Samarra Journal of Pure and Applied Science

www.sjpas.com

p ISSN: 2663-7405
e ISSN: 2789-6838

Cutting_Edge Hybrid Algorithms: Pelican Optimization Leveraging Conjugate Gradient and Osprey Techniques

Azhar Zaki Abdullah *, Ban Ahmed Mitras
Department of Mathematics, College of Computers Sciences and Mathematics, University of Mosul, Mosul, Iraq

https://doi.org/10.54153/sjpas.2025.v7i3.1122

Article Information
Received: 15/01/2025
Revised:16/02/2025
Accepted: 20/02/2025
Published: 30/09/2025
Keywords:

Corresponding Author
E-mail:
azhar.21csp26@student.uomosul.edu.iq
Mobile: 0773 849 8582

Abstract

This study investigates the performance improvement method of Pelican Optimization Algorithm (POA) through two different methods, namely Conjugate Gradient (CG) and Osprey Optimization Algorithm (OOA). The (POA) algorithm is a super-intuitive algorithm that has the ability to solve complex optimization problems. It has a distinctive exploration ability, but it has some problems in finding the solution accurately and quickly through the exploitation method. Therefore, (POA) was improved by exploiting the (CG) algorithm, which is included as an initial community for the (POA) algorithm in the first hybridization, which caused a simple improvement coupled with the original (POA) results. (POA) was hybridized by using the exploitation method of the (OOA) algorithm, which has proven its strength in finding the optimal solution accurately and quickly through the results, which were compared based on important measures such as the convergence rate, solution quality, and convergence speed shown through the graph, and the performance was evaluated through some global test functions. This new approach can be used in many scientific applications and fields such as economics, engineering, medicine, and biological sciences.

Introduction

Recent years have been an important stage in the study of algorithmic optimization due to the increasing complexity of real-world methods in many fields such as artificial intelligence, economics, and medicine. This is due to the ability of algorithms to explore large search areas in addition to their speed and effective ability to optimize these complex problems. Computational algorithms are considered one of the most important methods used in the real world for optimization in many fields such as energy and production, especially complex nonlinear problems that are difficult to solve using traditional methods. This has led to the search for new and advanced algorithms [1]. Urban expansion and global population growth have led to an increase in the demand for energy and the need for energy-efficient heat exchangers, especially heat exchangers, which have been achieved by integrating gray matter relationships (GRAs), genetic algorithms (GAs), and artificial neural networks (ANNs) to enhance energy efficiency [2]. Although there are many algorithms, heuristic algorithms are natural processes, such as firefly and cuckoo algorithms, swarm optimization, and others, which are widely used in optimization problems across different fields [3,4]. One of the nonlinear search methods is the conjugate gradient method, which is one of the most important methods used to solve unconstrained optimization problems. This method is effective with large-scale problems because it relies on first-order calculations and does not require second-order calculations [5]. The optimization conditions and basic properties of unconstrained optimization are among the most important studies that researchers seek to achieve in conjugate gradient algorithms, in addition to modified hybridization methods to improve numerical performance. With a focus on the focused analysis of how to update the search direction and methods of calculating the step length, which helps in performing the optimization [6,7]. The POA algorithm has been studied by a number of researchers and its exploratory ability has been highlighted, which is derived from its hunting style such as flying and diving in addition to its large-area search style [8]. This algorithm is suitable for solving optimization problems due to its ability to solve complex obstacles according to [9]. in order to reach an optimal solution, which usually needs to be optimized during the exploitation phase. However, by mimicking the same hunting methods of the osprey, which include diving and hovering near the prey, the osprey algorithm focuses more on rapid exploitation [10]. Using this approach, the algorithm may quickly optimize solutions at optimal locations; however, its tendency to over-exploit local optimization may impair performance on more complex and multimodal functions [11].One of the methods for evaluating the performance of (CG) for training (forward-feeding artificial neural networks), this method helped improve the parameters of the (CG) algorithm by making modifications to the parameters used in finding the search direction, which helped improve the convergence speed in addition to reducing the computation time [12]. Another method for optimization is the hybrid method, which is the process of merging two algorithms either through initial communities or by merging equations through exploration and exploitation methods. Such as developing a hybrid algorithm that relies on classical quantum computing with Penders partitioning to solve the problem of power systems and mixed programming problems [13]. As well as the group search algorithm and merging it with the particle swarm optimization algorithm, which has proven its effectiveness and ability to solve complex engineering and numerical problems [14]. As well as the group search algorithm and merging it with the particle swarm optimization algorithm, which has proven its effectiveness and ability to solve complex engineering and numerical problems [15]. One of the hybrid algorithms is the algorithm that combines the deep learning network and the Seagull Adapted Elephant Herding algorithm, which was used to detect security attacks supported by the Internet in cyber systems, and which has shown its effectiveness in improving and accuracy in detection [16]. In this paper, two different approaches are proposed to improve and hybridize the Pelican algorithm (POA) in two different ways. The first is by proposing the conjugate gradient method which will be used to improve the (POA) algorithm through initial communities. The second is by using the Osprey optimization algorithm (OOA) which will be integrated with the Pelican algorithm through the exploitation method.

Conjugate Gradient Algorithm (CG)

The CG algorithm is a mathematical method used in indirect optimization to find the minimum or maximum of a function. It is usually used in complex nonlinear equation systems to solve. This algorithm is also effective for solving positively definite symmetric linear systems, which are difficult to find by numerical or traditional methods. It relies on the conjugate search direction instead of the classical gradient direction. The algorithm works by directing the search in orthogonal directions used to improve convergence to the optimal solution with greater accuracy, efficiency and speed [17]. Below we propose a new approach to improve the conjugate gradient algorithm by proposing and deriving a new parameter which will be used to improve the Pelican algorithm (POA). Optimization can be defined as the method by which we can reach the best solutions to a given problem. Which has lower and upper bounds for the problem that contains n variables, where $n \geq 0$. The global minimum is the lower factor of the function and is the global minimum within the domain of the function. While the local minimum is the lowest point within its domain. The main goal of globally convergent algorithms is to find the global minimum, while locally convergent algorithms aim to determine the local minimum [18].

The proposed improved conjugate gradient (CG) method

Salihu et al. in 2023 proposed a new proof for the conjugate spectral gradient method as follows [19]:

$d_{k+1} = -\theta g_{k+1} + \beta_k^{BMIL} d_k \;\;, \; k \geq 0$

Where is β_k^{BMIL} a parameter defined in [20] as: $\beta_k^{BMIL} = \frac{g_{k+1}^T \; y_k}{\|d_k\|^2}$

$\theta = \eta + \frac{\|y_k\|}{\|d_k\|} \;\;, \eta > 0$, Where θ is the spectral coefficient. It is the angle between the search direction d_k and the negative gradient $-g_{k+1}$, and $\theta \in (\frac{\pi}{2}, \pi)$. , η is a parameter that adjusts the search direction during iterations to ensure convergence without resorting to linear search rules [19].

New let $d_{k+1}^{CG} = d_{k+1}^{SKA}$, where SKA is the proposed search direction in [20].

$-g_{k+1} + \beta_k^{New} d_k = -\theta g_{k+1} + \beta_k^{BMIL} d_k$, We get and multiply both sides of the equation by y_k^T we get.

$-y_k^T g_{k+1} + \beta_k^{New} y_k^T d_k = -\theta y_k^T g_{k+1} + \beta_k^{BMIL} y_k^T d_k$

By doing some steps to simplify the relationship we get.

$\beta_k^{New} = (1-\theta)\frac{y_k^T \; g_{k+1}}{y_k^T \; d_k} + \frac{y_k^T \; g_{k+1}}{\|d_k\|^2} . \frac{y_k^T \; d_k}{y_k^T \; d_k}$, where $\beta_k^{HS} = \frac{y_k^T \; g_{k+1}}{y_k^T \; d_k}$.

$$\beta_k^{New} = (1-\theta)\beta_k^{HS} + \frac{y_k^T \; g_{k+1}}{\|d_k\|^2} \qquad (1)$$

Modified algorithm steps

Step 1. Choose an initial value, mode, $g_0 = \nabla f(x_0)$, $d_0 = -g_0$, $k \geq 0$

Step 2. Calculate the step length value $\lambda_k \geq 0$, which should achieve Wolfe condition.

Step 3. If $\|g_{k+1}\| < \epsilon$ it stops, otherwise update the variables.$x_{k+1} = x_k + \lambda_k d_k$

Step 4. Calculate the new trend through, where $d_{k+1} = -g_{k+1} + \beta_k^{New} d_k$, β_k^{New} is the new update coefficient.

Step 5. Increase the meter k by 1, then return to step 2.

Convergence Analysis of New Conjugate Vector Method

We will investigate the necessary condition for convergence, namely the sufficient descent property of our algorithm. Importance of this property determines efficient convergence properties that search direction always landscape with negative gradient at every iteration of the algorithm. Consequently, it helps in the gradient descent of objective function to reach an optimal solution [21].

Assumption 1. Function f is bounded by set $S = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$. and function with the gradient, where then there is a Lipschitz constant $L > 0$ such that $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$, $\forall x, y \in S$.

Theorem 1. The value of search direction d_k used in the proposed algorithm. for CG, attains the property of adequate gradients for all k, subject to Wolfe's conditions on step size d_k are satisfied.

$$g_{k+1}^T d_{k+1} \leq -\mu\|g_{k+1}\|^2 \;, \quad \mu > 0 \qquad (2)$$

By the method of mathematical induction, we will prove the above relationship.

If $k = 0$, $\quad d_0 = -g_0 \Longrightarrow g_0^T d_0 = -\|g_0\|^2 < 0$ the relationship is true.

We assume that the relation is true for all values $k \geq 0$, that is,

$$g_k^T d_k < 0 \;, \; g_k^T d_k \leq -c\|g_k\|^2 \quad , \; c > 0 \qquad (3)$$

New we prove that the relationship at $k+1$.

$$d_{k+1} = -g_{k+1} + \beta_k^{New} d_k \qquad (4)$$

Multiplying both sides of equation (4) by the magnitude g_{k+1}^Tgives us

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \left[(1-\ \eta)\frac{y_k^T\ g_{k+1}}{y_k^T\ d_k} + \frac{y_k^T\ g_{k+1}}{\|d_k\|^2}\right] g_{k+1}^T d_k$$

Using Cauchy-Schwarz inequality we get, $y_k^T\ g_{k+1} \le \|y_k\| \|\ g_{k+1}\| y_k^T\ d_k \le \|y_k\| \|d_k\|$. then

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + (1-\ \eta)\frac{\|y_k\|\|d_k\|\ \|g_{k+1}\|^2}{\|y_k\|\|d_k\|} + \frac{\|y_k\|\|d_k\|\|g_{k+1}\|^2}{\|d_k\|^2}$$

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \left(1-\ \eta - \frac{\|y_k\|}{\|d_k\|}\right)\|g_{k+1}\|^2 + \frac{\|y_k\|\ \|g_{k+1}\|^2}{\|d_k\|}$$

$$g_{k+1}^T d_{k+1} \le -\eta \|g_{k+1}\|^2$$

Now we take the special cases of η

case 1: if $\eta = 1,$ then $g_{k+1}^T d_{k+1} \le -\|g_{k+1}\|^2$

case 2: if $\eta = \frac{\|y_k\|}{\|d_k\|}$, then $g_{k+1}^T d_{k+1} \le -\left(\frac{\|y_k\|}{\|d_k\|}\right)\|g_{k+1}\|^2$

case 3: if $\eta = 0.5$, then $g_{k+1}^T d_{k+1} \le -0.5\|g_{k+1}\|^2$

We notice that all cases achieve the relationship. $g_{k+1}^T d_{k+1} \le -\mu \|g_{k+1}\|^2$, $\mu > 0$

Comprehensive investigation of the convergence of the proposed algorithm.

We will now show that the proposed conjugate gradient method achieves the comprehensive convergence property by proving the following theorem:

Lemma 1. It is suggested that Assumption (1) is satisfied, and that the CG technique is also satisfied, since $d_k$is the The search direction for the slope and α_k is generated by the strong Wolff condition (SWP).), then if $\sum_{K=1}^{\infty}\frac{1}{\|d_{k+1}\|^2} = \infty$ then, $\lim_{k\to\infty} inf\|g_k\| = 0.$

Theorem 3. Suppose Assumption (1) and the proposed conjugate gradient method satisfied in the search direction d_k, also step length α_k is generated from conditions (SWP), then $\lim_{k\to\infty} inf\|g_k\| = 0.$

Proof. Since the algorithm satisfies the theorem (1), and if $g_{k+1} \ne 0$, we will prove that it is constrained from above$\|d_{k+1}\|$, and by taking $\|.\|$ for both sides of equation (4) we get,

$$\|d_{k+1}\| = \left\|-g_{k+1} + \beta_k^{New} d_k\right\|$$

$$\|d_{k+1}\| \le \|g_{k+1}\| + \left|\beta_k^{New}\right|\|d_k\| \tag{5}$$

$$\|d_{k+1}\| \le \|g_{k+1}\| + \left|\left[\left(1-\ \eta - \frac{\|y_k\|}{\|d_k\|}\right)\frac{\|y_k\|\ \|g_{k+1}\|}{\|y_k\|\ \|d_k\|} + \frac{\|y_k\|\ \|g_{k+1}\|}{\|d_k\|^2}\right]\right|\|d_k\|$$

$$\|d_{k+1}\| \le \|g_{k+1}\| + |(1-\ \eta)|\|g_{k+1}\|$$

Case 1. When $\eta = 1$, we get $\|\mathrm{d}_{k+1}\| \le \|\mathrm{g}_{k+1}\|$

$\|d_{k+1}\| \le \gamma_1$, $\gamma_1 = \|g_{k+1}\|$

$$\frac{1}{\|d_{k+1}\|} \ge \frac{1}{\gamma_1} \Rightarrow \sum_{k=1}^{\infty}\frac{1}{\|d_{k+1}\|^2} = \sum_{k=1}^{\infty}\frac{1}{{\gamma_1}^2} = \frac{1}{{\gamma_1}^2}\sum_{k=1}^{\infty} 1 = +\infty$$

Case 2. When $\eta = \frac{\|y_k\|}{\|d_k\|}$, we get $\|d_{k+1}\| \le \|g_{k+1}\| + \left|(1-\frac{\|y_k\|}{\|d_k\|})\right|\|g_{k+1}\|$

$\|d_{k+1}\| \le \gamma_2$, $\gamma_2 = \|g_{k+1}\| + \left|(1-\frac{\|y_k\|}{\|d_k\|})\right|\|g_{k+1}\|$

$$\frac{1}{\|d_{k+1}\|} \geq \frac{1}{\gamma_2} \Rightarrow \sum_{k=1}^{\infty} \frac{1}{\|d_{k+1}\|^2} = \sum_{k=1}^{\infty} \frac{1}{\gamma_2{}^2} = \frac{1}{\gamma_2{}^2} \sum_{k=1}^{\infty} 1 = +\infty$$

Case 3. When $\eta = 0.5$ we get $\|d_{k+1}\| \leq 1.5\|g_{k+1}\|$

$$\|d_{k+1}\| \leq \gamma_3 \quad , \quad \gamma_3 = 1.5\|g_{k+1}\|$$

$$\frac{1}{\|d_{k+1}\|} \geq \frac{1}{\gamma_3} \Rightarrow \sum_{k=1}^{\infty} \frac{1}{\|d_{k+1}\|^2} = \sum_{k=1}^{\infty} \frac{1}{\gamma_3{}^2} = \frac{1}{\gamma_3{}^2} \sum_{k=1}^{\infty} 1 = +\infty$$

Note that all cases are true. Therefore, $\lim_{k\to\infty} inf\|g_k\| = 0$

Pelican Optimization Algorithm (POA)

Pelican is a water bird of the pelican family. It has a huge body and long beak made up of a bag which it uses to catch fish and other various foodstuffs. This is an animal which is gregarious in nature. Pelican mostly in groups of a few hundred. The pelican can weigh anything from 2.75 to 15 kg. Its length is between 1.06 and 1.85 meters, which it is considered a large bird. It has a wingspan as large as three meters. This size provides both stability for long distance flights and a lot of lifting force for take-off. Pelicans have sophisticated social behaviors. A flock of pelicans will work together to find food. When it discovers a school of fish, the pelican begins to dive to depths anywhere from 10-20 meters. Once it has dived, it starts slowly flapping its broad wings on the water's surface thus moving the fish over towards areas shallower in depth, making them easier to capture.

The Pelican Optimization Algorithm (POA)is a population-based algorithm that relies on simple heuristics and rules to find optimal solutions. Each pelican represents a candidate solution in the search space. The algorithm iteratively updates these positions, using strategies which mimic the natural patterns foraging birds carry out as they look over landscapes in search of food.

Inspirations and Mathematical Model (POA)

The (POA) is based on the foraging behavior of pelicans and attempts to solve optimization problems.

The algorithm starts by initializing the population where the algorithm is based on a random set of individuals; in fact, each individual represents a candidate. The location of each pelican in the search space is determined by a certain mathematical formula.

$$x_{i,j} = l_j + rand.(u_j - l_j), i = 1,2,\dots,N. \ , j = 1,2,\dots,m. \qquad (6)$$

Where $x_{i,j}$is the fee of the j^{th} variable indicated through the i^{th} candidate solution, N is the length of the population, m is the broad range of problem variables, rand is a random number comes from (0, 1), l_j is the lower bound of the j^{th} problem variables, and u_j is the upper bound of thej^{th} problem variables [8].

Then it is converted into a population matrix where there is one candidate solution in each column of the population matrix, and each row represents a variable. Thus, the locations of pelicans for food gathering can be tracked.

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,d} & \cdots & x_{n,m} \end{bmatrix} \qquad (7)$$

Then evaluate the objective function The performance of each candidate is evaluated through the objective function vector, which helps in comparing the success rate of each individual in the herd. And to know the best places for each individual.

$$F = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_n) \end{bmatrix} \quad (8)$$

Pelicans rely on two main stages in their synchronized hunting strategy:

1- Exploration phase: It represents the phase in which the pelican searches, locates the prey and moves towards it. This phase represents the only possible algorithmic option for exploring areas by searching well in different areas for alternative solutions. The goal of the exploration phase is to avoid choosing in non-optimal alternative solutions by searching widely in solutions. Which can be represented mathematically as follows:

$$x_{i,j}^{p.1} = \begin{cases} x_{i,j} + rand.\left(p_{.1} - I.x_{i,j}\right) \ , & F_p < F_i \\ x_{i,j} + rand.\left(x_{i,j} - p_{.1}\right) \ , & otherwise \end{cases} \quad (9)$$

where $x_{i,j}^{p.1}$ represents the new state of the i^{th} swan, which is located in dimension j according to the first stage, and F_p is the value of the objective function. I is a random number that takes the value 1 or 2. Which is taken randomly at each iteration and for each element. When this value is 2, it gives more space to the element, which moves this member to another, more recent level of the search space. The parameter I affects the agent's ability to scan the search space correctly. Accordingly, according to the proposed work plan, when the objective function value for that position is improved, it becomes the new position of the swan. While the algorithm stops when moving to non-optimal positions during this type of update, which is known as active update.

$$X_i = \begin{cases} X_i^{p.1}, F_i^{p.1} & F_i^{p.1} < F_i\,, \\ X_i & otherwise\ , \end{cases} \quad (10)$$

where $X_i^{p.1}$ is the i^{th} pelican's new status and $F_i^{p.1}$ is its objective function value depending on phase.

Exploitation phase: In this strategy, the pelican uses its wings to glide along the water above which it lies, providing propulsion in shallow water streams where it can approach its prey. The search space shrinks with this movement. This movement is described in the mathematical model by gradually reducing the search space to the largest solution regions through the main equations: where the false agents (swans) are identified at the locations of the search space using this equation:

$$x_{i,j}^{p.2} = x_{i,j} + R.\left(1 - \frac{t}{T}\right).(2rand - 1).x_{i,j} \quad (11)$$

Where $x_{i,j}^{p.2}$ is the new location of the i^{th} swan in dimension *j*. The variables R and T can control how far the algorithm searches around the location of each individual swan. R is a constant set to 0.2. It helps constrain how far the algorithm searches for each swan. t is the current iteration of the algorithm. T is the total number of iterations of the algorithm.

At first, the search area is large. That is, the algorithm searches a large range of possible solutions. As the algorithm continues to run, the search area gets smaller. This allows the algorithm to better focus on the solutions that have been found. the algorithm is then updated to see if it should have found better locations. This is done using the equation.

$$X_i = \begin{cases} X_i^{p.2}, F_i^{p.2} & F_i^{p.2} < F_i\,, \\ X_i & otherwise\ , \end{cases} \quad (12)$$

where $X_i^{p.2}$ is the new position of the i^{th} pelican, and $F_i^{p.2}$ is the value of how good this position is. If the new position is better than the old position, the algorithm keeps it; otherwise, it stays in the old position.

Osprey optimization algorithm (OOA)

The Osprey is a bird of prey also known as the "fish hawk", and its diet mainly consists of fish. It is characterized by specifications as it is between 50 and 66 cm long, and weighs between 900 and 2100 grams, and the Osprey is characterized by a large wingspan ranging between 125 and 180 cm. The Osprey has a shiny brown body, a white chest, and white feet with strong black claws that enable it to catch prey. The most important feature of the Osprey is its sharp vision that enables it to detect fish under the surface of the water while flying at medium altitudes ranging between 10 and 40 meters. After determining the location of the prey, it swoops down on the fish and dives to catch it, then carries it to a safe place to eat it [22,23].

Inspirations and Mathematical Model (OOA)

The Osprey Optimization Algorithm (OOA) is a new heuristic meta-algorithm inspired by the hunting behavior of the osprey. It is a developmental study aimed at solving engineering optimization problems by balancing the exploration of the solution space with the exploitation of promising regions to achieve the best results [22].

The algorithm is initially initialized in the same approach as the Pelican Algorithm (POA) where each fish osprey is considered as a candidate solution and then transformed into a vector, thus forming an OOA set of elements where the location of each element is randomly initialized in the search space and transformed into a matrix of osprey locations, and then the values of the objective function of the problem are represented as shown in Equations (6,7,8). The objective function is a key measure to evaluate the accuracy of the candidate solutions which represents the best value of the eagle location which is updated in each iteration as the location and best candidate solution.

Exploration stage: Ospreys are excellent fish hunters because of their ability to detect the location of fish underwater due to their strong eyesight. Prey is hunted after determining its location by descending below the water surface. This method of updating the initial population of OOA is represented by modeling the bird's attack strategy on the prey, which depends largely on the search space and determining the best area and avoiding falling into possible local solutions. The location of each osprey relative to the locations of the rest of the individuals in the search space that has an objective function represents the best prey in the water. Determining the prey group relative to the predator is represented by the equation

$$FP_i = \{X_k | k \in \{1,2,\dots,N\} \wedge F_k < F_i\} \cup \{X_{best}\} \quad (13)$$

Where FP_i represents the set of fish that the i^{th} osprey can see. X_k is the total number of fish, X best is the best solution obtained, F_k is the performance value of the solution, and F_i is the performance value of the specific solution i^{th} that we are now analyzing.

After randomly detecting the position of one of the fish, the osprey attacks the prey, and according to the movement of the bird towards the prey, this movement is simulated through the following relations in which the new position of the osprey is calculated. The new position replaces the previous position of the bird according to,

$$xp_{i,j}^1 = x_{i,j} + r_{i,j} \cdot (SF_{i,j} - I_{i,j} \cdot x_{i,j}) \quad (14)$$

$$xp_{i,j}^1 = \begin{cases} XP_i^1, & lb_j < xp_{i,j}^1 < ub_j, \\ lb_j, & xp_{i,j}^1 < lb_j, \\ ub_j, & xp_{i,j}^1 > ub_j. \end{cases} \quad (15)$$

$$X_i = \begin{cases} X_i^{P_1}, & FP_i^1 < F_i, \\ X_i & otherwise, \end{cases} \quad (16)$$

Where XP_i^1 is the new i^{th} osprey position based on the previous stage of OOA, $xp_{i,j}^1$ is its j^{th} dimension, FP_i^1 is the fitness function value, SF_i is the location of the prey chosen by the predator, $SF_{i,j}$ is its j^{th} dimension, $r_{i,j} \in [0,1]$ are random numbers, and $I_{i,j}$ are random numbers from the set {1, 2}.

Exploitation Stage: This stage depends on the process of moving the fish after catching it to a suitable location for eating. The natural behavior of the bird is simulated by updating the OOA community. This strategy of moving the prey to a safe position leads to some changes in the search space, and thus an increase in the exploitation power of OOA in the local search areas and an increase in the convergence to a better solution than the explored solutions, as each position for eating the fish is calculated randomly as a safe position for eating as shown in the equation (17,18), and then the new position is improved through the objective function that replaces the previous position of the osprey as shown in equation (19).

$$xp_{i,j}^{*} = x_{i,j} + \frac{lb_j + r.(ub_j - lb_j}{t}, \; i = 1,2,\dots,N. \;, j = 1,2,\dots,m. \;, t = 1,2,\dots,T \qquad (17)$$

$$xp_{i,j}^{*} = \begin{cases} XP_i^{*}, & lb_j < xp_{i,j}^{*} < ub_j , \\ lb_j , & xp_{i,j}^{*} < lb_j , \\ ub_j , & xp_{i,j}^{*} > ub_j . \end{cases} \qquad (18)$$

$$X_i = \begin{cases} XP_i^{*}, & FP_i^{*} < F_i , \\ X_i & else, \end{cases} \qquad (19)$$

where XP_i^{*} is the new position of osprey i^{th} based on the current stage of OOA, $xp_{i,j}^{*}$ is its j^{th} dimension, FP_i^{*} is the fitness function value, $r_{i,j} \in [0,1]$ are random numbers, t is the algorithm's iteration counter, and T is the total number of iterations.

Hybrid The Pelican Optimization Algorithm (POA)

It is known that hybrid algorithms are the latest and new methods and approaches in artificial intelligence and computing fields. Where the original source algorithms can be utilized and improved to solve problems and achieve many common goals to solve many complex problems through systematic interconnection and cooperative relationships between the attributes and style of each algorithm.

We will now introduce the proposed algorithms (POA-CG) and (POA-AOO).

Hybrid Swarm Optimization (POA) Using the Developed of CG Algorithm.

In this section, we show how the POA algorithm can be improved using CG, a mathematical optimization technique that aims to reach an optimal solution within unconstrained optimization problems. As for the hybrid approach, the purpose is to use CG to generate and modify the initial solutions that will serve as the initial solutions included in the POA algorithm for the search field to be used to improve the solutions in the search field. This will also help in reducing the search time and determining the direction of the algorithm in search and exploitation which are the strengths of the hybrid. However, one does not prevent facing some challenges such as the complexity in computation and integration of the two techniques used and thus tuning the parameters can lead to poor results or results that are almost identical to the original. Therefore, it almost always needs a strong and high-precision management to address any problem related to hybridization between fitness algorithms and conjugate gradient algorithms, as this hybridization improves the performance of POA by improving the initial solutions and exploring more space but more management is needed to address the problems of complexity and effective combinatorial integration.

Flowchart Steps for Hybrid POA-CG Algorithm

1 -Start: In this step, parameters which include population size and maximum number of iterations, are defined for the algorithm setup.

2 -Population Initialization: As the second step, the positions of the pelican population are simulated within the boundaries of the search space.

3 -Objective function evaluation: In this function every pelican is evaluated and allocated an objective function.

4 -Identify best solution: Out of the group of current population, the best solution is determined.

5 -Exploration Phase (POA): It involves Exploring other new areas by pushing the pelicans closer to possible locations of the prey to allow the birds' exploration.

6 -Exploitation Phase (CG): After this, the best positions identified are surrounded by the Conjugate Gradient method to provide accurate local optimization.
7 -Add New Best Solution: Determine whether the better solution has been located and adjust the best solution which has been found.
8 -Convergence Check: Under these conditions, the process moves to the output stage; if not, it returns to the exploration stage.
9- Output: The final stage in which solution obtained through hybrid algorithm is displayed.

Hybridization of Pelican Optimization Algorithm (POA) by Osprey Optimization Algorithm (OOA)

The combination of the Pelican Optimization Algorithm (POA) and the Eagle Optimization Algorithm (OOA) is achieved through the heuristic search directions of the Pelican Optimization Algorithm (POA) and the exploitation directions of the Osprey Optimization Algorithm (OOA). In this way, the main goal is to optimize the search strategy so that the global search and local search for the best solutions are provided in a balanced manner. The algorithm adheres to a number of equations that define the movement of agents within a specific region of the search space and are modified to optimize the solutions in a window of multiple iterations.

$$X_{new} = X_{i,j} + rand(1,1).\left(X_{food} - I.X_{i,j}\right) \quad (20)$$

If the fitness function is better, it is accepted

$$\text{If } f_{new} < f(i) \Longrightarrow X_{i,j} = X_{new} \quad and \quad f(i) = f_{new} \quad (21)$$

Then we move to the exploitation phase, where this phase improves the solutions by taking advantage of the best performing solutions or specific fish positions to exploit the local areas

$$X_{new-P1} = X_{i,j} + rand(1,1).\left(X_{selected\,fish} - I.X_{i,j}\right) \quad (22)$$

If the fitness of this new position improves, the solution is updated

$$\text{If } f_{newP1} < f(i) \Longrightarrow X_{i,j} = X_{newP1} \quad and \quad f(i) = f_{newP1} \quad (23)$$

The algorithm in question is based on a probabilistic model that permits repositioning of agents depending on the fitness level of the surrounding agents. Due to the aforementioned features, the algorithm uses an effective approach wherein it attempts to explore potential zones while also seeking the best global solution in case no neighboring agents feature higher fitness levels as indicated in Equation (20). During the exploitation phase described in Equation (22,23), the size of the movements made by the agents is decreased incrementally so that the agents' convergence towards the optimal solution is performed steadily and precisely. The positions of the agents are modified only when the improvement occurs so that the improvement is guaranteed after every cycle of the iterations.

The gradual shrinkage phase comes, where after every 10 iterations, the search space shrinks to push the solutions towards zero, which gradually improves the accuracy

$$X_{i,j} = 0.9\,.X_{i,j} \quad (24)$$

Exploration in (POA) relies on a broad search of the solution space to discover the best possible results, but it can sometimes lead to local optimization. In contrast, exploitation in (OOA) focuses on improving the discovered solutions by reducing the search scope, which speeds up the process of finding the optimal solution. When (POA) is hybridized with (OOA), an effective balance is achieved between global search and local refinement, which enhances the algorithm's ability to find accurate solutions faster than using (POA) alone.

Through the results obtained using MATLAB R2021a, it was shown that the hybrid algorithm (POA-OOA) clearly outperformed its counterparts from the three algorithms (POA-CG1), (POA-CG2), (POA-CG3), in addition to the original algorithm (POA) through the illustrations shown in Figures (2). The comparison showed that (POA-OOA) reached the optimal solution by 100%, and this was verified on five global and exploration single-mode test functions, as shown in the tables 1.

Table 1: Standard test functions are used to evaluate the efficiency of computational algorithms.

Objective Function	Dimension	Range	F_{min}
$F_1 = \sum_{i=1}^{n} x_i^2$	30	$[-100,100]$	0
$F_2 = \sum_{i=1}^{n} \|x_i\| + \prod_{i=1}^{n} \|x_i\|$	30	$[-10,10]$	0
$F_3 = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2$	30	$[-100,100]$	0
$F_4 = max_i\{\|x_i\|. 1 \le i \le n\}$	30	$[-100,100]$	0
$F_5 = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5 \cdot 12, 5 \cdot 12]$	0

Table 2: Comparison outcomes of (POA), (POA_CG1), (POA_CG2), (POA_CG3), and (POA-OOA) With number of elements with 30 elements and 1000 iterations.

Function Symbol	POA	POA-CG1	POA-CG2	POA-CG3	POA-OOA
F_1	9.519e-214	3.3291e-276	1.1077e-248	1.3511e-241	0
F_2	3.8025e-107	4.8304e-134	3.0162e-126	2.9611e-127	0
F_3	1.0699e-212	3.0603e-262	6.2607e-237	4.6242e-245	0
F_4	1.3981e-108	1.5868e-133	2.387e-120	2.8769e-118	0
F_5	28.7066	28.3332	28.3715	27.2486	0

The results shown in Table 2 the table were obtained by repeating each algorithm 10 times and then calculating the final average for each algorithm in order to improve the reliability of the results, and to reduce the influence of randomness and individual abnormal and unnatural cases. This is shown in the figures 1-5.

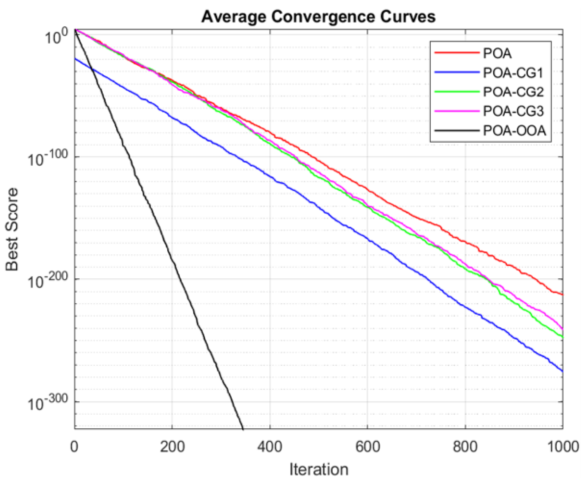

Fig 1. Function graph for F_1

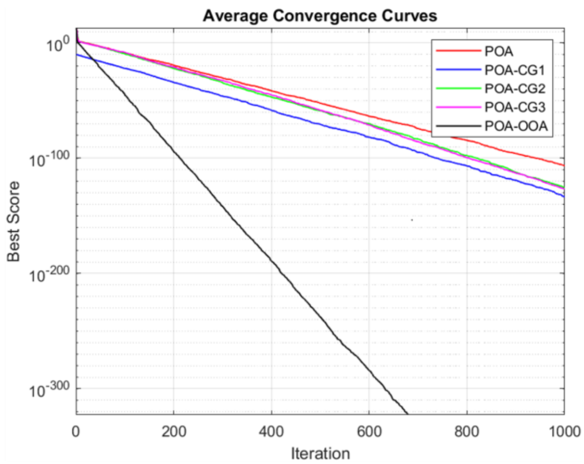

Fig 2. Function graph for F_2

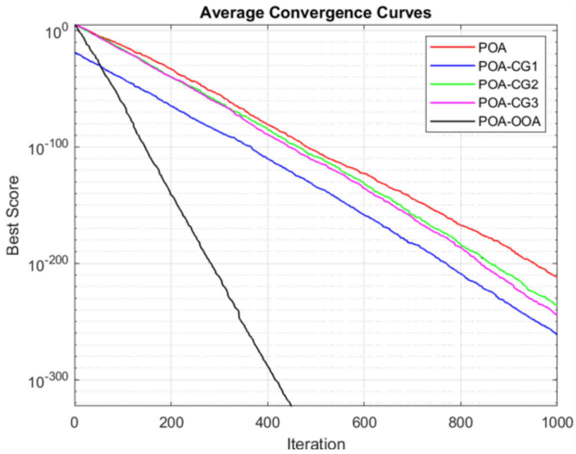

Fig 3. Function graph for F_3

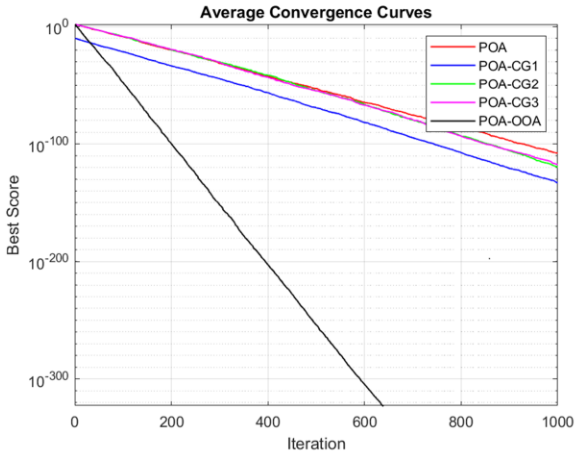

Fig 4. Function graph for F_4

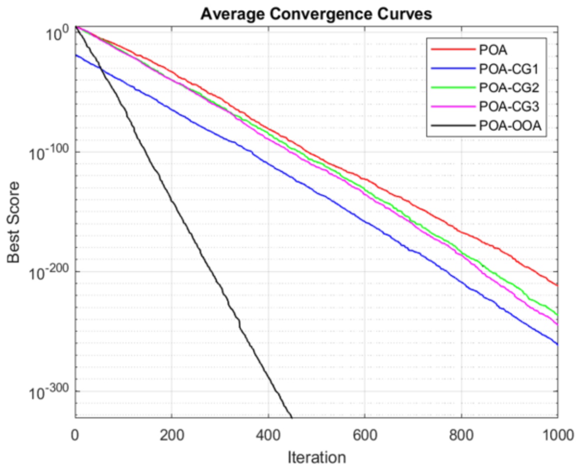

Fig 5. Function graph for F_5

The original POA algorithm shown in the graphs shows relatively slow convergence performance starting from a high starting value that slowly decreases making it less efficient than the hybrid algorithms. The faster convergence rate of POA-CG1 allows for higher quality solutions to be obtained more efficiently. Instead, POA-CG2 performs well in steady convergence which is somewhat slower than the rest of the algorithms but more consistent and stable. Conversely, POA-CG3 also resembles CG1 with good relative convergence speed and sometimes reaches the minimum faster than CG2. The hybrid algorithm POA-OOA dominates, as it is the fastest in converging to the optimal solution. This means that it is best used for solving problems that depend on accurate and fast minima. Therefore, POA-OOA is the most suitable while the rest include POA-CG1 and POA-CG3 which are very robust, however CG2 is better in terms of relative stability. The POA-CG optimization improves the convergence speed in the initial stages of the search by entering the initial community of the POA algorithm, which helps to efficiently guide the search towards the optimal solutions and reduces the number of iterations. While the hybrid POA-OOA improves the accuracy of the solutions by balancing exploration and exploitation, outperforming traditional methods such as PSO and GA in solving complex problems, combining wide coverage of the search space (POA) and focused search (OOA).

Conclusions

In this study, a new optimization algorithm based on a hybrid process of a new algorithm, the (POA-OOA) algorithm, was presented. The main inspiration for the proposed algorithm is to integrate the strategy and behavior of pelicans during hunting, these behaviors are diving towards their prey and moving their wings on the water surface, The (OOA) is inspired by the way ospreys catch fish. These different steps of the two algorithms were described through the exploration and exploitation stages as a mathematical model for use in the field of optimization for pelicans. In addition to the optimization method by integrating the proposed new CG algorithm with the (POA) algorithm, these algorithms were implemented through the MATLAB R2021a program. The results we obtained in optimizing the results of single-mode functions clearly showed the superiority of the hybrid algorithm (POA-OOA) over both the original algorithm (POA) by reaching the optimal solution. It also outperformed the proposed algorithms (POA-CG1), (POA-CG2), (POA-CG3) and slightly outperformed the original algorithm. These results demonstrated the ability to address design optimization problems in many fields in faster, more accurate and less costly ways.

References

1- Yahia, H. S., Zeebaree, S. R., Sadeeq, M. A., Salim, N. O., Kak, S. F., AL-Zebari, A., ... & Hussein, H. A. (2021). Comprehensive survey for cloud computing based nature-inspired algorithms optimization scheduling. Asian Journal of Research in Computer Science, 8(2), 1-16.
2- Venkatesh, B., Khan, M., Alabduallah, B., Kiran, A., Babu, J. C., Bhargavi, B., & Alhayan, F. (2023). Design Optimization of Counter-Flow Double-Pipe Heat Exchanger Using Hybrid Optimization Algorithm. *Processes*, 11(6), 1674.
3- Yang, X. S. (2020). *Nature-inspired optimization algorithms*. Academic Press.
4- Yang, X. S. (2020). Nature-inspired optimization algorithms: Challenges and open problems. *Journal of Computational Science*, 46, 101104.
5- Sato, H. (2022). Riemannian conjugate gradient methods: General framework and specific algorithms with convergence analyses. *SIAM Journal on Optimization*, *32*(4), 2690-2717.
6- Andrei, N. (2020). *Nonlinear conjugate gradient methods for unconstrained optimization*. Berlin, Heidelberg: Springer International Publishing.
7- Abdollahi, F., & Fatemi, M. (2021). An efficient conjugate gradient method with strong convergence properties for non-smooth optimization. *Journal of Mathematical Modeling*, *9*(3), 375-390.
8- Trojovský, P., & Dehghani, M. (2022). Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. *Sensors*, *22*(3), 855.
9- Alamir, N., Kamel, S., Megahed, T. F., Hori, M., & Abdelkader, S. M. (2023). Developing hybrid demand response technique for energy management in microgrid based on pelican optimization algorithm. *Electric Power Systems Research*, *214*, 108905.
10- Dehghani, M., & Trojovský, P. (2023). Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems. *Frontiers in Mechanical Engineering*, *8*, 1126450.
11- Zhou, L., Liu, X., Tian, R., Wang, W., & Jin, G. (2024). A Modified Osprey Optimization Algorithm for Solving Global Optimization and Engineering Optimization Design Problems. *Symmetry*, *16*(9), 1173.
12- Kamilu, K., Sulaiman, M. I., Muhammad, A. L., Mohamad, A. W., & Mamat, M. (2023). Performance evaluation of a novel conjugate gradient method for training feed forward neural network. *Math. Model. Comp*, *10*, 326-337.

13- Chang, C. Y., Jones, E., Yao, Y., Graf, P., & Jain, R. (2020). On hybrid quantum and classical computing algorithms for mixed-integer programming. *arXiv preprint arXiv:2010.07852*.
14- Braik, M., Al-Zoubi, H., Ryalat, M., Sheta, A., & Alzubi, O. (2023). Memory based hybrid crow search algorithm for solving numerical and constrained global optimization problems. *Artificial Intelligence Review*, *56*(1), 27-99.
15- Azevedo, B. F., Rocha, A. M. A., & Pereira, A. I. (2024). Hybrid approaches to optimization and machine learning methods: a systematic literature review. *Machine Learning*, 1-43.
16- Sagu, A., Gill, N. S., Gulia, P., Priyadarshini, I., & Chatterjee, J. M. (2024). Hybrid Optimization Algorithm for Detection of Security Attacks in IoT-Enabled Cyber-Physical Systems. *IEEE Transactions on Big Data*.
17- Yuan, G., Li, T., & Hu, W. (2020). A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems. *Applied numerical mathematics*, *147*, 129-141.
18- Chong, E. (2013). An Introduction to Optimization.
19- Salihu, N., Kumam, P., Awwal, A. M., Sulaiman, I. M., & Seangwattana, T. (2023). The global convergence of spectral RMIL conjugate gradient method for unconstrained optimization with applications to robotic model and image recovery. *Plos one*, *18*(3), e0281250.
20- Rivaie, M., Mamat, M., June, L. W., & Mohd, I. (2012). A new class of nonlinear conjugate gradient coefficients with global convergence properties. *Applied Mathematics and Computation*, *218*(22), 11323-11332.
21- Powell, M. J. (1986). Convergence properties of algorithms for nonlinear optimization. *Siam Review*, *28*(4), 487-500.
22- Ismaeel, A. A., Houssein, E. H., Khafaga, D. S., Abdullah Aldakheel, E., AbdElrazek, A. S., & Said, M. (2023). Performance of osprey optimization algorithm for solving economic load dispatch problem. *Mathematics*, *11*(19), 4107.
23- Somula, R., Cho, Y., & Mohanta, B. K. (2024). SWARAM: osprey optimization algorithm-based energy-efficient cluster head selection for wireless sensor network-based internet of things. *Sensors*, *24*(2).

Samarra Journal of Pure and Applied Science

www.sjpas.com

p ISSN: 2663-7405
e ISSN: 2789-6838

خوارزميات هجينة متطورة: تحسين بيليكان باستخدام تقنيات التدرج المترافق والعقاب

أزهار زكي عبد الله *، بان أحمد
قسم الرياضيات، كلية علوم الحاسوب والرياضيات، جامعة الموصل

معلومات البحث:
تاريخ الاستلام: 2025/01/15
تاريخ التعديل: 2025/02/16
تاريخ القبـــول: 2025/02/20
تاريخ الـــنشر: 2025/09/30
الكلمات المفتاحية:

معلومات المؤلف

الخلاصة:

مشاكل التحسين المعقدة. لديها قدرة استكشاف مميزة، ولكن لديها بعض المشاكل في إيجاد الحل بدقة وسرعة من خلال أسلوب الاستغلال. لذلك، تم تحسين من خلال استغلال خوارزمية، والتي تم تضمينها كمجتمع أولي لخوارزمية في التهجين الأول، مما تسبب في تحسن بسيط مقترنًا بنتائج الأصلية. تم تهجين باستخدام أسلوب استغلال خوارزمية ، والتي أثبتت قوتها في إيجاد الحل الأمثل بدقة وسرعة من خلال النتائج، والتي تمت مقارنتها بناءً على مقاييس مهمة مثل معدل التقارب وجودة الحل وسرعة التقارب الموضحة من خلال الرسم البياني، وتم تقييم الأداء من خلال بعض وظائف الاختبار العالمية. ويمكن استخدام هذا النهج الجديد في العديد من التطبيقات والمجالات العلمية مثل الاقتصاد والهندسة والطب والعلوم البيولوجية.