



Coronavirus herd immunity optimizer (CHIO)

Mohammed Azmi Al-Betar^{1,2} · Zaid Abdi Alkareem Alyasseri^{3,4} · Mohammed A. Awadallah⁵ · Iyad Abu Doush^{6,7}

Received: 18 April 2020 / Accepted: 12 August 2020 / Published online: 27 August 2020
© Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract

In this paper, a new nature-inspired human-based optimization algorithm is proposed which is called coronavirus herd immunity optimizer (CHIO). The inspiration of CHIO is originated from the herd immunity concept as a way to tackle coronavirus pandemic (COVID-19). The speed of spreading coronavirus infection depends on how the infected individuals directly contact with other society members. In order to protect other members of society from the disease, social distancing is suggested by health experts. Herd immunity is a state the population reaches when most of the population is immune which results in the prevention of disease transmission. These concepts are modeled in terms of optimization concepts. CHIO mimics the herd immunity strategy as well as the social distancing concepts. Three types of individual cases are utilized for herd immunity: susceptible, infected, and immuned. This is to determine how the newly generated solution updates its genes with social distancing strategies. CHIO is evaluated using 23 well-known benchmark functions. Initially, the sensitivity of CHIO to its parameters is studied. Thereafter, the comparative evaluation against seven state-of-the-art methods is conducted. The comparative analysis verifies that CHIO is able to yield very competitive results compared to those obtained by other well-established methods. For more validations, three real-world engineering optimization problems extracted from IEEE-CEC 2011 are used. Again, CHIO is proved to be efficient. In conclusion, CHIO is a very powerful optimization algorithm that can be used to tackle many optimization problems across a wide variety of optimization domains.

Keywords Coronavirus · COVID-19 · Herd immunity · Optimization · Nature inspired · Metaheuristic

1 Introduction

Optimization is the process of finding the best configurations of some entities following limited resources respecting predefined constraints [1]. The optimization process

can be utilized in several research domains such as health, engineering, mathematics, economics, linguistics, and science to optimize (minimize or maximize) their objective [2]. In order to tackle optimization problems, two types of optimization methods emerge deterministic-based and

✉ Mohammed Azmi Al-Betar
mohbetar@bau.edu.jo

Zaid Abdi Alkareem Alyasseri
zaid.alyasseri@uokufa.edu.iq

Mohammed A. Awadallah
ma.awadallah@alaqsa.edu.ps

Iyad Abu Doush
idoush@auk.edu.kw

¹ Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, P.O. Box 50, Al-Huson, Irbid, Jordan

² Department of Information Technology - MSAI, College of Engineering and Information Technology Ajman University, Ajman, UAE

³ Center for Artificial Intelligence, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

⁴ ECE Department, Faculty of Engineering, University of Kufa, Najaf, Iraq

⁵ Department of Computer Science, Al-Aqsa University, P. O. Box 4051, Gaza, Palestine

⁶ Computer Science Department, American University of Kuwait, Salmiya, Kuwait

⁷ Computer Science Department, Yarmouk University, Irbid, Jordan

approximation-based [3]. Traditionally, deterministic-based methods are utilized to tackle some optimization problems with small dimensions and less complexity. Although they can find an exact solution for the optimization problem, they suffer from some dilemmas such as they cannot be used to tackle the NP-hard problems; they require heavy mathematical derivation, especially for gradient-based techniques; they can easily be stuck in a local optima [4]. Thus, they are inefficient in tackling real-world problems. Consequently, the optimization research communities tend their attentions to utilize approximation methods for their optimization problems.

Approximation methods have stochastic components to intelligently overcome the deterministic-based dilemmas. The traditional approximation-based methods were heuristic-based in which the optimization problem is constructively tackled element by element until a complete solution is reached [5]. Heuristic methods are problem-specific where each optimization problem has its own heuristic methods; for example, graph coloring problems use saturation algorithm heuristic methods [6]. The heuristic-based approaches although they can easily find a solution for the optimization problem, and the quality of the constructed solution is not unfortunately respected. The ultimate objective of tackling optimization problem is not only to find any solution, but also to find a “good enough” solution. Therefore, the emergence of metaheuristic algorithms as an efficient approximation-based method acquired high attention due to its superior advantages.

Metaheuristic-based approaches provide a general optimization framework that can iteratively improve the current solution(s) using intelligent knowledge-acquisition operators with stochastic features controlled by tuned parameters until an optimal solution is reached [2]. The operators of the powerful metaheuristic algorithms can efficiently explore several regions in the problem search space as well as exploit the accumulative knowledge acquired during the search process. Exploitation and exploration are contradictory, and achieving the right balance between them during the search is the main algorithmic challenge. The main advantages of these metaheuristic algorithms are [4]: (1) Their simplicity is adapted for a wide range of optimization problems with very small tweaking. They are dealt with the optimization problem as black-box mathematically formulated in terms of objective function and solution representation in which the problem-specific knowledge is not necessarily deeply studied. (2) They do not require mathematical-derivative information in the initial search. (3) They can easily escape the local optima using their stochastic-based components. Interestingly, the most metaheuristic-based algorithms are originated from nature-inspired phenomena which can be categorized into four classes:

evolutionary-based, swarm-based, physical-based, and human-based algorithms [7, 8]. These categories of metaheuristic-based algorithms are summarized in Table 1.

Evolutionary algorithms (EA) are naturally inspired from the evolution process initiated with a population of random individuals. Generation after generation, the gens of the parent individuals in the population are recombined and mutated to come up with offspring individuals which are adopted based on the survival-of-the-fittest principle in the natural selection scheme. The first developed EA is genetic algorithm (GA) proposed by John Henry Holland in 1960 to utilize the Darwinian principle of natural evolution [9]. Swarm-based algorithms are normally inspired by the social behavior of animal swarms. The main merit of such class is their ability to collaboratively survive. The earlier and considered first swarm-based algorithms are particle swarm optimization (PSO) [10] which imitates the bird flocking social behavior. The particles (solutions) fly around their environment (search space) searching for the optimal position (global best). During the flying process, the best positions (local best) in the path to the optimal position are recorded. Other base swarm-based optimizer is ant colony optimization (ACO) [11], artificial bee colony (ABC) [12], and many others summarized in Table 1. Physical-based algorithms are inspired by the physical laws appeared in the universe. The base algorithm of such category is simulated annealing (SA) which imitates thermodynamics process when the metals are cooled and annealed [13]. Other physical-based algorithms are summarized in Table 1. Finally, human-based algorithm stimulates the human’s behavior, lifestyle, or perception. The base method of such class is harmony search algorithm (HSA) in which a group of JAZZ musicians plays the notes of their instruments, practice after practice until a pleasing harmony (optimal solution) is obtained [14]. Other popular human-based algorithms are fireworks algorithm (FA) [15] and many others as reported in Table 1.

Apparently, there are a plethora of nature-inspired algorithms which can be efficiently used for a wide range of optimization problems. However, according to the No Free Lunch (NFL) theorem, the optimization algorithm cannot work efficiently for all types of optimization problems [51]. Furthermore, most deterministic or even heuristic optimization is not workable for problems with nonlinearity and multimodality. Therefore, the tremendous developments of metaheuristic algorithms, although come up with very powerful algorithms, there is still a window to develop other nature-based metaheuristic algorithms with intelligence characteristics with hope to tackle some complex optimization problems powerfully.

Nowadays, human-based nature-inspired phenomenon is emerged with as algorithms such as HSA or β HC

Table 1 Nurtured-inspired optimization algorithms

Nurtured-inspired categories	Nurtured-inspired algorithms
Evaluation-based	Genetic algorithm (GA) [9], evolution strategy (ES) [16], genetic programming (GP) [17], and biogeography-based optimizer (BBO) [18]
Swarm-based	Particle swarm optimization (PSO) [10], ant colony optimization (ACO) [11], cuckoo search (CS) [19], bat algorithm (BA) [20], ant lion optimizer (ALO) [21], butterfly optimization algorithm (BOA) [22], dragonfly algorithm (DA) [23], fruit fly optimization algorithm (FOA) [24], grey wolf optimizer (GWO) [4], krill herd algorithm (KHA) [25], red deer algorithm (RDA) [26], bird mating optimizer (BMO) [27], flower pollination algorithm (FPA) [28], monarch butterfly optimization (MBO) [29], moth-flame optimization algorithm (MFO) [30], whale optimization algorithm (WOA) [7], firefly algorithm (FA) [31], artificial bee colony (ABC) [12], salp swarm algorithm (SSA) [32], Harris hawks optimization (HHO) [33], and crow search algorithm (CSA) [34]
Physical-based	Simulated annealing (SA) [13], multi-verse optimizer (MVO) [35], sine cosine algorithm (SCA) [36], water cycle algorithm (WCA) [37], electromagnetism-like mechanism (EM) [38], gravitational search algorithm (GSA) [39], charged system search (CSS) [40], big bang–big crunch (BBBC) [41], and Henry gas solubility optimization (HGSO) [42]
Human-Based	Fireworks algorithm (FA) [15], harmony search algorithm (HSA) [14], wisdom of artificial crowds (WAC) [43], β -hill climbing (β HC) [44], Tabu search (TS) [45], group search optimizer (GSO) [46], interior search algorithm (ISA) [47], seeker optimization algorithm [48], social-based algorithm (SBA) [49], and mine blast algorithm (MBA) [50]

achieve pleasing results when compared to other nature-inspired algorithms. This paper proposed a new human-based nature-inspired algorithm coronavirus herd immunity optimizer (CHIO). Quite recently, the novel 2019 coronavirus evolved and start to spread from Wuhan, China, since December 2019. Consequently, the virus spread across several countries and the World Health Organization (WHO) announces the name of the new contagious disease to be Coronavirus Disease (COVID-19) [52].

There are several differences between COVID-19 and influenza such as transmission speed, mortality rate, and reproductive number [53]. The mortality rate higher than influenza 3–4%; the reproductive number is higher with 2–2.5; the transmission speed is faster 3–5 days [53].

Herd immunity is proposed as one of the techniques to control the COVID19 epidemic outbreak [54]. The proposed algorithm relies on the concept of how to best protect the community against the disease by converting the majority of the susceptible population which is not infected to become immuned. The phases of herd immunity can be summarized as follows [54–56] first, a group of infected people will infect another group of people. Second, a large number of infected people will recover and become immuned and a small number of people will die. Finally, after some time the majority of the population will become protected against the virus.

Recent research confirms the impact of herd immunity on the spread of COVID-19 as neutralizing antibodies are detected at many individuals [57, 58]. Another researcher recommended that herd immunity is a plausible strategy against COVID-19 [59].

CHIO is modeled for continuous optimization. Initially, the population individuals are randomly generated

and marked as susceptible, and very few members are marked infected. According to the basic reproduction rate (BR_r), the herd immunity of the population is evolved using three rules of spreading the pandemic following social distancing concepts: susceptible, infected, and immuned cases rules. The population members are moved from susceptible to infected and from infected to immuned according to herd immunity threshold by adopting the survival-of-the-fittest principle. A few numbers of infected individuals will reach the fatality state. The search is stopped when the population reaches the state of herd immunity. In order to verify the efficiency of CHIO, 23 well-known benchmark functions are used for evaluation. The effect of parameters on CHIO performance is initially studied. Then, alternative social distancing strategies are analyzed. Finally, the comparative evaluation against seven well-regarded methods is provided. The comparative results prove the viability of the proposed CHIO. For more validations, three real-world engineering optimization problems extracted from IEEE-CEC 2011 are used. Again, CHIO is proved to be efficient. In a nutshell, the new CHIO is a very powerful human-based optimization method that is pregnant with tremendous and successful developments for those who are interested to tackle their problems using natural-inspired metaheuristic-based algorithms.

The remaining sections of this paper are as follows: The proposed CHIO algorithm and the concepts behind it are introduced in Sect. 2. The performance of the proposed algorithm is evaluated and analyzed in Sect. 3. Finally, the conclusion and some future directions are provided in Sect. 4.

2 Coronavirus herd immunity optimizer

Viruses are normally spread and evolved very quickly among individuals of the population. The health communities normally use a vaccine to build immunity against viruses. However, new viruses need a period of time until their vaccine discovered. In the meanwhile, the health care organizations recommend to treat the virus in one of two ways: i) They isolate the infected individuals from their surrounding communities and isolate all the people they contact. ii) They use herd immunity principle to stop pandemics where herd immunity accrued when a significant portion of a population is immune resulting in protecting susceptible individuals.

2.1 Inspiration

Viruses can be transmitted biologically and it can be replicated by the amplifying hosts [60]. The novel 2019 coronavirus (2019-nCoV) evolved and start to spread from Wuhan, China, since December 2019. Consequently, the virus spread across several countries and the World Health Organization (WHO) announces the name of the new contagious disease to be Coronavirus Disease (COVID-19) [52]. As of March 27, 2020, the number of cases reaches 532,279 in 199 countries and territories around the world¹.

The incubation period of the COVID-19 varies between 2.1 and 11.1 days [52]. As to yet, no powerful remedy for COVID-19 is found [52]. The fatality rate of COVID-19 can range between 0.25 and 3.0% [54].

Herd immunity means that the population has a large number of people that are protected from being infected (either by vaccination or natural infection) and as a result, the disease will stop from spreading. This happened because more than 60% (i.e., herd immunity threshold) of the population is recovered from the infection. Herd immunity can affect the epidemic transmission as it can downsize the spread of the infection [56]. Herd immunity is proposed as one of the techniques to control the COVID-19 epidemic outbreak [54]. Note that this approach applies the Darwinian theory about **survival-of-the-fittest** principle.

According to the social distancing, the COVID-19 can be transmitted from human to human if the person is in close contact to another person (within 1.8 meters), by the droplets originated when the infected person sneezes or coughs, or when the person touches his/her mouth, nose, or

eyes after contacting a surface or object that has the virus on it². The governments followed two approaches to control the spread of COVID-19 as still there is no vaccination available the in country lockdown or herd immunity³.

A normal person that is not immuned against the virus is called susceptible. Once infected with the COVID-19, the person becomes a transmitting case. Now, based on the strength of the person's immune system, s/he can be either recovered (i.e., immuned) or unfortunately dead. Generally speaking, the elderly immune system is usually weaker than young people because they would have other diseases such as diabetes, cardiovascular diseases, or cancer. As a result, the person's age plays an important role in being recovered or not. The average age of the people who are died in Italy is 81 years [61].

According to many researchers [54–56], the main phases of achieving herd immunity are as follows:

- A large number of infected people infect another large group of people.
- Most of the infected people are recovered, and a small number are dead.
- After a while, most of the population will have immunity against the disease.

2.2 Herd immunity

Herd immunity refers to a situation where enough people in a population have immunity to the infection to be able to effectively stop that disease from spreading. For herd immunity, it does not matter whether the immunity comes from vaccination, or from the people who had the disease. The crucial thing is that they are immune.

As more people become infected with COVID-19, the disease caused by the virus, there will be more people who recover and who are then immune to future infection.

Herd immunity is affected by the basic reproduction rate, which represents how many people will be probably infected from the transmitting cases. This can indicate how quickly the disease will spread in the population. Generally speaking, when the number of immune cases reaches to be a large percentage of the population (i.e., larger than 60%) the population will be shielded from having more infected cases, and such percentage is called herd immunity threshold.

The transmitting cases pass the infection, and the immune system of the infected person will preserve an immunological memory of the disease. This will enable the infected person to become immune against that virus in the future, and thus, it will stop the disease from circulation.

The coronavirus herd immunity concept is mathematically modeled to develop the proposed optimization algorithm. The algorithm relies on the concept of how to best

¹ <https://www.worldometers.info/coronavirus/>.

² <https://www.cdc.gov/coronavirus/2019-ncov/prepare/transmission.html>

³ <https://economictimes.indiatimes.com/news/economy/policy/india-needs-to-achieve-herd-immunity-to-effectively-counter-covid-19-swa-minathan-aiyar/articleshow/74860557.cms>

protect the community against the disease by converting the majority of the susceptible population which is not infected to become immuned. As a result, even the remaining susceptible cases will not be infected because the immuned population will not be transmitting the disease anymore.

2.3 Population hierarchy

The herd immunity population individuals can be classified into three types [62]: susceptible, infected (or confirmed), and immuned (or recovered) individuals [63]. Figure 2 shows how the three types of individuals are distributed. The figure is represented as a tree where the root is the infected individual and the edges point to the contacted people. The right part of the figure shows that if the root individual is immuned, the virus will not be spread to its contacted individual. Therefore, it is functionally utilized as a firewall against virus pandemics. These types of individuals can be defined as follows:

- *susceptible individuals* These individuals are not infected by the virus, but it can be infected when they contact other infected individuals (i.e., did not follow the recommended social distancing).
- *infected individuals* The individuals of this type have a confirmed case where they can transmit the virus to other susceptible individuals who are in direct contact with according to the social distancing factor.

- *immuned individuals* The individuals who are categorized as immuned are protected against the virus, and they are not affected by infected individuals. This type of individual can help the population to stop spreading the pandemic as can be shown in Fig. 1.

In order to represent the hierarchy of population when the CHIO is designed in terms of optimization context, the susceptible individuals take a large portion from the population. The second portion of the population is marked as infected individuals which are initiated by a small number which represent the first infected individuals appeared in the population, and this portion of the population grows up if they did not follow the recommendation of social distancing until all the infected individuals are either immuned (i.e., recovered) or dead. The last portion of the population is the immuned individuals which are initiated by null and grow up according to how many are the recovered cases in the population. In the last course of the run, the majority of individuals are immuned, and therefore, the pandemic is stopped. In CHIO, the improvement process is derived by susceptible, infected, and immuned individuals as shown in CHIO procedure section below.

2.4 Social distancing

The concept of social distancing is used in the case of virus pandemics as a strategy to reduce the spreading of infections [64]. Normally, the governments and health care institutions suggest such action to advise individuals to

Fig. 1 Herd immunity

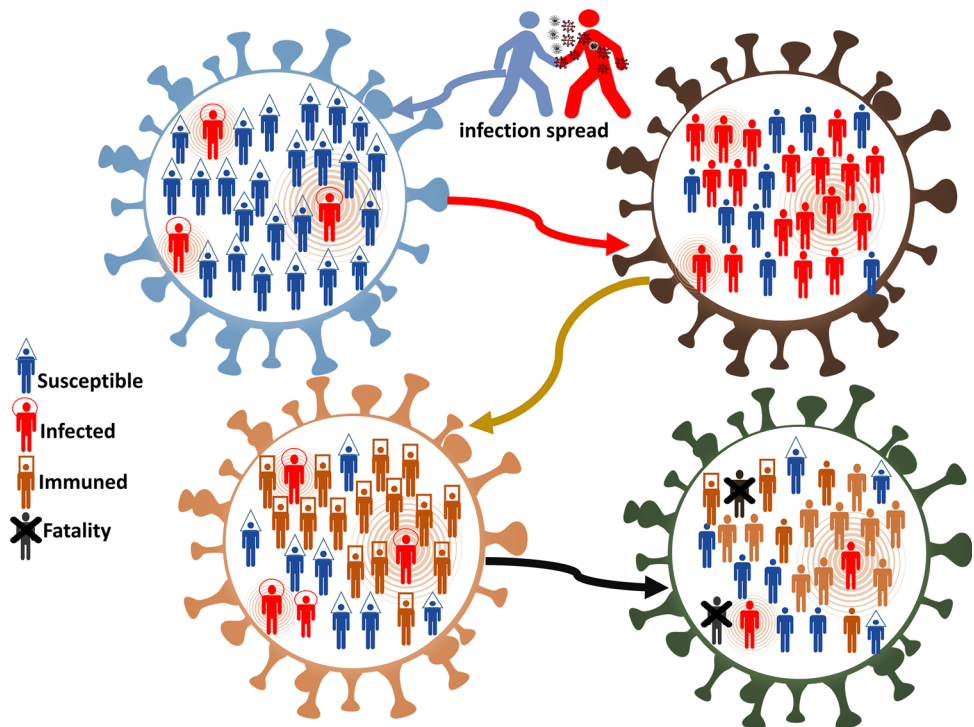


Fig. 2 Population hierarchy

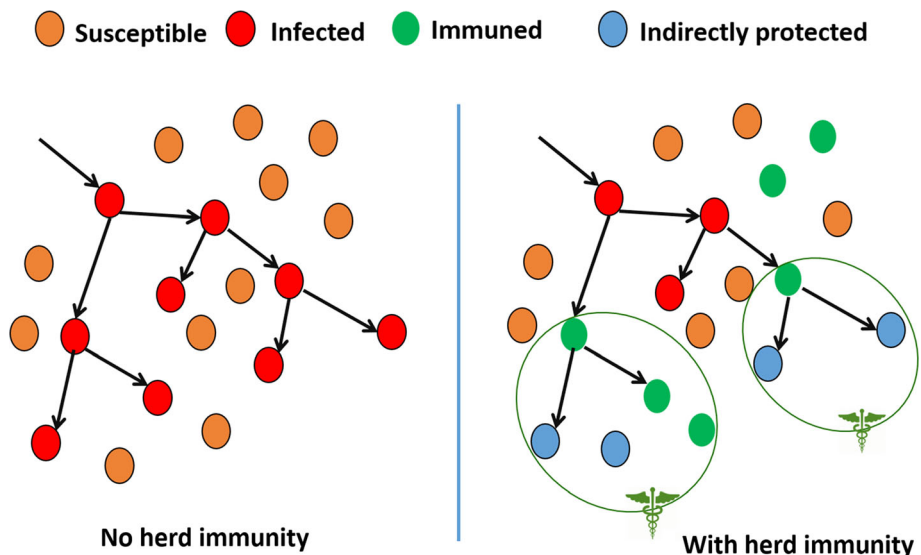
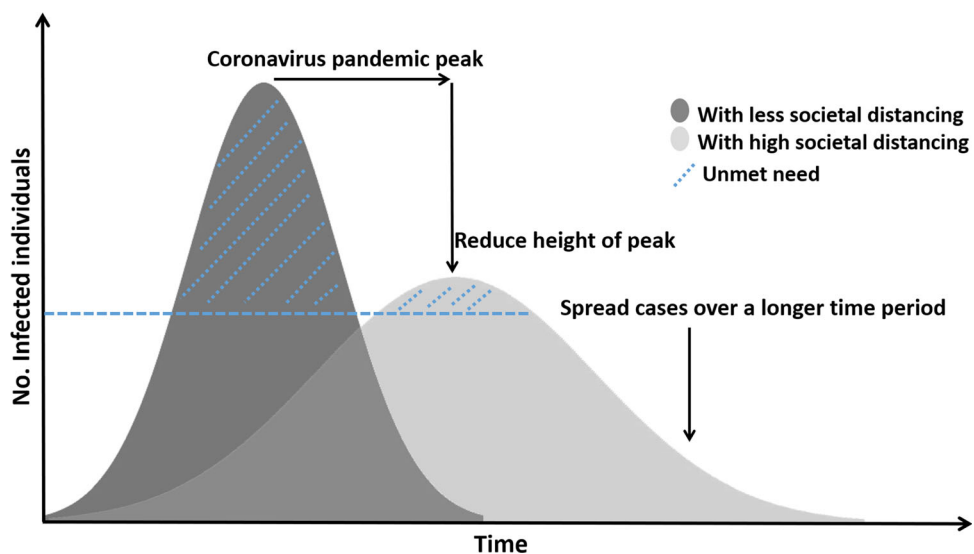


Fig. 3 Effect of social distancing on the spreading of virus pandemics in the population



keep a space of 2 meters (6.5 feet) between each other when going to crowded places [65]. Some other precautionary actions can avoid crowded places such as malls, schools, and universities.

The effect of social distancing is shown in Fig. 3. The spread of the disease would decline which can ultimately result in the outbreak of the pandemic. The transmission chains of the virus will be broken and would result in slowing down the spread of the disease and reaching the pandemic peak with a smaller number of infected cases [66]. Therefore, the country’s health care system would be able to continue to serve a smaller number of infected cases.

The two normal distribution charts presented in Fig. 3 show the effect of social distancing in controlling the spread of the pandemic. Apparently, the social distancing

would distribute the infected cases on a longer period of time which reduces the unmet need region (i.e., the health care services are not satisfactory).

In CHIO, the social distancing concept is achieved through taking the difference between the current individual and a selected individual from the population which might be susceptible, infected, or immuned.

2.5 Herd immunity real cases

The concept of using a controlled herd immunity to contain COVID-19 is used by some countries (e.g., UK and Sweden).

Sweden followed a controlled approach to herd immunity in which they keep schools, restaurants, and most businesses open and asked its citizens to voluntarily

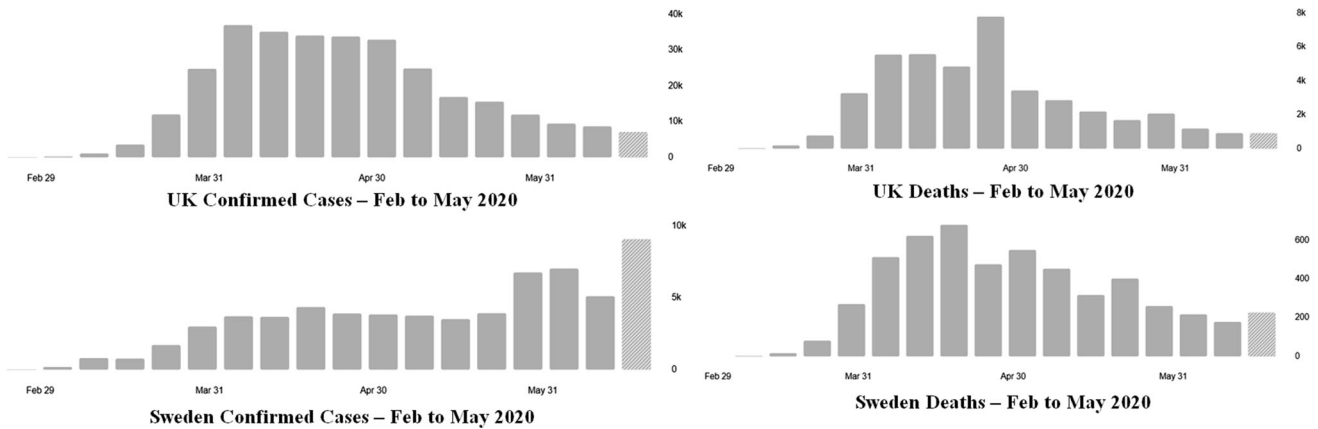


Fig. 4 COVID-19 confirmed and death cases in the UK and Sweden

Table 2 Bridge between COVID-19 and optimization concept

No	COVID-19 context	Optimization context
1	(infected, susceptible, immuned) Case	Solution
2	Social distancing	Pick random case and rely on the basic reproduction rate
3	Mortality rate	Reaching maximum age
4	Reproductive number	Basic reproduction rate
5	Transmission speed	Basic reproduction rate
6	Immunity rate	Fitness value
7	Possibility of infection	Weak fitness value and inherit COVID-19 features

practice social distancing [67]. The Swedish herd immunity takes longer than expected [68]. This is presented in Fig. 4 as it shows the number of confirmed cases and deaths in Sweden from February to May 2020. As of June 28, 2020, Sweden have 65,137 confirmed cases and 5,280 deaths [69].

Some countries (e.g., UK) allow the spread of the virus to increase the population herd immunity while protecting the elderly because they are the most vulnerable to this virus [54]. The UK government recommended using herd immunity to contain COVID-19 [70]. Figure 4 shows the decrease in the number of confirmed cases, and the number of deaths over time in the UK from February to May 2020. As of June 28, 2020, the UK have 310,254 confirmed cases and 43,514 deaths [71].

2.6 CHIO procedure

Herd immunity strategy is modeled in the proposed optimization algorithm. The concepts of COVID-19 are mapped to the optimization context in Table 2, and the CHIO is represented as a set of steps which thoroughly discussed below. The flowchart of CHIO algorithm is illustrated in Fig. 5 while CHIO is pseudocoded in Algorithm 1. The algorithm has six main steps discussed as follows:

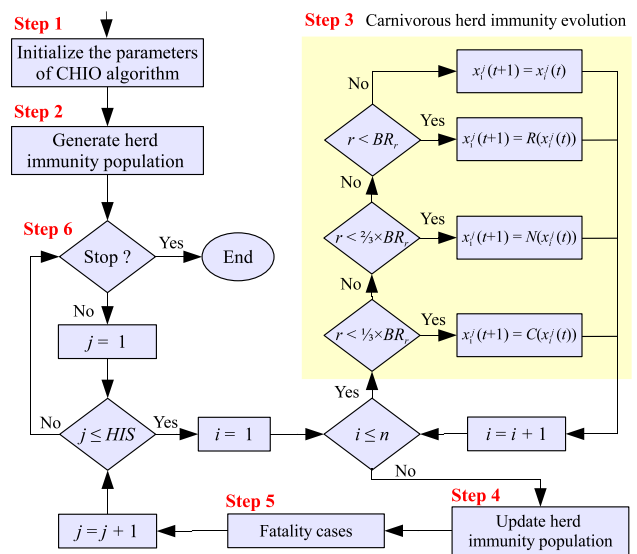


Fig. 5 Flowchart of CHIO algorithm

Step 1 Initialize parameters of CHIO and optimization problem In this step, the optimization problem is formulated in the context of objective function as follows:

$$\min_x f(\mathbf{x}) \quad \mathbf{x} \in [lb, ub] \tag{1}$$

where $f(\mathbf{x})$ is the objective function (or immunity rate) calculated for the case (or the individual) $\mathbf{x} = (x_1, x_2, \dots, x_n)$ where x_i is the gene (or the decision variable) indexed by i and n is the total number of genes in each individual. Note that the value range of each gene $x_i \in [lb_i, ub_i]$ where lb_i and ub_i represent the lower and upper bounds of gene x_i .

CHIO has four algorithmic parameters and two control parameters. The four algorithmic parameters are

- C_0 : which represents the number of initial infected cases where it is here initiated by one.
- Max_Itr : which is the maximum number of iterations.
- HIS : which is the population size.
- n : which is the problem dimensionality.

The CHIO has two main control parameters to be initialized in this step:

- Basic reproduction rate (BR_r) which controls the CHIO operators through spreading the virus pandemic between individuals.
- Maximum infected cases age (Max_{Age}): It determines the status of the infected cases where cases that reach Max_{Age} is either recovered or died.

Step 2 Generate herd immunity population Initially, CHIO randomly (or heuristically) generates a set of cases (individuals) as many as HIS . The generated cases are stored as two-dimensional matrix of size $n \times HIS$ in herd immunity population (HIP) as follows:

$$HIP = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \dots & \vdots \\ x_1^{HIS} & x_2^{HIS} & \dots & x_n^{HIS} \end{bmatrix} \tag{2}$$

where each row j represents a case \mathbf{x}^j , which is basically generated as follows: $x_i^j = lb_i + (ub_i - lb_i) \times U(0, 1)$, $\forall i = 1, 2, \dots, n$. The objective function (or immunity rate) for each case is calculated using equation (1). Furthermore, the status vector (\mathcal{S}) of length HIS for all cases in HIP is also initiated by either zero (susceptible case) or one (infected case). Note that the number of ones in (\mathcal{S}) is randomly initiated as many as C_0 .

Step 3 Coronavirus herd immunity evolution This is the main improvement loop of CHIO. The gene (x_i^j) of case \mathbf{x}^j is either remain the same or affected by social distancing using three rules according to the percentage of the BR_r as follows:

$$x_i^j(t+1) \leftarrow \begin{cases} x_i^j(t) & r \geq BR_r \\ C(x_i^j(t)) & r < \frac{1}{3} \times BR_r. \text{ //infected case} \\ N(x_i^j(t)) & r < \frac{2}{3} \times BR_r. \text{ //susceptible case} \\ R(x_i^j(t)) & r < BR_r. \text{ //immuned case} \end{cases} \tag{3}$$

where r generates a random number between 0 and 1. The three rules can be discussed as follows:

Infected case: Within the range of $r \in [0, \frac{1}{3}BR_r)$, the new gene value of $x_i^j(t+1)$ is affected by some social distancing which is achieved by the difference between current gene and a gene taken from an infected case \mathbf{x}^m such as

$$x_i^j(t+1) = C(x_i^j(t)) \tag{4}$$

where

$$C(x_i^j(t)) = x_i^j(t) + r \times (x_i^c(t) - x_i^j(t)) \tag{5}$$

Note that the value $x_i^c(t)$ is randomly chosen from any infected case \mathbf{x}^c based on the status vector (\mathcal{S}) such that $c = \{i | \mathcal{S}_i = 1\}$

Susceptible case: Within the range of $r \in [\frac{1}{3}BR_r, \frac{2}{3}BR_r)$, the new gene value of $x_i^j(t+1)$ is affected by some social distancing which is achieved by the difference between the current gene and a gene taken from a susceptible case \mathbf{x}^m such as

$$x_i^j(t+1) = N(x_i^j(t)) \tag{6}$$

where

$$N(x_i^j(t)) = x_i^j(t) + r \times (x_i^m(t) - x_i^j(t)) \tag{7}$$

Note that the value $x_i^m(t)$ is randomly spread from any susceptible case \mathbf{x}^m based on the status vector (\mathcal{S}) such that $m = \{i | \mathcal{S}_i = 0\}$.

Immuned case: Within the range of $r \in [\frac{2}{3}BR_r, BR_r)$, the new gene value of $x_i^j(t+1)$ is affected by some social distancing which is achieved by the difference between the current gene and a gene taken from an

immuned case \mathbf{x}^v such as

$$x_i^j(t+1) = R(x_i^j(t)) \tag{8}$$

where

$$R(x_i^j(t)) = x_i^j(t) + r \times (x_i^j(t) - x_i^m(t)) \tag{9}$$

Note that the value $x_i^v(t)$ is spread from the best immuned case \mathbf{x}^v based on the status vector (\mathcal{S}) such that

$$f(\mathbf{x}^v) = \arg \min_{j \sim \{k | \mathcal{S}_k = 2\}} f(\mathbf{x}^j).$$

Step 4 Update herd immunity population The immunity rate $f(\mathbf{x}^j(t+1))$ of each generated case $\mathbf{x}^j(t+1)$ is calculated and the current case $\mathbf{x}^j(t)$ is replaced by the generated case $\mathbf{x}^j(t+1)$, if better, such as $f(\mathbf{x}^j(t+1)) < f(\mathbf{x}^j(t))$. The age vector \mathcal{A}_j is also increased by one if $\mathcal{S}_j = 1$.

The status vector (\mathcal{S}_j) is updated for each case \mathbf{x}^j based on the herd immune threshold which utilizes the following equation:

$$\mathcal{S}_j \leftarrow \begin{cases} 1 & f(\mathbf{x}^j(t+1)) < \frac{f(\mathbf{x}^j(t+1))}{\Delta f(\mathbf{x})} \wedge \mathcal{S}_j = 0 \wedge is_Corona(\mathbf{x}^j(t+1)) \\ 2 & f(\mathbf{x}^j(t+1)) > \frac{f(\mathbf{x}^j(t+1))}{\Delta f(\mathbf{x})} \wedge \mathcal{S}_j = 1 \end{cases} \tag{10}$$

where $is_Corona(\mathbf{x}^j(t+1))$ is a binary value equal to one when the new case $\mathbf{x}^j(t+1)$ inherited a value from any infected case. The $\Delta f(\mathbf{x})$ is the mean value of the population immune rates such as $\frac{\sum_{i=1}^{HIS} f(x_i)}{HIS}$. Note that the individuals' immunity rate in the population will be changed based on the social distancing calculated before, if the newly generated individual immunity rate is better than the average immunity rate of the population. This means that we are starting to have a better-immuned population. If the newly generated population is strong enough to be immuned against the pandemic, then we reach the herd immunity threshold.

Step 5 Fatality cases In case the immunity rate ($f(\mathbf{x}^j(t+1))$) of the current infected case ($\mathcal{S}_j == 1$) could not improve for a certain number of iterations as specified by the parameter Max_Age (i.e., $\mathcal{A}_j \geq Max_Age$), then this case is considered died. After that, it is regenerated from scratch using $x_i^j(t+1) = lb_i + (ub_i - lb_i) \times U(0, 1)$, $\forall i = 1, 2, \dots, n$. Furthermore, \mathcal{A}_j and \mathcal{S}_j are set to zero. This can be useful to

diversify the current population and thus escaping local optima.

Step 6 Stop criterion CHIO repeats Step 3 to step 6 until the termination criterion which normally depends if the maximum number of iteration is reached. In this case, the total number of susceptible and immuned cases dominate the population. The infected cases are also disappeared.

Algorithm 1 CHIO pseudo-code

```

1: {----- Step 1: Initialize the CHIO parameters -----}
2: Initialize the parameters (HIS, Sr, and Max_Age).
3: {----- Step 2: Generate herd immunity population -----}
4: xij = lbi + (ubi - lbi) × U(0, 1),  ∀i = 1, 2, ..., n, and ∀j = 1, 2, ..., HIS.
5: Calculate the fitness of each search agent
6: Set Sj = 0  ∀j = 1, 2, ..., HIS.
7: Set Aj = 0  ∀j = 1, 2, ..., HIS.
8: {----- Step 3: Herd immunity evolution -----}
9: while (t ≤ Max_itr) do
10:   for j = 1 to HIS do
11:     is_Corona(xj(t + 1)) = false
12:     for i = 1 to N do
13:       if (r < 1/3 × BRr) then
14:         xij(t + 1) = C(xij(t))  {See Eq. (5)}
15:         is_Corona(xj(t + 1)) = true
16:       else if (r < 2/3 × BRr) then
17:         xij(t + 1) = N(xij(t))  {See Eq. (7)}
18:       else if (r < BRr) then
19:         xij(t + 1) = R(xij(t))  {See Eq. (9)}
20:       else
21:         xij(t + 1) = xij(t)
22:       end if
23:     end for
24:   {----- Step 4: Update herd immunity population -----}
25:   if (f(xj(t + 1)) ≤ f(xj(t))) then
26:     xj(t) = xj(t + 1)
27:   else
28:     Aj = Aj + 1
29:   end if
30:   if (f(xj(t + 1)) < f(xj(t+1))/Δf(x) ∧ Sj = 0 ∧ is_Corona(xj(t + 1)) then
31:     Sj = 1
32:     Aj = 1
33:   end if
34:   if (f(xj(t + 1)) > f(xj(t+1))/Δf(x) ∧ Sj = 1 then
35:     Sj = 2
36:     Aj = 0
37:   end if
38:   {----- Step 5: Fatality condition -----}
39:   if ((Aj ≥ Max_Age) ∧ (Sj == 1)) then
40:     xij = lbi + (ubi - lbi) × U(0, 1),  ∀i = 1, 2, ..., N.
41:     Sj = 0
42:     Aj = 0
43:   end if
44: end for
45: t = t + 1
46: end while
    
```

3 Experiments and results

In this section, the proposed CHIO algorithm is evaluated from various aspects by using a set of experiments conducted on 23 test functions. These test functions are circulated widely to evaluate newly established methods. The characteristics of these test functions are provided in Sect. 3.1. The experimental scenarios that are designed to study the behavior of CHIO algorithm are summarized in Sect 3.2. The sensitivity of CHIO to its control parameters: spreading rate (S_r) and maximum age of confirmed cases (Max_Age) are illustrated in Sects. 3.3 and 3.4, respectively.

Table 3 Characteristics of 23 test functions. (*n*: dimension, U: unimodal, M: multimodal)

Key Name	Test Functions	Range	<i>n</i>	C	$f(x^*)$
F1	Sphere	$\sum_{i=1}^n x_i^2$	30	U	0
F2	Schwefel's problem 2.22	$\sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	U	0
F3	Schwefel's problem 1.2	$\sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	U	0
F4	Schwefel's problem 2.21	$\max_i \{ x_i , 1 \leq i \leq n\}$	30	U	0
F5	Rosenbrock	$\sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	U	0
F6	Step	$\sum_{i=1}^n (x_i + 0.5)^2$	30	U	0
F7	Noise	$\sum_{i=1}^n ix_i^4 + random[0, 1)$	30	U	0
F8	Generalized Schwefel's problem	$\sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	M	-12569.5
F9	Rastrigin	$\sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	M	0
F10	Ackley	$-20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30	M	0
F11	Griewank	$\frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	M	0
F12	Generalized Penalized Function 1	$\frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4) y_i = 1 + \frac{x_i+1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 - a & < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	M	0
F13	Generalized Penalized Function 2	$0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] \right\} + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	M	0
F14	Shekel's Foxholes function	$\left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^n (x_i - a_{ij})^6} \right)^{-1}$	2	M	1
F15	Kowalik's function	$\sum_{i=1}^{11} \left[a_i - \frac{x_i (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	M	0.00030
F16	Six-hump camel back	$4x_1^2 - 2.1x_1^4 + \frac{5}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	M	-1.0316
F17	Branin	$(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10$	2	M	0.398
F18	Goldstein-Price function	$[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	M	3
F19	Hartman 1	$-\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2)$	3	M	-3.86

Table 3 (continued)

Key	Name	Test Functions	Range	n	C	$f(x^*)$
F20	Hartman 2	$-\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2\right)$	[0, 1]	6	M	- 3.32
F21	Shekel 1	$-\sum_{i=1}^5 [(X - a_i)(X - a_i)^7 + c_i]^{-1}$	[0, 10]	4	M	- 10.1532
F22	Shekel 2	$-\sum_{i=1}^7 [(X - a_i)(X - a_i)^7 + c_i]^{-1}$	[0, 10]	4	M	- 10.4028
F23	Shekel 3	$-\sum_{i=1}^{10} [(X - a_i)(X - a_i)^7 + c_i]^{-1}$	[0, 10]	4	M	- 10.5363

Thereafter, the effect of the social distancing strategies on the convergence behavior of the herd immunity evolution is analyzed in Sect. 3.5. Finally, the comparative evaluation against the state-of-the-art algorithms is discussed in Sect. 3.6.

3.1 Test functions

In order to evaluate the performance of the proposed CHIO, 23 common test functions are considered. All of these test functions are minimization problems, which are different in size and complexity. Table 3 provides the main characteristics of test functions used which includes the functions names, the test function key, the mathematical formulation of each test function, the rang which determines the boundary of the search space, the function dimensions (n), and the optimum solution $f(x^*)$. The category of each test function is also provided: unimodal (U) and multimodal (M). It should be noted that the unimodal test functions have a single optimum, while the multimodal test functions have more than one optimum. The unimodal test functions are used to evaluate the exploitation ability of the optimization algorithms, while the multimodal test functions are used to evaluate the exploration ability of the optimization algorithms [42]. As shown in Table 3, F1 – F7 are categorized as unimodal test functions, while F8 – F23 are categorized as multimodal test functions. Furthermore, The dimensions of the test functions F14 – F23 are fixed.

Figure 6 shows the 2D search space for each benchmark function and the convergence behavior of CHIO of the first solution in the first dimension for each benchmark function.

3.2 Experimental settings

The evaluations of the CHIO performance are tested and analyzed using different convergence scenarios. The sensitivity of CHIO to its two control parameters (i.e., BR_r and Max_{Age}) is studied as shown in Table 4: Sen1–Sen8. The convergence scenarios are conducted based on *ad hoc* strategy where the first set of scenarios study one operator and the remaining operators are remaining constant.

The effect of basic reproduction rate (BR_r) on the convergence of CHIO is studied using four values ($BR_r = 0.005$, $BR_r = 0.05$, $BR_r = 0.01$, and $BR_r = 0.5$) from Sen1 to Sen4. Note that BR_r determines the percentage of the population affected by the coronavirus pandemic. The smaller the value is, the slower the coronavirus spreading will be.

The effect of the maximum infected age (Max_{Age}) on the convergence of CHIO is studied using four numbers

Fig. 6 Functions and convergence plots

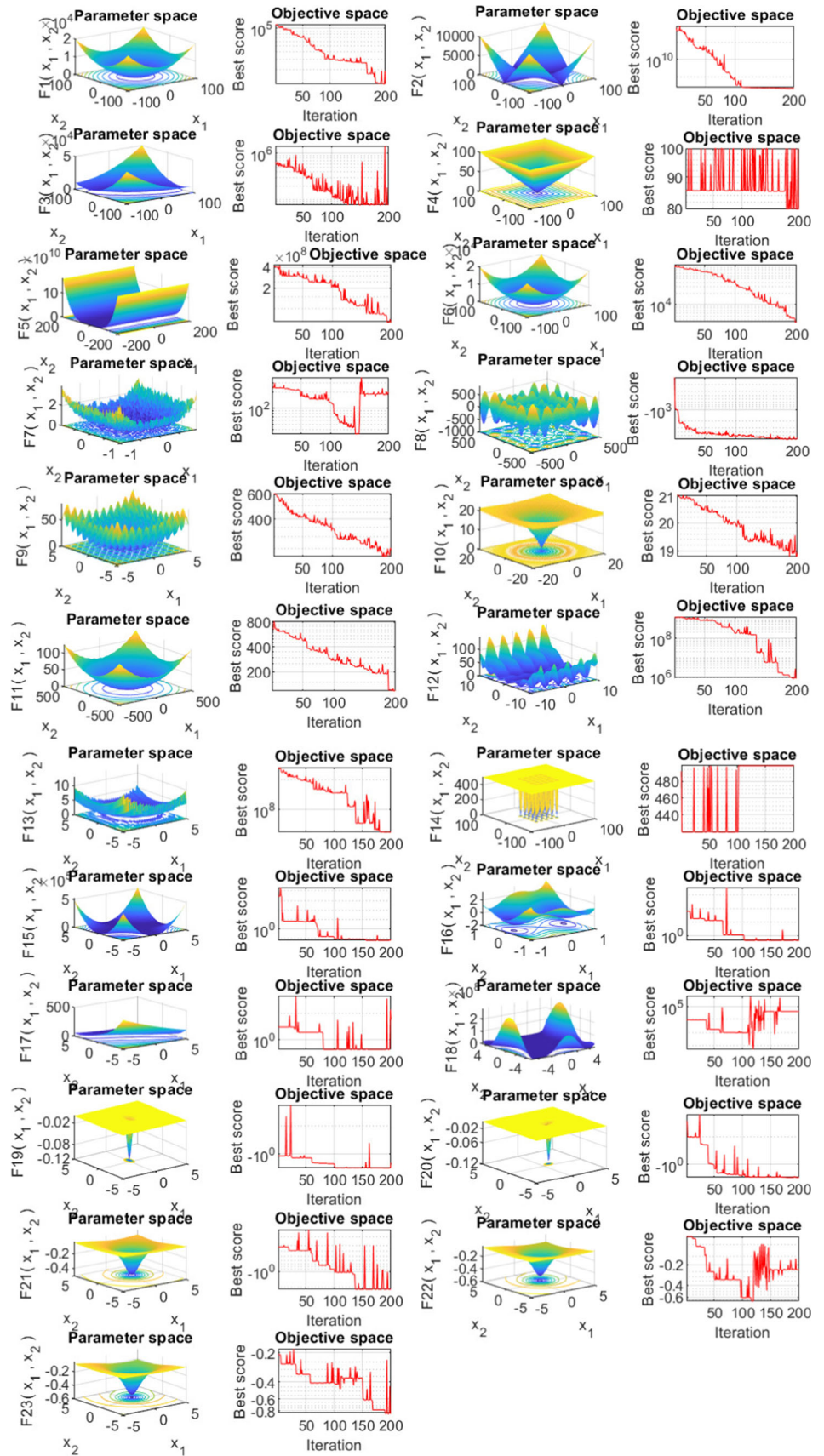


Table 4 Twelve experimental scenarios designed to evaluate the sensitivity of the proposed CHIO to its parameters

Scenario	BR_r	Max_{Age}	infected	susceptible	immuned	Notes
Sen1	0.005	100	Random	Random	Best	
Sen2	0.05					
Sen3	0.1					
Sen4	0.5					
Sen5		50	Random	Random	Best	
Sen6		100				Sen6 = Sen 2
Sen7		300				
Sen8		500				
Sen9			Random	Random	Random	
Sen10			Random	Random	Best	Sen10 = Sen 6
Sen11			Random	Best	Random	
Sen12			Random	Best	Best	

($Max_{Age} = 50$, $Max_{Age} = 100$, $Max_{Age} = 300$, and $Max_{Age} = 500$) from Sen5 to Sen8. As remembering, the Max_{Age} is the maximum number of iterations where the infected solution remains unimproved. Therefore, a new solution is constructed from scratch to replace the discarded solution.

The last four convergence scenarios (i.e., Sen9 – Sen12) are designed to study the social distancing strategy. Recall, the social distancing in herd immunity evolution step has three main rules for infection: susceptible, infected, and immuned. The first two rules update the generated solution based on the difference between the current solution and a randomly selected solution. The last rule updates the generated solution based on the difference between the current solution and the best solution. Sen9 – Sen12 study four possible combinations of social distancing strategies: random–random–random, random–random–best, random–best–random, and random–best–best.

Note that CHIO replicates 30 runs for each experimental scenario, the herd immunity size (HIS) used is 30, and the maximum number of iteration (i.e., Max_{itr}) is equal to 100,000. The results are statistically recorded in terms of best, mean, worst, and standard deviation for all designed scenarios.

3.3 Effect of the basic reproduction rate (BR_r)

The effect of the basic reproduction rate (BR_r) on the performance of CHIO using various values of BR_r (i.e., $BR_r = 0.005$, $BR_r = 0.05$, $BR_r = 0.1$, and $BR_r = 0.5$) has been studied here. The value of the parameter BR_r determines the speed of spreading the coronavirus pandemic across the population. A higher value of BR_r leads to a higher rate of spreading the disease and thus the exploration becomes large. The results recorded in Table 5

summarize the best, worst, mean, and standard deviation (Stdev.) of the 23 test functions over 30 replicated runs.

As can be noticed, CHIO in Sen2 can achieve the most best mean results. This is because a larger value of BR_r increases the exploration, and thus, the search will require longer time to converge. On the other hand, when the value of $BR_r = 0.001$, the exploration source of the generated individuals is not that much, and thus, fast convergence will be occurred.

As remembering, the first seven benchmark functions are unimodal and they have higher complexity to be solved due to their ruggedness in the search space. Sen2 can outperform other three scenarios in five out of seven benchmark functions. The next six benchmark functions (F8–F13) are multimodal, and their dimensions are flexible. The search space of this type of benchmark functions is not that complex in comparison with unimodal benchmark functions. Interestingly, Sen2 can excel all other designed scenarios for all multimodal flexible dimensions benchmark functions. The search space of the last ten benchmark functions of type multimodal and fixed dimensions is the simplest where Sen2 can outperform the other three scenarios in eight of ten benchmark functions. Apparently, using $BR_r = 0.01$ represented in Sen2 empower the convergence behavior of CHIO to achieve the right balance between the exploration and exploitation of the search space and thus the best performance. Therefore, the value of $BR_r = 0.01$ will be used in the experiments of the upcoming designed scenarios.

3.4 Effect of Max_{Age}

The effect of the maximum age of the infected cases (Max_{Age}) on the performance of CHIO using various values of ($Max_{Age} = 50$, $Max_{Age} = 100$, $Max_{Age} = 300$, and $Max_{Age} = 500$) is investigated in this subsection. The value

Table 5 Performance of CHIO algorithm using different settings of BR_r

Function		Sen1	Sen2	Sen3	Sen4
F1	Best	1.1915E−73	0.0000E+00	0.0000E+00	0.0000E+00
	Worst	9.3388E−02	2.1364E−16	2.0496E−04	5.3245E+03
	Mean	1.7232E−02	7.1578E−18	6.9120E−06	1.7251E+03
	Stdev.	1.7232E−02	3.8999E−17	3.7407E−05	1.4528E+03
F2	Best	1.6108E−35	3.6009E−179	5.9705E−27	7.2303E−14
	Worst	7.1296E+02	2.2255E−09	3.2453E−03	1.5847E+01
	Mean	2.8919E+01	1.0336E−10	1.1464E−04	6.6956E+00
	Stdev.	1.3216E+02	4.1628E−10	5.9190E−04	5.2267E+00
F3	Best	1.5125E+03	6.8219E−01	2.7157E−04	6.8886E−12
	Worst	6.5132E+03	1.4758E+02	1.1900E+02	4.1768E+03
	Mean	3.3855E+03	5.3496E+01	3.8246E+01	5.3926E+02
	Stdev.	1.1490E+03	4.1604E+01	3.6283E+01	1.1309E+03
F4	Best	2.0556E−01	1.4323E−14	1.7012E−20	4.1184E−29
	Worst	6.5107E+01	8.6072E−02	1.7094E−01	2.6488E+01
	Mean	5.7485E+00	1.2869E−02	3.6961E−02	5.0840E+00
	Stdev.	1.5723E+01	2.0007E−02	5.3978E−02	7.1575E+00
F5	Best	6.5458E−04	2.0602E−04	5.4126E−03	2.3175E+01
	Worst	9.6098E+01	1.2583E+00	1.8537E+01	1.1408E+06
	Mean	1.5704E+01	3.0925E−01	3.5249E+00	1.4873E+05
	Stdev.	2.4355E+01	4.4332E−01	4.5043E+00	2.7048E+05
F6	Best	0.0000E+00	0.0000E+00	0.0000E+00	1.2089E−03
	Worst	5.3341E−02	2.1121E−03	4.3919E−05	5.8506E+03
	Mean	1.7786E−03	7.0403E−05	1.4664E−06	1.2989E+03
	Stdev.	9.7386E−03	3.8561E−04	8.0180E−06	1.3499E+03
F7	Best	1.8090E−02	2.2048E−03	2.4062E−03	8.4391E−04
	Worst	5.7765E−02	7.5511E−03	8.9773E−03	1.5651E+00
	Mean	3.2510E−02	4.5852E−03	5.4953E−03	3.1551E−01
	Stdev.	9.9587E−03	1.3483E−03	1.6943E−03	4.0446E−01
F8	Best	− 1.2569E+04	− 1.2569E+04	− 1.2569E+04	− 1.2569E+04
	Worst	− 1.2451E+04	− 1.2569E+04	− 1.1401E+04	− 8.5119E+03
	Mean	− 1.2565E+04	− 1.2569E+04	− 1.2357E+04	− 1.1176E+04
	Stdev.	2.1538E+01	0.0000E+00	3.4241E+02	9.5714E+02
F9	Best	0.0000E+00	0.0000E+00	0.0000E+00	2.1068E−06
	Worst	1.3049E−04	2.1364E−16	2.9849E+00	1.0683E+02
	Mean	4.3838E−06	7.1578E−18	4.6432E−01	2.8554E+01
	Stdev.	2.3818E−05	3.8999E−17	8.5604E−01	2.7109E+01
F10	Best	2.2204E−14	1.5099E−14	1.5099E−14	6.6650E−05
	Worst	3.6980E−02	2.9142E−04	9.3130E−01	1.4479E+01
	Mean	1.2502E−03	1.0244E−05	3.1072E−02	5.3867E+00
	Stdev.	6.7487E−03	5.3185E−05	1.7003E−01	3.4723E+00
F11	Best	0.0000E+00	0.0000E+00	0.0000E+00	5.3118E−01
	Worst	5.9121E−02	1.1316E−05	3.6524E−02	4.9415E+01
	Mean	3.2896E−03	4.4131E−07	1.8787E−03	1.5387E+01
	Stdev.	1.1039E−02	2.0830E−06	7.4673E−03	1.1417E+01
F12	Best	1.5705E−32	1.5705E−32	1.5705E−32	1.5786E−32
	Worst	3.6124E−05	1.0144E−15	2.0264E−12	2.7489E+03
	Mean	2.5452E−06	3.3819E−17	6.9156E−14	2.1803E+02
	Stdev.	7.4542E−06	1.8520E−16	3.6972E−13	6.5459E+02

Table 5 (continued)

Function		Sen1	Sen2	Sen3	Sen4
F13	Best	1.3498E−32	1.3498E−32	1.3498E−32	1.7920E−30
	Worst	3.1017E−02	8.9261E−29	9.0058E−03	1.5117E+06
	Mean	1.3740E−03	2.9886E−30	3.0019E−04	9.5468E+04
	Stdev.	5.7660E−03	1.6294E−29	1.6442E−03	2.9145E+05
F14	Best	9.9800E−01	9.9800E−01	9.9800E−01	9.9800E−01
	Worst	9.9800E−01	9.9800E−01	9.9800E−01	9.9800E−01
	Mean	9.9800E−01	9.9800E−01	9.9800E−01	9.9800E−01
	Stdev.	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F15	Best	5.5859E−04	3.1027E−04	3.1755E−04	3.0749E−04
	Worst	1.0558E−03	7.4299E−04	6.1261E−04	7.2917E−04
	Mean	7.4781E−04	4.8287E−04	4.2580E−04	3.6978E−04
	Stdev.	1.0901E−04	1.1762E−04	8.8041E−05	1.0701E−04
F16	Best	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00
	Worst	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00
	Mean	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00
	Stdev.	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F17	Best	3.9789E−01	3.9789E−01	3.9789E−01	3.9789E−01
	Worst	3.9789E−01	3.9789E−01	3.9789E−01	3.9789E−01
	Mean	3.9789E−01	3.9789E−01	3.9789E−01	3.9789E−01
	Stdev.	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F18	Best	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00
	Worst	3.0031E+00	3.0000E+00	3.0000E+00	3.0000E+00
	Mean	3.0006E+00	3.0000E+00	3.0000E+00	3.0000E+00
	Stdev.	7.3438E−04	0.0000E+00	0.0000E+00	0.0000E+00
F19	Best	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00
	Worst	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00
	Mean	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00
	Stdev.	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F20	Best	− 3.3220E+00	− 3.3220E+00	− 3.3220E+00	− 3.3220E+00
	Worst	− 3.2584E+00	− 3.3220E+00	− 3.3220E+00	− 3.3220E+00
	Mean	− 3.3150E+00	− 3.3220E+00	− 3.3220E+00	− 3.3220E+00
	Stdev.	1.8824E−02	0.0000E+00	0.0000E+00	0.0000E+00
F21	Best	− 1.0153E+01	− 1.0153E+01	− 1.0153E+01	− 1.0153E+01
	Worst	− 7.8455E+00	− 1.0153E+01	− 1.0153E+01	− 1.0153E+01
	Mean	− 9.6105E+00	− 1.0153E+01	− 1.0153E+01	− 1.0153E+01
	Stdev.	8.4268E−01	0.0000E+00	0.0000E+00	0.0000E+00
F22	Best	− 1.0403E+01	− 1.0403E+01	− 1.0403E+01	− 1.0403E+01
	Worst	− 6.3599E+00	− 1.0403E+01	− 1.0403E+01	− 1.0403E+01
	Mean	− 9.4503E+00	− 1.0403E+01	− 1.0403E+01	− 1.0403E+01
	Stdev.	1.2046E+00	0.0000E+00	0.0000E+00	0.0000E+00
F23	Best	− 1.0536E+01	− 1.0536E+01	− 1.0536E+01	− 1.0536E+01
	Worst	− 7.3257E+00	− 1.0536E+01	− 1.0536E+01	− 1.0536E+01
	Mean	− 9.6771E+00	− 1.0536E+01	− 1.0536E+01	− 1.0536E+01
	Stdev.	9.9204E−01	0.0000E+00	0.0000E+00	0.0000E+00

Bold font refers to the best recorded result

Table 6 Performance of CHIO algorithm using different settings of Max_{Age}

Function		Sen5	Sen6	Sen7	Sen8
F1	Best	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	Worst	7.2289E−18	2.1364E−16	3.2278E−05	5.1485E+00
	Mean	2.4096E−19	7.1578E−18	1.4947E−06	1.7162E−01
	Stdev.	1.3198E−18	3.8999E−17	6.1433E−06	9.3998E−01
F2	Best	7.1594E−272	3.6009E−179	9.5408E−111	2.0725E−273
	Worst	1.0062E−03	2.2255E−09	2.6313E−02	2.3157E−02
	Mean	5.2922E−05	1.0336E−10	9.3354E−04	2.1896E−03
	Stdev.	2.0701E−04	4.1628E−10	4.7953E−03	6.3889E−03
F3	Best	2.5397E+00	6.8219E−01	4.6399E−01	2.3176E−01
	Worst	1.8566E+02	1.4758E+02	1.6224E+02	1.2954E+02
	Mean	6.8452E+01	5.3496E+01	5.4905E+01	4.4860E+01
	Stdev.	5.1942E+01	4.1604E+01	4.7754E+01	3.9250E+01
F4	Best	1.5053E−14	1.4323E−14	9.3573E−15	8.5950E−15
	Worst	1.3080E−01	8.6072E−02	1.2431E−01	7.4766E−02
	Mean	2.2018E−02	1.2869E−02	1.8668E−02	1.0511E−02
	Stdev.	3.6482E−02	2.0007E−02	3.0738E−02	1.9717E−02
F5	Best	6.7196E−04	2.0602E−04	3.1663E−03	1.1587E−03
	Worst	1.0773E+01	1.2583E+00	1.2215E+04	5.7580E+00
	Mean	7.3006E−01	3.0925E−01	4.0829E+02	1.3770E+00
	Stdev.	1.9983E+00	4.4332E−01	2.2299E+03	1.6805E+00
F6	Best	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	Worst	2.1015E−16	2.1121E−03	9.2760E+01	3.9408E−03
	Mean	7.0050E−18	7.0403E−05	3.0920E+00	1.5185E−04
	Stdev.	3.8368E−17	3.8561E−04	1.6936E+01	7.2334E−04
F7	Best	2.4045E−03	2.2048E−03	2.9042E−03	2.2303E−03
	Worst	8.9010E−03	7.5511E−03	1.1137E−02	2.3980E−02
	Mean	5.1565E−03	4.5852E−03	5.3464E−03	6.7864E−03
	Stdev.	1.5305E−03	1.3483E−03	1.6764E−03	5.1579E−03
F8	Best	− 1.2569E+04	− 1.2569E+04	− 1.2569E+04	− 1.2569E+04
	Worst	− 1.2451E+04	− 1.2569E+04	− 1.1148E+04	− 1.1912E+04
	Mean	− 1.2565E+04	− 1.2569E+04	− 1.2487E+04	− 1.2520E+04
	Stdev.	2.1544E+01	0.0000E+00	2.6373E+02	1.3638E+02
F9	Best	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	Worst	3.4106E−13	2.1364E−16	9.9502E−01	9.0767E+00
	Mean	3.4106E−14	7.1578E−18	3.3179E−02	8.3617E−01
	Stdev.	6.4380E−14	3.8999E−17	1.8166E−01	2.2866E+00
F10	Best	1.5099E−14	1.5099E−14	1.1546E−14	1.5099E−14
	Worst	3.9968E−14	2.9142E−04	2.6673E−03	5.0244E−04
	Mean	2.7534E−14	1.0244E−05	1.5374E−04	3.2880E−05
	Stdev.	7.1514E−15	5.3185E−05	5.3564E−04	1.0915E−04
F11	Best	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	Worst	7.4057E−03	1.1316E−05	1.3499E−04	1.1164E+00
	Mean	2.4690E−04	4.4131E−07	1.2273E−05	3.9179E−02
	Stdev.	1.3521E−03	2.0830E−06	3.1948E−05	2.0360E−01
F12	Best	1.5705E−32	1.5705E−32	1.5705E−32	1.5705E−32
	Worst	7.5517E−16	1.0144E−15	5.1535E−06	6.2437E−06
	Mean	2.5172E−17	3.3819E−17	1.7178E−07	2.2798E−07
	Stdev.	1.3787E−16	1.8520E−16	9.4090E−07	1.1387E−06

Table 6 (continued)

Function		Sen5	Sen6	Sen7	Sen8
F13	Best	1.3498E−32	1.3498E−32	1.3498E−32	1.3498E−32
	Worst	4.5096E−20	8.9261E−29	2.1435E−08	9.4083E−07
	Mean	1.5038E−21	2.9886E−30	7.4931E−10	5.0727E−08
	Stdev.	8.2333E−21	1.6294E−29	3.9111E−09	1.9867E−07
F14	Best	9.9800E−01	9.9800E−01	9.9800E−01	9.9800E−01
	Worst	9.9800E−01	9.9800E−01	9.9800E−01	9.9800E−01
	Mean	9.9800E−01	9.9800E−01	9.9800E−01	9.9800E−01
	Stdev.	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F15	Best	3.5295E−04	3.1027E−04	3.1141E−04	3.1057E−04
	Worst	8.0620E−04	7.4299E−04	6.3479E−04	6.8806E−04
	Mean	5.2965E−04	4.8287E−04	4.3896E−04	4.3822E−04
	Stdev.	1.1758E−04	1.1762E−04	9.8883E−05	9.3258E−05
F16	Best	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00
	Worst	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00
	Mean	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00
	Stdev.	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F17	Best	3.9789E−01	3.9789E−01	3.9789E−01	3.9789E−01
	Worst	3.9789E−01	3.9789E−01	3.9789E−01	3.9789E−01
	Mean	3.9789E−01	3.9789E−01	3.9789E−01	3.9789E−01
	Stdev.	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F18	Best	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00
	Worst	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00
	Mean	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00
	Stdev.	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F19	Best	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00
	Worst	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00
	Mean	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00
	Stdev.	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F20	Best	− 3.3220E+00	− 3.3220E+00	− 3.3220E+00	− 3.3220E+00
	Worst	− 3.3220E+00	− 3.3220E+00	− 3.3220E+00	− 3.3220E+00
	Mean	− 3.3220E+00	− 3.3220E+00	− 3.3220E+00	− 3.3220E+00
	Stdev.	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F21	Best	− 1.0153E+01	− 1.0153E+01	− 1.0153E+01	− 1.0153E+01
	Worst	− 1.0153E+01	− 1.0153E+01	− 1.0153E+01	− 1.0153E+01
	Mean	− 1.0153E+01	− 1.0153E+01	− 1.0153E+01	− 1.0153E+01
	Stdev.	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F22	Best	− 1.0403E+01	− 1.0403E+01	− 1.0403E+01	− 1.0403E+01
	Worst	− 1.0403E+01	− 1.0403E+01	− 1.0403E+01	− 1.0403E+01
	Mean	− 1.0403E+01	− 1.0403E+01	− 1.0403E+01	− 1.0403E+01
	Stdev.	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F23	Best	− 1.0536E+01	− 1.0536E+01	− 1.0536E+01	− 1.0536E+01
	Worst	− 1.0536E+01	− 1.0536E+01	− 1.0536E+01	− 1.0536E+01
	Mean	− 1.0536E+01	− 1.0536E+01	− 1.0536E+01	v1.0536E+01
	Stdev.	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00

Bold font refers to the best recorded result

of the parameter Max_{Age} determines the fatality condition of the infected cases. The infected cases that are reached the Max_{Age} threshold without improvement will be

destroyed, and a new solution will be rebuilt from scratch. The results recorded in Table 6 summarize the best, worst, mean, and standard deviation (Stdev.) of the 23 test

functions over 30 replicated runs. The best solution, as well as the best mean obtained, is highlighted in **bold** font. Note that in the results, the lowest is the best.

The results in Table 6 show that when the value of Max_{Age} is equal to 100, the best mean results are obtained. Note that the best results are highlighted in **bold**. The Max_{Age} refers to the number of iterations for which the infected cases remain unimproved. This operation can be considered as a source of exploration. The smaller the value of Max_{Age} , the higher the exploration. The value of $Max_{Age} = 100$ seems reasonable to diversify the search. However, there is no significant effect on the value of Max_{Age} on the results produced.

3.5 Study of social distancing strategies in herd immunity evolution

The effect of social distancing strategies on the herd immunity evolution step is studied using the last four convergence scenarios (i.e., Sen9 – Sen12). The social distancing in herd immunity evolution step has three main rules for infection: infected, susceptible, and immuned.

Sen9 changes the functionality of the social distancing strategy where the three rules update the generated solution based on the difference between the current solution and a randomly selected solution which is called a random–random–random social distancing strategy. Sen10 changes the functionality of the social distancing strategy and is called a random–random–best where the infected case and the susceptible case use a randomly selected solution while the immuned cases use the difference between the current solution and the best solution to update the values of the newly generated solution.

Sen11 assumed that the functionality of the social distancing strategy updates the newly generated solutions based on the random–best–random strategy where the infected case uses a randomly selected solution while the susceptible case uses the difference between the current solution and the best solution to update the values of the newly generated solution, while the immuned cases use the difference between the current solution and the randomly selected solution to update the values of the newly generated solution. The last scenario (Sen12) adopts random–best–best social distancing strategy where the infected rule uses the difference between the current solution and the randomly selected solution to update the values of the newly generated solution, while the rules of the susceptible and immuned cases use the difference between the current solution and the best solution to update the values of the newly generated solution.

The results recorded in Table 5 summarize the best, worst, mean, and standard deviation (Stdev.) of the 23 test

functions over 30 replicated runs. The best solution, as well as the best mean obtained, is highlighted in **bold** font. Note that in the results, the lowest is the best. The results summarized in Table 7 show that Sen9 can achieve the best mean results for 22 out of 23 benchmark functions. Recall, Sen9 uses a random–random–random social distancing strategy. This means that the stochastic strategy in social distancing is very efficient and empower the convergence strength of the proposed CHIO.

The convergence behaviour of CHIO using different social distancing strategies are illustrated in Fig. 7. As can be noticed, Sen9 adopted random–random–random social distance strategy shows the best convergence behaviour in comparison with other three social distancing strategies.

3.6 Comparison with the swarm-based optimization algorithms

In this section, the performance of the proposed CHIO algorithm is compared to seven swarm-based algorithms. The flower pollination algorithm (FPA) [28], bat algorithm (BA) [20], artificial bee colony (ABC) [12], sine cosine algorithm (SCA) [36], Harris hawks optimization (HHO) [33], salp swarm algorithm (SSA), and JAYA algorithm [72] are utilized to deeply investigate the efficiency of the proposed CHIO algorithm when compared against these algorithms.

It should be noted that all these algorithms are experimented using the same conditions in order to ensure fairness. These conditions include the maximum number of iterations is 100,000, the size of the population is 30, and the number of runs is 30 times. The algorithmic parameters of the other comparative algorithms are $p=0.8$ in FPA; $f_{min} = 0, f_{max} = 1, A_j = 0.95, r_j^0 = 0.1, \alpha = 0.95, \gamma = 0.95$, and $\epsilon = 0.001$ in BA; $a = 2$ in SCA; $limit = \text{Number of onlooker bees} \times n$ in ABC; and $v_0 = 0$ in SSA.

Table 8 exposes the experimental results of the proposed CHIO algorithm as well as the other comparative methods in terms of the best results, worst results, mean of the results, and the standard derivation when running these algorithms 30 independent runs. In Table 8, the best results are highlighted in bold font.

In terms of the best of the results, it can be observed from the results provided in Table 8 that all the algorithms obtained the same optimal results on four test functions (i.e., F14, F16, F17, and F18). This is because the dimensions of the solutions in these test functions are small, and the algorithms did not need big effort to reach the optimal results. On other hand, the HHO algorithm obtained the best results in 19 test functions, and this is the highest number of best results reached by one of the comparative algorithms, while the JAYA and FPA algorithms achieved

Table 7 Performance of CHIO algorithm using different social distancing strategies

Function		Sen 9	Sen 10	Sen 11	Sen 12
F1	Best	0.0000E+00	0.0000E+00	0.0000E+00	5.6986E−238
	Worst	6.9210E−192	2.1364E−16	6.5822E+02	2.6445E+03
	Mean	2.5633E−193	7.1578E−18	2.4379E+01	2.2585E+02
	Stdev.	0.0000E+00	3.8999E−17	1.2667E+02	6.9561E+02
F2	Best	3.1404E−284	3.6009E−179	6.9821E−251	1.5648E−43
	Worst	5.9875E−36	2.2255E−09	5.6621E−24	5.5872E−06
	Mean	1.9958E−37	1.0336E−10	1.8874E−25	1.8624E−07
	Stdev.	1.0932E−36	4.1628E−10	1.0338E−24	1.0201E−06
F3	Best	7.8863E−01	6.8219E−01	4.1503E−01	8.2422E+00
	Worst	8.3445E+01	1.4758E+02	5.5907E+01	1.4537E+02
	Mean	1.4510E+01	5.3496E+01	1.2282E+01	6.3900E+01
	Stdev.	1.9350E+01	4.1604E+01	1.4082E+01	3.5922E+01
F4	Best	1.0365E−13	1.4323E−14	4.1831E−12	7.4064E−08
	Worst	1.1238E−03	8.6072E−02	3.3300E+01	5.0596E+01
	Mean	1.1663E−04	1.2869E−02	1.9450E+00	2.2038E+00
	Stdev.	2.7763E−04	2.0007E−02	7.4795E+00	9.5325E+00
F5	Best	6.6029E−04	2.0602E−04	6.2803E−04	1.0870E−02
	Worst	1.0315E+00	1.2583E+00	3.2189E+06	6.0425E+06
	Mean	1.6330E−01	3.0925E−01	2.2734E+05	3.0083E+05
	Stdev.	2.5698E−01	4.4332E−01	7.0727E+05	1.1549E+06
F6	Best	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	Worst	0.0000E+00	2.1121E−03	3.9047E+03	5.5317E+03
	Mean	0.0000E+00	7.0403E−05	2.8332E+02	4.0554E+02
	Stdev.	0.0000E+00	3.8561E−04	8.5671E+02	1.2405E+03
F7	Best	1.8820E−03	2.2048E−03	1.7919E−03	3.4210E−03
	Worst	6.0311E−03	7.5511E−03	5.4759E−01	2.1158E+00
	Mean	2.9852E−03	4.5852E−03	3.0172E−02	7.5764E−02
	Stdev.	8.5775E−04	1.3483E−03	1.0903E−01	3.8530E−01
F8	Best	− 1.2569E+04	− 1.2569E+04	− 1.2569E+04	− 1.2569E+04
	Worst	− 1.2569E+04	− 1.2569E+04	− 9.7458E+03	− 1.0143E+04
	Mean	− 1.2569E+04	− 1.2569E+04	− 1.2389E+04	− 1.2457E+04
	Stdev.	0.0000E+00	0.0000E+00	6.8472E+02	4.6989E+02
F9	Best	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	Worst	0.0000E+00	2.1364E−16	4.6785E+01	4.8754E+01
	Mean	0.0000E+00	7.1578E−18	5.2139E+00	1.6251E+00
	Stdev.	0.0000E+00	3.8999E−17	1.3842E+01	8.9012E+00
F10	Best	1.5099E−14	1.5099E−14	1.5099E−14	1.8652E−14
	Worst	2.9310E−14	2.9142E−04	1.2547E+01	1.2153E+01
	Mean	2.0191E−14	1.0244E−05	8.2750E−01	1.4744E+00
	Stdev.	4.4435E−15	5.3185E−05	3.1494E+00	3.8314E+00
F11	Best	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	Worst	0.0000E+00	1.1316E−05	3.4024E+01	4.5916E+01
	Mean	0.0000E+00	4.4131E−07	2.0627E+00	3.6401E+00
	Stdev.	0.0000E+00	2.0830E−06	7.1892E+00	1.1629E+01
F12	Best	1.5705E−32	1.5705E−32	1.5705E−32	1.5705E−32
	Worst	1.5705E−32	1.0144E−15	3.4909E+05	9.9692E+06
	Mean	1.5705E−32	3.3819E−17	1.1636E+04	3.3236E+05
	Stdev.	0.0000E+00	1.8520E−16	6.3735E+04	1.8201E+06

Table 7 (continued)

Function		Sen 9	Sen 10	Sen 11	Sen 12
F13	Best	1.3498E−32	1.3498E−32	1.3498E−32	1.3498E−32
	Worst	1.3498E−32	8.9261E−29	6.3660E+06	4.4663E+05
	Mean	1.3498E−32	2.9886E−30	3.7489E+05	1.4888E+04
	Stdev.	0.0000E+00	1.6294E−29	1.4400E+06	8.1543E+04
F14	Best	9.9800E−01	9.9800E−01	9.9800E−01	9.9800E−01
	Worst	9.9800E−01	9.9800E−01	9.9800E−01	9.9800E−01
	Mean	9.9800E−01	9.9800E−01	9.9800E−01	9.9800E−01
	Stdev.	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F15	Best	3.0836E−04	3.1027E−04	3.1304E−04	3.2137E−04
	Worst	7.8177E−04	7.4299E−04	9.2408E−04	8.5773E−04
	Mean	4.5139E−04	4.8287E−04	5.3261E−04	5.6904E−04
	Stdev.	1.1921E−04	1.1762E−04	1.5902E−04	1.5870E−04
F16	Best	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00
	Worst	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00
	Mean	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00	− 1.0316E+00
	Stdev.	0.0000E+00	0.0000E+00	1.1709E−07	6.7752E−16
F17	Best	3.9789E−01	3.9789E−01	3.9789E−01	3.9789E−01
	Worst	3.9789E−01	3.9789E−01	3.9789E−01	3.9790E−01
	Mean	3.9789E−01	3.9789E−01	3.9789E−01	3.9789E−01
	Stdev.	0.0000E+00	0.0000E+00	1.6938E−16	1.8257E−06
F18	Best	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00
	Worst	3.0000E+00	3.0000E+00	3.0001E+00	3.0000E+00
	Mean	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00
	Stdev.	0.0000E+00	0.0000E+00	2.6272E−05	6.2293E−06
F19	Best	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00
	Worst	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00	− 3.8626E+00
	Mean	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00	− 3.8628E+00
	Stdev.	0.0000E+00	0.0000E+00	4.6999E−08	2.7174E−05
F20	Best	− 3.3220E+00	− 3.3220E+00	− 3.3220E+00	− 3.3220E+00
	Worst	− 3.3220E+00	− 3.3220E+00	− 3.3220E+00	− 3.3086E+00
	Mean	− 3.3220E+00	− 3.3220E+00	− 3.3220E+00	− 3.3215E+00
	Stdev.	0.0000E+00	0.0000E+00	7.0028E−07	2.4435E−03
F21	Best	− 1.0153E+01	− 1.0153E+01	− 1.0153E+01	− 1.0153E+01
	Worst	− 1.0153E+01	− 1.0153E+01	− 7.3062E+00	− 1.0153E+01
	Mean	− 1.0153E+01	− 1.0153E+01	− 1.0003E+01	− 1.0153E+01
	Stdev.	0.0000E+00	0.0000E+00	5.9311E−01	6.7700E−06
F22	Best	− 1.0403E+01	− 1.0403E+01	− 1.0403E+01	− 1.0403E+01
	Worst	− 1.0403E+01	− 1.0403E+01	− 4.5713E+00	− 1.0403E+01
	Mean	− 1.0403E+01	− 1.0403E+01	− 1.0072E+01	− 1.0403E+01
	Stdev.	0.0000E+00	0.0000E+00	1.2791E+00	3.0275E−05
F23	Best	− 1.0536E+01	− 1.0536E+01	− 1.0536E+01	− 1.0536E+01
	Worst	− 1.0536E+01	− 1.0536E+01	− 5.8221E+00	− 1.0536E+01
	Mean	− 1.0536E+01	− 1.0536E+01	− 1.0379E+01	− 1.0536E+01
	Stdev.	0.0000E+00	0.0000E+00	8.6070E−01	1.5540E−05

Bold font refers to the best recorded result

the best results in 16 test functions. Interestingly, the CHIO algorithm obtained the best results in 15 test functions, while ABC, SSA, SCA, and BA get the best results in 12,

10, 8, and 7 test functions, respectively. In comparison between the proposed CHIO algorithm and each of the comparative methods, it can be seen that the performance

Table 8 Performance of CHIO algorithm against other swarm-based algorithms

Function	CHIO	BAT	SSA	HHO	JAYA	FPA	SCA	ABC
F1	Best	5.8244E-06	3.6017E-10	0.0000E+00	0.0000E+00	1.2927E-68	0.0000E+00	2.3345E-16
	Worst	1.1328E-03	1.2276E-09	0.0000E+00	0.0000E+00	3.3459E-60	0.0000E+00	4.5276E-16
	Mean	2.5633E-193	7.0399E-10	0.0000E+00	0.0000E+00	1.2329E-61	0.0000E+00	3.1546E-16
	Stdev.	0.0000E+00	1.7096E-10	0.0000E+00	0.0000E+00	6.0969E-61	0.0000E+00	4.9325E-17
F2	Best	3.1404E-284	1.0175E-06	0.0000E+00	0.0000E+00	5.0467E-49	0.0000E+00	7.3926E-16
	Worst	5.9875E-36	2.4602E-06	0.0000E+00	0.0000E+00	2.0949E-46	0.0000E+00	1.0917E-15
	Mean	1.9958E-37	1.4181E-06	0.0000E+00	0.0000E+00	3.2563E-47	0.0000E+00	9.5129E-16
	Stdev.	1.0932E-36	3.0302E-07	0.0000E+00	0.0000E+00	4.6488E-47	0.0000E+00	6.8723E-17
F3	Best	7.8863E-01	1.6382E-11	0.0000E+00	0.0000E+00	6.4264E-36	3.7209E-79	4.2495E-01
	Worst	8.3445E+01	5.4407E-11	0.0000E+00	0.0000E+00	1.8345E-29	5.8476E-28	4.7319E+00
	Mean	1.4510E+01	3.1907E-11	0.0000E+00	0.0000E+00	7.0321E-31	1.9593E-29	1.3009E+00
	Stdev.	1.9350E+01	1.1184E-11	0.0000E+00	0.0000E+00	3.3398E-30	1.0674E-28	8.5311E-01
F4	Best	1.0365E-13	2.0848E-06	0.0000E+00	0.0000E+00	4.6804E-83	9.1151E-81	5.9336E-04
	Worst	1.1238E-03	4.2459E-02	0.0000E+00	0.0000E+00	1.6300E-73	4.1305E-45	3.0871E-02
	Mean	1.1663E-04	1.3337E-02	0.0000E+00	0.0000E+00	8.5571E-75	1.3768E-46	1.4670E-02
	Stdev.	2.7763E-04	6.1972E-03	5.9361E-07	0.0000E+00	1.8150E+00	7.5412E-46	9.0922E-03
F5	Best	6.6029E-04	1.3309E-03	2.9004E-05	2.6646E-19	0.0000E+00	2.5738E+01	1.7761E-04
	Worst	1.0315E+00	5.4626E-01	1.3386E+02	2.7537E-06	4.6121E-27	2.8843E+01	2.4223E-02
	Mean	1.6330E-01	3.0927E-01	9.1381E+00	2.6568E-07	7.1058E-28	2.6756E+01	8.2796E-03
	Stdev.	2.5698E-01	1.4211E-01	2.9857E+01	5.6154E-07	1.0409E-27	7.9304E-01	7.3267E-03
F6	Best	0.0000E+00	7.3260E-04	1.7724E-11	1.1350E-12	1.2663E+00	0.0000E+00	2.7110E-16
	Worst	0.0000E+00	1.2020E-03	4.6742E-11	1.6681E-08	2.3662E+00	3.6950E+00	4.5992E-16
	Mean	0.0000E+00	9.6083E-04	2.8460E-11	2.5973E-09	1.7168E+00	2.7519E+00	3.3343E-16
	Stdev.	0.0000E+00	1.1017E-04	7.9376E-12	3.5328E-09	2.4362E-01	2.7166E-01	5.6644E-17
F7	Best	1.8820E-03	2.0116E-05	1.6440E-05	4.3464E-08	3.1598E-04	1.2680E-05	1.4464E-02
	Worst	6.0311E-03	9.7654E-04	1.3617E-04	2.4894E-06	1.8851E-03	1.0451E-03	3.6492E-02
	Mean	2.9852E-03	2.8460E-04	6.0942E-05	6.1503E-07	8.0063E-04	1.9792E-04	2.7368E-02
	Stdev.	8.5775E-04	2.1320E-04	2.4372E-05	5.6438E-07	3.7976E-04	1.9364E-04	4.9410E-03
F8	Best	- 1.2569E+04	- 1.2569E+04	- 3.7358E+03	- 1.2569E+04	- 1.2569E+04	- 5.2221E+03	- 1.2569E+04
	Worst	- 1.2569E+04	- 1.2569E+04	- 2.7838E+03	- 1.2569E+04	- 9.6524E+03	- 1.2037E+04	- 4.3557E+03
	Mean	- 1.2569E+04	- 1.2569E+04	- 3.2996E+03	- 1.2569E+04	- 1.2414E+04	- 1.2533E+04	- 4.8295E+03
	Stdev.	0.0000E+00	0.0000E+00	2.9626E+02	3.8584E-06	5.3669E+02	1.2421E+02	2.2396E+02
F9	Best	0.0000E+00	1.3971E-03	3.9798E+00	0.0000E+00	2.0894E+01	0.0000E+00	0.0000E+00
	Worst	0.0000E+00	2.1810E-01	1.8904E+01	0.0000E+00	6.9959E+01	0.0000E+00	0.0000E+00
	Mean	0.0000E+00	3.4020E-02	1.1409E+01	0.0000E+00	4.1036E+01	0.0000E+00	0.0000E+00
	Stdev.	0.0000E+00	4.1965E-02	4.2913E+00	0.0000E+00	1.2778E+01	0.0000E+00	0.0000E+00

Table 8 (continued)

Function	CHIO	BAT	SSA	HHO	JAYA	FPA	SCA	ABC
F10	Best	1.2917E-02	1.1456E-06	8.8818E-16	4.4409E-15	4.4409E-15	4.4409E-15	2.2204E-14
	Worst	2.6318E-02	2.7045E-06	8.8818E-16	1.5099E-14	2.9570E+00	2.0091E+01	3.2863E-14
	Mean	2.2488E-02	2.1420E-06	8.8818E-16	1.0481E-14	2.0943E+00	4.7443E+00	2.8481E-14
	Stdev.	4.4435E-15	3.3870E-07	0.0000E+00	3.6315E-15	5.7074E-01	8.5925E+00	3.1893E-15
F11	Best	0.0000E+00	4.4278E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	Worst	0.0000E+00	2.2172E-02	0.0000E+00	3.9202E-02	9.9984E-02	0.0000E+00	0.0000E+00
	Mean	0.0000E+00	3.8227E-03	0.0000E+00	9.6025E-03	2.2508E-02	0.0000E+00	0.0000E+00
	Stdev.	0.0000E+00	6.4556E-03	0.0000E+00	9.8796E-03	2.0960E-02	0.0000E+00	0.0000E+00
F12	Best	1.5705E-32	6.1362E-06	1.9368E-16	1.0735E-01	1.5705E-32	1.7042E-01	2.0090E-16
	Worst	1.5705E-32	9.9548E-06	1.3567E-09	1.9790E+00	4.1467E-01	7.2099E-01	3.3297E-16
	Mean	1.5705E-32	8.0379E-06	2.1282E-10	7.9997E-01	3.1099E-02	2.3448E-01	3.0073E-16
	Stdev.	0.0000E+00	8.1735E-07	3.5150E-10	6.2292E-01	9.4892E-02	9.7074E-02	3.1383E-17
F13	Best	1.3498E-32	7.7966E-05	1.3841E-13	1.3498E-32	1.3498E-32	1.4198E+00	2.2467E-16
	Worst	1.3498E-32	1.1151E-02	8.0172E-09	1.0987E-02	2.1024E-02	1.9365E+00	3.2831E-16
	Mean	1.3498E-32	1.2242E-03	1.2539E-09	1.0987E-03	2.1658E-03	1.7143E+00	2.9817E-16
	Stdev.	0.0000E+00	3.3624E-03	1.8761E-09	3.3526E-03	5.2000E-03	1.3176E-01	2.1200E-17
F14	Best	9.9800E-01	9.9800E-01	9.9800E-01	9.9800E-01	9.9800E-01	9.9800E-01	9.9800E-01
	Worst	9.9800E-01	9.9800E-01	9.9800E-01	9.9800E-01	9.9800E-01	9.9800E-01	9.9800E-01
	Mean	9.9800E-01	9.9800E-01	9.9800E-01	9.9800E-01	9.9800E-01	9.9800E-01	9.9800E-01
	Stdev.	0.0000E+00	0.0000E+00	5.3044E-16	8.0766E-10	0.0000E+00	6.0527E-10	0.0000E+00
F15	Best	3.0836E-04	3.0750E-04	3.0749E-04	3.0749E-04	3.0749E-04	3.0795E-04	3.1349E-04
	Worst	7.8177E-04	3.0757E-04	3.0751E-04	1.2239E-03	3.0749E-04	3.1255E-04	3.8879E-04
	Mean	4.5139E-04	3.0753E-04	3.0749E-04	3.3803E-04	3.0749E-04	3.0992E-04	3.4468E-04
	Stdev.	1.1921E-04	1.8020E-08	4.1185E-04	1.6731E-04	1.0795E-19	1.3604E-06	2.1330E-05
F16	Best	1.0316E+00	1.0316E+00	1.0316E+00	1.0316E+00	1.0316E+00	1.0316E+00	1.0316E+00
	Worst	1.0316E+00	1.0316E+00	1.0316E+00	1.0316E+00	1.0316E+00	1.0316E+00	1.0316E+00
	Mean	1.0316E+00	1.0316E+00	1.0316E+00	1.0316E+00	1.0316E+00	1.0316E+00	1.0316E+00
	Stdev.	0.0000E+00	3.5650E-09	0.0000E+00	0.0000E+00	6.7752E-16	1.4940E-07	6.7752E-16
F17	Best	3.9789E-01	3.9789E-01	3.9789E-01	3.9789E-01	3.9789E-01	3.9789E-01	3.9789E-01
	Worst	3.9789E-01	3.9789E-01	3.9789E-01	3.9797E-01	3.9789E-01	3.9791E-01	3.9789E-01
	Mean	3.9789E-01	3.9789E-01	3.9789E-01	3.9789E-01	3.9789E-01	3.9789E-01	3.9789E-01
	Stdev.	0.0000E+00	1.6204E-09	0.0000E+00	1.5462E-05	0.0000E+00	5.7809E-06	0.0000E+00
F18	Best	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00
	Worst	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00
	Mean	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00
	Stdev.	0.0000E+00	1.2646E-07	0.0000E+00	3.8803E-15	4.8085E-16	9.0569E-10	1.8195E-13

Table 8 (continued)

Function	CHIO	BAT	SSA	HHO	JAYA	FPA	SCA	ABC
F19	Best	- 3.8628E+00	- 3.8628E+00	- 3.8628E+00	- 3.8628E+00	- 3.8628E+00	- 3.8627E+00	- 3.8628E+00
	Worst	- 3.8628E+00	- 3.8628E+00	- 3.8628E+00	- 3.8628E+00	- 3.8628E+00	- 3.8549E+00	- 3.8628E+00
	Mean	- 3.8628E+00	- 3.8628E+00	- 3.8628E+00	- 3.8628E+00	- 3.8628E+00	- 3.8559E+00	- 3.8628E+00
	Stdev.	0.0000E+00	4.0695E-07	2.0451E-15	2.1362E-15	2.7101E-15	2.6900E-03	2.7101E-15
F20	Best	- 3.3220E+00	- 3.3220E+00	- 3.3220E+00	- 3.3220E+00	- 3.3220E+00	- 3.1322E+00	- 3.3220E+00
	Worst	- 3.3220E+00	- 3.2030E+00	- 3.2031E+00	- 3.1896E+00	- 3.2031E+00	- 1.5784E+00	- 3.3220E+00
	Mean	- 3.3220E+00	- 3.2833E+00	- 3.2150E+00	- 3.3095E+00	- 3.2665E+00	- 2.9247E+00	- 3.3220E+00
	Stdev.	0.0000E+00	5.6979E-02	3.6278E-02	3.8128E-02	6.0328E-02	3.5932E-01	1.3550E-15
F21	Best	- 1.0153E+01	- 1.0153E+01	- 1.0153E+01	- 1.0153E+01	- 1.0153E+01	- 1.0120E+01	- 1.0153E+01
	Worst	- 1.0153E+01	- 1.0153E+01	- 1.0153E+01	- 1.0153E+01	- 2.6305E+00	- 4.9653E-01	- 1.0153E+01
	Mean	- 1.0153E+01	- 1.0153E+01	- 1.0153E+01	- 1.0153E+01	- 8.1432E+00	- 3.3205E+00	- 1.0153E+01
	Stdev.	0.0000E+00	2.6309E-07	3.4777E-13	2.6750E-10	2.9755E+00	3.2896E+00	2.6309E-07
F22	Best	- 1.0403E+01	- 1.0153E+01	- 1.0403E+01	- 1.0403E+01	- 1.0403E+01	- 1.0173E+01	- 1.0403E+01
	Worst	- 1.0403E+01	- 1.0153E+01	- 1.0403E+01	- 1.0403E+01	- 1.8376E+00	- 5.2404E-01	- 1.0403E+01
	Mean	- 1.0403E+01	- 1.0153E+01	- 1.0403E+01	- 1.0403E+01	- 8.5511E+00	- 5.3905E+00	- 1.0403E+01
	Stdev.	0.0000E+00	2.6309E-07	2.8255E-13	5.0350E-10	2.8232E+00	3.5424E+00	1.8506E-05
F23	Best	- 1.0536E+01	- 1.0153E+01	- 1.0536E+01	- 1.0536E+01	- 1.0536E+01	- 1.0414E+01	- 1.0536E+01
	Worst	- 1.0536E+01	- 1.0153E+01	- 5.1756E+00	- 1.0536E+01	- 2.4205E+00	- 9.4700E-01	- 1.0536E+01
	Mean	- 1.0536E+01	- 1.0153E+01	- 1.0000E+01	- 1.0536E+01	- 9.9954E+00	- 6.2118E+00	- 1.0536E+01
	Stdev.	0.0000E+00	2.6309E-07	1.6357E+00	4.0765E-10	2.0589E+00	2.7930E+00	2.2016E-05

Bold font refers to the best recorded result

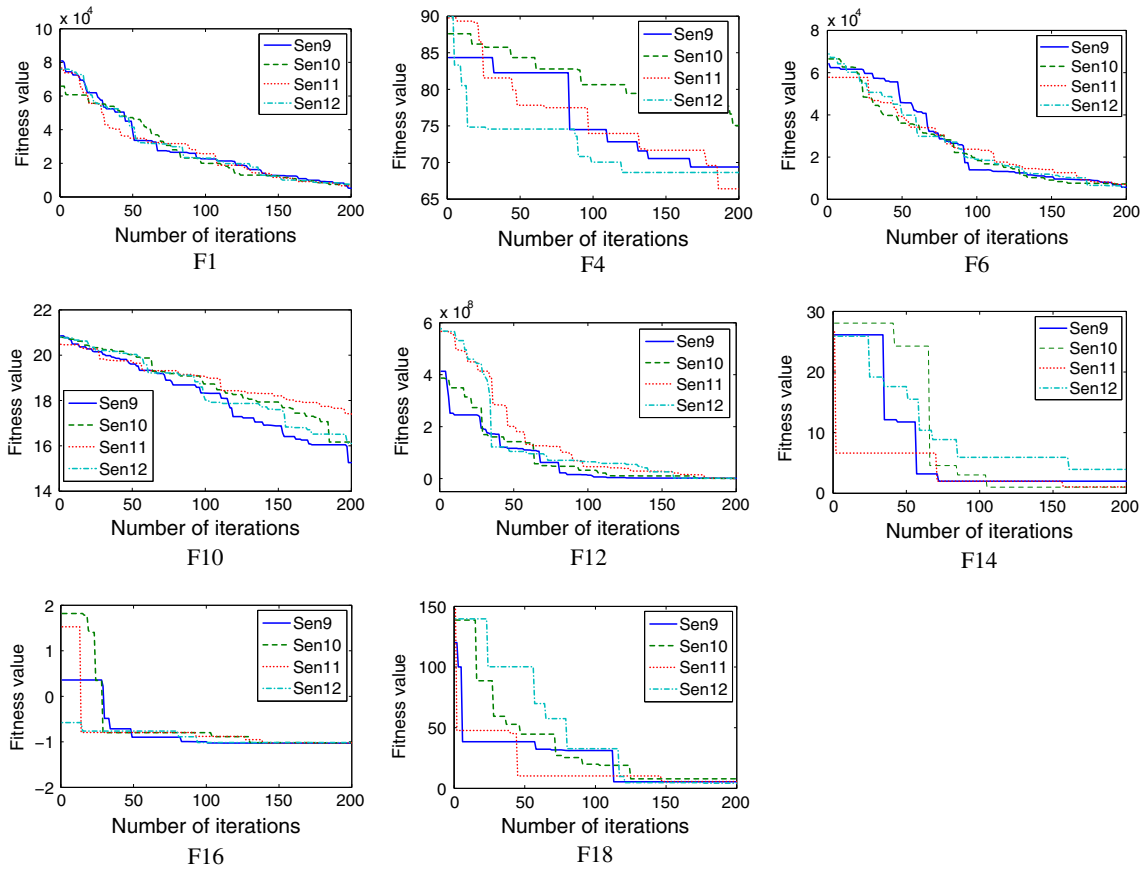


Fig. 7 Convergence plots of CHIO algorithm using different social distancing strategies

of the CHIO algorithm is similar or better than the BA, SSA, HHO, JAYA, FPA, SCA, and ABC algorithms in 20, 17, 16, 16, 19, 17, and 21 test functions, respectively.

Similarly, in terms of the results mean, it can be seen from Table 8 that the performance of all of the comparative algorithms is similar in four test functions (i.e., F14, F16, F17, and F18) as they reach the optimal results. The HHO obtained the best results in 18 test functions, and the proposed CHIO algorithm achieves the best results in 15 test functions. While the ABC and FPA get the best results in 12 and 10 test functions, respectively. The JAYA and SCA obtain the best results in eight test functions, while the BA and SSA get the best results in seven datasets. In comparison between the proposed CHIO algorithm and each of the comparative methods, it can be seen that the performance of the CHIO algorithm is similar or better than the BA, SSA, HHO, JAYA, FPA, SCA, and ABC algorithms in 20, 20, 15, 15, 20, 17, and 20 datasets, respectively.

Figure 8 illustrates the convergence behavior of the proposed CHIO algorithm against the other comparative algorithms. The *x-axis* represents the number of iterations, while the *y-axis* represents the values of the fitness function. It should be noted that eight out of the 23 test functions are considered in this figure to show the differences

between algorithms visually. Figure 8 elaborates that the proposed CHIO algorithm did not have fast convergence like the other comparative methods, where the convergence of the CHIO algorithm is gradually improved during the search. This allows CHIO to avoid the problem of getting stuck in local optima.

Figure 9 plots the Hamming distance between the solutions in the population for the proposed CHIO algorithm as well as the other comparative methods. It can be observed from Fig. 9 that the proposed CHIO algorithm can maintain a good distance between the population. This is because the infected case is killed when it is not improved after a certain number of iterations. Then, these cases are regenerated from scratch and thus solve the problem of fast convergence.

Friedman’s statistical test is used to illustrate the average rankings of the proposed CHIO algorithm when compared against other comparative methods. Table 9 shows the rankings where these rankings are calculated based on the best results recorded in Table 8. It is worthy to mention that the lower rankings indicate better performance, while the significant level $\alpha = 0.05$. Table 9 shows that HHO algorithm is ranked first, while the proposed CHIO algorithm is ranked third. The ρ -value computed by Friedman’s

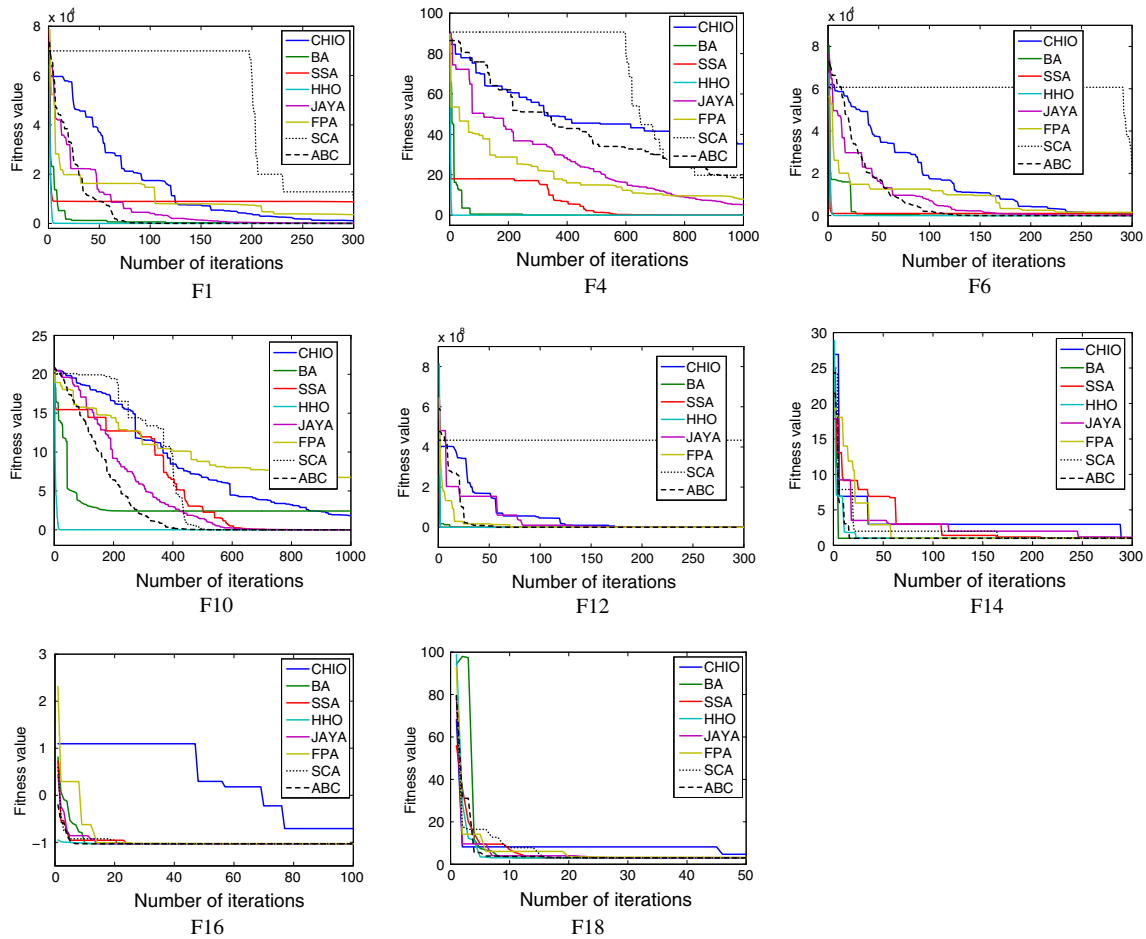


Fig. 8 Convergence plots of CHIO algorithm against the other swarm-based algorithms

test is $1.54E-5$, which is below the significant level. This value indicates that there are significant differences between the performance of the comparative methods.

Thereafter, the Holm’s procedure as a post hoc technique is used to confirm that there are significant differences among the controlled methods (the method with the first rankings) and the remaining comparative methods. It can be seen from the results recorded in Table 10 that the hypothesis is accepted. This means that there is a significant difference between the HHO algorithm and two of the other methods (BA, and SCA). On the other hand, there is no significant difference between HHO and the remaining comparative methods.

Table 11 illustrates the average rankings of the comparative methods, where these rankings are calculated based on the mean results recorded in Table 8. Table 9 points out that HHO algorithm is ranked first, while the proposed CHIO algorithm is placed second. The ρ -value computed by Friedman’s test is $3.97E-5$, which is below the significant level. This value indicates that there is a

significant difference between the performance of the comparative methods.

Thereafter, the Holm’s procedure as a post hoc technique is used to confirm that there is a significant difference between HHO and the other comparative methods. It can be seen from the results recorded in Table 12 that the hypothesis is accepted. It is clear that there is a significant difference between the HHO algorithm and four of the comparative methods (BA, JAYA, SSA, and SCA). On the other hand, there no significant difference between HHO and the remaining comparative methods including the proposed CHIO algorithm (please refers to the Table 12).

The performance of the CHIO algorithm has been evaluated using the Wilcoxon signed-rank statistical test [73] to verify whether there is a significant difference between CHIO and the other comparative algorithms. The Wilcoxon signed-rank is applied using the best results of 30 runs for each algorithm with P_value equal 0.05. Table 13 shows a pair-wise comparison against all algorithms and CHIO, showing whether two algorithms are

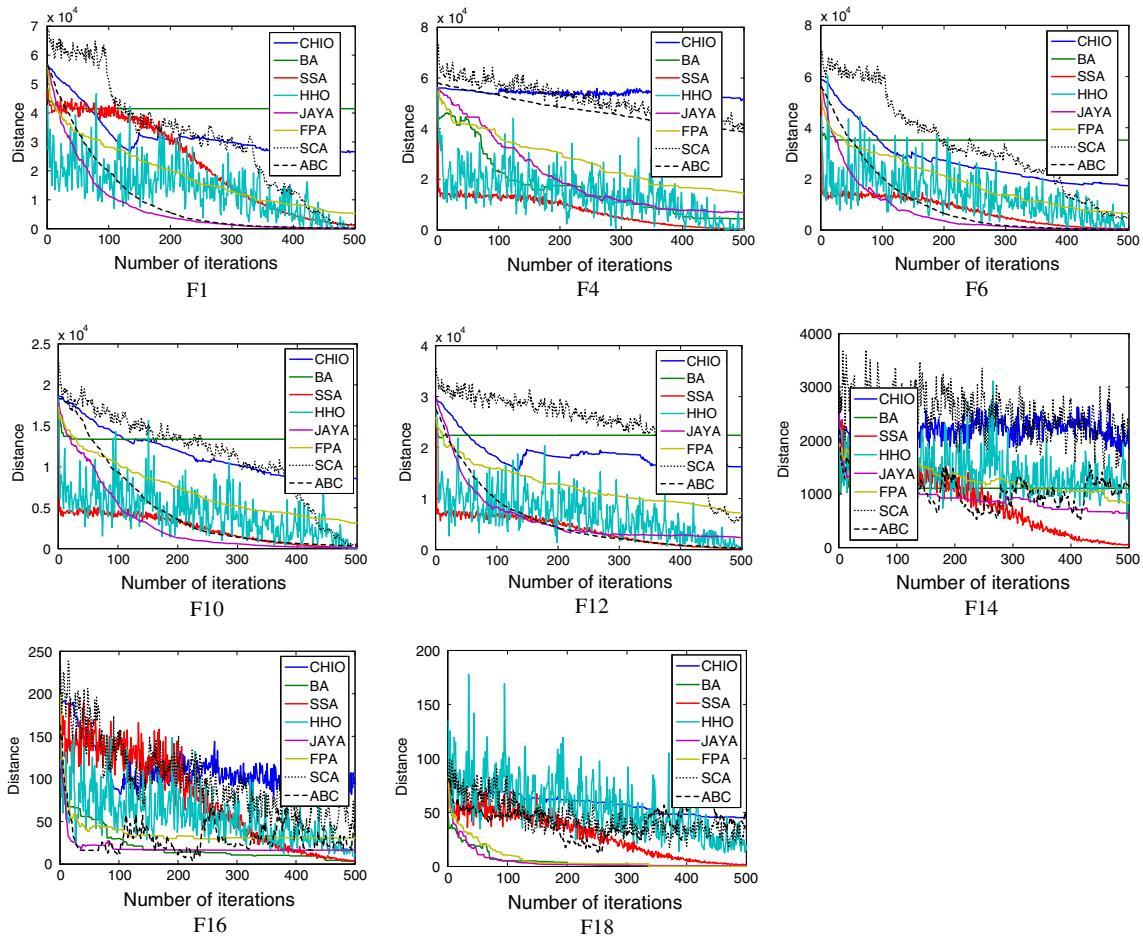


Fig. 9 Hamming distance of CHIO algorithm against the other swarm-based algorithms

Table 9 Average rankings of the algorithms calculated using Friedman’s test (based on the best results)

Order	Algorithm	Ranking
1	HHO	3.37
2	FPA	3.54
3	CHIO	3.83
4	JAYA	3.85
5	ABC	4.43
6	SSA	4.61
7	SCA	5.89
8	BA	6.48

Table 10 Holm’s results between the HHO algorithm and other comparative methods (Based on the best results)

Order	Algorithm	Adjusted p -value	(α /Rank)
7	BA	1.68E–05	0.0071
6	SCA	4.81E–04	0.0083
5	SSA	0.0863	0.0100
4	ABC	0.1403	0.0125
3	JAYA	0.5079	0.0167
2	CHIO	0.5274	0.0250
1	FPA	0.8097	0.0500

considered similar (‘-’) or not (‘++’) to each other. The most algorithm that has been considered similar to CHIO is ABC with 11 functions. The results show that CHIO and HHO are similar with ten functions. Finally, FPA is similar to CHIO in nine functions.

3.7 Real-world engineering optimization problems

For further validations, the applicability of the proposed CHIO algorithm on real-world optimization problems is discussed in this section. The proposed CHIO is tested using three bound-constrained real-world optimization problems which are:

Table 11 Average rankings of the algorithms calculated using Friedman's test (based on the mean results)

Order	Algorithm	Ranking
1	HHO	3.07
2	CHIO	3.26
3	ABC	3.74
4	FPA	4.24
5	SSA	4.87
6	JAYA	5.30
7	BA	5.48
8	SCA	6.04

Table 12 Holm's results between the HHO algorithm and other comparative methods (Based on the mean results)

Order	Algorithm	Adjusted p -value	(α /Rank)
7	SCA	3.74E-05	0.0071
6	BA	8.36E-04	0.0083
5	JAYA	0.0019	0.0100
4	SSA	0.0125	0.0125
3	FPA	0.1041	0.0167
2	ABC	0.3508	0.0250
1	CHIO	0.7865	0.0500

- Parameter estimation for frequency-modulated (FM) sound waves.
- The bifunctional catalyst blend optimal control problem.
- Transmission network expansion planning (TNEP) problem.

It should be noted that details of these problems are introduced in the 2011 IEEE Congress on Evolutionary Computation (IEEE-CEC 2011) [74].

The performance of the CHIO algorithm is compared with nine other comparative methods such as adaptive population-based simplex algorithm (APS) [75], adaptive differential evolution algorithm (ADE) [76], continuous differential ant-stigmergy algorithm (CDASA) [77], differential evolution (DE) [78], hybrid DE-with random hill climber (DE-RHC) [79], genetic algorithm with a new multi-parent crossover (GA-MPC) [80], hybrid DE algorithm with adaptive crossover operator (HDE) [81], hybrid EA-DE-Memetic algorithm (HMA) [82], intellects-masses optimizer (IMO) [83], artificial bee colony [84], accelerated artificial bee colony algorithm [84], and krill herd and artificial bee colony with information exchange (KHABC) [85].

It should be noted that the parameter settings of the CHIO algorithm followed the rules of IEEE-CEC 2011 by having the maximum number of iterations is set to 150,000 and the number of runs is set to 25 times. While the settings

of the other parameters include $HIS = 30$, $BR_r = 0.01$, and $Max_{Age} = 100$. These settings are obtained from the best results in the previous sections.

3.7.1 Parameter estimation for frequency-modulated (FM) sound waves

The performance of the proposed CHIO algorithm when compared against other comparative methods is reported in Table 14. It can be observed from Table 14 that the performance of the proposed CHIO is competitive when compared against other methods.

3.7.2 The bifunctional catalyst blend optimal control problem

Table 15 lists the optimization results of the proposed CHIO algorithm when compared against nine comparative methods. As it can be seen from Table 15, the CHIO algorithm obtained the optimal results in this problem based on the best results or the mean of the results. This proves the efficiency of the proposed CHIO algorithm on solving this kind of problems.

3.7.3 Transmission network expansion planning (TNEP) problem

The experimental results of the proposed CHIO algorithm against other comparative methods are presented in Table 16. Clearly, the best result obtained by the proposed CHIO algorithm for this problem is equivalent to the results in the literature in terms of the best results, the mean of the results, the median of the results, the worst results, and the standard deviation.

4 Conclusion and future work

In this paper, a new natural-inspired human-based meta-heuristic optimization algorithm is proposed which is called coronavirus herd immunity optimizer (CHIO) for global optimization problems. CHIO is inspired by the herd immunity strategy as a way to tackle the spreading of coronavirus pandemics (COVID-19). The population is initiated by several susceptible cases and very few (might be one) infected cases. During the herd immunity evolution, the population is evolved according to the basic reproduction rate (BR_r) affected by social distancing realized in the way of updating the newly generated individuals using three intervention with three possible cases: susceptible, infected, and immuned until the herd immunity is achieved in the population. During the search process,

Table 13 Wilcoxon signed-rank test evaluation between the proposed CHIO algorithm and other methods

Function	BA		SSA		HHO		JAYA	
	<i>P</i> Value	Results	<i>P</i> Value	Results	<i>P</i> Value	Results	<i>P</i> Value	Results
F1	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F2	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F3	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F4	<1.0E – 05	++	0.025	++	<1.0E – 05	++	<1.0E – 05	++
F5	0.00695	++	0.23885	–	<1.0E – 05	++	<1.0E – 05	++
F6	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F7	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F8	0.5	–	<1.0E – 05	++	0.5	–	0.1867	–
F9	<1.0E – 05	++	<1.0E – 05	++	0.5	–	<1.0E – 05	++
F10	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F11	<1.0E – 05	++	<1.0E – 05	++	0.5	–	<1.0E – 05	++
F12	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F13	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F14	0.5	–	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F15	<1.0E – 05	++	0.33724	–	<1.0E – 05	++	<1.0E – 05	++
F16	0.5	–	0.5	–	0.5	–	0.5	–
F17	0.5	–	0.5	–	0.5	–	0.5	–
F18	<1.0E – 05	++	0.5	–	0.5	–	0.5	–
F19	<1.0E – 05	++	0.5	–	0.5	–	0.5	–
F20	<1.0E – 05	++	<1.0E – 05	++	0.039	++	0.00048	++
F21	<1.0E – 05	++	0.5	–	0.5	–	0.00256	++
F22	<1.0E – 05	++	0.5	–	0.5	–	0.00144	++
F23	<1.0E – 05	++	0.038928	++	0.5	–	0.0777	–

Function	FPA		SCA		ABC	
	<i>P</i> Value	Results	<i>P</i> Value	Results	<i>P</i> Value	Results
F1	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F2	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F3	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F4	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F5	<1.0E – 05	++	<1.0E – 05	++	0.00014	++
F6	0.5	–	<1.0E – 05	++	<1.0E – 05	++
F7	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F8	0.00139	++	<1.0E – 05	++	0.5	–
F9	<1.0E – 05	++	0.5	–	0.5	–
F10	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F11	<1.0E – 05	++	0.5	–	0.5	–
F12	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F13	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F14	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F15	<1.0E – 05	++	<1.0E – 05	++	<1.0E – 05	++
F16	0.5	–	<1.0E – 05	++	0.5	–
F17	0.5	–	<1.0E – 05	++	0.5	–
F18	0.5	–	0.00074	++	0.5	–
F19	0.5	–	<1.0E – 05	++	0.5	–
F20	0.5	–	<1.0E – 05	++	0.5	–

Table 13 (continued)

Function	FPA		SCA		ABC	
	P Value	Results	P Value	Results	P Value	Results
F21	0.5	–	<1.0E – 05	++	0.5	–
F22	0.5	–	<1.0E – 05	++	0.5	–
F23	0.5	–	<1.0E – 05	++	0.5	–

++ means results is significant, and – means results is not significant

Table 14 Comparison results of parameter estimation for frequency-modulated (FM) sound waves problem over 25 runs and 150,000 function evaluations

Algorithm	Best	Median	Mean	Worst	Stdv
CHIO	9.0573E+00	1.7060E+01	1.8311E+01	4.0233E+01	8.1538E+00
ABC	2.7725E+00	NA	NA	NA	NA
AABC	3.6696E–01	NA	NA	NA	NA
APS 9	0.0000E+00	1.4813E+01	1.1935E+01	1.8698E+01	6.5169E+00
DE-RHC	5.0200E–20	NA	8.9100E+00	1.5600E+01	6.3700E+00
ADE	0.0000E+00	0.0000E+00	3.8526E+00	1.7021E+01	5.6900E+00
CDASA	3.2789E–18	1.1376E+01	1.0128E+01	2.1171E+01	7.0955E+00
GA-MPC	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
HDE	7.2093E–15	1.2362E–11	8.7697E–01	1.1757E+01	3.0439E+00
IMO	0.0000E+00	0.0000E+00	8.9894E–01	1.2306E+01	3.1266E+00
KHABC	1.2310E+01	NA	2.2310E+01	2.7790E+01	3.5300E+00
HMA	1.1674E–11	6.0847E–10	2.0949E+00	1.1374E+01	4.3064E+00
DE	0.0000E+00	1.0854E–27	6.0448E–13	1.3127E–11	2.6388E–12

Bold font refers to the best recorded result

Table 15 Comparison results of bifunctional catalyst blend optimal control problem over 25 runs and 150,000 function evaluations

Algorithm	Best	Median	Mean	Worst	Stdv
CHIO	1.1515E–05	1.1516E–05	1.1515E–05	1.1516E–05	4.5343E–10
APS 9	1.1515E–05	1.1515E–05	1.1515E–05	1.1515E–05	4.8070E–19
DE–RHC	1.1515E–05	NA	1.1515E–05	1.1515E–05	0.0000E+00
ADE	1.1515E–05	1.1515E–05	1.1515E–05	1.1515E–05	3.8043E–19
CDASA	1.1515E–05	1.1515E–05	1.1515E–05	1.1515E–05	1.6885E–24
GA-MPC	1.1515E–05	1.1515E–05	1.1515E–05	1.1515E–05	0.0000E+00
HDE	1.1515E–05	1.1515E–05	1.1515E–05	1.1515E–05	6.1087E–18
IMO	1.1515E–05	1.1515E–05	1.1515E–05	1.1515E–05	0.0000E+00
HMA	1.1515E–05	1.1515E–05	1.1515E–05	1.1515E–05	9.9711E–19
DE	1.1515E–05	1.1515E–05	1.1515E–05	1.1515E–05	2.0039E–19

Bold font refers to the best recorded result

some fatality cases are occurred according to the maximum number of iterations (Max_{Age}) when it remains unimproved with the infected state. The state of cases are updated during the search from susceptible to infected and from infected to immuned according to the herd immunity threshold, and it depends on the immunity rate of the generated cases.

The viability of the proposed CHIO is tested using 23 well-known benchmark functions with different size and

complexity: seven unimodal, six multimodal with flexible dimensions, and ten multimodal with fixed dimensions. These functions are well circulated in the literature to evaluate newly proposed optimization algorithms.

Initially, the effect of the control parameters (BR_r and Max_{Age}) on the convergence behavior of CHIO is studied. In conclusion, using a small value of BR_r is desired to strike the right balance between exploration and exploitation of the search space. Furthermore, the value of Max_{Age}

Table 16 Comparison results of transmission network expansion planning (TNEP) problem over 25 runs and 150,000 function evaluations

Algorithm	Best	Median	Mean	Worst	Stdv
CHIO	2.2000E+02	2.2000E+02	2.2000E+02	2.2000E+02	0.0000E+00
APS 9	2.2000E+02	2.2000E+02	2.2000E+02	2.2000E+02	0.0000E+00
DE–RHC	2.2000E+02	NA	2.2000E+02	2.2000E+02	0.0000E+00
ADE	2.2000E+02	2.2000E+02	2.2000E+02	2.2000E+02	0.0000E+00
CDASA	2.2000E+02	2.2000E+02	2.2000E+02	2.2000E+02	0.0000E+00
GA-MPC	2.2000E+02	2.2000E+02	2.2000E+02	2.2000E+02	0.0000E+00
HDE	2.2000E+02	2.2000E+02	2.2000E+02	2.2000E+02	0.0000E+00
IMO	2.2000E+02	2.2000E+02	2.2000E+02	2.2000E+02	0.0000E+00
HMA	2.2000E+02	2.2000E+02	2.2000E+02	2.2000E+02	0.0000E+00
DE	2.2000E+02	2.2000E+02	2.2000E+02	2.2000E+02	0.0000E+00

Bold font refers to the best recorded result

does not have a high impact on the performance of CHIO. However, using a small value of this parameter is necessary to diversify the search. The social distancing strategies in the herd immunity evolution are also investigated which are random–random–random, random–best–random, random–random–best, and random–best–best. In conclusion, The random–random–random social distancing strategies in the herd immunity evolution revealed the best performance of CHIO. For comparative evaluation, the proposed CHIO is compared against seven well-established comparative methods using the same benchmark functions. The comparative results show that CHIO is very competitive which is able to obtain 16 out of 23 new results for the test-bed functions. For more validations, three real-world engineering optimization problems extracted from IEEE–CEC 2011 are used. Again, CHIO is proved to be efficient.

As the proposed CHIO reveals very successful outcomes, CHIO can be widely used in the future for several kinds of real-world optimization problems. Furthermore, the optimization structure of CHIO can be improved by adapting its parameters to result in a parameter-less CHIO. Also for the future, the binary, discrete, multi-objective versions of CHIO can be proposed. Another future direction can use the herd immunity threshold as a stop condition for the algorithm.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

- Törn A, Žilinskas A (1989) Global optimization, vol 350. Springer, Berlin
- Osman IH, Laporte G (1996) Metaheuristics: a bibliography. *Annals Oper Res* 63(5):511–623
- Rothlauf F (2011) Optimization methods. Springer, Berlin, pp 45–102
- Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
- Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput Surv* 35(3):268–308
- Pardalos Panos M, Thelma M, Jue X (1998) The graph coloring problem: a bibliographic survey. Springer, Berlin, pp 1077–1141
- Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
- Fernando F, Adolfo R-O, Erik C, Andrade Ángel G, Marco P-C (2019) From ants to whales: metaheuristics for all tastes. *Artif Intel Rev* 2019:1–58
- Goldberg David E, Henry HJ (1988) Genetic algorithms and machine learning. Springer, Berlin
- James K, Russell E (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, vol 4, pp. 1942–1948. IEEE
- Dorigo M, Di Caro G (1999) Ant colony optimization: a new meta-heuristic. In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406), vol 2, pp. 1470–1477. IEEE
- Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
- Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. *Simulation* 76(2):60–68
- Tan Y, Zhu Y (2010) Fireworks algorithm for optimization. In: International conference in swarm intelligence, pp. 355–364. Springer
- Back T, Hoffmeister F, Schwefel H-P (1991) A survey of evolution strategies. In: Proceedings of the fourth international conference on genetic algorithms, vol 2. Morgan Kaufmann Publishers San Mateo, CA
- Koza JR, Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection, vol 1. MIT Press, Cambridge
- Simon D (2008) Biogeography-based optimization. *IEEE Trans Evol Comput* 12(6):702–713
- Yang X-S, Deb S (2009) Cuckoo search via lévy flights. In: 2009 World congress on nature & biologically inspired computing (NaBIC)

20. Yang X-S, Gandomi AH (2012) Bat algorithm: a novel approach for global engineering optimization. *Eng Comput* 29(5):464–483
21. Mirjalili S (2015) The ant lion optimizer. *Adv Eng Softw* 83:80–98
22. Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. *Soft Comput* 23(3):715–734
23. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput Appl* 27(4):1053–1073
24. Pan W-T (2012) A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowl Based Syst* 26:69–74
25. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. *Commun Nonlinear Sci Numer Simul* 17(12):4831–4845
26. Fathollahi-Fard AM, Hajiaghaei-Keshteli M, Tavakkoli-Moghaddam R (2020) Red deer algorithm (rda): a new nature-inspired meta-heuristic. *Soft Comput* 2020:1–29
27. Askarzadeh A (2014) Bird mating optimizer: an optimization algorithm inspired by bird mating strategies. *Commun Nonlinear Sci Numer Simul* 19(4):1213–1228
28. Yang X (2012) Flower pollination algorithm for global optimization. In: *International conference on unconventional computing and natural computation*, pp 240–249. Springer
29. Wang G-G, Deb S, Cui Z (2019) Monarch butterfly optimization. *Neural Comput Appl* 31(7):1995–2014
30. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl Based Syst* 89:228–249
31. Yang X-S et al (2008) Firefly algorithm. *Nat Inspir Metaheur Algorithms* 20:79–90
32. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191
33. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. *Future Gener Comput Syst* 97:849–872
34. Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Comput Struct* 169:1–12
35. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Comput Appl* 27(2):495–513
36. Mirjalili S (2016) Sca: a sine cosine algorithm for solving optimization problems. *Knowl Based Syst* 96:120–133
37. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput Struct* 110:151–166
38. Birbil Şİ, Fang S-C (2003) An electromagnetism-like mechanism for global optimization. *J Global Optim* 25(3):263–282
39. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) Gsa: a gravitational search algorithm. *Inf Sci* 179(13):2232–2248
40. Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. *Acta Mech* 213(3–4):267–289
41. Erol OK, Eksin I (2006) A new optimization method: big bang-big crunch. *Adv Eng Softw* 37(2):106–111
42. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S (2019) Henry gas solubility optimization: a novel physics-based algorithm. *Future Gener Comput Syst* 101:646–667
43. Yampolskiy RV, El-Barkouky A (2011) Wisdom of artificial crowds algorithm for solving np-hard problems. *Int J Bio-Inspir Comput* 3(6):358–369
44. Al-Betar MA (2017) β -hill climbing: an exploratory local search. *Neural Comput Appl* 28(1):153–168
45. Glover F, Laguna M (1998) Tabu search. In: *Handbook of combinatorial optimization*, pp 2093–2229. Springer
46. He S, Wu QH, Saunders JR (2006) A novel group search optimizer inspired by animal behavioural ecology. In: *2006 IEEE international conference on evolutionary computation*, pp 1272–1278. IEEE
47. Gandomi AH (2014) Interior search algorithm (isa): a novel approach for global optimization. *ISA Trans* 53(4):1168–1183
48. Dai Chaohua, Zhu Y, Chen W (2006) Seeker optimization algorithm. In: *International conference on computational and information science*, pp 167–176. Springer
49. Ramezani F, Lotfi S (2013) Social-based algorithm (sba). *Appl Soft Comput* 13(5):2837–2856
50. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. *Appl Soft Comput* 13(5):2592–2612
51. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1(1):67–82
52. Chih-Cheng L, Tzu-Ping S, Wen-Chien K, Hung-Jen T, Po-Ren H (2020) Severe acute respiratory syndrome coronavirus 2 (sars-cov-2) and corona virus disease-2019 (covid-19): the epidemic and the challenges. *Int J Antimicrob Agents* 55:105924
53. World Health Organization (2020) Q&a: influenza and covid-19-similarities and differences
54. Kwok KO, Lai F, Wei WI, Wong SYS, Tang JWT (2020) Herd immunity-estimating the level required to halt the covid-19 epidemics in affected countries. *J Inf* 80(6):e32–e33
55. Fine PEM (1993) Herd immunity: history, theory, practice. *Epidemiol Rev* 15(2):265–302
56. Ribeiro GS, Hamer GL, Diallo M, Kitron U, Ko AI, Weaver SC (2020) Influence of herd immunity in the cyclical nature of arboviruses. *Curr Opin Virol* 40:1–10
57. Randolph HE, Barreiro LB (2020) Herd immunity: Understanding covid-19. *Immunity* 52(5):737–741
58. Wu F, Wang A, Liu M, Wang Q, Chen J, Xia S, Ling Y, Zhang Y, Xun J, Lu L, et al. (2020) Neutralizing antibody responses to sars-cov-2 in a covid-19 recovered patient cohort and their implications
59. Syal K (2020) Covid-19: herd immunity and convalescent plasma transfer therapy. *J Med Virol* 13:13
60. Weaver SC, Reisen WK (2010) Present and future arboviral threats. *Antiviral Res* 85(2):328–345
61. Remuzzi A, Remuzzi G (2020) Covid-19 and italy: what next? *Lancet* 395:497
62. Anderson RM, May RM (1990) Immunisation and herd immunity. *Lancet* 335(8690):641–645
63. Lavine JS, King AA, Bjørnstad ON (2011) Natural immune boosting in pertussis dynamics and the potential for long-term vaccine failure. *Proc Natl Acad Sci* 108(17):7259–7264
64. Long NJ (2020) From social distancing to social containment: reimagining sociality for the coronavirus pandemic. *Med Anthropol Theory*
65. Jefferson T, Foxlee R, Del Mar C, Dooley L, Ferroni E, Hewak B, Prabhala A, Nair S, Rivetti A (2008) Physical interventions to interrupt or reduce the spread of respiratory viruses: systematic review. *Bmj* 336(7635):77–80
66. Glass RJ, Glass LM, Beyeler WE, Min HJ (2006) Targeted social distancing designs for pandemic influenza. *Emerg Inf Dis* 12(11):1671
67. HospiMedica International staff writers (2020) Sweden’s coronavirus strategy targeting herd immunity could be adopted globally, say analysts
68. Jung F, Krieger V, Hufert FT, Küpper J-H (2020) Herd immunity or suppression strategy to combat covid-19. *Clin Hemorheol Microcircul (Preprint)*:1–5

69. World Health Organization (2020) Covid-19 sweden data
70. Cohen J, Kupferschmidt K (2020) Countries test tactics in 'war' against covid-19
71. World Health Organization (2020) Covid-19 uk data
72. Rao R (2016) Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int J Ind Eng Comput* 7(1):19–34
73. Wilcoxon F (1992) Individual comparisons by ranking methods. In: *Breakthroughs in statistics*, pp 196–202. Springer
74. Das S, Suganthan PN (2010) Problem definitions and evaluation criteria for cec 2011 competition on testing evolutionary algorithms on real world optimization problems. Jadavpur University, Nanyang Technological University, Kolkata, pp 341–359
75. Omran MGH, Clerc M (2018) Aps 9: an improved adaptive population-based simplex method for real-world engineering optimization problems. *Appl Intell* 48(6):1596–1608
76. Asafuddoula M, Ray T, Sarker R (2011) An adaptive differential evolution algorithm and its performance on real world optimization problems. In: 2011 IEEE congress of evolutionary computation (CEC), pp 1057–1062. IEEE
77. Korošec P, Šilc J (2011) The continuous differential ant-stigmergy algorithm applied to real-world optimization problems. In: 2011 IEEE congress of evolutionary computation (CEC), pp 1327–1334. IEEE
78. Zamuda A, Brest J (2018) On tenfold execution time in real world optimization problems with differential evolution in perspective of algorithm design. In: 2018 25th international conference on systems, signals and image Processing (IWSSIP), pp 1–5. IEEE
79. LaTorre A, Muelas S, Peña J-M (2011) Benchmarking a hybrid de-rhc algorithm on real world problems. In: 2011 IEEE congress of evolutionary computation (CEC), pp 1027–1033. IEEE
80. Elsayed SM, Sarker RA, Essam DL (2011) Ga with a new multi-parent crossover for solving ieeec-2011 competition problems. In: 2011 IEEE congress of evolutionary computation (CEC), pp 1034–1040. IEEE
81. Reynoso-Meza G, Sanchis J, Blasco X, Herrero JM (2011) Hybrid de algorithm with adaptive crossover operator for solving real-world numerical optimization problems. In: 2011 IEEE congress of evolutionary computation (CEC), pp 1551–1556. IEEE
82. Singh HK, Ray T (2011) Performance of a hybrid ea-de-memetic algorithm on cec 2011 real world optimization problems. In: 2011 IEEE congress of evolutionary computation (CEC), pp 1322–1326. IEEE
83. Omran MGH, Alsharhan S, Clerc M (2018) A modified intellects-masses optimizer for solving real-world optimization problems. *Swarm Evol Comput* 41:159–166
84. Gothania B, Mathur G, Yadav RP Accelerated artificial bee colony algorithm for parameter estimation of frequency-modulated sound waves
85. Wang H, Yi J-H (2018) An improved optimization method based on krill herd and artificial bee colony with information exchange. *Memetic Comput* 10(2):177–198

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.