ISSN No:-2456-2165

Using the Conjugate Gradient & Parallel Tangent to Improve the Ant Lion Algorithm

Radhwan Basim Thanoon
M.sc. department of mathematics
Educational directorate of Ninevah provence Iraq

Zyiad Adrees Hamad
M.sc. department of mathematics
Educational directorate of Ninevah provence Iraq

Abstract:- In this paper, A hybrid algorithm is proposed, the(ALO) Ant lion algorithm, Along with classical algorithms. The hybrid algorithm optimizes the randomly generated primary community, dependent on a Conjugate Gradient algorithm (CG) and once with a Parallel Shadow algorithm (Partan). The test was applied to (15) high-performance optimization functions of different dimensions and different frequencies. This algorithm relies on the fishing mechanism and is of five main steps for hunting prey such as random walke of ants, building pits, hunting ants in pits, hunting prey, and rebuilding potholes. The (ALO) algorithm is a meta-heuristic algorithm Which was presented in (2015) by *Merjalili*. A hybrid algorithm was used to solve the NP-Hard optimization problem. The hybrid algorithm gave excellent results in terms of exploration, avoiding falling into local solutions, good convergence and working to determine the best (optimal) location

Keywords:- *Ant lion Optimization, meta-Heuristic algorithms, Conjugate Gradient methods, Parallel Tangent method, unconstrained Optimization.*

I. INTRODUCTION

Human has sought in the past to find some solutions to his daily problems, and after experience, a general vision has been formed for him about solutions to these problems in terms of their efficiency in providing accuracy, safety and speed with the best and optimal solution. In the hands of some leading scientists, the area of the Optimization field has expanded greatly in recent years due to the tremendous development in computer technologies, software, high speed and parallel processors as well as artificial neural networks that have a significant impact on many life applications[1].

Optimization is a means of obtaining the best results in specific circumstances. Therefore, optimization techniques were used in many industrial fields, including the aircraft and automotive and electrical industries. Optimization appears everywhere on the ground, in natural phenomena that extend and expand even in our lives. Everyday and presents itself in many physical situations, for example, the reflection of light on a mirror corresponds to the shortest path that affects the mirror, so that optimism can be a model of physical reality[2].

One of the basic principles in our world is to search for the optimum situation or the optimum state, Any better choosing the best choice among all possible decisions in the real life environment. Optimization, or Mathematical Programming, is one of the oldest sciences, and it expands and extends even in our daily lives. The researcher *Yuqi He*, a professor at Harvard University and a member of the American National Engineering Academy, described the importance of optimizm by saying (Optimization is the cornerstone of the development of civilization). As long as there is a human being, he will make every effort to reach the best in many areas, because he desires to reach the greatest degree of happiness in his life with the least amount of effort possible [3].

Optimization can be defined as science to determine the "best" solutions to mathematical problems, which can also be examples of physical facts. It also includes studying the optimal conditions of the problems and their typical structure, determining the mathematical methods of the solution, studying such methods, and computerizing them. It has a wide range of theoretical and practical applications in this field [4].

II. OPTIMIZATION ANT LION

It is a small insect, long wings, and it is characterized by a mesh sweat to about 60 mm, the head is elongated and presented in the form of a short hose, and the sensing pods are short. The larvae of this insect are prey in the form of a short hose in topical pits, made in the soil and fed with the blood of ants. So it is considered a beneficial insect. It is divided into two parts in the way of feeding, living and growing when it is a complete insect feeding on the nectar of flowers. But if it is in the stage of growth and is called the larva and feed on the blood of ants. In this paper, we will focus on the (Ant lion) in the larva stage[5].

This algorithm is based on naturalism presented by *Mirijlalili* in 2015. This algorithm simulates the technique of hunting an (Ant lion) in the natural, the ant lion larva works on a conical hole using a circular path in the hole and throws the sand out of the hole by Its huge jaws, and after preparing the hole it works to bury the caterpillar itself inside the hole under the sand and is waiting in an attempt to catch inside the hole and the edges around the conical are sufficient to drop insects inside the hole and grab the prey and pull it into the soil. After eating the prey, the ant lion will throw the remains of the prey out of the hole and rearrange the hole again for the next catch [6].

➢ *Operators of the Alo Algorithm:*

The ant lion algorithm (AlO) simulates the interaction between ant lion and ants in the hole to model this interaction. Required by ants and in the research space ant lion hunts using the hole. The ants move randomly (Stochastic) in the natural when searching for food. A random movement selects the ant movement form as follows[7].

$$X(t) = [0, cumsum(2r(t_1) - 1), cumsum(2r(t_2) - 1), \dots, cumsum(2r(t_n) - 1)] \quad (1)$$

Whereas cum calculates the cumulative sum, n represents the maximum iterations, r (t) is a random function and can be defined as follows:

$$r(t) = \left\{ \begin{matrix} 1 \; if \; rand > 0.5 \\ 0 \; if \; rand \le 0.5 \end{matrix} \right\} \quad (2)$$

Where t: represents the random walking step, and the rand is a random number generated between [0,1].

The location of the ants is stored using, by optimization, using the following matrix:

$$M_{Ant} = \begin{bmatrix} A_{11} & A_{12} & \vdots & A_{1d} \\ A_{21} & A_{22} & \vdots & A_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \vdots & A_{nd} \end{bmatrix} \quad (3)$$

Since M_{Ant}represents a storage array of locations for each ant element, $A_{ij}$where $(j - th)$shows the value of the variable, $(i - th)$is the dimension of ants, n is the number of ants, and d is the number of variables, we should note that Ants are similar to flocks of birds (PSO) and individuals in the genetic algorithm, the position of the ants indicates the parameters for a particular solution. The M_{Ant}matrix stores all the ant positions (variables of all solutions) during the strongest during optimization. The matrix [8] to evaluate an ant is used for the fitness function (goal) during the optimization. The next matrix stores the fitness value of all the ants.

$$M_{oA} = \begin{bmatrix} f([A_{11} & A_{12} & \vdots & A_{1d}]) \\ f([A_{21} & A_{22} & \vdots & A_{2d})] \\ \vdots & \vdots & \ddots & \vdots \\ f([A_{n1} & A_{n2} & \vdots & A_{nd})] \end{bmatrix} \quad (4)$$

Whereas, M_{oA} represents a storage array of locations for each ant element, A_{ij} where $(j - th)$ shows the value of the variable$(i - th)$) is the dimension of ants, , n is the number of ants, , d is the number of variables, and , f is a function (goal). In addition to ants, we assume that ants are also hiding somewhere in the search space. In order to memorise their positions and fitness values, the following matrices are used:

$$M_{Antlion} = \begin{bmatrix} A_{11} & A_{12} & \vdots & A_{1d} \\ A_{21} & A_{22} & \vdots & A_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \vdots & A_{nd} \end{bmatrix} \quad (5)$$

Where $M_{Antlion}$ is the matrix for storing the location of each ant lion $A_{ij}$where $(j - th)$ shows the value of the variable, $(i - th)$ is the dimension of the ants, n is the number of ants, and d is the number of variables.

$$M_{OAL} = \begin{bmatrix} f([A_{11} & A_{12} & \vdots & A_{1d}]) \\ f([A_{21} & A_{22} & \vdots & A_{2d})] \\ \vdots & \vdots & \ddots & \vdots \\ f([A_{n1} & A_{n2} & \vdots & A_{nd})] \end{bmatrix} \quad (6)$$

WhereM_{OAL} is the matrix to provide physical fitness for each prisoner, A_{ij} where $j - th$ shows the value of the variable, $i - th$ is (the dimension) of ants, n is the number of ants, d is the number of variables, and it is the goal function. During improvement, the following conditions are applied [6].Ants move around the search area using a different random walk.

- Random paths are applied to all the dimensions of the ants.
- Random paths are affected by ant traps
- Ants can build pits appropriate for their fitness (larger pit).
- An ant lion with larger pits has a greater chance of catching ants.
- Every ant can be discovered by an ant lion in every occurrence and elite (the best ant lion).
- The random walk range is reduced to adapt to simulated ant sliding toward the ant lion.
- If the ant becomes tougher (stronger), this means that it has also been pulled under the sand by the ant lion
- An ant lion repositions itself to new prey and rearranges the pit to catch another prey.

➢ *Random Walks of Ant:*

The base of the random path depends on equation (7), so that the ant happens its location with the random path change in each ideal step. Each search space has limits (variable scope). It cann't be used directly to update the ant site. In order to maintain random paths within the search space. It was applied using the following equation:

$$x_i^t = \frac{(x_i^t - a_i) * (d_i - c_i^t)}{(d_i^t - a_i)} + c_i \quad (7)$$

Where $a_i$is the minimum random walk of the variable i, $d_i$is the maximum amount of random walk in the first variable, c_i^t is the minimum of the variable i in the frequency t, and c_i^tindicates the maximum of the variable i in the frequency t. Equation. (7) It should be applied at all iterations to ensure random paths occur within the search space [9]:

➢ *Trapping in ant lion pits:*

As discussed above, the random ant course is affected by ant traps. In order to model this assumption mathematically, the following equations have been proposed:

$$c_i^t = Antlion_j^t + c^t \quad (8)$$
$$d_i^t = Antlion_j^t + d^t \quad (9)$$

Where c_i^t is the minimum for all variables in iteration t, d ^ t indicates the vector including the maximum of all variables in iteration t , c_i^t is the minimum for all variables i, d_i^t j is the maximum for all variables i and Ant liont The position of the choice j appears in t repetition. In equation (8) and (9) it was found that ants walk randomly into a ball hole defined by vectors c and d around a specific ant lion [6].

➢ *Bulding Trap:*

In order to model the ability of hunting an ant lion, you build a larger pit to increase the opportunity in hunting, using the pattern of a large circular pit, it is assumed that the ants are trapped in only one choice of the ant lion and the ants are chosen based on the value of their fitness during the improvement. A large pit helps ants attract more sliding ants to a lion, which gives the greatest opportunity for hunting. [9].

➢ *Silding ants towards ant lion:*

With the mechanisms proposed so far, an ant lion can make traps proportional to their fitness and ants are required to move randomly. However, the ant lion starts throwing sand into the hole as soon as you realize the ant is in the trap. This behavior slides down the trapped ant who is trying to escape. For mathematically modeling this behavior, a random walk radius of ants isreduced to an adaptive field overload. The following equations:

$$c^t = \frac{c^t}{I} \quad (10)$$
$$d^t = \frac{d^t}{I} \quad (11)$$

Whereas, c^t is the minimum for all variables in repetition t, and d^tindicates the vector including the maximum of all variables in t, whereas $I = 10^w \frac{t}{T}$where t is the current repetition, T is the limit The maximum number of iterations, and w is a fixed constant based on the current iteration ($w = 2$ When$t > 0.1T$ ، $w = 3$ When w $t > 0.5T$ ، $w = 4$ When $t > 0.75T$ ، $w = 5$ When $t > 0.9T$ ، و $w = 6$ When $t > 0.95T$). Basically, the constant w can adjust the precision of exploitation of these equations by shrinking the radius of the ant update positions and simulating the ant sliding process within the pits. This guarantees the exploitation of the search space.

➢ *Catching prey and re-bulding the pit:*

The last stage of hunting is when an ant reaches the bottom of a hole and is caught in the ant's jaw. After this stage, the ant lion pulls the ants into the sand and consumes their body. To simulate this process, it is assumed that catching prey occurs when ants go inside the sand from their corresponding counterpart. Wanted, then an ant lion to update its position to the last position of the hunted ant to enhance its opportunity to hunt new prey. The following equation is proposed in this regard

$$Antlion_j^t = Ant_i^t \ \ if \ \ f(Ant_i^t) > f\left(Antlion_j^t\right) \quad (12)$$

Selection position appears $Antlion_j^t$ From the current iteration, Ant_i^t indicates the position of the ants from the current iteration [7].

➢ *Elitism:*

Elite is an important feature of evolutionary algorithms that allow them to maintain the best solutions obtained at any stage of the optimization process. In this study the best ant lion obtained so far in each iteration is preserved and considered the elite. Since the elite is the best site for my ant lion, it should be able to influence the movements of all ants during repetition. Each ant should walk randomly around a specific group and elite simultaneously as given

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \quad (13)$$

Where R_A^t is the random walk around the ants specified in the repetition t, R_E^t is the random walk around the elite in the repetition t, Ant_i^t indicates the position of the ant in the repetition [8].

III. IMPORTANCE RESEARCH

The importance of the research lies in improving the performance of the ant lion algorithm by hybridizing it with the classical methods which are the Conjugate Gradient Method and Parallel Tangent Method. It is a suggested method or method for solving unrestricted, high-measurement optimization issues that are considered difficult issues (NP-hard) to help researchers and specialists to take advantage of the proposed hybrid algorithm to solve this type of problem as well as all issues that fall within its framework.

IV. RESEARCH OBJECTIVE

The aim of the research is to propose a hybrid algorithm based on the ant lion algorithm. My Agency: To propose a new hybrid algorithm consisting of the ant-lion algorithm with the conjugated directional algorithm called “Ant-CG” and the parallel shadow algorithm called “Ant-PARTAN”. It works to solve high-volume issues.

➢ *Step Algorithm:*

- A random preparation for a community of ants and an ant lion, calculating the best value from the fitness function for both ants and an ant lion, choosing the best ant lion, and finding the cumulative account for the best elite.
- When selecting the best and the value is unsatisfactory for all ants, the ant lion determines the circular motion of the search area

- We are working on updating the situation through the ant lion using equations (10) and (11).
- Create a random walk using equation (7).
- Update the position of each ant using Equation (13).
- Calculate the value of the fitness function for all ants.
- The possibility of replacing ants with an ant lion compared to if the ant was tested.
- Then the elites are modernized, as they get an ant lion better than the elite.
- Repeat again.

V. CONJUGATE GRADIENT METHODS

Attention was paid to the associated gradient methods for two reasons, The first is that these methods are among the oldest and best known techniques for solving systems of linear equations of large dimensions. The second reason is that these methods can be adapted to solve problems of nonlinear optimization [9].

These methods have advantages that place them between the steepest descent method and the Newton method, Because these methods only require calculation of the first derivatives, they do not need to calculate and store the second derivatives needed by the Newton method, and they are faster than the steepest descent method, that is, they overcame the slow convergence of this method and because they do not need to calculate the hessian matrix or any of its pproximations They are widely used to solve optimization issues of high dimension [4].

The two types are known for these methods. The first type is linear conjugation methods, also known as quadratic conjugation, and is sometimes known as pure conjugate gradient, These methods are used to minimize convex quadratic functions. The second type is known as nonlinear conjugate gradient methods, also known as non-quadratic conjugate gradient, used to minimize convex general functions or nonlinear general functions.

In the method of linear conjugate Gradient was first proposed by Hestenes and Stiefel in (1950) as an iterative method originally used as an alternative to Gaussian elimination to solve large linear systems with a positive definite matrix of positive coefficients on the computer [10].

Fletcher and Reeves developed the Hestenes and Stifle method and introduced the first method of conjugation of nonlinearity (1990). It is one of the first known techniques to solve nonlinear optimization issues of large dimensions. Over the years many different methods have been proposed along the lines of the original version of this method, and some of these methods are widely used in application [9].

➢ *Classical Conjugate Gradient Method*

Linear Gradient is an iterative method to solve the problem of miniaturization

$$\min f(x) = \frac{1}{2}x^T Gx - b^T x + c \qquad (14)$$

b is a vector , c constant value , and G is a positive-definite matrix of the n × n type. Equation (14) can be obtained in an equivalent manner as a system of linear equations:

$$Gx = b \qquad (15)$$

This makes the only solution to the problem (14) the same solution to the system (15). Thus, we can prepare the method of co-gradation either algorithm to solve linear or technical systems to find the smallest value of quadratic convex functions. [11].

Note that the gradient of (f) is equal to the residual (negative gradient) of the linear system, ie:

$$\nabla f(x) = Gx - b \qquad (16)$$

While $x = x_k$

$$g_k = Gx_k - b \qquad (17)$$

A noteworthy feature is that the method of gradient conjugation has the potential to generate a vector group with property, As we can calculate the new direction d_{k+1} using the previous direct d_k and the current gradient g_k, And is selected so that it is a linear composition of ($-g_k$) (the direction of extreme descent) and d_k (the previous direction) and we do not need to know all the previous directions $d_0, d_1, \ldots, d_{n-1}$ for the associated group where d_{k+1} is associated with all the previous trends, and this attribute makes this method requires storage space and few calculations we write

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k \qquad (18)$$

Where β_{k+1} represents a numerical quantity, defined as d_{k+1} and d_k are correlated for the matrix G, we choose the first search direction at the initial point $d_0 = -g_0$ (The direction of extreme descent) [12], that the convergence rate of the gradient methods is linear unless iteration is repeated [13].

➢ *Classical Conjugate Gradient Method*

Step 1 : choose an initial value x_0 , put $d_0 = -g_0$ ، $k = 0$ and $\varepsilon > 0$.

Step 2 : calculate the length of step $\alpha_k > 0$، *satisfy Wolf condition*

$$f(x_k + \alpha d_k) \le f(x_k) + c_1 \alpha g_k^T d_k$$
$$g_k^T + g_{k+1} d_k \ge c_1 g_k^T d_k$$

Whereas $0 < c_1 < c_2 < 1$

Step 3 : calculate $x_{k+1} = x_k + \alpha_k d_k$ if $\|g_{k+1}\| < \varepsilon$ Then stop

Step 4 : calculate β_{k+1} and generate the vector $d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$

ISSN No:-2456-2165

Step 5 : put $k = k + 1$ and go to step 2 .

The associated gradient algorithms differ according to the different tests of the parameter β_{k+1} in step 4 .

The most popular formulas were summarized in the following table:

$\beta_{k+1}^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}$	Hestenes -Stiefel 1952
$\beta_{k+1}^{FR} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$	Fletcher-Reeves(FR) 1964
$\beta_{k+1}^{PR} = \frac{g_{k+1}^T y_k}{g_k^T g_k}$	Polak-Ribiere (PR) 1969
$\beta_{k+1}^{PR+} = max\left\{0, \frac{g_{k+1}^T y_k}{g_k^T g_k}\right\}$	Polak-Ribiere+ (PR+) 1984
$\beta_{k+1}^{CD} = \frac{g_{k+1}^T g_{k+1}}{-d_k^T g_k}$	Dixon (CD) 1987
$\beta_{k+1}^{LS} = \frac{g_{k+1}^T y_k}{-d_k^T g_k}$	Liu-Story (LS) 1991
$\beta_{k+1}^{DY} = \frac{g_{k+1}^T g_{k+1}}{d_k^T y_k}$	Dai-Yuan(DY) 1999

Table 1:- Different options for the parameter β_{k+1}in the conventional conjugation algorithms [14].

VI. PARALLEL TANGENT GRADIENT

➢ *Zigzagging Phenomenon:*

The descent algorithm tends to be near the optimal point as small orthogonal steps are taken, Which is known as the Zigzagging phenomenon. To understand the Zigzagging phenomenon, let us look at an objective function with concentric ellipse lines with the minimum limit for the best point of P * as shown in Figure 1 below [6]:

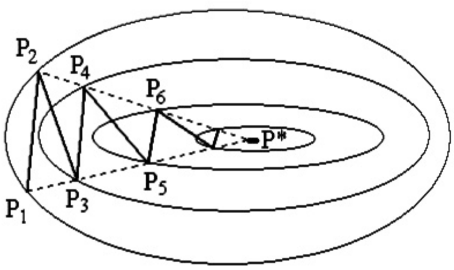

Fig 1:- The Zigzagging phenomenon in the ellipse

➢ *Parallel Tangent Gradient:*

Parallel Tangent Gradient : Parallel Tangent is a technique that combines many simple characteristics of gradient methods. It uses the geometrical property in the formation of quadratic functions. It works in the form of elliptical lines. It has several formulas, including the gradient of the parallel Tangent , which is multidimensional in the combined gradient methods. Therefore, this technique is based on the improvement of the steep way. In Figure (1) we can see two ascending rectangles approaching the optimal point P*. This indicates that the search is from point P_3 and does not lead to the direction of point P_4. but along the straight line connecting P_1 and P_3[15]. The procedure to be achieved in the objective function to the minimum on successive straight lines is to determine the direction alternating position by the previous point or gradient directions. This method does not involve the construction of a reciprocal light with a conjugate vector that can be created through two direction vector with the conjugate. The property is based on convergence and is called the parallel Tangent method [16].

In Fig. 1 this way the optimal point P* can be located after three steps as :

- From P_1 to P_2 along the gradient to P_1.
- From P_2 to P_3 along the gradient to P_2.
- At last from P_3 to P_1 along the gradient to P_3.[17]

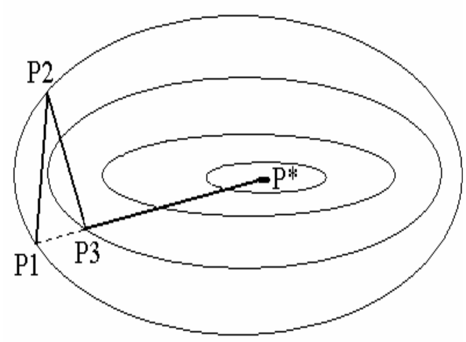

Fig 2:- Represents the base of the parallel tangent gradient in finding the optimal point P*

Figure 2 shows a diagram of the parallel tangent gradient algorithm. Note that the dots were numbered. When individual points, for example, after the point P_3, mean (p_5 , p_7 , p_9 …..) It is the result of gradation in the search, while the post even P_2 points mean (p_4 , p_6 , p_8 …..) acceleration can be obtained, in other words using a point difference system, be either odd or even. The P_{2k} point of the conjugation points can be determined by acceleration from P_{2k-4} to P_{2k-1} when K=2,3,…..,N that means $P_{2k} = \Omega(P_{2k-1}, P_{2k-4})$, $K \geq 2$

The Ω is an acceleration function, so acceleration requires a process of taking the minimum point on the line that falls between the odd points and even points.

Consider that the gradient of the parallel tangent shown in Figure (2) reduces different functions in the number of variables. P_0 is the starting point, then P_2 and P_3 are standard gradually steps, and an optimal duplicates is followed for the two steps. In the first step, the acceleration and in the second step is the standard gradient [18].

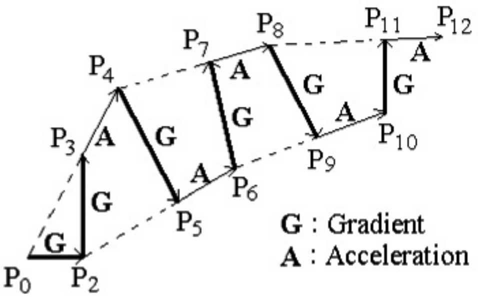

Fig 3:- The path taken by the gradient-PARTAN.

➢ *Parallel Tangent Algorithm:*

1. Choose a starting point and assume $d_0 = - g_0$

So $x_1 = x_0 + \lambda d_0$

after this choosing $d_2 = - g_2$

Then, the fourth point is created by moving in one direction with ($x_3 – x_1$).

$d_3 = - (x_3 – x_1)$

This point is referred to as an acceleration step after the determination of whether to follow the process in succession with gradual rotation and acceleration of steps. And so on

2. $d_i = -g_i$ for $i = 0,2, ... ,2n - 2$

$d_i = - (x_i - x_{i-2})$ for $i = 3,5, ... ,2n - 1$

In this way we will reach the (minimum) in (n) of the square surfaces dimensions not exceeding 2n steps.

In general the gradient is interrelated and not reciprocal and follows the following characteristics:

1. The direction of the search is most appropriate $d_i^T g_i < 0$
2. The vectors $(x_2 - x_0), (x_2 - x_0), ..., (x_{2n} - x_{2n-2})$ mutually conjugancy.
3. Points $x_4, x_6, ..., x_{2n}$,The minimum search area is a straight extend by $d_1 \& d_2$, $g_2 \& g_4, ... (d_1, d_2, ..., d_{2n})$
4. Gradient vectors $g_2, g_2, ..., g_{2n}$be orthogonal.

The parallel Tangent algorithm stops when $\| g_{k+1} \|$ is small enough.

The ideal calculation should end in n of iterations , We choose an elementary point x_0, especially when the algorithm approaches k <n of iterations if the Hessian Matrix and only k function had special values. These properties are tracked because of repetitive relationships in d_i which are designed to ensure that the search direction is associated with the Hessian Matrix. The behavior of the parallel tangent algorithm depends on the exact accuracy of the calculation over the accuracy of these numerical results [19].

VII. PROPOSED ALGORITHM

A new hybrid algorithm has been proposed by linking the ant lion algorithm with classic algorithms which are divided into two parts: the parallel shadow algorithm is divided into three sections which are the conjugated direction (CG) and the derivative difference $(\boldsymbol{g_k - g_{k-1}})$ If the point is my pair and the difference of points if the point is odd $(\boldsymbol{x_i - x_{i-1}})$ **.**The main goal of this work is to improve the algorithm for examples of ant lions .

As for the steps of the hybrid algorithm, they are as follows:

➢ A random preparation for a community of ants and an ant lion, calculating the best value from the fitness function for both ants and an ant lion, choosing the best ant lion, and finding the cumulative account for the best elite.
➢ Entering random values depending on the conjugated direction once and parallel shadow again.
➢ When selecting the best and the value is unsatisfactory for all ants, the ant lion determines the circular motion of the search area
➢ We are updating the situation through the ant lion using equations (10) and (11).
➢ Create a random walk using equation (7).
➢ Update the position of each ant using Equation (13).
➢ Calculate the value of the fitness function for all ants.
➢ The possibility of replacing ants with an ant lion compared to if the ant was tested.
➢ Then the elites are modernized, as they get an ant lion better than the elite.
➢ Repeat again.

VIII. PRACTICAL SIDE

In order to evaluate the performance of the hybrid algorithm in solving optimization problems, the hybrid algorithm was chosen using (15) functions from standard functions in order to compare with the original algorithm before hybridization, Table (2) shows the details of the test functions as well as the special range and minimum value (f_{min}) and minimums The upper limits of each function and the number of iterations (1000) have been used repeatedly and the dimensions that indicate the number of variables in the design. As for the number of research elements, it was in table (3) that consists of swarm of (10) elements, and in table (4) the swarm consists of (20) an item.

Function	Dim	Range	$Fmin$
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100,100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[−10,10]	0
$F_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_i)^2$	30	[−100,100]	0
$F_4(x) = \max_i\{\lvert x_i \rvert . 1 \le i \le n\}$	30	[−100,100]	0
$F_5(x) = \sum (100 * (x_{i+1}{:}\,\mathrm{dim}\ \ ) - (x_i{:}\,\mathrm{dim}\ \ )^2 + ((x_{i+1}{:}\, dim - 1) - 1)^2$	10	[−30,30]	
$F_6(x) = \sum_{i=1}^{n} ([x_i\ \ + 05])^2$	30	[−100,100]	0
$F_7(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	[−500,500]	−418.9829
$F_8(x) = \sum_{i=1}^{n} \lvert x_i^2 - 10\cos(2\pi x_i) + 10 \rvert$	30	[−5.12,5.12]	0
$F_9(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	[−32,32]	0
$F_{10}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[−600,600]	0
$F_{11}(x) = \frac{\pi}{n}\{10\sin(\pi y_i) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1}) + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	30	[−50,50]	0
$y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i + a)^m & x_i < a \end{cases}$			
$F_{12}(x) = 1.0\{\sin^2(3\pi x_i) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + \sin^2(\pi x_{i+1}) + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)\}$	30	[−50,50]	0
$F_{13}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1 - (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^4$	4	[−5,5]	0.00030
$F_{14}(x) = 4x_i^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[−5,5]	−1.0316
$F_{15}(x) = [1 + (x_1\ + x_2\ + 1)^2(19 - 14x_1\ + 3x_1^2 - 14x_2\ + 6x_1x_2 + 3x_2^2)] * [30 + (2x_1\ - 3x_2\ )^2(18 - 32x_1\ + 12x_1^2 + 48x_2\ - 36x_1x_2 + 2x_2^2)]$	2	[−5,5]	−1.0316

Table 2:- shows the details of the test functions and the minimum value for each function

		Ant–PARTAN			
Function	**Ant**	**CG**	$X_K - X_{K-1}$	$G_K - G_{K-1}$	**Ant_CG**
FI	8.3143e-09	6.0682e-15	2.254e-15	6.8331e-15	0
F2	0.0010586	2.0998e-08	2.4762e-08	1.6447e-08	0
F3	0.00017321	1.2765e-14	2.0253e-14	8.3542e-15	0
F4	0.0059878	4.0771e-08	4.2866e-08	4.4024e-08	0
F5	3.381	1.7742e-06	0.0001456	7.4908e-05	0.00023949
F6	1.1123e-08	5.2253e-09	1.0075e-08	5.63e-09	2.1104e-08
F7	-2373.5251	-2045.7002	-1987.0637	-1896.3608	-3083.8654
F8	20.8941	0	0	0	0
F9	0.00010502	2.8008e-08	3.9519e-08	2.3728e-08	8.8818e-16
F10	0.22388	1.5321e-14	5.107e-15	4.3965e-14	0
F11	3.8172	1.8596e-08	4.6238e-08	5.603e-08	1.5044e-07
F12	0.010988	6.7085e-07	1.066e-06	0.021039	6.4622e-07
F13	0.0013346	0.0007896	0.00076673	0.00078827	0.0012967
F14	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
F15	3	3	3	3	3

Table 3:- testing the ant lion algorithm using 10 elements and iterations for only 1000

		Ant-PARTAN			
Function	Ant	CG	$X_K - X_{K-1}$	$G_K - G_{K-1}$	Ant_CG
FI	3.8483e-09	4.1901e-15	8.3396e-15	8.7496e-16	0
F2	3.1676e-05	1.7142e-08	1.7548e-08	9.6064e-09	0
F3	9.613e-05	8.3265e-15	6.4849e-15	4.8982e-15	0
F4	0.0015428	2.6721e-08	5.4033e-08	2.6257e-08	0
F5	96.7172	5.2345e-06	4.6289e-05	3.2159e-07	6.1677e-05
F6	2.593e-09	7.3338e-09	6.4917e-09	8.661e-09	3.7822e-09
F7	-2669.8156	-2106.9753	-2165.2043	-2162.4652	-2280.3171
F8	17.9092	0	0	0	0
F9	2.0133	2.168e-08	2.8497e-08	2.2263e-08	8.8818e-16
F10	0.28048	9.1038e-15	1.088e-14	1.7764e-14	0
F11	3.3116e-09	3.5293e-09	9.8838e-10	1.051e-07	3.4427e-09
F12	4.8259e-09	1.0201e-08	7.6071e-09	0.021024	3.3374e-09
F13	0.020367	0.00030932	0.00069527	0.00064486	0.00054517
F14	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
F15	3	3	3	3	3

Table 4:- testing the ant lion algorithm using 20 elements, and repeating for only 1000

IX. DISCUSSION OF NUMERICAL RESULTS

The results in the previous tables show the success of the hybrid algorithm in each of the ant lion algorithm with the conjugated direction as well as the ant lion algorithm with the parallel shadow algorithm in finding the optimum solution for (15) functions of the standard high-measurement test functions compared to the original algorithm itself and this confirms the success of Hybridization process. To note that the functions $(F_1, F_2, F_3, F_4, F_8, F_{10})$ have given an ideal solution with the algorithm of the conjugated trend in both tables.

As for the functions (F_5 وF_9, F_{12}, F_{13}) In the same hybrid algorithm, it was observed that the results improved in comparison with the original algorithm, but the results did not reach the optimum degree. As for the rest of the functions, which are ($F_6, F_7, F_{11}, F_{14}, F_{15}$) we may observe the results either It is stable between the original algorithm and the hybrid algorithm, or there is no noticeable improvement in it.

In the case of the results with the ant-lion algorithm with the shadow algorithm, we may notice that the best (optimal) solution that the algorithm has been able to connect to automatically only at the function (F_8) in both tables is both. As for the rest of the functions, I worked between improving some functions or the stability of some other functions. In both cases, hybridization leads to obtaining results better than the original algorithm in some functions and not in all functions. Either way, on the one hand, the best conjugate directional algorithm or parallel shadow algorithm through crossbreeding with post-intuitive algorithms prove that the conjugated directional algorithm is better in hybridization .

X. CONCLUSIONS

Hybridization of the post-intuitive algorithm with classic algorithms contributed to improving the performance of single post-intuitive algorithms by increasing the speed of convergence, and also led to an improvement in the quality of the resulting solutions by increasing their exploratory capabilities, as numerical results demonstrated the ability of hybrid algorithms to solve optimization issues Different. The results of the Ant-CG algorithm and the Ant-Partan algorithm were compared with the optimization algorithm of the ant lion itself, which led to excellent results, as the most optimal comprehensive solution was obtained for most of the test functions, and this is what the research results showed.

REFERENCES

[1]. Chong, E.K. and S.H. Zak, *An introduction to optimization*. 2004: John Wiley & Sons.
[2]. Gould, N., *An introduction to algorithms for continuous optimization*. 2006, Oxford University Computing Laboratory Notes.
[3]. Weise, T., *Global optimization algorithms-theory and application*. Self-Published Thomas Weise, 2009.
[4]. Sun, W. and Y.-X. Yuan, *Optimization theory and methods: nonlinear programming*. Vol. 1. 2006: Springer Science & Business Media.
[5]. Ali, E., S.A. Elazim, and A. Abdelaziz, *Ant Lion Optimization Algorithm for optimal location and sizing of renewable distributed generations*. Renewable Energy, 2017. **101**: p. 1311-1324.
[6]. Mirjalili, S., *The ant lion optimizer*. Advances in engineering software, 2015. **83**: p. 80-98.

[7]. Petrović, M., et al., *The ant lion optimization algorithm for flexible process planning.* Journal of Production Engineering, 2015. **18**(2): p. 65-68.
[8]. Mirjalili, S., P. Jangir, and S. Saremi, *Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems.* Applied Intelligence, 2017. **46**(1): p. 79-95.
[9]. Nocedal, J. and S.J. Wright, *Numerical optimization 2nd*. 2006, Springer.
[10]. Hestenes, M.R. and E. Stiefel, *Methods of conjugate gradients for solving linear systems*. Vol. 49. 1952: NBS Washington, DC.
[11]. Pedregal, P., *Introduction to optimization*. Vol. 46. 2004: Springer Science & Business Media.
[12]. Weise, T., *Global optimization algorithms-theory and application.* Self-published, 2009. **2**.
[13]. Powell, M.J.D., *Restart procedures for the conjugate gradient method.* Mathematical programming, 1977. **12**(1): p. 241-254.
[14]. Andrei, N. *Conjugate gradient algorithms for molecular formation under pairwise potential minimization*. in *Mathematical modeling of environmental and life sciences problems. Proc. of the 5th Workshop, Constanta, Romania, 10–13 September*. 2006.
[15]. Hestenes, M.R. and E. Stiefel, *Methods of conjugate gradients for solving linear systems.* Journal of research of the National Bureau of Standards, 1952. **49**(6): p. 409-436.
[16]. Metras, B.A. and K.B. Rasheed, *Two New Approaches for PARTAN Method.*
[17]. Moallem, P., et al. *A novel fast backpropagation learning algorithm using parallel tangent and heuristic line search*. in *Proceedings of the 10th WSEAS international conference on Computers*. 2006. World Scientific and Engineering Academy and Society (WSEAS).
[18]. Moallem, P., S.A. Monadjemi, and B. Mirzaeian, *PTGMVS: Parallel tangent gradient with modified variable steps for a reliable and fast MLP neural networks learning.* KUWAIT JOURNAL OF SCIENCE AND ENGINEERING, 2008. **35**(1B): p. 65.
[19]. Pedregal, P., *Introduction to optimization*. Vol. 46. 2006: Springer Science & Business Media.