COVID-19 Optimizer Algorithm, Modeling and Controlling of Coronavirus Distribution Process

Eghbal Hosseini, Kayhan Zrar Ghafoor, Ali Safaa Sadiq, Mohsen Guizani, Ali Emrouznejad

***Abstract*—The emergence of novel COVID-19 causing an overload on public health sector and a high fatality rate. The key priority is to contain the epidemic and prevent the infection rate. It is imperative to stress on ensuring extreme social distancing of the entire population and hence slowing down the epidemic spread. Further, there is a need for an efficient optimizer algorithm that can solve NP-hard in addition to applied optimization problems. This paper first proposes a novel COVID-19 optimizer Algorithm (CVA) to cover almost all feasible regions of the optimization problems. We also simulate the coronavirus distribution process in several countries around the globe. Then, we model a coronavirus distribution process as an optimization problem to minimize the number of COVID-19 infected countries and hence slow down the epidemic spread. Furthermore, we propose three scenarios to solve the optimization problem using most effective factors in the distribution process. Simulation results show one of the controlling scenarios outperform the others. Extensive simulations using several optimization problems show that the CVA technique performs best with up to 15%, 37%, 53% and 59% increase compared with Volcano Eruption Algorithm (VEA), Gray Wolf Optimizer (GWO), Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), respectively.**

***Index Terms*—COVID-19, Coronavirus Distribution Process, Coronavirus Simulated Algorithm, Controlling Coronavirus Distribution**

I. INTRODUCTION

In December 2019, the Chinese authorities notified the world that a virus was spreading to their communities. Later, the virus spread to other countries with cases doubling within days. This virus is called Sever Acute Respiratory Syndrom-Related Coronavirus 2 (COVID-19) and is simply referred to as the coronavirus. Once this virus swarm the lungs after few days of infection, it can have devastating effects and manifesting cough, severe headache and fever [1]–[3]. The majority of infected people will get through with relatively mild symptoms, but many cases become severe or even critical. Later, the World Health Organization (WHO) announced that this COVID-19 outbreak should be treated as a pandemic because of its rapid spread and the fact that most of the infected people are not immune to it [4].

For COVID-19 vaccine, many people are asking when will an effective vaccine be ready? While a vaccine is not available yet, the average reproduction rate of coronavirus is between 1.4 to 2.5; meaning that with each infection a further 2.5 people will be infected. In particular, the initial global infection rate was doubling daily in the first few weeks of 2020. In response, governments imposed lockdowns to slow the spread of the virus. Manual tracking of recent contacts history of confirmed coronavirus cases is practised by health officials. This manual contact tracing is labor intensive and requires more resources to manage large numbers of infected people [5]–[7]. In particular, infections spread more rapidly in large gatherings, such as celebrations, funerals, etc. Such scenarios will burden authorities to their maximum potentials [8], [9]. The awkward process of finding and tracing infected people will increase the probability of coronavirus occurrence and infection.

In such contagious viral infection, developing a good model is significant to make a decision on controlling the spread of the virus. Each type of virus infections has its own characteristics in terms of spreading and manifesting of symptoms. Derivation of a mathematical model to tackle the infection requires very thorough analysis of the biological characteristics of the virus [10], [11]. In particular, a good mathematical model should consider the infection through someone with no symptoms of COVID-19 clinical presentation. Currently, there are few existing mathematical models on the spreading pattern of COVID-19. In an attempt, the authors in [12] developed a real-time forecasting algorithm based on phenomenological models to evaluate short-term predictions of the cumulative number of confirmed reported cases. However, the coronavirus causes an emergency situation worldwide and needs a robust control model based on its specific characteristics. This paper presents a COVID-19 optimizer Algorithm (CVA) that mimic the behavior of COVID-19 spreading across countries over the period of December 2019 to the time of writing this paper. This paper also models the behavior of this pandemic transmission with its different stages that was experienced so far and shows that how the model can be utilized for making decisions by governments and authorities in practising optimum lockdowns. Further, the proposed CVA can cover almost all feasible regions of the optimization problems. We also simulated the coronavirus distribution process in several countries around the globe. The proposed mathematical model in this paper formulates a coronavirus distribution process as an

Eghbal Hosseini is with the Mechanical and Energy Engineering Department, Erbil Technical Engineering College, Erbil Polytechnic Univeristy, Kurdistan Region, Iraq. E-mail: kseghbalhosseini@gmail.com.

Kayhan Zrar Ghafoor is with the Department of Software Engineering, Salahaddin University-Erbil, Iraq. kayhan@ieee.org.

Ali Safaa Sadiq is with the Wolverhampton Cyber Research Institute, School of Mathematics and Computer Science, University of Wolverhampton, Wolverhampton, WV1 1LY, UK. Ali.Sadiq@wlv.ac.uk.

Mohsen Guizani is with Department of of Computer Science and Engineering, Qatar University, Qatar. mguizani@ieee.org

Ali Emrouznejad is with the Department of Operations and Information Management, Aston Business School, Aston University, Birmingham, UK

The researchers can access the implementation and programming code in https://github.com/eghbal11/Eghbal/blob/master/CVA.m

optimization problem to minimize the number of COVID-19 infected countries and with the aim to slow down the epidemic spread. Then, three scenarios are proposed to solve the optimization problem using the most effective factors in the distribution process. Simulation results show one of the controlling scenarios outperforms the others. Extensive simulations using several engineering optimization problems show the efficiency of the proposed CVA algorithm. Further, the proposed CVA technique can be applied in several research fields to solve constrained, unconstrained and engineering problems. In the existing literature, researchers use behavior of animals and insects to derive meta-heuristic algorithms whereas our proposed CVA is inspired from the COVID-19 infection process.

The rest of this paper is organized as follows. Section II explains the background of the literature on a developed CVA and controlling the spread of COVID-19. This is followed by presenting the CVA optimization algorithm and the proposed mathematical model to slow down the fast spread of the COVID-19 in section III. Finally, Section IV concludes the paper.

II. RELATED WORK

Recently, Meta-heuristic algorithms are widely used in optimizing the process of a wide-range of systems within different real-world scenarios. In the state-of-the-art, numerous number of optimization algorithms have been developed, which are mainly categorized into deterministic and stochastic algorithms. In a deterministic category, the algorithm constantly discovers a similar solution for a particular optimization problem when it starts from similar initial searching steps. This kind of algorithm is beneficial as they offer reliable performance due to its ability of obtaining a definite solution after each specific run. Nevertheless, they are also facing possibility of trapping within the local optima solution due to their lack of randomness manners, which could increase the possibility of exploring different promising rejoins while resolving a given optimization problem. On the other hand, as a second category, stochastic type of algorithms can fill-in the gap of a limited explored solution by gaining assistance from inherent stochastic operators that exist with such algorithms. As a result, various solutions could be obtained in return of each run, though the starting point is kept the same. However, the obtained results out of the stochastic algorithms are most likely to be unreliable compared with deterministic type approaches. This tradeoff drives the development of new algorithms to achieve more reliable and effective optimization methods. This was highly reflected by the wide-range of the proposed algorithms in the last decades, which are trying to find the best tuning method between randomness and quality of obtained solution over acceptable course of iterations.

Therefore, many related approaches have been developed, which are classified as a collective method that creates many random solutions and change them throughout the optimization process that follows a predefined strategy. By doing so, a more efficient method could be achieved in identifying the global optimum solution within a given search space of a problem. A set of well-known representatives of such algorithms are Genetic Algorithm (GA) [13], Particle Swarms Optimization (PSO) [14], Ant Colony Optimization (ACO), and Differential Evolution (DE). Obtaining wide-range of solutions normally reduce the chances of trapping with the local optimal solution of a given problem, which is considered one of the winning points of stochastic algorithms. Yet, achieving each solution enquires an evaluation function and finding the optimal collaboration between each solution, which is a challenging task. Nevertheless, such kind of algorithms introducing some challenges but they are widely used nowadays. This is due to the fact that their simplicity in solving problems by taking simple strategies inspired by some naive nature based collaborative behavior. Hence, several algorithms have been developed as a way to push further the advantages of such algorithms and lessen their limitation/drawbacks. We can name for example, Artificial Bee Colony (ABC) [15] optimization, Cuckoo Search (CS) [16], Firefly Algorithm (FA) [17], Gray Wolf Optimizer (GWO) [18], Salp Swarm Algorithm (SSA) [19], Whale Optimization Algorithm (WOA) [20], Laying Chicken ALgorithm [21], Big Bang Algorithm [22] and Swine Influenza Inspired Optimization (SIIO) [23].

On the other hand, taking a simple correlation factor between the value of R and the transmission speed of an infection of any similar virus, the higher the R-value is the higher the number of infected people. Hence, from this point we can highlight the main concept behind the powerfulness of novel COVID-19's transmission speed in comparison with another seasonal flu. As was reported in [24], a seasonal flu can be transferred from one person to another within an average of 1.4, while COVID-19 transfers within an average of 2-3; which formed the initial stages of the outbreak.

It is important to highlight how likely the volume of infected people behaves. This relies on several factors, such as the number of vulnerable people in the population, the period of time that each person carries the virus asymptomatically, the probability of each person getting in contact with others (number of social contacts) and the possibility of transferring the virus to them as well as the frequency of having a group of people visiting a common place (such as shopping malls, airports, train stations, etc.) over a similar period of time. Taking the aforementioned factors into consideration, governments and authorities in charge can optimally plan the method of controlling the spread of such epidemic, which is highly related to maintaining the rate of value R (where R represents the reproduction rate). Accordingly, one can have a better understanding of the health ecosystem and thus manage to save more lives. Consider China, for instance, they had to immediately model the R rate and measure the transmission speed to take a quick decision within the right time and right measure.

III. THE COVID-19 INSPIRED ALGORITHM (CVA)

The details of our proposed CVA algorithm is discussed in this section. The details include mathematical equations, simulation of coronavirus distribution and other related steps. The exploration as well as exploitation processes of CVA algorithm in finding the optimal solution of an optimization problem are outlined in this section.

Coronavirus in nature has started spreading rapidly due to its high transmission behavior and speed of the R-value, (as stated within initial estimates of reinfection average of 2-3), within each country and then outbreaks in that country. These outbreaks and export processes have been inspired the proposed CVA algorithm in generating its initial solutions and populations mimicking COVID-19 spreading behavior. During the process of COVID-19 spreading, some confirmed cases (solutions in CVA) may not outbreak in specific region or export to other countries. This is due to that fact that those cases (solutions in CVA) are recovered or already passed away. Considering this scenario, these solutions will be removed from the feasible list in the next population and the algorithm will carry on exploring the remaining best solutions. Then, the CVA performs the spreading process of remaining best solutions to the rest of population.

A. *COVID-19 Outbreak and Spreading*

The initial solution is randomly set in a region of feasible space and initial population is generated based on the outbreak in that region. Possible solutions will also be exported to other regions, the same as coronavirus distribution process across cities and countries. We have defined the exporting process of the virus to other countries as the power of export, which could be different for each country based on its community's behavior and social activities. Here, for simplification this power is defined as a constant p for all countries, which is a percentage of the number of solutions in the initial population, $0 < p < 1$. Let n is the number of solutions for initial population, so $(1 - p)$ of n are generated close to the initial solution based on the following equation 1:

$$||x_i - x_0|| < r, i = 1, 2..., (1 - p).n \tag{1}$$

where x_0 is the initial solution, r is a positive constant number which restricts solutions in a specific region. When the generated number r exceeds its upper bound, the outbreak scenario will occur. As it was based on the inspiration of COVID-19 outbreak in a country, the CVA triggers the outbreak to reinforce the search by exploiting this part of population intensively; looking for the optimal solution. The parameter p is the percentage of the number of solutions in the initial population and generated far from the region of the initial solution in the feasible space based on following equation:

$$x_i = x_0 + \lambda.(rand), i = (1 - p).n + 1, (1 - p).n + 2, ..., n \tag{2}$$

Where *rand* is a real random number between $(-1, 1)$ and λ is a large enough positive number. The above generation process has been inspired from the COVID-19 export to other countries; this part of population would not trigger the outbreak only after two iterations in our simulation. The reason behind this assumption is to obtain more realistic simulation, where there is less possibility of having all people exposed/infected by the virus at the first instant. So there is an opportunity to control and quarantine infected cases before they could lead to an outbreak. In fact, the initial population of the CVA displays the first generation of people that were exposed to the virus. In the process of coronavirus distribution in our inspired algorithm, the population distributes the virus all over the region, so it uses this initial population to obtain more outbreak and export to cover all feasible regions in the optimization problems' spaces.

B. *Recovery, Remove and Transmission*

In the distribution and infection process of the coronavirus, there are two categories of people: *active cases* who have the virus and can transmit it to others whereas *closed cases* is defined as the solutions that are either were recovered from the COVID-19 symptoms or dead. In this context, in our COVID-19 optimizer and after generation of the initial population, there are two kinds of solutions in each population: *closed cases* that are not contributing to transmit to other regions of feasible space and are removed from the population. The second type of solutions, named *active cases*, can transmit the virus immediately or after a while (two iterations) of exported solutions.

In CVA, transmission power is a factor that enables spreading of the solutions to other feasible regions in the searching space for generating the next population. This factor could be different for each solution (person) based on its communities compared with others. Furthermore, the transmission power is a positive integer constant, t, for all active solutions and it is zero for closed cases. Hence, the number of active solutions, n_{act}, is obtained from the following equation:

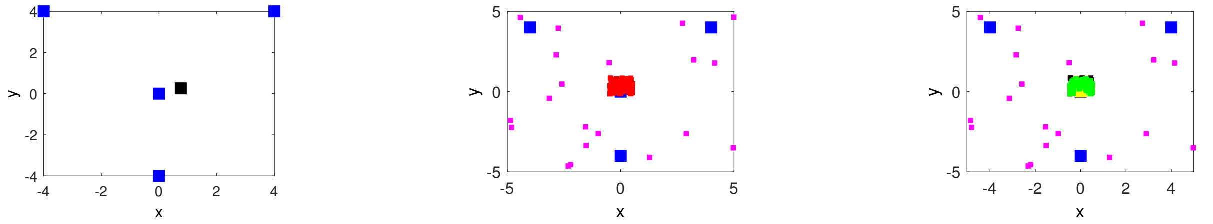
$$n_{act} = act.n \tag{3}$$

where n is the number of solution in the current population and act is the rate of active solutions in the current population (we empirically set $act = 0.3$). In the simulation, all solutions of the population are sorted based on their functions' values and among n solutions where the best n_{act} will be selected as active cases. Then, the remaining solutions $(n - n_{act})$ will be removed. The remaining solutions are defined as the closed cases and are represented by the following equation :

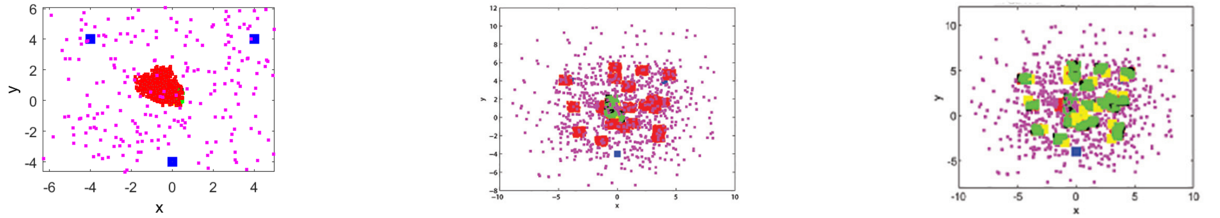
$$n_{clc} = n - n_{act} \tag{4}$$

where n_{clc} is the number of closed cases that means all recovered and dead solutions.

In our proposed CVA algorithm, the best solution will be found out from the active cases, n_{act} and saved accordingly. This process will be repeated over throughout the course of iterations and eventually obtain the best solution n_{act}, which will be announced as the optimal solution. For generating the next population, each solution of active cases can transmit the virus to new feasible solutions named t, where t is defined as the power of infection transmission. Hence, the number of solutions in the next population, n_{next}, can be obtained from Equation 5:



(a) Optimal solutions and initial solution (b) Initial population (c) Recovery and best solution Of population



(d) Outbreak in initial region and more spreading (e) Outbreak in other regions (f) Surrounding optimal solutions

Fig. 1. The process of COVID-19 spreading for a given optimization problem

$$n_{next} = n_{act} + t.n_{act} = (t+1).n_{act} \tag{5}$$

Using equations 3 and 5, we get:

$$n_{next} = (t+1).act.n \tag{6}$$

Equation 6 calculates the number of solutions in the next population that has a direct relation with n, act, and t. Thus, by increasing the number of solutions, the rate of active solutions and infection transmission power, the speed of distribution will be increased accordingly.

C. The Procedure of COVID-19 Inspired Algorithm

The principal steps of the coronavirus inspired algorithm in R^n are proposed as follows:

1) The initial feasible solution is created randomly. This solution implies the initial region or country where the virus starts. The parameters of the algorithm are: number of solutions, n; an arbitrary small positive number, r; number of iterations, k with $k = 1$ at the first iteration; the power of export, p; the percentage of the recovery, rec; the rate of active solutions, act; the infection transmission power, t; a positive small number, ϵ_1, and the maximum number of iterations, M_1. It is worth to mention that the parameters of the CVA technique are empirically set.
2) The initial population with n solutions is generated as follows: $1 - p$ percentage of solutions are generated very close to the solution with radius r and other p percentage are generated far from the region of the initial solution in the feasible space. The coronavirus spreads some solutions from the initial attempt to other regions because of human social interaction all over the world.
3) Based on the coronavirus termination process, some of the patients in step 2 that have been recovered or died, these solutions are removed from the population.
4) The COVID-19 is spreading in a specific country on one hand and will be transmitted to other countries after sometime on the other hand. Likewise, the CVA technique is composed of two categorises of solutions that are contributed to find the optimal value. More precisely, solutions in the initial region transmit promptly (red points in Fig. 1) whereas the rest of solutions will be transmitted after two iterations (magnet points in Fig. 1).
5) The best solutions of each generation are found and saved. Then, the best solution among them will be selected as the optimal solution by the algorithm. If $k > 1$, go to the next step, otherwise let $k = k+1$ and go back to step 3.
6) If $d((F(x^{k-1}), F(x^k)) = (\sum_{i=1}^{n}(F(x_i^{k-1}) - F(x_i^k))^2)^{1/2}) < \epsilon_1$ or the number of iteration is more than M_1, the algorithm will be terminated. $x^{(k-1)} and x^k$, are the best solutions in two consecutive generations. Otherwise, let $k = k+1$ and go to the step 2.

Fig. 1 shows the process of CVA to find the optimal solutions, Fig. 1a shows the optimal solution (blue points) and the initial solution (black point) in a given optimization problem. As shown in Fig. 1b, the initial population includes solutions, which can outbreak and spread (red points) as well as solutions exposed to be spread to other regions, but would not outbreak after two iterations (magnate points). The recovery process is illustrated in Fig. 1c; it includes closed cases (green points, recovered, and black points, dead) and the best solution of the current population (large yellow points). Fig. 1d shows more outbreak and spreading from active solutions. Further, Fig. 1e shows the spreading solution outbreak in their regions and exporting to other districts. Eventually, Fig. 1f depicts the surrounding optimal solutions by populations. More specifically, the yellow points are the best among populations and our proposed CVA selects the best among those best solutions.

To show the performance of the proposed CVA, we use the benchmark functions of well-known optimization problems. Table I shows mathematical formulations of the problems, Table II presents the optimal values of proposed algorithm and Figs. **??**, **??** and **??** show the process of finding the optimal

TABLE I. Optimization test functions examples 1-8

Examples	Equation
Example 1	$f(x,y) = exp(-(x-4).^2 - (y-4).^2) + exp(-(x+4).^2 - (y-4).^2) + 2 * exp(-(x).^2 - (y).^2) + 2 * exp(-(x).^2 - (y+4).^2)$
Example 2 - Deckkers-Aarts Function	$f(x,y) = 10^5 x^2 + y^2 - (x^2+y^2)^2 + 10^{-5}(x^2+y^2)^4$
Example 3 - Himmelblau Function	$f(x,y) = (x^2+y-11)^2 + (x+y^2-7)^2$
Example 4 - Sphere Function	$f(\mathbf{x}) = f(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} x_i^2$
Example 5 - Rosenbrock Function	$f(x,y) = \sum_{i=1}^{n} [b(x_{i+1} - x_i^2)^2 + (a - x_i)^2]$
Example 6 - Bukin N. 6 Function	$f(x,y) = 100\sqrt{\lvert y - 0.01x^2 \rvert} + 0.01\lvert x + 10 \rvert$
Example 7 - Holder-Table Function	$f(x,y) = -\lvert sin(x) * cos(y) * exp(\lvert 1 - \frac{\sqrt{x^2+y^2}}{\pi} \rvert) \rvert$
Example 8 - McCormick Function	$f(x,y) = sin(x+y) + (x-y)^2 - 1.5x + 2.5y + 1$

solution using the CVA algorithm for the given optimization problems. In all benchmark examples, we set the number of agents n to 100, the rate of active cases (act) to 0.3, the number of iterations (M) to 1; and the epsilon to 0.5. Fig. **??** presents the behavior of the original function, Fig. **??** finds the optimal solutions when the CVA is applied. Similarly, the proposed CVA can get the optimal solution shown in Figs. **??** and **??**.

In order to evaluate the performance of our CVA, we used the proposed algorithm to solve more optimization problems. In particular, Fig. **??**, Fig. **??** and Fig. **??** show the results of optimization problems, which is presented in examples 4, 5 and 6, respectively. As can be seen, the proposed CVA could find the optimal solution with a short amount of time for all test functions in examples 4, 5 and 6.

The performance of the proposed algorithm is high since it can produce efficient solutions with only one iteration. Furthermore, the factor *act* is set to 0.3. Increasing this factor accelerates the convergence of the algorithm.

TABLE II. Results of the proposed CVA for Examples 2-9

	(x,y)	OS by CVA	Optimal Solution	IS
Example 1	(-0.0084, 0.0072)	1.9998	2	Random between (-1.5,1.5)
Example 2	(-0.0143, -10.7994)	-1.1615e+04	−24771.09375	Random between (0,1)
Example 3	(3.0039,1.9995)	0.0005	0	Random between (-3,2)
Example 4	(-0.0077,0.0166)	0.0003	0	Random between (-5,5)
Example 5	(0.9176,0.8431)	0.0069	0	Random between (0,2)
Example 6	(-8.2987,0.6886)	1.0044	0	Random between (-10,1)
Example 7	(-8.1329,-9.1749)	-16.9743	−19.2085	Random between (-8,-6)
Example 8	(-0.5481,-1.5564)	-1.9131	−1.9133	Random between (-2,1)

D. Performance Evaluation

This section presents the performance evaluation of the proposed CVA and most related meta-heuristic algorithms, such as the Volcano Eruption Algorithm (VEA) [25], the Grey Wolf Optimizer (GWO) [18], the PSO [14] and the GA [13]. Several well-known optimization problems have been used as benchmark test functions to verify the validity of the CVA performance evaluation. In Particular, the CVA is used to solve two different categories of test functions: unimodal and the multi-modal. Unimodal test functions that have just a global optimum, but multi-modal test functions have a global optimum as well as multiple local optima.

In the experiments, we set the number of agents to 20, the maximum number of iterations to 100 and the epsilon to 0.1. In order to give credibility to the results, every value in Table III represents 50 runs of the algorithm. Further, the results in Table III show that the proposed algorithm is efficient on both the unimodal and multi-modal test functions as compared to existing algorithms in the literature.

E. The Control of Coronavirus Spreading

This section first presents the importance of preventing the spread of COVID-19 by developing a mathematical controlling model. The control of COVID-19 spreading is adopted for person-to-person infection and spreading among countries. The proposed mathematical model is derived to overcome the spread of COVID-19 among countries. The proposed control model maximizes the number of safe countries, while fulfilling the constraints of formulae 8 to 13.

The problem of COVID-19 spreading is defined as follow:
Definition:
During the coronavirus spread, there are N regions (countries) with four states at k periods of time:

1) Countries are assumed in a safe state.
2) Countries are assumed to be safe, but they are susceptible to COVID-19 infection.
3) Countries are infected with COVID-19 that can be transmitted to other countries.
4) Countries can be infected with COVID-19, but do not transmit to other countries

In the aforementioned states, countries in state 4 do not need assistance from neighbouring regions due to the imposed lockdown to reduce and control the disease spread. Countries in states 2 and 3 can still receive aid since they are still not in lockdown yet. However, in this paper we ignore this assumption for simplicity purpose. Moreover, help can be made to countries in state 2 by finding and quarantine people who recently arrived from other countries in state 3. Finally, help for countries in state 3 can be made by testing people that have visited an infected country. For instance, applying COVID-19 test to all passengers who plan to leave such country. By this help, countries in state 3 would be quarantined and changed to state 4.

According to the aforementioned assumptions and states to control COVID-19 spreading, the objective is to maximize the number of countries staying at state 1 during the fast spreading process. To achieve this goal, the control of COVID-19

TABLE III. Comparison of CVA with VEA, GWO, PSO and GA meta-heuristic techniques for several types of optimization problems

F	CVA		VEA		GWO		PSO		GA	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
Function 1	0.0107	0.0106	0.0142	0.0112	0.2080	0.1195	0.5774	0.3185	18.7254	4.9352
Function 2	1.9435	0.7396	1.7367	0.5892	2319.19	1237.109	3.5523	2.8537	27187.58	2745.82
Function 3	9.1690	6.8371	12.4784	8.3213	89.1347	37.9576	99.8320	24.6287	273.2519	29.5521
Function 4	1.0749	0.6921	1.3269	0.8263	9.4525	3.4676	4.2950	1.3083	18.5965	0.3517
Function 5	4.0734	2.9351	5.1245	3.2689	14.4316	5.9230	8.7162	4.9291	68.6618	6.0623

spreading is modeled as a mixed binary-integer programming problem as follows:

$$\max \quad \sum_{k=1}^{m} x_{1t_k} \tag{7}$$

subject to

$$x_{1t_k} = x_{1t_{k-1}} + \sum_{k=1}^{N} c_{i2t_{k-1}} * h_{it_{k-1}} - \sum_{k=1}^{N} c_{i3t_{k-1}} * S_{it_{k-1}}, \tag{8}$$

$$x_{2t_k} = \sum_{k=1}^{N} c_{i3t_{k-1}} * S_{it_{k-1}}, \tag{9}$$

$$x_{3t_k} = x_{3t_{k-1}} + \sum_{k=1}^{N} c_{i2t_{k-1}} * (1 - h_{it_{k-1}}) - \sum_{k=1}^{N} c_{i3t_{k-1}} * h_{it_{k-1}}, \tag{10}$$

$$x_{4t_k} = x_{4t_{k-1}} + \sum_{k=1}^{N} c_{i3t_{k-1}} * h_{it_{k-1}}, \tag{11}$$

$$x_{1t_1} = N - 1, x_{2t_1} = 0, x_{3t_1} = 1, x_{4t_1} = 0 \tag{12}$$

$$\sum_{j=1}^{4} x_{jt_k} = N, for\ each\ k = 1, 2, 3....m \tag{13}$$

where m is the time period, N is the number of countries, $0 \leq x_{jt_k} \leq$ N, $c_{ijt_k} \in \{0, 1\}$, $e_{ijt_k} \geq 0$,i=1,2,3,4...N, j=1,2,3,4, K=1,2,...,m. Variable x_{jt_k} is the number of countries in state j at a stipulated time period t_k, variable c_{ijt_k} is 1 if i_{th} country be in state j at stage t_k, and it is 0 otherwise. Moreover, variable h_{it_k} is 1 if i_{th} country at stage t_k gets help otherwise its value is 0. S_{it_k} is the spreading rate country i has at stage t_k. The value of S_{1t_k}=S_{2t_k}=S_{4t_k}=0, because only countries at state 3 can spread the COVID-19 to other countries for all k, whereas the value of S_{3t_k} is a positive integer. The objective function is utilized to maximize the number of countries stay in safe state in m periods of time. Constraint 8 to 13 are for $k = 2, 3, \ldots, m$ number of safe countries (state 1) at the k_{th} stage = countries in state 1 plus countries in state 2, but they get help (help is a key measures to prevent COVID-19 spreading) minus the COVID-19 infected countries, which are responsible in spreading the virus to other countries in state 3 and stage (k-1). Further, constraint 9 originally is:

$$x_{2t_k} = x_{2t_k} + \sum_{k=1}^{N} c_{it_{k-1}3} * S_i - \sum_{k=1}^{N} c_{it_{k-1}2} * h_i - \sum_{k=1}^{N} c_{it_{k-1}2} * (1 - h_i) \tag{14}$$

As shown in formula 14, the number of countries in state 2 at the k_{th} stage are shrinking in case of getting help at $(k-1)th$ stage, whereas this number is increasing if they don't get help. So, all countries in this state will be changed to states 1 or 3. Furthermore, countries that are infected from countries at state 3 will transit their state to 2. Constraint 10 shows the number of countries in state 3 at the $(k-1)th$ stage. Constraint 10 is derived from the number of countries in state 3. Countries in state 2 don't get help with excluding countries in state 3, where to get help at $k - 1_{th}$ stage. Further, constraint 11 is derived from the number of countries in state 4 (closed case) at stage k, which is derived from countries in state 4 and also countries in state 3, where to get help at stage $k - 1$. Constraint 12 shows the state of countries at the beginning of coronavirus which starts in a country. Constraint 12 indicates that the summation of the number of countries in all states should be equal to the number of countries, N, at all period of time. The state transition diagram of the proposed COVID-19 spreading control is illustrated in Fig. **??**.

F. Normal Distribution

In this sub-section, the process of coronavirus distribution is presented (Fig.**??**a). A comparison has been conducted that shows the distribution of the results with different scenarios. Assuming a scenario with the number of countries (region or people) is N=50, number of agents (virus) is n=50, number of running iterations (stages or periods of time) is 5, power of exporting for each country is 20 percentage of solutions (this can be based on the communication in each country, but in this paper, this is fixed for each country). Two main scenarios are simulated to illustrate the distribution of the virus among cities as detailed below.

G. Scenario I

In this scenario, countries allocated in state 3 may get help and change to state 4. In fact, countries that are categorized under state 3, they could export virus to other countries; hence helping them can be more effective in saving other countries. Help for state 3, based on the definition, is to control all commuting of the country. For example, testing passengers who exit from the country, so that they won't export the virus to their destination country. By this kind of help, the country will be changed to the lockdown status (state 4). But, based on the obtained results and extensive simulation of this scenario, it is not recommended to control the spreading of the virus. The reason is that countries in state 3 can export the virus to other countries, because of asymptomatic transmission. Even before officials find out the start of infections in a country, the people who infected with COVID-19 may have travelled to other regions. In fig. **??**b, large green points are safe countries (state 1), green points with small magnet point in the center are countries in state 2 (exposed of getting the virus), green points with red points in the center are countries in state 3 (with virus and can export), green points with yellow points are for countries in state 4 (with virus but not exported).

H. Scenario II

1) Scenario II - A: In this scenario, not only countries with state 3 will be changed to state 4, but also countries that are exposed to get the virus in order to get help and change to state 1. Here, we suppose all countries in state 2 are recognized before, their state is changed to state 3. However, this consideration is a little hard and costly in reality. But, we could observe from simulation that the results can be substantially effective. The reason obtained from the simulation was that, the most powerful factor of the coronavirus distribution is the power of exporting it to other countries. That in turn will move them to state 3 at the same time from the countries, where just got the virus.

In Fig. **??**c, large green points are safe countries (state 1), green points with yellow points in the center are countries that got the virus already. But with getting help, their status has changed to state 4 (not exporting the virus anymore). On the other hand, green points with small magnet points in the center are countries in state 2 (expose to getting the virus), green points with white points in the center are countries in state 2, but with getting help, they will be changed to state 1.

From Fig. **??**c, it is clear that all countries will converge to a safe state, (except the initial country), after only one iteration of the virus distribution. This is because the fact that the initial country has been controlled such that it could not export the virus anymore. Hence, this country will be avoided from exporting the virus to some other countries as it is in state 3. Similarly, all countries in state 2 are infected and obtained help to be safe, so there will be no other countries to export the virus to others. The main challenge of this scenario is we may not recognize all countries that were exposed to the virus (state 2) before their status is changed to state 3. So to be realistic, we assume that the percentage of these countries can not be recognized right after starting virus in the initial country.

Algorithm 1 Scenario II-B Algorithm

```
Input: countries in state 2 and 3
Output: countries in state 1 and 4
x0: initial feasible solution
 xbest: Best solution of population
 hi: help to country i
 ci2: country i in state 2
 ci3: country i in state 3
 n2: number of countries in state 2
 n3: number of countries in state 3
 N: Number of countries
 k=1
 n: number of solutions
 p: power of export
 act: rate of active solutions
 t: transmission power
 rand: real random number between (-1,1)
 L: a large positive number
 M: maximum number of iterations
 epsilon, r: two positive small numbers
 while $k < itr$ do
     for $i \leftarrow 1$ to $(1 - p) * n$ do
         generate $x_i$ such that ——xi-x0——¡r
         for $i \leftarrow (1 - p) * n$ To $n$ do
             let $x_i = x_0 + L * rand$
         for $i \leftarrow 1$ To $n3$ do
             apply $c_{i3} * h_i$ for country i
         for $i \leftarrow 1$ To $n2$ do
             apply $c_{i2} * h_i$ for country i
         x0, y0=(xbest, ybest) k++ n=(t+1)*act*n
```

2) Scenario II - B: In this scenario, we assume 50 percentage of countries in state 2 will be recognized and get help before their states are changed to state 3. The power of export of countries in state 3 is also very effective in this scenario as well as in the power of distribution. Export power for each country is the rate of communication with other countries, so by increasing the number of countries (people) in our simulation, this power should be increased too, because the community of countries has a direct relation with the number of countries. Algorithm 1 illustrates the procedure of scenario II-B.

Fig. **??**d shows the process of scenario II-B in 5 iterations. Based on Fig. **??**d, 34 countries have been saved from being exposed to the virus, which can be a significant achievement. The results would be much better as much as recognizing a number of countries in state 2. In other words, during the coronavirus, we should focus on countries or people who have been exposed to the virus. Luckily, these achievements are valid for people too, so authors strongly suggest quarantining a recognized sick person and all people that had contact with a positive virus carrier recently. By this scenario, the virus cannot transmit to others and it will be controlled very soon, hence its dissemination will be substantially mitigated. Fig. **??** shows a comparison of normal distribution, scenario I, scenario II – A and scenario II – B with different numbers of people (country). We can observe from scenario II-A that the virus distribution can be controlled with very high performance, while scenario I presents the approximate behavior of the normal distribution of the virus.

Scenario I represents the mass quarantine strategy in which the government authorities around the globe have implemented to prevent the COVID-19 spreading. However, practising lockdown strategy slams the economy and increases unemployment rate. Therefore, we strongly suggest deploying scenario II – A, which is based on the obtained results it could control the virus very quickly (for both countries and people). Besides, the use of scenario II – B that could also control the distribution processes much better than the conventional way of scenario I.

Analyzing the complexity of any algorithm will quantify how fast it can run as a function of the input size. The running time of an algorithm depends on several factors: input size as well as the used hardware and software. Here, we are interested in the rate of time growth according to the input size and the algorithm used. In Algorithm 1, there are 17 assignments and several procedures in order to make countries stay safe. In particular, the complexity ($O(max((1-p),n,M,n3,n2,n*p)))$) of the proposed algorithm can be derived as follows:

$$O(max((1-p),n,M,n3,n2,n*p))) = (1-p)\times n\times M[5+2 \times n_3 + 2\ times 2\times n_2 + 2\times(n\times p+1)] + \frac{17}{(1-p)\times n\times M} \tag{15}$$

Referring to equation 15, the maximum variable is n, which is number of countries. In worst case scenarios, the complexity of our proposed algorithm is $O(n^3)$.

IV. CONCLUSION

This paper answers an important question; how can governments and health authorities be able to protect people back to their normal life in the era of COVID-19 spreading? To achieve this goal, we modeled the coronavirus distribution process as an optimization problem to minimize the number of COVID-19 infected countries and hence slow down the epidemic spread. This will certainly ease the lockdown measures. Then, three scenarios are proposed to solve the optimization problem using the most effective factors in the distribution process. We also propose a novel COVID-19 optimizer Algorithm (CVA) to cover almost all feasible regions of the optimization problem. This is confirmed through a simulated coronavirus distribution process in several countries around the globe. Extensive simulations using several engineering optimization problems have shown the efficiency of the proposed CVA algorithm. For the future research directions, we will pursue the re-infection process where countries that have viruses and do not export to the other countries (state 4) may spread COVID-19 to other countries (state 3).

REFERENCES

[1] J. Chen, L. Wu, J. Zhang, L. Zhang, D. Gong, Y. Zhao, S. Hu, Y. Wang, X. Hu, B. Zheng *et al.*, "Deep learning-based model for detecting 2019 novel coronavirus pneumonia on high-resolution computed tomography: a prospective study," *medRxiv*, 2020.

[2] M. Holshue, C. DeBolt, and L. S. First, "Novel coronavirus in the united states," *N Engl J Med*, vol. 2020, p. 31, 2019.

[3] H. S. Maghdid, A. T. Asaad, K. Z. Ghafoor, A. S. Sadiq, and M. K. Khan, "Diagnosing covid-19 pneumonia from x-ray and ct images using deep learning and transfer learning algorithms," *arXiv preprint arXiv:2004.00038*, 2020.

[4] "Cnn health," https://edition.cnn.com/2020/03/11/health/coronavirus-pandemic-world-health-organization/index.html, Accessed in March, 2020.

[5] B. N. Wetsman, "What is contact tracing?" https://www.theverge.com/2020/4/10/21216550/contact-tracing-coronavirus-what-is-tracking-spread-how-it-works, April,2020.

[6] H. S. Maghdid, K. Z. Ghafoor, A. S. Sadiq, K. Curran, and K. Rabie, "A novel ai-enabled framework to diagnose coronavirus covid 19 using smartphone embedded sensors: Design study," *arXiv preprint arXiv:2003.07434*, 2020.

[7] A. Wilder-Smith, C. J. Chiew, and V. J. Lee, "Can we contain the covid-19 outbreak with the same measures as for sars?" *The Lancet Infectious Diseases*, 2020.

[8] H. Shelani. Number of covid-19 cases reaches 176 in kurdistan region; iraq's total now 703. [Online]. Available: https://www.kurdistan24.net/en/news/64f6c6f0-6416-471f-b018-47c133ecc8c3note=

[9] H. S. Maghdid and K. Z. Ghafoor, "A smartphone enabled approach to manage covid-19 lockdown and economic crisis," *arXiv preprint arXiv:2004.12240*, 2020.

[10] R. M. Anderson and R. M. May, "Population biology of infectious diseases: Part i," *Nature*, vol. 280, no. 5721, pp. 361–367, 1979.

[11] B. C. Malathesh, G. S. Gowda, C. N. Kumar, M. Narayana, and S. B. Math, "Response to: Rethinking online mental health services in china during the covid-19 epidemic," *Asian journal of psychiatry*, 2020.

[12] K. Roosa, Y. Lee, R. Luo, A. Kirpich, R. Rothenberg, J. Hyman, P. Yan, and G. Chowell, "Real-time forecasts of the covid-19 epidemic in china from february 5th to february 24th, 2020," *Infectious Disease Modelling*, vol. 5, pp. 256–263, 2020.

[13] J. H. Holland *et al.*, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[14] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[15] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *Journal of global optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[16] X.-S. Yang and S. Deb, "Cuckoo search via lévy flights," in *2009 World congress on nature & biologically inspired computing (NaBIC)*. IEEE, 2009, pp. 210–214.

[17] M. Shehab and A. T. Khader, "Modified cuckoo search algorithm using a new selection scheme for unconstrained optimization problems," *Current Medical Imaging*, vol. 16, no. 4, pp. 307–315, 2020.

[18] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

[19] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.

[20] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in engineering software*, vol. 95, pp. 51–67, 2016.

[21] E. Hosseini, "Laying chicken algorithm: A new meta-heuristic approach to solve continuous programming problems," *J Appl Computat Math*, vol. 6, no. 344, p. 2, 2017.

[22] H. Eghbal, "Big bang algorithm: A new meta-heuristic approach for solving optimization problems," *Asian Journal of Applied Sciences*, vol. 10, no. 3, pp. 134–144, 2017.

[23] S. S. Pattnaik, D. G. Jadhav, S. Devi, and R. K. Ratho, "Swine influenza inspired optimization algorithm and its application to multimodal function optimization and noise removal." *Artif. Intell. Research*, vol. 1, no. 1, pp. 18–30, 2012.

[24] What we scientists have discovered about how each age group spreads covid-19. (2020, March). [Online]. Available: https://www.theguardian.com/commentisfree/2020/mar/17/scientists-age-groups-covid-19-workplaces-shops-restaurants

[25] E. Hosseini, A. Safaa Sadiq, K. Ghafoor, D. B. Rawat, M. Saif, and X. Yang, "Volcano eruption algorithm for solving optimization problems," in *Neural Computing and Applications*, vol. In press. Springer.