, trial solution vectors are sampled utilizing a crossover operator allowing a cooperative interaction between the mutated vector and its related target vector. Finally, the selection mechanism makes a comparison between the trial solution and target solution to maintain the survival of the fittest vector. An individual with having a fitter solution among these two vectors is going to be used for the upcoming iteration. The next section will explain these algorithm steps in details:

D-dimensional N-sized population members are represented as $M_i^G = \{m_{i,1}^G, m_{i,2}^G, \ldots, m_{i,D}^G\}$ where i=1,2,3,….,N symbolizes the population size; D refers to the dimensionality of the problem and G denotes the current generation. Based on the target vector $M_{i,j}^G$ to be manipulated, a mutant vector can be produced employing the below given six well-known mutation strategies:

$$\text{"rand/1"} \rightarrow Y_i^G = M_{n_1}^G + F \times (M_{n_2}^G - M_{n_3}^G) \tag{7}$$

$$\text{"rand/2"} \rightarrow Y_i^G = M_{n_1}^G + F \times (M_{n_2}^G - M_{n_3}^G) + F \times (M_{n_4}^G - M_{n_5}^G) \tag{8}$$

$$\text{"best/1"} \rightarrow Y_i^G = M_{best}^G + F \times (M_{n_1}^G - M_{n_2}^G) \tag{9}$$

$$\text{best/2} \rightarrow Y_i^G = M_{best}^G + F \times (M_{n_1}^G - M_{n_2}^G) + F \times (M_{n_3}^G - M_{n_4}^G) \tag{10}$$

$$\text{"current} - \text{to} - \text{best/1"} \rightarrow Y_i^G = M_i^G + F \times (M_{best}^G - M_{n_1}^G) + F \times (M_{n_2}^G - M_{n_3}^G) \tag{11}$$

$$\text{"current} - \text{to} - \text{pbest/1"} \rightarrow Y_i^G = M_i^G + F \times (M_{best}^G - M_i^G) + F \times (M_{r_1}^G - M_{r_2}^G) \tag{12}$$

In the above equations, randomization is occurred by n_1, n_2, n_3, n_4, and n_5 which are integers between [1, N] and are also different from the index (i) of the current population member. F is the scale factor whose numerical value varies from 0 to 1 responsible for controlling the relative

magnitude between the difference vector. M_{best}^G is the best solution in the population obtained during consecutive iterations. After sampled solutions are mutated, a crossover operation is put into practice to decide between the target vector M_i^G and mutated vector Y_i^G to generate a set of trial solutions $Z_i^G = \{z_{i,1}^G, z_{i,2}^G, \ldots, z_{i,D}^G\}$. With using the binomial crossover operator, trial solutions are produced as defined in the following procedure

$$z_{i,j}^G = \begin{cases} y_{i,j}^G & if(rand(0,1) \leq CR \;\; or \;\; j = j_{rand}) \\ m_{i,j}^G & else \end{cases} \tag{13}$$

Where $CR \in [0,1]$ is a crossover parameter that controls the number of individuals to be copied from the target vector M_i^G and Y_i^G to the trial vector Z_i^G ; *rand(0,1)* is a uniform random number defined in the range [0,1]; j_{rand} is an integer between 1 to D, which ensures at least one individual in trial vector Z_i^G should be different from the target vector M_i^G. Once the trial vector is produced, a selection mechanism is employed to select the fitter individuals among the trial vector Z_i^G and the target vector M_i^G. The selection mechanism is performed by the following equation

$$M_{i,j}^{G+1} = \begin{cases} M_{i,j}^G & if \;\; f(M_i^G) \leq f(Z_i^G) \\ Z_{i,j}^G & \text{otherwise} \end{cases} \tag{14}$$

2.3. The Proposed Method

The Eagle Strategy framework composed of modified search equations of Barnacles Mating Optimizer (BMO) and Differential Evolution (DE). According to the base paper discussing the merits of BMO, it was emphasized that this algorithm can balance the exploration and exploitation phases efficiently. However, we mainly aim to benefit from the diversification ability of BMO reinforced by the random numbers generated by the Levy Flights and Lozi chaotic map. Exploitation in which fertile areas of the search region are finetuned will be executed by an ensemble of modified mutation strategies of DE. Utilizing different mutation strategies for a single population has been an issue among the metaheuristic community for a while. With such a combination of different mutation equations, the algorithm will have the chance to explore undiscovered paths or valleys on the search domain more efficiently and tenaciously compared to the solution procedure in which only one mutation strategy is concerned with. During the iterative process, different DE variants used in the ensemble become powerful and supportive as they collaborate by sharing the valuable domain information using their distinctive capabilities. Literature studies concerning the applications of DE variants into a single framework show the importance to use different DE operators in one algorithm. Qin et al. [39] proposed a solution strategy based on a self-adaptive DE in which different mutation schemes of ensemble algorithms are dynamically selected relying on their success in previous iterations. In another research paper, Gong et al. [40] adopted an ensemble of DE variants using an adaptive strategy selection which involves two different methods including Probability Matching and Adaptive Pursuit. These two intelligent techniques are autonomously employed by each individual in the population to determine the most suitable strategy decided by their comparative influence on the optimization problem. Ali et al. [41] aimed to boost up the solution diversity by subdividing the entire community into independent sub-populations and apply different mutation strategies to each individual in the subdivided group. This multi-population approach uses the domain information of the best solution.

The proposed method is composed of two cascaded stages as considered in the original paper propounded by Yang and Deb [24]. Crude global exploration is accomplished by the modified search equations of BMO, which has been experienced to be an efficient explorer based on the extensive numerical experiments made on unimodal and multimodal testbeds by the authors. Here, in this case, Levy flight based random number generation is used instead of a uniform random number controlled by the genotype frequency for BMO. Literature comprises plenty of Levy flight strengthened metaheuristic algorithms whose successful engineering applications ranging from NOX emission prediction of a boiler [42] to optimal design of steel space frames [43]. This wide diversity in the utilization of Levy flights highly depends on its enormous capability to escape local optimum points as well as faster solution convergence, both actions result from the imposed intrinsic probing mechanism. The search mechanism reinforced by the random numbers generated by Levy flights provides beneficial behaviors to the base algorithm such that it allows minimizing the possibility to visit previously explored spaces in the solution domain. Therefore, this mechanism enables the algorithm to avoid stagnation in the fitness land space by circumventing any local optima in the search space thanks to the intermingled long and short jumps across the search domain that occurred by the Levy flights. This type of flight pattern takes advantage of reaching distant regions of the solution space without revisiting the already explored spaces, thereby eliminating a huge amount of computational burden that would be resulted from the excessive number of function evaluations [44]. One can see the impacts of Levy Flights even in the dispersion of gas molecules and movement of particles in fluids when turbulent flow conditions are prevalent

Generation of random numbers from Levy distribution consists of two constructive steps. The first step is concerned with deciding the randomized direction of Levy Flight based on uniform distribution and the second step is related to producing sequential steps drawn from the chosen Levy distribution. Mantegna's algorithm [31] is considered in this study to draw a random number from symmetric Levy stable distribution as formulated below:

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)\cdot\beta\cdot 2^{\left(\frac{\beta-1}{2}\right)}} \right\}^{\frac{1}{\beta}} \tag{15}$$

Where Γ is the gamma function and β parameter varies between 0 and 2. In this study, this β parameter is taken as 1.5 [24]. Based on the numerical value σ_u obtained from Eq. (15) and $\sigma_v = 1$, step length *s* can be computed by the following equation

$$s = \frac{u}{|v|^{1/\beta}} \tag{16}$$

Where u and v are random numbers respectively following a Gaussian distribution of zero mean and deviation σ_u^2 and σ_v^2.

The modified search equation of BMO considered in this study is reconstructed by merging Eq.(4) and Eq.(5) and utilizing a chaotic random number generator by Lozi map rather than a uniformly distributed random number varying in the range between 0 and 1. Eleven different chaotic maps are tested for random number generation and the Lozi map gives the best solution outcomes, therefore this map is considered in this phase of the algorithm. The proposed modified manipulation equation responsible for the exploration phase of the algorithm takes the final form given in Algorithm 1.

Algorithm 1 – Exploration phase
$X_1 = randperm(X)$
$X_2 = randperm(X_1)$
$for\ i\ =\ 1$ to N
$for\ j\ =\ 1$ to D
$X_{new}^{i,j} = \left(X_1^{i,j} \times levy_1^{i,j} + X_2^{i,j} \times levy_2^{i,j}\right) \times \left(2 \cdot \varphi_1^{i,j} - 1\right)$
end
end

In the above-defined Algorithm 1, *randperm()* function shuffles the row elements of the given input matrix; *X* represents *N*-sized *D*-dimensional population individuals to be mutated for upcoming generations; $levy_1$ and $levy_2$ are two different random numbers drawn from Levy distribution calculated in terms of step size as given Eq.(16); φ is a chaotic random number generated by Lozi map. Lozi chaotic map [45] is a two-dimensional piecewise linear map whose representative dynamical equations are similar to that of Henon map [46] and formulated as given below:

$$\begin{aligned} x(t+1) &= 1 - a \cdot |x(t)| + y(t) \\ y(t+1) &= b \cdot x(t) \end{aligned} \tag{17}$$

Where the parameters considered in the above equation *a* and *b* are respectively 1.7 and 0.5 as suggested in Caponetto et al [45]. Chaotic sequences generated by the Lozi map are normalized in the numerical range between 0 and 1 with the below-given procedure

$$y_{norm}^i = \frac{y^i - \min(y)}{\max(y) - \min(y)} \tag{18}$$

Where y^i stands for the i^{th} element in the chaotic sequence; max(y) and min(y) are respectively the maximum and minimum valued element in the sequence. The exploitation phase of the algorithm is constructed by the ensemble of two well-known mutation strategies of DE including “DE/rand/1” and “DE/best/1” with some modifications on mutative equations. Proposed modifications have been made based on the trial-and-error procedure. Plenty of combinatorial interactions of DE variants have been evaluated and benchmarked against different types of optimization problems involving challenging real-world design problems as well as unimodal and multi-modal conventional test problems for the exploitation phase. The below-defined procedure described in Algorithm 2 yields the most favorable solution outcomes among the contestant alternative mutation schemes and proposed for intensifying on the promising candidates obtained in the exploration phase

Algorithm 2 – Exploitation phase

X1 = randperm(*X*) ; *X2 = randperm*(*X1*) ;

X3 = randperm(*X2*) ; *X4 = randperm*(*X3*)

oz1=onezero()

oz2=onezero()

$for\ i = 1$ to N

$for\ j = 1$ to D

$if(rand(0,1)_1 < rand(0,1)_2)$

$$X_{new}^{i,j} = X^{i,j} + (2\varphi_2^{i,j} - 1) \cdot oz_1^{i,j} \cdot (X_1^{i,j} - X_2^{i,j})$$

else

$$X_{new}^{i,j} = X_{best}^{j} + oz_2^{i,j} \cdot (X_3^{i,j}\varphi_3^{i,j} - X_4^{i,j}\varphi_4^{i,j})$$

end

end

end

Where rand $(0,1)_1$ and rand$(0,1)_2$ are random numbers different from each other generated by uniform distribution and Xbest is the best individual in the population obtained so far. Function onezero() produces an integer either 0 or 1 and assigns this number to a random member of the population. This kind of assignment increases the solution diversity to some extent as experienced in Differential Search [47] and Artificial Cooperative Search [48] algorithms, which is also deduced by the extensive analysis over varying types of test problems. Algorithm 2, in essence, concurrently exploits the fertile areas employing the best solution Xbest and diversifies the search space within the current iteration thanks to the ensemble of mutation strategies of Differential Evolution. Numerical experiments over many test functions reveal that too much intensification and probing around the best solution eventually leads to premature convergence, therefore we choose to impose a balanced effect of exploration and exploitation rather than mere exploitation in this phase. Within this context, the first part of Algorithm 2 is concerned with exploration while the second part dealing with exploitation. Table 1 provides the pseudo-code of the proposed hybrid Eagle Strategy.

Table 1. Pseudo-code representation of the proposed Eagle Strategy

Initialization phase
Initialize algorithm parameters
N: Population size
X : Population matrix
f(x) : Objective function
Up and Low: Upper and lower bounds of the search space
Maxiter : Maximum number of iteration
While (iter < Maxiter) do
//Exploration//
X1 = randperm(X)
X2 = randperm(X1)
for i = 1 to N
for j = 1 to D
X_new^(i,j) =(X_1^(i,j) 〖levy〗_1^(i,j)+X_2^(i,j) 〖levy〗_2^(i,j))·(2φ_1^(i,j)-1)
end
end

Xnew = Boundary_check(Xnew)
X = Update(X , Xnew)
Retain the best solution (Xbest)

//Exploitation//
X1 = randperm(X); X2 = randperm(X1); X3 = randperm(X2); X4 = randperm(X3);
oz1 = onezero()
oz2 = onezero()
for i = 1 to N
for j = 1 to D
if (rand(0,1)1 < rand(0,1)2)
X_new^(i,j)=X^(i,j)+(2φ_2^(i,j)-1)· 〖oz〗_1^(i,j)·(X_1^(i,j)-X_2^(i,j))
else
X_new^(i,j)=X_best^j+ 〖oz〗_2^(i,j)·(X_3^(i,j) φ_3^(i,j)-X_4^(i,j) φ_4^(i,j))
end
end
end

Xnew = Boundary_check(Xnew)
X = Update(X , Xnew)
Retain the best solution (Xbest)
Update random numbers drawn from Levy flights (levy1,2) and Lozi chaotic map (φ_1,2,3,4)
Increment the iteration counter (iter++)
end
Output the best result (Xbest)

3. Numerical experiments on benchmark problems

This section deals with assessing the effectiveness of the proposed hybrid Eagle Strategy (EAGLE) over different types of optimization benchmark functions. A testbed of 40 optimization benchmark functions composed of unimodal and multimodal optimization problems will be solved by the proposed method and corresponding results will be compared those acquired by the reputable literature optimizers of Barnacles Mating Optimizer (BMO) [26], Manta-Ray Foraging Optimizer (MANTA-RAY) [49], Harris Hawks Optimizer (HARRIS) [50], Particle Swarm Optimization (PSO) [3], Sine-Cosine Algorithm (SINECOS) [51], Butterfly Optimization (BFLY) [52], Fruit-Fly Optimization (FF) [50], Grey Wolf Optimization (GWO) [54], Spotted Hyena Optimizer (SH) [55], Crow Search Algorithm (CROW) [56], Differential Evolution (DE) [25], Multi-Verse Optimizer (MVO) [57], and Cuckoo Search (CUCKOO) [58]. Because of the randomness inherent in the metaheuristic algorithms, a single algorithm run may not be a decisive factor for performance judgment. 50 independent algorithm runs along with 2000 function evaluations have been performed for each compared algorithms to conquer the stochasticity. Among them, the best of 30 algorithm runs is considered for assessment of the prediction capability of the optimizers. The majority of the metaheuristics involve adjustable algorithm parameters to be tuned for different optimization test problems. One of the main advantages of the EAGLE algorithm is that it includes no tuneable parameters, which eliminates the tedious and exhaustive process of trial-and-error based problem-specific parameter tuning. Table 2 reports the parameter configuration of the algorithms that will be benchmarked against the EAGLE algorithm for performance evaluation. Adjustable parameters of the algorithms are calibrated by trial and error and final results are evaluated in terms of statistical analysis retained after successive algorithm runs. Algorithms that have not taken place in Table 2 have no tunable parameters. Numerical experiments and algorithm implementations are executed in the Java environment and performed on a personal computer having 6.0 GB RAM at 2.50 GHz CPU. Benchmark problems utilized in this study for performance analysis are categorized into two different groups including multimodal and unimodal functions, many of those have been frequently applied for judging the capabilities of stochastic optimization algorithms in the literature. Table 3 to Table 6 report the statistical results of the unimodal and multimodal test functions for each compared algorithm. The mean and standard deviation results are given in the tables to compare the effectiveness of the compared algorithms. Functions f1 to f22 are in the category of multimodal test functions. These types of test functions comprise plenty of local optimum points, which makes them an efficient testbed for assessing the exploration capability of the applied algorithm. For the f1-Levy algorithm, HARRIS provides the best performance while EAGLE and GWO achieving second and third best predictions. For the f2-Ackley algorithm, EAGLE outperforms the remaining algorithms. EAGLE, BMO, and HARRIS algorithms obtain the global best solution for each independent run for f3-Griewank and f4-Rastrigin test functions. FF algorithm acquires the best robustness and accuracy for the f5-Zakharov test function. BMO and EAGLE come with second and third best algorithms for this problem. EAGLE surpasses the compared algorithms in terms of solution efficiency for the f6-Alpine test function. Although very similar results are obtained, EAGLE slightly outperforms the remaining algorithms for the f7-Penalized1 test function. HARRIS obtains the minimum deviation results for the f8-Quintic function whereas EAGLE is the second-best performer in these terms. The predictive results obtained by EAGLE for the f9-Csendes test function are superior compared to the other approaches. BFLY retains the best predictive performance for the f10-Schaffer test function. EAGLE obtains the global best answer of f11-Inverted cosine function each algorithm run and becomes the best performer for this case. EAGLE and BMO find the global optimum solution of the f12-Wavy test function for each independent run. Deviation results produced by the f13-Hyperellipsoid function are much better than those of the compared algorithms. FF algorithm is significantly superior in finding the optimal results for the f14-Pathological test function. MANTA-RAY obtains the most robust solutions for the f15-Salomon test function. Although predictive results obtained from the compared algorithms are far away from the global optimum solutions for f16-Ackley N4, EAGLE becomes the dominant algorithm in performing the exploration of the search space, therefore produces the minimum error deviation. EAGLE and BMO obtain the global optimum solution for each consecutive run for the f17-Exponential test function. HARRIS gives the minimum mean deviation values for the f18-Trid function while EAGLE has been significantly outperforming by the remaining algorithms. MVO generates competitive optimal results for the f19-Styblinski-Tang function whereas results found by EAGLE are quite satisfactory. EAGLE yields the most accurate predictions for f20-Yang1 and f21-Yang2 test functions while unequivocally outperformed by FF for the f22-Yang4 test function. It can be concluded from the predictive performances of the algorithms that EAGLE is very competitive in exploring the search space efficiently such that it obtains the global optimum solution of 30 dimensional f3-Griewank, f4-Rastrigin, f12-Inverted cosine, f13-Wavy, and f17-Exponential test functions for each algorithm run and comes with second or third best performing algorithm in most of the multi-modal test functions. The good exploration behavior of EAGLE can be attributed to two different facts. One fact is those anomalous stochastic movements of Levy flights generated by the long jumps over the search space. Another fact is that random numbers generated by chaotic Lozi map have better dynamical and statistical characteristics, which not only help the algorithm to produce more diverse sample solutions to reach unexplored regions in the objective landscape but also accelerates the convergence speed to some extent compared to the solutions found by stochastic random numbers with uniform distribution.

Table 2. Parameter configurations for the compared algorithms

Algorithm	Parameter	Value
BMO	Penis length (*pl*)	7.0
PSO	Cognitive (c_1) and social (c_2) factors	c_1=2.0, c_2=2.0
	Inertia weight(w_1)	Decreasing linearly from 0.6 to 0.1
BFLY	Switch probability (*p*)	*p*=0.8
	Modular modality (*m*)	*m*=0.01
	Power exponent (*pe*)	Linearly increased from 0.1 to 0.3
GWO	Convergence parameter (*a*)	Decreasing linearly from 2.0 to 0.0
SH	Convergence parameter (*h*)	Decreasing linearly from 5.0 to 0.0
CROW	Flight lenght (*fl*), Awareness probability (*AP*)	*fl*=0.1, *AP*=0.1
DE	Scale Factor (*F*), Crossover Rate (*CR*)	*F*=0.9, *CR*=0.5
MVO	Wormhole Existence Probability (*WEP*)	Decreasing linearly from 1.0 to 0.2
CUCKOO	Discovering alien egg probability (p_a)	p_a=0.2
	Algorithm parameter(α)	α=1.0
	Algorithm parameter(β)	β=1.5

Table 3. Statistical results of the multimodal test functions from Levy to Ackley N4

	f_1 - Levy	**f_2 - Ackley**	**f_3 - Griewank**	**f_4 - Rastrigin**
	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev
EAGLE	1.24E+00±1.71E-01	**1.22E-15±1.47E-15**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
BMO	1.70E+00±2.66E-01	3.19E-11±6.63E-11	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
MANTA-RAY	1.40E+00±2.92E-01	4.90E-04 ±4.05E-04	1.63E-02±5.19E-02	4.62E+01±4.97E+01
HARRIS	**7.72E-01±2.11E-01**	1.68E-10±3.84E-10	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
PSO	8.77E+00±3.06E+00	1.90E+00±5.37E-01	3.52E-01±1.43E-01	1.85E+02±4.18E+01
SINECOS	3.15E+00±2.21E+00	1.84E-01±1.66E-01	1.88E-01±2.22E-01	1.47E+02±6.03E+01
BFLY	3.21E+00±5.69E-02	1.37E-11±1.12E-12	2.19E-14±2.99E-15	4.06E-12±8.29E-12
FF	3.11E+00±7.19E-02	2.42E-03±7.15E-06	6.58E-07±5.74E-09	2.18E-03±1.61E-05
GWO	1.27E+00±2.15E-01	5.53E-03±1.45E-03	4.68E-06±4.66E-06	1.79E+01±5.05E+00
SH	2.24E+00±1.46E+00	3.06E-02±2.09E-02	6.21E-02±1.00E-01	1.00E+02±5.28E+01
CROW	1.21E+01±4.00E+00	3.39E+00±4.38E-01	7.66E-01±1.22E-01	2.80E+02±3.21E+01
DE	1.09E+01±2.40E+00	2.86E+00±3.43E-01	7.23E-01±1.10E-01	2.50E+02±1.87E+01
MVO	2.08E+01±8.21E+00	5.40E-01±8.76E-02	9.16E-02±3.19E-02	2.09E+02±4.01E+01
CUCKOO	5.10E+01±1.37E+01	5.31E+00±4.96E-01	9.82E-01±5.03E-02	3.53E+02±3.55E+01
	f_5 - Zakharov	**f_6 - Alpine**	**f_7 - Penalized1**	**f_8 - Quintic**
	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev
EAGLE	1.02E-01±6.81E-01	**6.21E-21±9.04E-21**	**3.23E-02±1.41E-02**	2.31E+01±4.41E+00
BMO	**6.42E-02±5.61E-01**	6.63E-12±2.14E-11	2.03E-01±1.23E-01	4.76E+01±9.76E+00
MANTA-RAY	1.91E+03±2.68E+03	9.81E-01±5.01E+00	8.23E-02±7.53E-02	5.81E+01±1.90E+01
HARRIS	1.36E+02±1.42E+02	1.40E-10±3.38E-10	3.51E-02±1.17E-02	**2.18E+01±3.57E+00**

PSO	3.73E+02±8.54E+01	8.43E+00±3.37E+00	5.42E-01±3.48E-01	2.32E+02±2.05E+02
SINECOS	2.72E+02±9.16E+01	2.33E+00±3.61E+00	5.43E-01±1.89E-01	1.19E+02±3.47E+01
BFLY	6.06E+02±1.63E+02	1.32E-10±5.69E-11	8.81E-01±1.40E-01	1.29E+05±2.33E+04
FF	1.70E-02±1.43E-04	1.82E-03±6.77E-06	1.67E+00±9.00E-06	1.16E+02±6.24E-04
GWO	1.47E+01±9.75E+00	1.25E-02±3.85E-03	7.39E-02±4.08E-02	4.72E+01±1.20E+01
SH	8.03E+01±6.11E+01	3.04E+00±4.67E+00	3.02E-01±1.49E-01	7.53E+02±1.53E+01
CROW	4.54E+02±7.19E+01	1.73E+01±3.25E+00	1.21E+00±5.91E-01	9.66E+02±9.25E+02
DE	5.42E+02±6.62E+01	1.95E+01±2.32E+00	1.90E+00±6.26E-01	3.82E+02±1.59E+02
MVO	2.64E+02±7.93E+01	1.20E+01±3.46E+00	1.72E+00±1.06E+00	6.22E+01±1.38E+01
CUCKOO	1.67E+03±6.74E+02	3.76E+01±5.51E+00	5.04E+00±1.25E+00	2.10E+03±1.13E+03
	***f_9* - Csendes**	***f_{10}* - Schaffer**	***f_{11}* - Inverted cosine**	***f_{12}* - Wavy**
	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev
EAGLE	**1.82E-76±1.45E-75**	1.43E-03±4.17E-04	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
BMO	7.36E-55±4.93E-54	2.09E-03±9.76E-04	5.35E-18±4.83E-17	**0.00E+00±0.00E+00**
MANTA-RAY	4.06E-01±2.29E+00	3.55E-02±1.79E-02	6.69E-06±1.25E-05	3.21E-01±2.05E-01
HARRIS	8.59E-51±8.38E-50	1.97E-03±1.13E-03	2.30E-17±2.01E-16	7.38E-03±6.76E-02
PSO	2.34E+03±4.48E+03	8.14E-02±1.47E-02	1.11E+01±3.65E+00	7.27E-01 ±5.20E-02
SINECOS	1.87E+02±7.17E+02	5.12E-02±1.69E-02	2.46E-03±6.33E-01	5.73E-01±1.22E-01
BFLY	1.98E+06±3.92E+06	**7.17E-05±5.41E-04**	7.16E-13±5.77E-13	6.36E-14±2.06E-14
FF	7.74E-18±2.64E-18	1.01E-03±4.28E-03	1.46E-04±1.11E-06	1.85E-05±1.31E-07
GWO	1.70E-09±5.14E-09	2.17E-02±3.05E-02	2.66E-02±6.79E-02	2.01E-01±4.86E-02
SH	2.42E-01±9.55E-01	2.63E-02±1.09E-02	1.65E-01±2.82E-01	3.81E-01±1.68E-01
CROW	1.04E+04±7.98E+03	9.81E-02±1.50E-02	2.82E+01±8.51E+00	8.15E-01±2.41E-02
DE	4.96E+03±3.78E+03	1.19E-01±1.49E-02	2.07E+01±4.91E+00	7.35E-01±2.13E-02
MVO	1.82E-01±2.33E-01	7.23E-02±1.93E-02	3.98E+00±8.04E-01	7.39E-01±3.52E-02
CUCKOO	1.95E+04±1.21E+04	1.31E-01±2.01E-01	7.14E+01±1.62E+01	8.57E-01±2.05E-02
	***f_{13}* - Hyperellipsoid**	***f_{14}* - Pathological**	***f_{15}* - Salomon**	***f_{16}* - Ackley N4**
	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev
EAGLE	**1.13E-32±4.40E-32**	3.27E+00±4.01E-01	9.98E-02±1.26E-06	**-6.31E+01±5.12E+00**
BMO	3.35E-18±2.56E-17	5.00E+00±4.50E-01	9.78E-02±1.60E-02	-1.71E+01±1.71E+01
MANTA-RAY	7.97E-05±1.27E-04	5.41E+00±3.33E-01	**3.63E-02±1.41E-01**	-3.75E+01±1.55E+01
HARRIS	1.33E-16±6.86E-16	3.34E+00±5.17E-01	9.98E-02±1.73E-16	-6.06E+01±1.05E+01
PSO	7.72E+02±4.78E+02	4.20E+00±5.05E-01	1.15E+00±1.43E-01	-4.03E+01±1.21E+01
SINECOS	1.35E+01±1.53E+01	4.79E+00±2.85E-01	7.08E-01±1.50E-01	5.06E+00±9.55E+00
BFLY	1.00E+05±1.41E+04	4.75E+00±4.41E-01	1.01E-01±3.15E-03	5.60E+01±9.23E+00
FF	2.92E-03±2.45E-05	**3.99E-08±6.61E-10**	5.97E-01±2.05E-01	8.71E+01±5.01E-01
GWO	6.89E-03±5.34E-03	8.84E-01±4.41E-01	3.52E-01±7.51E-01	-5.49E+01±9.67E+00
SH	1.87E-01±2.91E-01	4.71E+00±4.46E-01	5.52E-01±1.13E-01	2.76E+01±1.10E+01
CROW	2.86E+03±1.03E+01	4.86E+00±2.99E-01	1.26E+00±1.17E-01	2.79E+00±1.36E+01
DE	1.99E+03±5.05E+02	4.03E+00±3.18E-01	1.45E+00±1.07E-01	1.05E+00±8.18E+00
MVO	7.55E+02±2.89E+02	3.72E+00±4.53E-01	1.03E+00±1.84E-01	-1.84E+01±1.69E+01
CUCKOO	1.29E+04±3.59E+03	5.61E+00±2.37E-01	1.83E+00±1.83E-01	4.46E+01±1.22E+01

Table 4. Statistical results of the multimodal test functions from Exponential to Yang4

	***f_{17}* - Exponential**	***f_{18}* - Trid 6**	***f_{19}* - Styblinski-Tang**
	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev
EAGLE	**-1.00E+00±0.00E+00**	-1.47E+02±4.25E+01	-8.43E+02±4.52E+01
BMO	**-1.00E+00±0.00E+00**	-2.49E+01±1.32E+01	-8.47E+02±1.03E+02
MANTA-RAY	-9.99E-01±2.36E-06	-1.07E+02±1.47E+02	-8.06E+02±5.57E+02
HARRIS	-1.00E+00±3.31E-17	**-1.42E+03±1.43E+02**	-8.49E+02±8.56E+01
PSO	-5.22E-02±7.34E-02	8.51E+01±3.46E+02	-8.90E+02±5.84E+01
SINECOS	-9.91E-01±1.36E-01	5.32E+01±9.45E+01	-4.73E+02±3.88E+01
BFLY	-5.87E-109±5.53E-108	7.55E+03±1.74E+03	3.61E+03±2.24E+03
FF	-9.99E-01±4.09E-08	2.85E+01±3.62E-01	3.14E-01±4.70E+00
GWO	-9.99E-01±2.74E-05	-1.32E+02±8.87E+01	-7.46E+02±6.86E+01
SH	-9.97E-01±2.97E-03	-6.86E+00±3.83E+01	-6.48E+02±4.93E+01
CROW	-3.52E-04±1.37E-03	5.62E+02±3.88E+02	-6.64E+02±7.91E+01
DE	-8.01E-04±1.56E-03	-5.46E+02±2.71E+02	-6.69E+02±4.63E+01
MVO	-8.86E-01±3.44E-02	-5.43E+02±3.22E+02	**-9.63E+02±3.81E+01**
CUCKOO	-3.39E-05±7.48E-05	1.83E+03±5.78E+02	-5.42E+02±6.25E+01
	***f_{20}* - Yang1**	***f_{21}* - Yang2**	***f_{22}* -Yang4**
	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev
EAGLE	**1.24E-63±7.79E-63**	**1.05E-08±1.10E-08**	1.65E-12±3.85E-13
BMO	1.61E-21±8.50E-21	1.46E-05±2.17E-05	6.47E-11±1.40E-10
MANTA-RAY	6.05E+01±2.97E+02	3.87E-05±7.37E-05	1.46E+01±2.15E+01
HARRIS	9.66E-06±6.28E-05	1.20E-07±1.60E-07	2.95E-12±1.05E-12
PSO	1.24E+08±5.40E+08	1.04E-07±3.97E-07	1.62E+01±2.72E+01
SINECOS	2.89E+05±2.88E+06	3.60E-06±4.74E-06	1.80E+01±2.17E+01
BFLY	5.29E+17±1.50E+18	8.99E-07±1.06E-06	2.98E-11±7.81E-12
FF	4.96E-03±3.47E-03	8.91E-01±2.91E-01	**-9.81E-01±6.24E-05**
GWO	2.84E-04±6.01E-04	3.05E-08±1.26E-07	1.89E-13±3.48E-13
SH	1.58E+02±6.62E+02	8.99E-07±2.66E-06	2.17E-11±1.55E-11
CROW	1.32E+08±6.90E+08	2.02E-06±2.33E-06	1.46E-11±1.03E-11
DE	3.98E+08±2.50E+09	1.89E-07±1.95E-07	1.11E-11±4.34E-12
MVO	7.67E+08±4.19E+09	2.99E-07±9.38E-07	3.44E-13±2.42E-13
CUCKOO	6.41E+14±2.51E+15	4.17E-04±4.77E-04	1.17E-09±5.53E-10

Table 5. Comparative analysis of the statistical results of unimodal test functions from Sphere to Yang 3

	f_{23} - **Sphere**	f_{24} - **Rosenbrock**	f_{25} - **Brown**	f_{26} - **Streched sine wave**
	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev
EAGLE	**2.76E-34±9.26E-34**	**2.75E+01±4.64E-01**	**1.11E-33±3.61E-33**	**3.33E-10±2.02E-10**
BMO	4.18E-21±1.51E-20	2.84E+01±2.72E-01	1.02E-19±5.24E-19	1.41E-05±2.08E-05
MANTA-RAY	2.67E-06±1.38E-05	2.69E+02±1.29E+03	5.48E-04±1.58E-03	3.47E-01±2.41E-01
HARRIS	8.62E-20±1.90E-19	2.85E+01±2.04E-01	5.61E-18±3.12E-17	1.65E-04±2.40E-04
PSO	7.69E+00±4.39E+00	7.94E+03±6.99E+03	3.29E+16±2.61E+17	4.22E+01±5.70E+00
SINECOS	1.58E-01±2.98E-01	6.55E+02±1.95E+03	1.99E+00±4.21E+00	8.94E+00±3.97E+00
BFLY	7.14E-13±4.04E-13	1.89E+06±2.99E+06	2.38E+93±1.81E+94	3.74E-08±1.84E-08
FF	1.10E-05±7.30E-08	2.86E+01±1.12E-02	2.12E-05±1.43E-07	6.02E+00±1.15E+00
GWO	6.85E-05±9.26E-05	2.86E+01±6.14E-01	3.01E-04±1.90E-04	3.86E+00±6.93E-01
SH	4.62E-03±6.08E-03	5.13E+01±5.61E+01	1.50E-01±7.78E-01	8.33E+00±5.47E+00
CROW	2.64E+01±8.35E+00	2.39E+04±1.24E+04	9.90E+10±6.01E+11	4.41E+01±4.82E+00
DE	1.77E+01±4.61E+00	1.69E+04±6.96E+03	1.87E+15±1.38E+16	4.74E+01±3.81E+00
MVO	5.42E-01±1.98E-01	3.08E+02±1.95E+02	2.62E+02±1.41E+02	5.19E+01±8.43E+00
CUCKOO	6.95E+01±1.87E+01	6.58E+04±2.62E+04	7.81E+34±8.59E+35	6.95E+01±3.80E+00
	f_{27} - **Powell singular**	f_{28} - **Sum of different powers**	f_{29} - **Sum of squares**	f_{30} - **Bent cigar**
	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev
EAGLE	**7.93E-32±2.74E-31**	**9.87E-70±5.87E-69**	**2.38E-33±5.54E-33**	**1.28E-28±3.44E-28**
BMO	2.92E-18±1.60E-17	2.12E-23±9.94E-23	8.67E-19±8.06E-18	9.57E-14±8.32E-13
MANTA-RAY	1.54E-02±7.03E-02	2.39E+02±1.32E+03	9.14E-06±1.37E-05	5.38E-01±6.93E-01
HARRIS	3.49E-16±1.41E-15	3.40E-16±2.31E-15	1.57E-17±8.17E-17	4.34E-12±2.23E-11
PSO	4.78E+03±3.72E+03	4.87E+06±1.82E+07	8.98E+01±4.75E+01	6.88E+06±5.15E+06
SINECOS	9.58E+01±2.39E+02	2.09E+04±1.17E+06	1.60E+00±2.21E+00	8.83E+04±1.76E+05
BFLY	9.81E+05±1.76E+05	5.87E+17±1.62E+18	2.88E+03±2.22E+03	4.95E+08±4.58E+07
FF	1.18E-03±9.42E-06	3.99E-04±4.30E-06	1.65E-04±1.23E-06	1.05E+01±8.53E-02
GWO	1.05E-02±8.02E-03	1.29E-07±7.25E-07	9.57E-04±7.03E-04	5.26E+01±3.84E+01
SH	3.32E+00±9.19E+00	1.98E+00±6.93E+00	4.23E-02±7.37E-02	2.57E+03±3.04E+03
CROW	1.69E+04±9.77E+03	1.00E+08±4.13E+08	2.91E+02±8.61E+01	2.35E+07±7.31E+06
DE	2.58E+04±8.75E+03	1.03E+07±3.95E+07	2.01E+02±5.74E+01	1.49E+07±3.50E+07
MVO	1.13E+02±9.17E+01	5.49E+04±2.37E+05	2.99E+01±1.61E+01	4.89E+05±1.66E+05
CUCKOO	4.71E+04±2.37E+04	7.21E+13±2.37E+14	9.16E+02±2.37E+02	6.47E+07±1.58E+07
	f_{31} - **Discus**	f_{32}- **Different powers**	f_{33} - **Dixon-Price**	f_{34} - **Yang 3**
	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev
EAGLE	**3.45E-34±7.32E-34**	**4.22E-26±3.36E-25**	**6.66E-01±4.43E-06**	8.07E-01±9.79E-03
BMO	1.88E-19±7.01E-19	4.49E-14±3.27E-13	7.00E-01±9.12E-02	5.83E-01±3.91E-01
MANTA-RAY	4.18E-06±1.01E-05	1.45E-03±3.45E-03	9.06E-01±6.63E-01	8.84E-01±3.12E-01
HARRIS	1.10E-15±7.51E-15	9.73E-12±2.65E-12	6.68E-01±1.56E-03	8.58E-01±1.74E-02
PSO	1.63E+01±9.92E+00	4.50E+00±1.93E+00	2.48E+03±1.93E+03	8.71E-01±1.28E-02
SINECOS	2.46E-01±4.64E-01	7.41E-01±9.29E-01	8.99E+01±1.71E+02	8.75E-01±1.06E-02
BFLY	4.49E-12±7.88E-12	9.41E-11±8.23E-11	1.07E+06±1.78E+05	8.69E-01±1.08E-02
FF	1.60E-01±4.52E-03	5.90E-04±3.89E-06	9.79E-01±1.46E-02	**-9.99E-01±3.47E-07**
GWO	1.47E-04±8.32E-05	4.63E-04±3.20E-04	7.53E-01±1.48E-01	9.79E-01±4.67E-04
SH	9.78E-03±1.52E-02	1.05E-01±1.89E-01	6.01E+00±1.46E+01	6.57E-01±1.58E-02
CROW	3.91E+01±1.18E+01	1.04E+01±2.93E+00	1.06E+04±5.62E+03	8.59E-01±1.14E-02
DE	3.21E+01±8.22E+00	8.39E+00±1.65E+00	6.35E+03±2.89E+03	7.86E-01±1.12E-01
MVO	3.38E+02±1.50E+02	3.86E-01±1.77E-01	5.53E+01±3.87E+01	8.10E-02±1.86E-02
CUCKOO	3.69E+03±2.19E+03	3.14E+01±7.62E+00	3.78E+04±1.66E+04	8.98E-01±2.13E-02

Table 6. Mean and standard deviation error analysis of unimodal test functions from Schwefel 2.20 to Dropwave

	f_{35} - **Schwefel 2.20**	f_{36} - **Schwefel 2.21**	f_{37} - **Schwefel 2.22**
	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev
EAGLE	**3.07E-20±4.43E-20**	**3.42E-119±2.39E-118**	**2.47E-20±2.29E-20**
BMO	1.04E-10±7.04E-10	6.44E-34±4.69E-33	4.66E-11±1.55E-10
MANTA-RAY	2.43E-05±2.33E-05	6.90E-59±1.73E-58	3.39E-05±4.84E-05
HARRIS	3.53E-09±1.31E-08	1.02E-23±5.68E-23	6.49E-10±1.12E-09
PSO	8.07E+00±2.05E+00	3.09E-15±1.68E-14	8.31E+00±2.47E+00
SINECOS	1.40E-01±1.25E-01	1.16E-27±6.90E-27	1.33E-01 ±1.09E-01
BFLY	1.74E-10±9.35E-11	3.79E-14±4.61E-14	7.29E-11±3.66E-11
FF	1.81E-02±7.19E-05	4.01E-04±1.19E-05	1.81E-02±6.83E-05
GWO	1.40E-02±6.43E-03	1.37E-36±7.65E-36	1.33E-02±4.24E-03
SH	1.37E-02±9.09E-03	9.04E-65±6.31E-64	1.09E-02±6.51E-03
CROW	1.53E+01±2.53E+00	5.33E-18±2.14E-17	1.69E+01±2.88E+00
DE	1.31E+01±1.94E+00	4.10E-12±9.60E-12	1.58E+01±2.52E+00
MVO	5.53E+00±1.68E+00	1.35E-04±1.38E-04	4.79E+01±1.86E+02
CUCKOO	3.52E+01±4.49E+00	5.41E-18±9.82E-18	2.74E+05±1.57E+06
	f_{38} - **Schwefel 2.23**	f_{39} - **Schwefel 2.25**	f_{40} - **Dropwave**
	Mean±Std.dev	Mean±Std.dev	Mean±Std.dev
EAGLE	**7.09E-116±5.59E-115**	8.79E+00±1.67E+00	-9.37E-01±7.54E-03
BMO	6.25E-83±5.90E-82	1.53E+01±1.70E+00	**-9.44E-01±2.12E-02**
MANTA-RAY	2.78E+04±2.48E+05	4.12E+01±8.18E+01	-7.34E-01±1.13E-01
HARRIS	1.39E-83±1.14E-82	**3.50E+00±1.28E+00**	-9.36E-01±1.65E-15
PSO	3.24E+05±4.61E+05	2.64E+02±1.54E+02	-4.79E-02±1.51E-02
SINECOS	3.74E+05±1.48E+06	5.62E+01±8.55E+01	-2.25E-01±1.30E-01
BFLY	2.77E+09±9.92E+08	1.52E+04±3.76E+03	-4.61E-03±7.11E-04

FF	2.17E-31±7.42E-33	2.38E+01±5.04E-01	-3.63E-01±2.00E-01
GWO	2.38E-13±1.42E-12	1.37E+01±2.25E+00	-6.13E-01±7.11E-02
SH	7.84E+01±5.69E+02	2.33E+01±8.42E+00	-2.89E-01±9.70E-02
CROW	1.76E+06±2.19E+06	4.16E+02±1.97E+02	-3.32E-02±8.24E-03
DE	9.69E+05±1.05E+06	4.54E+02±1.49E+02	-2.46E-02±5.06E-03
MVO	4.82E+00±1.53E+01	2.24E+01±2.56E+01	-1.29E-01±4.32E-02
CUCKOO	4.44E+06±7.11E+07	7.95E+02±3.02E+02	-2.93E-02±7.08E-03

Table 5 and Table 6 provide the deviation results of each algorithm for unimodal test functions. Unimodal test functions have only one global optimum point and are efficient testbeds for assessing the exploitative performance of the algorithm on promising regions. In general overview, it is observed that the optimization capabilities of the EAGLE algorithm in solving unimodal algorithms are much superior compared to multimodal optimization problems. Success in finding competitive results for unimodal test functions can be attributed to the modified DE-reinforced ensemble search mechanism which can provide reasonable solution outcomes thanks to the effective balance between exploration and exploitation maintained in this phase of the hybrid algorithm. EAGLE produces better results than the other methods for the f23-Sphere function. None of the contestant algorithms obtain the optimal value of the f24-Rosenbrock test function. The global optimal solution of this problem occupies the narrow and parabolic valley, which is easy to locate, however convergence to the global optimum point in this valley is relatively difficult. Nevertheless, EAGLE obtains competitive mean and standard deviation results for this test problem. EAGLE significantly outperforms the other algorithms for the f25-Brown test function. For the f26-Streched sine wave function, EAGLE is the best performing method while BFLY is the second-best method among them. Prediction superiority of EAGLE is so evident for f27-Powell singular, f28-Sum of different powers, f29-Sum of squares, and f30-Bent cigar, f31- Discus, and f32-Different powers test functions as this optimizer retains much better statistical results in terms of solution accuracy and persistence. Despite the inefficacy in finding the optimum solution of f33 -Dixon-Price, EAGLE becomes the most prominent algorithm among the compared methods for this case. None of any algorithm, except FF, even gets closer to the optimal solution of the f34-Yang3 test function. The prediction capability of EAGLE for f35-Schwefel 2.20, f36-Schwefel 2.21, f37-Schwefel 2.22, and f38- Schwefel 2.23 is quite remarkable. HARRIS outperforms the compared methods for the f39-Schwefel 2.25 test function while EAGLE becoming the second-best optimizer. BMO is superior to the other optimizers concerning finding the optimal solution of the f40-Dropwave test function, which is highly complex and nonlinear.

4. Solving a transient heat conduction problem

In this section, the one-dimensional transient heat conduction problem is converted into a nonlinear optimization problem and solved by different literature optimizers along with the proposed hybrid Eagle Strategy. Temperature distribution in a slab $T(x,t)$ is conventionally obtained by applying the separation of variables method on governing equations associated with mathematical modeling of heat dissipation in the one-dimensional medium [59]. One dimensional transient heat conduction in a finite medium is modeled as a boundary value problem whose successful solution is accomplished by the application of the separation of variables method. That is, consider a slab $0 \le x \le L$ as a heat transfer medium whose initial temperature is $T = F(\mathrm{x})$, dissipates heat through convection mechanism at times $t > 0$ from boundary surfaces to the surrounding ambient at zero temperature. For generality, it is assumed that convective heat transfer coefficients at two boundaries are not the same. With taking into account these assumptions, mathematical modeling of one-dimensional heat conduction equation can be formulated as

$$\frac{\partial^2 T(x,t)}{\partial x^2} = \frac{1}{\alpha}\frac{\partial T(x,t)}{\partial t} \text{ in } 0 \le x \le L, t > 0 \quad (25)$$

$$-k_1\frac{\partial T}{\partial x} + h_1 T = 0 \text{ at } x = 0, t > 0 \quad (26)$$

$$-k_2\frac{\partial T}{\partial x} + h_2 T = 0 \text{at} x = L, t > 0 \quad (27)$$

$$T = F(\mathrm{x}) \text{for } t = 0 \text{ in } 0 \le x \le L \quad (28)$$

Where x represents the spatial coordinate of the heat transfer medium, t is time, L is the length of the heat transfer medium, $T(x,t)$ is temperature distribution in the slab as a function of time (t) and space (x), α is the heat diffusivity; k_1 and k_2 are thermal conductivities of two different boundary surfaces, h_1 and h_2 are convective heat transfer coefficients of two different boundaries.

Obeying the rules of separation of variables method, the expression temperature distribution in the heat transfer medium $T(x,t)$ is separated in two different terms equated by the following

$$T(x,t) = X(x) \cdot \Gamma(t) \quad (29)$$

Where $X(x)$ is a space-variable function and $\Gamma(t)$ is the time-variable function whose analytical formulation is given below

$$\Gamma(t) = e^{-\alpha\beta^2 t} \quad (30)$$

And the space-variable function $X(\beta, x)$ should satisfy the below-defined eigenvalue problem

$$\frac{d^2 X(x)}{dx^2} + \beta X(x) = 0 \text{ in } 0 < x < L \quad (31)$$

$$-k_1\frac{dX}{dx} + h_1 X = 0 \text{ at } x = 0 \quad (32)$$

$$-k_2\frac{dX}{dx} + h_2 X = 0 \text{ at } x = L \quad (33)$$

The eigenfunctions $X(\beta, x)$ are orthogonal within the current forms of Eq.(31) to Eq. (33) That is,

$$\int_0^L X(\beta_m, x)X(\beta_m, x)\,dx = \begin{cases} 0 & for\ m \ne n \\ N(\beta_m) & for\ m = n \end{cases} \quad (34)$$

Then, the solution of the set of equations described in Eq.(25) to Eq.(28) can be reformulated in a simplified form with the following expression

$$T(x,t) = \sum_{n=1}^{\infty} c_n X(\beta_n, x)e^{-\alpha\beta_m^2 t} \quad (35)$$

And reformulation of the initial temperature conditions can be expressed by

$$F(x) = \sum_{n=1}^{\infty} c_n X(\beta_n, x) \text{ in } 0 < x < L \quad (36)$$

Eq.(36) is the reexpression of the initial temperature condition $F(x)$ defined in the interval $0 < x < L$by eigenfunctions $X(\beta, x)$ of the eigenvalue problem. Assume that the above formulation is permissible and c_n coefficients in Eq.(36) can be obtained by operating both side of the equations by the integral operator $\int_0^L X(\beta_{\mathrm{n}}, x)\,dx$and taking advantage of the orthogonality property of the eigenfunctions, one can obtain the following equation

$$c_n = \frac{1}{N(\beta_n)}\int_0^L X(\beta_n, x)F(x)dx \quad (37)$$

And the norm $N(\beta_n)$expressed in Eq.(34) is given by

$$N(\beta_n) = \int_0^L X^2(\beta_n, x) \cdot dx \quad (38)$$

Integration of Eq.(37) into Eq.(35) yields the temperature distribution in a slab in the final form of

$$T(x,t) = \sum_{n=1}^{\infty} e^{-\alpha\beta_n^2 t} \frac{X(\beta_n,x)}{N(\beta_n)} \int_0^L X(\beta_n, x')F(x')dx' \quad (39)$$

In the above equation, temperature diffusion in a slab is modeled utilizing eigenfunction, normalized integral function, and time-variable function. Based on the boundary conditions mathematically described in Eq.(31) to Eq.(33), the analytical solution of the eigenfunctions $X(\beta_n, x)$ for the eigenvalue problem is given by

$$X(\beta_n, x) = \beta_n \cos(\beta_n x) + H_1 \sin(\beta_n x) \quad (40)$$

The eigenvalues β_n can be acquired by solving the roots of the following transcendental equation

$$\tan(\beta_n L) = \frac{\beta_n(H_1+H_2)}{\beta_n^2 - H_1 H_2} \quad (41)$$

And the normalization integral $N(\beta_n)$ for the related problem is

$$N(\beta_n) = \frac{1}{2}\left[(\beta_n^2 + H_1^2)\left(L + \frac{H_2}{\beta_n^2+H_2^2}\right) + H_1\right] \quad (42)$$

Where

$$H_1 = \frac{h_1}{k_1} \text{ and } H_2 = \frac{h_2}{k_2} \quad (43)$$

One can easily conclude from the above-defined equations that once the eigenvalues β_n are calculated from its associated transcendental equation, eigenfunctions $X(\beta_n, x)$and the normalization integral $N(\beta_n)$will become known quantities. Thereby, temperature distribution $T(x,t)$ across the heat transfer medium can be obtained by using Eq.(39). However, problems may occur in obtaining precise solutions of the transcendental equation, which requires the application of the suitable algorithmic procedure. Different solution strategies in the literature can be found for solving Eq.(41) including graphical methods and different types of root-finding algorithms. Albeit, each different method has intrinsic drawbacks, complicating to obtain the accurate solution of the problem. Graphical methods can be utilized to locate the roots of the transcendental equations, however, it is not guaranteed to obtain the accurate position of the each root. Accurate values of the roots of transcendental equations can be alternatively determined by using derivative-based optimization methods such as Newton Raphson algorithm, root-finding methods such as the Bisection algorithm, Secant methods, etc. However, these mentioned methods needs the correct region where the each root resides. There are two option to overcome this issue. The first option is to utilize graphical methods to locate the region of the roots. The second option is to make reasonable initial estimates and then utilize an iterative method to find the roots. The second option may not be a preferable idea because it is not always possible to make good initial guesses. Corresponding eigenvalues of the each root can be found by an optimization technique to overcome the difficulties in determining the accurate values of the roots. For example, Eq.(41) can be reorganized as an optimization problem,

$$f(\beta_n) = \tan(\beta_n L) - \frac{\beta_n(H_1+H_2)}{\beta_n^2 - H_1 H_2} \quad (44)$$

Root Mean Sqaure Error (RMSE) metric is used to formulate the objective function of the optimization problem,

$$\text{argmin } f_{RMSE}(\beta_n) = \sqrt{\frac{1}{N}\sum_{n=1}^{N} f(\beta_n)^2} \quad (45)$$

Where N is the number of transcendental roots (eigenvalues) to be applied for solving Eq.(39).

4.1 Case study

Consider a slab $0 \leq x \leq L$ whose initial temperature is $F(x) = T_0$ constant, as simply visualized in Figure 3. Heat is dissipated into the ambient at zero temperature from two boundary surfaces for times $t > 0$. With known thermal conductivities (k_1 and k_2) and convective heat transfer coefficients (h_1 and h_2) of each boundary, mathematical modeling of the problem can be formulated by the equations from Eq.(26) to Eq.(28). By solving these set of equations, temperature distribution $T(x,t)$ in the heat transfer medium can be obtained by the following expression, which is the rearranged form of Eq.(39) by incorporating Eq.(40) and Eq.(42)

$$T(x,t) = \sum_{n=1}^{\infty} e^{-\alpha\beta_n^2 t} \frac{T_o(\beta_n \cos(\beta_n x) + H_1 \sin(\beta_n x))}{2\left[(\beta_n^2 + H_1^2)\left(L + \frac{H_2}{\beta_n^2+H_2^2}\right) + H_1\right]} \int_0^L \left(\beta_n \cos(\beta_n x') + H_1 \sin(\beta_n x')\right) dx' \quad (46)$$

Where β_n eigenvalues positive roots of Eq.(44). Assume that length of the slab (L) is 1.0 m; thermal conductivities of each boundary are respectively k_1=13.0 W/m.K and k_2=50.0 W/m.K; convective heat transfer coefficients of boundary surfaces are respectively h_1=100.0 W/m^{2}K and h_2=200 W/m^{2}K; thermal diffusivity of the slab is α=1.1E-5 m^2/s, and initial slab temperature is taken as T_0=25°C. Figure 4 illustrates the sequence of normalized values of $\beta_n$eigenvalues obtained by different algorithms. It should be noted that several eigenvalues (n) stand for the dimensionality of the optimization problem, which is considered to be n=30 for this case. Eigenvalues obtained by the compared algorithms are normalized with the exact analytical solutions found by the Bisection root-finding method. As it is seen, respective eigenvalues retained by the EAGLE method fully agree with the exact eigenvalues while other solutions found by the compared algorithms are far away from the exact solutions. Deviations in the early eigenvalues are quite remarkable, this is because of the extensive nonlinearities that occurred in the defined search regions, which have not been well coped with the contestant algorithms. Figure 5 shows the temperature distribution in the slab obtained by the Bisection root-finding method and compares the results with those found by the proposed EAGLE along with optimization algorithms of SH, GWO, SEAGULL, SINECOS, WHALE, HARRIS, PSO, and BFLY. It is observed that temperature distribution in the slab retained by solving Eq.(44) through the Bisection-root finding method for different elapsed times is almost identical to those found by the EAGLE algorithm. Other compared algorithms fail to predict accurate one-dimensional temperature distribution due to their incapabilities in finding correct values of $\beta_n$eigenvalues. Table 7 reports the statistical results of the objective function defined in Eq.(45) obtained for the compared algorithms. The superiority of the EAGLE is so evident that even the worst solution acquired by this method is much better than the best solution of the second-best performing algorithm, which is PSO for this case. Figure 6 shows the convergence plots of the objective functions for different algorithms. After 3000 iterations, the solution is converged to its optimum point for the EAGLE algorithm.

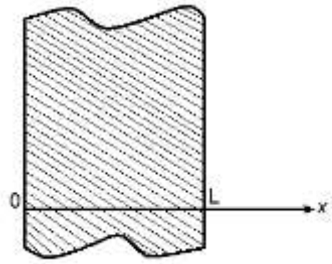


Fig. 3. Schematic view of a heat transfer medium

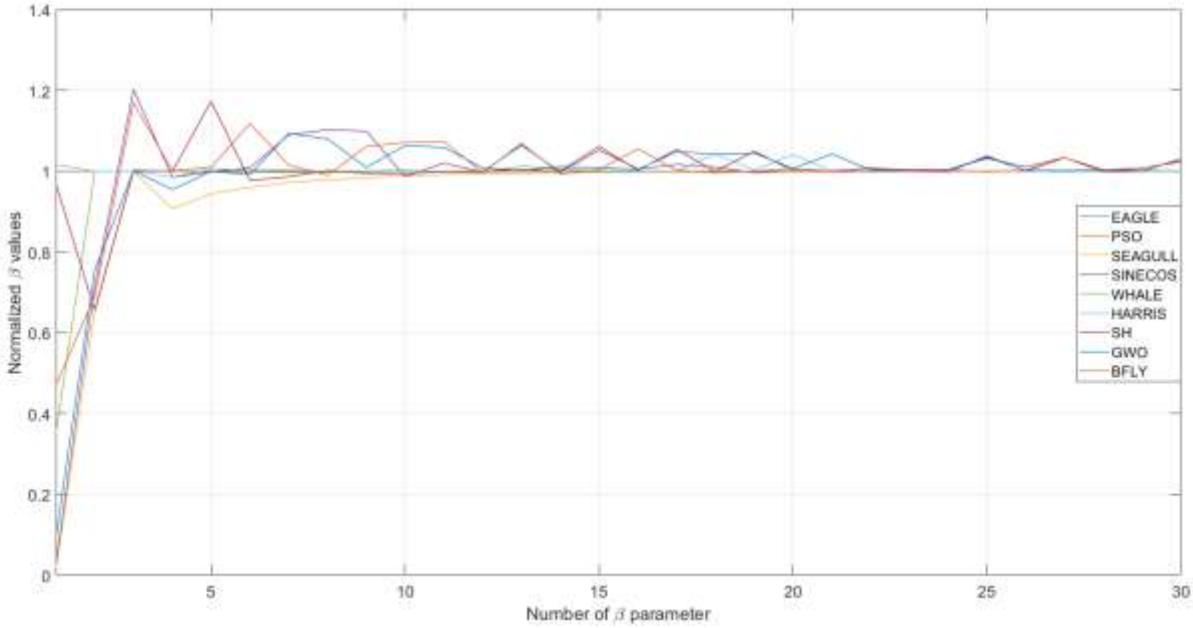


Fig. 4. Numerical values of normalized β parameters for the case study

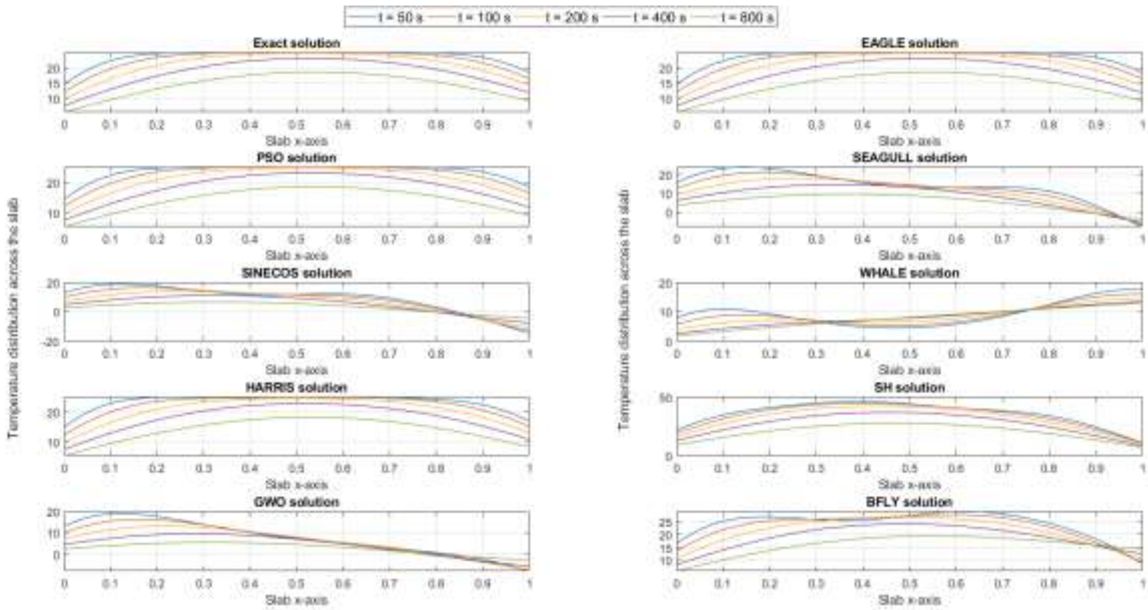


Fig. 5. Temperature distribution along the slab found by different algorithms for the case study

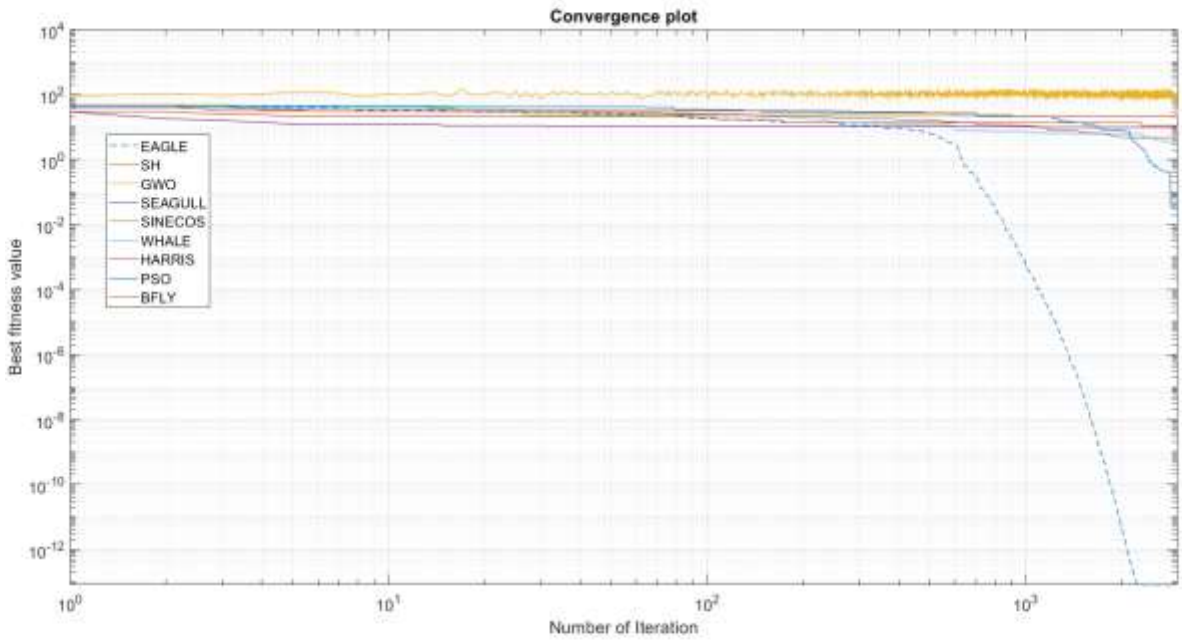


Fig. 6. Convergence plots for the case study

Table 7. Statistical results for the case study

	Best	Std. Dev.	Mean	Worst
EAGLE	8.0518E-14	1.3873E-15	8.1068E-14	8.7568E-14
SH	7.3177E+00	1.2377E+00	9.7899E+00	1.2786E+00
GWO	1.8849E+01	4.4840E+00	2.6197E+01	3.5794E+01
SEAGULL	9.8494E+00	4.7187E+00	1.5985E+01	3.5253E+01
SINECOS	2.1628E+01	1.3671E+00	2.4417E+01	2.7293E+01
WHALE	3.3480E+00	3.1824E+00	7.5697E+00	1.5646E+00
HARRIS	4.6677E+00	2.8974E+00	9.4215E+00	1.7090E+01
PSO	3.3033E-02	9.2369E-01	1.1679E+00	5.0121E+00
BFLY	2.1001E+01	1.6621E+00	2.4363E+01	2.8312E+01

5 Conclusion

This study suggests an alternative approach for solving one-dimensional transient heat conduction problems. Rather than applying conventional root-finding methods to obtain the tendencies of temperature distribution in a heat transfer medium, an iterative stochastic optimization framework based on Eagle Strategy is proposed. Eagle strategy is the combination of crude global search and intensive local search mechanisms. In this Eagle Strategy concept, ameliorated manipulation equations of Barnacles Mating Optimizers are utilized as a global search mechanism while the ensemble of two Differential Evolution variants is considered as a local exploiter. Favorable merits of Levy flights and chaotic Lozi map are meticulously implemented on the related phases to enhance the total probing efficiency of the optimization algorithm. The created synergy between these complementary search mechanisms constructs the Eagle Strategy framework. The proposed method has been applied on forty benchmark problems composed of unimodal and multimodal test functions to verify its effectiveness on multidimensional benchmark problems. Comparative results between the literature optimizers reveal that the Eagle Strategy framework can maintain reasonable accuracy for test problems and outperforms the compared algorithms in most of the cases in terms of solution efficacy. Finally, a case study concerning transient heat conduction have been solved. One can conclude from this research study that the Eagle strategy optimization framework can be a promising alternative to the class of hybrid metaheuristic algorithms. Another remarkable conclusive outcome of this research study is that tedious drawbacks of root finding methods can be easily eliminated by converting the related equation into an optimization problem.

References

[1] X.S. Yang, "Nature-Inspired Metaheuristic Algorithms," Luniver Press, UK, 2008.

[2] X.S. Yang and S. Deb, "Cuckoo search via Levy Flights," in: *World Congress on Nature & Biologically Inspired Computing, IEEE Publications*, Coimbatore, India, 9-11 December 2009, pp. 210-214.

[3] J. Kennedy and R. Eberhart, "Particle Swarm Optimization". in: *Proceedings of IEEE International Conference on Evolutionary Computation*, Perth, WA, Australia, 27 November- 1 December 1998, pp. 69-73.

[4] E. Kaya, S. Korkmaz, M.A. Sahman and A.C. Cinar, "DEBOHID: A differential evolution based oversampling approach for highly imbalanced datasets," *Expert Syst. Appl.*, 169. 114482, 2021.

[5] M.A. Sahman, A.C. Cinar, I. Saritas, A. Yasar, "Tree-seed algorithm in solving real-life optimization problems," in: *IOP Conference Series: Materials Science and Engineering*, Vol. 675, No. 1, pp. 012030, 2019.

[6] M.A. Sahman, A.C. Cinar, "Binary tree-seed algorithms with S-shaped and V-shaped transfer functions," *Int. J. Int. Syst. Appl. Eng.*, 7 (2), pp. 111-117, 2019.

[7] F. Mohammadi and H. Abdi, "A modified crow search algorithm MCSA for solving economic load dispatch problem," *Appl. Soft. Comput.*, vol. 71, pp. 51-65, 2018.

[8] A. Fathy, M. Abd-Elaziz and A.G. Alharbi, "A novel approach based on hybrid vortex search algorithm and differential evolution for identifying the optimal parameters of PEM fuel cell," *Renew. Eng.*, vol. 146, pp. 1833-1845, 2020.

[9] X.S. Yang, M. Karamanoglu, T.O. Ting and Y.X. Zhao, "Applications and analysis of bio-inspired eagle strategy for engineering optimization," *Neural Comput. Appl.*, vol. 25, pp. 411-420, 2014.

[10] X.S. Yang and S. Deb, "Two-stage eagle strategy with differential evolution," *Int. J. Bio-Inspir. Com.*, vol. 3, pp. 77-84, 2011.

[11] X.S. Yang and A.H. Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Eng. Comput.*, vol. 29, no. 5, pp. 464-483, 2012.

[12] Z.W. Geem, J.H. Kim and G.V. Loganathan, "A new heuristic algorithm: Harmony search," *Simulation*, vol. 76, pp. 60-68, 2001.

[13] A. Kaveh and S. Talathari, "A novel metaheuristic optimization method: charged system search," *Acta Mech.*, vol. 213, pp. 267-286, 2010.

[14] D. Simon, "Biogeography-Based Optimization," *IEEE Trans. Evol. Comput.*, vol. 12, pp. 702-713, 2008.

[15] L. Bianchi, M. Dorigo, L.M. Gambardella and W.J. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," *Natural Comput.*, vol. 8, pp. 239-287, 2009.

[16] D.H. Wolpert and W.G. Macready, "No Free Lunch Theorems for Optimization," *IEEE Trans. Evol. Comput.*, vol. 1, pp. 67, 1997.

[17] K. Sörensen, "Metaheuristics-the metaphor exposed," *Int. Trans. Oper. Res.*, vol. 22, pp. 3-18, 2015.

[18] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, pp. 268-308, 2003.

[19] N. Singh, L.H. Son, F. Chiclana and J.P. Magnot, "A new fusion of salp swarm with sine cosine for optimization of nonlinear functions," *Eng. Comput.*, vol. 36, pp. 185-212, 2020.

[20] I.B. Aydilek, "A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems," *Appl. Soft. Comput.*, vol. 66, pp. 232-249, 2018.

[21] N.N. Son, C.V. Kien and H.P.H. Anh, "Parameters identification of Bouc-Wen hysteresis model for piezoelectric actuators using hybrid adaptive differential evolution and Jaya algorithm," *Eng. Appl. Artif. Intel.*, vol. 87, 103317, 2020.

[22] G. Dhiman, "ESA: a hybrid bio-inspired metaheuristic optimization approach for engineering problems," *Eng. Comput.*, vol. 37, pp. 323-353, 2021.

[23] X. Zhang, Q. Kang and X. Wang, "Hybrid biogeography-based optimization with shuffled frog leaping algorithm and its application to minimum spanning tree problems," *Swarm Evol. Comput.*, vol. 49, pp. 245-265, 2019.

[24] X.S. Yang and S. Deb, "Eagle strategy using Levy walk and firefly algorithms for stochastic optimizatio," ***in: J.R. Gonzalez et al. (Eds), Nature Inspired Cooperative Strategies for Optimization***, NICSO 2010, vol. 284, pp. 101-111, 2010.

[25] T. Storn and K. Price, "Differential Evolution – simple and efficient heuristic for global optimization over continuous spaces,"*J. Global Optim.*, vol. 11, pp. 341-359, 1997.

[26] M.H. Sulaiman, Z. Mustaffa, M.M. Saari and H. Daniyal, "Barnacles Mating Optimizer: A new bio-inspired algorithm for solving engineering optimization problems," *Eng. Appl. Artif. Intel.*, vol. 87, 103330, 2020.

[27] R. Lozi, "Un attracteur etrange? du type attracteur de Henon," *J. Phys.*. vol. 39, no. 5, pp. 9-10, 1978.

[28] A.H. Gandomi, X.S. Yang, S. Talathari and S. Deb, "Coupled eagle strategy and differential evolution for constrained global optimization," *Comput. Math. Appl.*, vol. 63, pp. 191-200, 2012.

[29] S. Talathari, A.H. Gandomi, X.S. Yang, S. Deb, "Optimal design of frame structures using the Eagle Strategy with Differential Evolution," *Eng. Struct.*, vol. 91, pp. 16-25, 2015.

[30] H. Yapıcı and N. Çetinkaya, "An Improved Particle Swarm Optimization Algorithm Using Eagle Strategy for Power Loss Minimization," *Math. Probl. Eng.*, 1063045, 2017.

[31] K.G. Dhal, A. Namtirtha, M.I. Quraishi and S. Das, "Grey level image enhancement using Particle Swarm Optimization with Levy Flight: An Eagle Strategy Approach," *IJIRSET*, vol. 5, pp. 79-86, 2016.

[32] S. Xu, Y. Wang and Z. Wang, "Parameter estimation of proton exchange membrane fuel cells using eagle strategy based on JAYA algorithm and Nelder-Mead simplex method," *Energy*, vol. 173, pp. 457-467, 2019.

[33] A.M. Reynolds and C.J. Rhodes, "The Levy flight paradigm: random search patterns and mechanisms," *Ecology*, vol. 90, pp. 877-887, 2009.

[34] S.W. Guo and E.A. Thompson, "Performing the exact test of Hardy-Weinberg proportion for multiple alleles," *Biometrics*, vol. 48, pp. 361-372, 1992.

[35] M. Baranzadeh, C.S. Davis, C.J. Neufeld, D.W. Coltman and A.R. Palmer, "Something Darwin didn't know about barnacles: sperm cast mating in a common stalked species," *Proc. R. Soc. Lond. [Biol]*, vol. 285, 20122919, 2013.

[36] R. Storn, "On the usage of differential evolution for function optimization," *in: Proceedings of North American Fuzzy Information Processing*, IEEE, Berkeley, CA, USA, 1996.

[37] S. Das and P.N. Suganthan, "Differential evolution: A survey of the state-of-art," *IEEE Trans. Evol. Comput.*, vol. 15, pp. 4-31, 2011.

[38] S. Das, S.S. Mullick and P.N. Suganthan, "Recent advances in differential evolution-An updated survey," *Swarm Evol. Comput.*, vol. 27, pp. 1-30, 2016.

[39] A.K. Qin, V.L. Huang and P.N. Suganthan, "Differential Evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, pp. 398-417, 2009.

[40] W. Gong, A. Fialho, Z. Cai and H. Li, "Adaptive strategy selection in differential evolution for numerical optimization: An empirical study," *Inf. Sci.*, vol. 181, pp. 5364-5386, 2011.

[41] M.Z. Ali, H. Awad and P.N. Suganthan, "Multi-population differential evolution with balanced ensemble of mutation strategies for large scale global optimization," *Appl. Soft. Comput.*, vol. 33, pp. 304-327, 2015.

[42] X. Li, P. Niu and J. Liu, "Combustion optimization of a boiler based on the chaos and Levy flight vortex search algorithm," *Appl. Math. Model.*, vol. 58, pp. 3-18, 2018.

[43] I. Aydogdu, A. Akın and M.P. Saka, "Design optimization of real-world steel space frames using artificial bee colony algorithm with Levy flight distribution," *Adv. Eng. Softw.*, vol. 92, pp. 1-14, 2016.

[44] M. Chawla and M. Duhan, "Levy Flights in Metaheuristics Optimization Algorithms – A review," *Appl. Artif. Intel.*, vol. 32, pp. 802-821, 2018.

[45] R. Caponetto, L. Fortuna, S. Fazzino and M. Xibilia, "Chaotic sequences to improve the performance of evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, pp. 289-304, 2003.

[46] M. Hennon, "A two-dimensional mapping with a strange attractor," *Commun. Math. Phys.*, vol. 50, pp. 69-77, 1979.

[47] P. Civicioglu, "Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm," *Comput. & Geosci.*, vol. 46, pp. 229-247, 2012.

[48] P. Civicioglu, "Artificial cooperative search algorithm for numerical optimization problems," *Inf. Sci.*, vol. 229, pp. 58-76, 2013.

[49] W. Zhao, Z. Zhang and L. Wang, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications," *Eng. Appl. Artif. Intel.*, vol. 87, 103300, 2020.

[50] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja and H. Chen, "Harris hawks optimization: Algorithm and applications," *Futur. Gener. Comput. Syst.*, vol. 97, pp. 849-872, 2019.

[51] S. Mirjalili, “SCA: A Sine Cosine Algorithm for solving optimization problems,” *Knowl.-Based Syst.*, vol. 96, pp. 120-133, 2016.

[52] S. Arora and S. Singh, “Butterfly optimization algorithm: a novel approach for global optimization,” *Soft Comput.*, vol. 23, pp. 715-734, 2019.

[53] W.T. Pan, “A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example,” *Knowl.-Based Syst.*, vol. 26, pp. 69-74, 2012.

[54] S. Mirjalili and S.M. Mirjalili, “Lewis A Grey Wolf Optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46-61, 2014.

[55] G. Dhiman and V. Kumar, “Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications,” *Adv. Eng. Softw.*, vol. 114, pp. 48-70, 2017.

[56] A. Askarzadeh, “A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm,” *Comput. & Struct.*, vol. 169, pp. 1-12, 2016.

[57] S. Mirjalili, S.M. Mirjaili and A. Hatamlou, “Multi-verse optimizer: a nature-inspired algorithm for global optimization,” *Neural Comput. Appl.*, vol. 27, pp. 495-513, 2016.

[58] X.S. Yang and S. Deb, “Engineering optimization by Cuckoo Search,” *Int. J. Math. Model. Numer. Optim.*, vol. 1, pp. 330-343, 2010.

[59] N. Ozisik, “Heat conduction,” John Wiley & Sons, New-York, 1993.