



A novel optimization algorithm (Lion-AYAD) to find optimal DNA protein synthesis



Samaher Al-Janabi ^{b,*}, Ayad Alkaim ^a

^a Department of Chemistry, Faculty of Science for Women (SCIW), University of Babylon, Babylon, Iraq

^b Department of Computer Science, Faculty of Science for Women (SCIW), University of Babylon, Babylon, Iraq

ARTICLE INFO

Article history:

Received 25 May 2018

Revised 8 February 2020

Accepted 22 January 2022

Available online 12 February 2022

Keywords:

Intelligent bioinformatic analysis

Optimization

Synthesis

Lion-AYAD

DNA

PSO

COA

WOA

LOA

Spirally searching and Bubble net searching mechanism

ABSTRACT

In this paper, we present a new algorithm to find the optimal proteins generated through DNA synthesis. The algorithm executes in five stages: in the first stage, it takes a DNA sequences and consider it as the initial populations of lions, determined the main positions of each lion and the main distances among lions and goal point then consider this distance as fitness of that lions, after that sort the lions based on their fitness to preparing it to the second stage. The second stage develops lion optimization algorithm (LOA) by adding four new features on it, each feature performance one task, a replacing the kernel of LOA (i.e., searching machnizam) by spirally searching & Bubble net searching to increase the accuracy, at the same time reduce the execution time to reach of the goal achieve by A Smart feature. The main purpose of the third stage is determining lion active or more yauld where each lion in population need update the positions and fitness after each move in searching space to reach of their goal, this achieved through Yauld feature. The fourth stage applies the Cooperative features to convert the active sequence of DNA (i.e., Yauld lion) into mRNA after that built tRNA from it after splitting it into triplet to start to generate the proteins. Synthesis of all triplet of tRNA to generated final proteins result by new optimization algorithm achieved based on deep composite that satisfies the four rules, this feature called Deep feature and represent the final stage of the algorithm. The new algorithm appears as a pragmatic optimization model, it proves their robust to work with dynamic length of DNA sequence. It increases accuracy and reduces execution times.

© 2022 THE AUTHORS. Published by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

DNA replication is semi-conservative in nature. DNA is comprised of a double-stranded helical base of nucleotides. This double strand “unzips” at the beginning of the replication process, separating into two single strands, which can be copied. Each of the two strands is then replicated, resulting in two new daughter strands of DNA from the parent helix, while conserving the original

base sequence of the parent strand. (a) **Continuous DNA Replication**, and (b) **Discontinuous DNA Replication**.

Optimisation is an overall term used in many areas, intelligent computation in different types also in different sciences to reach of the best values (i.e., These values take multi names such as minimum, maximum, less cost, high performance, etc.) Based on the nature of the problem and what the researchers need to prove from this problem (i.e., Based on the objective function/functions of the problem). Sometimes the field of used determined the specific definition of term optimization.

In this paper, we will focus on optimization of the side algorithms by enhancing the search machnizam and add new capabilities into one of the optimization algorithms that begin search on optimal values based on specific strategy apply on initial population and in each step attempts to find the best path and values allowed it to near from their first sub goal, Then Second etc. to reach of the final goal (i.e., search sequence of sub goals to reach of the final solution of problem). The main properties should con-

* Corresponding author at: Department of Computer Science, Faculty of Science for Women (SCIW), University of Babylon, Babylon, Iraq.

E-mail addresses: samaher@itnet.uobabylon.edu.iq (S. Al-Janabi), alkaim@iftc.uni-hannover.de (A. Alkaim).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.



Production and hosting by Elsevier

tain in any optimization algorithm are robustness; efficiency and accuracy.

Synthesis have multi definitions based on the purpose of it such as: "(a) the composition or combination of parts or elements so as to form a whole. (b) the production of a substance by the union of chemical elements, groups, or simpler compounds or by the degradation of a complex compound. (c) The combining of often diverse conceptions into a coherent whole also: the complex so formed". While another definition "(a) deductive reasoning. (b) The dialectic combination of thesis and antithesis into a higher stage of truth" As a result, we can define synthesis as the procedure opposite analysis procedure, where Synthesis refer to combine separate elements or components in order to form a coherent whole

Lion Optimization Algorithm is one of the optimization algorithm derivatives from the natural community behaviours and cooperative among lions such as prey capturing, mating, territorial marking, defense and the other behaviors. We can summarization it's performance by find the best solution. Therefore, this paper will focus on develop that algorithm by add new prosperities on it to find the optimal proteins from DNA Sequences. **Table 1** shown compare among the main optimization techniques

2. Related works

Optimization is the act of getting the finest result beneath given circumstances. The optimization divided into (maximization minimization, single variable multi variables, with without constrains). There are two types of objective functions optimization, single objective function and multiple objective function. Optimization has three basic elements which are decision variables, objective, and constraints. An optimization algorithm is distinguished a procedure which is executed iteratively by comparing various solutions till an optimum or a satisfactory solution is found, there are many types of optimization namely continuous optimization, bound constrained optimization, constrained optimization, derivative-free optimization, discrete optimization, global optimization, linear programming and nondifferentiable optimization. Multi researchers work in the field optimization.

Alkaim and Al_Janabi [14] propose a new prediction method focused on the development of MARS data mining by replacing the kernel with a multiobjectives optimization feature. To find a parallel gas flaring rate based on three main gases (CO₂, CH₄, N₂O) that improves the efficiency and decreases the time spent finding the optimum result. In this analysis, too, measure the oil rate (GOR) of gas. Zawbaa et al. [1] suggested a binary ALO for the collection of the functions. The findings show that binary ALO will outperform traditional PSO and GA in a substantial way. Zawbaa et al. [2] have enhanced ALO's usefulness in the treatment of function selection tasks by introducing a new ALO (CALO) based on confusion. In Lab VIEW TM Gupta et al. [3] built a new toolkit based on ALO. Yamany et al. [4] employed ALO to refine MLP's link vectors. They used only four datasets and the findings show that ALO will outperform ACO, PSO, and GA.

Rajan and Malakar [5] suggested ALO (MALO) for the treatment of Optimum Reactive Control Dispatch (ORPD) tasks in power systems and the results confirm the MALO's robustness and accuracy. Al_Janabi el. al. [13] build model called Multi Objectives Renewable Energy Generation (MORE-G) based on constructions and multi objective optimization methods. Through design two models of optimization; first called Develop Cuckoo Optimization Model (DCOM) based on enhance the performance of Cuckoo search optimization by apply the horizontal combination on it, While the second Develop the CapsNet Model (DCapsNet) based on using multi layers network and each layer apply different activation function then combination results through linear combination to satisfy the optimization concept.

Kamboj et al. [6] have used ALO to solve a nonconvex ORPD problem in an efficient way. Ali et al. [7] used basic ALO to handle effective distributed generation allocation and sizing (DG). The comparative findings showed ALO's superiority over other meta-heuristics, such as Artificial Bee Colony (ABC), Firefly Algorithm (FA), Cuckoo Quest (CS), Ant Colony Optimization (ACO), GA and PSO. Even the basic ALO was used to handle the combined planning and scheduling tasks [8]. Raju et al. [9] developed an ALObased method for controller optimisation at the same time. Dubey et al. [10] applied simple ALO to arranging hydrothermal-wind (HTPGS) functions.

Table 1

Compare among the Optimization Techniques.

Techniques of Optimization	Advantages	Disadvantages
Discrete Optimization	<ul style="list-style-type: none"> Give efficient solutions. 	<ul style="list-style-type: none"> Generate a series of persistent sub-problems. A complicated way. Give illogical results.
Unconstraint Optimization	<ul style="list-style-type: none"> Helps solve the most common problems. 	<ul style="list-style-type: none"> The classification is based on the information available on the function to be optimized.
Constraint Optimization	<ul style="list-style-type: none"> Works on applications with clear restrictions. 	<ul style="list-style-type: none"> The method is more complicated than Unconstrained Optimization.
None, One or Multi objectives	<ul style="list-style-type: none"> You find a solution that fully satisfies your problem. 	<ul style="list-style-type: none"> Some goals are replaced by restrictions.
Deterministic Optimization	<ul style="list-style-type: none"> Number of steps to find the goal limited and known Most time led to near from local optimization 	<ul style="list-style-type: none"> All the variables related and not related to the goal take into account when design the optimization model, therefore, this led to more computations and complexity
Stochastic Optimization	<ul style="list-style-type: none"> The cost is expected lowest. High cost very Limited estimation to appear. Need to the previous information to discovery it. In real time the information can adapts. Can work with the total or group or part of information 	<ul style="list-style-type: none"> Very difficult compute the probabilities. Require compute the probability of all scenarios. Not suitable all the time

The findings demonstrate ALO's strengths and benefits in obtaining better-quality technologies while meeting a lot of real-world hydro, thermal, and wind generation constraints. Mirjalili et al. [11] suggested the multiobjective version of ALO (MOALO), and the findings show the high utility of this approach in real-world engineering solutions.

Al-Janabi and Alkaim [12] present a novel tool called DRFLLS to fine the optimal values to missing values. By developing random forest algorithm, seven categories of similarity measures were defined. These categories are person similarity coefficient, simple similarity, and fuzzy similarity (M1, M2, M3, M4 and M5). Al-Janabi and Madi [15] present study based on intelligent computation through deep analysis to five prediction data mining techniques to determine which one achieve an optimal biomedical analysis, this study appear not the mathematical computation always lead to optimal solution as popular but the nature of data is consider the main parameter in determined the optimal way.

3. Main optimization algorithms

This section focuses on the discovery and analysis the behaviours of the main optimization algorithms to compare the behaviours and performance of new algorithm Lion-Ayad with them later.

3.1. Particle swarm optimization (PSO)

Particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality; for more detail see [17]. In general, can summarize the main steps of PSO in algorithm 1

Algorithm #1: PSO

```

Input: X, P           //X: problem size, P: population size
Output: Y           //Y : Pgbest: the best solution of population
1.   A: Population of agents,  $p_i$ : Position of agent  $a_i$  in the solution space, f: Objective function
2.    $v_i$  : Velocity of agent's  $a_i$ ,  $V(a_i)$ : Neighborhood of agent  $a_i$  (fixed)
3.    $[x^*]$  = PSO()
4.   P = Particle_Initialization()
5.   For  $i=1$  to  $it\_max$ 
6.     For each particle  $p$  in  $P$  do
7.        $f_p = f(p)$ 
8.       IF  $f_p$  is better than  $f(pBest)$ 
9.          $pBest = p$ 
10.      End IF
11.    End for
12.     $gBest = best\ p\ in\ P$ 
13.    For each particle  $p$  in  $P$  do
14.       $v = v + c1 * r1 * (pBest - p) + c2 * r2 * (gBest - p)$ 
15.       $p = p + v$ 
16.    End for
17.  End for

End PSO

```

of solve any problem. Mahdi and Al-Janabi [22] present a study for the main challenges in the healthcare field. The goal is to identify patients who will be admitted to the hospital within the next year by using historical claims data including Information about patient and analysis it to solve the problem. Through build the predictor that satisfies the three points: "Speed execution", "abilities to deal with very huge databases" and "predict the number of days the patients spend in the hospital in the next years".

here: c_1 : acceleration factor related to pbest, c_2 : acceleration factor related to gbest, r_1 : random number between 0 and 1, r_2 : random number between 0 and 1.

3.2. Cuckoos optimization algorithm (COA)

Cuckoos are a bird family with a special breeding technique which is more aggressive relative to the species of other birds.

Cuckoo is an algorithm inspired by nature, built on the basis of cuckoo bird reproduction. See [13,18] for more information, also Algorithm 2

its principal problem. With the aim of increasing global convergence speed and achieving better performance, the WOA optimization process will introduce chaos. In the proposed chaotic WOA methods, numerous chaotic maps are considered for tuning WOA

Algorithm #2: COA

```

Input: X, P           //X: problem size, P: population size
Output: Y          //Y : Pgbest: the best solution of population
1: Determined the constraints for each dataset.
2: Set of parameters: t: Number of Iteration, n: Number of nests host,
   Fi: fitness function
3: Determined the objective function f(x) , x=(x1,x2.....xd)T
4: Generate initial population of n host nests xi (i=1,2.....n)
5: While (t< Max Generation) or (stop criteria)
6:   Get a cuckoo randomly (say, i) and replace its solution by
      Perform Lévy flights
7:   Evaluate its fitness Fi
8:   Choose a nest among n (say, j) randomly
9:   | IF Fi < Fj then
10:    |   Replace j by the new solution
11:   | End if
12:   a fraction pa of the worse nests is abandoned and new ones
      are built
13:   Keep the best nests
14:   Rank the nests and find the current best
15:   Pass the current best solutions to the next generation
16: End while

```

Cuckoo optimization algorithm is very good for problems of global optimization, as it preserves a balance between local random walk and global random walk. The equilibrium between local and global random walks is regulated by a parameter $p_a \in [0,1]$. The random local and global walks are characterized by the

Eq. (1) and (2), respectively [18].

$$X_i^{t+1} = X_i^t + \alpha S H(p_a - \varepsilon) (X_j^t - X_k^t) \quad (1)$$

$$X_i^{t+1} = X_i^t + \alpha L(S, \lambda) \quad (2)$$

's main parameter which helps to monitor exploration and exploitation. In the recent past, the implementations of the algorithm have been widely used in various field because of its optimum output and quality [19,20]. The definition of each parameters shown under Table 6.

$$D = |C * X_{k_{rand}} - X(t)| \quad (3)$$

$$X(t+1) = X_{k_{rand}} - A * D \quad (4)$$

$$X(t+1) = D * \exp(b * l) * \cos(2 * \pi * l) + \dot{X}(t) \quad (5)$$

3.3. Whale optimization algorithm (WOA)

The whale optimization algorithm (WOA) is a proposed bio-inspired metaheuristic algorithm which focuses on the collective hunting behavior of the humpback whales. As with other metaheuristic algorithms, WOA is facing slow convergence speed as

3.4. Lion optimization algorithm (LOA)

Lion Optimization Algorithm (LOA) is one of the main optimization algorithms described in algorithm 4 also LOA have multi parameters [16] and very different from Ant lion optimization (ALO) in strategies of works and parameters [21]; Chaser new

Algorithm #3: WOA

```

Input: X, P           //X: problem size, P: population size
Output: Y           //Y : Pgbest: the best solution of population
Initialize a, A, C, l and p
1: Calculate the fitness of each search agent
2: X*= the best search agent
3: While (it < Maxiter)
4:   For each search agent
5:     IF (p < 0.5)
6:       IF (|A| < 1)
7:         Update the position of the current search agent by the
           equation (1)
8:       Else if (|A| = 1)
9:         Select a random search agent (X_rand)
10:        Update the position of the current search agent by the
            equation (3)
11:      End if
12:      Else if (p = 0.5)
13:        Update the position of the current search by the by the equation (2)
14:      End if
15:    End for
16:    Calculate the fitness of each search agent
17:    Update X* if there is a better solution
18:    it=it+1
19:    Update a, A, C, l and p
20: End While
21: Return X*

```

and Chaser are the new and existing positions of each chaser lion, respectively; rand represents a random number between zero and one; and Prey indicates the position of the prey. Winger i; new indicates the new position of Winger hunter lion, and H2 is a random unit vector perpendicular to vector W, Cheater new represents the new position of the Winger hunter lion, and Cheater is the present position of the Winger hunter lion. Male Lion new indi-

cates the new position of the male lion, and Territory. Best Positions represent the collective understanding of the pride's members about the search space. The cumulative best positions of the resident lions (including both males and females) are sorted from the best to the worst according to the corresponding fitness; E is an excursion constant; T R indicates the territory ratio. For more details see [16].

Algorithm #4 : LOA**Input:** X, P // X : problem size, P : population size**Output:** Y // Y : P_{best} : the best solution of population

Initialize the total population of lions and lionesses

1: Calculate the fitness of each search agent

2: $X^* =$ the best search agent3: **While** ($it < Maxiter$)4: **For** each search agent

5: Sort all females according to their fitness

6: Group the lionesses into three general groups

7: From each group select one member randomly to become as lion of that group

8: Update the female lion position as following (6) to (12)

$$\text{Chaser}_{new} = \text{Chaser} + H_1 * \text{rand} + (D) * (2 \text{rand} - 1) \quad (6)$$

$$H_1 = (\text{Prey} - \text{Chaser}) \quad (7)$$

$$\text{Winger}_{new} = \text{Prey} + H_2 * \text{abs}(W) * \text{rand} + (D) * (2 \text{rand} - 1); \quad (8)$$

$$W = \text{Prey} - \text{Winger} \quad (9)$$

$$|W| = \sqrt{W_1^2 + \dots + W_n^2}. \quad (10)$$

$$\text{Cheater}_{new} = \text{Prey} + H_3 * \text{rand} + (D) * (2 \text{rand} - 1) \quad (11)$$

$$H_3 = (\text{Prey} - \text{Cheater}) \quad (12)$$

9: Great territory Matrix

$$\text{Territory}_i = \text{Best Positions} * \text{rand} * TR \quad (13)$$

$$\text{Best Positions} = \text{sort}[\text{Male best Positions}; \text{Female best Positions}] \quad (14)$$

10: Update the position of Lions

$$\text{Male Lion}_{new} = \text{Territory} + E * (D) * (2 \text{rand} - 1) \quad (15)$$

11: **End for**

12: Calculate the fitness of each search agent

13: Update X^* if there is a better solution14: $it = it + 1$ 15: **End While**16: Return X^* **4. Lion-AYAD optimization algorithm (Lion-AYAD)**

The main points of the research methodology can represent by the following steps, Algorithm 5 summarized all steps required to generate proteins from the DNA sequences synthesis, we refer to that new optimization algorithm by name Lion-AYAD, the main idea of that algorithm generated from the add new properties to one of the optimization algorithms that prove good performance, Fig. 3 presents the main activities of research work, while Fig. 4 shown the architecture of Lion-AYAD algorithm. Fig. 5 shown the Contribution of each triple concentrations in generation the 20 types of Proteins.

A. Main parameters of new optimizer determined in that step includes number of Population; Max number of Iteration; Current Lion-AYAD. position; initial Lion-AYAD. fitness; area that any lion searching it represent by low boundary and upper boundary.

B. Split the DNA sequences through preprocessing step, for make the algorithm more dynamic, we deal with short and long sequence. At the same times each sequence can have different number of start and end points.

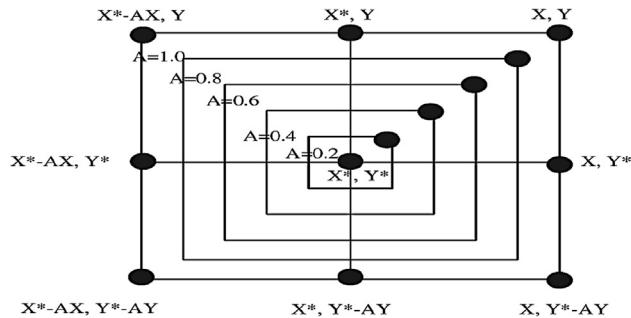


Fig. 1. Bubble net searching mechanism to update the position.

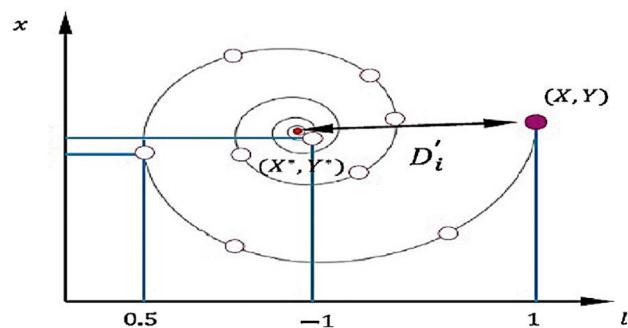


Fig. 2. Spiral. Searching Mechanism to update the position.

C. Build the Optimizer through generated the initial populations of lions, determined the main positions of each lion and the main distances among lions and goal then consider this distance as fitness of that lions, sort the lions based on their fitness.

D. Develop the Lion Optimization Algorithm (LOA) by replacing the kernel of LOA in searching machnizam by using a smart feature (i.e., spirally searching & Bubble net searching) to increase the accuracy, at the same time reduce the execution time to reach of the goal.

E. Each lion based on step D will move from the recent position forward new sub goal as attempting to reach of the final goal, therefore, all lions need to update the positions and fitness of it. The main purpose of this step is determining lion active or more yauld, this step achieved through yauld feature.

F. To start by generated the proteins, we can't work on the main DNA sequence therefore in A cooperative step, we will first generated mRNA molecules through; Build-mRNA-for-each-gene Then Build-protein-from- mRNA. Finally, Build-tRNA-for- protein after split it into triplet.

G. Apply four rules through synthesis each triplet tRNA to generated proteins by Lion-AYAD optimization Algorithm this achieved final feature add by that algorithm that depend on deep composite that satisfy the rules.

- Rules for all triplets have the 2nd position is **U generated set of Proteins**.
- Rules for all triplets have the 2nd position is **C generated set of Proteins**.
- Rules for all triplets have the 3rd position is **A generated set of Proteins**.
- Rules for all triplets have the 4th position is **G generated set of Proteins**.

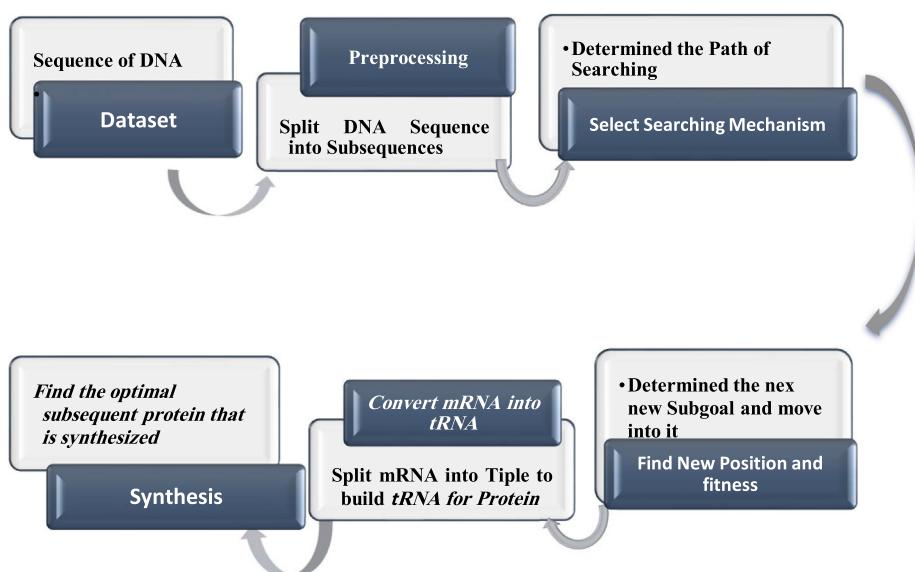
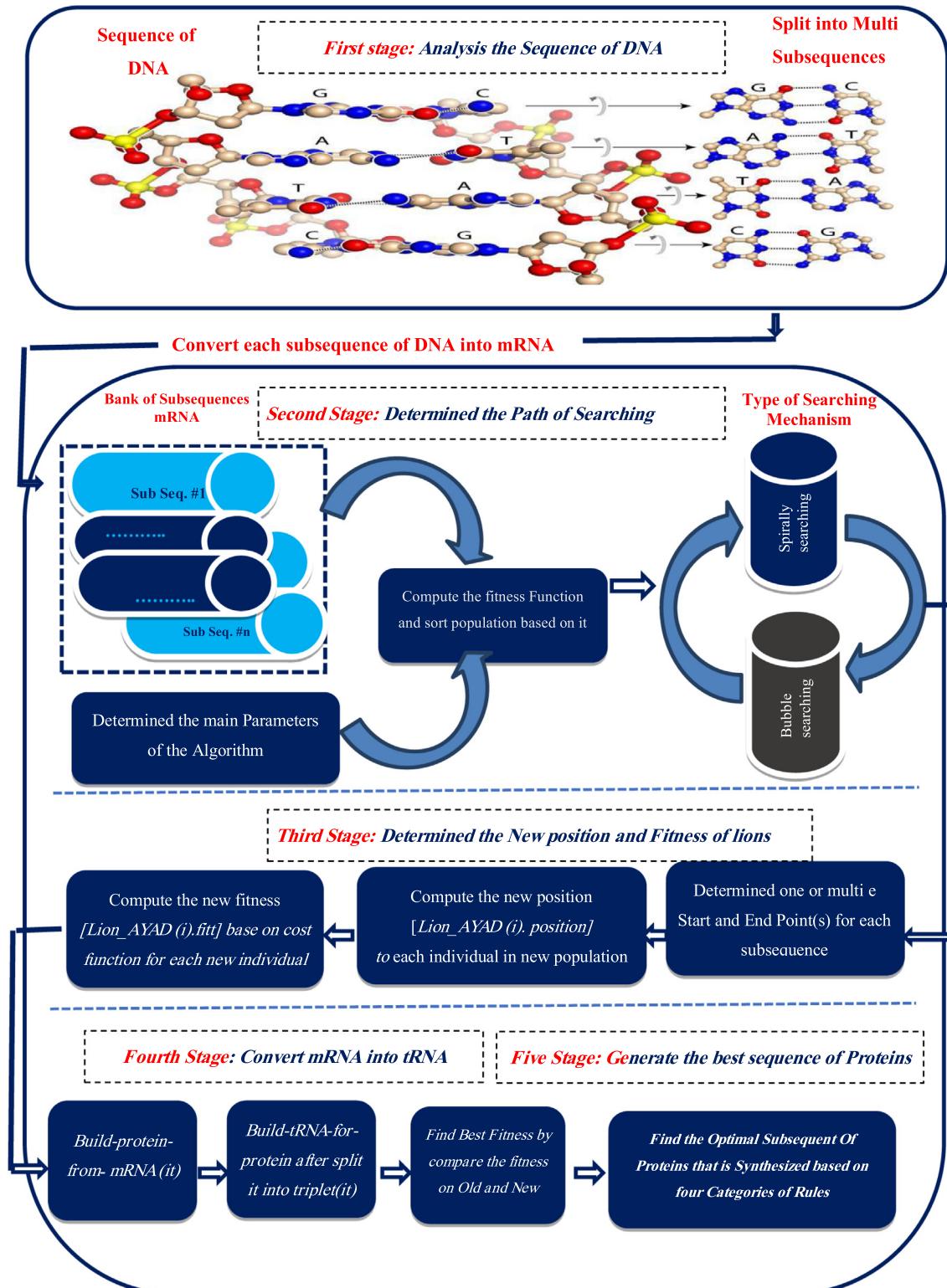


Fig. 3. Main Activities to A Novel Optimization Algorithm.

**Fig. 4.** The Architecture of Lion-AYAD.

Algorithm #5: Lion-AYAD

Input: Long or Short DNA Sequence contain four terms (*A, T, C, G*)
Output: Set of Optimal Protein Synthesis
Initialization: number of Population(*nPop*)=100; Max number of Iteration (*MaxIt*)=500; Empty_lion-AYAD.position = [] ; Empty_Lion-AYAD.fitf = [] ; Lion-AYAD=repmat(Empty_lion-AYAD, *nPop*, 1), lb: low boundary, ub: upper boundary
// Pre-Processing Stage
1: For each DNA sequence in GenBank
2: Call split DNA sequence (start-position end-position)
3: End For
// Build Optimizer (Lion-AYAD)
4: For each lion-AYAD(*i*) in the population (*nPop*)
5: Lion-AYAD (*i*).position = Call split DNA sequence (lb, ub, 1, length of sequence); // Determine initial Position
6: Lion-AYAD (*i*).fitf=costfunction(Lion-AYAD (*i*).position); // Determine initial fitness
7: IF Lion-AYAD (*i*).fitf < bestsol.fitf
8: bestsol.position = Lion-AYAD (*i*).position;
9: bestsol.fitf = Lion-AYAD (*i*).fitf;
10: End If
11: End For
// A Smart Feature: determined the searching mechanism.
12: For each iteration(*i*)to MaxIt
13: For each lion-AYAD(*i*) in the population (*nPop*)
14: For each Subsequence(*j*)in DNA //J length of sub sequence
15: Select *r1*, *r2* and *p* randomly in rang (0,1)
16: Compute *A* = 2^*a^*r1-a ; *c* = 2^*p2 ;
17: IF *p*<0.5 // First mechanism
18: If abs(*A*)<=1
19: *D*=abs(*c**bestsol.position(*j*)- Lion-AYAD(*i*).position(*j*));
20: Lion-AYAD(*i*).position(*j*)=bestsol.position(*j*)-*A***D*;
21: Else
22: *K*=randperm(*nPop*);
23: *K*(*K*=*i*)=[];
24: *k*=*K*(1);
25: *D*=abs(*c**Lion-AYAD(*k*).position(*j*)- Lion-AYAD(*i*).position(*j*));
26: Lion-AYAD(*i*).position(*j*)=Lion-AYAD (*k*).position(*j*)-*A***D*;
27: End If
28: Else // Second mechanism
29: *b*=*I*; *I*=unifrnd(-1,1);
30: *D*=abs(bestsol.position(*j*)- Lion-AYAD(*i*).position(*j*));
31: Lion-AYAD(*i*).position(*j*)= $D^*exp(b^*i^*cos(2^*\pi i^*)+bestsol.position(j))$;
32: End If
33: End For
// Yaud Feature: determine bounded for most effect DNA subsequence (Start & End Points)
34: Lion_AYAD (*i*).position=max(Lion_AYAD (*i*).position,lb);
35: Lion_AYAD (*i*).position=min(Lion_AYAD (*i*).position,ub);
36: Lion_AYAD (*i*).fitf=costfunction(Lion_AYAD (*i*).position);
37: End For
// A Cooperative Feature: Generated mRNA molecules
38: Build-mRNA-for-each-gene(*it*)
39: Build-protein-from-mRNA(*it*)
40: Build-tRNA-for protein after split it into triplet(*it*)
41: *a*=2-it*(2)/MaxIt // update a base on MaxIt
42: Lion_AYAD (best)=min([Lion_AYAD.fitf]); // Sort and find best lion
43: IF Lion_AYAD (best).fitf<bestsol.fitf
44: bestsol=Lion_AYAD (best);
45: End If
46: bestcost(it)=bestsol.fitf;
// Deep Feature: find the optimal subsequent protein that is synthesized.
47: For all triplets have the 2nd position is U // Convert the sequence of mRNA into Protein
48: ["UUU" "Phe"] ["UUC" "Phe"] ["UUA" "Leu"] ["UUG" "Leu"]
49: ["CUU" "Leu"] ["CUC" "Leu"] ["CUA" "Leu"] ["CUG" "Leu"]
50: ["AUU" "Ile"] ["AUC" "Ile"] ["AUU" "Ile"] ["AUG" "Met"]
51: ["GUU" "Val"] ["GUC" "Val"] ["GUA" "Val"] ["GUG" "Val"]
52: End For
53: For all triplets have the 2nd position is C //Convert the sequence of mRNA into Protein
54: ["UCU" "Ser"] ["UCC" "Ser"] ["UCA" "Ser"] ["UCG" "Ser"]
55: ["CCU" "Pro"] ["CCC" "Pro"] ["CCA" "Pro"] ["CCG" "Pro"]
56: ["ACU" "Thr"] ["ACC" "Thr"] ["ACA" "Thr"] ["ACG" "Thr"]
57: ["GCC" "Ala"] ["GCC" "Ala"] ["GCA" "Ala"] ["GCG" "Ala"]
58: End For
59: For all triplets have the 3rd position is A //Convert the sequence of mRNA into Protein
60: ["UAU" "Tyr"] ["UAC" "Tyr"] ["UAA" "Stop"] ["UAG" "Stop"]
61: ["CAU" "His"] ["CAC" "His"] ["CAA" "Gln"] ["CAG" "Gln"]
62: ["AAU" "Asn"] ["AAC" "Asn"] ["AAA" "Lys"] ["AAG" "Lys"]
63: ["GAU" "Asp"] ["GAC" "Asp"] ["GAA" "Glu"] ["GAG" "Glu"]
64: End For
65: For all triplets have the 4th position is G //Convert the sequence of mRNA into Protein
66: ["UGU" "Cys"] ["UGC" "Cys"] ["UGA" "Stop"] ["UGG" "Trp"]
67: ["CGU" "Arg"] ["CGC" "Arg"] ["CGA" "Arg"] ["CGG" "Arg"]
68: ["AGU" "Ser"] ["AGC" "Ser"] ["AGA" "Arg"] ["AGG" "Arg"]
69: ["GGU" "Gly"] ["GGC" "Gly"] ["GGA" "Gly"] ["GGG" "Gly"]
70: End For
End Lion_AYAD

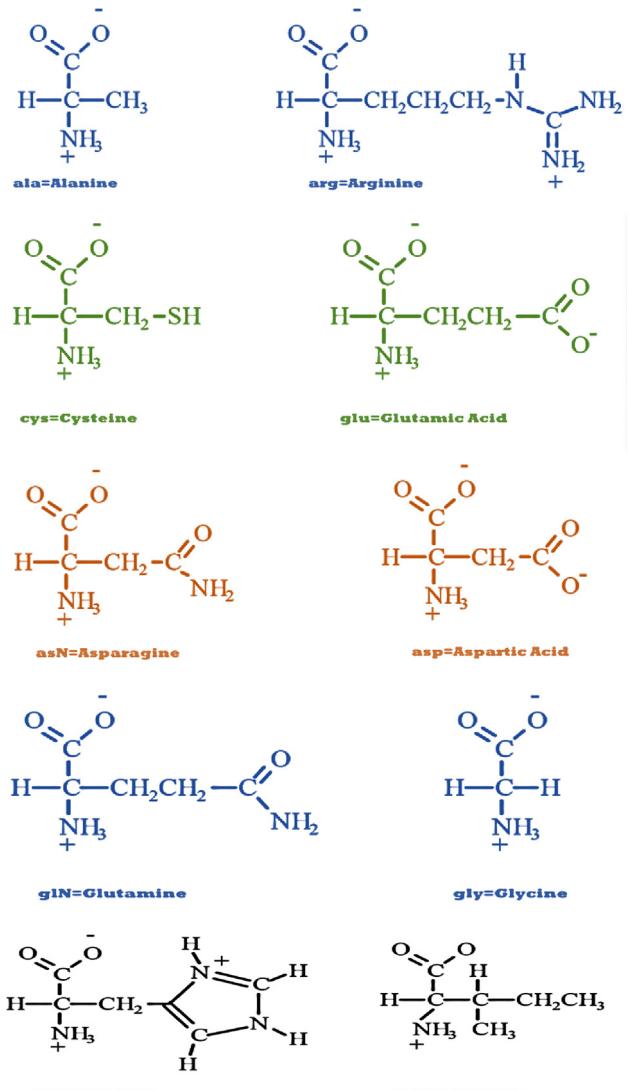


Fig. 5. Contribution of each triple concentrations in generation the 20 types of Proteins.

5. Results

To test the performance and behaviours of Lion-AYAD algorithm, we will take multi case studies different in the length of DNA sequence also show the types of proteins generated from each DNA sequence synthesis, the details of each case study will explain below in Fig. 6 and Fig. 7.

A. First Case Study: Implementation Lion-AYAD on Short Sequence of DNA

- Proteins Produced from original DNA = 1: Protein: -Met-Trp

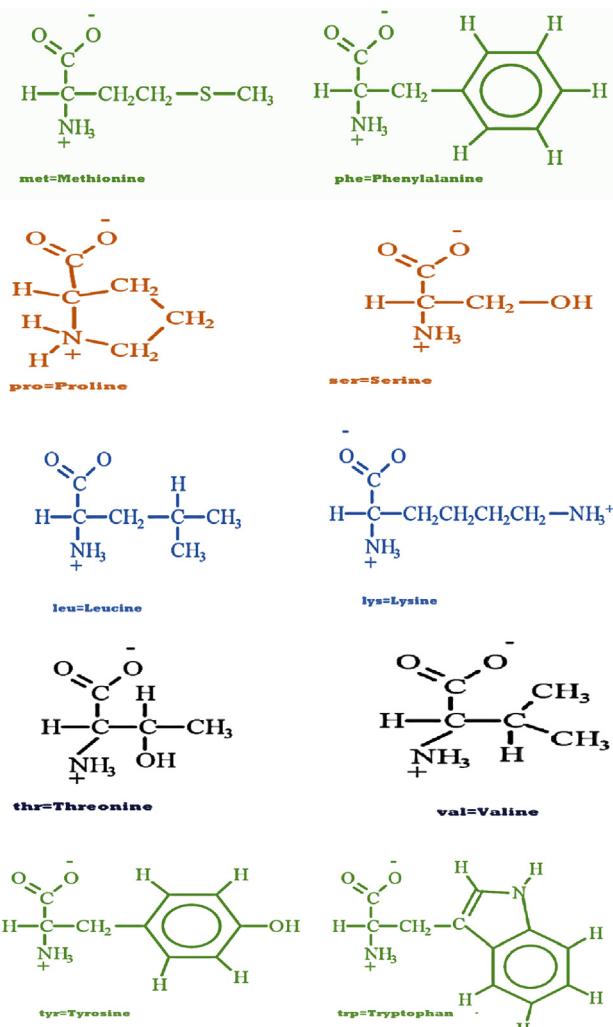


Fig. 5 (continued)

- Proteins Produced from copy of DNA = 1: Protein: -Met-Trp

B. Second Case Study: Implementation Lion-AYAD on Sequence have one Cut Point

In this case study, we take the following an active sequence that have one start point represent in blue color is ATG and one end point represent in red color is TAG while the sequence is [GGGATG-GACACCTTATCATTGCTACTAGCGACCAGTTGAG-TAGCTTCGTCGGTGA]. This sequence contain **39** item and generated sequence of proteins contain **13** different types of proteins as shown in [Table 2](#) through apply the main steps of lion-AYAD algorithm on the active sequence of DNA as represent in [Fig. 8](#) (a, b, c, d, e, f, and g).

C. Third Case Study: Implementation Lion-AYAD on Sequence of DNA have Two Cut Points

In this case study, we get active a sequence has two start points represent in blue color are ATG and two end points represent in red color are TAG and TAA while a sequence is [AGT_{ATG}AAAACCCAC-GATGGTAGCCCGAGATTGAGATGTGGTTTTCAGATAACAGATGTG-GAGCAGCTTTAG]. This active sequence contain **57** items while through apply Lion-AYAD algorithm on it generated two sequences of proteins, first contain **6** different types of proteins, second contain **5** different types of proteins as represent in [Table 3](#) and [Fig. 9](#) (a and b).

D. Fourth Case Study: Implementation Lion-AYAD Algorithm on Active Sequence of DNA have Three Cut Points

In this case study, we get active a sequence has three start points represent in blue color are ATG and three end points represent in red color are TAG and TAA while a sequence is [GAT_{ATG}TGGTAGCCCGAGATGTGGTTTTCAGATAACAGATGTG-GAGCAGCTTTAG]. This active sequence contain **57** items while through apply Lion-AYAD algorithm on it generated three sequences of proteins, [Table 4](#) described that proteins that distributed as (2 types in first group, 5 in second group while 5 different types of proteins in third group). [Fig. 10](#) (a and b) show all details.

A. First Case Study: Implementation Lion-AYAD on Short Sequence of DNA

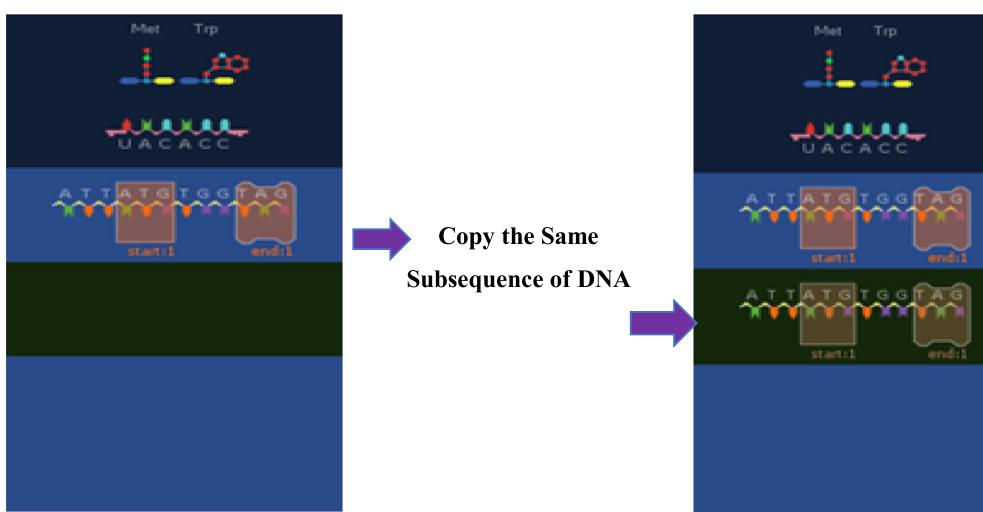


Fig. 6. Generate the proteins from short sequence [ATTATGTGGTAG].

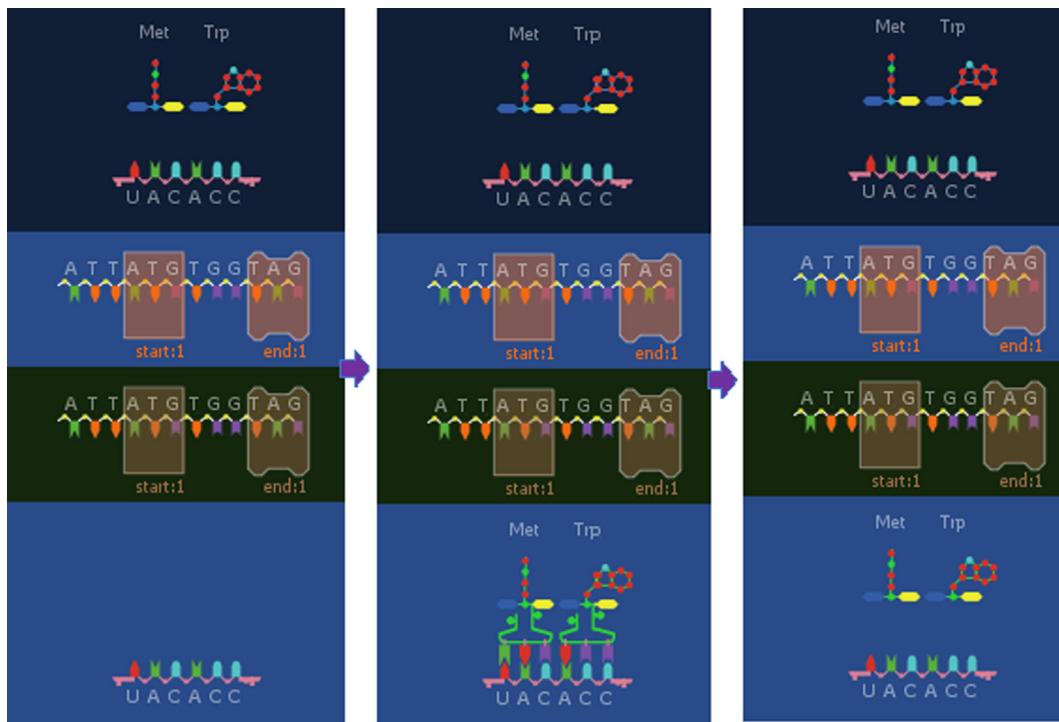


Fig. 7. Short sequence generated two proteins Met and Trp.

Table 2

Proteins generated base on Second Case Study.

Proteins Produced from original DNA = 1	Proteins Produced from copy of DNA = 1
Protein gen #1 -Met-Asp-Thr-Leu-Ser-Phe-Ala-Thr-Ser-Asp-Gln-Phe-Glu	Protein gen #1 -Met-Asp-Thr-Leu-Ser-Phe-Ala-Thr-Ser-Asp-Gln-Phe-Glu

E. Fifth Case Study: Implementation Lion-AYAD Algorithm on Active Sequence of DNA have Eight Cut Points

In this case study, we get active a sequence has eight start and end points; a sequence is [GGGATGGACACCTTAT CATTGCTAC TAGCGACCAGTTGAGTAGCTCGTCGGTGAATTGGCACACAT GACTGGGAATGACCCATTGACCCAGACGTGGTTTGAGTGGTGT TATGAGGGGGACCCGAGATGTGGTTTGAAATAGACAAGTAGACCC TAATAGACAGTATGAAAACCCACGAGTGGTAGCCCCGAGATTGA GATGTGGTTTCAGATAACAG]. This active sequence contain **297** items while through apply Lion-AYAD algorithm on it generated eight sequences of proteins, these eight sequences contain **99** proteins distributed among **8 groups** especially (**13,25,5,26,13,5,6, and 5**) different types of proteins as explain in Table 5. While all steps shown in Fig. 11 (a, and b).

6. Discussion & conclusion

This paper present multi points can be summarization as; A novel approach for generate proteins using DNA Sequence Synthesis. Implies enhanced searching mechanism in the lion optimization algorithm. Incorporating four new features called “AYAD” with Lion optimization algorithm lead to increase statistical performance. The best results in term of generate proteins includes only calling behavior of four new features. Optimal lion-AYAD find best behavior to generate all truth proteins from DNA sequence.

Before summarization the main features and characteristic of that algorithm will analysis traditional optimization algorithms based on the four points (i.e., advantages, disadvantages, primary parameters and Secondary parameters) compare with the algorithm called Lion-AYAD present in that work, in general, optimization tries to solve a variety of problems by producing many technical works to find optimal or reasonable solutions for a specific problem. In this section, the major properties of five optimization algorithms have been considered and a comparison among them is shown in Table 6.

After that, we analyzed the main parameters of each optimization algorithm with the novel algorithm (lion-AYAD), where each one of the optimization algorithms has a number of parameters that can be primary factors in building the optimization model or

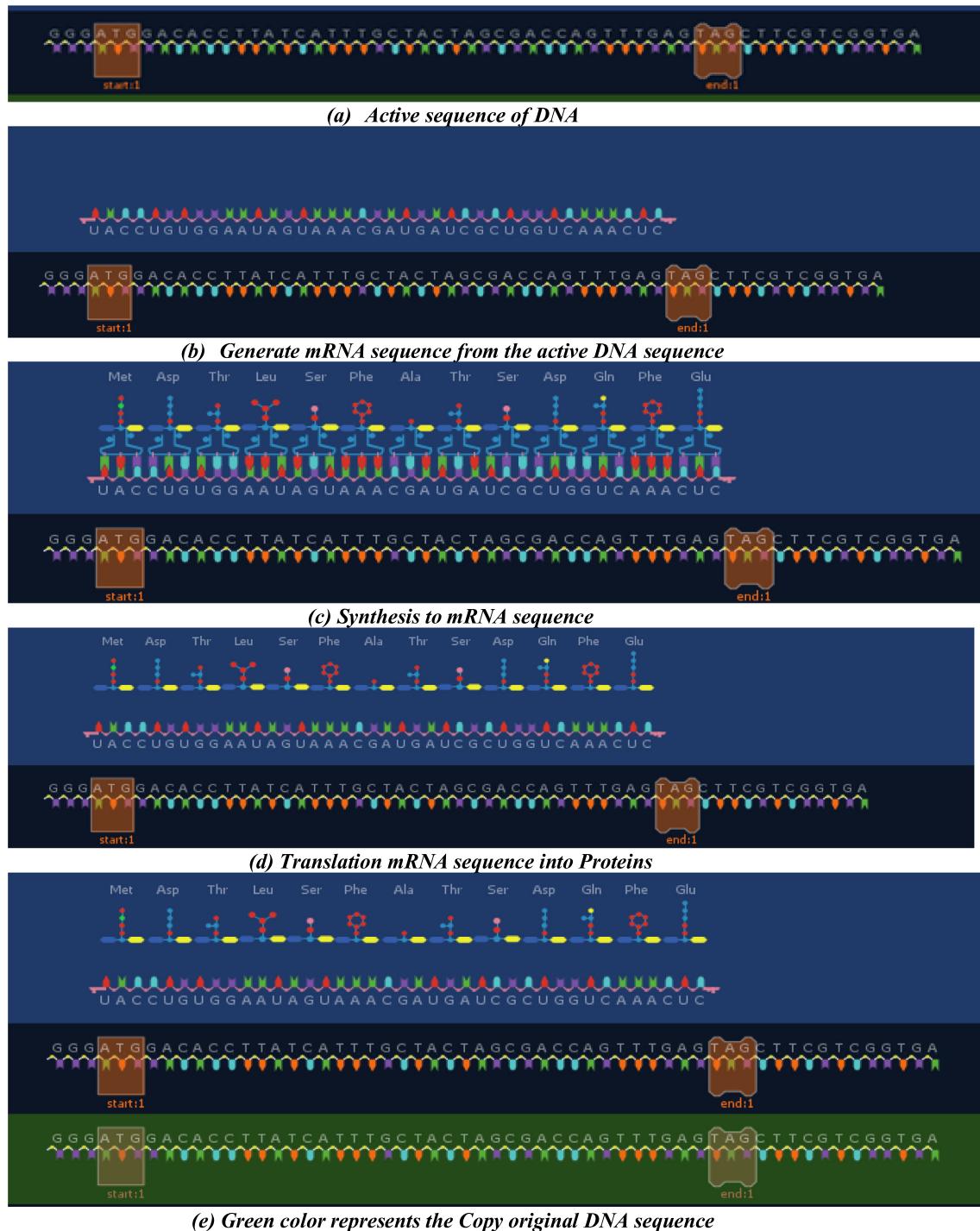


Fig. 8. Main Steps of Proteins generated through Lion-AYAD Algorithm.

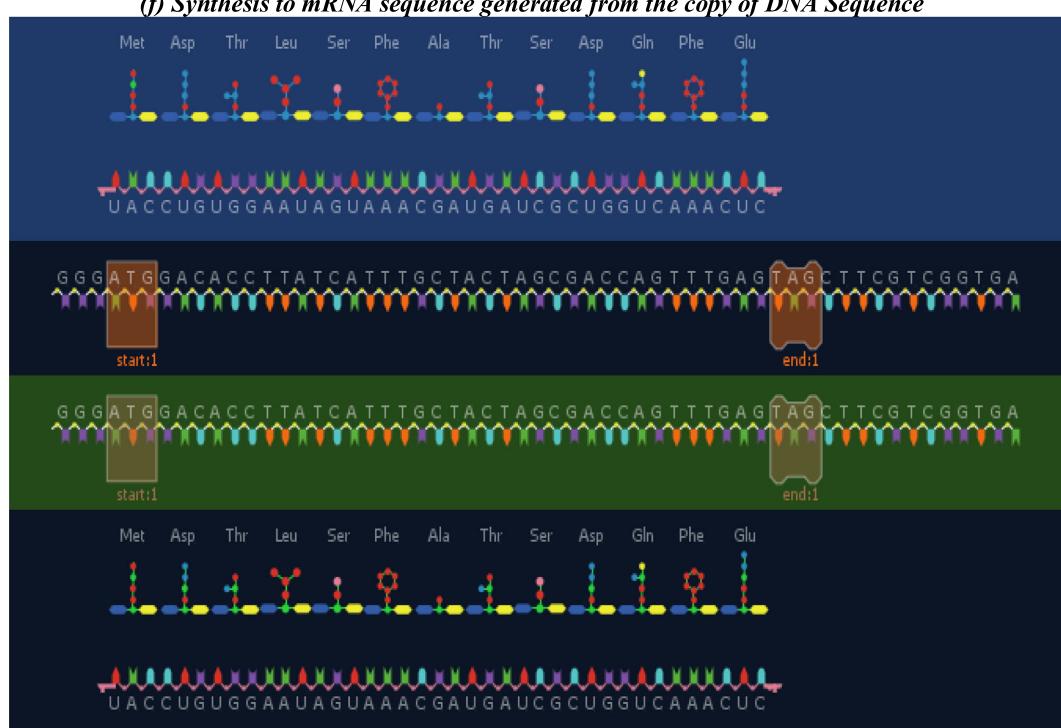
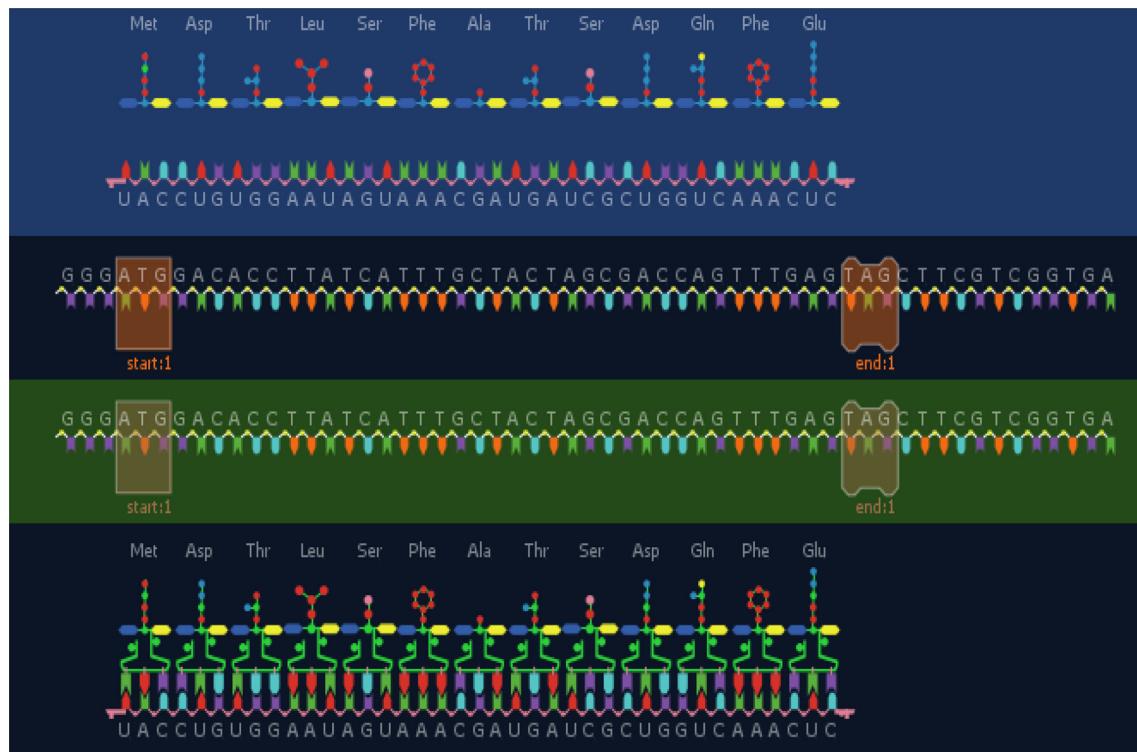


Fig. 8 (continued)

Table 3

Proteins generated base on Third Case Study.

Proteins Produced from original DNA = 2	Proteins Produced from copy of DNA = 2
Protein gen #1 -Met-Lys-Thr-His-Glu-Trp	Protein gen #1 -Met-Lys-Thr-His-Glu-Trp
Protein gen #2 -Met-Trp-Phe-Phe-Arg	Protein gen #2 -Met-Trp-Phe-Phe-Arg

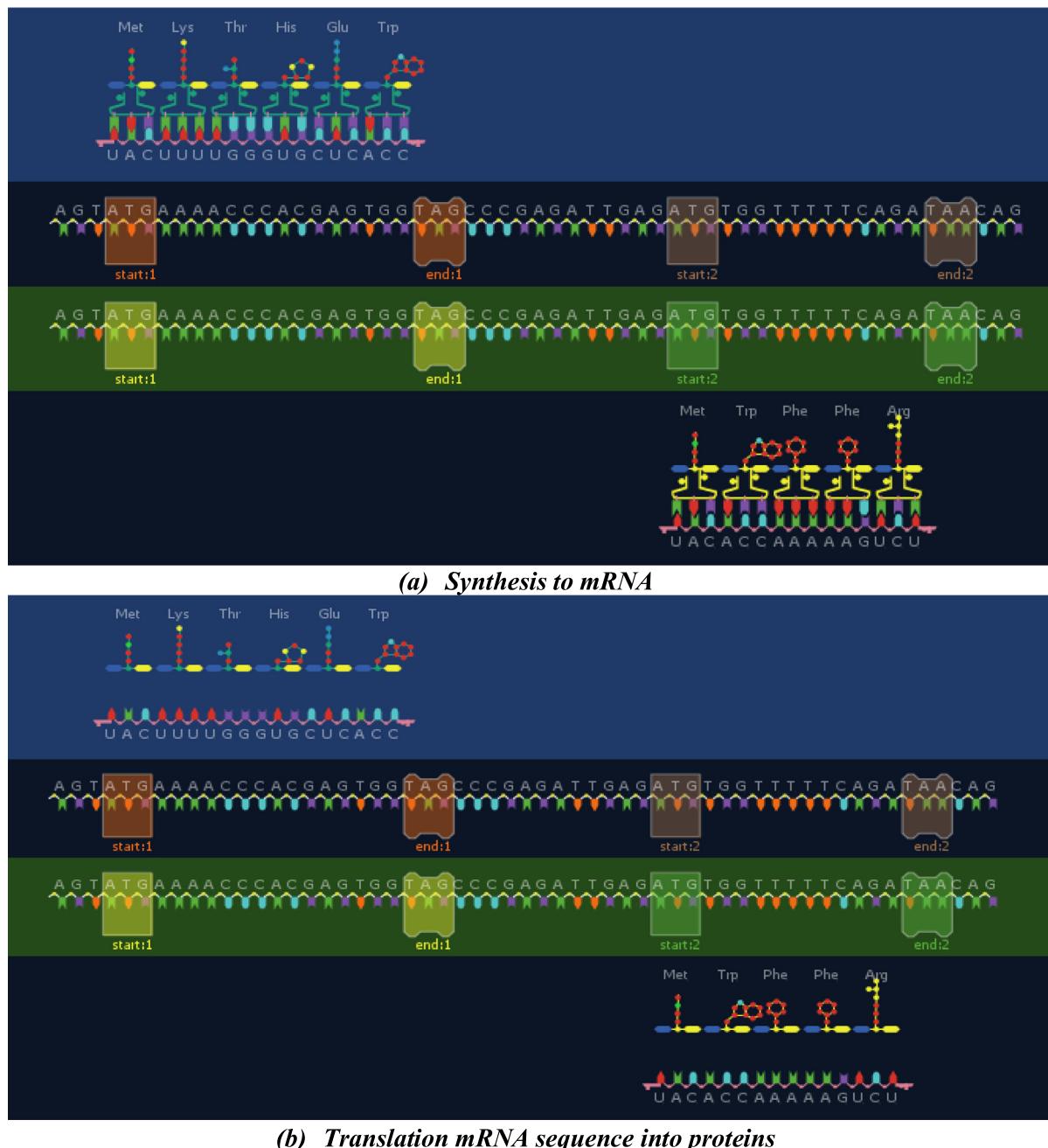
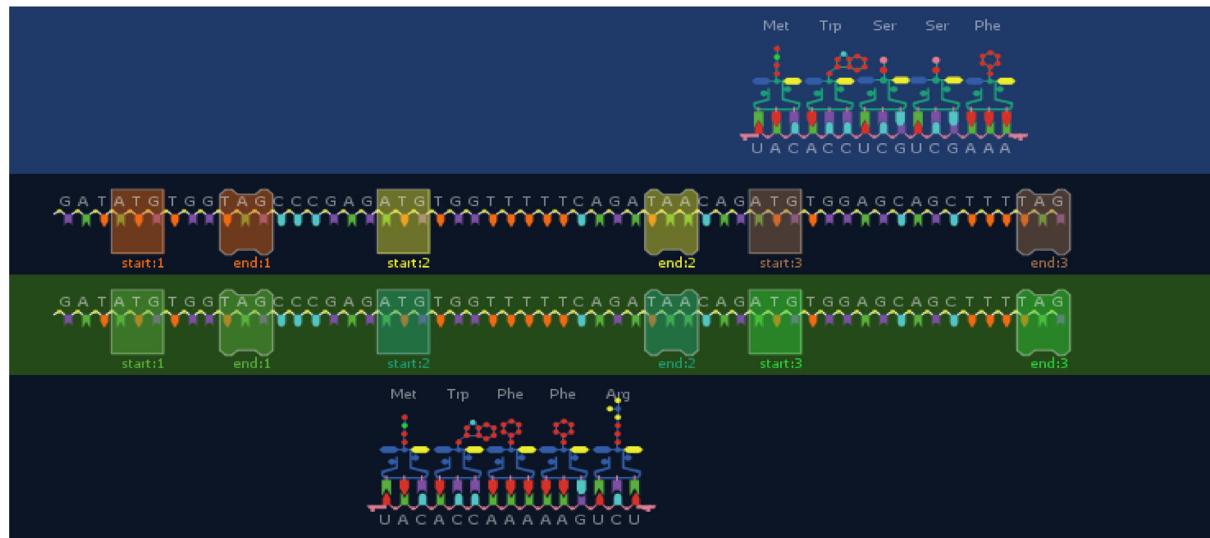


Fig. 9. Proteins generated by lion-AYAD from Active Sequence of DNA have two cut points.

Table 4

Proteins generated base on Fourth Case Study.

Proteins Produced from original DNA = 3	Proteins Produced from copy of DNA = 3
Protein gen #1: -Met-Trp	Protein gen #1: -Met-Trp
Protein gen #2: -Met-Trp-Phe-Phe-Arg	Protein gen #2: -Met-Trp-Phe-Phe-Arg
Protein gen #3: -Met-Trp-Ser-Ser-Phe	Protein gen #3: -Met-Trp-Ser-Ser-Phe



(a) Synthesis to mRNA Sequence generated from the Active Sequence of DNA have Three Cut Points



(b) Translation mRNA into Portions

Fig. 10. Proteins generated by Lion-AYAD from Active Sequence of DNA has Three Cut Points.

Table 5a

Proteins generated base on Fifth Case Study.

Proteins Produced from original DNA = 8	Proteins Produced from copy of DNA = 8
Protein gen #1: -Met-Asp-Thr-Leu-Ser-Phe-Ala-Thr-Ser-Asp-Gln-Phe-Glu	Protein gen #1: -Met-Asp-Thr-Leu-Ser-Phe-Ala-Thr-Ser-Asp-Gln-Phe-Glu
Protein gen #2: -Met-Thr-Gly-Asn-Asp-Pro-Ile-Asp-Pro-Asp-Val-Val-Phe-Glu-Trp-Cys-Tyr-Glu-Gly-Asp-Pro-Arg-Cys-Gly-Phe	Protein gen #2: -Met-Thr-Gly-Asn-Asp-Pro-Ile-Asp-Pro-Asp-Val-Val-Phe-Glu-Trp-Cys-Tyr-Glu-Gly-Asp-Pro-Arg-Cys-Gly-Phe
Protein gen #3: -Met-Trp-Phe-Phe-Arg	Protein gen #3: -Met-Trp-Phe-Phe-Arg
Protein gen #4: -Met-Thr-Pro-Leu-Thr-Gln-Thr-Trp-Phe-Leu-Ser-Gly-Val-Met-Arg-Gly-Thr-Arg-Asp-Val-Val-Phe-Glu-Ile-Asp-Lys	Protein gen #4: -Met-Thr-Pro-Leu-Thr-Gln-Thr-Trp-Phe-Leu-Ser-Gly-Val-Met-Arg-Gly-Thr-Arg-Asp-Val-Val-Phe-Glu-Ile-Asp-Lys
Protein gen #5: -Met-Arg-Gly-Thr-Arg-Asp-Val-Val-Phe-Glu-Ile-Asp-Lys	Protein gen #5: -Met-Arg-Gly-Thr-Arg-Asp-Val-Val-Phe-Glu-Ile-Asp-Lys
Protein gen #6: -Met-Trp-Phe-Leu-Lys	Protein gen #6: -Met-Trp-Phe-Leu-Lys
Protein gen #7: -Met-Lys-Thr-His-Glu-Trp	Protein gen #7: -Met-Lys-Thr-His-Glu-Trp
Protein gen #8: -Met-Trp-Phe-Phe-Arg	Protein gen #8: -Met-Trp-Phe-Phe-Arg

Table 5b

Proteins generated base on Fifth Case Study.

Proteins Produced from original DNA = 8	Proteins Produced from copy of DNA = 8
Protein gen #1: -Met-Asp-Thr-Leu-Ser-Phe-Ala-Thr-Ser-Asp-Gln-Phe-Glu	Protein gen #1: -Met-Asp-Thr-Leu-Ser-Phe-Ala-Thr-Ser-Asp-Gln-Phe-Glu
Protein gen #2: -Met-Thr-Gly-Asn-Asp-Pro-Ile-Asp-Pro-Asp-Val-Val-Phe-Glu-Trp-Cys-Tyr-Glu-Gly-Asp-Pro-Arg-Cys-Gly-Phe	Protein gen #2: -Met-Thr-Gly-Asn-Asp-Pro-Ile-Asp-Pro-Asp-Val-Val-Phe-Glu-Trp-Cys-Tyr-Glu-Gly-Asp-Pro-Arg-Cys-Gly-Phe
Protein gen #3: -Met-Trp-Phe-Phe-Arg	Protein gen #3: -Met-Trp-Phe-Phe-Arg
Protein gen #4: -Met-Thr-Pro-Leu-Thr-Gln-Thr-Trp-Phe-Leu-Ser-Gly-Val-Met-Arg-Gly-Thr-Arg-Asp-Val-Phe-Glu-Ile-Asp-Lys	Protein gen #4: -Met-Thr-Pro-Leu-Thr-Gln-Thr-Trp-Phe-Leu-Ser-Gly-Val-Met-Arg-Gly-Thr-Arg-Asp-Val-Phe-Glu-Ile-Asp-Lys
Protein gen #5: -Met-Arg-Gly-Thr-Arg-Asp-Val-Val-Phe-Glu-Ile-Asp-Lys	Protein gen #5: -Met-Arg-Gly-Thr-Arg-Asp-Val-Val-Phe-Glu-Ile-Asp-Lys
Protein gen #6: -Met-Trp-Phe-Leu-Lys	Protein gen #6: -Met-Trp-Phe-Leu-Lys
Protein gen #7: -Met-Lys-Thr-His-Glu-Trp	Protein gen #7: -Met-Lys-Thr-His-Glu-Trp
Protein gen #8: -Met-Trp-Phe-Phe-Arg	Protein gen #8: -Met-Trp-Phe-Phe-Arg

Table 6

Comparison among Main optimization algorithms with Lion-AYAD.

Algorithm	Advantages	Disadvantages	Primary parameters	Secondary Parameters
PSO	<ul style="list-style-type: none"> Can be simple to implement Have few parameters to adjust Able to run parallel computation Can be robust Have higher probability and efficiency in finding the global optima Can converge fast Do not overlap and mutate Have short computational time Can be efficient for solving problems presenting difficulty to find accurate mathematical models 	<ul style="list-style-type: none"> Can be difficult to define initial design parameters Cannot work out the problems of scattering Can converge prematurely and be trapped into a local minimum especially with complex problems 	X,P,Y	c1,c2,r1,r2
COA	<ul style="list-style-type: none"> It satisfies the global convergence requirements. It supports local and global search capabilities . It uses Lévy flights as a global search strategy. 	<ul style="list-style-type: none"> Algorithm is still undergoing rapid development and improvement, and the processing of some links still needs continuous optimization 	X,Y,F,N	T
WOA	<ul style="list-style-type: none"> has the ability to avoid local optima and get a globally optimal solution appropriate for solving different constrained or unconstrained optimization problems for practical applications 	<ul style="list-style-type: none"> not good at exploring the search space. 	X , Y , C , A , D,F , X _{rand}	t , n , p , a , r , l , b
LOA	<ul style="list-style-type: none"> It has two types of motion are spiral and square The method of searching the solution area at random and Avoid getting caught in your local Optima The space of the solution arises exp with the problem to be solved It works to find and replace the worst solution with the best solutions. The results of this algorithm confirm its high performance compared to other optimization algorithms. 	<ul style="list-style-type: none"> The search space may be large and therefore it takes more time to find the best solution. 	Npop , X , Y , F , S	M , UB , LB
Lion-AYAD	<ul style="list-style-type: none"> Dynamic machinizam can work with different length of sequence (small, middle and long sequence) Reduce the time complexity for find optimal protein High accuracy because in each step fined optimal solution before move to the next steps Add the new four features of lion make it more robust 	<ul style="list-style-type: none"> Need more computation Reminded need from user determine the start of sequence 	X , Y , C , A , D,F , X _{rand}	M , UB , LB

Where X is input dataset, Y is the goal, F is the activation function, N is the number of nests, and T number of iteration, W is weights, Jc is Jumping constant, E is error ratio, Npop number of population, S is seed, M is mating, UB is upper dimensions, LB is the lower dimensions, Xrand = randomly selected solution vector, C & A is coefficient vectors, D is the direction of motion, t is current iteration, r is the random vector, a is the vector decreasing from 2 to 0 throughout the iteration, p is the probability of motion, spiral or square, b is the logarithmic spiral constant, and l is the random number in the range [-1, 1]

Table 7

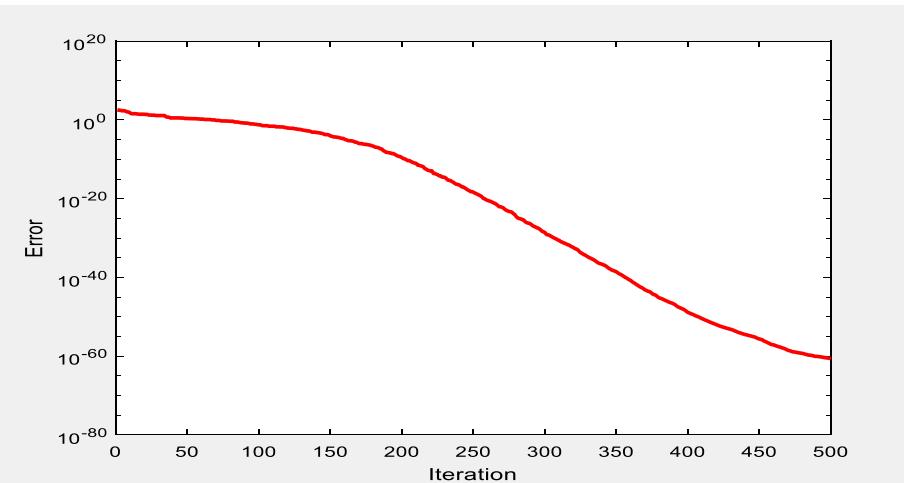
Main Parameters effect on optimization techniques.

Algorithm	Parameters																											
	X	Y	F	N	T	W	Jc	E	N _{pop}	S	M	UB	LB	C	A	D	X _{rand}	t	n	p	a	r	l	p	r1	r2	c1	C2
PSO	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	v	v	v	v		
COA																												
WOA	v	x	v	x	x	x	x	x	x	x	x	x	x	v	v	v	v	v	v	v	v	v	x	x	x	x		
LOA	v	x	x	x	x	x	v	v	v	v	v	v	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
LION-AYAD	x	v	x	v	x	v	v	v	v	v	v	v	v	x	x	x	x	x	x	x	v	v	v	v	x	x		

Table 8

Comparison among the evaluation measures, which affect Optimization Techniques.

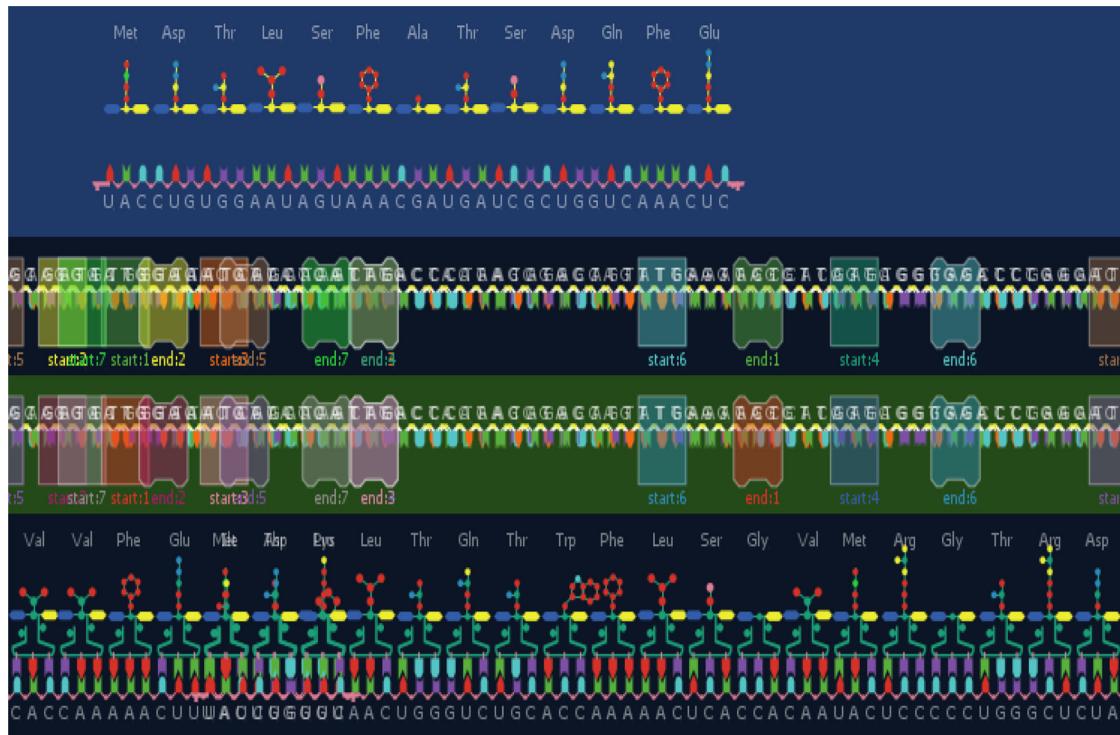
Evaluation-Measures	PSO	COA	WOA	LOA	LION-AYAD
Accuracy	84.87%	72.89%	91.54%	90.01%	92.51%
True positive (TP)	86%	81%	88%	87%	89%
Precision	97%	83%	92%	93%	91%
Recall	91.16%	77.19%	89.81%	84.81%	82.21%
F-Measure	93.58%	83.61%	93.82%	95.23%	97.79%

**Fig. 11.** Proteins generated by active DNA Sequence have eight Cut Points.

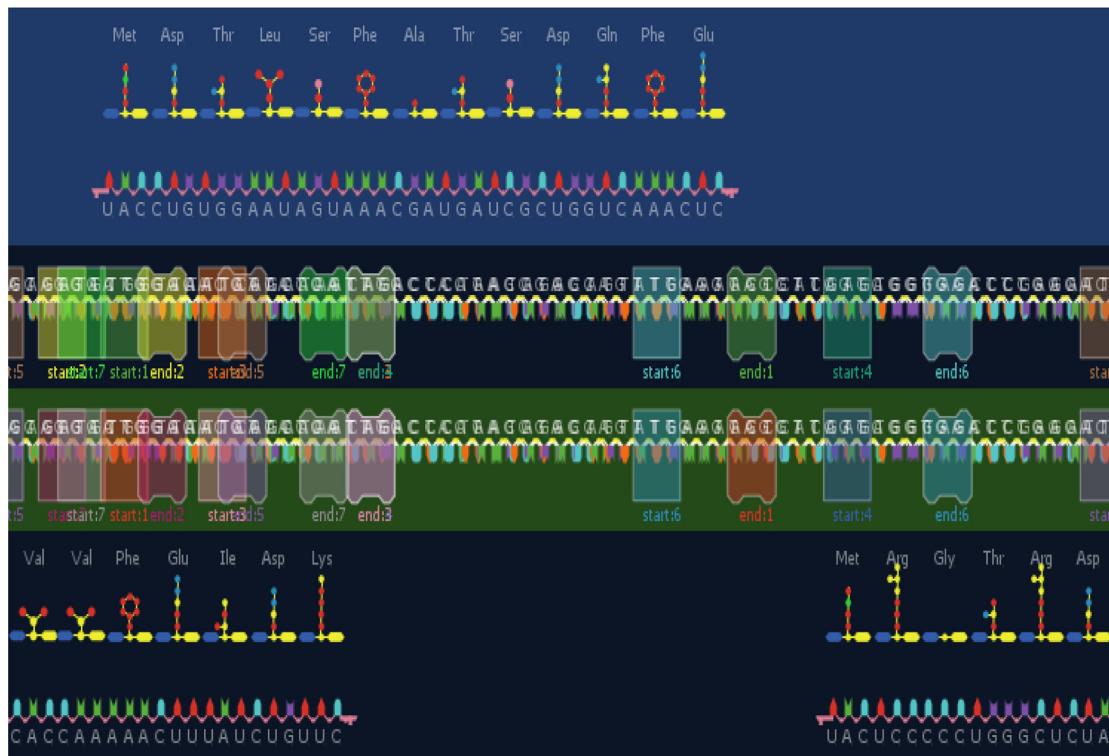
it can be secondary ones, which contribute to the primary parameters to provide an optimum solution. That used to build a new Lion-AYAD optimizer, it is necessary to analyze these parameters and make the comparison for the presence and absence of each one to determine their effects in the optimization techniques. **Table 7** shows compression between optimization techniques parameters.

As a results, we can summarized the main advantages of Lion-AYAD is the dynamic of application on different length of sequence this meaning the length of sequence become dynamic where the algorithm can work on small, middle and long sequence, also the determined the active sequence from multi sequence become

achieved automatically through the first features of the algorithm, the algorithm in each steps searching the short path to transform from the start point to goal and the lions move through sequence of sub-goals by search on the best position before travelling to it this satisfy by the second features of the algorithm. Generating mRNA molecules for the active sequence of DNA (i.e., lion have best value fitness determined through second features of that algorithm) through: Build mRNA for each gene (i.e., move in specific region known the start and end point of it) Then Build protein from mRNA Finally, Build tRNA for protein after split it into triplet this achieved by add the third properties to algorithm under title cooperative. How can synthesis each triple of tRNA to generated pro-



(a) Synthesis to mRNA Sequence generated from Active Sequence of DNA have Eight Cut Points



(b) Translation mRNA sequence into Portions

Fig. 12. The Relation between the Number of Iteration and Error.

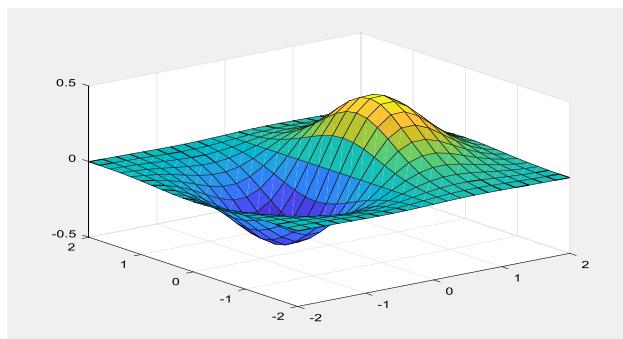


Fig. 13. The Values of Fitness for Lions all Iterations.

teins by Lion-AYAD optimization Algorithm this achieved through applied four types of rules in the final feature add by algorithm called deep composite of triple of tRNA.

The Lion-AYAD algorithm is appear as pragmatic optimization model as explained in [Table 8](#) and [Fig. 12](#) that shown the error rate generation in each iteration, while [Fig. 13](#) presented the values of fitness of lions. [Fig. 15](#) displayed all the results of algorithm from grouping the lions, reduce the cost and increase find the optimal fitness (i.e., solution). Lion-AYAD can apply in other field such as chemistry to reduce the cost, time and human effort by determined the Main Rules (i.e., limitations) to avoiding incorrect interactions of materials, at the same time as optimal model to find optimal new materials can product through interaction among materials computationally before apply the experiment in lab.

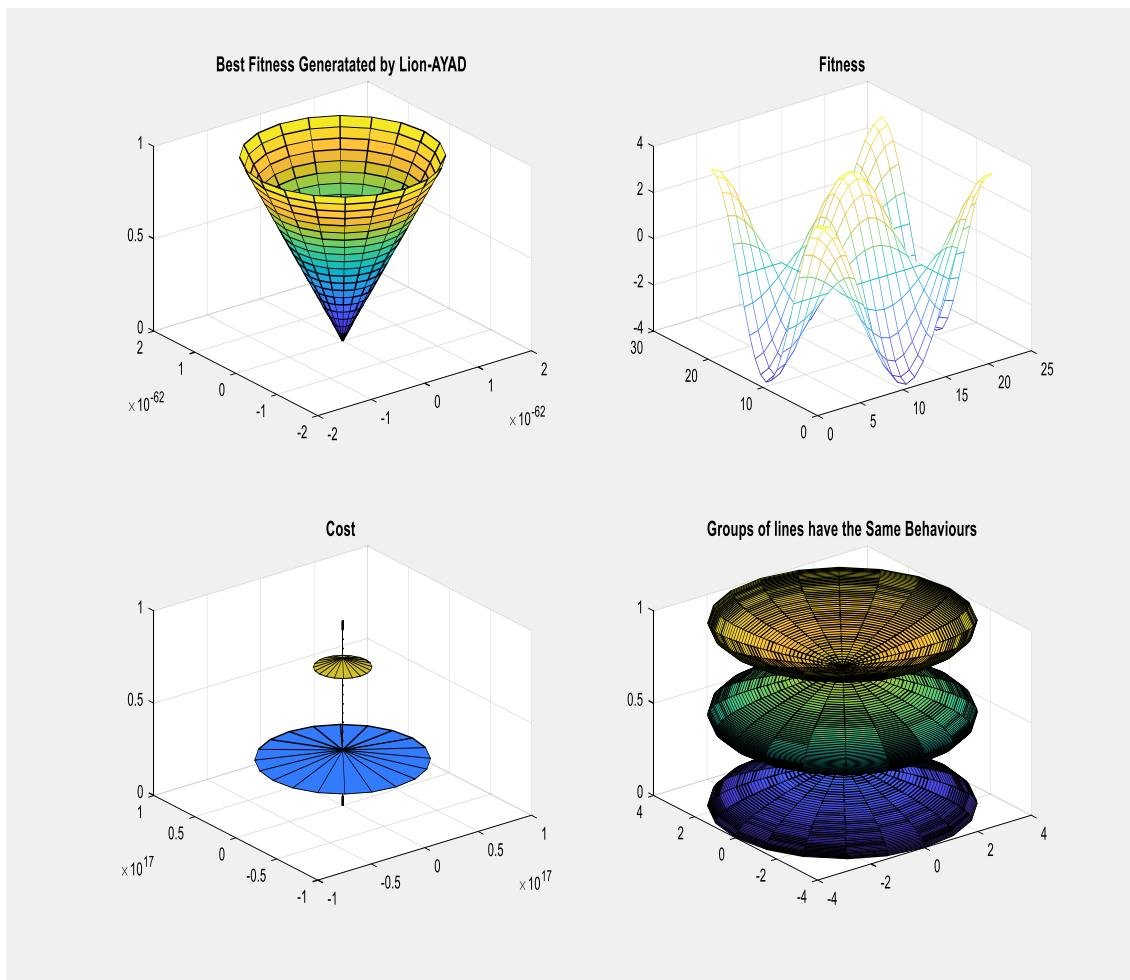


Fig. 14. Results of (a) Best Fitness generated by Lion-AYAD, (b) The Values of Fitness for Best Population, (c) Explain how the Value of cost function reduce to reach of best values, (d) Best number groups of lines population that have the same behaviors results by Lion-AYAD.

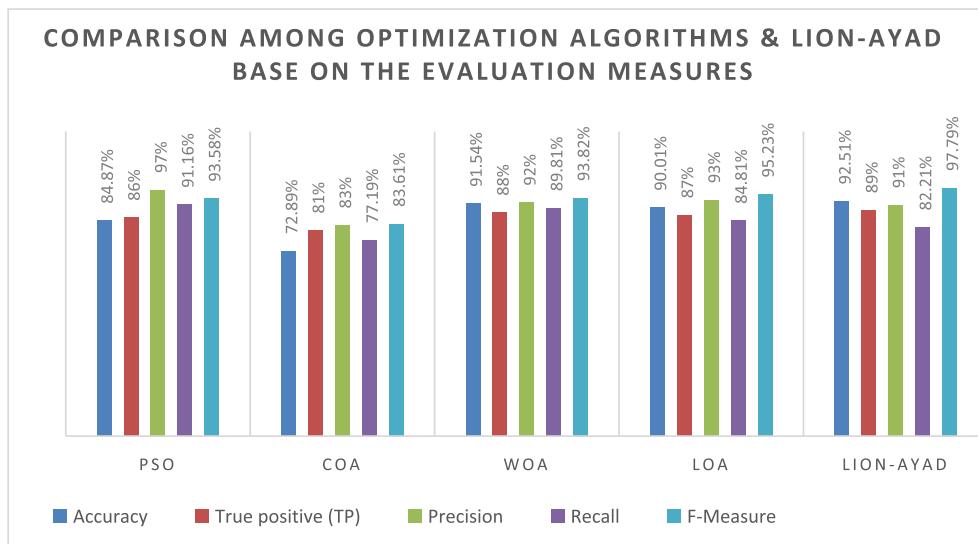


Fig. 15. Comparison among Optimization algorithms & Lion-AYAD base on the evaluation measures.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the author.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] Emry E, Zawbaa HM, Hassanien AE. Binary ant lion approaches for feature selection. *Neurocomputing* 2016;213:54–65.
- [2] Zawbaa HM, Emry E, Grosan C, Bongard J. Feature selection via chaotic antlion opti-mization. *PLoS ONE* 2016;11(3):e0150652.
- [3] Gupta S, Kumar V, Rana K, Mishra P, Kumar J. Development of ant lion optimizer toolkit in labview. In: 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH). p. 251–6.
- [4] Yamany W, Tharwat A, Hassanin MF, Gaber T, Hassanien AE, Kim TH. A new multi-layer perceptrons trainer based on ant lion optimization algorithm. In: 2015 Fourth International Conference on Information Science and Industrial Applications (ISI). IEEE; 2015. p. 40–5.
- [5] Rajan A, Jeevan K, Malakar T. Weighted elitism-based ant lion optimizer to solve optimum var planning problem. *Appl Soft Comput* 2017;55:352–70.
- [6] Kamboj VK, Bhadoria A, Bath SK. Solution of non-convex economic load dispatch problem for small-scale power systems using ant lion optimizer. *Neural Comput Appl* 2017;28(8):2181–92.
- [7] Ali ES, Abd Elazim SM, Abdelaziz AY. Optimal allocation and sizing of renewable distributed generation using ant lion optimization algorithm. *Electr Eng* 2018;100(1):99–109.
- [8] Petrović M, Petronjević J, Mitić M, Vuković N, Miljković Z, Babić B. The ant lion optimization algorithm for integrated process planning and scheduling. In: Applied Mechanics and Materials. Trans Tech Publ.; 2016. p. 187–92.
- [9] Raju M, Saikia LC, Sinha N. Automatic generation control of a multi-area system using ant lion optimizer algorithm based pid plus second order derivative controller. *Int J Electr Power Energy Syst* 2016;80:52–63.
- [10] Dubey HM, Pandit M, Panigrahi BK. Ant lion optimization for short-term wind integrated hydrothermal power generation scheduling. *Int J Electr Power Energy Syst* 2016;83:158–74.
- [11] Mirjalili S, Jangir P, Saremi S. Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. *Appl Intelligence* 2017;46(1):79–95.
- [12] Al-Janabi S, Alkaim AF. A nifty collaborative analysis to predicting a novel tool (DRFLLS) for missing values estimation. Springer, *Soft Comput* 2020;24(1):555–69. doi: <https://doi.org/10.1007/s00500-019-03972-x>.
- [13] Al-Janabi S, Alhashmi S, Adel Z. Design (More-G) Model Based on Renewable Energy & Knowledge Constraint. In: Farhaoui Y, editor. *Big Data and Networks Technologies*. BDNT 2019. Lecture Notes in Networks and Systems. Cham: Springer; 2020. doi: https://doi.org/10.1007/978-3-030-23672-4_20.
- [14] Alkaim AF, Al-Janabi S. Multi Objectives Optimization to Gas Flaring Reduction from Oil Production. In: Farhaoui Y, editor. *Big Data and Networks Technologies*. BDNT 2019. Lecture Notes in Networks and Systems. Cham: Springer; 2020. doi: https://doi.org/10.1007/978-3-030-23672-4_10.
- [15] Al-Janabi S, Mahdi MA. Evaluation prediction techniques to achievement an optimal biomedical analysis. *Int J Grid Utility Comput* 2019;10(5):512–27.
- [16] Maziar Y, Fariborz J. 'Lion Optimization Algorithm (LOA): A nature-inspired metaheuristic algorithm. *J Comput Des Eng* 2016;3(1):24–36. doi: <https://doi.org/10.1016/j.cde.2015.06.003>.
- [17] Al-Janabi S, Alkaim A, Al-Janabi E, Aljeboree A, Mustafa M. Intelligent forecaster of concentrations (PM2.5, PM10, NO2, CO, O3, SO2) caused air pollution (IFCsAP). *Neural Comput Appl* 2021;33(21):14199–229. doi: <https://doi.org/10.1007/s00521-021-06067-7>.
- [18] Al-Janabi S, Alkaim AF, Adel Z. An Innovative synthesis of deep learning techniques (DCapsNet & DCOM) for generation electrical renewable energy from wind energy. *Soft Comput* 2020;24:10943–62. doi: <https://doi.org/10.1007/s00500-020-04905-9>.
- [19] Al-Janabi S, Kadhum G. Synthesis Biometric Materials Based on Cooperative Among (DSA, WOA and gSpan-FBR) to Water Treatment. In: Abraham A, editor. *Proceedings of the 12th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2020)*. SoCPaR 2020. *Advances in Intelligent Systems and Computing*. Cham: Springer; 2021. doi: https://doi.org/10.1007/978-3-030-73689-7_3.
- [20] Al-Barmani Z, Al-Janabi S. Intelligent Data Mining Techniques to Verification of Water Quality Index. In: Abraham A, Hanne T, Castillo O, Gandhi N, Nogueira Rios T, Hong TP, editors. *Hybrid Intelligent Systems*. HIS 2020. *Advances in Intelligent Systems and Computing*. Cham: Springer; 2021. doi: https://doi.org/10.1007/978-3-030-73050-5_58.
- [21] Al-Janabi S, Salman AH. Optimization Model of Smartphone and Smart Watch Based on Multi Level of Elitism (OMSPW-MLE). In: Misra S, Kumar TA, Piuri V, Garg L, editors. *Artificial Intelligence for Cloud and Edge Computing, Internet of Things (Technology, Communications and Computing)*. Cham: Springer; 2022. doi: https://doi.org/10.1007/978-3-030-80821-1_7.
- [22] Mahdi MA, Al-Janabi S. A Novel Software to Improve Healthcare Base on Predictive Analytics and Mobile Services for Cloud Data Centers. In: Farhaoui Y, editor. *Big Data and Networks Technologies*. BDNT 2019. Lecture Notes in Networks and Systems. Cham: Springer; 2020. doi: https://doi.org/10.1007/978-3-030-23672-4_23.