JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Overview Of the Adaptive Coati Optimization Algorithm

T.Nishitha[1],Dr .Akhil Khare[2]
Ph.D Scholar, Dept of CSE, Osmania University.
Professor of CSE, MVSR Engineering College, Osmania University.

Abstract
Adaptive Coati Optimization algorithm is a new nature-inspired optimization algorithm based on the behavior of coatis in their search for food. The algorithm shows the social behavior of coati's where individuals share the location of sources of food. This sharing helps guide the population toward promising areas of the search space. Coatis adjust their movement patterns based on the success of previous searches. This algorithm uses a collective approach where multiple coatis work together to solve complex optimization problems.
Keywords: optimization, search of food, search space.

Introduction

The Adaptive Coati Optimization Algorithm (ACOA) is an innovative optimization approach based on the behavior of coatis in their search for food. ACOA improves the configuration by imitating the adaptive hunting and defensive systems of coatis, which are inspired by their strategic behaviors. This ensures that ACOA has better learning and prediction capabilities. Initialize a population of coatis each representing a good solution to the optimization problem. Distribute coatis randomly within the search space. Based on a predefined objective function that reflects the quality of the solution evaluate the fitness of each coati. Coatis share information about their positions and the corresponding fitness values. Based on the shared information, each coati decides its next move using a probabilistic rule, considering its own position and best-known positions of other coatis.

According to adaptive strategies, coatis move through work space that may involve random walks, guided movements towards better positions. The movement is adjusted dynamically, with a balance between exploration and exploitation. After each iteration, update the best-known solutions based on the coatis current positions and fitness values. Adapt the algorithm parameters dynamically to maintain an optimal balance between exploration and exploitation. The algorithm run continuously until a specified stopping condition is reached. Due to its flexibility and adaptability, this algorithm can be applied to a wide range of optimization problems.

The algorithm balances exploration and exploitation dynamically, that allows for efficient search and convergence to optimal or near-optimal solutions. It can handle large search spaces effectively and so suitable for high-dimensional problems.

I. Related works

Introducing the Adaptive Coati Optimization Algorithm (ACOA). It is a swarm intelligence optimization technique that give natural behaviors of the animals. This algorithm draws inspiration from the foraging behavior of honeybees. Artificial Bee Colony (ABC) Algorithm optimizes solutions by simulating the search for food [4], a concept elucidated by Karaboga, D [4]. Inspired by the food-foraging behavior of squirrels [9], a concept elucidated by Jain, M., Singh, V., & Chhabra, J. K. [9]. In the realm of coatis searching for food, two distinct roles are observed:

In the Figure 1 presented below, the coatis utilize their small ears, little black paws, and their lengthy no prehensile tail for balance and signaling. One of the coatis favourite preys is the green iguana. In groups, coatis will follow iguanas because they are typically spotted in trees.While some swiftly attack it, others climb trees in an attempt to scare the iguana into collapsing. Coatis are nevertheless susceptible to raptor attacks. Jaguarundis, anacondas, tayras, dogs, foxes, boa constrictor snakes, maned wolves, and jaguarundis are some of the predators of coati. The subsequent section elaborates on how the behaviours of these two groups are translated into mathematical equations.

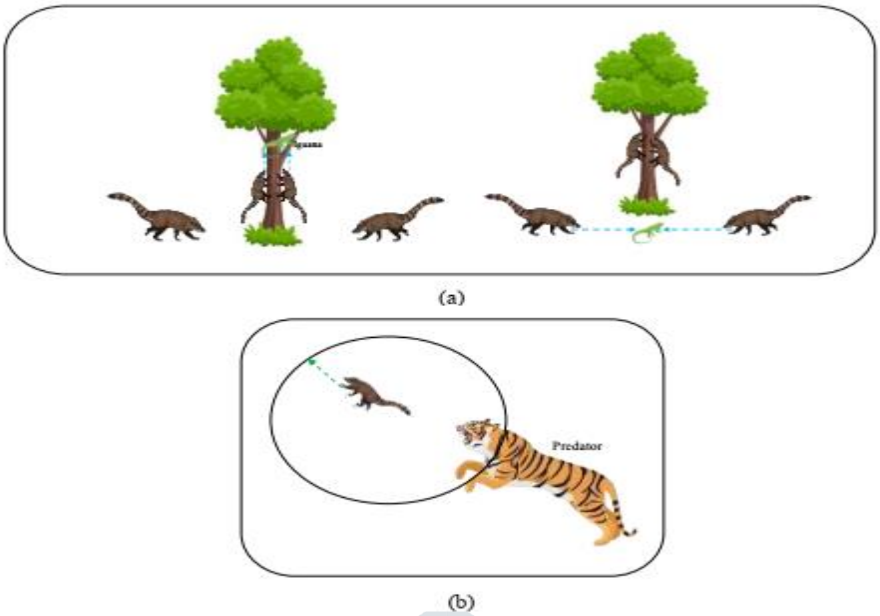

Fig. 1 (a) Attacking behaviour and (b) Escaping behaviour of coati

$$X=\begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m}=\begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & . & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,m} \\ \vdots & . & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,m} \end{bmatrix}_{N\times m} \quad (1)$$

$$X_i: \quad x_{i,j}=lb_j+r\cdot(ub_j-lb_j), \quad i=1,2,\ldots,N, \quad j=1,2,\ldots,m, \quad (2)$$

$$F=\begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1}=\begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \quad (3)$$

The matrix X represents randomly generated initial positions of the population as shown in Eq. (1).Here N denotes the number of coatis, M represents the number of dimensions in the problem space. The position of coati i defined by Eq. (2).Where lb_j and ub_j are lower and upper values of the decision and r is the random variable between the interval 0 and 1. The objective function of each position is calculated and formed a vector F with dimension NX1 as shown in Eq. (3).

$$X_i^{P1}: x_{i,j}^{P1}=x_{i,j}+r\cdot\left(\text{Iguana}_j-I\cdot x_{i,j}\right), \text{ for } i=1,2,\ldots,\lfloor\tfrac{N}{2}\rfloor \text{ and } j=1,2,\ldots,m \quad (4)$$

$$\text{Iguana}^G: \text{Iguana}_j^G=lb_j+r\cdot(ub_j-lb_j), j=1,2,\ldots,m, \quad (5)$$

$$X_i^{P1}: x_{i,j}^{P1}=\begin{cases} x_{i,j}+r\cdot\left(\text{Iguana } a_j^G-I\cdot x_{i,j}\right), & F_{\text{Iguana}}<F_i, \\ x_{i,j}+r\cdot\left(x_{i,j}-\text{Iguana}_j^G\right), & \text{else} \end{cases} \quad (6)$$
$$\text{for } i=\lfloor\tfrac{N}{2}\rfloor+1,\lfloor\tfrac{N}{2}\rfloor+2,\ldots,N \text{ and } j=1,2,\ldots,m$$

$$X_i=\begin{cases} X_i^{P1}, F_i^{P1}<F_i \\ X_i, \text{ else} \end{cases} \quad (7)$$

$$lb_j^{\text{local}}=\frac{lb_j}{t}, ub_j^{\text{local}}=\frac{ub_j}{t}, \text{ where } t=1,2,\ldots,T \quad (8)$$

$$X_i^{P2}: x_{i,j}^{P2}=x_{i,j}+(1-2r)\cdot\left(lb_j^{\text{local}}+r\cdot\left(ub_j^{\text{local}}-lb_j^{\text{local}}\right)\right), \quad i=1,2,\ldots,N, j=1,2,\ldots,m, \quad (9)$$

$$X_i=\begin{cases} X_i^{P2}, F_i^{P2}<F_i \\ X_i, \text{ else} \end{cases} \quad (10)$$

Exploration Phase:

During the exploration phase, the coatis are divided into two groups. The first group climbs a tree to frighten their prey while the second group waits below the tree to catch the prey as it falls. This behavior allows the coatis to thoroughly explore the problem space. The prey here is referred as the Iguana and the position of the best individual in the population is considered the position of the Iguana. The positions of the first group of coatis, which climb the tree are determined by Eq. (4). When the Iguanas fall to the ground, their positions are randomized according to Eq. (5).

The second group of coatis which waits under the tree, moves based on Eq. (6). If the new position of a coati offers better fitness value than its previous position, it is updated otherwise it retains its old position as specified in Eq. (7).

Exploitation Phase:
When a predator attacks a coati, the coati moves to a new random position nearby as described in Eq. (8). and Eq. (9). This new position is only accepted if it results in an improved fitness value as indicated by Eq. (10). where suffix denotes the position and fitness of coati in the second phase.

Flowchart of the ACOA

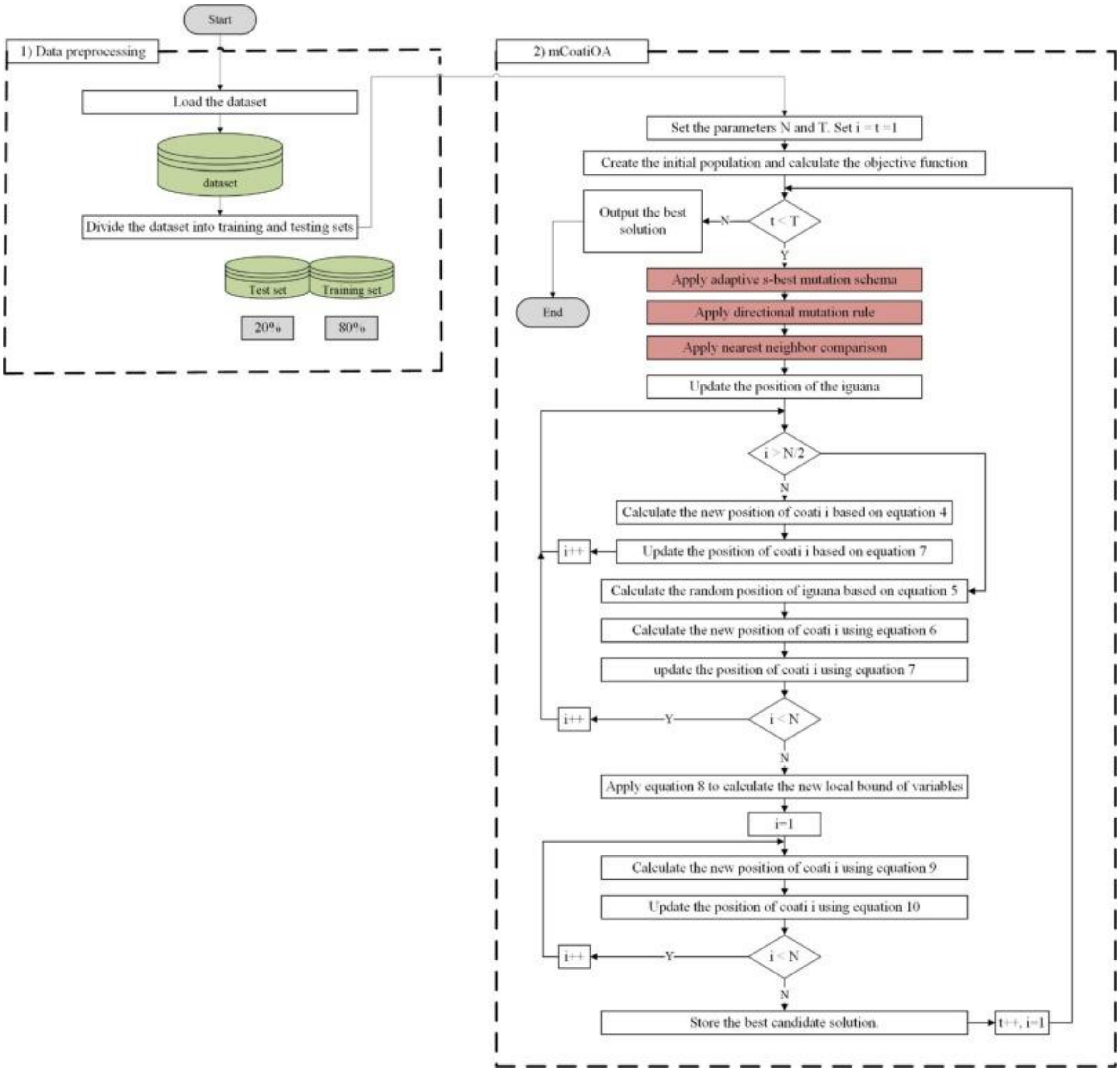

Fig. 2 Flowchat of ACOA

Algorithm of ACOA

```
Initialization:
• Initialize the CoatiOA parameters which are population size N and the number of
  iterations T.
• Initialize the problem parameters which are, problem dimensions D and lower and
  upper bounds lb, ub.
• Randomly initialize CoatiOA positions.
Compute the fitness function for the initial generation using Eq. (3).
while t ≤ T do
   update the iguana position based on the position of the best population.
   Phase1: exploration phase
   for each coati i in the first half of the populations do
      calculate the new position of coati i based on Eq. (4).
      update the position of coati i based on Eq. (7).
   end for
   for each qoati i in the second half of the populations do
      Calculate the random position of iguana based on Eq. (5).
      Calculate the new position of coati i using Eq. (6).
      update the position of coati i using Eq. (7).
   end for
   Phase2: exploitation phase
   Apply Eq. (8) to calculate the new local bound of variables.
   for each coati i in N do
      Calculate the new position of coati i using Eq. (9).
      Update the position of coati i using Eq. (10).
   end for
   Store the best candidate solution.
   Set iteration counter t = t + 1.
end while
output the best solution.
```

Fig. 3 Pseudocode of the Adaptive Coati Optimization Algorithm

II. CONCLUSION

In conclusion, the Adaptive Coati Optimization Algorithm stands out as an innovative optimization method, which offers an innovative method to solving complex optimization problems. Its ability to balance exploration and exploitation and its adaptive nature allow the algorithm to efficiently navigate large and diverse search spaces.

REFERENCES

[1]. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, Australia, 1942-1948. IEEE. doi:10.1109/ICNN.1995.488968

[2] Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: A new meta-heuristic. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, 2, 1470-1477. IEEE. doi:10.1109/CEC.1999.782657

[3] Holland, J. H. (1992). Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT Press.

[4] Karaboga, D. (2005). An Idea Based on Honey Bee Swarm for Numerical Optimization (Technical Report-TR06). Erciyes University, Engineering Faculty, Computer Engineering Department.

[5] Yang, X.-S., & Deb, S. (2009). Cuckoo search via Lévy flights. 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), 210-214. IEEE. doi:10.1109/NABIC.2009.5393690

[6] Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. In: J. R. González et al. (eds), Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Studies in Computational Intelligence, 284, 65-74. Springer, Berlin, Heidelberg.

[7] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. Advances in Engineering Software, 69, 46-61.

[8] Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. Proceedings of the 5th International Conference on Stochastic Algorithms: Foundations and Applications (SAGA 2009), 169-178. Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-04944-6_14

[9] Jain, M., Singh, V., & Chhabra, J. K. (2019). A novel nature-inspired algorithm for optimization: Squirrel search algorithm. Swarm and Evolutionary Computation, 44, 148-175. doi:10.1016/j.swevo.2018.02.013