Cuttlefish Algorithm – A Novel Bio-Inspired Optimization Algorithm

Adel Sabry Eesa, Adnan Mohsin Abdulazeez Brifcani, Zeynep Orman

Abstract— In this paper, a new meta-heuristic bio-inspired optimization algorithm, called Cuttlefish Algorithm (CFA) is presented. The algorithm mimics the mechanism of color changing behavior used by the cuttlefish to solve numerical global optimization problems. The patterns and colors seen in cuttlefish are produced by reflected light from different layers of cells including (chromatophores, leucophores and iridophores) stacked together, and it is the combination of certain cells at once that allows cuttlefish to possess such a large array of patterns and colors. The proposed algorithm considers two main processes: *reflection* and *visibility*. Reflection process is proposed to simulate the light reflection mechanism used by these three layers, while the visibility is proposed to simulate the visibility of matching pattern used by the cuttlefish. These two processes are used as a search strategy to find the global optimal solution. Efficiency of this algorithm is also tested with some other popular biology inspired optimization algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO) and Bees Algorithm (BA) that have been previously proposed in the literature. Simulations and obtained results indicate that the proposed CFA is superior to other algorithms.

Index term— Cuttlefish algorithm, Reflection, Visibility, Optimization, Chromatophores, Iridophores, Leucophores, Test functions.

1 Introduction

GLOBAL optimization is a field with applications in many areas of science, engineering, economics, and others, where mathematical modeling is used. Without loss of generality, the optimization maybe defined as the search for a vector x_0 in a possible solution set X minimizing a target function f so that $\forall x \in U \subseteq X$: $f(x) \geq f(x_0)$. For $U = X$, x_0 is called a global optimum, otherwise it is called a local optimum of f in X [29]. Global optimization algorithms are usually broadly divided into deterministic and meta-heuristic [10]. Deterministic algorithms tend to use gradient technique and find greater use in solving unimodal problems. While meta-heuristic models tend to learn as they run, and tend to be more intelligent and adaptive. Meta-heuristic methods are usually faster in locating a global optimum than deterministic ones. The components of any meta-heuristic algorithms are: intensification and diversification, or exploitation and exploration [28]. Diversification means to generate diverse solutions so as to explore the search space on a global scale, while intensification means to focus the search in a local region knowing that a current good solution is found in this region. A good balance between intensification and diversification should be found during the selection of the best solutions to improve the rate of algorithm convergence. The selection of the best ensures that solutions will converge to the optimum, while diversification via randomization allows the search to escape from local optima and, at the same time, increases the diversity of solutions. A good combination of these two major components will usually ensure that global optimality is achievable.

Most of meta-heuristic algorithms are nature-inspired such as Ant Colony Optimization ACO, Particle Swarm Optimization PSO, Bees Algorithm BA, etc. that have previously been proposed by researchers. Some of these studies [1] have been inspired by animal behaviors for developing optimization techniques. For example, ACO algorithm proposed by Dorigo et al. [2], is inspired by the research on the behavior of ant colonies. BA proposed by D.T. Pham et al. [3], is inspired by the food foraging behavior of honey bees. PSO algorithm proposed by Kennedy and Eberhart [4], models the social behavior of bird flocking or fish schooling.

Recently, new meta-heuristic approaches are presented by several researchers. For example, collective animal behavior CAB algorithm proposed by Erik Cuevas et al. [5] is inspired from a group of animals which interact with each other that is based on the biological laws of collective motion. A gravitational search algorithm GSA, proposed by Esmat Rashedi et al. [6] is based on the law of gravity and mass interactions. Bumble bees mating optimization BBMO algorithm presented inYannis Marinakis et al. [7] simulates the mating behavior of the bumble bees. Parliamentary optimization algorithm POA proposed by Ali Borji [8] is motivated from human social behaviors in political environments. Bat Algorithm BA proposed by Xin-She Yang [9] is based on the echolocation behavior of bats. Firefly algorithm FA proposed by Xin-She Yang [11] is based on flashing characteristics of fireflies.

In this paper, a new meta-heuristic optimization algorithm that is inspired based on the mechanism of color changing behavior of cuttlefish is presented to find the optimal solution in numerical optimization problems. The patterns and colors seen in cuttlefish are produced by reflected light from different layers of cells stacked together, and it is the combination of certain cells at once that allows cuttlefish to possess such a large array of patterns and colors. The proposed algorithm mimics the light reflection process through the combination of these layers, and the visibility of matching pattern process used by cuttlefish to match its background. The algorithm divides the population (cells) into four groups, each group works independently sharing only the best solution. Two of them used as a global search, while others used as a local search.

This paper is organized as follows. In Section 2, cuttlefish skin components and color changing behavior is introduced. In Section 3, the proposed CFA algorithm and its characteristics are described in detail. Section 4 presents the experimental results and the comparative study. Finally, conclusions are given in Section 5.

2 Cuttlefish Skin Components

Cuttlefish [12, 13] is a type of cephalopods which is well-known for its abilities to change its color to either seemingly disappear into its environment or to produce stunning displays. The patterns and colors seen in cephalopods are produced by different layers of cells [14] stacked together including chromatophores, leucophores and iridophores, and it is the combination of certain cells operations of reflecting light and matching patterns at once that allows cephalopods to possess such a large array of patterns and colors. These layers are described as follows:

Chromatophores: are groups of cells that include an elastic saccule that holds a pigment, as well as 15-25 muscles attached to this saccule [15]. These cells are located directly under the skin of cuttlefish. When the muscles contract, they stretch the saccule allowing the pigment inside to cover a larger surface area. When the muscles relax, the saccule shrinks and hides the pigment [16].

Iridophores: are found in the next layer under the chromatophores [17, 18]. Iridophores are layered stacks of platelets [19] that are chitinous in some species and protein based in others. They are responsible for producing the metallic looking greens, blues and golds seen in some species, as well as the silver color around the eyes and ink sac of others. Iridophores work by reflecting light [19] and can be used to conceal organs, as is often the case with the silver coloration around the eyes and ink sacs. Additionally, they assist in concealment and communication.

Leucophores: these cells are responsible for the white spots occurring on some species of cuttlefish, squid and octopus [15]. Leucophores are flattened, branched cells that are thought to scatter and reflect incoming light. In this way, the color of the leucophores will reflect the predominant wavelength of light in the environment [20]. In white light they will be white, while in blue light they will be blue. It is thought that this adds to the animal's ability to blend into its environment.

2.1 Mechanisms of skin color change

Chromatophores cells contain red, orange, yellow, black, and brown pigments. But a set of mirror-like cells (iridophores and leucophores) allows cuttlefish skin to assume all the rich and varied colors of its environment. The appearance of the cuttlefish thus depends on which skin elements affect the light incident on the skin. Light may be reflected by either chromatophores or by reflecting cells (iridophores or leucophores) or a combination of both, and it is the physiological changeability of the chromatophores and reflecting cells that enables the cuttlefish to produce such a wide repertoire of optical effects. A diagram in Fig1 of Cuttlefish skin detailing the three main skin structures (chromatophores, iridophores and leucophores), two example states (a, b) and three distinct ray traces (1, 2, 3) show the sophisticated means by which cuttlefish can change reflective color [27].

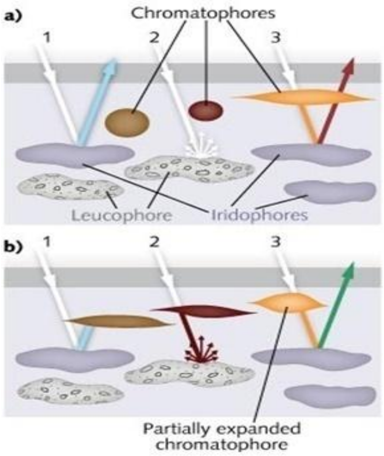

Fig. 1. Diagram of cuttlefish skin detailing the three main skin structures (chromatophores, iridophores and leucophores)

3 Proposed Cuttlefish algorithm (CFA)

The proposed algorithm mimics the work of the three cell layers that are used by cuttlefish to change its skin colors. To do this, we reordered the six cases shown in Fig1 to be as shown bellow.

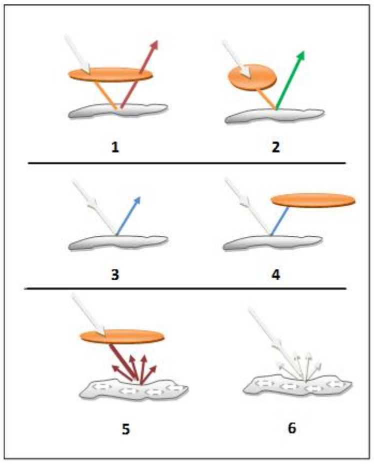

Fig. 2. Reorder of the six cases in Figure 5

From Fig2 we can assume two main processes (*reflection* and *visibility*). Reflection process represents the mechanism used by cuttlefish to reflect incoming light and it can by any case of the six cases considered in Fig2. While visibility is representing the matching pattern clarity that the cuttlefish try to simulate the patterns appear in its environment. We assumed that the final pattern is the global optimum solution, while visibility is the difference between the best solution and the current solution. The proposed cuttlefish algorithm CFA is designed based on these two processes (*reflection* and *visibility*) and they used as a search strategy to find the new solutions. The formulation of finding the new solution (*newP*) using reflection and visibility is described in (1).

$$newp = reflection + visibility \qquad (1)$$

As other meta-heuristic optimization algorithms, CFA starts with random solutions for initializing the population. Then the six cases shown in Fig6 are applied until stop condition is meeting. The main steps of CFA algorithm are summarized as follow:

1- Initialize the population with random solutions, calculate and keep the best solution and the average value of the best solution's points.
2- Use interaction operator between chromatophores and iridophores cells in case 1 and 2, to produce a new solution based on the reflection and the visibility of pattern (global search).
3- Use iridophores cells operators in case 3 and 4 to calculate new solutions based on the reflected light coming from best solution and the visibility of matching pattern (local search).
4- Use leucophores cells operator in case 5 to produce new solution by reflecting light from the area around the best solution and visibility of the pattern (local search).
5- Use leucophores cells operator in case 6 for random solution by reflecting incoming light (global search).

3.1. Initialization

First the population *P* (*cells*) of *N* initial solutions *P* = *cells* = {*points*$_1$, *points*$_2$, ..., *points*$_N$}, is spread over *d*-dimensional problem space at random positions (*points*) using (2).

$$P[i].points[j] = random * (upperLimit - lowerLimit) + lowerLimit \qquad (2)$$
$$i = 1, 2, \ldots, N;\ j = 1, 2, \ldots, d$$

Where *upperLimit* and *lowerLimit* are the upper and the lower limits in the problem domain and *random* is a random number between (0, 1).

Each individual *points*$_i$ in of the population represents a single cell and it is associated with two values, fitness and a vector of *d*-dimension continuous values. After that the best solution will be kept in *Best*, and the average of the *Best* points are calculated and stored in AV_{Best}. Then the population is divided into four groups of cells. Each group will work independently sharing only the best solution, two of them (group 1 and 4) are work as a local search, while group 2 and 3 are work as a global search.

3.2 Group 1, Simulation of case 1 and 2

Reflected color (light) shown in Fig2 case (1 and 2) is produced due the interaction between chromatophores and iridophores cells, each chromatophors cell will contract or relax its muscles to stretch or shrink its saccule. While iridophores cells will reflect the light that is coming from chromatophors cells. The reflected light, my penetrates the chromatophors cells or not.

The stretch and shrink process in chromatophors cells and the reflected light from iridophores cells and visibility of the pattern used by cuttlefish to match its background, are used to find a new solution. The formulations of these processes are described in (3) and (4), respectively.

$$reflection_j = R * G_1[i].Points[j] \qquad (3)$$
$$visibility_j = V * (BestPoints[j] - G_1[i].Points[j]) \qquad (4)$$

In (3) and (4), G_1 represents a group of chromatophors cells used to simulates case (1 and 2). *i*, is the *i*th cell of group G_1. *Points*[*j*] represent the *j*th point of *i*th cell. *Best.Points* represents the best solution points. *R*, represents the reflection degree used to find the stretch range of the saccule when the muscles of the cell is in contract or relax. *V*, represents the visibility degree of the final view of the pattern. The value of *R* and *V* are calculated as follows:

$$R = random() * (r_1 - r_2) + r_2 \qquad (5)$$
$$V = random() * (v_1 - v_2) + v_2 \qquad (6)$$

Where, *random*() function is a function used to produce a random numbers between (0, 1). r_1, r_2 are two constant values used to find the stretch interval of the chromatophors cells. While v_1 and v_2 are two constant values used to find the interval of the visibility's degree of the final view of the pattern. Sometime the value of *R* or value of *V* is just set to 1, otherwise it will be calculated. In this group only value of *V* is set to 1 and *R* will be calculated.

To illustrate the two processes, stretching and shrinking in chromatophors cells, and the reflection process in iridophores, consider the value of r_1 and r_2 are set to 2 and -1, respectively, *x*=*Points*[*j*] = 10, *Best* = -5. The value of *R* then found as follows:

$$R = [random() * (2 - (-1))] - 1$$

If *random*() = 0, then *R* will equal to -1. If *random*() =1, then *R* will equal to 2. Since, *x* = 10, then the interval of the stretch and the shrink process in chromatophors cells will be between (-*x* and 2*x*) as showing in Fig3. Based on Fig3, the reflected light from iridophores will be any value between (-10, 20). The reflected values between (0, 20) represents case 1 in Fig2, while the reflected values between (0, -10) represents case 2. Then the *newP* can be found using (1) as follows:

$$newp[j] = reflection_j + visibility_j$$

Since, *Best* = -5, x = 10, then the visibility = -5-10 = -15. Thus the new solution (*newP*) will be any point in the interval (-10, 20) plus (-15). For example: if reflection = 17, the *newP* = 2. In this example, the new solution will be better than the current solution if the value of reflected light is between (-5, 20), otherwise the new solution will be ignored.

Shortly, the search space of the problem is too big, thus these operation reduce the search space to be between two specific values such as in the example between (-10, 20). This group work as a global search uses the value of each point to found new area around best solution with specific interval.

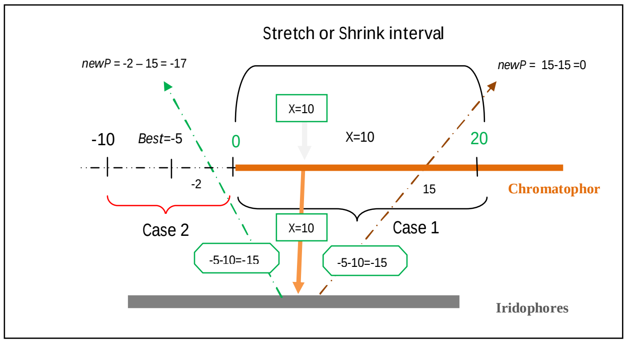

Fig. 3. Reflection and visibility processes, case (1, 2)

3.3. Group 2, Simulation of case 3 and 4

As described before the iridophores cells are light reflecting cells. From Fig2, case (3 and 4), the iridophores cells will reflect incoming light from the outside (environment), and the reflected color is a specific color. Iridophores cells are assisting in concealment or used to conceal organs. We assumed that the concealed organs are represented by the best solution. So the formulation of finding the visibility is remaining as it, while the formulation of finding the reflection is rewritten as follows:

$$reflection_j = R * Best.Point[j] \qquad (7)$$

For this group the value of *R* is set to 1, while the value of *V* will be calculated.

To illustrate consider the same example used with Group 1. *Best* = - 5, *x* = 10, v_1 = 1.5, v_2 = -1.5, and the difference between *Best* and *x* is (*difference* = -5 -10 = -15). The value of *V* is calculated as follows:

$$V = [random() * (1.5 - (-1.5))] - 1.5$$

If *random*() = 0, then *V* will be equal to (-1.5). If *random*() =1, then *V* will be equal to (1.5). Since, *difference* = -15, then the range of the visibility of the current pattern and the best pattern (best solution) will be between (1.5 * *difference*, -1.5 * *difference*) = (-7.5, 7.5), as showing in Fig4. From Fig4, the reflected light (*newP*) from iridophores will be any value between (-7.5, 7.5) plus best solution (*Best*). The reflected value between (-7.5, -5,) represents case 3 in Fig2, while the reflected value between (-5, 7.5) represents case 4. In this example, the new solution will be better than the current solution if the value of reflected light is between (-5, 7.5), otherwise the new solution will be ignored. This group woks as a local search uses the difference between the best solution and the current solution to produce an interval around the best solution as a new search area.

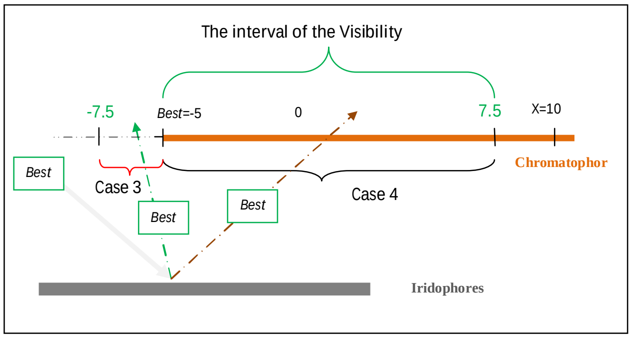

Fig. 4. Reflection and visibility processes, case (3, 4)

3.4. Group 3, Simulation of case 5

Leucophores cells are work as a mirror. In this way, the cells will reflect the predominant wavelength of light in the environment. In white light they will reflect the white, in brown light they will reflect brown and etc., In this case (case 5 in Fig2) the light is coming through chromatophors cells with specific color. The reflected light is very similar to the light that coming from the chromatophors cells. In order to cover the similarity between the incoming color and the reflected color, we assumed that the incoming color is the best solution (*Best*), and the reflected color could be any value around the *Best*. The interval that is used around the *Best* is produced by visibility. The two Equations (3) and (4) of finding the reflection and the visibility are modified as follows:

$$reflection_j = R * Best.Points[j] \qquad (8)$$

$$visibility_j = V * (Best.Points[j] - AV_{Best}) \qquad (9)$$

Where, AV_{Best} is the average value of the *Best* points. The value of *R* is set to 1, while the value of *V* will be calculated.

To illustrate, consider *Best* with two points (9, -3), v_1, v_2 are set to (1,-1), respectively. AV_{Best} = (9-3)/2 = 3. The difference between the *Best* and the AV_{Best} is (*difference*$_1$ = 9-3 =6), (*difference*$_2$ = -3 -3 = -6). Thus the visibility will represents the interval between (-*differnce*$_1$, *differenc*$_2$) = (-6, 6), calculated based on the values of v_1 and v_2. The *newP* could be any value between the interval (*Best.Points*[*j*] +6) and (*Best.Points*[*j*] -6) as showing in Fig5. Also this group works as a local search, but this time uses the difference between the best solution points and the average value of *Best* points to produce a small area around the best solution.

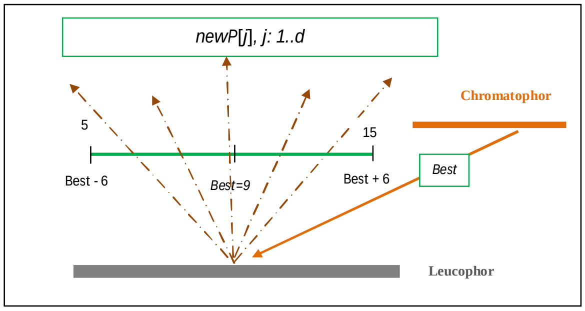

Fig. 5. Reflection and visibility processes, case (5)

3.5. Group 4, Simulation of case 6

In this case, the leucophores cells will just reflect the incoming light from the environment. This operator allows the cuttlefish to blend itself into its environment. As a simulation, one can assume that any incoming color from the environment will be reflected as it and can be represented by any random solution. Thus this case (case 6 in Fig2) works as initialization uses (2) to find the new solutions which is described previously in section 3.1.

Fig6, shows the pseudo code for the CFA in its simplest form. The algorithm is initialized with *N* cells being placed randomly in the search space using (2), and the fitness of the population is evaluated in Step 2, and the best solution is kept in *Best*. In Step 3, the population is divided into four groups. In Step 5, the average of the *Best* points are calculated and stored in AV_{Best}. In Steps 6, 7, 8 and 9, for each cell in each group, a new solution is produced by using the equations (3, 4), (7, 4), (8, 9) and (2), respectively. If current solution is better than the *Best*, then the *Best* is replaced with the new solution. These steps are repeated until the stopping criterion is met which is given in Step 4. Finally, the best solution (*Best*) is returned in Step 11. The general principle of the proposed CFA is shown in Fig7.

```
1.  Initialize population (P[N]) with random solutions. Assign the
    values of $r_1$, $r_2$, $v_1$, $v_2$.
2.  Evaluate fitness of the population. Keep the best solution in Best.
3.  Divide population (cells) into four groups ($G_1$, $G_2$, $G_3$ and $G_4$).
4.  While (stopping criterion is not met)
5.  Calculate the average value of the best solution, and store it in
    $AV_{Best}$.
6.  Case (1, 2)
    For each cell in $G_1$ do
        generate new solution using Equation (3 and 4)
        if( the new solution is better than the Best)
            replace the Best with it.
        if( the new solution is better than the current solution)
            replace the current solution with it.
    end
7.  Case(3, 4)
    For each cell in $G_2$ do
        generate new solution use Equations (7, 4)
        if( the new solution is better than the Best)
            replace the Best with it.
        if( the new solution is better than the current solution)
            replace the current solution with it.
    end

8.  Case(5)
    For each cell in $G_3$ do
        generate new solution use Equations (8, 9)
        if( the new solution is better than the Best)
            replace the Best with it.
        if( the new solution is better than the current solution)
            replace the current solution with it.
    end

9.  Case(6)
    For each cell in $G_4$ do
        generate a random solution using Equations(2)
        if( the new solution is better than the Best)
            replace the Best with it.
        if( the new solution is better than the current solution)
            replace the current solution with it.
    end
10. End While.
11. Return the best solution (Best)
```

Fig. 6. Pseudo-code of the basic Cuttlefish algorithm

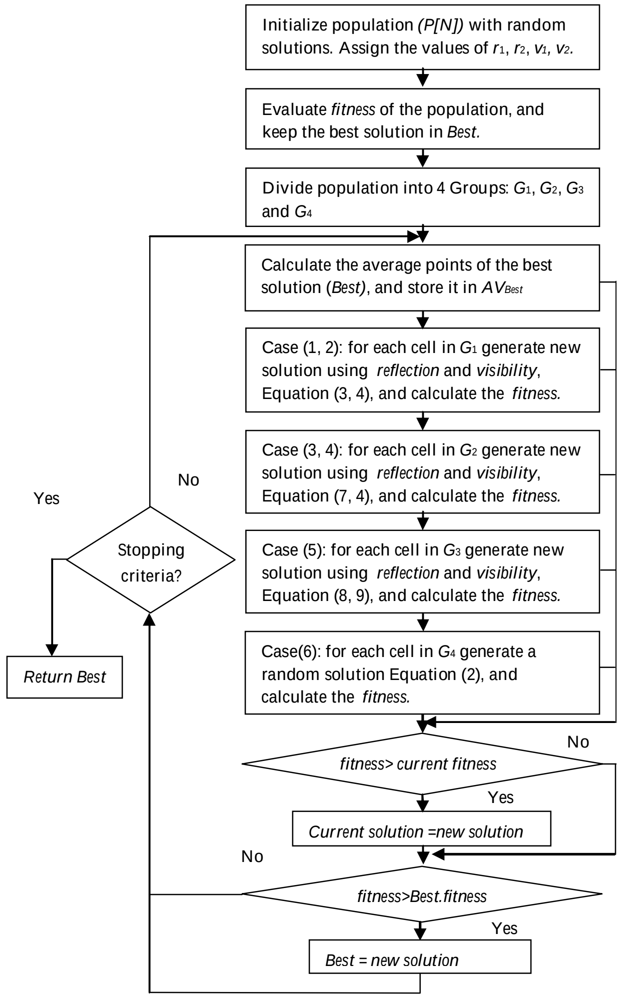
Initialize population (P[N]) with random solutions. Assign the values of r_1, r_2, v_1, v_2.
Evaluate fitness of the population, and keep the best solution in Best.
Divide population into 4 Groups: G_1, G_2, G_3 and G_4
Calculate the average points of the best solution (Best), and store it in AV_{Best}
Case (1, 2): for each cell in G_1 generate new solution using reflection and visibility, Equation (3, 4), and calculate the fitness.
Case (3, 4): for each cell in G_2 generate new solution using reflection and visibility, Equation (7, 4), and calculate the fitness.
Case (5): for each cell in G_3 generate new solution using reflection and visibility, Equation (8, 9), and calculate the fitness.
Case(6): for each cell in G_4 generate a random solution Equation (2), and calculate the fitness.
Yes
No
Stopping criteria?
Return Best
fitness> current fitness
No
Yes
Current solution =new solution
No
fitness>Best.fitness
Yes
Best = new solution

Fig. 7. General principle of CFA

4 Experiments and Validation

To test the performance of the CFA algorithm, Rosenbrock's valley function [22] with 16 dimensions is used. Fig8 shows a two-dimensional view of this function. Rosenbrock's valley is a classical optimization problem which is also known as Banana function or the second function of De Jong. The global optimum lies inside a long, narrow, parabolic shaped flat valley. To find the valley is trivial, however convergence to the global optimum is difficult and hence this problem has been frequently used to test the performance of optimization algorithms. This function has the following definition:

$$f(x)=\sum_{i=1}^{n-1}\left[100(x_{i+1}-x_i^2)^2+(1-x_i)^2\right], \qquad -2.048\le x_i\le 2.048, \qquad i=1,\ldots,n \qquad (10)$$

$$F\_\min(X)=0, \qquad X(1,1,\ldots 1)$$

Fig9 shows how the fitness values evolve with the number of function evaluations. The results are averages for 100 independent runs with population size equal to 60. It can be easily seen that after approximately 250,000 function evaluations, the CFA algorithm is able to find solutions close to the optimum.

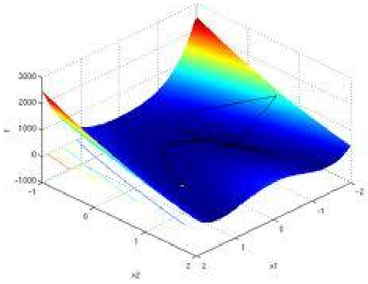

Fig. 8. Rosenbrock's valley function in 2D

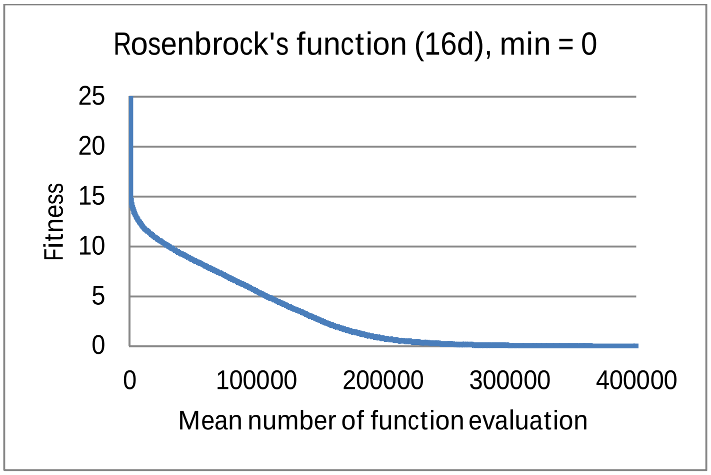

Fig. 19. Evolution of fitness with the mean number of function evaluation, Rosenbrock's fun. with 16d

We have also applied CFA to 12 well known test functions [22] listed in Table 1 in order to compare its performance with other well-known algorithms such as GA, PSO, and BA.

TABLE 1
TEST FUNCTIONS

Function Name	Interval	Function	Global Optimum

1. De Jong	[-5.12, 5.12]	$F\_\min = \sum_{i=1}^{d} x_i^2$	X(0, 0, …, 0)
			F = 0

2. Griewangk	[-600, 600]	$F\_\min = \sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos(\frac{x_i}{\sqrt{i}}) + 1$	X(0, 0, …, 0)
			F = 0

3. Ackley	[-32.768, 32.768]	$F\_\min = -a.\exp(-b.\sqrt{\frac{1}{n}\sum_{i=1}^{d} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{d}\cos(cx_i)) + a + \exp(1)$
		a = 20, b=0.2, c=2π.
			X(0, 0, …,0)
			F = 0

4. Rastrigin	[-5.12, 5.12]	$F\_\min = 10m + \sum_{i=1}^{d}[x_i^2 - 10\cos(2\pi x_i)]$	X(0, 0, …, 0)
			F = 0

5. Axis Parallel hyber-ellipsoid	[-5.12, 5.12]	$F\_\min = \sum_{i=1}^{d}(i * x_i^{\ 2})$	X(0, 0, …, 0)
			F = 0

6. Martin and Gaddy	[0, 10]	$F\_\min = (x_1 - x_2)^2 + [(x_1 + x_2 - 10)/3]^2$	X(5, 5)
			F = 0

7. Rosenbrock's valley	[-2.048, 2.048]	$F\_\min = \sum_{i=1}^{d-1}[100(x_{i+1} - x_i^2)^2]$	X(1, 1)
			F = 0

8. Easom	[-100, 100]	$F\_\min = -\cos(x_i)\cos(x_2)\exp(-(x_1-\pi)^2 - (x_2-\pi)^2)$	X(π, π)
			F = -1

9. Shubert	[-10, 10]	$F\_\min = \sum_{i=1}^{5} i\cos((i+1)x_1 + i)\sum_{i=1}^{5} i\cos((i+1)x_2 + i)$	18 global min. F = -186.7309

10. Schwefel	[-500, 500]	$F\_\min = \sum_{i=1}^{d}[-x_i \sin(\sqrt{\lvert x_i \rvert})]$	X(420.9687, 420.9687)
			F= -418.9829n

11. Goldstein-Price	[-2, -2]	$F\_\min = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2]$
		$*[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2]$
			X(0, -1)
			F=3

12. Shekel's Fox-Holes	[-50, 50]	$F\_\min = \left[\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right]^{-1}$	X(-32, -32)
			F = 1

There are various ways to test the performance of an algorithm with other algorithms [23]. In this paper, we have used three approaches:

- The first approach is to compare the number of function evaluations for a given tolerance or accuracy.
- The second approach is to compare their accuracies for a fixed number of function evaluations.
- The last approach is to compare the mean best fitness solution MBF for a fixed number of iteration. MBF is the average (mean) of the best solutions in last generation for each run.

For genetic algorithms, we have used the real-value GA version [24] with elitism, with the mutation probability equal to 0.05, and the blending crossover [25] methods with the probability equal to 0.95, and roulette wheel selection. For PSO [26], the values of c_1 and c_2 are set to 1.49445 while the inertia factor ω is set to 0.729. For BA [3] and proposed CFA, Table 2 and 3 respectively, describe the parameters values that are used with the different test functions. We run each algorithm for 100 times to make effective comparisons.

For all experiment, the simulations have been carried out using C# on a Pentium Dual-Core CPU 2.20 GHz laptop, 2 GB RAM. Population size for all algorithms in all experiments is fixed to 50.

TABLE 2
BEES ALGORITHM PARAMETERS

function	n	m	e	n to m	n to e	ngh
1. De jong	25	5	2	2	12	0.3
2. Griewangk	28	3	1	5	15	5
3. Ackley	24	4	2	4	11	0.5
4. Rastrigin	23	4	1	5	16	0.1
5. Axis Parallel hyber-ellipsoid	28	3	1	5	15	0.1
6. Martin and Gaddy	25	5	2	2	12	0.1
7. Rosenbrock	25	5	2	2	12	0.1
8. Easom	24	4	2	4	11	1
9. Shubert	24	4	2	4	11	0.5
10. Schwefel	24	4	2	4	11	4
11. Goldstein-Price	28	3	1	5	15	0.1
12. Shekel's Fox-Holes	24	4	2	4	11	1

TABLE 3
CFA ALGORITHM PARAMETERS

No. function	r_1	r_2	v_1	v_2
1. De jong	1	-0.5	1	-1
2. Griewangk	0.4	-0.2	1	-1
3. Ackley	1	-0.5	0.5	-0.5
4. Rastrigin	1	-0.5	0.3	-0.3
5. Axis Parallel hyber-ellipsoid	1	-0.5	1	-1
6. Martin and Gaddy	1	-1	1	-1
7. Rosenbrock	1	-0.5	1.2	-0.2
8. Easom	2	-1	0.5	-0.5
9. Shubert	1	-0.5	0.5	-0.5
10. Schwefel	3	-1	2	-2
11. Goldstein-Price	0.5	-0.2	1	-1
12. Shekel's Fox-Holes	1	-0.5	2	-2

In the first experiment, we used a fixed tolerance ε ≤ 10^{-5} and 1,000,000 function evaluation for each run. The algorithm is stopped when the difference between the obtained minimum fitness and the global optimum is less than or equal to the fixed tolerance, and we run each algorithm for 100 times so that we can do meaningful statistical analysis.

The obtained results shown in Table 4 are averages for 100 independent runs. The form 13094 ±12664.30 (100%) in Table 4 means that the average number (mean) of function evaluation is 13094, standard division is ±12664.30, and the success rate of finding the global optima for this algorithm is 100%. The token (****) means that there is no obtained data with the current algorithm.

From Table 4 we can see clearly that the performance of the proposed CFA is much better than the other algorithm in all cases except function 9 and 12, CFA is perform better in term mean number of function evaluation when compared with the obtained result using PSO, this means that the CFA is faster than PSO . While PSO is perform better in term of standard division.

TABLE 4
COMPARISON RESULTS IN TERM MEAN NUMBER OF FUNCTION EVALUATION, STANDARD DIVISION, AND SUCCESS RATE, (100 RUN, 20,000 ITERATIONS, 1000,000 FUNCTION EVALUATIONS)

Function	GA	PSA	BA	**CFA with**
1. d=120	13094 ±12664.30 (100%)	974121.5 ±147163.88 (3%)	****	**1765.5 ±285.630 (100%)**
2. d=120	14284 ±13501.646 (100%)	****	****	**3903 ±1020.216 (100%)**
3. d=120	14940± 12977.675 (100%)	****	****	**3376.5 ±392.58 (100%)**
4. d=120	13087.5 ±511843.98 (100%)	****	****	**2098 ±402.49 (100%)**
5. d=120	12451.5 ±13476.34 (100%)	****	****	**2295.5 ± 275.51 (100%)**
6. d=2	****	****	50917.5 ±52862.752 (100%)	**382.5 ±116.27 (100%)**
7. d=16	****	732037.5 ± 1105.39 (83%)	****	**636299.5 ±116711.6 (95%)**
8. d=2	****	3098 ±361.657 (100%)	575023.5 ± 368099.464 (72%)	**416.5 ±109.785 (100%)**
9. d=2	****	4113.5 **±580.166** (100%)	****	**1264** ±1986.38 (100%)
10. d=2	****	****	****	****
11. d=2	105868 ±91770.94 (100%)	2381.5 ±395.7054 (100%)	998449 ±15432.255 (1%)	**639.5 ±114.52 (100%)**
12. d=2	973056.5 ±141303.11 (4%)	1816.5 **± 659.6232** (100%)	4094 ±4689.6496 (100%)	**1279** ± 938.99 (100%)

In the second experiment, and to compare the speed of proposed CFA with other algorithm, the number of function evaluations is fixed to 10,000 and the algorithm is stopped when the difference between the obtained minimum fitness and the global optimum is less than 0.001.We run each algorithm for 100 times to make effective comparisons

Table 5 describes the results that are obtained from the experiments. The results are averages for 100 independent runs. The form 743.5(100%) in Table 5 means that the average number (mean) of function evaluation is 743.5 and the success rate of finding the global optima for this algorithm is 100%. The token (****) means that there is no obtained data with the current algorithm.

For the first five test functions 1, 2, 3, 4 and 5 in Table 5, we can see that the GA performs better than both PSO and BA. While PSO, is perform better than both GA and BA for functions 7, 8, 9, 10, 11 and 12. BA performs better than both GA and PSA with success rate 100% using function 6. From Table 5, it is also obvious that the CFA is faster and much superior to other algorithms in terms of accuracy and efficiency for all test functions.

TABLE 5
COMPARISON OF CFA WITH GA, PSA AND BEES ALGORITHM FOR THE SECOND EXPERIMENT IN TERM MEAN NUMBER OF FUCNTION EVALUATION AND SUCCESS RATE, (100 RUN, 200 ITERATION, 10,000 FUNCTION EVALUATIONS)

Function	GA	PSA	BA	**CFA**
1. d=120	6962 (59%)	****	****	**1311 (100%)**
2. d=120	6889.5 (53%)	****	****	**3052 (100%)**
3. d=120	7426.5 (50%)	****	****	**2336.5 (100%)**
4. d=120	6919.5 (58%)	****	****	**2220 (100%)**
5. d=120	7116.5 (53%)	****	****	**1703.5 (100%)**
6. d=2	9901 (1%)	9707.5 (3%)	1448 (100%)	**236 (100%)**
7. d=2	9900.5 (1%)	1407.5 (100%)	7197 (46%)	**968.5 (100%)**
8. d=2	****	2094 (100%)	5868 (72%)	**335.5 (100%)**
9. d=2	****	3046 (100%)	****	**876 (100%)**
10. d=2	****	3622 (86%)	5385 (85%)	**560 (100%)**
11. d-2	5731 (72%)	1465 (100%)	9628.5 (7%)	**446 (100%)**
12. d=2	9999 (1%)	1447 (100%)	2753 (93%)	**893.5 (100%)**

In the last experiment, the number of iterations is fixed to 1000, each run takes 50,000 function evaluations; the algorithm will stop when it finish its iterations. The results over 100 run for this experiment are reported in Table 6, considering the performance of mean best fitness. The best result for each function is boldfaced. According to this table, for function 1 to 6, the proposed CFA is performed better than other algorithms. However, for functions 7, 8 and 9, CFA produces similar results to PSO. For function 10, the results seem to be the same for both BA and CFA. The four algorithms produced the same results for function 11 with small difference among them. While for function 12, the results seem to be the same for CFA, BA, and PSO.

TABLE 6
COMPARISON RESULTS IN TERM MEAN BEST FITNESS, (100 RUN, 1000 ITERATION, 50,000 FUNCTION EVALUATIONS)

Function	GA	PSO	BA	**CFA**
1.De Jong's (d=120)	0.0001	85.357	883.14	**5.662E-193**
2. Griewangk (d=120)	0.0055	296.817	1122.47	**0.0**
3. Ackley (d=120)	0.009	17.382	20.73	**2.114E-15**
4. Rastrigin (d=120)	3.57E-06	814.046	1875.104	**0.0**
5. Axis Parallel hyber-ellipsoid (d=120)	7.814E-05	6195.604	49820.122	**1.782E-185**
6. Martin and Gaddy (d=2)	0.048	0.108	9.9478E-06	**2.314E-31**
7. Rosenbrock (d=2)	0.203	**0.0**	0.00018	8.16E-32
8. Easom (d=2)	-0.125	-1	-0.9998	-1
9. Shubert (d=2)	-183.373	-186.731	-186.65	**-186.730909**
10. Schwefel (d=2)	-770.016	-817.831	-837.965	**-837.965774**
11. Goldstein-Price (d-2)	3.000097	2.99999991	3.005	**2.99999999999991**
12. Shekel's Fox-Holes	2.683	0.9999998	0.99999984	0.99999984

5 CONCLUSIONS

In recent years, various meta-heuristic inspired optimization methods have been developed. In this paper, a new meta-heuristic optimization algorithm called Cuttlefish Algorithm (CFA) is introduced. The algorithm is inspired based on the color changing behavior of cuttlefish to find the optimal solution. The patterns and colors seen in cuttlefish are produced by reflected light from different layers of cells including (chromatophores, leucophores and iridophores) stacked together, and it is the combination of certain cells at once that allows cuttlefish to possess such a large array of patterns and colors. In this paper, the simulation of light reflecting and visibility of the matching patterns processes used by these cells layers are formulated. The results obtained by the proposed CFA in all cases provide superior results when compared with GA, PSO, and BA. As a future work, more study on CFA parameters is needed.

REFERENCES

[1] A. Jorge, Ruiz-Vanoye, D. –P. Ocotlan, C. Felipe, S. Andres, Ma. De los Angeles, V. – R Gustavo, A. –L. Roberto, *Meta-Heuristics Algorithms based on the Grouping of Animals by Social Behavior for the Traveling Salesman Problem*, International Journal of Combinatorial Optimization Problems and Informatics, Vol. 3, No. 3, 2012.

[2] Dorigo, Marco, *Ant colony optimization*, Massachusetts Institute of Technology, 2004.

[3] D.T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri , S. Rahim , M. Zaidi, *The Bees Algorithm – A Novel Tool for Complex Optimization*, Manufacturing Engineering Centre, Cardiff University, 2005.

[4] J. Kennedy, and R. Eberhart, Particle *Swarm Optimization*, IEEE International Conference on Neural Networks, 1995.

[5] C. Erik, G. Mauricio, Z. Daniel, P. –C. Marco, and G. Guillermo, *An Algorithm for Global Optimization Inspired by Collective Animal Behavior*, Hindawi Publishing Corporation Discrete Dynamics in Nature and Society, 2012.

[6] R. Esmat, N. –P. Hossein, S. Saeid, *GSA: A Gravitational Search Algorithm, Elsevier*, Information Sciences, 2009.

[7] M. Yannis, M. Magdalene, and M. Nikolaos, *A Bumble Bees Mating Optimization Algorithm for Global Unconstrained Optimization Problems*, NICSO, SCI 284, 2010.

[8] B. Ali, *A new Approach to Global Optimization Motivated by Parliamentary Political Competitions*, ICIC International, 2008.

[9] Y. Xin-She, *A New Metaheuristic Bat-Inspired Algorithm*, Springer, 2010.

[10] M. Kalyani, C. S. Suresh, B. Poornasatyanarayana, *Population based meta-heuristic techniques for solving optimization problems: A selective survey*, international journal of Emerging Technology and Advanced Engineering IJETAE, Vol. 2 Issue 11, 2012.

[11] Y. Xin-She, *Firefly algorithms for multimodal optimization*, in: Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences, Vol. 5792, 2009.

[12] M. M. Lydia, J. D. Eric, and T. H. Roger, N. M. Justin, *Mechanisms and behavioural functions of structural coloration in cephalopods*, J. R. Soc. Interface, 2008.

[13] http://www.thecephalopodpage.org/.

[14] Y. Jarred, C. L. Alexandra, G. Allyson, H. J. St. H. Debra, T. Lindsay, M. Michelle and J. T. Nathan, *Principles underlying chromatophore addition during maturation in the European cuttlefish*, Sepia officinalis, Experimental Biology 214, 3423-3432, 2011.

[15] R. T. Hanlon, and J. B. Messenger, *Cephalopod Behavior*, Cambridge: Cambridge University Press, 1996.

[16] E. Florey, *Ultrastructure and function of cephalopod chromatophores*, Am. Zool. 1969.

[17] R. T. Hanlon, K. M. Cooper, B. U. Budelmann. and T. C. Pappas, *Physiological color change in squid iridophores I, Behavior, morphology and pharmacology in Lolliguncula brevis*, Cell and Tissue Research. 259, 1990.

[18] K. M. Cooper, R. T. Hanlon, and B.U. Budelmann, *Physiological color change in squid iridophores II, Ultrastructural mechanisms in Lolliguncula brevis,* Cell and Tissue Research. 259, 1990.

[19] R. A. Cloney, and S. L. Brocco, *Chromatophore organs, reflector cells, iridocytes and leucophores in cephalopods*, Am. Zool. 1983.

[20] D. Froesch,. and J. B. Messenger, *On leucophores and the chromatic unit of Octopus vulgaris*, J. Zool, 1978.

[21] T. Wilson, and J. W. Hastings, *Bioluminescence*. Annu. Rev. Cell Devel. Biol. 14:197-230, 1998.

[22] M. Marcin, S. Czesław. *Test functions for optimization needs*, 2005.

[23] Y. Xinjie, G. Mitsuo, *Introduction to Evolutionary Algorithms*, Springer-Verlag London Limited, 2010.

[24] L. H. Randy, and E. H. Sue, *Practical Genetic Algorithms Second Edition*, John Wiley & Sons, ISBN: 978-0-471-45565-3, Inc, 2004.

[25] J. R. Nicholas, *Forma analysis and random respectful recombination*, In Proc. 4th Int. Conf. on Genetic Algorithms, San Mateo, CA: Morgan Kauffman,1991.

[26] R. C. Eberhart, Y. Shi, *Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization*, Evolutionary Computation, 2000, Proceedings of the 2000 Congress, Vol. 1, IEEE, 2000.

[27] K. Eric, M. M. Lydia, T. H. Roger, B. D. Patrick, R. N. Rajesh, F. Eric and H. Jason, *Biological versus electronic adaptive coloration: how can one inform the other*, J. R. Soc. Interface, 2012.

[28] C. Blum, and A. Roli, *Metaheuristics in combinatorial optimization: Overview and conceptual comparison*, ACM Comput. Surv., 35, 268-308. 2003.

[29] K. Dervis, and B. Bahriye, *Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems*, Proceedings of the 12th international Fuzzy Systems Association world congress on Foundations of Fuzzy Logic and Soft Computing, Springer, Verlag Berlin, Heidelberg, 2007.

- *Adel Sabry Eesa is currently pursuing Ph.D degree program in computer science in Zakho University, Duhok City, KRG-Iraq. E-mail: adelsabryissa@gmail.com*
- *Adnan Mohsin Abdulazeez Brifcani, Department of IT, Duhok Technical Institute, Duhok Polytechnic University, Duhok City, KRG-Iraq. Email: adnan_brifcani@yahoo.com*
- *Zeynep Orman, Department of Computer Engineering, Faculty of Engineering, Istanbul University, 34320, Avcilar, Istanbul, Turkey. ormanz@istanbul.edu.tr*