International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 12, Issue 4 [April. 2023] PP: 188-192

A Brief Review on Some Recently Proposed Natural Inspired Algorithms

E. F. Kelash, M. A. Elsayed, D. A. Hammad, M. A. Elsisy
Basic Engineering Sciences Department, Benha Faculty of Engineering, Benha University, Benha, EGYPT
Corresponding Author: E. F. Kelash

ABSTRACT: *This paper provides a brief review of some natural inspired algorithms that have been proposed in recent years. The algorithms covered in this review include Fennec fox optimization, mountaineering team-based optimization, total interaction algorithm, and serval optimization algorithm. Each algorithm is briefly introduced, and their mathematical model and main procedure are discussed. This review aims to provide an overview of these algorithms and to highlight their potential for solving complex optimization problems. Overall, this review provides valuable insights for researchers and practitioners who are interested in natural inspired algorithms and their applications.*

Date of Submission: 06-04-2023 Date of acceptance: 18-04-2023

I. INTRODUCTION

Natural-inspired optimization is considered as one of the most active areas in optimization today. It refers to a class of optimization algorithms that are inspired by principles or phenomena found in nature. These algorithms are often used to solve complex optimization problems that are difficult or time-consuming to solve using traditional mathematical optimization techniques [1]. Natural-inspired optimization algorithms are based on the idea that nature has evolved efficient and effective strategies for solving optimization problems over millions of years of evolution, and that these strategies can be emulated or adapted to solve human-engineered problems. These algorithms are used to solve complex optimization and search problems and are often employed in fields such as artificial intelligence [2], optimization [3], data mining [4], robotics [5], and pattern recognition [6]. These algorithms can be classified into five classes based on the area of inspiration [3]. 1) Swarm intelligence algorithms, 2) Evolutionary algorithms, 3) Human-based algorithms, 4) Bio inspired (not swarm based) algorithms, and 5) Physics-based algorithms.

Swarm intelligence algorithms are a type of computational optimization algorithm that are inspired by the collective behavior of social insects, such as ants, bees, and termites, which exhibit complex and coordinated behavior without central control. Swarm intelligence algorithms are used to solve complex problems, optimize systems, and make decisions by mimicking the self-organizing and decentralized behavior of swarms [3]. Examples of popular swarm intelligence are particle swarm optimization (PSO) [7], ant colony optimization (ACO) [8] and fish school search (FSS) [9].

Evolutionary algorithms are a family of optimization algorithms inspired by the principles of biological evolution [10]. The main idea behind evolutionary algorithms is to mimic the natural process of evolution by iteratively generating new candidate solutions, evaluating their fitness based on a predefined objective function, and selecting the fittest individuals for reproduction to create the next generation of solutions. The process continues until a satisfactory solution is obtained. The most common types of evolutionary algorithms include genetic algorithms [11], evolutionary programming [12], evolutionary strategies [13], and genetic programming [14].

Human search algorithms are a class of algorithms that rely on human intelligence and intuition to search for solutions to complex problems. Unlike machine-based algorithms, which rely on computational power and efficiency, human search algorithms rely on human creativity and problem-solving skills [15]. These algorithms include driving training-based optimization (DTBO) [16], chef-based optimization algorithm (CBOA) [17] and tabu search algorithm (TSA) [18].

Bio-inspired algorithms are a class of optimization algorithms that are inspired by natural processes and biological systems. Unlike swarm-based algorithms that simulate the collective behavior of a group of individuals, bio-inspired algorithms focus on the individual behavior of organisms and their interaction with the environment [19]. Examples are: flower pollination algorithm (FPA) [20], Egyptian vulture algorithm (EVA) [21], and Paddy field algorithm (PFA) [22].

Physics-inspired optimization algorithms are a class of algorithms that are based on the principles of physics and natural laws. These algorithms are inspired by the way particles move and interact in physical systems, and they use mathematical models to simulate these behaviors in order to find optimal solutions to complex optimization problems. This category includes Gravitational Search Algorithm (GSA) [23], simulated annealing (SA) [24], and vortex search algorithm (VSA) [25].

In this paper we focus on recently proposed metaheuristic algorithms used in the field of optimization. Each of the following sections discusses one of the targeted algorithms.

II. FENNEC FOX OPTIMIZATION

Fennec Fox Optimization (FFA) is a recently proposed nature-inspired algorithm that mimics two behaviors of the fennec fox, a small nocturnal fox found in the Sahara Desert [26]. The algorithm is designed to optimize complex problems by mimicking the hunting behavior of the fennec fox, which is known for its ability to locate prey in the harsh desert environment. The algorithm is inspired by the ability of digging to capture its prey and the strategy of escaping when attacked by wild predators. FFA is a population-based algorithm. In this algorithm, foxes construct search groups among themselves with the aim to catch prey. Each fox represents one candidate in the search space. Initially, the fennec foxes' positions are randomly generated as in equation (1).

$$X_i: x_{i,j} = lb_j + rand.\left(ub_j - lb_j\right), \quad i = 1, \dots, N, j = 1, \dots, m \tag{1}$$

where X_i denotes the i^{th} fox, $x_{i,j}$ represents its dimension, N is the number of candidates, m is the number of decision variables, $rand$ is a random number in the interval [0, 1]. lb_j and ub_j represents the lower and upper bounds of the j^{th}, respectively.

As known in metaheuristic algorithms, the search process consists of two main phases: exploration and exploitation. In the exploration phase, the algorithm mimics the escaping behavior of the Fennec fox when attacked by wild predators like caracals and striped hyenas. This species of foxes can change their direction of movement suddenly in addition to their high speed. Depending on this escape strategy, the candidate which represents a Fennec fox can cover all the search space escaping from being captured in local optimal solutions.

In the exploitation phase, the algorithm simulates the digging behavior of the Fox to catch its prey. The Fennec fox has long ears which enable it to hear and find the exact location of the prey beneath sand. This behavior gives the algorithm a high locality. Thus, it provides power to the exploitation process. The authors of [26] have applied the algorithm to 68 benchmark problems in addition to a number of engineering optimization problems such as tension compression spring design optimization problem, and welded beam design problem. The results show a promising performance of the FFA when compared to a set of famous metaheuristic algorithms such as GA, PSO, and TLBO. The main procedure as proposed in [26] is shown in Fig. 1.

Step 1: Identify the search space and generate initial population.
Step 2: Evaluate the objective value based on the initial population.
Step 3: Find the neighborhood radius of the i^{th} fox in the j^{th} dimension.
Step 4: Find the new suggested status of the i^{th} fox.
Step 5: Update the position of the i^{th} fox.
Step 6: Find a random position for the i^{th} fox representing the escaping phase.
Step 7: Evaluate the objective function.
Step 8: Update the status of the i^{th} fox.
Step 9: Return to step 3, until reaching a termination condition.

Fig. 1 The main procedure of the FFA

III. MOUNTAINEERING TEAM-BASED OPTIMIZATION

Mountaineering team-based optimization (MTBO) is a novel optimization technique for dealing with complicated optimization problems. The algorithm was first presented in 2023 [27]. It is considered as one of the human-based algorithms as it mimics human intelligence and intuition to search for solutions to complex problems. The algorithm relies on the cooperative behavior of humans and the ability to communicate among themselves to reach the top of a mountain. Mountain top in real life represents the global solution of a certain objective function in optimization. As stated in [27], the algorithm has many advantages. Some of them are: 1) Compared to other popular metaheuristics, the algorithm has a superiority in terms of the performance and the goodness of the reached solutions, 2) The algorithm depends on simple concepts which make it easy to be implemented, 3) The convergence rate of the MTBO is high when compared to some other algorithms.

A mountaineering team is a group of humans aiming to reach the top of a mountain in some region. The leader of this team is the most experienced member of the group. As aforementioned, the mountaintop is considered as the global optimal solution to the handled problem. The populations of the algorithm are a set of mountaineers. The algorithm consists of four phases as follows:

The first phase is dedicated to coordinate mountaineering. In this phase, the most experienced member of the group is chosen to be the leader. This member is considered as the best solution for the search space in the optimization problem. This member guides the search towards the best global solution. Therefore, all members change their positions to the position of leader as formulated in (2).

$$X_i^{new} = X_i + rand \times (X_{leader} - X_i) \tag{2}$$

In addition, the members in the group are ordered from the best to the worst. Therefore, each member in the group is guided not only by the leader, but also by the member in front. (2). Can be reformulated to be:

$$X_i^{new} = X_i + rand \times (X_{leader} - X_i) + rand \times (X_{ii} - X_i) \tag{3}$$

In the second phase, the effect of nature disasters on the search process is considered. Nature disasters are life threating for the members of the group, in addition they can obstruct the access to the desired mountaintop. In optimization, this means that the algorithm could be captured in local optima instead of reaching the global one. The behavior of members depends on the human intelligence, and they try to escape from avalanches worst case X_{worst} as possible. The algorithm can escape avalanches or in other words local optima using the formula:

$$X_i^{new} = X_i - rand \times (X_{worst} - X_i) \tag{4}$$

In the third phase, the members coordinate among themselves to overcome nature disasters. Depending on the cooperative behavior of humans, when a disaster occurs, all the team try to save the trapped member. The reason is that the performance of the algorithm depends on the average of the positions of all members X_{avg}, thus the i^{th} member is guided toward the average position using the formula:

$$X_i^{new} = X_i + rand \times (X_{avg} - X_i) \tag{5}$$

The main procedure of the algorithm is shown in Fig. 2.

Step 1: Identify the search space and generate initial population.
Step 2: Evaluate the objective value based on the initial population.
Step 3: Move the population to new positions using equations 2 to 5.
Step 4: Evaluate the objective function depending on the new positions of members.
Step 5: Update the position of the i^{th} member with worst contributions.
Step 6: Return to step 3, until reaching a termination condition.

Fig. 2 The main procedure of the MTBO

IV. TOTAL INTERACTION ALGORITHM

Total interaction algorithm (TIA) is a novel simple swarm metaheuristic [28]. The algorithm basic idea is the ability of the population agents to interact among themselves. In each iteration, each agent interacts with all other agents with the aim to reach the possible global optimal solution. In the initial step, the population candidates are generated randomly using a uniform distribution in terms of the lower and upper bounds of the search space as in (6).

$$X = U(lb, ub) \tag{6}$$

In the second step, the algorithm starts to replace the current solution with the newly reached solution in the new iteration if the new solution is better than the current best solution as in (7).

$$X_b' = \begin{cases} X, & if\ f(X) < f(X_b) \\ X_b, & otherwise \end{cases} \tag{7}$$

The movement of the regarded candidate has two possibilities. The first one is to move closer to the other solution if the other solution is closer to the global optimum, otherwise it will go away it. This behavior is formulated as:

$$X_{c,i,j} = \begin{cases} X_i + r_1.\left(X_j - r_2.X_i\right), & if\ f\left(X_j\right) < f(X_i) \\ X_j + r_1.\left(X_i - r_2.X_j\right), & otherwise \end{cases} \tag{8}$$

After that, the solution with the highest quality is chosen to be the new candidate as stated in (9).

$$X_{cc,i'} = \begin{cases} X_{c,i,j}, & if\ f(X_{c,i,j}) < f(X_{cc,i}) \\ X_{cc,i}, & otherwise \end{cases} \quad (9)$$

Finally, the same procedure will be used to decide if the final agent will replace the current solution.

V. SERVAL OPTIMIZATION ALGORITHM

Serval optimization algorithm (SOA) is a new bio-inspired algorithm for solving optimization problems [29]. The algorithm mimics the hunting process of serval in nature. It is known that the serval hunts and kills its prey in three stages. Firstly, it uses its good hearing ability to identify the prey's position and keeps calm for 15 minutes observing it. After that, it starts to attack the prey by moving towards it and jumping in the air towards it. The final stage is a chasing phase at which the serval runs and jumps to catch the fleeing prey and finally killing it and feeding on it. This behavior inspired the authors of [29] as it can provide a good explorative and exploitative ability.

First, the algorithm is initialized. The servals represent the population of the algorithm. Each serval denotes a candidate solution for the problem. The algorithm consists of two steps which are exploration and exploitation. The exploration step represents the prey selection and attacking phase. The selection of the prey depending on the good hearing ability of servals, then attacking it are the two processes which guide the exploration phase of the algorithm. In this step, big changes in the serval's position occur. This big change helps the algorithm to go through all the search space increasing the power of exploration of the algorithm. Serval position is updated using (10).

$$x_{i,j}^{p1} = x_{i,j} + r_{i,j}.\left(P_j - I_{i,j}.x_{i,j}\right),\ i = 1,2,\ldots,N, and\ j = 1,2,\ldots,d \quad (10)$$

If the new position of the serval updates the fitness function, then the previous position is replaced as:

$$X_i = \begin{cases} X_i^{p1}, & if\ f(X_i^{p1}) < f(X_i) \\ X_i, & otherwise \end{cases} \quad (11)$$

where X_i^{p1} represents the new position of the i^{th} serval in the first phase of the algorithm, $r_{i,j}$ is a random number between 0 and 1, $I_{i,j}$ is either 1 or 2, N denotes the total number of servals, and d represents the dimension of the problem.

In the SOA, the exploitation process is represented by the chase and kill phase. After the attack happens, the serval makes several tries to control the prey by chasing it. This chase process causes small changes in the position of the serval which means small changes in the solution which increase the exploitative power of the algorithm. The new position of the serval can be formulated as:

$$x_{i,j}^{p2} = x_{i,j} + \frac{r_{i,j}(ub_j - lb_j)}{t},\ i = 1,2,\ldots,N, j = 1,2,\ldots,d, and\ t = 1,2,\ldots,T \quad (12)$$

If the new position improves the fitness function, it will replace the old one using (13).

$$X_i = \begin{cases} X_i^{p2}, & if\ f(X_i^{p2}) < f(X_i) \\ X_i, & otherwise \end{cases} \quad (13)$$

The main procedure of the algorithm is shown in Fig. 3.

Step 1: Identify the search space and generate initial population.
Step 2: Evaluate the objective value based on the initial population.
Step 3: Update the best member to identify the prey location.
Step 4: Start the exploration phase: Calculate the new position of the i^{th} serval and evaluate the objective function.
Step 5: Start the exploitation phase: Calculate the new position of the i^{th} serval and evaluate the objective function.
Step 6: Return to step 3, until reaching a termination condition.

Fig. 2 The main procedure of the SOA

VI. CONCLUSION

This paper has provided a comprehensive review of some of the natural inspired algorithms that have been proposed in recent years. The algorithms covered in this review are diverse, and they include Fennec fox optimization, mountaineering team-based optimization, total interaction algorithm, and serval optimization algorithm. Each algorithm has been briefly introduced, and their mathematical model and main procedure have been discussed. The review has also highlighted the potential of these algorithms in solving complex optimization problems.

REFRENCES

[1]. Zang, H., Zhang, S., & Hapeshi, K. (2010). A review of nature-inspired algorithms. Journal of Bionic Engineering, 7(4), S232-S237.

[2]. Naganna, S. R., Deka, P. C., Ghorbani, M. A., Biazar, S. M., Al-Ansari, N., & Yaseen, Z. M. (2019). Dew point temperature estimation: application of artificial intelligence model integrated with nature-inspired optimization algorithms. Water, 11(4), 742.

[3]. Fister Jr, I., Yang, X. S., Fister, I., Brest, J., & Fister, D. (2013). A brief review of nature-inspired algorithms for optimization. arXiv preprint arXiv:1307.4186.

[4]. Hemeida, A. M., Alkhalaf, S., Mady, A., Mahmoud, E. A., Hussein, M. E., & Eldin, A. M. B. (2020). Implementation of nature-inspired optimization algorithms in some data mining tasks. Ain Shams Engineering Journal, 11(2), 309-318.

[5]. Karpenko, A. P., & Leshchev, I. A. (2019). Nature-inspired algorithms for global optimization in group robotics problems. Smart Electromechanical Systems: Group Interaction, 91-106.

[6]. Jino Ramson, S. R., Lova Raju, K., Vishnu, S., & Anagnostopoulos, T. (2019). Nature inspired optimization techniques for image processing—A short review. Nature Inspired Optimization Techniques for Image Processing Applications, 113-145.

[7]. Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). IEEE.

[8]. Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. IEEE computational intelligence magazine, 1(4), 28-39.

[9]. Bastos Filho, C. J., de Lima Neto, F. B., Lins, A. J., Nascimento, A. I., & Lima, M. P. (2008, October). A novel search algorithm based on fish school behavior. In 2008 IEEE international conference on systems, man and cybernetics (pp. 2646-2651). IEEE.

[10]. Yu, X., & Gen, M. (2010). Introduction to evolutionary algorithms. Springer Science & Business Media.

[11]. Holland, J. H. (1992). Genetic algorithms. Scientific american, 267(1), 66-73.

[12]. Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. IEEE Transactions on Evolutionary computation, 3(2), 82-102.

[13]. Beyer, H. G., & Schwefel, H. P. (2002). Evolution strategies–a comprehensive introduction. Natural computing, 1, 3-52.

[14]. Koza, J. R. (1994). Genetic programming II (Vol. 17). Cambridge: MIT press.

[15]. Reverdy, P. B. (2014). Human-inspired algorithms for search A framework for human-machine multi-armed bandit problems (Doctoral dissertation, Princeton University).

[16]. Dehghani, M., Trojovská, E., & Trojovský, P. (2022). A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process. Scientific reports, 12(1), 9924.

[17]. Trojovská, E., & Dehghani, M. (2022). A new human-based metahurestic optimization method based on mimicking cooking training. Scientific Reports, 12(1), 14861.

[18]. Gallego, R. A., Romero, R., & Monticelli, A. J. (2000). Tabu search algorithm for network synthesis. IEEE Transactions on Power Systems, 15(2), 490-495.

[19]. Fan, X., Sayers, W., Zhang, S., Han, Z., Ren, L., & Chizari, H. (2020). Review and classification of bio-inspired algorithms and their applications. Journal of Bionic Engineering, 17, 611-631.

[20]. Yang, X. S. (2012). Flower pollination algorithm for global optimization. In Unconventional Computation and Natural Computation: 11th International Conference, UCNC 2012, Orléan, France, September 3-7, 2012. Proceedings 11 (pp. 240-249). Springer Berlin Heidelberg.

[21]. Sur, C., Sharma, S., & Shukla, A. (2013). Egyptian vulture optimization algorithm–a new nature inspired meta-heuristics for knapsack problem. In The 9th International Conference on Computing and InformationTechnology (IC2IT2013) 9th-10th May 2013 King Mongkut's University of Technology North Bangkok (pp. 227-237). Springer Berlin Heidelberg.

[22]. Premaratne, U., Samarabandu, J., & Sidhu, T. (2009, December). A new biologically inspired optimization algorithm. In 2009 international conference on industrial and information systems (ICIIS) (pp. 279-284). IEEE.

[23]. Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. Information sciences, 179(13), 2232-2248.

[24]. Bertsimas, D., & Tsitsiklis, J. (1993). Simulated annealing. Statistical science, 8(1), 10-15.

[25]. Doğan, B., & Ölmez, T. (2015). A new metaheuristic for numerical function optimization: Vortex Search algorithm. Information sciences, 293, 125-145.

[26]. Trojovská, E., Dehghani, M., & Trojovský, P. (2022). Fennec Fox Optimization: A New Nature-Inspired Optimization Algorithm. IEEE Access, 10, 84417-84443.

[27]. Faridmehr, I., Nehdi, M. L., Davoudkhani, I. F., & Poolad, A. (2023). Mountaineering Team-Based Optimization: A Novel Human-Based Metaheuristic Algorithm. Mathematics, 11(5), 1273.

[28]. Kusuma, P. D., & Novianty, A. (2023). Total Interaction Algorithm: A Metaheuristic in which Each Agent Interacts with All Other Agents. International Journal of Intelligent Engineering and Systems, 16(1), 224-234.

[29]. Dehghani, M., & Trojovský, P. (2022). Serval Optimization Algorithm: A New Bio-Inspired Approach for Solving Optimization Problems. Biomimetics, 7(4), 204.