Applied Soft Computing Journal 104 (2021) 107197

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Expeditious COVID-19 similarity measure tool based on consolidated SCA algorithm with mutation and opposition operators

Mohamed Issa

Computer and Systems Department, Faculty of Engineering, Zagazig University, Zagazig, Egypt
Faculty of Computers and Informatics, Nahda University, Beni Suef, Egypt

ARTICLE INFO

Article history:
Received 12 September 2020
Received in revised form 9 February 2021
Accepted 15 February 2021
Available online 20 February 2021

Keywords:
COVID-19
Bioinformatics
Pairwise global alignment
Sine–Cosine optimization algorithm
Particle swarm optimization algorithm

ABSTRACT

COVID-19 is a global pandemic that aroused the interest of scientists to prevent it and design a drug for it. Nowadays, presenting intelligent biological data analysis tools at a low cost is important to analyze the biological structure of COVID-19. The global alignment algorithm is one of the important bioinformatics tools that measure the most accurate similarity between a pair of biological sequences. The huge time consumption of the standard global alignment algorithm is its main limitation especially for sequences with huge lengths. This work proposed a fast global alignment tool (G-Aligner) based on meta-heuristic algorithms that estimate similarity measurements near the exact ones at a reasonable time with low cost. The huge length of sequences leads G-Aligner based on standard Sine–Cosine optimization algorithm (SCA) to trap in local minima. Therefore, an improved version of SCA was presented in this work that is based on integration with PSO. Besides, mutation and opposition operators are applied to enhance the exploration capability and avoiding trapping in local minima. The performance of the improved SCA algorithm (SP-MO) was evaluated on a set of IEEE CEC functions. Besides, G-Aligner based on the SP-MO algorithm was tested to measure the similarity of real biological sequence. It was used also to measure the similarity of the COVID-19 virus with the other 13 viruses to validate its performance. The tests concluded that the SP-MO algorithm has superiority over the relevant studies in the literature and produce the highest average similarity measurements 75% of the exact one.

1. Introduction

Coronavirus disease 2019 (COVID-19) is a contagious virus created as a result of an evolution of severe acute respiratory syndrome coronavirus (SARS-CoV). The infected people were detected firstly in December 2019 in Wuhan city (China) and became a dilapidation pandemic when it flare-up through most of the countries. Until December 2020, more than 70 million infected cases were reported include 1.5 million deaths as reported by the World Health Organization (WHO). COVID-19 is a critical human disease that infects the liver, nervous systems, and respiratory system [1,2].

It transmits from bats to human and it has high mobility for transmission from human to human through the air, close personal contact, touching surfaces containing viral particles, and rare stool contamination [3].

Therefore, Intensive efforts are being made to analyze the virus to design a drug for it and model the pandemic spreading to overcome the devastating proliferation of COVID-19 [4,5]. Studies have been performed for modeling the physical transportation of COVID-19 in the air [6,7]. In [8], the stability of the transmission model of COVID-19 that was developed based on the SEIR model was investigated under different control strategies. In [9], The transmission of the airborne germ of COVID-19 was provided from a physics view based on fluid dynamics analysis methodology. Another trial was done to minimize the indoor transmission of COVID-19's airborne [10]. The effect of weather on the transmission of COVID-19 was researched also [11,12]. Besides, another research direction of COVID-19 is testing the influence of temperature on its spreading [13–15].

Computational intelligent tools were developed for analyzing the behavior of the virus such [16–23] besides intelligent diagnosis tools of COVID-19 were proposed [19,22,24–27]. Also, docking tools are developed for docking antibodies and peptides against the ligands of protein of COVID-19 [28,29]. Also, the internet of things research was performed to overcome the spreading of COVID-19 [30–32]. All these research efforts aim for analyzing the pandemic and designing a drug for it. One of the important tools that aid in analyzing COVID-19 for constructing its phylogenetic trees, predicting its structure is measuring the similarity of COVID-19 with other viruses. The pairwise global

E-mail addresses: Mamohamedali@eng.zu.edu.eg, Mohamed.elsayed@nub.edu.eg.

https://doi.org/10.1016/j.asoc.2021.107197
1568-4946/© 2021 Elsevier B.V. All rights reserved.

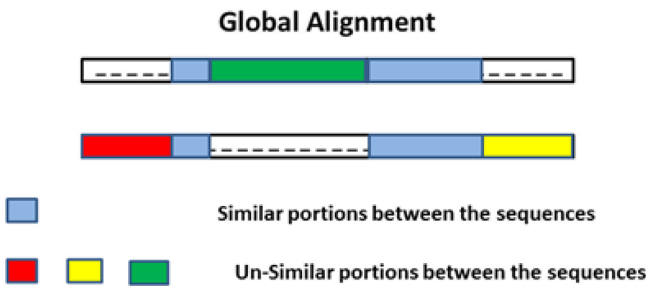

Global Alignment
Similar portions between the sequences
Un-Similar portions between the sequences

Fig. 1. Aligning two biological sequences using global alignment.

alignment algorithm proposed by Needleman–Wunsch (NW) [33] is the most accurate technique for measuring the similarity of two biological sequences. Pairwise global alignment aligns the entire of the two sequences not the portion of sequences such as local sequence alignment [34] such as in Fig. 1. A similar portion of the two sequences is colored in blue while the other portions align with gaps '-' to shift the similar ones.

The most similar viruses of COVID-19 can be detected by using the NW algorithm to align them against the huge biological databases of viruses.

The main limitation of the NW global alignment algorithm is its huge consuming time especially for huge length sequences (COVID-19 has a length exceeds 7000 bases pair) however it provides the most accurate alignment results.

Hence, a fast global alignment tool G-Aligner) is needed to be developed for fast primary scanning of databases to detect the viruses with the highest similarity scores (near to the exact ones founded by the NW algorithm). The aim of this primary scanning is to filter the huge number of sequences in the database into some sequences with reasonable similarity scores as near as possible from the exact score which decreases the search time. NW algorithm can be used to align these filtered sequences to measure the accurate and exact similarity score. Fig. 2 shows the rule of using the accelerated global aligner (G-Aligner) technique to align COVID-19 with another virus to test the similarity between them. The results of similarity will be used with other applications such as designing drugs, prediction of protein structure, and constructing the phylogenetic tree of COVID-19.

The pairwise global alignment algorithm was accelerated in the literature using hardware acceleration devices such as using Graphical Processing Unit (GPU) devices [35–38] and Field Programmable Gate Arrays (FPGAs) [39–44]. These quick versions of global alignment propose efficient speedup when using massive parallelization devices but are cost money.

Hence, the necessity of this work is developing a low cost accelerated global aligner tool (G-Aligner) that can produce a fast measurement of the similarity between pair of biological sequences with reasonable results near to the exact ones produced by the NW algorithm. The main innovation is using the stochastic search of meta-heuristic algorithms [45] in designing the G-Aligner tool.

The meta-heuristic algorithm is a search-based algorithm that accelerates exploring the search space of the problem based on random movement to find the best solutions [45]. The meta-heuristic algorithm mimics the search methods from nature, physics, or humans [45] such as the Sine–Cosine Optimization algorithm (SCA) [46] which mines the search space by attracting the search agents toward the best candidates based on the sine and cosine operators. Besides, Particle Swarm Optimization (PSO) [47] mimics the search strategy from the bird flocking from nature. Also, there are a lot of released algorithms such as Ions Motion Optimization (IMO) [48], Lightning Attachment Procedure Optimization [49]. Gravitational Search Algorithm (GSA) [50], Electromagnetic Field Optimization (EFO) [51], Moth-Flame Optimization (MFO) [52], and other hundreds of algorithms are developed.

Meta-heuristic algorithms are succeeded to enhance the performance of many bioinformatics tools such as protein folding prediction [53,54], protein structure prediction [55–57], Drug discovery [58,59], local alignment [60], and other applications which are reviewed in [61,62] and that motivates for using meta-heuristic algorithms for accelerating the pairwise global alignment.

In this work, pairwise global alignment is formulated as an optimization problem where the objective is accelerating the execution time while producing a reasonable similarity score near to the exact one. G-Aligner was implemented using SCA [46] and the performance of it was validated on a two set of experimental biological data. The first was a set of Homo sapiens biological sequences and the second was COVID-19 virus protein versus 13 proteins of other viruses. The G-Aligner based on SCA reduced the execution time significantly over various lengths of biological sequences but achieved an average similarity score of 39% of the exact one measured by NW global alignment [33]. Hence, the performance of G-Aligner based on SCA needs to be enhanced.

In the literature, there are previous studies to enhance the performance of SCA such as improving SCA by applying the opposition of solutions to increase the exploration of SCA (ISCA) [63]. In m-SCA [64], SCA was enhanced by applying the opposition on the solutions besides adding a self-adaptive parameter was added in the updating equations of SCA to enhance the exploitation of promising regions of search space.

Besides, SCA was merged with other algorithms to enhance the performance of it such as SCA-DE [65] and SCA-PSO [66]. In SCA-DE [65], SCA updates the search agents for several iterations and at the last of each iteration, DE was used to update the solutions based on the updating mechanism of DE.

In SCA-PSO [66], SCA updates its search agents for some iteration based on updating equation of SCA and the best fitness of each search agent of SCA are saved in addition to the best global solutions among all search agents. Then, PSO is used to update the search agents of SCA based on updating the equation of PSO toward the best solution achieved by each search agent and the best global solution and the new solutions will be updated using SCA again. SCA-DE and SCA-PSO were developed to get the benefit of the two algorithms (efficient exploration of the search space using SCA beside the efficient exploitation of DE and PSO).

The chaotic Sine–Cosine algorithm was merged with chaotic firefly (CSCF) [67] where when updating each agent it can be updated using chaotic SCA or chaotic firefly depends on the fitness value of each solution. There was 5 version of embedding the chaos parameters on the two algorithms while the updating equation was used chosen randomly. SCA-GWO [68] is integration between SCA and Gray Wolf Optimization (GWO) algorithm [69] that developed to benefit from the advantage of SCA for exploration and GWO for the exploitation of the search space. In SCA-GWO, SCA was executed first for all agents toward the best global solution founded and the best solution each agent visit. Then GWO was used to update the solutions for exploitation.

Besides, another different hybrid scheme of SCA and PSO (ASCA-PSO) was proposed that performs SCA and PSO in parallel as a two-layer. The bottom one responsible for exploring the search space using SCA while the upper one intensifies the best solutions founded from the bottom layer using PSO. ASCA-PSO succeeded to accelerate the performance of the local alignment algorithm [60].

G-Aligner was implemented using all the enhanced versions of SCA in the literature and ASCA-PSO produced the highest

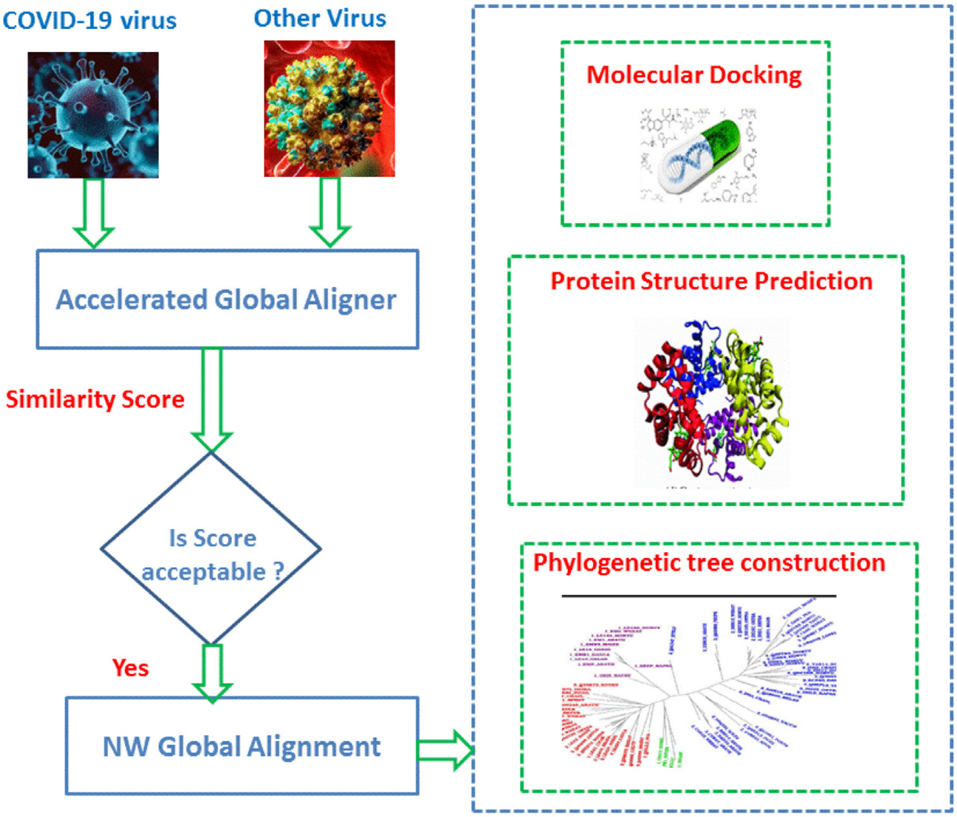


Fig. 2. The rule of using G-Aligner with NW alignment and other applications of analyzing COVID-19.

similarity scores (51.5% of the exact ones). The poor results of G-Aligner based on ASCA-PSO since its fall in local minima after some iteration and its exploration capability of SCA need to be enhanced.

The main argument of this work is enhancing the performance of ASCA-PSO for G-Aligner using the mutation operator in the updating equations of SCA to increase the efficiency of the exploration of the search space and applying the opposition on the solutions that fall in local minima to avoid it.

The main contributions in this paper are concluding as follows:

1- A fast and low cost pairwise global alignment technique was proposed based on meta-heuristic algorithms.
2- The SCA algorithm was improved by hybrid with PSO using mutation operators and opposition operators to enhance the exploration capability and avoiding trapping in local optima.
3- The performance of the proposed algorithm (SP-MO) was validated on a set of IEEE benchmark mathematical functions.
4- G-Aligner based on SP-MO algorithm was validated to measure the similarity of COVID-19 virus with other 13 viruses and achieved similarity scores 75% of the exact ones measured by NW global alignment algorithm at a reasonable time.

The structure of the article is organized as follows: Section 2 presents the basic information pairwise global alignment algorithm. Section 3 presents the preliminaries of SCA, PSO, and ASCA-PSO. In Section 4, the global alignment technique based on meta-heuristic (G-Aligner) is presented and in Section 5 the proposed SP-MO is presented. Section 6 proposes the experimental results of validating SP-MO. Finally, Section 7 concludes the proposed work and results.

2. Pairwise global alignment algorithm

NW global alignment algorithm [33] is the standard algorithm for performing pairwise global alignment and produce accurate alignment results. It depends on the dynamic programming approach [70] which calculates all possible alignments. It aligns the similar residues between two biological sequences hence a gaps '-' are need to be inserted to shift the similar residues.

The algorithm starts by constructing a scoring matrix that has a size of row and column (m+1) and (n+1) in order where m and n are the lengths of the two sequences. The first row and column are filled with the negative index of the cells. For example, the first cell in row and column has a score (0), the second cell in the first row has a score (-1) and the third one has a score (-2), and so on and the same for the first column. The scores of cells that are starting from the second row and column until the final cell of the scoring matrix will be computed according to Eq. (1) based on the corresponding residues of the two sequences.

$$\text{Score}\ (s, j) = \max \left\{ \begin{array}{c} \text{Score}\ (i-1, j-1) + \text{Similarity}\ (\text{Seq}_A\ (i)\ , \text{Seq}_B\ (j)) \\ \max_{k=1:i-1}\ (\text{Score}\ (i, k) + g_0 + k\ g_e) \\ \max_{k=1:i-1}\ (\text{Score}\ (k, j) + g_0 + k\ g_e) \end{array} \right\} \tag{1}$$

where the sequences to be aligned are represented as Seq_B and Seq_A and have lengths n and m respectively, i and j are the indices of the row and column where 1< i < m and 1< j < n. The scoring of gaps is based on linear gap ($g_o + kg_e$) penalty to penalize the scoring of consecutive gaps where the score of an open gap is g_o, score of the extended gap is g_e and k is several inserted gaps [71].

Similarity () is a function that is used to measure the similarity between residues of proteins. A different scoring schemes for measuring the similarity of residues of proteins such as

BLOcked SUbstitution Matrix (BLOSUM) and Point Accepted Mutation (PAM) [72]. The simple scoring scheme has penalized a score (+1) if the two residues are similar otherwise give (0) and this scheme which was applied in this paper and can be adjusted.

The second stage of alignment is tracing back to align the sequences that are performed after finishing the computing of the scoring matrix. The tracing starts from the final cell at the bottom right of the scoring matrix and finishes at the upper left cell. At each cell, it has three movements toward one of the upper, left, or diagonal cells. The next movement occurs toward the cell that has the maximum score. If the movement is diagonal then it means aligning similar two corresponding residues (one from each sequence). If the movement toward upper that means aligning one residue from the sequences in the row with a gap. If the movement toward the left then aligns a residue from the sequence in the column with a gap and this tracing movement resuming until reaching the start cell at first row and column.

After constructing the alignment, the similarity between the two sequences can be computed according to Eq. (2) where A and B represent the aligned sequences (with insertion gaps) and L denotes the length of (A or B).

$$\mathrm{Alignment}_{\mathrm{Score}} = \sum_{i=1}^{L} \left\{ \begin{array}{cc} \text{if } \mathrm{A_i} == \mathrm{B_i}\,, & \text{penalize } (+1) \text{ or positive score} \\ \text{otherwise} \quad , & \text{penalize zero} \end{array} \right\} \tag{2}$$

The time complexity of the exact global alignment algorithm (NW) is O(n^3), where (n) denotes the length of the sequences to be aligned (assuming the two sequences have the same length). It is clear that from the time complexity by increasing the length of sequences the execution time will be huge especially for sequences with huge lengths. Hence, there is a motivation to decrease the execution time of the NW algorithm by proposing a developed version of it based on meta-heuristic algorithms.

3. Preliminaries

The following subsections propose a brief description of the Sine–Cosine optimization algorithm (SCA), Particle Swarm Optimization (PSO) algorithms, and the hybrid algorithm of SCA and PSO algorithms (ASCA-PSO).

3.1. Sine-Cosine optimization algorithm (SCA)

SCA is a population-based optimization algorithm that depends on sine and cosine mathematical operators for updating the agents as in Eqs. (3) and (4).

$$\mathrm{P_i^{t+1}} = \left\{ \begin{array}{ll} \mathrm{P_i^t} + \mathrm{r_1} \sin(\mathrm{r_2}) \left|\mathrm{r_3 P_{gbest}} - \mathrm{P_i^t}\right| & \mathrm{r_4} < 0.5 \\ \mathrm{P_i^t} + \mathrm{r_1} \cos(\mathrm{r_2}) \left|\mathrm{r_3 P_{gbest}} - \mathrm{P_i^t}\right| & \mathrm{r_4} \geq 0.5 \end{array} \right\} \tag{3}$$

$$\mathrm{r_1} = \mathrm{a}\left(1 - \frac{\mathrm{t}}{\mathrm{T}}\right) \tag{4}$$

where (P_{i}) is the solution of the search agent (i), (P_{gbest}) the global best solution, (t) is the current iteration number, (T) is the maximum number of iterations, (r_1) is the parameter responsible for determining the next region of search and increase the exploration of search space for the higher value of it. (a) is a scaling factor that balances between the exploration and exploitation of SCA. Meanwhile, (r_2) defines the direction of movement toward or outwards (P_{gbest}), and (r_3) controls the effect of destination on current movement. (r_4) is used to switch between sine and cosine functions as in Eq. (3).

The steps of SCA are presented in Algorithm (1). The time complexity of SCA is O(T*n*C_{SCA}) where (n) is the size of populations and (C_{SCA}) is the time cost of updating all populations per one iteration, and (T) is the number of iterations.

3.2. Particle swarm optimization (PSO)

PSO is a swarm optimization algorithm that mimics the attitude of birds flocking for flying. It has a stochastic search strategy that depends mainly on the global communications between the search agents, where all search agents modify their movements pointed to the global search agents that finds the global solution. Besides, it memorizes the best solutions each search agent pass through, which influences the new update of it as stated in Eq. (5), and this memorization of location enhances the exploitation phase of PSO.

The updating equations of PSO are represented as Eqs. (5) and (6), where the particle ($\mathrm{P^{gbest}}$) has the global best position (solution) among all search agents and the best personal position ($P_{\mathrm{i}}^{\mathrm{best}}$) that each search agents found during the previous iterations.

$$v_i\,(t+1) = w * v_i\,(t) + c_1 rand\left(P_i^{best} - P_i\,(t)\right) + c_2 rand\left(P^{gbest} - P_i\,(t)\right) \tag{5}$$

$$P_i\,(t+1) = P_i\,(t) + v_i\,(t+1) \tag{6}$$

where v_i represents the velocity of the ith particle, c_1 and c_2 are the local the global best position coefficient in order. w is the inertia coefficient that estimates the influence of the prior velocity on the new estimated velocity. *rand* () is a uniformly distributed random variable in the range (0–1). The PSO search strategy as SCA in the algorithm (1) except the updating equations in step 6 will be Eqs. (5) and (6).

PSO has a complexity of time $O(T*n*c_{pso})$ where T, n, and c_{pso} express the number of iterations, the number of search agents, and the cost time of modifying the position of one search agent, respectively. PSO has a main advantage is the interchanging of information between search agents, which gives it more reliability to achieve an approximate optimal solution with acceptable convergence speed besides robustness. Besides, the agents move toward the best location it achieved in the previous iteration which makes its exploitation of the search space more efficient.

3.3. Two layer hybrid SCA and PSO (ASCA-PSO)

ASCA-PSO consists of two layers, the bottom layer (exploration layer) contains search agents that updates their movement according to the updating equations of SCA. Each search agent in the upper layer represents the best solution found from each group in the bottom layer that updates its movement using the PSO algorithm. There is a global solution (y_{gbest}) that represents the best solution founded among the agents of upper and bottom layers and it represents the output optimal solution.

The search agents in the bottom layer are divided into (M) groups where each group contains (N) search agents. Each group has the best agent in the upper layer ($\mathrm{y^k}$, k: 1 to M) represents the best solution founded from the search agents in the bottom group, and all best agents in the upper layer are moved according to the updating equation of PSO. Each search agent in the bottom layer updates its movement according to Eq. (7).

$$x_{ij}^{t+1} = \left\{ \begin{array}{l} x_{ij}^t + r_1 \sin(r_2) \left|r_3 y_i^t - x_{ij}^t\right| r_4 < 0.5 \\ x_{ij}^t + r_1 \cos(r_2) \left|r_3 y_i^t - x_{ij}^t\right| r_4 \geq 0.5 \end{array} \right\} \tag{7}$$

where, ($x_{\mathrm{i,j}}$) represents the solution of search agent in the bottom layer, ($\mathrm{y^i}$) is the best solution founded of the group (i) in the bottom layer, (i) and (j) represent the indices of solutions in the top and bottom layer respectively.

Besides, Eqs. (8) and (9) represents the updating movement equations of the search agents in the upper layer toward (y_{gbest})

Algorithm (1) Sine Cosine Algorithm

```
Initialize a set of population solutions (P_i), algorithm parameters ( r_1, r_2, r_3, and r_4)
Repeat
    Evaluate the objective function based on population solution
    Update the best solution obtained so far (P_gbest)
    Update r_1, r_2, r_3 and r_4
    Update the next position of population solutions using Eq. 3 and Eq. 4
Until (T < maximum number of iterations)
Return the best solution (P_gbest) obtained as the global optimum
```

which represents the best global solution founded among all search agents in the upper and bottom layers.

$$v_i^{t+1} = w * v_i^t + c_1 rand \left(y_i^{pbest} - y_i^t\right) + c_2 rand \left(y^{gbest} - y_i^t\right) \tag{8}$$

$$y_i^{t+1} = y_i^t + v_i^t \tag{9}$$

The synergy of execution SCA and PSO in two-layer form is performed by executing the first group in the bottom layer which explores the search space using the updating strategy of SCA based on Eq. (7) and update (y^1) in the upper layer and (y_{gbest}) if a better fitness solution is founded. Then the agent (y^1) is updated based on PSO updating strategy based on Eqs. (8) and (9) to intensify the best solution founded from the exploration in the bottom layer. Then the second group in the bottom layer is executed and (y^2) in the upper layer is updated for intensification of the search space and then the third group in the bottom and so on.

Hence, as shown in this hybridization mechanism after some exploration of the search space using SCA, it is intensified using PSO around the best-explored regions produced by the bottom layer. This mechanism of hybridization will increase the diversity of produced solutions and enhance the quality of solutions which was proved for enhancing the Fragmented Local Aligner Technique (FLAT) [60]. This advantage motivates to use of ASCA-PSO for optimization of the global sequence alignment algorithm.

ASCA-PSO algorithm has a time complexity of O(TM $\left(Nc_{sca} + c_{pso}\right)$) where N and M are the number of search agents in the bottom and top layer in order. c_{pso} and c_{sca} are the time cost for updating each search agent for PSO and SCA in order, and T is the number of iterations.

4. Pairwise global alignment based on meta-heuristic algorithms (G-Aligner)

This section presents the procedure of performing the pairwise global alignment based on a stochastic search using meta-heuristic algorithms. The global alignment algorithm can be formulated as an optimization problem where the desired output is finding the best alignment between the two sequences by matching the similar residues of proteins in a reasonable time smaller than that is consumed by NW global alignment algorithm.

To match the similar residues it is needed to insert gaps in different positions on the aligned sequences to shift the matching residues. Hence, the objective is inserting gaps (which can be 30% of the length of the sequences to be aligned) at locations in the aligned sequences that maximize the similarity of biological sequences.

The solution to this optimization problem is the locations of gaps in each sequence that maximize the similarity scores and the fitness is the similarity score of the aligned sequences that is estimated based on Eq. (2).

In case of the sequences have different lengths then it is needed first to equalize the aligned sequences by adding extra gaps to the shorter sequence that equal the difference between the length of the two sequences. For example, Fig. 3 shows the representation of solutions for performing the global alignment based on meta-heuristics algorithms.

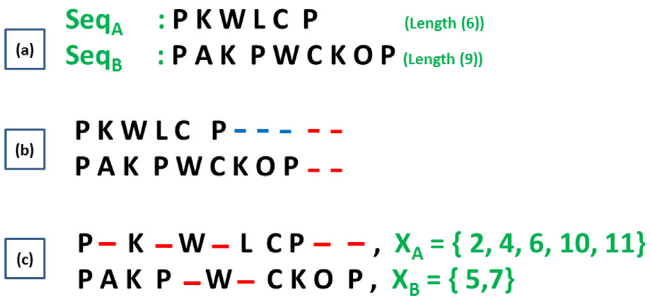

(a)
Seq_A :PKWLC P (Length (6))
Seq_B :PAK PWCKOP (Length (9))
(b)
PKWLC P- - - - - -
PAK PWCKOP- -
(c)
P– K –W– L CP– –, X_A = { 2, 4, 6, 10, 11}
PAK P –W– CKO P, X_B = { 5,7}

Fig. 3. Representation of the solution of the global alignment based on stochastic algorithms.

In part (a) of Fig. 3, the pair of sequences to be aligned have different lengths hence in part (b) the difference between lengths filled by blue gaps to equalize the two sequences and the red gaps are extra gaps inserted as a 30% of the shorter length for example. In part (c) of Fig. 3, the gaps are inserted in random locations to shift the residues to align similar ones. As shown X_A and X_B keeps the indices of gaps in each sequence hence X_A and X_B together represent one solution.

The similarity scores are estimated using Eq. (2) (fitness function) and each solution moves its gaps indices (positions in each sequence) toward the indices of the best solution founded (the solution that has the maximum similarity score) based on the updating mechanism of the meta-heuristic technique used.

The general procedure of performing the global alignment based on the meta-heuristic technique as follows :

1- Constructing the aligned arrays of the two sequences for (N) solutions after equalization of the lengths of the pair of sequences and estimating the extra gaps as a specified percentage of the shorter sequence.
2- Initialize (N) solutions with indices in each sequence by spreading the gaps over the entire length of sequences in random locations.
3- Find the best solution from the N solutions that give the maximum similarity score based on Eq. (2).
4- Update the solution toward the best solution founded according to the updating equation of the meta-heuristic techniques.
5- Evaluating the similarity scores of the updated solutions and repeat from step (3) for some iterations.
6- Output the aligned sequences according to the best location of gaps that maximize the similarity score of the two sequences.

5. The improved SCA algorithm based on mutation operator and opposition (SP-MO) for G-Aligner

This section proposes the procedure of the improved SCA algorithm based on integration with PSO and using mutation and opposition operators. In the proposed algorithm (SP-MO), the agents (x) are divided into some groups which update their movements based on SCA algorithm for exploring the search space. Then, before updating the following group, the best agent of the current group (y) is determined and is updated toward the global best agent of all agents ($\mathrm{y}^{\mathrm{gbest}}$) based on PSO operators for the exploitation of the search space. The global best solution ($\mathrm{y}^{\mathrm{gbest}}$) is updated if any solutions achieved better fitness.

This integration mechanism between SCA and PSO balance the exploration and exploitation of the search space. As mentioned in Section 4, the locations of gaps in each sequence represent the solution of the optimization problem which needs to be moved over the entire length of the sequences to maximize the similarity score. Due to the huge length of the sequences, there is a high possibility for trapping in local minima which are defined for the G-Aligner in the following.

If the locations of gaps of two solutions are near hence the fitness of the two solutions are the same approximately. Hence, the local minima of G-Aligner are represented as the locations of solutions that become nearby and be in the approximate stable form (movements in a small range of locations) which leads to approximate fitness (similarity score). Fig. 4 represents the local minima of G-Aligner where the vertical blue arrow represents the alignment score scale (fitness of solutions) and the seven circles represent the fitness of the solutions. Circle (2) has the highest similarity score and circles (3), (4), and (6) have fitness larger than the average score (red line) while circles (1), (5), and (7) have a fitness smaller than the average line.

The solutions are attracted toward the global best solution (circle 2), hence as shown in the figure solutions (3), (4), and (6) have fitness approximate to the best solutions (circle 2) which means the locations of inserted gaps of the solutions are in near positions. While locations of gaps in solutions (1), (5), and (7) are located in positions that are far from that of the best solutions so they produce lower similarity scores than that of the best position (smaller than the average score). So, solutions (3), (4), and (6) become stable move with small step movement which mean they lie in local minima. Hence, if the solutions (3), (4), and (6) are opposed that may produce better fitness and enhance the best fitness founded.

The condition to determine if a solution trapped in local minima is the difference between its fitness and the best fitness founded being lower than the average fitness among all fitness of solutions. As shown in Fig. 4, for the circle (6) the distance to best solution (circle 2) is L_1 is lower than average fitness (L_3) so it needs to opposed while L_2 is larger than L_3 which means the locations of gaps of solution (7) are far from that of the best solution.

The opposing is occurred on the search agents of the upper layer of ASCA-PSO due to it influences the movement updating of the search agent of the bottom layer. Therefore, search agents in the bottom layer also explored and enhance their fitness if their best solution in the upper layer was opposed. The condition of a solution trapped in local minima and the procedure of opposing as follows:

```
If ($F^{gbest} – F^i$) < Average ($F^i$)
    1. Find the opposition of the solution (i) in the upper layer as follow :
        - $y_A(i) = L_A - y_A(i)$ , if ($y_A(i) = 0$ ) then ($y_A(i) = 1$)
        - $y_B(i) = L_B – y_B(i)$ , if ($y_B(i) = 0$ ) then ($y_B(i) = 1$)
    2. Estimate ($F^i$) after apply the opposition.
    3. If ($F^i > F^{gbest}$) then $y^{A\_gbest} = y_A(i)$ , $y^{B\_gbest} = y_B(i)$
```

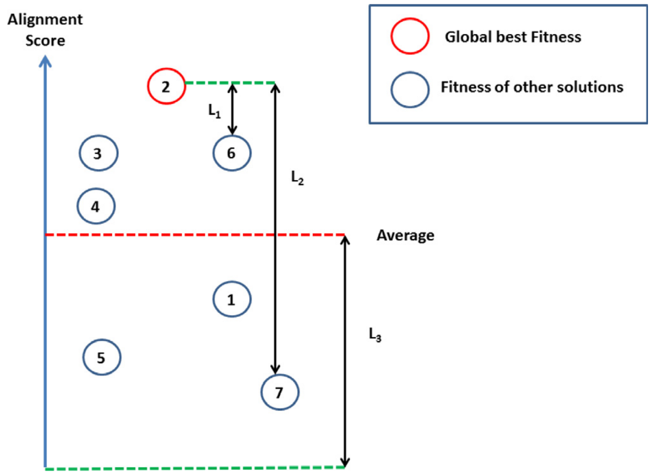


Fig. 4. The local minima of the proposed G-Aligner based on the SP-MO algorithm.

where (y_A) and (y_B) represents the array of a position of the inserted gaps in the aligned sequences A and B, i is the index of the search agent of the upper layer, ($\mathrm{F}^{\mathrm{gbest}}$) is the global best alignment score among all search agents of the bottom and upper layer, (F^{i}) is the alignment score of the search agent (i).

A mutation operator is applied to the updating equations of the bottom layer (search agents of SCA) to increase the avoidance of the local minima of the problem. Since mutation operators are succeeded to enhance the exploration capability of many meta-heuristic techniques to increase the diversity of generated solutions [73–78]. Two common mutation operators are used to enhance the meta-heuristic techniques are Gaussian mutation (GM) and Cauchy mutation (CM) operators.

The previous study proved that CM operator has an efficient search capability more than GM operator [73,77,79,80]. The main reason behind that is CM operator has a broader distribution in the horizontal direction more than the vertical one however the GM operator has a broader distribution but in the vertical direction. Hence, this is the main motivation to use the CM operator.

The density function of the CM operator is used as follow :

$$f_{(0,g)}\left(\gamma\right) = \frac{g}{\pi(g+\gamma^2)},\ \gamma = \tan(\pi(rand-0.5)) \tag{10}$$

Where g is the proportion parameter and is assigned value (1) [77], rand is a uniform random generator function in the range (0,1).

The updating equation of SP-MO after adding the CM are as follow :

$$x^{i,j} = \left\{ \begin{array}{ll} x^{i,j} + M_{Op} r_1 \sin(r_2) \left| r_3 y^i - x^{i,j} \right| & r_4 < 0.5 \\ x^{i,j} + M_{Op} r_1 \sin(r_2) \left| r_3 y^i - x^{i,j} \right| & r_4 \geq 0.5 \end{array} \right\} \tag{11}$$

$$\mathbf{v}^{\boldsymbol{i}} = \mathbf{w}{*}\mathbf{v}^{\boldsymbol{i}} + \mathbf{c}_1\mathbf{rand}\left(\mathbf{y}^{\boldsymbol{gbest}} - \mathbf{y}^{\boldsymbol{i}}\right) + \mathbf{c}_2\mathbf{rand}\left(\mathbf{y}^{\boldsymbol{i}}_{\boldsymbol{best}} - \mathbf{y}^{\boldsymbol{i}}\right) \tag{12}$$

$$\mathbf{y}^{\boldsymbol{i}} = \mathbf{y}^{\boldsymbol{i}} + \mathbf{v}^{\boldsymbol{i}} \tag{13}$$

where ($\mathrm{x}^{\mathrm{i,j}}$) represents the position of a gap in the sequences, (i) is the index of the group and (j) is the index of agent in the group. (y^{i}) is the best solution of search agents in the group (i) and is updated according to Eqs. (12) and (13) (updating equation of PSO) toward the global best solution founded among all search agents in ($\mathrm{y}^{\mathrm{gbest}}$). ($M_{\mathrm{op}}$) represents the Cauchy mutation operator used to increase the exploration of the solution in the bottom layer to increase the diversity of solutions.

Table 1
Benchmark of mathematical test functions (Dimension = 50).

Function		Bounds	F_{min}
F_1	$-20e^{-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}} - e^{-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)}} + 20 + e^1$	$[-32, 32]$	0
F_2	$10d + \sum_{i=1}^{d}[x_i^2 - 10\cos(2\pi x_i)]$	$[-5.12, 5.12]$	0
F_3	$\sum_{i=1}^{d}\frac{x_i^2}{4000} - \prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	0
F_4	$\sum_{i=1}^{d} ix_i^4$	$[-1.28, 1.28]$	0
F_5	$\sum_{i=1}^{d} x_i^2$	$[-5.12, 5.12]$	0
F_6	$\sum_{i=1}^{D} \lfloor x_i + 0.5 \rfloor^2$	$[-100, 100]$	0
F_7	$\sum_{i=1}^{d}\left[\sum_{j}^{i}(x_i)\right]^2$	$[-65, 65]$	0
F_8	$(x_1 - 1)^2 + \sum_{i=2}^{d} i(2x_i^2 - x_{i-1})^2$	$[-10, 10]$	0
F_9	$\sum_{i=1}^{d} x_i + \prod_{i=1}^{d} x_i$	$[-10, 10]$	0
F_{10}	$max_i\{\lvert x_i\rvert, 1 \le i \le D\}$	$[-100,100]$	0
F_{11}	$\sum_{i=1}^{D}\lvert x_i\sin(x_i) + 0.1x_i\rvert$	$[-10,10]$	0
F_{12}	$\sum_{i=1}^{D}\sin(x_i) * (\sin(\frac{ix_i^2}{\pi}))^{20}$	$[0, \pi]$	0
F_{13}	$(\sum_{i=1}^{D} x_i^2)^2$	$[-100,100]$	0
F_{14}	$\left(\sum_{i=1}^{D}\lvert x_i\rvert\right)\left(e^{\sum_{i=1}^{D} - \sin(x_i^2)}\right)$	[-2π, 2π]	0
F_{15}	$\left[\sum_{i=1}^{D}\sin 2(x_i) - e^{-\sum_{i=1}^{D}x_i^2}\right](e^{-\sum_{i=1}^{D}\sin 2\sqrt{\lvert x_i\rvert}})$	$[-10,10]$	0

6. Experimental results and discussion

The performance of the proposed SP-MO algorithm was evaluated on a set of uni-modal and multimodal benchmark mathematical functions of IEEE CEC [81]. Besides, the optimized global alignment technique (G-Aligner) using SP-MO was tested on real biological protein sequences (Homo sapiens proteins) were gathered from NCBI to validate its performance for measuring the similarity between pair of sequences. The founded similarity of scores is compared with that founded by the exact NW global alignment [33] to validate the quality of the solution of G-Aligner. Besides, the similarity of COVID-19 protein was measured with the other 13 viruses to validate the performance of G-Aligner based on SP-MO. The results of the SP-MO algorithm were compared with other recent development of SCA in the literature such as m-SCA, ISCA, SCA-DE, SCA-PSO and ASCA-PSO, SCA-GWO, and CSCF.

6.1. Evaluation of SP-MO's performance on mathematical benchmark functions

In this section, the developed SP-MO algorithm was tested on 15 mathematical benchmark functions (unimodal and multimodal) that are described in Table 1. Table 2 shows the average optimum results (30 independent runs) for the proposed SP-MO algorithm versus other algorithms in the literature to find the optimum value of the mathematical functions in Table 1.

SP-MO has superiority over other algorithms for all functions by finding the minimum fitness of the functions near the optimum. While SCA-DE, ASCA-PSO, SCA-PSO, SCA-GWO, and CSCF achieved near the optimum for some functions with lower accuracy than that of SP-MO. SCA-DE provided poor results for functions (F_1, F_2, F_6, $F_{9,}$F$_{10}$, F_{11}, F_{13} and F_{14}) and ASCA-PSO is poor for functions (F_1, F_2, F_6, $F_{9,}$F$_{10}$, F_{11}, F_{13} and F_{14}). SCA-PSO provided poor results for the functions (F_1, F_2, F_3, $F_{4,}$F$_6$, F_7, F_9, F_{11}, F_{13}, and F_{14}) and SCA-GWO is poor for all the function except (F_7, F_8, F_{10}, $F_{12,}$ and F_{15}).

For the rest of the algorithms, it provided poor results in all functions approximately which reflects the powerful of the proposed method (SP-MO). The addition of the mutation operator and avoiding trapping in the local minima with aiding of applying the opposition to the solutions become in nearby aid for efficient exploration in the search space.

Besides, the synergy of exploration and exploitation of the search space using SCA and PSO enhance the provided quality of solutions.

Table 3 shows the standard deviation of the results that were provided using SP-MO in comparison with other algorithms in the literature. As shown in Table 3 the proposed method SP-MO provided the lowest standard deviation while other algorithms produce higher standard deviation. That reflects the robustness of SP-MO and shows the significance of using mutation operator and opposition to intensify the search space more accurately.

So, evaluating the performance of SP-MO for finding the optimal value of benchmark mathematical functions concludes its superiority over other algorithms in the literature in terms of quality of solution and robustness.

6.2. Estimating the similarity of biological sequences using G-Aligner based on SP-MO algorithm

In this section, the performance of G-Aligner based on SP-MO was evaluated in measuring the similarity of biological sequences (set of Homo sapiens) and finding the similarity of COVID-19 virus with other viruses. NW alignment algorithm provides the accurate alignment score (similarity) hence it was used as a reference in comparison.

The performance of G-Aligner based on different techniques in terms of execution times was tested on a set of biological sequences each pair have a product of its lengths ranges from 100,000 to 9,000,000. The G-Aligner was implemented on MATLAB software toolkit on a computer machine that has a processor Core I3 (3.14 GHz for each processor) and 4 GB RAM. The number of iterations of the meta-heuristic techniques is 200, the search agents of different techniques were assigned as in Table 4 according to the product of lengths of the aligned sequences. Table 5 shows the setting of parameters of the meta-heuristic techniques used for implementing the G-Aligner.

Fig. 5 shows the execution time of NW global alignment against G-Aligner based on meta-heuristic techniques for aligning pair of biological sequences have a product of lengths of their sequences ranges from 100000 to 9000000. As shown in figure G-Aligner based on various meta-heuristic techniques consumes a smaller execution time than that of NW global alignment especially for longer sequences. This test verifies the significant computational time improvement of G-Aligner over NW global alignment. However, SP-MO consumes the greater execution time that is due to its big-time complexity than other meta-heuristic algorithms were used in the test which represents one of its main limitations.

6.2.1. Measuring the similarity Homo sapiens proteins using G-Aligner

The performance of G-Aligner based on the various meta-heuristic technique for measuring the similarity score of pair of sequences are evaluated using a set of pair of biological protein sequences (Homo sapiens proteins) gathered from NCBI. The experimental results were executed based on the parameter setting of Table 2 while Eq. (2) was used as the fitness function to score the similarity (+1 used for similar residues and otherwise is 0).

G-Aligner was implemented using the proposed SP-MO and the results were compared versus the results of standard SCA, PSO, ISCA, m-SCA, SCA-PSO and SCA-DE, ASCA-PSO, SCA-GWO,

Table 2
The average results for all algorithms for 30 independent runs.

F	SP-MO	m-SCA	SCA	PSO	ISCA	SCA-DE	ASCA-PSO	SCA-PSO	SCA-GWO	CSCF
F_1	3.20E−07	0.37	2.3	1.21	1.20	1.50E−14	2.30E−15	0.62	2.60E−13	0.034
F_2	2.30E−06	0.68	82.5	11.69	3.20	0.09	0.12	2.10	0.21	3.67
F_3	8.50E−17	0.13	0.28	0.26	0.37	2.10E−4	0.002	0.008	4.50E−05	0.00243
F_4	0	0.80	1.10	0.7	0.34	0.007	4.20E−3	0.06	3.01E−03	8.98E−03
F_5	5.20E−68	7.00E−06	1.85E−14	0.08	0.07	1.90E−16	7.30E−71	9.10E−20	8.50E−12	10.4E−03
F_6	3.40E−15	0.71	12.1	5.20	7.89	7.60E−08	6.20E−06	1.20	0.98	2.45
F_7	4.5E−128	1.52	90	120.8	3.20	2.30E−16	4.10E−68	5.73	6.70E−10	3.95
F_8	4.80E−14	4.20E−02	2.40	18.71	0.30	3.48E−04	8.52E−06	1.77	6.54E−09	1.24
F_9	2.20E−65	0.08	5.2	3.50	0.09	2.50E−04	4.10E−10	0.06	1.03	2.34
F_{10}	1.6E−137	0.13	2.76	2.63	0.40	4.50E−26	0.004	0.12	3.45E−03	0.245
F_{11}	2.19E−04	3.40	5.13	4.31	1.03	1.20	3.25	1.4	0.97	1.89
F_{12}	0	2.30	4.93	6.20	1.42	4.60E−03	3.60E−02	0.0008	0.004	4.9E−03
F_{13}	0	1.30	4.20	3.65	0.43	3.65E−05	2.10E−02	0.05	8.64E−06	1.045
F_{14}	2.07E−06	0.70	6.70	3.56	0.93	1.30E−03	0.47	1.03	0.067	2.53
F_{15}	3.67E−13	3.20	7.23	4.23	2.10	2.30E−04	0.004	0.03	3.7E−03	0.00078

Table 3
Standard Deviation of SP-MO versus comparative algorithms.

F	SP-MO	m-SCA	SCA	PSO	ISCA	SCA-DE	ASCA-PSO	SCA-PSO	SCA-GWO	CSCF
F_1	3.20E−07	0.757	3.41	1.21	0.97	0.62	0.65	0.37	0.068	0.236
F_2	0	2.09	17.6	11.69	2.68	1.36	3.64	8.68	0.543	0.326
F_3	8.84E−22	0.234	0.37	0.26	0.30	0.03	0	0.006	0.017	0.085
F_4	1.49E−28	0.702	0.34	0.97	0.90	0.06	0	0.12	0.039	0.466
F_5	1.85E−14	0.624	1.27	0.08	0.80	0	3.12E−12	7.00E−06	0.0234	0.443
F_6	0	0.554	14.4	3.42	0.71	2.34	2.63	0.71	2.06	0.026
F_7	0.003	0.016	0.48	2.14	0.02	6.10E−09	6.20E−06	0.02	6.8E−10	0.019
F_8	0	0.001	2.03	4.20	1.80E−03	4.50E−06	2.80E−13	1.80E−16	4.3E−06	0
F_9	0	2.668	18.49	120.8	3.42	0.08	11.07	0.08	0.018	3.056
F_{10}	0	0	5.60	18.71	3.05E−04	1.77	0.29	3.05E−02	1.371	0
F_{11}	6.38E−08	0.062	18.18	0.88	0.08	0.06	0.18	0.08	0.020	0.034
F_{12}	0	0	1.20	4.49E−02	4.40E−04	7.10E−08	9.20E−09	4.40E−04	7.0E−08	0
F_{13}	2.65E−07	0.881	7.78	2.63	1.13	0.12	0.48	0.73	0.021	0.075
F_{14}	0.0052	2.652	5.13	4.31	3.40	1.4	0.02	2.31	0.699	2.136
F_{15}	1.90E−16	1.638	3.54	1.23	2.10	8.30E−07	4.36E−8	1.00E−03	2.1E−07	1.284

Table 4
The number of search agents used for G-Aligner according to each technique.

m × n	Search agents
100000	10
150000	20
400000	30
700000	50
900000	80
1200000	100
1700000	130
2000000	150
2500000	180
3000000	200
3500000	220
4500000	250
6000000	300
7000000	350
8000000	380
9000000	400

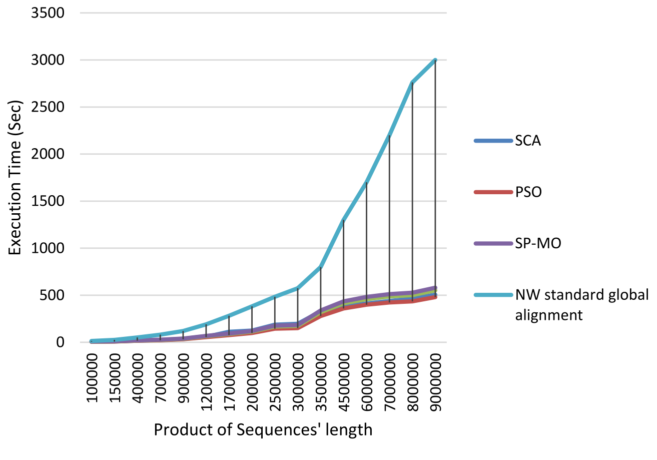


Fig. 5. Execution time of NW-Alignment versus G-Aligner based on different metaheuristic techniques.

and CSCF) and the results of NW global alignment algorithms were used as a reference to validate the results of G-Aligner.

Table 6 presents the similarity scores measured by G-Aligner using SP-MO and other techniques. The first column shows the protein ID of all biological sequences of data used in the test. G-Aligner based on SP-MO provided the highest score for all pairs of comparisons with an average score of 75% of that provided by the exact global alignment algorithm (NW) which proves its powerful capability over all other algorithms in the comparison. G-Aligner based on SCA and PSO achieved approximate results where the average similarity scores are 39% for SCA and 38% for PSO relative to that measured by NW global alignment. ISCA and m-SCA achieve average similarity scores of 42% which have no enhancement of SCA for G-Aligner. However, m-SCA produces a smaller standard deviation of the experimental results than ISCA and SCA as shown in Table 7.

The G-Aligner based on hybrid techniques SCA-DE and SCA-PSO achieved average similarity scores of 46% and 49% in order and the two techniques have smaller standard deviation than

Table 5
The setting values for the parameters of G-Aligner based on a different technique.

	Algorithm	Parameter	Value
NW Alignment		Match g_e g_o	+1.0 −0.5 −1.0
G-Aligner	SCA m-SCA ISCA	a	20
	PSO	Inertia Coefficient Local coefficient (C1) Global coefficient (C2)	0.2 1.5 1.5
	SP-MO ASCA-PSO SCA-PSO	Inertia Coefficient Local coefficient (C1) Global coefficient (C2) A	0.2 1.5 1.5 20
	SCA-DE	Beta P_{CR} A	0.3 0.3 20

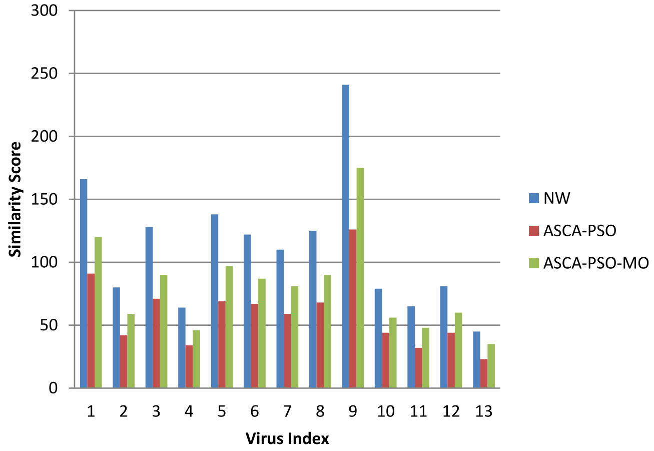


Fig. 6. Similarity scores of aligning COVID-19 against 13 viruses based on G-Aligner using SP-MO versus ASCA-PSO and NW algorithm [33].

SCA. The hybrid of SCA and PSO (ASCA-PSO) has a great enhancement by over the performance of SCA where it achieves 51.5% of the exact similarity scores founded by NW global alignment. ASCA-PSO has a standard deviation results approximate to SCA-DE and SCA-PSO. SCA-GWO provided an average score of 45% while CSCF had an average score of 44%.

The proposed technique SP-MO for G-Aligner achieved the highest similarity scores with 75% of the exact similarity scores founded by NW global alignment with the lowest standard deviation among all techniques as shown in Table 7. That verifies the superiority of G-Aligner based on SP-MO over the other algorithms in the literature for finding the highest similarity score near to the exact one as possible in small time.

6.2.2. Measuring similarity of COVID-19 versus other viruses using G-Aligner

The G-Aligner performance was validated by measuring the similarity of the COVID-19 virus with other viruses where all the protein of viruses gathered from NCBI. The viruses are (1) Middle East respiratory syndrome coronavirus (MERS-CoV), (2) Malaria, (3) Hepatitis C, (4) Hepatitis B, (5) Epstein–Barr virus (HHV-4), (6) Influenza A, (7) Influenza B, (8) Simian immunodeficiency virus, (9) Trachea Infections, (10) Severe acute respiratory syndrome coronavirus (SARS-CoV), (11) Dengue virus, (12) Cowbox virus and (13) Alveolar proteinosis.

Fig. 6 presents the comparisons of measuring the similarity of COVID-19 virus with viruses using G-Aligner based on SP-MO, ASCA-PSO, and NW global alignment as a reference. The horizontal line represents the index of the virus while the vertical one represents the scale of similarity score. As shown in the figure G-Aligner proposes a similarity score using SP-MO higher than that of ASCA-PSO and is 75% of that measured by NW global alignment.

Fig. 7 shows the comparison of similarity scores between COVID-19 and other viruses using G-Aligner based on SP-MO and the enhancement version of SCA in the literature review. As shown in Fig. 7(a), m-SCA and ISCA achieved approximate scores little better than that of SCA and PSO and in Fig. 7(b) SCA-DE and SCA-PSO achieved approximate scores but ASCA-PSO beat them. In Fig. 7(c), SP-MO beat SCA-GWO and CSCF with a significant difference.

From Figs. 6 and 7 we can conclude them G-Aligner based on SP-MO has the superiority of measuring the similarity scores with the highest similarity scores of aligning COVID-19 with 13 viruses that are 75% of the score measured by NW global alignment in a reasonable time. SP-MO beat all algorithms in the literature due to its hybrid mechanism which is based on the balance between exploration and exploitation using SCA and PSO in order. Besides, the mutation and opposition operator enhance the exploration of the search space especially for sequences with huge lengths. Besides, SP-MO has the advantage of avoidance the trapping in local optima where there is a condition if the solutions become nearby then apply the opposition operator to diverse the solution.

The main advantages of G-Aligner based on SP-MO as follows:

1- Measuring the similarity score of pair of biological sequences with a reasonable percentage of that measured by NW global alignment algorithm (the exact ones) in very small time especially with sequences with huge length at low cost.
2- It can work offline or online and with any scoring weight for measuring similarity.
3- It is easy to develop G-Aligner in the future by replacing the meta-heuristic technique to test its performance and develop it.

The main limitation of the proposed method (SP-MO) as follows:

1- It consumes execution time more than that of the ASCA-PSO algorithm.
2- Implementing G-Aligner, provided a similarity score of 75% of the exact result that measured by NW global alignment which needs more enhancement.

Table 6
The average similarity scores using G-Aligner based on meta-heuristic techniques versus exact scores of NW global alignment.

Protein ID (length)		NW	SCA	PSO	ISCA	m-SCA	SCA-DE	SCA-PSO	ASCA-PSO	SCA-GWO	CSCF	SP-MO
1	Q08AH3 Q9ULC5	94	36	34	40	41	44	46	39	38	35	69
2	P18089 Q6P093	53	22	25	24	23	25	25	22	21	18	39
3	Q9Y2D8 Q5TYW2	96	36	34	39	41	43	46	37	39	36	70
4	Q9UBJ2 Q8NE71	107	41	38	41	46	50	50	46	47	44	77
5	Q9H172 Q9H222	131	50	47	52	58	59	63	51	53	50	98
6	Q12979 Q96P50	129	53	49	57	58	58	65	58	57	54	90
7	Q12979 Q15027	125	46	41	53	57	59	62	55	52	49	98
8	Q9UG63 O95870	72	28	29	32	30	33	34	28	31	28	56
9	Q8WWZ7 Q96GR2	93	34	33	40	39	42	45	38	38	35	66
10	O95870 Q6H8Q1	83	33	35	38	34	37	40	34	31	28	60
11	O95342 Q96J66	192	74	69	78	80	83	95	79	77	74	138
12	Q8IUA7 Q6H8Q1	209	75	76	81	91	95	104	92	91	87	165
13	P55198 Q96J66	126	45	40	59	53	59	60	54	54	50	97
14	Q8NFM4 Q9BZC7	156	59	57	68	68	68	78	64	64	60	113
15	Q9UKV3 Q07912	171	63	60	73	70	75	86	71	71	67	124
16	A8K2U0 Q8NFM4	172	60	55	67	72	81	87	79	80	76	128
17	O60706 Q9UKV3	158	57	55	65	69	74	80	70	74	70	115
18	O43306 Q6IQ32	140	53	49	54	59	61	70	56	60	56	106
19	Q7Z5R6 Q8N961	27	13	11	14	12	13	14	8	12	8	20
20	A0PJZ0 Q96IX9	26	39	39	58	50	49	48	46	46	42	80
21	Q96IX9 P86434	20	8	8	10	10	10	11	10	7	3	15
22	Q96IU4 Q969K4	37	16	14	19	19	19	20	14	14	10	27
23	P14060 Q7L8J4	50	20	20	25	26	26	25	22	23	18	38
24	J3QRE5 H7C0G5	29	11	12	14	15	15	16	12	15	10	22
25	P04229 P13761	245	94	90	115	127	130	121	122	126	121	183
26	P14060 Q7L8J4	157	63	57	76	82	78	79	72	75	70	116
27	Q8R4X Q8VD53	32	14	12	16	17	16	17	11	12	7	23
28	P68510 P63101	148	61	58	68	73	72	79	69	68	63	105
29	A0PJZ0 Q96IX9	26	11	11	12	13	14	14	11	11	9	20

3- It was tested on real biological sequences that have a product of lengths up to 9,000,000 which need to increase the length of sequences and test its performance to develop it.

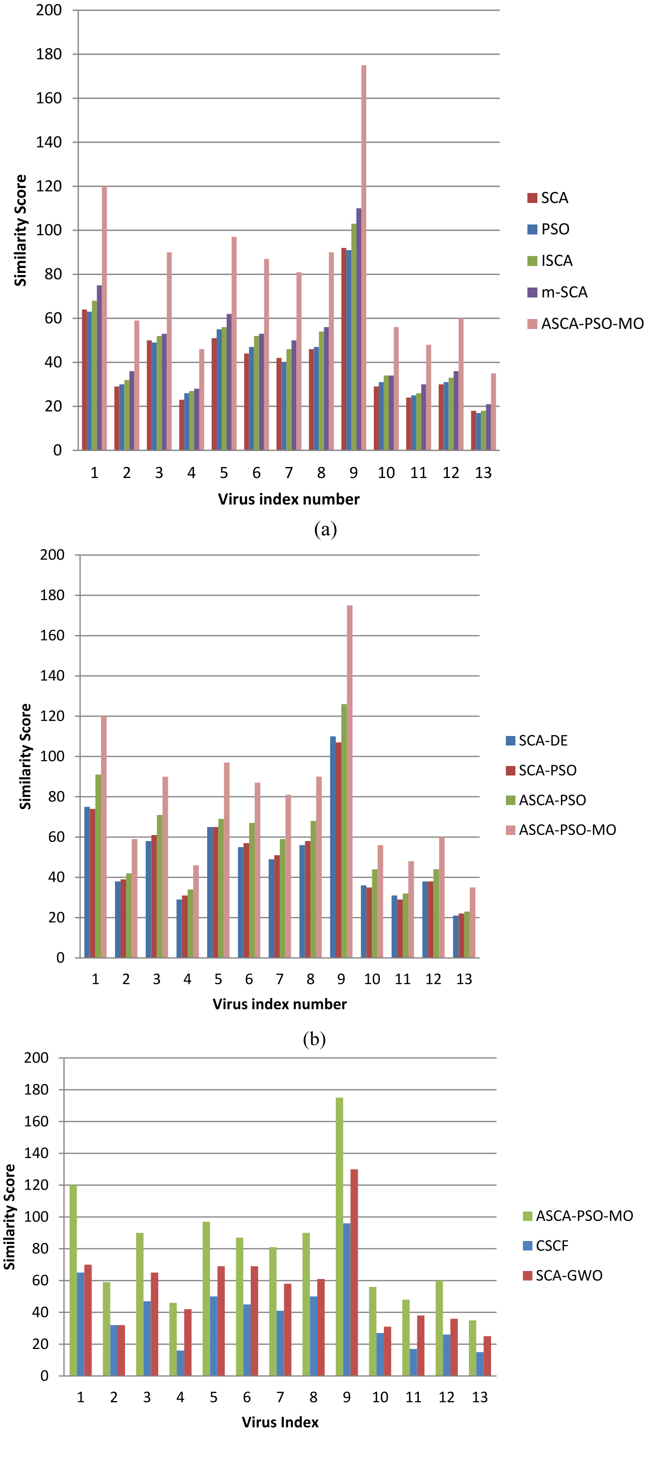
7. Conclusion

This work proposed an accelerated global alignment technique (G-Aligner) based on meta-heuristic algorithms to measure the

Table 7
The standard deviation of G-Aligner based on different meta-heuristic for a 20 independent run.

Protein ID (length)		SCA	PSO	ISCA	m-SCA	SCA-DE	SCA-PSO	ASCA-PSO	SCA-GWO	CSCF	SP-MO
1	Q08AH3 Q9ULC5	2.33	3.57	2.5	0.9	0.91	0.3	0.57	1.11	1.01	0.61
2	P18089 Q6P093	1.84	2.01	1.41	0.75	0.63	0.26	1.58	0.83	0.92	0.80
3	Q9Y2D8 Q5TYW2	2.69	1.81	1.27	1.01	0.91	0.59	0.78	1.11	1.16	0.37
4	Q9UBJ2 Q8NE71	2.46	2.75	1.93	0.94	0.4	1.06	0.88	0.6	0.57	0.24
5	Q9H172 Q9H222	3.17	4.53	3.17	1.15	0.74	0.85	2.42	0.94	1	0.8
6	Q12979 Q96P50	1.64	3.97	2.78	0.69	0.48	0.07	0.7	0.68	0.83	0.44
7	Q12979 Q15027	2.46	2.1	1.47	0.94	0.75	1.39	1.22	1.09	1.86	0.55
8	Q9UG63 O95870	2.27	2.31	1.62	0.88	0.42	1.01	1.36	0.76	1.09	0.63
9	Q8WWZ7 Q96GR2	2.37	1.63	1.14	0.91	1.1	0.88	1.04	1.44	1.78	0.4
10	O95870 Q6H8Q1	1.96	2.95	2.07	0.79	0.47	0.8	0.89	0.81	1.12	0.74
11	O95342 Q96J66	4.5	9.27	6.49	1.55	0.82	2.42	2.66	1.16	1.29	0.91
12	Q8IUA7 Q6H8Q1	2.37	3.17	2.22	0.91	1	0.72	4.05	1.34	1.22	0.23
13	P55198 Q96J66	2.72	1.23	0.86	1.01	0.61	1.49	0.94	0.95	1.7	0.4
14	Q8NFM4 Q9BZC7	3.77	1.4	0.98	1.33	0.78	0.37	0.77	1.12	0.99	0.48
15	Q9UKV3 Q07912	1.07	2.4	1.68	0.52	0.43	0.02	1.58	0.77	1.4	0.72
16	A8K2U0 Q8NFM4	5.96	2.88	2.01	1.99	1.86	4.04	1.31	2.2	2.76	0.84
17	O60706 Q9UKV3	1.84	1.84	1.29	0.75	0.4	0.42	0.41	0.74	0.67	0.95
18	O43306 Q6IQ32	2	1.96	1.37	0.8	0.53	0.45	1.55	0.87	1.01	0.66
19	Q7Z5R6 Q8N961	1.4	1.4	0.98	0.62	0.9	0.91	0.95	1.07	1.44	0.36
20	A0PJZ0 Q96IX9	2.25	3.44	2.41	0.88	0.71	0.99	1.85	0.88	0.86	0.61
21	Q96IX9 P86434	2.52	2.64	1.78	0.43	0.21	0.85	0.97	0.38	0.3	0.28
22	Q96IU4 Q969K4	2.31	2.5	1.63	0.72	0.43	0.70	0.94	0.6	0.72	0.25
23	P14060 Q7L8J4	2.58	2.41	1.51	0.73	0.38	0.88	0.84	0.55	1.16	0.16
24	J3QRE5 H7C0G5	2.48	2.28	1.64	0.88	0.25	0.73	0.96	0.42	1.23	0.45
25	P04229 P13761	2.33	2.25	1.81	0.75	0.33	0.90	1.2	0.5	1.43	0.28
26	P14060 Q7L8J4	2.22	2.36	1.65	0.44	0.43	0.92	0.86	0.6	0.75	0.47
27	Q8R4X Q8VD53	2.24	2.34	1.44	0.6	0.43	0.93	0.78	0.6	0.63	0.36
28	P68510 P63101	2.55	2.38	1.71	0.67	0.41	0.84	1.30	0.58	0.73	0.19
29	A0PJZ0 Q96IX9	2.48	2.33	1.64	0.87	0.47	0.94	1.03	0.64	0.97	0.55

similarity score of the pair of biological sequences in a small time at a low cost. The main benefit of G-Aligner it can scan biological databases fastly to filter the highest similarity sequences to a query one with acceptable similarity measurements near to the exact ones. The developed algorithm (SP-MO) was tested on a set of benchmark mathematical functions in comparison with recent related work in the literature. SP-MO algorithm has superiority over the relevant studies in the literature by finding the best



(a)

(b)

(c)

Fig. 7. Similarity scores of aligning COVID-19 against 13 viruses based on G-Aligner using SP-MO versus various stochastic techniques in the literature.

minimum fitness values of all functions with the lowest standard deviation. Besides, G-Aligner based on SP-MO was validated by measuring the similarity of COVID-19 virus with the other 13 viruses. G-Aligner using SP-MO succeeded to measure the similarity with 75% of the exact one but in execution time very smaller than that of the exact global alignment.

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] Y. Chen, Q. Liu, D. Guo, Emerging coronaviruses: genome structure, replication, and pathogenesis, J. Med. Virol. 92 (4) (2020) 418–423.

[2] X.-Y. Ge, et al., Isolation and characterization of a bat SARS-like coronavirus that uses the ACE2 receptor, Nature 503 (7477) (2013) 535–538.

[3] S. Cauchemez, et al., Transmission scenarios for middle east respiratory syndrome coronavirus (MERS-CoV) and how to tell them apart, in: Euro surveillance: bulletin Europeen sur les maladies transmissibles= European communicable disease bulletin, 18, (24) 2013.

[4] M.A. Tipaldi, et al., How to manage the COVID-19 diffusion in the angiography suite: experiences and results of an Italian interventional radiology unit, SciMedicine J. 2 (2020) 1–8.

[5] D. Hanscom, et al., Polyvagal and global cytokine theory of safety and threat Covid-19–Plan B, SciMedicine J. 2 (2020) 9–27.

[6] L.A. Anchordoqui, J.B. Dent, T.J. Weiler, A physics modeling study of COVID-19 transport in air, SciMedicine J. 2 (2020) 83–91.

[7] X. Sun, S. Wandelt, A. Zhang, How did COVID-19 impact air transportation? A first peek through the lens of complex networks, J. Air Transp. Manag. 89 (2020) 101928.

[8] A. Intissar, A mathematical study of a generalized SEIR model of COVID-19, SciMedicine J. 2 (2020) 30–67.

[9] L.A. Anchordoqui, E.M. Chudnovsky, A physicist view of COVID-19 airborne infection through convective airflow in indoor spaces, SciMedicine J. 2 (2020) 68–72.

[10] L. Morawska, et al., How can airborne transmission of COVID-19 indoors be minimised?, Environ. Int. 142 (2020) 105832.

[11] P.K. Sahoo, et al., Is the transmission of novel coronavirus disease (COVID-19) weather dependent?, J. Air Waste Manage. Assoc. (2020) 1–4.

[12] P.K. Sahoo, et al., COVID-19 pandemic: an outlook on its impact on air quality and its association with environmental variables in major cities of Punjab and Chandigarh, India, J. Air Waste Manage. Assoc. (2020) 1–12.

[13] A. Tobías, T. Molina, Is temperature reducing the transmission of COVID-19?, Environ. Res. 186 (2020) 109553.

[14] T. Jamil, et al., No evidence for temperature-dependence of the COVID-19 epidemic, Front. public health 8 (2020) 436.

[15] M. Holtmann, et al., Low ambient temperatures are associated with more rapid spread of COVID-19 in the early phase of the endemic, in: Environ. Res., 2020.

[16] S. Lalmuanawma, J. Hussain, L. Chhakchhuak, Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review, Chaos Solitons Fractals (2020) 110059.

[17] M.A. Al-Qaness, et al., Optimization method for forecasting confirmed cases of COVID-19 in China, J. Clin. Med. 9 (3) (2020) 674.

[18] B. Pirouz, et al., Investigating a serious challenge in the sustainable development process: analysis of confirmed cases of COVID-19 (new type of coronavirus) through a binary classification using artificial intelligence and regression analysis, Sustainability 12 (6) (2020) 2427.

[19] M. Jamshidi, et al., Artificial intelligence and COVID-19: Deep learning approaches for diagnosis and treatment, IEEE Access 8 (2020) 109581-109595.

[20] M. Abdel-Basset, et al., A hybrid COVID-19 detection model using an improved marine predators algorithm and a ranking-based diversity reduction strategy, IEEE Access 8 (2020) 79521–79540.

[21] H. Alabool, et al., Artificial Intelligence Techniques for Containment COVID-19 Pandemic: A Systematic Review, 2020.

[22] F.B. Hamzah, et al., Coronatracker: worldwide COVID-19 outbreak data analysis and prediction, Bull. World Health Organ 1 (2020) 32.

[23] B.B. Hazarika, D. Gupta, Modelling and forecasting of COVID-19 spread using wavelet-coupled random vector functional link networks, Appl. Soft Comput. (2020) 106626.

[24] L. Wynants, et al., Prediction models for diagnosis and prognosis of covid-19 infection: systematic review and critical appraisal, BMJ 369 (2020).

[25] C. Monaghan, et al., Artificial Intelligence for COVID-19 Risk Classification in Kidney Disease: Can Technology Unmask an Unseen Disease? MedRxiv, 2020.

[26] M. Nour, Z. Cömert, K. Polat, A novel medical diagnosis model for COVID-19 infection detection based on deep features and Bayesian optimization, Appl. Soft Comput. (2020) 106580.

[27] G. Marques, D. Agarwal, I. de la Torre Díez, Automated medical diagnosis of COVID-19 through efficientnet convolutional neural network, Appl. Soft Comput. (2020) 106691.

[28] P.S. Sen Gupta, et al., Binding insight of clinically oriented drug famotidine with the identified potential target of SARS-CoV-2, J. Biomol. Struct. Dyn. (2020) 1–7.

[29] R. Kong, et al., COVID-19 docking server: An interactive server for docking small molecules, peptides and antibodies against potential targets of COVID-19, 2020, arXiv preprint arXiv:2003.00163.

[30] E. Lamptey, D. Serwaa, The use of zipline drones technology for COVID-19 samples transportation in Ghana, HighTech Innov. J. 1 (2) (2020) 67–71.

[31] M. Angurala, et al., An internet of things assisted drone based approach to reduce rapid spread of COVID-19, J. Saf. Sci. Resil. 1 (1) (2020) 31–35.

[32] A. Kumar, et al., A drone-based networked system and methods for combating coronavirus disease (COVID-19) pandemic, Future Gener. Comput. Syst. 115 (2020) 1–19.

[33] S.B. Needleman, C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, J. Mol. Biol. 48 (3) (1970) 443–453.

[34] T.F. Smith, M.S. Waterman, Identification of common molecular subsequences, J. Mol. Biol. 147 (1) (1981) 195–197.

[35] N. Ahmed, et al., GASAL2: a GPU accelerated sequence alignment library for high-throughput NGS data, BMC Bioinformatics 20 (1) (2019) 520.

[36] A.H. Mohamed Issa, Ibrahim Ziedan, Ahmed Alzohairy, Maximizing occupancy of GPU for fast scanning biological database using sequence alignment, J. Appl. Sci. Res. 13 (6) (2017).

[37] L. Alawneh, et al., A scalable multiple pairwise protein sequence alignment acceleration using hybrid CPU–GPU approach, Cluster Comput. (2020) 1–12.

[38] D. Sundfeld, et al., Using GPU to accelerate the pairwise structural RNA alignment with base pair probabilities, Concurr. Comput.: Pract. Exper. 32 (10) (2020) e5468.

[39] S. Kasap, K. Benkrid, Y. Liu, Design and implementation of an FPGA-based core for gapped BLAST sequence alignment with the two-hit method, Eng. Lett. 16 (3) (2008).

[40] Y. Liu, et al., An fpga-based web server for high performance biological sequence alignment, in: 2009 NASA/ESA Conference on Adaptive Hardware and Systems, IEEE, 2009.

[41] K. Benkrid, et al., High performance biological pairwise sequence alignment: FPGA versus GPU versus cell BE versus GPP, Int. J. Reconfigurable Comput. 2012 (2012).

[42] K. Benkrid, Y. Liu, A. Benkrid, A highly parameterized and efficient FPGA-based skeleton for pairwise biological sequence alignment, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 17 (4) (2009) 561–570.

[43] R. Chamberlain, et al., Method and Apparatus for Protein Sequence Alignment using FPGA Devices, Google Patents, 2008.

[44] T. Ramdas, G. Egan, A survey of FPGAs for acceleration of high performance computing and their application to computational molecular biology, in: TENCON 2005 2005 IEEE Region 10, IEEE, 2005.

[45] E.-G. Talbi, Metaheuristics: From Design To Implementation, Vol. 74, John Wiley & Sons, 2009.

[46] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133.

[47] Kennedy, Particle swarm optimization, Neural Netw. (1995).

[48] B. Javidy, A. Hatamlou, S. Mirjalili, Ions motion algorithm for solving optimization problems, Appl. Soft Comput. 32 (2015) 72–79.

[49] H. Shareef, A.A. Ibrahim, A.H. Mutlag, Lightning search algorithm, Appl. Soft Comput. 36 (2015) 315–333.

[50] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.

[51] H. Abedinpourshotorban, et al., Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm, Swarm Evol. Comput. 26 (2016) 8–22.

[52] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.

[53] C.-H. Yang, et al., Protein folding prediction in the HP model using ions motion optimization with a greedy algorithm, BioData min. 11 (1) (2018) 17.

[54] X. Zhao, Advances on protein folding simulations based on the lattice HP models with natural computing, Appl. Soft Comput. 8 (2) (2008) 1029–1040.

[55] B. Bošković, J. Brest, Genetic algorithm with advanced mechanisms applied to the protein structure prediction in a hydrophobic-polar model and cubic lattice, Appl. Soft Comput. 45 (2016) 61–70.

[56] A. Morshedian, J. Razmara, S. Lotfi, A novel approach for protein structure prediction based on an estimation of distribution algorithm, Soft Comput. 23 (13) (2019) 4777–4788.

[57] A.E. Márquez-Chamorro, et al., Soft computing methods for the prediction of protein tertiary structures: A survey, Appl. Soft Comput. 35 (2015) 398–410.

[58] H. Pérez-Sánchez, G. Cano, J. García-Rodríguez, Improving drug discovery using hybrid softcomputing methods, Appl. Soft Comput. 20 (2014) 119–126.

[59] P.F. Leonhart, et al., A biased random key genetic algorithm for the protein–ligand docking problem, Soft Comput. 23 (12) (2019) 4155–4176.

[60] M. Issa, et al., ASCA-PSO: Adaptive sine cosine optimization algorithm integrated with particle swarm for pairwise local sequence alignment, Expert Syst. Appl. 99 (2018) 56–70.

[61] M. Muppalaneni, M. Ma, S. Gurumoorthy, Soft Computing and Medical Bioinformatics, Springer, 2019.

[62] A.F. Ali, A.-E. Hassanien, A survey of metaheuristics methods for bioinformatics applications, in: Applications of Intelligent Optimization in Biology and Medicine, Springer, 2016, pp. 23–46.

[63] M.Abd. Elaziz, D. Oliva, S. Xiong, An improved opposition-based sine cosine algorithm for global optimization, Expert Syst. Appl. 90 (2017) 484–500.

[64] S. Gupta, K. Deep, A hybrid self-adaptive sine cosine algorithm with opposition based learning, Expert Syst. Appl. 119 (2019) 210–230.

[65] H. Nenavath, R.K. Jatoth, Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking, Appl. Soft Comput. 62 (2018) 1019–1043.

[66] H. Nenavath, R.K. Jatoth, S. Das, A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking, Swarm Evol. Comput. (2018).

[67] B.A. Hassan, CSCF: a chaotic sine cosine firefly algorithm for practical application problems, Neural Comput. Appl. (2020) 1–20.

[68] S. Gupta, et al., Sine cosine grey wolf optimizer to solve engineering design problems, Eng. Comput. (2020) 1–27.

[69] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.

[70] T.H. Cormen, Introduction To Algorithms, MIT press, 2009.

[71] J. Xiong, Essential Bioinformatics, Cambridge University Press, 2006.

[72] D.W. Mount, Comparison of the PAM and BLOSUM amino acid substitution matrices, Cold Spring Harbor Protoc. 2008 (6) (2008) p. pdb. ip59.

[73] R. Salgotra, U. Singh, Application of mutation operators to flower pollination algorithm, Expert Syst. Appl. 79 (2017) 112–129.

[74] Q. Zhang, et al., Chaos-induced and mutation-driven schemes boosting salp chains-inspired optimizers, IEEE Access 7 (2019) 31243–31261.

[75] H. Jia, et al., Dynamic harris hawks optimization with mutation mechanism for satellite image segmentation, Remote sens. 11 (12) (2019) 1421.

[76] Y. Xu, et al., Enhanced moth-flame optimizer with mutation strategy for global optimization, Inform. Sci. 492 (2019) 181–203.

[77] H. Wang, et al., Opposition-based particle swarm algorithm with Cauchy mutation, in: 2007 IEEE Congress on Evolutionary Computation, IEEE, 2007.

[78] X. Zhang, et al., Gaussian mutational chaotic fruit fly-built optimization and feature selection, Expert Syst. Appl. 141 (2020) 112976.

[79] G.-G. Wang, et al., Opposition-based krill herd algorithm with Cauchy mutation and position clamping, Neurocomputing 177 (2016) 147–157.

[80] S. Sapre, S. Mini, Opposition-based moth flame optimization with Cauchy mutation and evolutionary boundary constraint handling for global optimization, Soft Comput. 23 (15) (2019) 6023–6041.

[81] M. Jamil, X.-S. Yang, A literature survey of benchmark functions for global optimisation problems, Int. J. Math. Model. Numer. Optimis. 4 (2) (2013) 150–194.