American Journal of Artificial Intelligence
2021; 5(2): 56-65
http://www.sciencepublishinggroup.com/j/ajai
doi: 10.11648/j.ajai.20210502.12
ISSN: 2639-9717 (Print); ISSN: 2639-9733 (Online)

Artificial Corona-Inspired Optimization Algorithm: Theoretical Foundations, Analysis, and Applications

Alia Youssef Gebreel

Operations Research, Faculty of Graduate Studies for Statistical Research, Cairo University, Cairo, Egypt

Email address:
y_alia400@yahoo.com

To cite this article:
Alia Youssef Gebreel. Artificial Corona-Inspired Optimization Algorithm: Theoretical Foundations, Analysis, and Applications. *American Journal of Artificial Intelligence*. Vol. 5, No. 2, 2021, pp. 56-65. doi: 10.11648/j.ajai.20210502.12

Received: July 28, 2021; **Accepted**: August 7, 2021; **Published**: August 27, 2021

Abstract: One of the important parts of computer science is Artificial Intelligence (AI). It deals with the development of machines that can take decisions like humans on their own. Currently, AI can solve many difficult real-world problems because it works much better and faster than humans. Researchers of operations research also are turning their heads towards AI instead of traditional systems. Meanwhile, there are several AI models to solve mathematical optimization problems. They depend heavily on a random search, but many of their solutions have been efficient at finding absolute optimum. This means that it is necessary to choose another optimization model to get quite the optimum value. This paper introduces an artificially intelligent algorithm in order to find the optimal solution for a given computational problem that minimizes or maximizes a particular function. It is inspired by the corona virus that spreads throughout the world and infects healthy people. Its structure simulates the stages of virus transmission and treatment. Because the starting point is so important for converging to the global optimum, corona virus approach has guided researchers to select the starting point and parameters. Actually, this point depends on three real numbers as the corona virus affects three main parts of the human body (nose, throat, respiratory). The proposed algorithm has been found to be an optimal key to different applications. It doesn't require any derivative information and it is simple in implementation with few parameters setting. Finally, some numerical examples are presented to illustrate the algorithm studied here. The computational results show that it has high performance in finding an optimal solution within reasonable time.

Keywords: Artificial Intelligent Algorithms, Corona Virus (CV), Optimal Solution

1. Introduction

In December 2019, everyone was interested in corona virus that has spread all over the world. It is an infectious acute respiratory disease and its complications may lead to death. This current disease is first reported in Wuhan, China, and transmitted from bats to humans. Its name is derived from a relation to the proteins that give it its coronary shape as shown in Figure 1.

The corona virus is also named COVID-19 that is an emerging contagious disease caused by the SARS-CoV-2 virus. It infects people of all ages, especially elder people and persons with chronic diseases [1-5].

It could be transmitted from human to human by inhalation or contact with infected droplets and the incubation period ranges from 2 to 14 days. In addition, there are opinions say that individuals who remain asymptomatic could transmit the virus [3, 5].

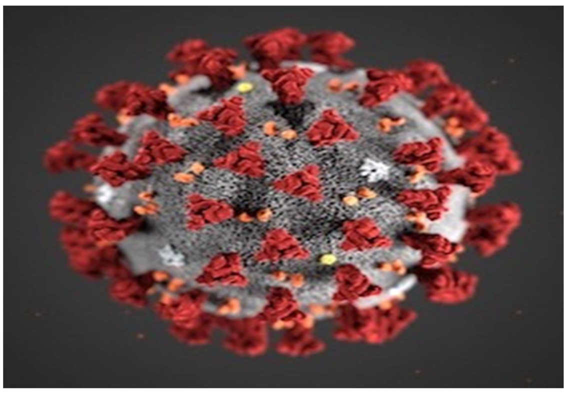

Figure 1. *A graphical representation of the ultra-structural morphology of corona virus.*

The stages of corona virus (COVID 19) and its treatment are sketched basically as shown in the following *Figure 2*.

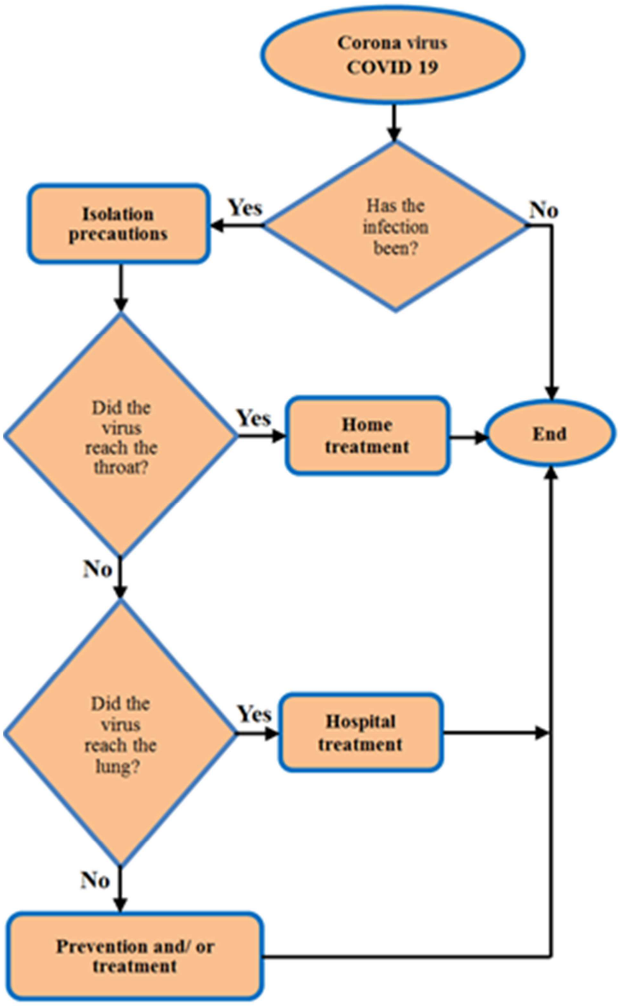

Figure 2. *A diagram representation of the stages of corona virus (Covid 19) and its treatment.*

Generally, hospitalized patients are classified into two categories, the general COVID-19 which has been defined according to the following criteria: obvious relief of respiratory symptoms (for example, cough, chest distress, and shortness of breath) after treatment, maintaining normal body temperature for more than three days without the use of corticosteroids or antipyretics, improving radiological abnormalities in the chest scanner or X-rays after treatment, a hospital stay of fewer than 10 days. Otherwise, it was classified as COVID-19 refractory [3].

Hence, health education on knowledge for disease prevention and control is also important to control and reduce the corona virus infection rate. Treatment is essentially supportive; the role of antiviral agents is yet to be established. Prevention entails home isolation of suspected cases and those with mild illnesses and strict infection control measures at hospitals that include contact and droplet precautions [3, 5].

To simulate this disease for solving mathematical programming problems, it is known that traditional methods have been replaced by AI algorithms to solve such problems. They are the fastest-growing field of computer science and technology. These modern optimization algorithms have been successfully employed in different applications. Also, they are called heuristic or meta-heuristic algorithms. Where, heuristic search refers to a search strategy that attempts to optimize a problem by iteratively improving the solution based on a given heuristic function or a cost measure. Most of the existing meta-heuristic algorithms imitate natural or scientific phenomena, e.g. evolution in evolutionary algorithms, functions of the brain in neural networks, physical annealing of metal plates in simulated annealing, human memory in tabu search, human's food in bacterial foraging, and the music improvisation process in harmony search. Following a similar trend, a new artificial intelligent algorithm can be conceptualized from the corona virus activity. Where, a doctor always desires to provide the best adequate treatment to his patients, which can be obtained by numerous practices [6-9].

Some studies have been published for applications of corona virus in operations research such as:

The authors of the paper [2] introduced a brief overview of the meta-heuristic COVID-19 method concerning its applications for electricity demand time series forecasting and hybridize deep learning models. It found that corona virus optimization algorithm is easily transformed into a multi-virus meta-heuristic, in which different corona virus strains search for the best solution in a collaborative way.

But, *Ahmed et al.* in paper [1] presented an improved hybrid classification approach for COVID-19 images by combining the strengths of convolutional neural networks to extract features and a swarm-based feature selection algorithm for selecting the most relevant features. The results showed that this approach has better performances in both classification accuracy and the number of extracted features that have a positive effect on resource consumption and storage efficiency.

The natural-inspired human-based meta-heuristic optimization algorithm is proposed by *Mohammed Al-Betar et al.* [10], which is inspired by the herd immunity strategy as a way to tackle the spreading of corona virus pandemics (COVID-19).

Furthermore, *G Dhiman et al.* in the article [11] illustrated the multi-objective optimization and a deep-learning methodology for the detection of infected corona virus patients with X-rays. They concluded that the performance of their study is better than the other ten competitor models based on the statistical research to select the best classification pattern.

In general, all intelligent optimization algorithms are started with random point(s) and their result shows time-consuming process. For this reason, the research proposes an algorithm to find the optimal solution for mathematical programming problems. It relies on the deterministic value for each decision variable that is selected from three real numbers; rather than random numbers.

The remaining sections are organized as follows. Section 2 briefly outlines some types of intelligent optimization algorithms. Followed by, section 3 describes in detail the proposed algorithm for optimization problems. Section 4 presents some numerical examples to solve linear and nonlinear optimization problems with their obtained results. Finally, section 5 concludes the paper and provides future work.

2. Some of Intelligent Optimization Algorithms

There are several intelligent optimization techniques to generate acceptable solutions, even though they cannot guarantee optimality. They can incorporate problem-specific knowledge to improve the quality of the solutions. This section reviews some of them. These models include genetic algorithm, hill-climbing, simulated annealing, tabu search, particle swarm, ant colony, neural network, bacterial foraging, and harmony search algorithm.

2.1. Genetic Algorithm

Genetic Algorithm (GA) imitates the mechanics of natural selection and evaluation. It involves the following stages to solve any optimization problem:

1) Representation of feasible solution to a problem as a population of randomly generated individuals and happens in generations.
2) Evaluation of the population using fitness function.
3) Generation of new population using the genetic operators (crossover and mutation).
4) Selection of new population from the current population [12, 14].

2.2. Hill-climbing

Hill-climbing is a mathematical optimization technique which belongs to the family of local search. It is an iterative algorithm that starts with an arbitrary solution to a problem, then attempts to find a better solution by making an incremental change to the solution [15-16].

2.3. Simulated Annealing

Simulated Annealing (SA) is a local search based on meta-heuristic technique that proposed by Kirkpatrick et al. in 1983. It simulates the process of heating and cooling metals. SA technique differs from other search methods in accepting inferior solutions with certain probability. This is calculated using the Boltzmann probability as:

$$P_B=e^{-\varphi/T} \tag{1}$$

Where, T is the temperature that is decreasing according to some cooling schedule during the search process, and φ is the difference between the best solution's fitness value and the generated trial's fitness value. In this way, it attempts to avoid getting stuck in the local optimal solutions [13, 17, 18].

2.4. Tabu Search

Tabu Search (TS) is a local heuristic search procedure. It is guided by using an adaptive or flexible memory structure. The information is stored in a Tabu list that contains all moves. The quality of each solution is measured by an objective function. In order to find the best admissible move in a neighborhood, the move that has the shortest tabu tenure is selected. If the TS is not converging, the search is reset randomly [14, 19].

2.5. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is inspired by the behavior of bird flocking and fish schooling. It is one of the evolutionary computation techniques. So, it likes GA in starting with an initial population of random solutions. But, PSO is unlike other algorithms in which each potential solution (called a particle) is also assigned a randomized velocity and then flown through the problem hyperspace [19].

2.6. Ant Colony

Ant Colony (AC) is inspired by the behavior of ant colonies. A number of ants cooperate with each other by sharing their experiences in finding a solution. In real ant colonies, this shared memory is represented by the deposition of pheromones on the trail as ants forage for food. If the trail is a "successful" one (in terms of increasing the food supply), its usage will be increased, otherwise "evaporation" takes place and the trail is eventually abandoned. In artificial systems, this effect is represented by a modification to the probability of choosing the next element of the solution [6].

2.7. Neural Network

Neural Network (NN) is modeled on the mechanism of the human brain. This network has the capacity of learning from examples, memorizes, and creates relationships amongst data. It works based on three main stages: the designing stage, training stage, and testing stage [20].

2.8. Bacterial Foraging Algorithm

The bacterial Foraging Algorithm (BFA) is proposed by Kevin M. Passino in 2000. It simulates the life cycles and the foraging behaviors of Escherichia coli bacteria that live in the human intestine to solve the optimization problems. There are three processes on a population of simulated cells: chemotaxis, reproduction, and elimination-dispersal. It is unfortunate that BFO gives poor performance for multimodal and high dimensional functions as compared to other optimization techniques like GA and PSO [12, 21].

2.9. Harmony Search Algorithm

Harmony Search Algorithm (HSA) is developed by Zong Woo Geem et al. in 2001. It is inspired by the natural musical performance process that occurs when a musician searches for a better state of harmony. Consequently, HSA starts by initializing the problem and algorithm parameters. Then, initializing the harmony memory (HM), improvising a new harmony from the HM, and updating the HM until the termination criterion is satisfied.

It is clear that HS searches for the optimal solution through multiple solution vectors as in GA. But, its reproduction process is different from GA. While GA generates a new offspring from two parents in the population, HS generates it from all of the existing vectors stored in HM [12, 22].

These heuristic search algorithms have exponential time and space complexities as they store complete information of the path including the explored intermediate nodes [23].

To overcome the difficulties in the above optimization models, the corona virus approach is suggested.

3. The Proposed Algorithm

This section introduces a corona-based idea to optimization problems. Basically, just as corona virus infects three parts of the human body (nose, throat, respiratory), so decision variables can be assigned with any one of the three real values $(0.5, \sqrt{0.5}, 1)$ based on computational intelligence in the optimization process. It should be noted that there is a relation of these three values. Where, the second value is the square root of the first, and the third value is double the first value. Accordingly, the value "0.5" is considered as the lifeblood of this algorithm. Additionally, just as the corona virus infects the person(s) based on infection spread or treatment or death, decision variables in computer memory can be improved based on the objective function.

3.1. Procedure of Corona Algorithm

In the optimization process, the solution is estimated by putting values of decision variables to objective or fitness function and evaluating the function value with respect to several aspects such as cost, profit, efficiency, and/or error. Similarly in an intelligent corona algorithm, the solution of an optimization problem is provided based on the following four steps as shown in *Figure 4*:

Step 1. Initialize the optimization problem:

In this step, the initial population consists of one patient to each decision variable. Essentially, there are two ways in which a mathematical problem can be optimized:

1) The objective function is to minimize the infection spread of corona virus.
2) The objective function is to maximize the treatment of corona virus.

So, the optimization problem is specified as follows:

Optimize (minimize or maximize): f(x),

Subject to: $x \in S=\{g_m(x) \leq 0, m=1, 2,..., M,$

$$h_l(x)=0,\ l=1,\ 2,...,\ L\ \} \tag{2}$$

Where:

f(x) is an objective function,

$x=(x_1, x_2, \ldots, x_n)^T$ is a set of decision variables, which optimizes (minimizes or maximizes) the objective function,

n is a number of decision variables,

T is the transpose operator,

S is a feasible set, defined as

$S=\{x \mid x \in R^n, g_m(x) \leq 0, m=1, 2, \ldots, M, h_l(x)=0, l=1, 2,\ldots, L\}$,

The inequality $g_m(x)$ and equality $h_l(x)$ are real valued functions defined on S,

M and L are the numbers of inequality and equality constraints, respectively.

In addition, the termination criterion (maximum number of searches) is determined (arbitrary).

Step 2. Prepare a memory vector:

The corona search (CS) algorithm has memory storage as a vector, named corona memory (CM). Where, a group of decision variables (that contains a population) is stored in this vector. So, it can call an infected population. The problem constraints $(g_1(x), g_2(x), \ldots, g_M(x), h_1(x), h_2(x), \ldots, h_L(x))$ and the objective function value are also stored next to these decision variables. In this step, the CS vector is initially filled with any one of the three values $(0.5, \sqrt{0.5}, 1)$. The corresponding objective function value and constraints are calculated and stored in CM as follows:

$$CM=[x_1, x_2, \ldots, x_n, g_1(x), g_2(x), \ldots, g_M(x), h_1(x), h_2(x), \ldots, h_L(x),\ f(x)]^T. \tag{3}$$

CM is similar to one vector of the harmony memory in the harmony algorithm. It is a memory location that stores the solution vector (set of decision variables). Interestingly, CM is the most important part of corona search; it considers the heart of this algorithm.

Step 3. Update the corona memory (CM):

In this step, a new corona vector $(x=x_1, x_2, \ldots, x_n)^T$ is generated based on the following formula:

$$x_{i+1}=a_1\ x_i + a_2, \tag{4}$$

Where:

a_1 and a_2 are real numbers. These arbitrary values are the spreading rates of corona virus whether internal or external factors, respectively. In most cases, the value of a_2 is zero that is expressed as a traveling rate or any external factor.

The possible range of values for each decision variable is determined as $x_i^L \leq x_i \leq x_i^U$, i=1, 2, …, n. Where, x_i^L and x_i^U are the lower and upper bounds for each decision variable, respectively. The lower bound x_i^L is any one of the initial values $(0.5, \sqrt{0.5}, 1)$ for each decision variable x_i. But, the upper bound (x_i^U) of x_i is calculated by putting all variables at zero values in the set of constraints except x_i. Then the upper bound has a maximum value of x_i. Obviously, when a patient dies or he (she) is treated, the corresponding value of x_i is zero. Thus, this operation is introduced as follows:

$$x_i \begin{cases} 0 \text{ If a patient is died or treated or social isolation} \\ x_i^L \leq x_i \leq x_i^U, i = 1, 2, \ldots n. \qquad Otherwise \end{cases} \tag{5}$$

The value of each decision variable obtained as already mentioned is examined to determine whether it should be feasible and improve the solution. If the new value is better, then it is included in the corona vector. Else, it is excluded from this vector.

The formula (5) considers the mastermind of the algorithm, and it enables the algorithm to be more efficient than other artificial intelligent systems. This algorithm requires only two parameters: a_1, and a_2 that are used to improve the solution. So, the good information captured in the current iteration can be well utilized to generate the global solution easily within reasonable time. Moreover, this step likes the process of search direction or step length of each bacterium in the bacterial foraging algorithm (BFA).

Step 4. Checking the termination criterion:

After each iteration of the algorithm, two tasks are carried out:

1) Check whether the current solution is optimal (if yes, it is obtained),
2) If not, repeat Steps 2 and Steps 3 until the termination criterion (or a maximum number of iterations) is satisfied.

There is no doubt that this step simulates the following events of COVID-19: Death, isolation, traveling to any place, and the recovery or spreading of the corona virus. It is clear that the end program means reaching the optimum value; which simulates the death or life of the patient.

All of these steps are illustrated using pseudo-code as in the following *Figure 3*:

CV Algorithm

Begin

1- Initialize the CVA:

Define fitness objective f(**x**)= (**x**= $x_1, x_2, \dots, x_n)^T$, and the constraints = $g_m(\mathbf{x})$ and $h_j(\mathbf{x})$,

Define (a_1), and (a_2),

Define maximum number of iterations (NI).

2- Generate a new solution:

Selecting the initial point from these three values: $(\mathbf{0.5}, \sqrt{\mathbf{0.5}}, \mathbf{1})$, with taking into consideration that:

Max values of x_i= the upper bound (x_i^U) of x_i, i= 1, …, n.

3- Calculate the fitness function:

while (*Max number of iterations* < NI) do

  while (V_i < *number of variables*) do

   use the selected point to calculate the fitness function

   if the new solution is better than the worst solution

   then replace the worst solution by the new one with

    update the corona memory based on:

    $x_{i+1} = a_1 x_i + a_2$

   end if

  end while

end while

Optimal solution = find the current best solutions

End

Figure 3. *Pseudo-code of CV algorithm.*

Theorem 1:

If the value of each decision variable follows a linear equation, then the intersection of them is considered as a convex set that has an optimal solution.

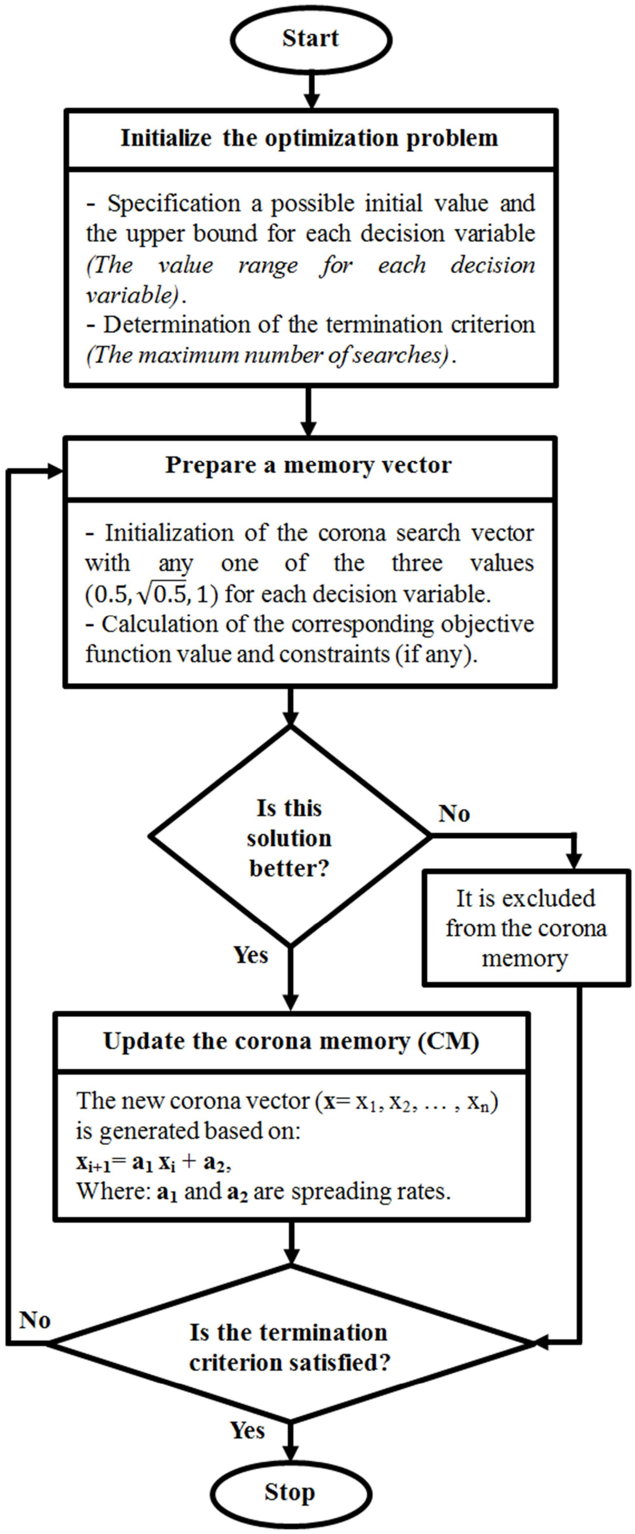

Figure 4. *Optimization procedure of the corona search algorithm.*

Proof:

The proof of this theorem is a simple consequence of linear programming optimization theory and the definition of convex set [24-25]. Since every decision variable follows a linear equation, then the intersection of them forms a convex shape. Additionally, from the characteristics of a convex set that has an optimal solution, the proof is given.

3.2. The Analogy Between Corona Virus Activity and Problem Optimization

There is an analogy between corona virus activity and problem optimization as follows:

1) The corona virus activity ↔ Decision variable.
2) The infection spread ↔ Value range. ({One of the three real numbers: $(0.5, \sqrt{0.5}, 1)$, and the upper bound of each variable}).
3) Eliminate the corona virus ↔ Solution vector. (The population of infected people who will die or treatment or social isolation).
4) The corona virus ↔ Fitness function.
5) Infection and treatment step ↔ Iteration.
6) Infected population ↔ Memory vector.
7) Corona operators:

The spreading rates of corona virus are:

a_1: The corona virus infection rate or internal rate or direct influencer or the degree of corona virus (in minimization case) or the improvement rate (in maximization case).

a_2: The traveling rate or external rate or indirect influencer.

Remarks:

1) Corona algorithm is governed by the three real numbers $(0.5, \sqrt{0.5}, 1)$, the upper bound of each variable, and the corona formula ($x_{i+1}=a_1\, x_i + a_2$).
2) The formula above makes the corona algorithm's work is easy and accurate.

3.3. The Major Advantages of the Corona Virus Optimization Algorithm

The proposed optimization algorithm has some advantages such as:

1) The arbitrary values are already set according to the corona virus activity easily.
2) The algorithm can execute by Excel (or MATLAB or any other software) program to find an optimal solution. It has the ability to end after a few or several iterations because it finishes after the optimal solution is obtained.
3) The corona virus optimization algorithm has the ability to finish its task in a fewer number of iterations with reasonable time (comparing with others such as genetic algorithm, neural network, bacteria foraging algorithm).
4) This algorithm is suitable for discrete and continuous variables. Also, it has a better chance to find the global optimal solution.
5) The stochastic random searches and derivative information are unnecessary in the mathematical of corona search. Whereas, it requires two parameters (a_1 and a_2) only, which have deterministic values to guide the search process for finding the optimal solution easily.
6) The actual values (with low or high degree) for the spreading rates are reported due to the ability of corona virus to cause infection.

4. Numerical Examples

The following examples are taken from the optimization literature, which are used to show the validity and effectiveness of the artificial corona algorithm.

Example 1:

Consider the following linear-programming problem with four variables, four equality constraints, and four bounds:

Min: $f(x)=2x_1+ 6x_2+ x_3+ x_4$,

Subject to: $x_1+ 2\, x_2 + x_4=6$,

$x_1+ 2\, x_2+ x_3+ x_4=7$,

$x_1+ 3\, x_2+ x_3+ 2x_4=7$,

$x_1+ x_2+ x_3=5$,

$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$.

This problem is presented by David and Yinyu [26]. Its optimal solution using LINDO software is $x_1^*=4$, $x_2^*=0$, $x_3^*=1$, $x_4^*=2$, and the objective function value is 11.

To solve this problem by the proposed algorithm, the steps are as follows:

1) *Initial corona memory (CM):* It seems that the objective function has a greater value multiplying by the second variable "x_2", and followed by the first variable "x_1". Therefore, it can be starting by initial solution as $(0.5, 0.5, 0.7, 1)^T$ to find the optimal solution. The corresponding value of the objective function "f(x)" is 5.7, but the four constraints are not satisfied.
2) *Update the corona memory:* To update the memory of corona vector, it can be seen firstly the last constraint or the second constraint. The upper bound of x_1, x_2, x_3, x_4 are calculated as 7, 5, 7, 7, respectively. Thus, it is necessary to set the variable "x_2" at zero value, "x_1"=0.5 × 8=4.0, "x_3"=0.7 × 1 + 0.3=1.0, "x_4"=1 × 2=2.0. In this case, all constraints (C_1, C_2, C_3, C_4) are satisfied and the corresponding objective function f(x) has a minimum value of 11.0.

Table 1. *The optimal results of example (1) by CA.*

Data	Initial solution	Update solution
x_1	0.5	0.5 × 8 + 0.0=4.0
x_2	0.5	0.5 × 0 + 0.0=0.0
x_3	0.7	0.7 × 1 + 0.3=1.0
x_4	1.0	1.0 × 2 + 0.0=2.0
C_1	2.5	6
C_2	3.2	7
C_3	3.3	7
C_4	1.7	5
f(x)	5.7	11

Table 1 shows these operations for obtaining the optimal solution. It is obvious that the optimal solution obtained using the artificial corona algorithm is the same solution previously presented.

Example 2:

This example has been previously solved by Zong Geem [27] and Quan [28]. It consists of minimizing two variables as shown in *Figure 5*. The six-hump camel-back function is defined as:

Min: $f(x)=(4\, x_1^2 - 2.1\, x_1^4 + \frac{1}{3}\, x_1^6) + x_1\, x_2 + (- 4\, x_2^2 + 4\, x_2^4)$,

Subject to: $-5 \leq x_i \leq 5$, i=1, 2.

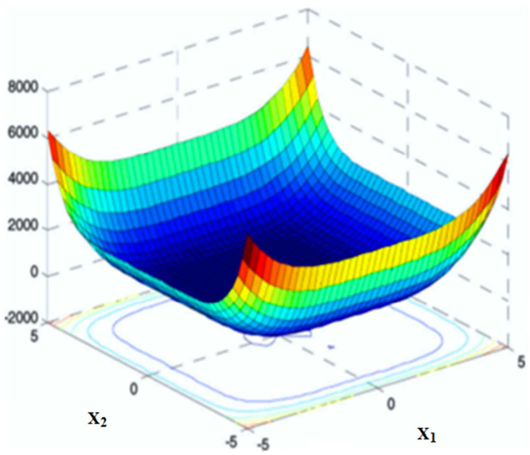

Figure 5. *2D Six-hump camel-back function.*

The comparison of results is shown in Table 2. It can be seen that the global solution of this problem obtained using the presented algorithm is better than those reported previously in the literature.

Example 3:

Consider the following nonlinear problem:

Min: $(2x - 3y)$,

Subject to: $x^2 + y^2 \le 1$,

$x + y \le 1$.

The optimal solution of this problem by Kuhn-Tucker algorithm is $x^*=0$, $y^*=1$, and $f^*=-3$. For solution by CA, the initial values for each decision variable are x=1.0, and y=0.5. But, the upper bound of x and y is one.

As shown in *Table 3*, the optimal solution by the proposed method is $x^*=-0.5547003$, $y^*=0.832050225$, $C_1=1.0$, $C_2=0.277349925$, and $f^*=2x^* - 3y^*=-3.605551275$.

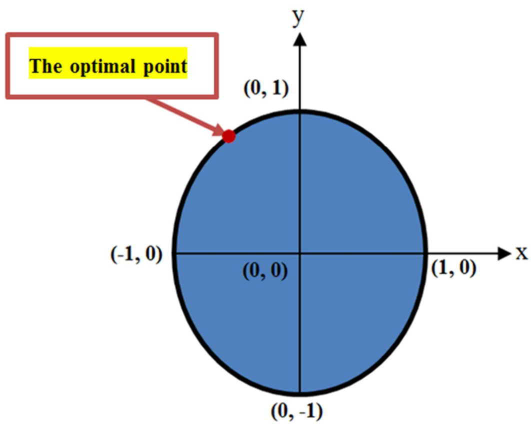

Figure 6. *The feasible region of example 3.*

Table 2. *The optimal results for minimization of the six-hump camel-back function.*

Memory vector	Proposed method			Other methods	
x_1	0.5	0.5 × -1	- 0.5 × 0.17967	- 0.08983	- 0.08975
x_2	$\sqrt{0.5}$	$\sqrt{0.5}$	$\sqrt{0.5}$ + 0.0055	0.7126	0.7127
f(x)	0.227512	- 0.4796	- 1.0316284335	- 1.0316284276	- 1.031628401
Explanation	*Initial solution*	*update solution*	*Optimal solution by CA*	*Optimal Exact solution*	*Optimal solution by HA*

Table 3. *The optimal results of example (3) by CA.*

Data	Initial solution	Update solution	
x	1.0	1.0 – 0.446=– 0.554	1.0 – 1.5547003=– 0.5547003
y	0.5	0.5	0.5 + 0.332050225=0.832050225
C_1	1.25	0.556916	1.0
C_2	1.5	– 0.054	0.277349925
f(2x – 3y)	0.5	– 2.608	–3.605551275

It is clear that the optimal solution by algorithm corona is more accurate than others.

Example 4:

This problem has been solved before by M. Mahdavi, M. Fesanghary, and E. Damangir [29] using an improved harmony search algorithm. It is stated as follows:

Min: $f(x)=(x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$,

Subject to: $4.84 - (x_1 - 0.05)^2 - (x_2 - 2.5)^2 \ge 0$,

$x_1^2 + (x_2 - 2.5)^2 - 4.84 \ge 0$,

$0 \le x_1 \le 6, 0 \le x_2 \le 6$.

The problem has two decision variables (x_1, x_2), two nonlinear inequality constraints, and four boundary conditions. The unconstrained objective function has a minimum solution at (3, 2) with a function value of f(x)=0. But, this solution is no more feasible. Where, the feasible region is a narrow crescent-shaped region with the optimum value lying on the second constraint as shown in the *Figure 7*.

The previous best solution reported by the authors of the paper [29] was obtained at $x^*=(2.2468258, 2.381863)$ with the corresponding function value=13.5908421.

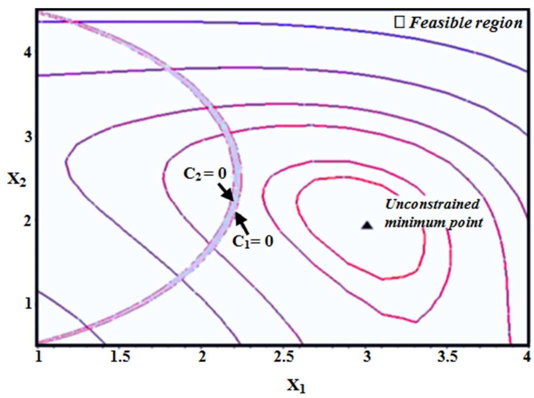

Figure 7. *The constrained function of example 4.*

The optimal solution by corona algorithm is provided at

x^*=(2.2468256, 2.381859053734) with the corresponding function value $f^*(x^*)$=13.59084169, C_1=0.0, and C_2=0.2221826. Of course, corona's solution is better than the previous paper.

Example 5:

The following problem has been presented by Valdimir [30] as a multi-objective:

Min: $f_1=-(25(x_1-2)^2+(x_2-2)^2+(x_3-1)^2+(x_4-4)^2+(x_5-1)^2)$,

Min: $f_2=(x_1^2+x_2^2+x_3^2+x_4^2+x_5^2+x_6^2)$,

Subject to:

$x_1+x_2-2\geq 0$,

$-x_1-x_2+6\geq 0$,

$x_1-x_2+2\geq 0$,

$-x_1+3x_2+2\geq 0$,

$-(x_3-3)^2-x_4+4\geq 0$,

$(x_5-3)^2+x_6-4\geq 0$,

$x_1, x_2, x_6 \in [0: 10], x_4 \in [0: 6], x_3, x_5 \in [1: 5]$.

To get the ideal (utopia) point, the optimal solution is obtained for each objective function of this problem. Firstly, the individual optimal for objectives by LINGO software are as follows:

The optimal solution for the first objective is f_1^* (x_1=0, x_2=2, x_3=5, x_4=0, x_5=5, x_6=0.3513184)=- 148. The second objective optimal is f_2^*($x_1=x_2=x_3=x_5$=1, $x_4=x_6$=0)=4.

Secondly, the optimal solution for each objective is obtained by corona algorithm as follows:

1) The initial solution: x_1=0.5, x_2=1, x_3=1, x_4=1, x_5=1, $x_6=\sqrt{0.5}$, and the corresponding objective function is -46.25.
2) The upper bound for each variable is as follows: x_1, x_2, x_6=10, x_4=6, and x_3, x_5=5. But as seen from this problem, the lower bound for two variables x_3 and x_5 is one.
3) The obtained optimal solution of the first objective is: f_1^* (x_1=5, x_2=1, x_3=5, x_4=0.00, x_5=5, x_6=0.00)=- 274. The second objective optimal is the same as solution of LINGO software: f_2^*($x_1=x_2=x_3=x_5$=1, $x_4=x_6$=0)=4. The following *Table 4* presents the optimal solutions of this example obtained using the proposed algorithm and comparing these results with solutions solved by LINGO software.

Table 4. *Optimal results for each objective of example (5).*

Data	The first objective function			The second objective function by two approaches
	Proposed method		LINGO software's solution	
	The initial solution	The final solution		
x1	0.50	0.5 × 10 + 0.00=5.00	0.00	1
x2	1.00	1.00	2.00	1
x3	1.00	1 × 5=5.00	5.00	1
x4	1.00	1 × 0=0.00	0.00	0
x5	1.00	1 × 5=5.00	5.00	1
x6	$\sqrt{0.5}$	$\sqrt{0.5}$ × 0.0=0.00	0.3513184	0
C1	- 0.50	4.00	0.00	0
C2	4.50	0.00	4.00	4
C3	1.50	6.00	0.00	2
C4	4.50	0.00	8.00	4
C5	-1.00	0.00	0.00	0
C6	$\sqrt{0.5}$	0.00	0.3513184	0
f_1^*	- 46.25	-274	-148.00	- 42
f_2^*	4.75	76	54.1234246	4
$f_1^*+f_2^*$	-41.50	-198	-93.8765754	-38

5. Conclusion

This study proposes a new artificial intelligent methodology by simulating the corona virus to get the optimal solution.

Firstly, the values of the decision variables are selected from the three real numbers $(0.5, \sqrt{0.5}, 1)$. Secondly, the upper bound of each variable is calculated by putting other variables at zero values. After that, the upper bound of each variable has the highest value of each variable. These bounds of each variable in the corona search algorithm control the generation of a new solution based on the linear formula. This formula is useful to get the optimal solution easily. The corona algorithm doesn't need to use any derivatives or randomness.

Moreover, it is compared with other artificial algorithms such as GA, BA, and HA on solving the same mathematical optimization problems. The results show that CA yields the same or better solutions than the previous solutions by these algorithms. Also, it presents the optimum value better than those solved by LINGO software.

Besides, this artificial tool offers the optimal solution for difficult or impossible problems reported previously in the literature.

Some features of the proposed algorithm are simplicity in concept, ease of implementation, and generating the optimal solution with high accuracy and speed. These features make this approach more efficient, robust, and effective to apply in many real-world optimization problems.

Finally, this work shows that the proposed approach is different from others for solving mathematical optimization problems.

For future research, this algorithm is expected to solve the multi-objective optimization programming problems to get the best efficient solution.

Acknowledgements

Thanks are to ALLAH for his guidance and supports in showing us the path.

References

[1] Ahmed T. Sahlol, DaliaYousri, Ahmed A. Ewees, Mohammed A. A. Al-qaness, Robertas Damasevicius, & Mohamed Abd Elaziz (2020). COVID-19 image classification using deep features and fractional-order marine predators algorithm. *Scientific Reports*, *Nature* research, 10: 15364, https://DOI.org/10.1038/s41598 -020-71294-2, PP. 1-15.

[2] F. Martı´nez-A´ lvarez, G. Asencio-Corte´ s, J. F. Torres, D. Gutie´rrez-Avile´ s, L. Melgar-Garcı´a, R. Pe´rez-Chaco´ n, C. Rubio-Escudero, J. C. Riquelme, & A. Troncoso (2020). Coronavirus optimization algorithm: A bioinspired meta heuristic based on the COVID-19 propagation model, *Big Data, Mary Ann Liebert, Inc.*, DOI: 10.1089/big.2020.0051, 8 (4), (*Research gate*).

[3] Hayat Ouassou, Loubna Kharchoufa, Mohamed Bouhrim, Nour Elhouda Daoudi, Hamada Imtara, Noureddine Bencheikh, Amine ELbouzidi, & Mohamed Bnouham (2020). The pathogenesis of coronavirus disease 2019 (COVID-19): Evaluation and prevention. *Journal of Immunology Research*, *Hindawi*, https://doi.org/10. 1155/2020/1357983, PP. 1-7.

[4] Mohammed A. A. Al-qaness, Ahmed A. Ewees, Hong Fan, & Mohamed Abd El Aziz (2020). Optimization method for forecasting confirmed cases of COVID-19 in China. *Journal of Clinical. Medicine*, DOI: 10.3390/jcm9030674, 9 (674), PP. 1-15.

[5] Tanu Singhal (2020). A Review of Coronavirus Disease-2019 (COVID-19), *The Indian Journal of Pediatrics*, 87 (4), PP. 281–286.

[6] Colin R. Reeves (1996). Heuristic search methods: A review, *Conference Paper*, *https://www.Researchgate.net/publication/263587732*.

[7] Krishna Nand Patel, Scahin Raina, & Saurabh Gupta (2020). Artificial intelligence and its models, *Journal of Applied Science and Computations (JASC)*, 7 (2), PP. 95-97.

[8] Prithwish Chakraborty, Gourab Ghosh Roy, Swagatam Das, Dhaval Jain, & Ajith Abraham (2009). An improved harmony search algorithm with differential mutation operator, *Fundamenta Informaticae*, 95, PP. 1–26.

[9] Reza Sirjani, Azah Mohamed, & Hussain Shareef (2012). Heuristic optimization techniques to determine optimal capacitor placement and sizing in radial distribution networks: A comprehensive review, *Przeglad Elektrotechniczny*, ISSN 0033-2097, R. 88 NR 7a.

[10] Mohammed Azmi Al-Betar, Zaid Abdi Alkareem Alyasseri, Mohammed A. Awadallah, & Iyad Abu Doush, (2020). Coronavirus herd immunity optimizer (CHIO), *Neural Computing and Applications*, Springer-Verlag London Ltd., https://doi.org/10.1007/s00521-020-05296-6, PP. 1-32.

[11] Gaurav Dhiman, V. Vinoth Kumar, A Kaur, & A. Sharma (2021). DON: Deep learning and optimization-based framework for detection of novel coronavirus disease using X-ray images. *Interdisciplinary Sciences: Computational Life Sciences*, https://doi. org/10.1007/s12539- 021- 00418- 7, PP. 1-13.

[12] Alia Youssef Gebreel (2018). An overview of genetic algorithm, bacterial foraging algorithm, and harmony search algorithm, *Global Scientific Journals*, 6 (9), PP. 165-189.

[13] P. D. D. Dominic, Ahmad Kamil, P. Parthiban, & M. Punniyamoorthy (2009). A comparative representation approach to modern heuristic search methods in a job shop. *The International Journal of Applied Management and Technology*, 6 (2), PP. 39-60.

[14] Timo Pukkala, & Mikko Kurttila (2004). Examining the performance of six heuristic optimisation techniques in different forest planning problems, *Silva Fennica*, *research articles*, 39 (1), PP. 67-80.

[15] Julia Borghoff, Lars R. Knudsen, & Krystian Matusiewicz, (2011). Hill climbing algorithms and trivium, Springer-Verlag Berlin Heidelberg, PP. 57-73.

[16] Sheldon H. Jacobson, & Enver Yücesan, (2004). Analyzing the performance of generalized hill climbing algorithms, *Journal of Heuristics*, Kluwer Academic Publishers, Manufactured in the Netherlands, 10, PP. 387-405.

[17] Andreas Fink, & Stefan Voss (1998). Applications of modern heuristic search methods to pattern sequencing problems, *Computers & Operations Research*, PP. 1-20.

[18] Baran Hekimoğlu, & Serdar Ekinci (2020). Optimally designed PID controller for a DC-DC buck converter via a hybrid whale optimization algorithm with simulated annealing, *https://www.researchgate.net/publication/339782631*, DOI: 10.5152/Electrica.19034, 20 (1), PP. 19-27.

[19] K. Y. Lee, & M. A. El-Sharkawi (2008). Modern heuristic optimization techniques with applications to power systems, *Power System Analysis, Computing*, *and Economics Committee*, *IEEE Power Engineering Society*.

[20] Alia Youssef Gebreel (2016). An adaptive interactive multi-objective optimization approach based on decision neural network, *International Journal of Scientific & Engineering Research*, 7 (8), PP. 1178-1185.

[21] K. M. Bakwad, S. S. Pattnaik, B. S. Sohi, S. Devi, B. K. Panigrahi, & Sastry V. R. Gollapudi (2010). Multimodal function optimization using synchronous bacterial foraging optimization technique, *IETE Journal of Research*, 56 (2), PP. 80-87.

[22] M. Fesanghary, E. Damangir, and I. Soleimani (2009). Design optimization of shell, & tube heat exchangers using global sensitivity analysis and harmony search algorithm, *Applied Thermal Engineering*, 29, PP. 1026–1031.

[23] Girish P Potdar, & R C Thool (2014). Comparison of various heuristic search techniques for finding shortest path, *International Journal of Artificial Intelligence & Applications (IJAIA)*, 5 (4), PP. 63-74.

[24] Dimitri P. Bertseka (2009). Convex optimization theory, *Massachusetts Institute of Technology*.

[25] Mokhtar S. Bazaraa, John J. Jarvis, & Hanif D. Sherali (2010). Linear programming and network flows, Fourth edition, A John Wiley & Sons, Inc., Publication.

[26] David G. Luenberger, & Yinyu Ye (2008). Linear and nonlinear programming, (Chapter 3-The simplex method), PDF, Third edition, PP. 73.

[27] Zong Woo Geem (2008). Novel derivative of harmony search algorithm for discrete design variables, *Applied Mathematics and Computation*, 199, PP. 223–230.

[28] Quan-Ke Pan, P. N. Suganthan, M. Fatih Tasgetiren, & J. J. Liang (2010). A self-adaptive global best harmony search algorithm for continuous optimization problems. *Applied Mathematics and Computation*, 216, PP. 830–848.

[29] M. Mahdavi, M. Fesanghary, & E. Damangir (2007). An improved harmony search algorithm for solving optimization problems", *Science Direct*, *Applied Mathematics and Computation*, 188, PP. 1567–1579.

[30] Valdimir Sevasty Anov (2013). Hybrid multi-gradient explorer algorithm for global multi-objective optimization, *American Institute of Aeronautics and Astronautics*, 10 (94-99), PP. 1-15, eartius, Inc.