American Journal of Integrated STEM Education, 2025, 8-17
https://publishingjournals.org/stem
Volume: 2 | Number: 1 (2025) January

Using Emperor Penguins Colony Algorithm (EPC) To Hybrid Archimedes Optimization Algorithm (AOA)To Solve Optimization Problems

Ayad Hamad Khalaf
Tikrit University

***Correspondence :** Ayad Hamad Khalaf
drayadlohebe@tu.edu.iq

Received: date
Accepted: date
Published: date

Abstract

This research paper included the study of two intelligent algorithms, the Emperor Penguin Colony algorithm and the Archimedes optimization algorithm. A hybrid algorithm called AOA-EPC was proposed, which included improving the Archimedes algorithm based on the penguin algorithm by making the two algorithms work together to reach the best solutions. The solutions are enhanced through both algorithms. The results of the proposed hybrid algorithm were excellent. It can avoid falling into local solutions in addition to the accuracy of the results. It can also reach the optimal solution in record time. Moreover, it can find the best solutions with the least number of swarm elements. The steps of the proposed algorithm were created, as well as the flowchart for it. A table of the results we obtained and the optimal approximation for most functions used by measuring numerical optimization was displayed. Ten functions were addressed, which can be generalized to the rest of the other functions. This paves the way for applying this algorithm to any function that can be used in engineering, statistics and other science applications, as it is an excellent algorithm. The results of the proposed algorithm AOA-EPC were compared with both the AOA and EPC algorithms, and the numerical results showed the superiority of the hybrid algorithm.

Keywords

Optimization, Intelligence Techniques, Swarm Intelligence, Heuristic Algorithms, Hybrid, Evolutionary Computation, Algorithms.

Introduction

The expansion of the base of use of applied mathematics was an incentive to innovate new applications that keep pace with the significant development at present so that these applications are highly efficient and can support this development. This, in turn, paves the way for the creation of modern mathematical methods and programs that, in turn, use high-performance computing to simulate natural phenomena and can solve various problems in various sciences and engineering. With this vast amount of data and issues facing life, it has become challenging to find optimal solutions to solve these problems and extract meaningful information. For this reason, several innovative algorithms have been established to help solve these problems. These intelligent algorithms in numerical optimization play an essential and significant role in finding feasible solutions to these problems. Several algorithms exist to solve numerical problems and engineering systems, including specific algorithms, random algorithms inspired by nature (Cuevas et al, 2020), and classical algorithms. Random algorithms are also divided into intuitive and heuristic (Malik et al, 2021) algorithms, designed based on cooperation between community members to reach a global solution) or goal. For example, these were used Algorithms by Mirjalili, Saydali is SCA algorithm (Mirjalili, 20166) to find optimal solutions. A large group of hybrid algorithms have been used in previous studies. An example of this is the sine and cosine algorithms with IWO and BA algorithm (Khalaf et al, 2021) and SCA- CG (Khalaf et al, 2020) and AOA [9] by KHalaf and Ban . It can be noted that optimization is one of the branches of computational sciences that seeks to answer the question of "what is best" and find the maximum or minimum value of a function. There is no single method available to solve optimization problems efficiently (Wahdan, 2020). Therefore, many optimization methods have been developed to solve different problems. An example is the mathematical programming techniques generally studied in operations research (Zhang, 2006).

Methods

Emperor Penguins Colony Algorithm

The emperor penguin is the tallest and heaviest species left on Earth and is restricted to Antarctica. It is one of the largest birds on Earth, with an individual reaching 122 centimeters in height and weighing between 22 and 45 kilograms. It lays eggs and follows a specific algorithm to protect them and its body. It takes 115 days for an egg to become a chick. To protect itself and the egg from the low temperatures and icy winds to which it is exposed and to avoid freezing to death during the incubation period, emperor penguins gather and clump together in tightly packed groups or masses. In the middle of this gathering, there is a lot of heat, and to use the heat by individuals and equalize their body temperatures, they make a spiral-like movement towards the center (Gilbert et al, 2008).

Mathematical Model for Emperor Penguin's Colony

The mechanism of designing and modeling the emperor penguin colony algorithm is based on the spiral motion of the emperor penguins. The body temperature and thermal radiation of the body are calculated, and then, given the distance and gravity, each penguin performs a spiral-like motion. This motion is called the spiral motion. At first, the position and cost of each penguin are calculated, and then the penguins' prices are compared with each other. The penguins always go to the penguin with the lowest absorption cost (high heat density). The heat density and distance determine this cost. All solutions are sorted, and the best one is chosen (Harifi, 2023).

Some rules for this algorithm:

1. Because of the absorption coefficient, all of the penguins in the first group are attracted to one another and radiate heat.
2. Every penguin's body surface area is the same.
3. Regardless of the atmosphere, the penguin absorbs all of the thermal radiation and the effects of the earth's surface.
4. Penguins radiate heat in a linear fashion.
5. The penguin is attracted according to the amount of heat.
6. To calculate the intensity of heat and its attraction, The transmission of heat must be computed, and the body surface area of each penguin must be determined in order to estimate the heat radiation of all of them (Harifi et al, 2019) (Al-Taie et al, 2022).

Body Surface Area

To calculate the body surface area, the area of the penguin's main body is first calculated using the following equation:

$$A_{trunk} = 2\pi *(ab/e)* \sin^{-1} *e + 2\pi b^2 \qquad (1.1)$$

e is calculated using the following equation:

$$e = sqrt(a^2 - b^2) \qquad (1.2)$$

The term "atrunk" describes the region of the main body, and "a" denotes the half of the body length from the neck where the penguin's feathers are white in the belly, or 0.34. B is equal to 0.16 and denotes the length of the secondary axes, or the main trunk's radius. Next, use the cone equation to determine the beak, as indicated by the equation below:

$$A_{beak} = \pi rs \qquad (1.3)$$

Since A beak is the area of the beak, r is half the largest cross-sectional area of the beak and is equal to 0.02, and S is the chord that is the length of the beak and is equal to 0.11.
The area of the head is calculated using the ball equation as shown in the following equation:

$$A_{head} = nd^2 - \pi r^2 \qquad (1.4)$$

Where Ahead represents the area of the head, d is the average height of the head, and r is above.
The fins are measured using the rectangle equation as in the following equation:

$$A_{fipper} = L Xw \qquad (1.5)$$

Where L represents the length, and W represents the width.

By placing the foot on 1 mm^3 of graph paper, One foot in touch with the ground has a major surface area of 0.0036 square meters, according to calculations of that region. The penguin often rests on the wrist joint, so this area is also calculated at 0.0006 m. T is the foot's thickness, which is calculated from the average thickness of the metatarsus and is equal to(0.014). Eventually, an emperor penguin's total body surface area becomes 0.56 m^2 .[13]

Heat Transfer

When two bodies have different temperatures, heat is a type of energy that is transported from one body to another or from one location to another. Heat transfer always occurs from the warmer body to the colder body and is accompanied by a change in the substance's intrinsic energy. Radiation is used to determine heat transfer, and the dispersed heat transfer model is taken into consideration when estimating an emperor penguin's heat exchange. Heat transmission from the trunk, head, fins, and feet is gathered in this model using the equation that follows:

$$total = q_{trunk} + q_{head} + q_{flippers} + q_{feet} \quad (1.6)$$

However, the radiation emitted by each part of the body must be calculated from the surface area and determined according to the following equation:

$$Q_{penguin} = A\, \mathcal{E} \sigma T^4{}_s \quad (1.7)$$

Where $Q_{penguin}$ represents the heat transfer per unit time, A represents the total surface area calculated in the previous subsection, which is 0.56 m2, Ɛ represents the emissivity of penguin feathers, which is equal to 0.98, σ represents Stefan's constant, and Ts is the absolute temperature in Kelvin, which is 35 °C and equal to 308.18 .[3]

Heat Intensity and Attractiveness

Photons transport energy and are absorbed by raising the electronic energy levels in radiation-induced heat transfer. Since photon sources can be classified as surface, point, or linear, they can be either concentrated or homogenous[7]. The heat source is linear in the emperor penguin colony algorithm. Photons do not slow down or stop due to matter but weaken and scatter. Therefore, to attenuate the photon, we need an equation that uses the linear source, and equation (8-1) explains this. The following equation is the photon attenuation equation:

$$I=I_0\, e^{-mx} \quad (1.8)$$

Where M refers to the attenuation coefficient, X is the distance between two linear sources, I0 is the initial heat density, and I refers to the heat density .

Finally, the gravity is calculated using the following equation:

$$Q = A\, \mathcal{E} \sigma T^4{}_s\, e^{-mx} \quad (1.9)$$

Coordinated Spiral Movements

In this case, the system's structure has uncertain boundaries in a spiral pattern around the centre, where the temperature is warmest in the cluster's centre and coldest at the periphery. The penguins do not crowd to gain an individual advantage. The entire cluster has a very slow spiral motion where each penguin has a turn in all positions. Suppose there are two types of penguins: i and j. The penguin that requires heat always gives way to the one that is warmer. Since penguin j is warmer in this instance, the spiral motion is from i to j. See Figure (6.1) a. The value of gravity Q measures the average distance penguin I travelled towards penguin j, and K is the new position of penguin i. The following equation can represent this spiral motion:

$$x_n = ae^{b\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}} + (Q)e^{b\tan^{-1}\frac{y_i}{x_i}}\right\}} \cos\left\{\frac{1}{b}\ln\left\{(1-Q) + (Q)e^{b\tan^{-1}\frac{y_i}{x_i}}\right\}\right\}$$

$$y_n = ae^{b\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}} + (Q)e^{b\tan^{-1}\frac{y_i}{x_i}}\right\}} \sin\left\{\frac{1}{b}\ln\left\{(1-Q) + (Q)e^{b\tan^{-1}\frac{y_i}{x_i}}\right\}\right\}$$

(1-10)

Since a and b are constants determined randomly, and the angle information is predetermined, the spiral motion can become orderly. It is better not to be limited to a monotonous spiral path, so there is a need to add a random vector to increase the variety and move it to a new location; the following equation illustrates this:

$$x_n = ae^{b\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}} + (Q)e^{b\tan^{-1}\frac{y_i}{x_i}}\right\}\cos\left\{\frac{1}{b}\ln\left\{(1-Q)+(Q)e^{b\tan^{-1}\frac{y_i}{x_i}}\right\}\right\}} + \Phi\,\epsilon\mathrm{i}$$

$$y_n = ae^{b\frac{1}{b}\ln\left\{(1-Q)e^{b\tan^{-1}\frac{y_i}{x_i}} + (Q)e^{b\tan^{-1}\frac{y_i}{x_i}}\right\}\sin\left\{\frac{1}{b}\ln\left\{(1-Q)+(Q)e^{b\tan^{-1}\frac{y_i}{x_i}}\right\}\right\}}$$

Where Φ represents the mutation factor in the path change, and ϵi represents the value of the random vector. The distribution in this algorithm is uniform.

Archimedes Optimization Algorithm (AOA)

Archimedes Optimization Algorithm Like any other community-based meta-intuitive algorithm, this algorithm starts with an initial population (candidate solutions) and random sizes, densities, and accelerations (Harifi, 2020).

At first, each individual is initialised with its random position in the fluid. Then, the fitness function for each individual in the initial population is calculated. Archimedes's Optimization Algorithm iterates until each iteration is completed. The density and size of each individual are updated, and the acceleration is updated based on the collision status with any other neighbouring individual. The updated density, size, and acceleration determine the individual's new position (Hashim et al, 2021) (Salah, 2022).

The following steps illustrate the mathematical model of the algorithm.

Step (1): Initialize the positions of all individuals using

$$o_k = lb + rand \times (ub - lb), \quad k = 1,2, \dots \dots, N \qquad (1.12)$$

Where o_k is the *kth* individual of a set of *N* individuals, *lb* and *ub* are the minimum and maximum search space, and the initial volume (*vol*) and density (*den*) of each individual are calculated as follows:

$$den_k = rand \qquad (1.13)$$

$$vol_k = rand \qquad (1.14)$$

Finally, configure the acceleration using

$$acc_k = lb + rand \times (ub - lb) \qquad (1.15)$$

This step evaluates the initial population and determines the organism with the best fitness.

Step (2): Updating the densities and volumes for each k and iteration t+1 is done using

$$den_k^{t+1} = den_k^t + rand \times (den_{best} - den_k^t) \qquad (1.16)$$

$$vol_k^{t+1} = vol_k^t + rand \times (vol_{best} - vol_k^t) \qquad (1.17)$$

Where den_{best} is the density of the best individual, vol_{best} is the size of the best individual so far, and *rand* is a random number (Harifi, 2021).

Step (3): The transfer process and the density factor

At first, the collision occurs between the individuals, and after some time, the individuals try to reach the equilibrium state with the help of the transfer operator TF, which converts the search from exploration to exploitation using

$$TF = \exp\left(\frac{t-T}{T}\right) \quad (1.18)$$

As TF increases gradually with time until it reaches 1, d is the density reduction factor and is calculated using

$$d^{t+1} = \exp\left(\frac{t-T}{T}\right) - \left(\frac{t-T}{T}\right) \quad (1.19)$$

Since d^{t+1} increases with time and allows convergence in the already defined region, properly handling this variable will ensure a balance between exploration and exploitation in this algorithm.

Step (4): (1) Exploration phase (collision between individuals)

If *TF≤0.5*, a collision between individuals occurs, determine a random substance (mr) and the acceleration for step t+1 using

$$acc_k^{t+1} = \frac{den_{mr} + vol_{mr} \times acc_{mr}}{den_k^{t+1} \times vol_k^{t+1}} \quad (1.20)$$

Where acc_k^{t+1}, den_k^{t+1}, vol_k^{t+1} are the acceleration, density and volume of individual k and, $den_{mr}$are the acceleration, density and volume of random matter.

Step (4): (2) Exploitation phase (no collision between individuals)

If *TF>0.5*, there is no collision between individuals and update acceleration for iteration t+1

$$acc_k^{t+1} = \frac{den_{best} + vol_{best} \times acc_{best}}{den_k^{t+1} \times vol_k^{t+1}} \quad (1.21)$$

Since acc_{best} is the acceleration of the best individual.

Step (4):(3) Print the acceleration

Print the acceleration to calculate the percentage change

$$acc_{k-norm}^{t+1} = u \times \frac{acc_k^{t+1} - min(acc)}{max(acc) - min(acc)} + l \quad (1.22)$$

*S*ince u and l are constants and $\mathrm{acc}_{\mathrm{k-norm}}^{\mathrm{t+1}}$ determines the percentage of steps, if the individual k is far from the optimal level, the acceleration value will be high, which means that the individual will be in the exploration phase, otherwise in the exploitation phase.

Step (5): Update the location

If TF≤0.5, i.e. exploration phase, the location of the k_th object for the next iteration t+1

$$x_k^{t+1} = x_k^t + c_1 \times rand \times acc_{k-norm}^{t+1} \times d \times (x_{rand} - x_k^t) \quad (1.23)$$

c_1 is constant else if $TF > 0.5$ stage of exploitation so updating the site is$x_k^{t+1} = x_{best}^t + F \times c_2 \times rand \times acc_{k-norm}^{t+1} \times d \times (A \times x_{best} - x_k^t)$ (1.24)

Since C_2 is a constant and A increases with time and is directly proportional to TF and is known as $A = c_3 \times TF.A$ (1.25)

It increases with time within the range [C_3×0.3,1] and by initially taking a certain percentage of the best position, as it starts with a low rate. There will be a large difference between the current location and the best location, and thus, the size of the random walk step will be high as the search continues. This percentage increases gradually to reduce the difference between the current and best locations, leading to the appropriate balance between exploration and exploitation.

F to change the direction of movement using

$$F = \begin{Bmatrix} +1 & if\ p \leq 0.5 \\ -1 & if\ p > 0.5 \end{Bmatrix} \quad (1.26)$$

Since $p = 2 \times rand - c_4$

Step (6): Evaluation

Each individual is evaluated using the function f, and the best solution found so far is found and $x_{best}, acc_{best}, den_{best}, vol_{best}$. [5][10].

Proposed algorithm (AOA-EPC)

This algorithm was proposed based on the AOA and EPC algorithms, where the first was hybridised based on the second. This algorithm's basis is that the AOA algorithm's local solutions are improved based on the EPC algorithm. Therefore, the results will be excellent, the local loophole can be overcome, and the optimal global values will be achieved. The work of this algorithm is integrated and efficient, so the two algorithms work together to achieve the desired goal. This algorithm mathematically simulates the behaviour of the AOA and EPC algorithms. The properties of the AOA algorithm, inspired by nature EPC, a random swarm algorithm, were utilised within a specific framework. An algorithm with excellent specifications was produced to avoid falling into local problems. It also proved its ability to reach the global optimal solution. It was applied to a group of countries for testing examples, and excellent results were obtained. They were explained in the tables of the practical side of the research and the following (Alweshah, 2023).

General steps of this algorithm (AOA-EPC)

1. Generating the initial random community.
2. Calculating the fitness function for each value in this community.
3. Improving the values of the solutions using the AOA algorithm, where the solution (element) is improved, and a fitness function is calculated for it.
4. Improving the resulting values using the EPC algorithm
5. Calculating the final fitness and arranging the solutions
6. Choosing the best solution
7. Comparing the best solution with the new solution and choosing the best between them.
8. If the number of repetitions exceeds the allowed limit, perform steps 3-7; otherwise, stop.

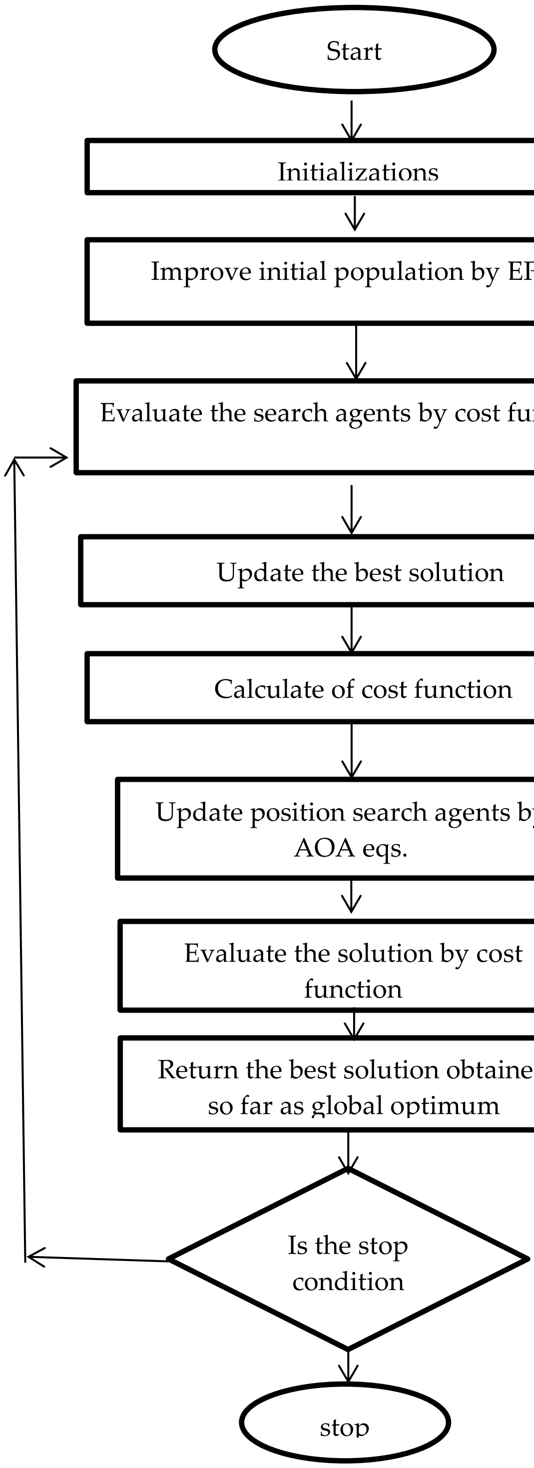

Figure 1. The flow chart Proposed AOA-EPC algorithm

The aim of hybridization

The goal of hybridization between the AOA and EPC algorithms is to obtain a high-performance algorithm. When using the AOA algorithm, which has notable mathematical properties that depend on the collision between its elements, and a mathematical model with high specifications, then combining it with a random intelligent swarm algorithm inspired by nature with another mathematical model, we aspire to obtain an algorithm that carries two types of properties and, therefore, its efficiency is very high, and this is what was achieved.

Result

Table 1. Comparison of the proposed algorithm AOA-EPC with the algorithms AOA and EPC at the number of elements N=30 and the number of iterations T=500

Function	EPC	Archives	EPC-AOA
F1	3.63E-100	6.96E-156	0
F2	5.75E-50	8.66E-88	0
F3	1.84E-99	4.64E-108	0
F4	1.00E-50	2.74E-79	0
F5	3.29E-198	2.17E-295	0
F6	-2.49E-01	-3.52E+02	-1.26E-03
F7	0.00E+00	1.5810E-322	0
F8	4.44E-15	-7.01E+01	3.24E-25
F9	0.00E+00	0.00E+00	0
F10	-1.87E-01	8.88E-16	8.88E-16

Discussions

To evaluate the work of the proposed hybrid algorithm, it was necessary to show the practical results by applying this algorithm to the measurement functions for optimization (Mirjalili, 2016). When using the programs, the results shown in Table (1) below appeared, and after comparing them with the original algorithms, they showed a high superiority of the hybrid algorithm.

Conclusion

When comparing the results of each of the AOA algorithms and the EPC algorithms separately and comparing them with the proposed algorithm AOA-EPC, we find a The numerical results show the big difference and superiority of the improved algorithm over the others. It was also concluded that the strength of the proposed algorithm lies in several elements:

First: It reaches the optimal global solution with the least number of elements for the swarm.

Second: It requires fewer iterations.

Third: The time taken to reach the optimal solution is much less.

References

Al-taie, Z. S. I. M., & Mitras, B. A. (2022). "Hybrid computational optimization for recogniza under foot," University of Mosul

Alweshah, M. (2023). Intrusion detection for the internet of things (IoT) based on the emperor penguin colony optimization algorithm. *Journal of Ambient Intelligence and Humanized Computing*, *14*(5), 6349–6366. https://doi.org/10.1007/s12652-022-04407-6

Cuevas, E., Fausto, F. & González, A. (2020). *New advancements in swarm algorithms: operators and applications*. Springer.

Gilbert,C., Robertson, G., Le Maho, Y. & Ancel, A. (2008).“How do weather conditions affect the huddling behaviour of emperor penguins?,” *Polar Biol.*, vol. 31, pp. 163–169, 2008.

Hammel, H. T. (1956). “Infrared emissivities of some arctic fauna,” *J. Mammal.*, vol. 37, no. 3, pp. 375–378.

Harifi, S., Khalilian, M., Mohammadzadeh, J. & Ebrahimnejad, S. (2019). “Emperor Penguins Colony: a new metaheuristic algorithm for optimization,” *Evol. Intell.*, vol. 12, pp. 211–226.

Harifi, S. (2020). Optimizing a Neuro-Fuzzy System Based on Nature-Inspired Emperor Penguins Colony Optimization Algorithm. *IEEE Transactions on Fuzzy Systems*, *28*(6), 1110–1124. https://doi.org/10.1109/TFUZZ.2020.2984201

Harifi, S. (2021). Optimization in solving inventory control problem using nature inspired Emperor Penguins Colony algorithm. *Journal of Intelligent Manufacturing*, *32*(5), 1361–1375. https://doi.org/10.1007/s10845-020-01616-8

Harifi, S. (2023). Swarm based automatic clustering using nature inspired Emperor Penguins Colony algorithm. *Evolving Systems*, *14*(6), 1083–1099. https://doi.org/10.1007/s12530-023-09507-y

Hashim, F. A. K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, “Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems,” *Appl. Intell.*, vol. 51, pp. 1531–1551, 2021.

Khalaf, A. H. & Mitras, B. A. (2020). “Using Modified Conjugate Gradient Method to Improve SCA,” in *Journal of Physics: Conference Series*, IOP Publishing, 2020, p. 12050.

Khalaf, A. H. & Mitras, B. A. (2021). “Two-hybrid sine cosine algorithm based on Invasive weed optimization algorithm and Bat algorithm,” in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, 2021, p. 12053.

Khalaf, A. H. & Mitras, B. A. (2021). “Use Two Hybrid SCA-AOA and Fuzzy SCA-AOA Algorithms in Information Security,” *Open Access Libr. J.*, vol. 8, no. 4, pp. 1–14, 2021.

Le Maho, Y., Delclitte, P. & Chatonnet, J. (1976). “Thermoregulation in fasting emperor penguins under natural conditions,” *Am. J. Physiol. Content*, vol. 231, no. 3, pp. 913–922.

Malik, H., Iqbal, A., Joshi, P., Agrawal, S. & Bakhsh, F. I. (2021). *Metaheuristic and evolutionary computation: algorithms and applications*, vol. 916. Springer..

Mirjalili, S. (2016). “SCA: a sine cosine algorithm for solving optimization problems,” *Knowledge-based Syst.*, vol. 96, pp. 120–133.

Salah, M. W. H. (2022). “Hybrid Meta-Heuristics Algorithms Using Unconstrained Optimization for Text Processing",” University of Mosul.

Taler, J. & Duda, P. (2006). *Solving direct and inverse heat conduction problems*. Springer.

Wahdan, H. G. (2020). Two-modified emperor penguins colony optimization algorithms. *Revue d'Intelligence Artificielle*, *34*(2), 151–160. https://doi.org/10.18280/ria.340205

Zhang, H. (2006). *Gradient methods for large-scale nonlinear optimization*. University of Florida.