



# A novel metaheuristic algorithm inspired by COVID-19 for real-parameter optimization

Soleiman Kadkhoda Mohammadi<sup>1</sup> · Daryoush Nazarpour<sup>1,2</sup> · Mojtaba Beiraghi<sup>1</sup>

Received: 21 September 2021 / Accepted: 6 January 2023 / Published online: 9 March 2023  
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2023

## Abstract

In this modern world, we are encountered with numerous complex and emerging problems. The metaheuristic optimization science plays a key role in many fields from medicine to engineering, design, etc. Metaheuristic algorithms inspired by nature are among the most effective and fastest optimization methods utilized to optimize different objective functions to minimize or maximize one or more specific objectives. The use of metaheuristic algorithms and their modified versions is expanding every day. However, due to the abundance and complexity of various problems in the real world, it is always necessary to select the most proper metaheuristic method; hence, there is a strong need to create new algorithms to achieve our desired goal. In this paper, a new and powerful metaheuristic algorithm, called the coronavirus metamorphosis optimization algorithm (CMOA), is proposed based on metabolism and transformation under various conditions. The proposed CMOA algorithm has been tested and implemented on the comprehensive and complex CEC2014 benchmark functions, which are functions based on real-world problems. The results of the experiments in a comparative study under the same conditions show that the CMOA is superior to the newly-developed metaheuristic algorithms including AIDO, ITGO, RFOA, SCA, CSA, CS, SOS, GWO, WOA, MFO, PSO, Jaya, CMA-ES, GSA, RW-GWO, mTLBO, MG-SCA, TOGPEAe, m-SCA, EEO and OB-L-EO, indicating the effectiveness and robustness of the CMOA algorithm as a powerful algorithm. As it was observed from the results, the CMOA provides more suitable and optimized solutions than its competitors for the problems studied. The CMOA preserves the diversity of the population and prevents trapping in local optima. The CMOA is also applied to three engineering problems including optimal design of a welded beam, a three-bar truss and a pressure vessel, showing its high potential in solving such practical problems and effectiveness in finding global optima. According to the obtained results, the CMOA is superior to its counterparts in terms of providing a more acceptable solution. Several statistical indicators are also tested using the CMOA, which demonstrates its efficiency compared to the rest of the methods. This is also highlighted that the CMOA is a stable and reliable method when employed for expert systems.

**Keywords** Optimization · Real-world optimization functions · Engineering optimization · Optimization algorithms · Coronavirus metamorphosis optimization algorithm (CMOA)

## 1 Introduction and literature review

✉ Daryoush Nazarpour  
d.nazarpour@urmia.ac.ir

Soleiman Kadkhoda Mohammadi  
soleiman.kadkhodamohammadi@iau.ac.ir

Mojtaba Beiraghi  
mojtaba.beiraghi@iau.ac.ir

<sup>1</sup> Department of Electrical Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

<sup>2</sup> Department of Electrical Engineering, Urmia University, Urmia, Iran

In general, there are one or more objective functions in any given optimization problem, which sometimes may require different conditions to be optimized. The main aim is to find the best solution for the control variables of the problem, which of course must satisfy a set of conditions and constraints. Hence, optimization in various problems is a process to determine the control variables of the problem in the designs, which must ultimately achieve the objective or objectives in a way that the necessary energy or cost is

minimized and the desired profit is maximized [1–3]. Optimization methods are generally grouped into two categories: exact algorithms and heuristic algorithms [4]. The former can be used to solve small or medium-sized problems, but the latter is utilized to solve complex problems and obtain near-optimal solutions in proper time [5]. The various heuristic optimization algorithms available can be applied only to specific optimization problems [4] and they are possible to trap in local optimizations. To overcome this issue, metaheuristic algorithms were proposed [5].

In general, modern evolutionary and inspirational algorithms can be divided into four categories based on structure and inspired model, including (1) bio-inspired algorithms (BIA), (2) swarm-intelligence-based algorithms (SI), (3) natural science-based algorithms (NSA) such as electromagnetic laws in physics, and (4) natural phenomena-based algorithms (NPA) mimics the phenomena that occur in nature. Thus, a new algorithm can be inspired by evolutionary natural phenomena, swarm behaviors of creatures, physical laws, and human concepts. Examples of this classification are given as follows:

## 1.1 Bio-inspired algorithms (BIA)

Genetic algorithms (GAs) [6] are a family of computational models inspired by the concept of evolution. This category of algorithms encodes probabilistic or candidate solutions, or probabilistic hypotheses for a particular problem in a chromosome-like data structure. GAs preserve vital information stored in these chromosome-like data structures by applying recombination operators. Plant intelligence using plants nervous system [7] has been introduced based on the biology, habit, and lifestyle of weeds in 2017 by Akyol and Alatas. Virulence optimization algorithm (VOA) using the optimal mechanism of viruses [8] is introduced based on the growth and reproduction of viruses in human cells. It is similar to GAs from many perspectives and is a novel evolutionary algorithm. Chemotherapy science using the chemotherapy method [9] is a population-based optimization algorithm introduced by Salmani and Eshghi in 2017. The inspiration of this algorithm is that chemotherapy attempts to destroy cancer disease and malignant cells and replace them with acceptable solutions, i.e., with healthy cells.

Kidney-inspired algorithm (KA) using kidney process in the human body [10] is a type of BIAs. A kidney is one of the important organs of the body that is responsible for cleansing the blood and eliminating pollution from the body. In this algorithm, the author selects and filters the appropriate populations and sends them to the next iteration of the algorithm to continue the optimization cycle.

Artificial infectious disease algorithm using the SEIQR epidemic model, S (susceptibility), E (exposure), I (infection), Q (quarantine), and R (recovery) [11] attacks the bodies of people in society, infects them, and through infectious disease, it will automatically spread through the people who are infected in the community and among the whole population. Hence, the author has been inspired by the same model of attack, spread, and recovery to present this algorithm.

Colonial competitive differential evolution (CCDE) [12] algorithm divides the population into several groups, with each group playing the role of a colonizer with its colonies. Then, a special mutation is used in each group for that colonialists and colonies so that the algorithm covers a wide range of differential evolution algorithms.

The tree growth algorithm (TGA) using tree competition for acquiring light and foods [13] is designed based on the growth of trees and their germination in forest environments. It has two phases of diversification and intensification. The author of this paper claims that the proposed algorithm is a robust algorithm for continuous optimization.

Invasive tumor growth (ITGA) algorithm using tumor growth mechanism [14] is derived from invasive gland growth. In this algorithm, the population is the gland cells that try to grow and multiply, which are divided into proliferative, quiescent, and dying cell groups). Testing this algorithm on the test functions CEC2005, CEC2008, CEC2010 as well as the support vector machine (SVM) parameter optimization problem indicates that this new algorithm is powerful and effective. Invasive weed optimization (IWO) [15] is one of the most fundamental and recent algorithms based on the growth and multiplication of weeds to a certain extent by the user. Each weed forms a clone that begins to spawn and form other weeds around it.

Imperialist competitive algorithm (ICA) [16] is an inspiration by the competitive practice between the old imperialists, such as Britain, France, and the Netherlands, which divides the population into several groups, each group representing an imperialist and his colonies. The basis of this algorithm is absorbing the colonies by the respective emperor and the internal revolution in the colonies and finally the fall of the weaker empires in an imperialist rivalry between them.

JAYA algorithm [17] is a simplified differential evolutionary algorithm without any control parameters or any type of crossover that has been widely used by researchers in recent years due to its simplicity.

Emperor penguin optimizer (EPO) via the behavior of emperor penguin [18] is based on the territory and boundaries of the group and swarm, the size of the boundary temperature and the surrounding boundaries, the

distance between members, and also finding an effective stimulus between members.

Weighted superposition attraction (WSA) using weighted superposition of active fields [19] is a new type of particle swarm optimization (PSO), which is based on two concepts: superposition mechanisms and attracted movement of agents. Hence, this algorithm can converge to the optimal solution very quickly.

Path planning algorithm (PPA) using plant growth mechanism [20] is based on the embryo and the roots of seeds and eggs, as well as the reaction of trees and buds to the light source because the trees move toward the light source and want to receive more share of sunlight. A sine-cosine algorithm using sine cosine functions [21] is an evolutionary algorithm in which coefficients and control parameters are placed in sine and cosine functions.

## 1.2 Swarm-intelligence-based algorithms (SI)

Ant colony optimization (ACO) [22] is based on the natural behavior and life of the ant colony and the worker ants working in the colonies. The foraging process in these colonies is considerably efficient and the workers naturally find a logical and optimal way between their place of living and different food sources, which is the basis of optimization in this algorithm.

Harris hawks optimization (HHO) using Harris' hawk's behavior [23] is based on the coordinated behavior and hunting of Harris hawks in nature. This algorithm is a new type of hunting and hunter algorithm.

Yellow saddle goatfish using collaborative behaviors found in fish [24] is an algorithm in which the population consists of predatory fish groups whose hunting strategy is inspired by chaser and blocker roles in coordinated, multi-group behavior.

Mouth brooding fish (MBF) using the behavior of mouth brooding fish [25] is inspired by the life cycle of the mouth brooding fish, which includes the movement and dispersion phases and the conservation behavior of these fish. Pathfinder algorithm (PA) using the collective movement of an animal group [26] is a new type and model of the PSO algorithms, which is based on coordinated movement and is inspired by different groups of animals with strong leaders in each group. Sailfish optimizer (SO) using a group of hunting sailfish [27] divided the population into two types. The first group, i.e., sailfish, searches for the best current solution. The second group, i.e., sardines, is the population used to generate population diversity in the entire search space of the problem.

Lion optimization algorithm (LOA) using the behavior of lion [28] is inspired by a specific lifestyle and coordinated and group behavior in hunting prey. The most important steps of this algorithm include hunting,

definition, moving toward a safe place, roaming, mating, defense, migration, and lion population equilibrium.

An artificial flora algorithm (AFA) using the behavior of flora is presented in [29]. This plant has two special characteristics, both in reproduction and migration. Migration in this plant is done by moving seeds by any means such as wind or animals to more suitable environments than the current environment for growth and reproduction. That cycle is used in this algorithm for optimization operation.

The cooperation search algorithm (CSA) via the team cooperation behaviors in the modern enterprise was introduced in [30]. This behavior is based on three operators: team communication, reflective learning, and dualistic competition. Team communication helps the global search, reflective learning creates a balance between exploration and exploitation, and dualistic competition is used to select better solutions and send them to the next cycle.

The squirrel search algorithm (SSA) using the behavior of southern flying squirrels is discussed in [31]. As the name implies, this algorithm is based on the effort and dynamic search of squirrels among trees and the environment or problem environment for food. This algorithm is designed using how squirrels travel long distances. The algorithm has several control parameters that are updated by the algorithm itself in the optimization cycle.

Phasor particle swarm optimization (PPSO) [32] is a new version of PSO in which each member has an angle from which all PSO control parameters are placed in sine and cosine functions of this angle. Therefore, it is a simpler version without a control parameter with more powerful optimization.

Collective decision optimization (CDO) using the social behavior of human-based [33] consists of the characteristics of deciding by humans to find a more optimal solution in each iteration of the algorithm, which includes the phases of experience, others, group thinking, leader, and innovation. The result given in the relevant paper indicates the effectiveness of the proposed method.

Pity beetle algorithm (PBA) using aggregation behavior, searching for nest and food was introduced in [34]. Pity beetle can live in suitable places in the forest and reap and harvest the leaves of the weakest trees and at the same time, the population multiplies tremendously, which is a big threat to the health of trees. The author has entered this process into the optimization cycle and achieved acceptable results.

## 1.3 Natural science-based algorithms (NSA)

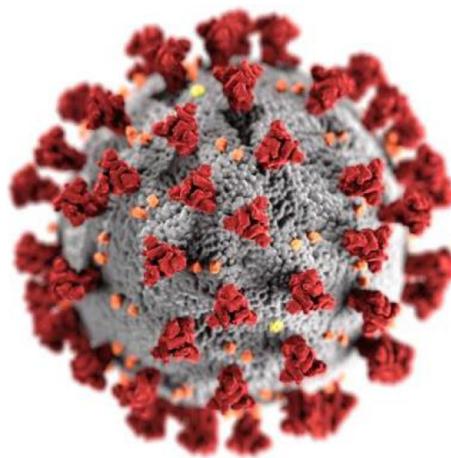
A new and efficient optimizer by Franklin's and Coulomb's laws theory (CFA) [35] divides the population into several groups. Then, the population is modeled into two categories, weak and strong, which are equivalent to positive

and negative charges. Finally, in each iteration of the algorithm, the action of absorption and repulsion and also the population conduction between members and groups is performed which is the basis of optimization in this algorithm. Simulated annealing (SA) algorithm [36] is often utilized to estimate the global optimization in optimization problems with large or discrete search space. This algorithm is based on the metallurgy annealing process. During the annealing process, the physical and sometimes the chemical characteristics of the material change, during which the metal is first heated and then kept at a certain temperature and then gradually cooled. This changes the free movement of the molecules. Moreover, the heating and cooling rates and cooling also have many effects on the metal structure, which forms the basis of optimization in this algorithm.

Yin-Yang-pair optimization (YYO) using a specific mechanism or physical event is adopted in [37]. This algorithm, which is a new lightweight optimizer, is based on maintaining a balance between exploration and exploitation in the problem space. The algorithm has three controllable parameters that are determined by the user according to the problem structure. Electromagnetic field optimization (EFO) using the behavior of electromagnets with different polarities was used in [38]. One of the notable advantages of this algorithm is the use of a nature-inspired ratio called the golden ratio. The artificial chemical reaction optimization algorithm (ACROA) [39] is a robust algorithm with low control parameters based on the use of chemical activity types and events. Gravitational search algorithm (GSA) [40], which is inspired by the physical laws of gravitational attraction and Newtonian motion, forces individuals or algorithms that are random objects in space or dimension of the problem to search the problem space. Atom search optimization (ASO) using the atomic motion model was used in [41]. In this algorithm, the population members, which are the equivalent of atoms, absorb each other through Lennard-Jones forces, and this is the basis of optimization in this algorithm.

Thermal exchange optimization (TEO) using Newton's law of cooling was employed in [42]. Newton's law of cooling states the ratio of heat loss from a body due to heat flow between a body and its surroundings caused by temperature differences. A group of populations in the algorithm is modeled as a cold object and the rest play the role of the environment or other populations in which the temperature is transferred between the two groups of the population, which is the basis of optimization of this algorithm.

Henry gas solubility optimization (HGSO), which mimics the behavior governed by Henry's law [43] is a novel and multi-group type of PSOs that has more control parameters. However, the author claims that this algorithm



**Fig. 1** Coronavirus (Source: centers for disease control and prevention (CDC): PHIL)

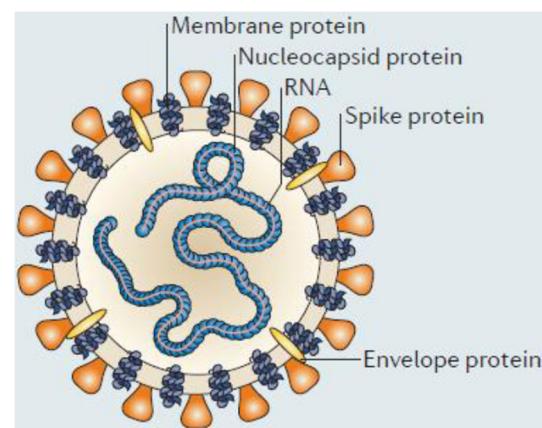
is more powerful and efficient, although the author deals with more complexity than that of the PSO algorithm.

Water evaporation optimization (WEO) using the evaporation of water molecules was incorporated in [44]. The basis of this algorithm is the vaporized tiny particle of water molecules on a solid surface with different wettability values. This algorithm consists of two optimization phases: droplet evaporation and monolayer evaporation.

Heat transfer search (HTS) using heat transfer and the law of thermodynamics was provided in [45]. The population of this algorithm is the same molecules from the system that absorb each other through different modes of heat transfer such as conduction, convection, and radiation.

#### 1.4 Natural phenomena-based algorithms (NPA)

The very optimistic method (VOMMI) using real-life practices of successful persons was discussed in [46]. The author claims that VOMMI is a simpler and more efficient algorithm than many of the recently proposed algorithms.



**Fig. 2** Virus structure of a typical coronavirus [58]

The optimization process of this algorithm is based on the best particles in the current iteration of the algorithm. It also uses two factors: chance and effort. Volleyball premier league algorithm (VPLA) using competition and interaction among volleyball teams [47] is composed of learning, substitution, and coaching terms like the real world, which has made the algorithm achieve acceptable and effective results compared to other algorithms.

Supernova optimizer (SO) using supernova phenomena [48] is a new type of artificial intelligence algorithm based on inspiration from a supernova physical phenomenon. The main feature of this algorithm is to improve exploration, exploitation, and local minima avoidance effectively compared to other algorithms.

Rain-fall optimization (RFO) using the behavior of raindrops [49] is an algorithm without a control parameter. As its name implies, the population of the algorithm is composed of raindrops and the basis of the operation of this algorithm is that water particles tend to enter the pool through the deepest and shortest pass when they fall on a rock physically. Particles always consider the position of their neighbors and mimic their movement.

Queuing search algorithm (QSA) using human activities in queuing [50] has no control parameter. The author in [50] models several social and communication phenomena. For instance, customers or the population of the algorithm are actively and purposefully lined up to provide faster and better service. Each customer service is mainly affected by the performance of itself or its employees, and ultimately a customer can be influenced by other people during service.

The galactic swarm optimization (GSO) algorithm using the motion of galaxies, stars, and superclusters of galaxies under the influence of gravity [51] is a new version of PSO algorithms and is based on exploration and exploitation phases.

Turbulent flow of water-based optimization (TFWO) via nature search phenomenon, i.e., whirlpools, is introduced in [52]. Whirlpools disturb water flow. Whirlpools have a central cavity that swallows the surrounding objects. The population of the algorithm is composed of these objects, and the whirlpools represent the number of groups of the algorithm. The central cavity plays the role of the best member of each group, sucking objects toward itself, making them better positioned according to the laws of particle optimization.

The application of optimization methods and algorithms in various fields of science from economics to chemistry, etc. is evident and has been tested, implemented, and analyzed in various design and optimization problems. To achieve an acceptable optimal solution, researchers urgently need a strong, fast, as well as simple and robust algorithm for optimization in various fields. In recent years, many optimization methods and metaheuristic algorithms inspired by various phenomena have been introduced [3].

Football game-based optimization (FGBO) imitates football games, in which the population produced in the algorithm consists of football clubs while the players are considered as variables. FGBO has four stages, namely, league holding, player transfer, practice, and promotion and relegation [60]. Wild geese algorithm (WGA) mimics the natural life and death of wild geese [61]. Aquila optimizer (AO) imitates the foraging behavior of Aquila [62]. Hunger games search (HGS) is another algorithm that makes use of hunger driven connections [63].

Red fox optimization (RFO) is an inspiration from how foxes live and hunt their prey [64].

Arithmetic optimization algorithm (AOA) is another method that uses mathematic operators including multiplication (M), division (D), subtraction (S), and addition (A) [65].

**Table 1** A summary of the inspiration for the proposed CMOA algorithm

Phase modeled in CMOA algorithm	Phase in internal and genetic changes of coronavirus
Algorithm population	Mass of viruses
Population position change and population movement within the problem range	Internal and genetic changes of the virus
Repeat algorithm	Time of change and life of viruses
The movement of a member to a member closest to itself and its repair by acquiring the characteristics of the virus	Crossover
The best member of the population	Inactivated virus
Creating a new member through the action of mutation	Cross-activation: The intersection of the active virus genome with the inactive virus genome
The target member moves rapidly due to the defined evolutionary operator and changes the position in the dimensions of the problem	Incremental reactivation
Each member changes position based on the information of the closest member	Non-genetic mutations
Each member randomly picks up information from the other two members	Genotypic mixing

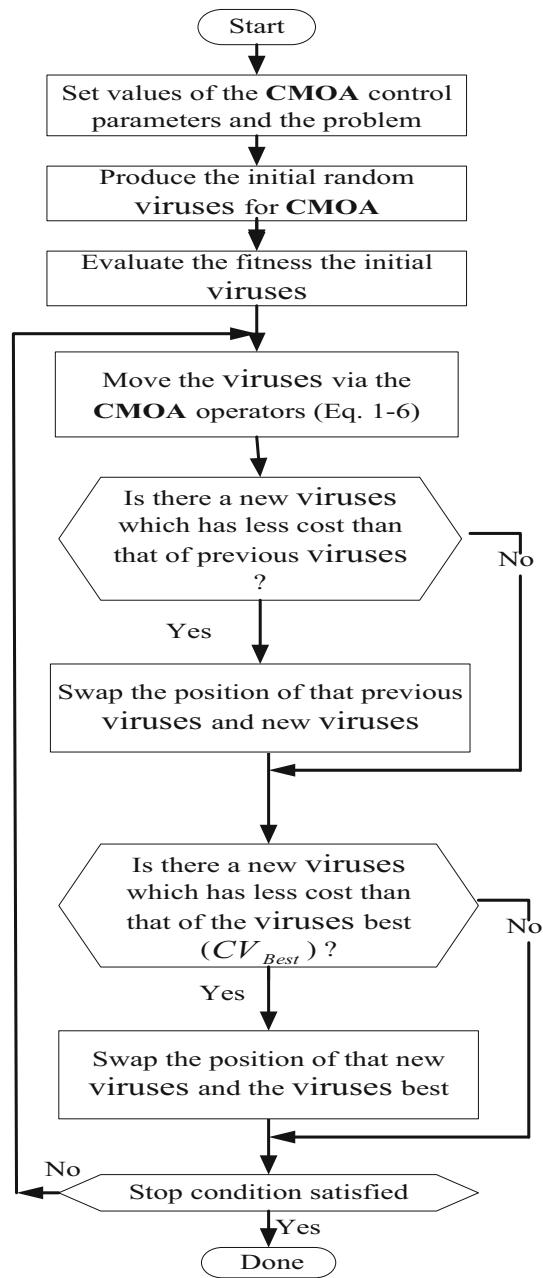
These references [67] model coronavirus spread and how they infect healthy individuals, in which the virus leads to a large population of the infected who pass away or spread the infection. The infected grows over time, but considering social distance or death toll and the number of saved ones, the population of the infected decreases gradually. Coronavirus optimization algorithms evolve and result in mutations and search the space.

Coronavirus her immunity optimizer (CHIO) is a novel algorithm as a solution to address the coronavirus pandemic. Social distancing and herd immunity are the two strategies adopted by CHIO [66]. In a general conclusion, it can be said that in some previous algorithms, the user encounters an imbalance between the local and global search operator in the algorithm for better optimization, which results in trapping in the local optimal solution, or some algorithms require complex calculations in which the user has trouble using that algorithm. Also, the parameter values may not be selected correctly and harm the performance of the algorithm in the optimization. Moreover, some algorithms need a longer time and more iterations to reach an acceptable solution.

For example, equilibrium optimizer (EO) [90] is one of these novel and successful algorithms that gained the attention of researchers, and it was used in many papers such as [116–120].

For fractional-order proportional–integral–derivative plus derivative (FOPIDD) controller [118], a self-adaptive hybrid equilibrium optimizer (SAHEO) for distributed jobshop rescheduling problem [119], enhanced equilibrium optimization ( $E^2O$ ) to application engineering [116], stock market prediction [120], a modified EO for optimal allocation of PVs in radial distribution systems [117]

According to the no free lunch (NFL) theorem, there is no perfect algorithm for solving all optimization problems. Consequently, novel and more efficient algorithms are needed. To this end, the present study models and presents a new metaheuristic solution called CMOA, the results of which are discussed in the following, indicating the value of this algorithm for future studies. To show the performance and optimization effect of the proposed algorithm for optimizing functions in the real world, the CEC2014 standard test functions are employed, which include 30 different functions including unimodal, simple multimodal, hybrid, and composition test functions, which cover most real functions of the real world. On the other hand, to highlight the superior performance of the proposed algorithm over other new algorithms and to make a comparison between them, six recently published standard algorithms are considered in this study, including artificial infectious disease optimization (AIDO), invasive tumor growth optimization (ITGO), rain-fall optimization algorithm (RFOA),

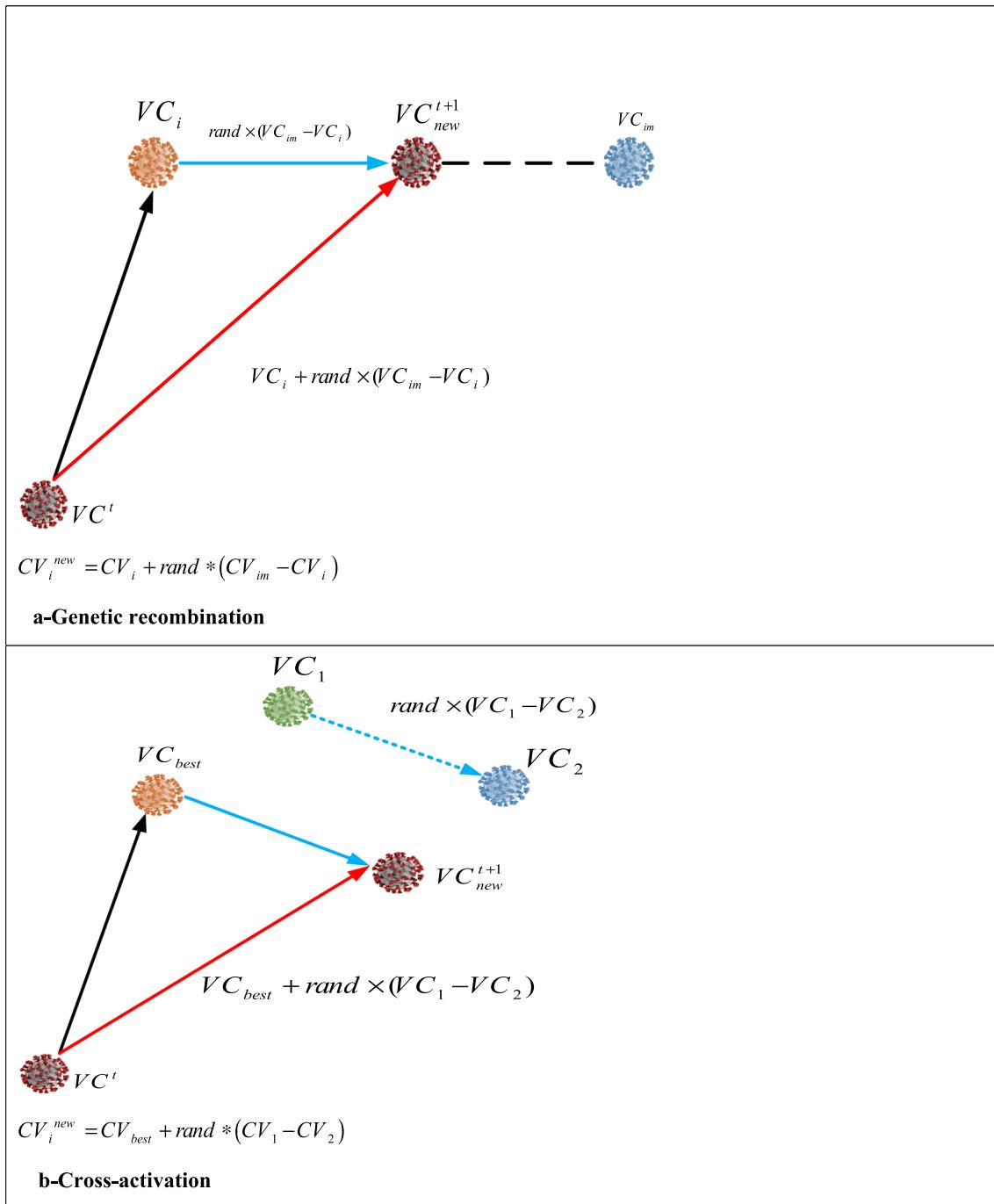


**Fig. 3** Flowchart of the CMOA algorithm

sine cosine algorithm (SCA), and cooperation search algorithm (CSA).

## 2 Coronavirus metamorphosis optimization algorithm (CMOA)

Coronaviruses (COVIDs), first identified in the 1960s, are large ribonucleic acid (RNA) viruses that infect a wide range of domestic animals as well as bats [53]. As it is obvious to everyone, COVID-19 is a deadly and very

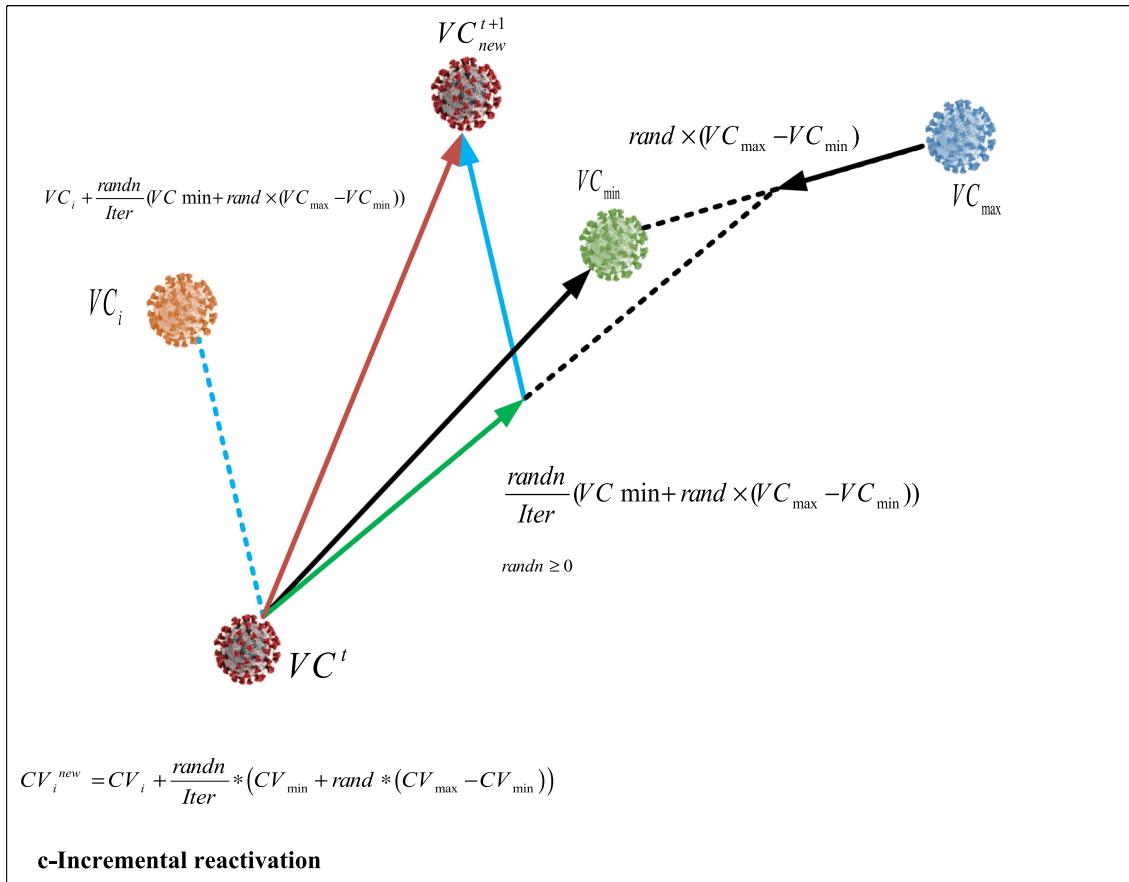


**Fig. 4** Coronavirus updating (Bilateral genetic mutations in viruses)

rapidly spreading virus that has affected the whole world today, both rich and poor [54, 55]. Since the development of the COVID-19 pandemic, different mutations have been identified in the genome of the virus that has led to the development of various variants of the causative agent worldwide. The mutations have made it easier for the virus to attach to human cells and increase the spread of the virus. The coronavirus has different mutations based on its

genetic structure, and the virus is growing every day [68–71].

As the coronavirus multiplies and spreads, mutations in the virus with the RNA genome of the virus increase so that one or two random mutations occur in the virus each month, causing each mutation with different infectivity [72, 73]. Some other mutations can multiply and infect the virus and in turn, lead to increased immune escape [71]. On



**Fig. 4** continued

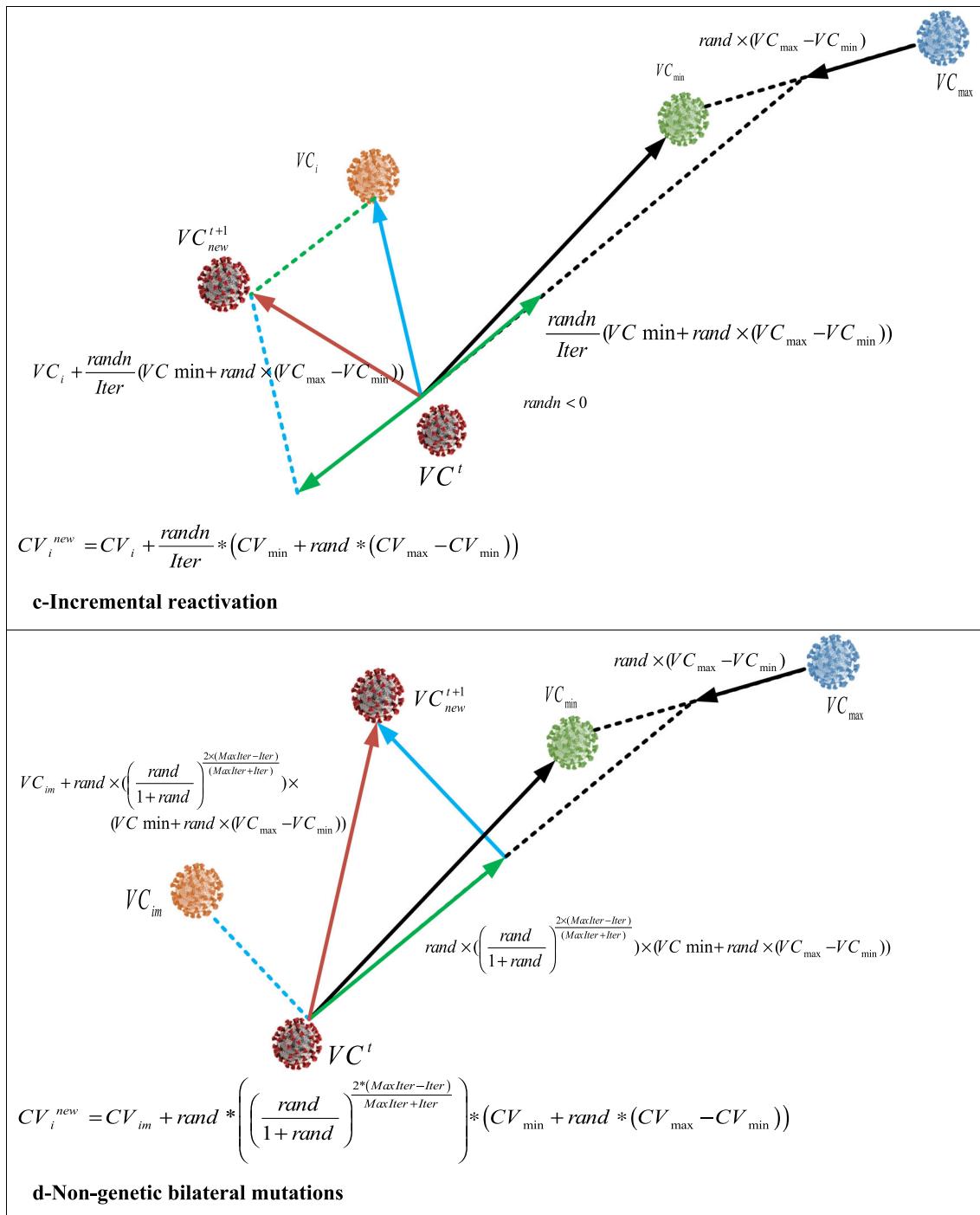
the other hand, humans have started to fight this virus very quickly. COVID-19, like any other virus, has been able to survive and spread by adapting to different environments and conditions such as different seasons. This, i.e., internal and genetic changes and adaptations of COVID-19 for survival, establishes the basis of the proposed algorithm, meaning that the members of the algorithm play the role of this virus, which evolve and reach better positions by evolving and spreading. Coronaviruses search and optimize with their internal changes to make them more resilient and more viable. In this paper, unlike the real world where the coronavirus plays a destructive role, it plays the effective role of an optimizer and a searcher. The environment around us is made up of different conditions such as the effect of heat or sunlight on viruses. Hence, viruses will adapt to these conditions over time to survive or change internally, weaken and disappear, or start reproducing and spreading in a suitable place [56, 57]. On the other side, the structure of the coronavirus, like any other population structure, the closest member or the most similar member to each member, will play an important role in its evolution and mutation or path, which is one of the basic foundations of this algorithm. Viruses also use different mutation

structures to evolve depending on different conditions. Although it is not yet known exactly how long COVID-19 stays on living surfaces, it probably behaves similarly to other coronaviruses. Figure 1 illustrates the general family of these types of viruses. A robust mathematical model inspired by this behavior of virus is adopted for the suggested algorithm. Most importantly, we are inspired here by the bilateral genetic mutation of viruses to implement genetic modifications in the coronavirus algorithm (Fig. 2).

Utilizing all the motion and mutation models of the virus mentioned above, a powerful and new standard algorithm has been extracted in this study, called metamorphosis optimization algorithm (CMOA). The mathematical modeling for the optimization process and the implementation of the proposed CMOA is described in the following subsections.

### 3 Mathematical modeling of the CMOA algorithm

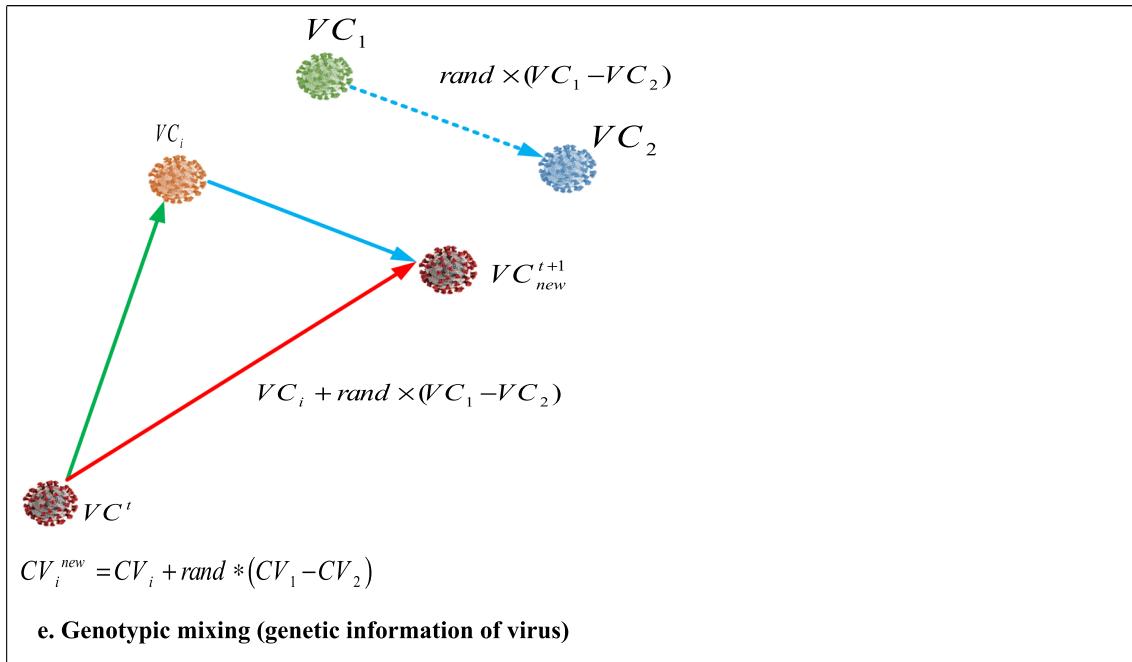
We need to consider many hypotheses to create a meta-heuristic algorithm. In the CMOA algorithm, several hypotheses will be described below. The CMOA algorithm

**Fig. 4** continued

starts with an initial population  $CV_i$  for  $i = 1: Npop$ ,  $Npop$  the population number of the algorithm, which is randomly generated. This initial population of the algorithm, which will remain constant throughout the algorithm, will play the same role as coronaviruses, and then bilateral mutations in the corona genome of the viruses will occur.

### 3.1 Bilateral genetic mutations in viruses

Bilateral genetic mutations in viruses occur when the genomes of two active viruses enter the cell at the same time and infect it, and the host cell replicates both. In such cases, the probability of bilateral genetic mutations between the two viruses is very high. In general, in this



**Fig. 4** continued

case, the genome of viruses loses some part and gains some other part. In such cases, several possibilities can be considered. Inspired by these possibilities and considering several hypotheses to transfer to the optimization space described below, we created the CMOA. A summary of the mathematical modeling and inspiration for the proposed CMOA algorithm based on the internal and genetic variations of the virus is summarized in Table 1.

### 3.1.1 Genetic recombination

Recombinant is the cross-repair of parts of the genome of two active viruses that have entered the cell at the same time and infected it. In fact, as in the real world, each virus usually interacts with the virus closest to it. Hence, we have assumed and modeled the crossover phase as the movement of the target organ to the nearest organ or virus to improve its position.

For the  $i$ th virus ' $CV_i$ ', we selected "cross-virus",  $CV_{im}$ , as the first virus with the closest location to  $CV_i$ . This cross virus,  $CV_{im}$ , is determined using the following equation:

$$dij = \left| \sum_{k=1}^D CV_{ik} - \sum_{k=1}^D CV_{jk} \right|, j = 1 : Npop \quad (1)$$

In the above equation,  $D$  is the number of dimensions of the problem to be optimized. The first virus to produce the smallest value of  $dij$  will be selected as  $CV_{im}$  for the  $i$ th member. In this case, the coronavirus will be updated and changed as follows:

$$CV_i^{new} = CV_i + rand * (CV_{im} - CV_i) \quad (2)$$

where  $rand$  denotes generates random numbers between 0 and 1 equal to the number of dimensions of the problem.

### 3.1.2 Cross-activation

The intersection of the active virus genome with the inactive virus genome, which simultaneously entered the cell and infected it, sometimes leads to the intersection of part of their genome, and both viruses from these changes acquire new characteristics. As mentioned earlier, we need to consider many assumptions to create a meta-heuristic algorithm. Therefore, the intersection of the active virus genome with the inactive virus genome, which may be one of the two viruses that lead to a new virus, is assumed to be the best member of the population by moving and infiltrating members and viruses. This will lead to a new virus in the process, which may accept or reject the new member based on its position.

Here the inactive virus is assumed the best member of the population, that is, the member with the minimum value of the objective function,  $CV_{best}$ . In this case, if such a reaction occurs, this operation is modeled as follows by selecting two other random members, namely,  $CV_1$  and  $CV_2$ :

$$CV_i^{new} = CV_{best} + rand * (CV_1 - CV_2) \quad (3)$$

**Table 2** CEC2014 test functions

Function name	Function formulation
<i>Unimodal functions</i>	Note: $M$ -rotation matrix
High conditioned elliptic function	$F_1(x) = f_1(M(x - o_1)) + 100$
Rotated bent cigar function	$F_2(x) = f_2(M(x - o_2)) + 200$
Rotated discus function	$F_3(x) = f_3(M(x - o_3)) + 300$
<i>Multimodal functions</i>	
Shifted and rotated rosenbrocks function	$F_4(x) = f_4(M(\frac{2.048(x-o_4)}{100}) + 1) + 400$
Shifted and rotated ackleys function	$F_5(x) = f_5(M(x - o_5)) + 500$
Shifted and rotated weierstrass function	$F_6(x) = f_6(M(\frac{0.5(x-o_6)}{100}) + 1) + 600$
Shifted and rotated griewanks function	$F_7(x) = f_7(M(\frac{600(x-o_7)}{100}) + 1) + 700$
Shifted rastrigins function	$F_8(x) = f_8(\frac{5.12(x-o_8)}{100}) + 800$
Shifted and rotated rastrigins function	$F_9(x) = f_9(M(\frac{5.12(x-o_9)}{100})) + 900$
Shifted Schwefels Function	$F_{10}(x) = f_{10}(\frac{1000(x-o_{10})}{100}) + 1000$
Shifted and Rotated Schwefels Function	$F_{11}(x) = f_{11}(M(\frac{1000(x-o_{11})}{100})) + 1100$
Shifted and Rotated Katasuura Function	$F_{12}(x) = f_{12}(M(\frac{5(x-o_{12})}{100})) + 1200$
Shifted and Rotated Happy Cat Function	$F_{13}(x) = f_{13}(M(\frac{5(x-o_{13})}{100})) + 1300$
Shifted and Rotated HGBat Function	$F_{14}(x) = f_{14}(M(\frac{5(x-o_{14})}{100})) + 1400$
Shifted and Rotated Expanded Griewanks plus Rosenbrocks Function	$F_{15}(x) = f_{15}(M(\frac{5(x-o_{15})}{100})) + 1500$
Shifted and Rotated Expanded Ssaffers F6 Function	$F_{16}(x) = f_{16}(M(x - o_{16})) + 1600$
<i>Hybrid functions</i>	
$F_{17} = f_9(M_1z_1) + f_8(M_2z_2) + f_1(M_3z_3) + 1700$	$p = [0.3, 0.3, 0.4]$
$F_{18} = f_2(M_1z_1) + f_{12}(M_2z_2) + f_8(M_3z_3) + 1800$	$p = [0.3, 0.3, 0.4]$
$F_{19} = f_7(M_1z_1) + f_6(M_2z_2) + f_4(M_3z_3) + f_{14}(M_4z_4) + 1900$	$p = [0.2, 0.2, 0.3, 0.3]$
$F_{20} = f_{12}(M_1z_1) + f_3(M_2z_2) + f_{13}(M_3z_3) + f_8(M_4z_4) + 2000$	$p = [0.2, 0.2, 0.3, 0.3]$
$F_{21} = f_{14}(M_1z_1) + f_{12}(M_2z_2) + f_4(M_3z_3) + f_9(M_4z_4) + f_1(M_5z_5) + 2100$	$p = [0.1, 0.2, 0.2, 0.2, 0.3]$
$F_{22} = f_{10}(M_1z_1) + f_{11}(M_2z_2) + f_{13}(M_3z_3) + f_9(M_4z_4) + f_5(M_5z_5) + 2200$	$p = [0.1, 0.2, 0.2, 0.2, 0.3]$
<i>Notes</i>	
$z_1 = [yS_1, \forall S_2, \dots, yS_{n1}],$	
$z_2 = [yS_{n1+1}, yS_{n1+2}, \dots, yS_{n1+n2}],$	
$z_n = \left[ yS_{\sum_{i=1}^{N-1} n_i + 1}, yS_{\sum_{i=1}^{N-1} n_i + 2}, \dots, yS_D \right],$	
$y = x - o_i, S == \text{randperm}(1 : D), p_i : \text{Percentage of } g_i(x)$	
$n_1 = [p1^D], n_2 = [p1^D], \dots, nN - 1 = [pN - 1^D], nN = D - \sum_{i=1}^{N-1} n_i$	
<i>Composition functions</i>	
$F_{23} = \omega_1 * F'_4(x) + \omega_2 * [1e^{-6}F'_1(x) + 100] + \omega_3 * [1e^{-26}F'_2(x) + 200] + \omega_4 * [1e^{-6}F'_3(x) + 300] + \omega_5 * [1e^{-6}F'_1(x) + 400] + 2300\sigma = [10, 20, 30, 40, 50]$	
$F_{24} = \omega_1 * F'_{10}(x) + \omega_2 * [F'_9(x) + 100] + \omega_3 * [F'_{14}(x) + 200] + 2400\sigma = [20, 20, 20]$	
$F_{25} = \omega_1 * 0.25F'_{11}(x) + \omega_2 * [F'_9(x) + 100] + \omega_3 * [1e^{-7}F'_1(x) + 200] + 2500\sigma = [10, 30, 50]$	
$F_{26} = \omega_1 * 0.25F'_{11}(x) + \omega_2 * [F'_{13}(x) + 100] + \omega_3 * [1e^{-7}F'_1(x) + 200] + \omega_4 * [2.5F'_6(x) + 300] + \omega_5 * [10F'_7(x) + 400] + 2600\sigma = [10, 10, 10, 10, 10]$	
$F_{27} = \omega_1 * 10F'_{14}(x) + \omega_2 * [10F'_9(x) + 100] + \omega_3 * [2.5F'_{11}(x) + 200] + \omega_4 * [25F'_6(x) + 300] + \omega_5 * [1e^{-6}F'_1(x) + 400] + 2700\sigma = [10, 10, 10, 20, 20]$	
$F_{28} = \omega_1 * 2.5F'_{15}(x) + \omega_2 * [10F'_{13}(x) + 100] + \omega_3 * [2.5F'_{11}(x) + 200] + \omega_4 * [5F'_{16}(x) + 300] + \omega_5 * [1e^{-6}F'_1(x) + 400] + 2800\sigma = [10, 20, 30, 40, 50]$	
$F_{29} = \omega_1 * F'_{17}(x) + \omega_2 * [F'_{18}(x) + 100] + \omega_3 * [F'_{19}(x) + 200] + 2900\sigma = [10, 30, 50]$	
$F_{30} = \omega_1 * F'_{20}(x) + \omega_2 * [F'_{21}(x) + 100] + \omega_3 * [F'_{22}(x) + 200] + 3000\sigma = [10, 30, 50]$	
<i>Notes</i>	
$\omega_i = \frac{1}{\sqrt{\sum_{j=1}^D (x_j - o_{ij})^2}} \exp(-\sum_{j=1}^D (x_j - o_{ij})^2 / 2D\sigma_i^2)$	

**Table 3** Mean statistical results of CMOA

Function		<i>Npop</i> = 30	<i>Npop</i> = 45	<i>Npop</i> = 60	<i>Npop</i> = 90
F1	Unimodal	<b>96,148</b>	109,387	110,735	168,801
		<b>1</b>	2	3	4
F2		0.000592	<b>0.000249</b>	0.0326	3.42
		2	<b>1</b>	3	4
F3		<b>2.58</b>	20.92	102.02	519.05
		<b>1</b>	2	3	4
F4	Simple multimodal	12.41	<b>0.1527</b>	0.1851	0.5436
		4	<b>1</b>	2	3
F5		<b>20.87</b>	<b>20.87</b>	20.94	20.94
		<b>1</b>	<b>1</b>	2	2
F6		21.09	19.06	14.64	<b>12.34</b>
		4	3	2	<b>1</b>
F7		0.0282	0.00369	0.00615	<b>0.0</b>
		4	2	3	<b>1</b>
F8		70.14	70.14	59.03	<b>55.22</b>
		3	3	2	<b>1</b>
F9		75.78	74.79	54.06	<b>52.57</b>
		4	3	2	<b>1</b>
F10		1943	2276	2281	<b>1690</b>
		2	3	4	<b>1</b>
F11		3505	<b>2901</b>	3500	3003
		4	<b>1</b>	3	2
F12		<b>1.71</b>	2.43	1.97	2.22
		<b>1</b>	4	2	3
F13		0.2703	0.2660	0.2232	<b>0.1916</b>
		4	3	2	<b>1</b>
F14		0.248	0.251	<b>0.219</b>	0.223
		3	4	<b>1</b>	2
F15		15.52	5.15	<b>4.77</b>	6.17
		4	2	<b>1</b>	3
F16		11.02	<b>10.89</b>	10.90	11.47
		3	<b>1</b>	2	4
F17	Hybrid	<b>11,821</b>	15,918	20,043	15,720
		<b>1</b>	3	4	2
F18		1213	622	<b>155</b>	368
		4	3	<b>1</b>	2
F19		7.60	6.00	<b>5.31</b>	5.58
		4	3	<b>1</b>	2
F20		<b>260</b>	353	581	1101
		<b>1</b>	2	3	4
F21		8589	11,313	5740	<b>8512</b>
		2	4	3	<b>1</b>
F22		235	<b>108</b>	117	136
		4	<b>1</b>	2	3

**Table 3** (continued)

Function		<i>Npop</i> = 30	<i>Npop</i> = 45	<i>Npop</i> = 60	<i>Npop</i> = 90
F23	Composition	<b>315.24</b> 1	<b>315.24</b> 1	<b>315.24</b> 1	<b>315.24</b> 1
F24		231 4	227 3	224 2	<b>223</b> <b>1</b>
F25		209 4	208 3	206 2	<b>205</b> <b>1</b>
F26		116.9 2	116.9 2	<b>100.34</b> <b>1</b>	<b>100.34</b> <b>1</b>
F27		401.33 3	445.17 4	<b>401.06</b> <b>1</b>	401.13 2
F28		1105 4	1050 3	975 <b>1</b>	<b>944</b> <b>1</b>
F29		1,805,616 4	1624 3	<b>1243</b> <b>1</b>	1302 2
F30		2866 4	2489 3	2124 2	<b>1566</b> <b>1</b>
<i>Nb/Nw/Mean</i>		7/16/2.933	7/4/2.467	8/2/2.033	13/5/2.033

Results of Coronavirus Metamorphosis Optimization Algorithm (CMOA)

### 3.1.3 Incremental reactivation

Sometimes inactive viruses enter a cell and begin to repair and amend each other, resulting in an active virus with new characteristics. Consequently, a new active and violent virus is likely to emerge. We have assumed and modeled the probable phase of incremental reactivation for each virus equivalent to the evolutionary phase in evolutionary algorithms. That is, here in each iteration, as in evolutionary algorithms, if this phase occurs, the virus begins to repair itself with evolutionary action, and as a result, an active virus with new properties is created. According to its new fitting function is accepted if it is stronger than the previous virus and otherwise rejected. We assume that the virus or organ in question has grown rapidly and the environment around the neighboring organ is considered for the virus to evolve. The new position and structure of the *i*th member are obtained from the following equation:

$$CV_i^{new} = CV_i + \frac{randn}{Iter} * (CV_{min} + rand * (CV_{max} - CV_{min})) \quad (4)$$

where *randn* represents a one-dimensional normal distribution with a random value between infinitely negative and infinitely positive. Also, *CV<sub>min</sub>* and *CV<sub>max</sub>* are the minimum and maximum values of the desired problem space with dimension *D* and *Iter* is the current iteration number of the algorithm.

### 3.1.4 Non-genetic bilateral mutations

When two viruses enter a cell and multiply at the same time, phenotypic mixing occurs if the genetic information of the first virus, named *CV<sub>im</sub>*, is replaced in the capsid of the *i*th virus. This phase in the proposed algorithm is such models and assumed that the *i*th member mimics and follows the closest member to it based on the first equation, and tries to obtain the evolved information of this member and gain a better position, i.e.:

$$CV_i^{new} = CV_{im} + rand * \left( \left( \frac{rand}{1 + rand} \right)^{\frac{2 * (MaxIter - Iter)}{MaxIter + Iter}} \right) * (CV_{min} + rand * (CV_{max} - CV_{min})) \quad (5)$$

In the above equation, *MaxIter* represents the maximum number of iterations of the algorithm.

### 3.1.5 Genotypic mixing (genetic information of virus)

Sometimes a cell mixes with two different viruses at the same time, in which case one virus genetically removes the information from the other two viruses, i.e., *CV<sub>1</sub>* and *CV<sub>2</sub>*, and genotypic mixing occurs. In other words, in optimization science and inspirational algorithms in random members 1 and 2 of the total population of the algorithm, this leads to genotype mixing, or in other words in optimization science, causes the evolution and improvement of the position of member *i*. However, this new member will be accepted and will replace the previous member if the

**Table 4** Best statistical results of CMOA

Function		<i>Npop</i> = 30	<i>Npop</i> = 45	<i>Npop</i> = 60	<i>Npop</i> = 90
F1	Unimodal	<b>12,471</b>	41,545	43,922	85,579
		<b>1</b>	2	3	4
F2		<b>4.24e – 07</b>	8.37e – 06	9.27e – 05	0.0434
		<b>1</b>	2	3	4
F3		<b>0.0975</b>	1.31	6.95	290
		<b>1</b>	2	3	4
F4	Simple multimodal	0.0346	<b>7.46e – 05</b>	0.00741	0.00227
		4	<b>1</b>	3	2
F5		20.83	<b>20.72</b>	20.84	20.87
		2	<b>1</b>	3	4
F6		17.64	13.99	9.77	<b>9.08</b>
		4	3	2	<b>1</b>
F7		<b>0.0</b>	<b>0.0</b>	<b>0.0</b>	<b>0.0</b>
		<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
F8		53.73	59.69	47.75	<b>32.83</b>
		3	4	2	<b>1</b>
F9		48.75	59.69	46.76	<b>43.77</b>
		3	4	2	<b>1</b>
F10		<b>975</b>	1657	2013	1192
		<b>1</b>	3	4	2
F11		2780	<b>2004</b>	2954	2264
		3	<b>1</b>	4	2
F12		0.779	2.073	<b>0.4481</b>	1.95
		2	4	<b>1</b>	3
F13		0.204	0.184	<b>0.1167</b>	0.156
		4	3	<b>1</b>	2
F14		0.193	0.175	<b>0.124</b>	0.202
		3	2	<b>1</b>	4
F15		8.14	4.09	<b>3.50</b>	4.82
		4	2	<b>1</b>	3
F16		10.45	<b>10.10</b>	10.45	10.33
		3	<b>1</b>	3	2
F17	Hybrid	<b>1661</b>	5663	5272	8392
		<b>1</b>	3	2	4
F18		175.7	166	<b>13.50</b>	59.93
		4	3	<b>1</b>	2
F19		5.85	4.22	<b>3.64</b>	4.24
		4	2	<b>1</b>	3
F20		<b>131.9</b>	259.3	238.3	276.5
		<b>1</b>	3	2	4
F21		<b>1034</b>	4973	1074	1428
		<b>1</b>	4	2	3
F22		28.34	<b>25.61</b>	25.99	63.63
		3	<b>1</b>	2	4
F23	Composition	<b>315.24</b>	<b>315.24</b>	<b>315.24</b>	<b>315.24</b>
		<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>

**Table 4** (continued)

Function	<i>Npop</i> = 30	<i>Npop</i> = 45	<i>Npop</i> = 60	<i>Npop</i> = 90
F24	225.07	223.04	222.06	<b>221.83</b>
	4	3	2	<b>1</b>
F25	206.14	203.55	205.08	<b>200.0</b>
	4	2	3	<b>1</b>
F26	100.19	100.19	100.30	<b>100.13</b>
	2	2	3	<b>1</b>
F27	400.62	<b>400.58</b>	400.78	401.06
	2	<b>1</b>	3	4
F28	1005	957	937	<b>874</b>
	4	3	2	<b>1</b>
F29	1138	1059	1111	<b>893</b>
	4	2	3	<b>1</b>
F30	1361	<b>728</b>	935	873
	4	<b>1</b>	3	2
<i>Nb/ Nw/Mean</i>	9/12/2.633	9/4/2.233	8/3/2.333	10/9/2.4

Results of Coronavirus Metamorphosis Optimization Algorithm (CMOA)

new member has a better position and a better objective value than the previous member.

Which is modeled as the following relation:

$$CV_i^{new} = CV_i + rand * (CV_1 - CV_2) \quad (6)$$

The important point is that only one of the above will be done randomly in each iteration for a coronavirus. Here we will introduce, hypothesize, and model four inspirational possibilities that will determine which of the above changes will play a role in coronavirus transformation. This is described below:

$$P1 = \frac{\text{Exp}\left(\frac{-\text{Iter}}{2*(\text{MaxIter}+\text{Iter})}\right)}{\left(\frac{1}{1+rand}\right)^{\frac{2*(\text{MaxIter}-\text{Iter})}{\text{MaxIter}+\text{Iter}}} + \text{Exp}\left(\frac{-\text{Iter}}{2*(\text{MaxIter}+\text{Iter})}\right)} \quad (7)$$

$$P2 = \frac{\left(\frac{1}{1+rand}\right)^{\frac{2*(\text{MaxIter}-\text{Iter})}{\text{MaxIter}+\text{Iter}}}}{\left(\frac{1}{1+rand}\right)^{\frac{2*(\text{MaxIter}-\text{Iter})}{\text{MaxIter}+\text{Iter}}} + \text{Exp}\left(\frac{-\text{Iter}}{2*(\text{MaxIter}+\text{Iter})}\right)} \quad (8)$$

$$P3 = \frac{\text{Exp}\left(\frac{-|f(CV_i)|}{\text{Iter}}\right)}{\left(\frac{1}{1+rand}\right)^{\frac{2*(\text{MaxIter}-\text{Iter})}{\text{MaxIter}+\text{Iter}}} + \text{Exp}\left(\frac{-\text{Iter}}{2*(\text{MaxIter}+\text{Iter})}\right)} \quad (9)$$

$$P4 = \frac{\text{Exp}\left(\frac{-|f(CV_{im})|}{\text{Iter}}\right)}{\left(\frac{1}{1+rand}\right)^{\frac{2*(\text{MaxIter}-\text{Iter})}{\text{MaxIter}+\text{Iter}}} + \text{Exp}\left(\frac{-\text{Iter}}{2*(\text{MaxIter}+\text{Iter})}\right)} \quad (10)$$

$$PE_i = rand^{rand} \quad (11)$$

Equations (8–10) model the probability values, in which  $P1$  is the probability of recombination in the coronavirus,  $P2$  is the probability of cross-activation, both of which are equal for all

members.  $P3$  is the probability of occurrence of incremental reactivation in the  $i$ th coronavirus, and  $P4$  is the probability of occurrence of bilateral non-genetic mutations in the  $i$ th coronavirus. As can be seen from the appearance of  $P3$  and  $P4$ , the two can be different for particles. Furthermore, if none of the above probabilities are met, genotypic mixing will occur in the coronavirus. Also,  $PE_i$  indicates the environmental conditions for the occurrence of one of the transformations for the coronavirus. Now after obtaining the new structure and position of the coronavirus, if the value of the objective function of this member is better than the current position of the coronavirus, it will be replaced, otherwise, the coronavirus or a member of some population will maintain its current position and this cycle continues until a certain number of iterations or an acceptable optimal solution determined by the user is reached. And if the new position of the virus  $CV_i^{new}$  has a better fitness function than the current member ( $CV_i$ ), it replaces the current position. Also, it should be noted that the best member obtained at each iteration is stored the best solution ( $CV_{best}$ ) and is printed out. This process is shown as follows:

```

if f(CV_i^{new}) ≤ f(CV_i)
    CV_i = CV_i^{new}
    f(CV_i) = f(CV_i^{new})
if f(CV_i) ≤ f(CV_{best})
    CV_{best} = CV_i
    f(CV_{best}) = f(CV_i)
end
end

```

### 3.2 The following presents a summary of the CMOA algorithm pseudocode:

**Step 1** Generate the  $N_{pop}$  number of initial populations or viruses of the CMOA algorithm and calculate their fitness

genetic transformation and mutation for the  $i$ th member, as well as for the whole population. If the new position of the new member had a better objective function, it substitutes the current position of that member.

```

for i = 1:Npop
    CVim, P1, P2, P3, P4 and PEi
    if PEi < P1
        CVinew = CVi + rand * (CVim - CVi)
    elseif PEi < P2
        CVinew = CVbest + rand * (CV1 - CV2)
    elseif PEi < P3
        CVinew = CVi + randn / Iter * (CVmin + rand * (CVmax - CVmin))
    elseif PEi < P4
        CVinew = CVim + rand * (rand / (1 + rand))2 * (MaxIter - Iter) / MaxIter + Iter * (CVmin + rand * (CVmax - CVmin))
    else
        CVinew = CVi + rand * (CV1 - CV2)
    end
    if f(CVinew) ≤ f(CVi)
        CVi = CVinew
        f(CVi) = f(CVinew)
        if f(CVi) ≤ f(CVBest)
            CVBest = CVi
            f(CVBest) = f(CVi)
        end
    end
end

```

or cost value.

```

for i = 1 : Npop
    CVi = CVmin + rand * (CVmax - CVmin)
end

```

**Step 2** Obtain the values of  $P1, P2, P3, P4$ , and  $PE_i$  and the  $CV_{im}$  member for the  $i$ th member and then do the

**Step 3** If the number of selected iterations for the algorithm is over, stop; otherwise, go to *Step 2* and continue this cycle.

In order to better illustrate the process of optimizing the proposed flowchart algorithm, its performance is shown in the following Figs. 3 and 4:

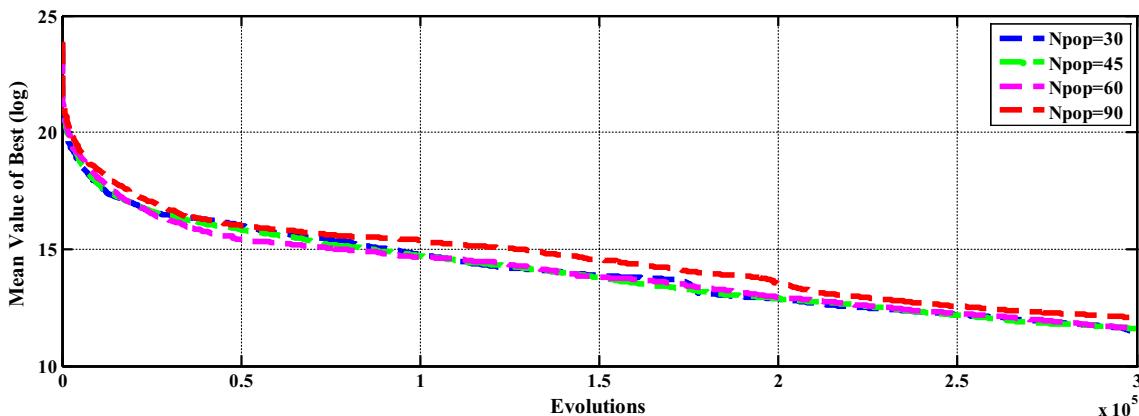
**Table 5** Standard deviation (STD) statistical results of CMOA

Function		<i>Npop</i> = 30	<i>Npop</i> = 45	<i>Npop</i> = 60	<i>Npop</i> = 90
F1	Unimodal	83,671	82,605	<b>60,015</b>	132,311
		3	2	<b>1</b>	4
F2		0.000855	<b>0.0003</b>	0.0632	3.46
		2	<b>1</b>	3	4
F3		<b>2.25</b>	14.56	103	227
		<b>1</b>	2	3	4
F4	Simple multimodal	27.3	<b>0.170</b>	0.24	0.940
		4	<b>1</b>	2	3
F5		0.0285	0.0773	0.056	0.052
		<b>1</b>	4	3	2
F6		3.8	3.22	3.66	<b>2.205</b>
		4	2	3	<b>1</b>
F7		0.0535	0.00903	0.0103	<b>0.0</b>
		4	2	3	<b>1</b>
F8		12.36	<b>9.11</b>	13.18	14.74
		2	<b>1</b>	3	4
F9		18.20	13.55	<b>6.50</b>	7.72
		4	3	<b>1</b>	2
F10		673	539	<b>295</b>	388
		4	3	<b>1</b>	2
F11		701	545	553	<b>481</b>
		4	2	3	<b>1</b>
F12		0.709	<b>0.191</b>	0.819	0.226
		3	<b>1</b>	4	2
F13		0.0457	0.0582	0.0663	<b>0.0265</b>
		2	3	4	<b>1</b>
F14		0.0728	0.062	0.0511	<b>0.0249</b>
		4	3	2	<b>1</b>
F15		7.99	<b>0.974</b>	1.56	1.36
		4	<b>1</b>	3	2
F16		<b>0.394</b>	0.485	0.4402	0.674
		<b>1</b>	3	2	4
F17	Hybrid	<b>8261</b>	10,595	13,500	8945
		<b>1</b>	3	4	2
F18		1163	602	<b>143</b>	584
		4	3	<b>1</b>	2
F19		1.78	1.46	<b>0.932</b>	1.51
		4	2	<b>1</b>	3
F20		132	<b>83.7</b>	311	885
		2	<b>1</b>	3	4
F21		5643	<b>3963</b>	4098	5489
		4	<b>1</b>	2	3
F22		116	89.0	61.2	<b>36.1</b>
		4	3	2	<b>1</b>

**Table 5** (continued)

Function		<i>Npop</i> = 30	<i>Npop</i> = 45	<i>Npop</i> = 60	<i>Npop</i> = 90
F23	Composition	2.36e – 13	2.49e – 13	<b>0.0</b>	1.87e – 13
		3	4	<b>1</b>	2
F24		5.37	6.23	<b>1.44</b>	1.46
		3	4	<b>1</b>	2
F25		2.12	2.76	<b>1.39</b>	2.93
		2	3	<b>1</b>	4
F26		40.69	40.71	<b>0.05083</b>	0.113
		3	4	<b>1</b>	2
F27		0.571	107.99	0.404	<b>0.053</b>
		3	4	2	<b>1</b>
F28		97.09	78.80	<b>27.16</b>	50.2
		4	3	<b>1</b>	2
F29		4,418,745	545	<b>152</b>	273
		4	3	<b>1</b>	2
F30		2619	1774	1360	<b>613</b>
		4	3	2	<b>1</b>
<i>Nb/ Nw/Mean</i>		4/15/3.067	7/5/2.5	11/3/2.133	8/7/2.3

Results of Coronavirus Metamorphosis Optimization Algorithm (CMOA)



**Fig. 5** Convergence characteristic of the CMOA algorithm for different populations for function number 1

## 4 Simulation results

### 4.1 Test functions

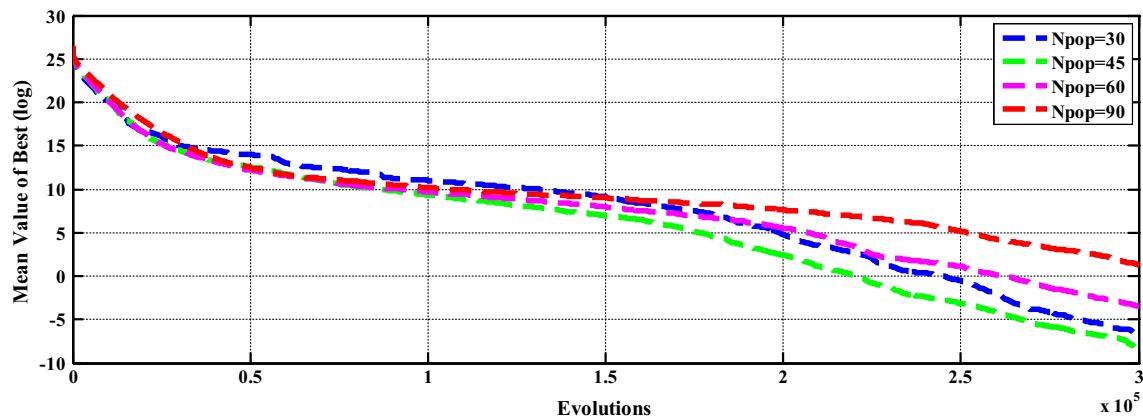
The real world is full of a wide range of new and emerging optimization problems with optimization functions much different and more complex than conventional test functions, which in some papers are used as selective functions to illustrate the performance of an algorithm. Hence, optimization researchers in recent years have introduced a variety of test functions that can incorporate the properties of real-time problem functions. CEC2014 test functions [59] are examples of these functions that have been adopted successfully in many papers. Thereby, this paper

employs these functions to test the performance of the proposed CMOA algorithm compared with other algorithms. A brief description of these functions is tabulated in Table 2.

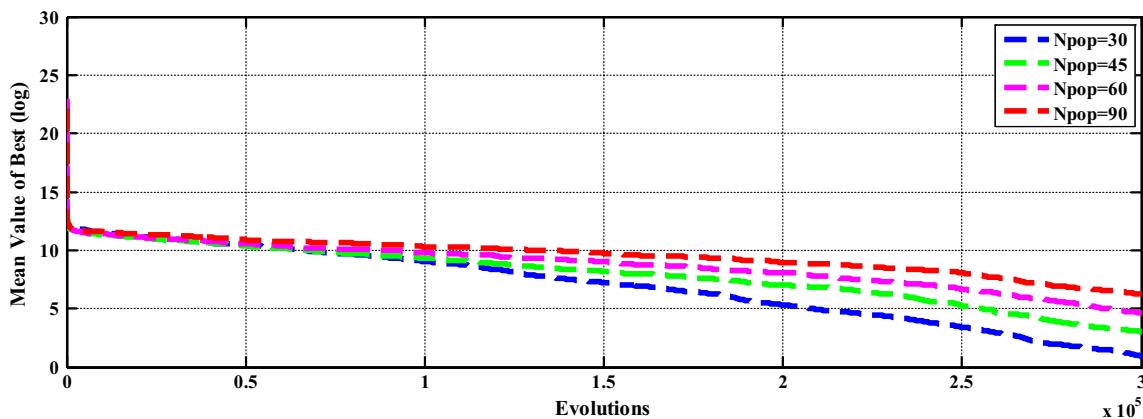
### 4.2 CMOA review

#### 4.2.1 Review of different populations in the CMOA algorithm

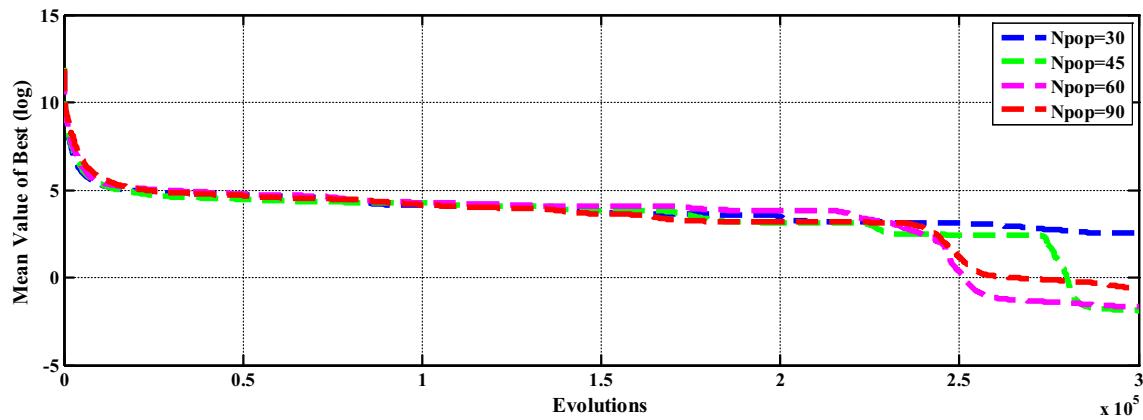
Fortunately, the proposed algorithm does not have a special and new control parameter, and like any other algorithm, the user must select the number of populations and iterations of the algorithm. Four different populations of 30, 45,



**Fig. 6** Convergence characteristic of the CMOA algorithm for different populations for function number 2



**Fig. 7** Convergence characteristic of the CMOA algorithm for different populations for function number 3

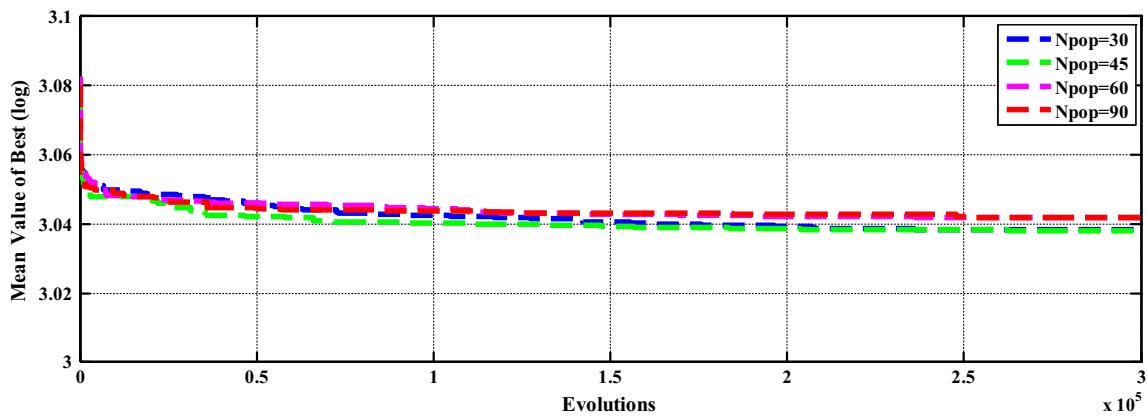


**Fig. 8** Convergence characteristic of the CMOA algorithm for different populations for function number 4

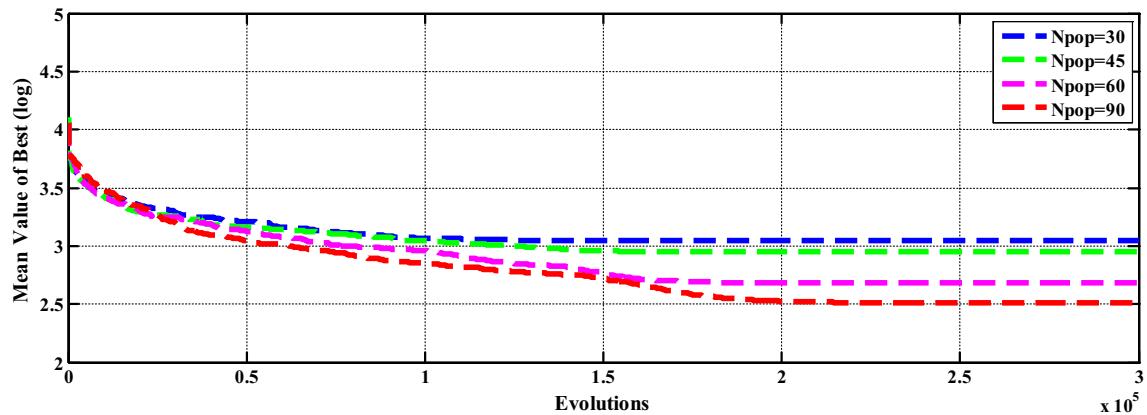
60, and 90, in this study, are selected for the CMOA algorithm. One system is used for all tests and the number of executions is 25, the number of dimensions for each function is 30, and the number of iterations is 300,000.

Tables 3, 4 and 5 present the results obtained for the mean, best, and standard deviation (STD) values,

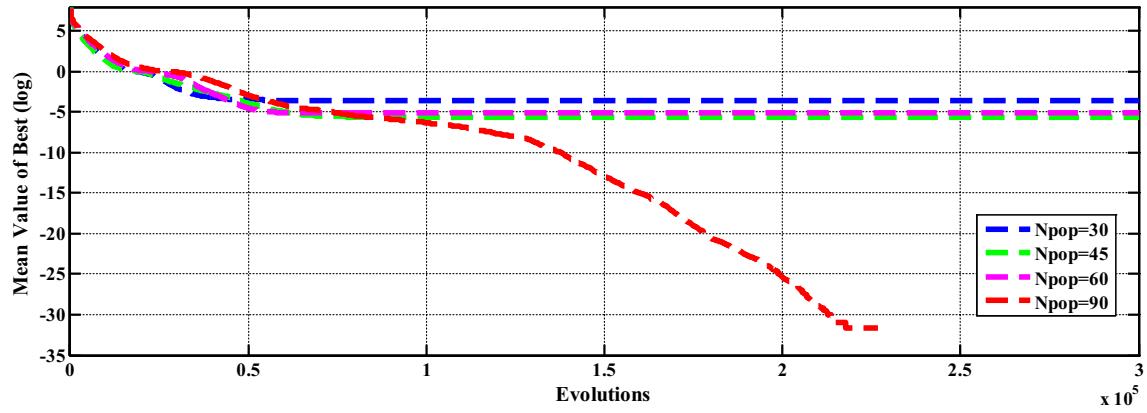
respectively. The last row of each table shows three parameters for each algorithm:  $N_b$  means the number of functions the algorithm obtained the best result,  $N_w$  is the number of times with the worst result, and Mean denotes the average rank of the algorithm for all functions. At a glance, it can be said that a smaller population performs



**Fig. 9** Convergence characteristic of the CMOA algorithm for different populations for function number 5



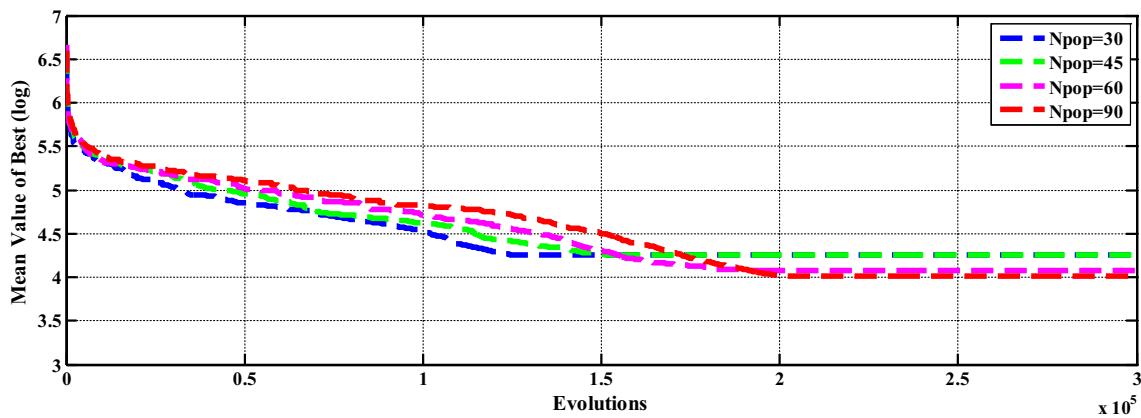
**Fig. 10** Convergence characteristic of the CMOA algorithm for different populations for function number 6



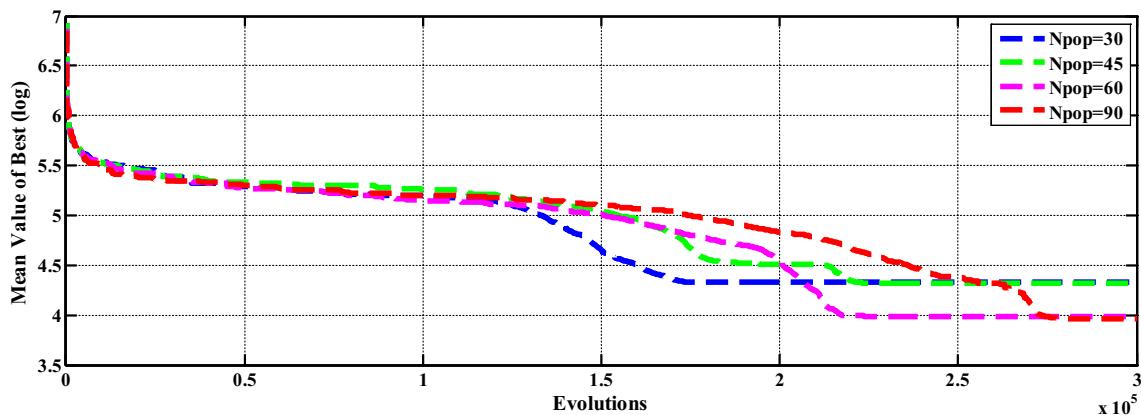
**Fig. 11** Convergence characteristic of the CMOA algorithm for different populations for function number 7

better for unimodal functions in a fair situation, and a larger population performs better for multimodal functions. Nonetheless, generally speaking, the choice of population 60 can be a good option for this algorithm in this dimension. Moreover, convergence characteristics for 30 test

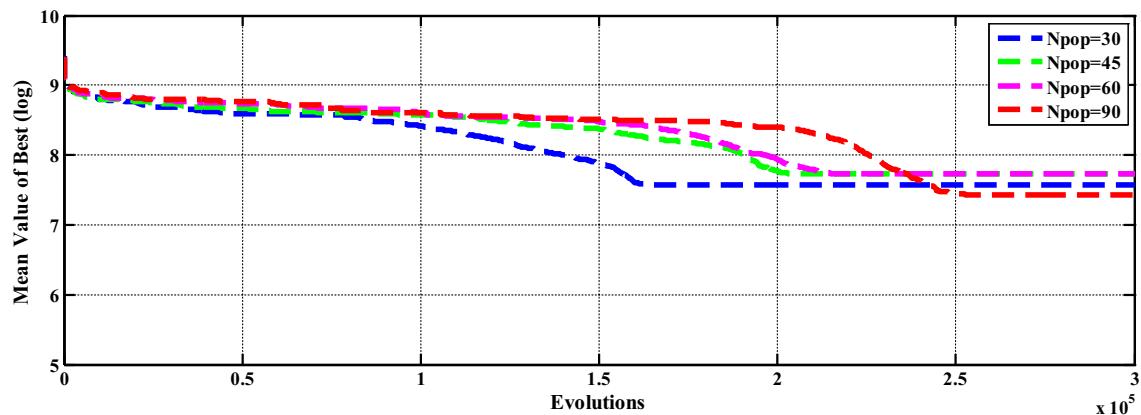
functions are also depicted in Figs. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33 and 34. These results indicate that for most test functions the algorithm is being optimized to the last moment and rarely traps in the local optima.



**Fig. 12** Convergence characteristic of the CMOA algorithm for different populations for function number 8



**Fig. 13** Convergence characteristic of the CMOA algorithm for different populations for function number 9



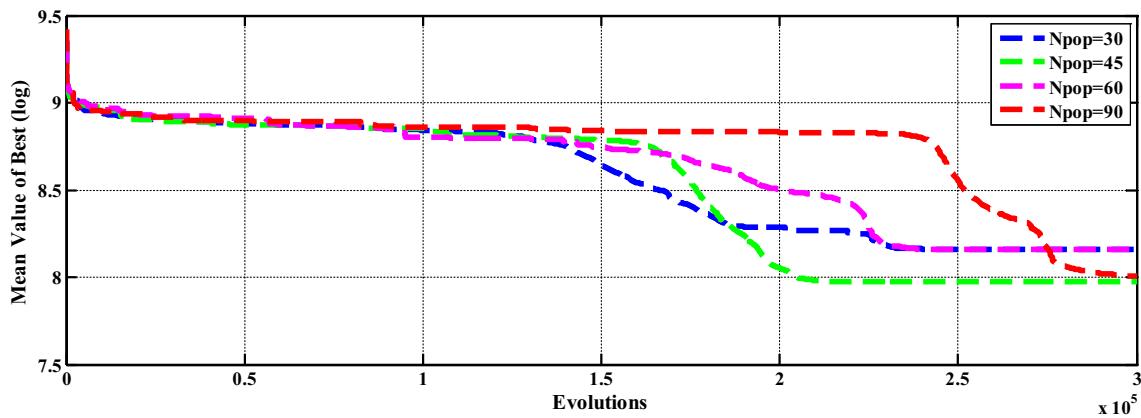
**Fig. 14** Convergence characteristic of the CMOA algorithm for different populations for function number 10

#### 4.2.2 Investigation of probability coefficients in CMOA algorithm

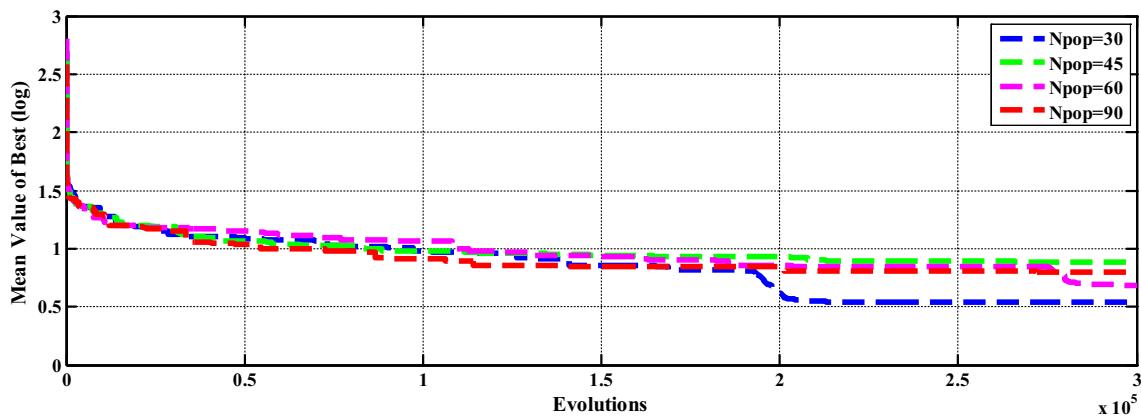
Figures 35, 36 and 37 show the characteristics of the changes in random functions  $P_1$ ,  $P_2$ ,  $P_3$ ,  $P_4$ , and  $PE$  for test function 1 during a random execution. As can be seen,  $PE$  occupies the whole space and plays the role of

environmental conditions, while  $P_1$  and  $P_2$  have opposite behavior.  $P_3$  and  $P_4$  have almost the same values here, and these two parameters obtain a value after approximately 175,000 iterations, thus affecting the function of the virus.

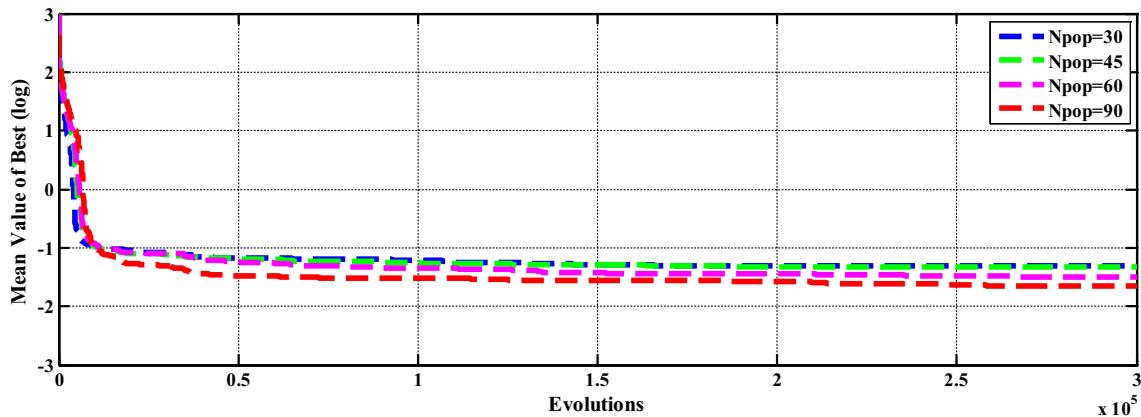
The coronavirus algorithm undergoes major changes in such evolution, which can be seen in the convergence figures, i.e., Figures 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,



**Fig. 15** Convergence characteristic of the CMOA algorithm for different populations for function number 11



**Fig. 16** Convergence characteristic of the CMOA algorithm for different populations for function number 12

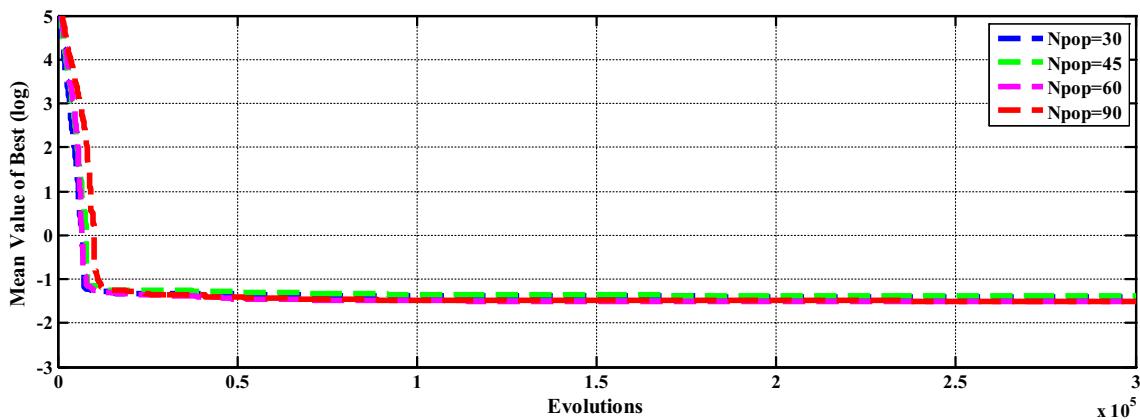


**Fig. 17** Convergence characteristic of the CMOA algorithm for different populations for function number 13

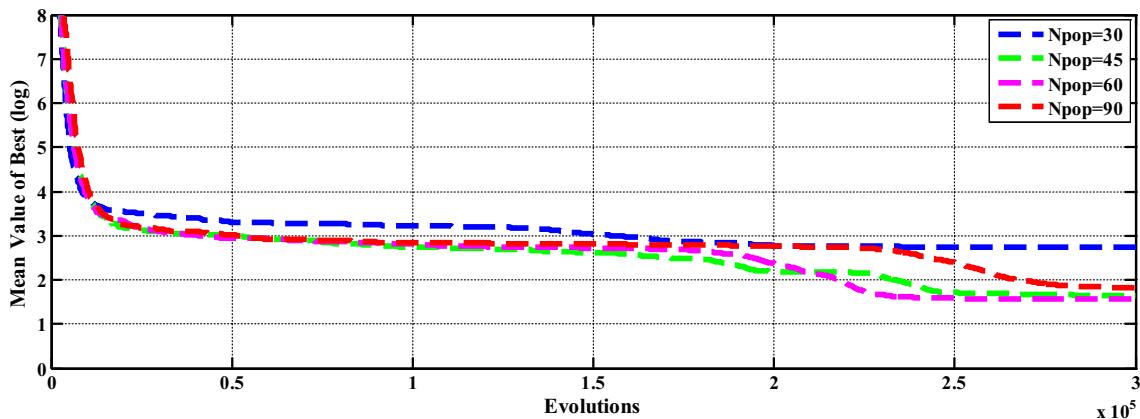
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33 and 34 (this can be seen from these figures that the algorithm undergoes major changes in such evolutions).

### 4.3 A comparative study of CMOA with new metaheuristic algorithms

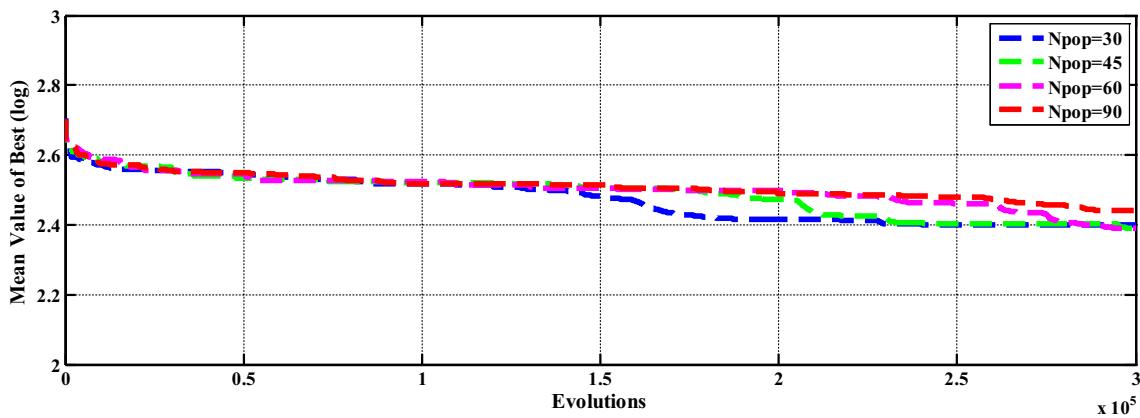
To show the performance of the proposed algorithm in a competitive study and equal and fair conditions, we have



**Fig. 18** Convergence characteristic of the CMOA algorithm for different populations for function number 14



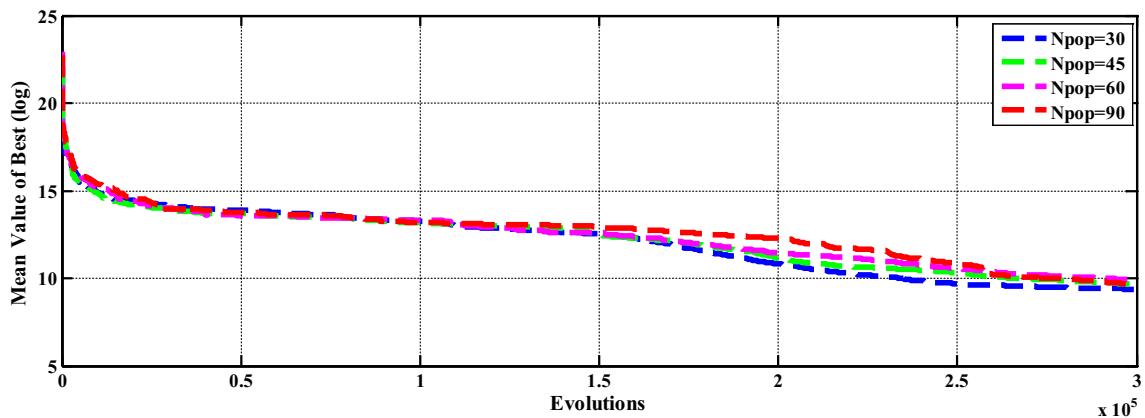
**Fig. 19** Convergence characteristic of the CMOA algorithm for different populations for function number 15



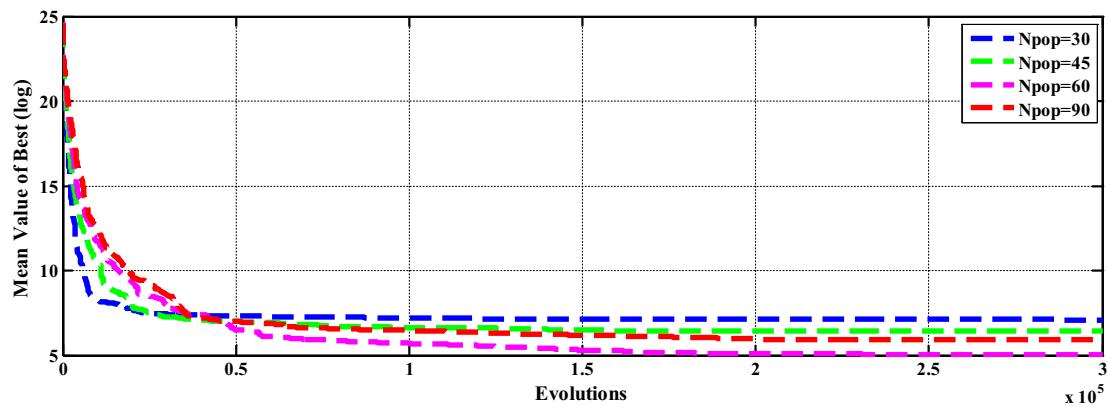
**Fig. 20** Convergence characteristic of the CMOA algorithm for different populations for function number 16

used several metaheuristic algorithms introduced in recent years. To this end, the test functions of 2014, including 30 modern standard optimization functions including unimodal, simple multimodal, hybrid, and composition test functions, which represent real-world problems and have been successfully implemented in many papers, have been

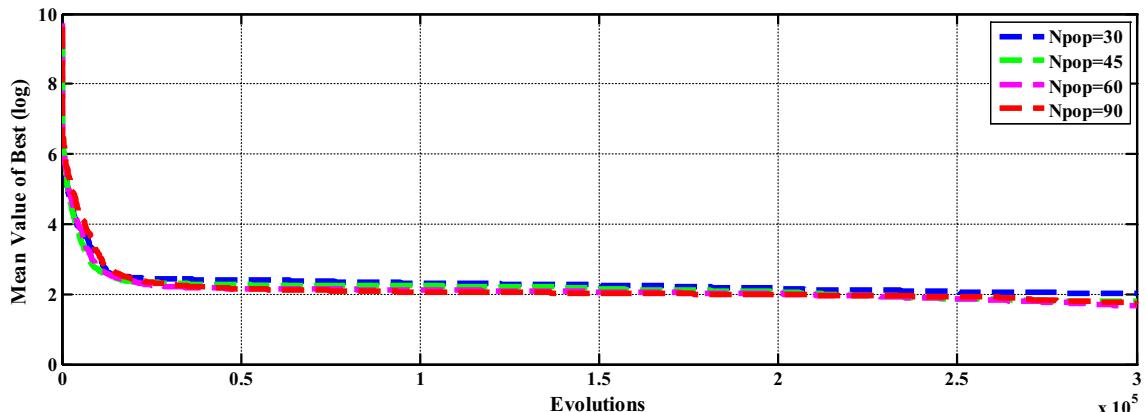
selected as criteria for measuring the power of the proposed CMOA algorithm. One system is utilized for all optimization algorithms the number of executions for each test function is 25, the number of dimensions for each function is 30, and the maximum number of fitness evaluations in all parts of the paper is 300,000 for any algorithm for the test



**Fig. 21** Convergence characteristic of the CMOA algorithm for different populations for function number 17



**Fig. 22** Convergence characteristic of the CMOA algorithm for different populations for function number 18



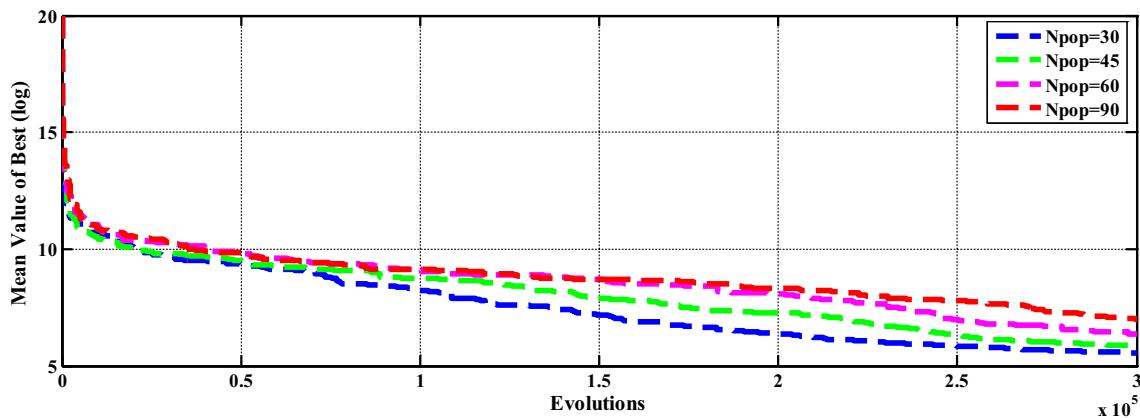
**Fig. 23** Convergence characteristic of the CMOA algorithm for different populations for function number 19

functions of CEC2014. The control parameters and population size of each algorithm are were selected based on the main reference of the algorithm given below.

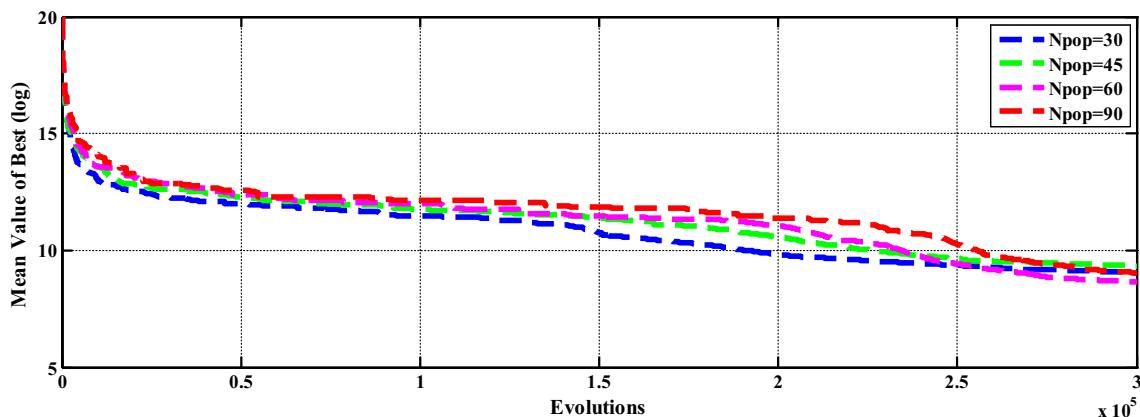
These algorithms include.

1. Artificial infectious disease optimization (AIDO) with  $N_{pop} = 60$  [11]

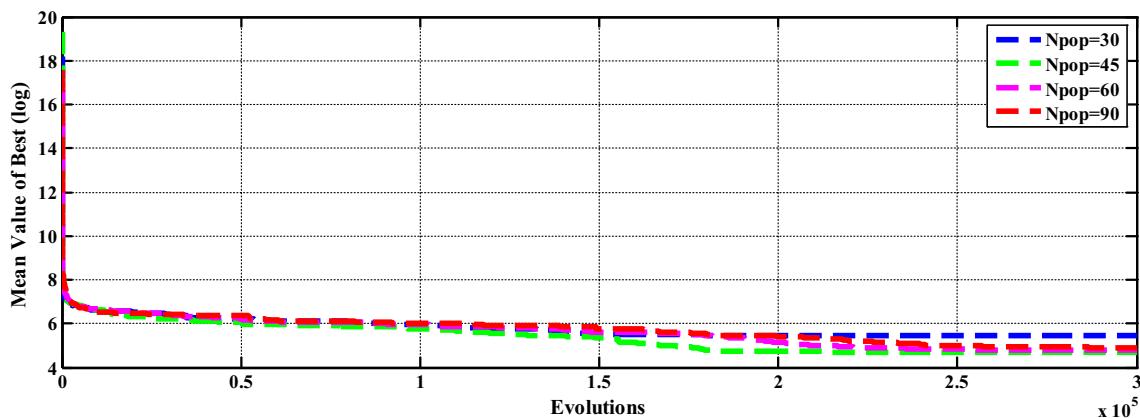
2. Invasive tumor growth optimization (ITGO) with  $N_{pop} = 50$  [14]
3. Rain-fall optimization algorithm (RFOA) with  $N_{pop} = 50$  [49]
4. Sine cosine algorithm (SCA) with  $N_{pop} = 60$  [21]
5. Cooperation search algorithm (CSA) with  $N_{pop} = 50$  [30]



**Fig. 24** Convergence characteristic of the CMOA algorithm for different populations for function number 20



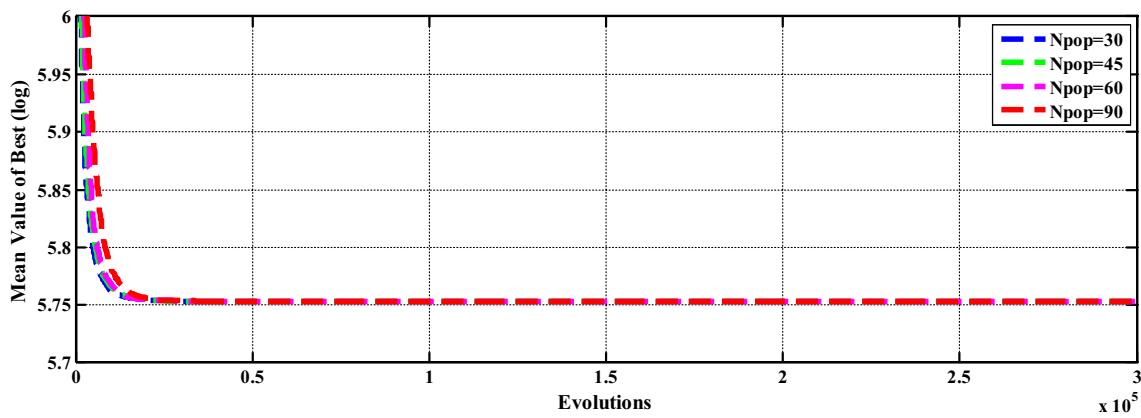
**Fig. 25** Convergence characteristic of the CMOA algorithm for different populations for function number 21



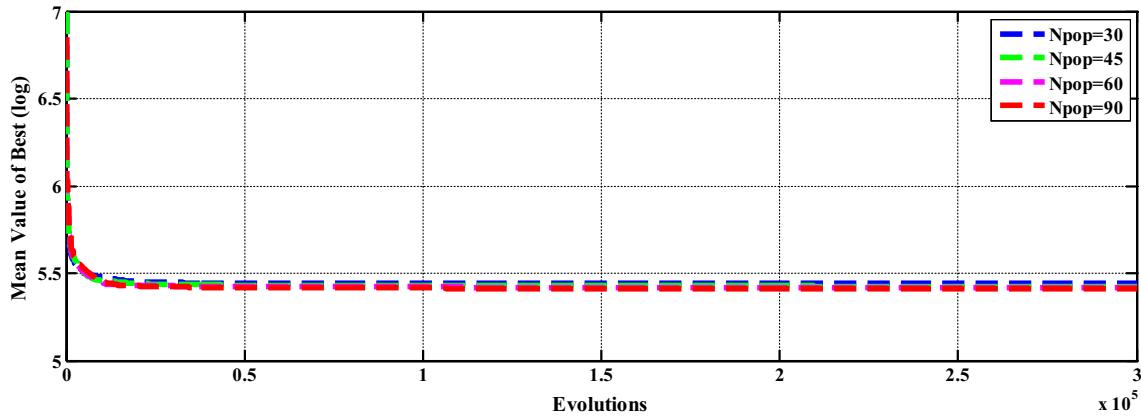
**Fig. 26** Convergence characteristic of the CMOA algorithm for different populations for function number 22

Tables 6, 7 and 8 illustrate the final results obtained by metaheuristic algorithms based on three measurement criteria, namely best and mean values and standard deviation (STD) of 25 times independent execution for each optimal function in each algorithm. What is clear from these tables is that the CMOA has been able to obtain the highest

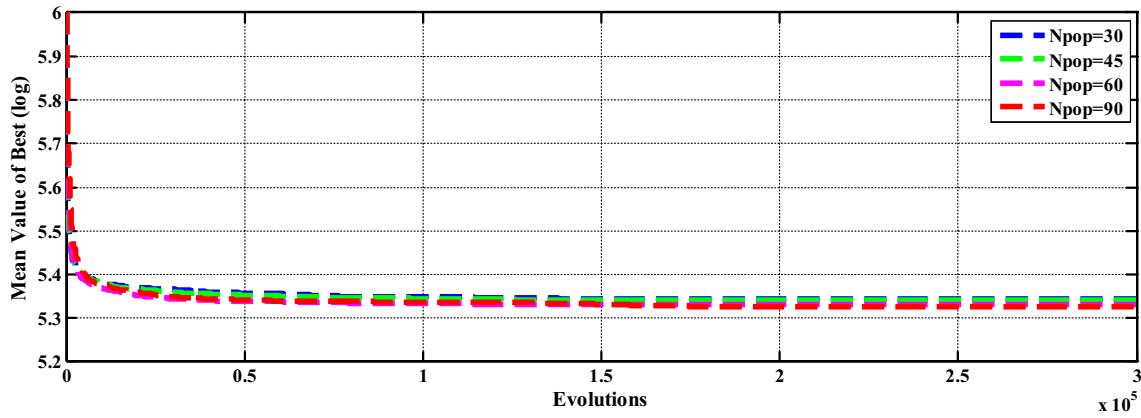
*Nb*, meaning that in most functions the best indices and results belong to the CMOA. The proposed CMOA algorithm succeeded to achieve a stunning performance compared to the five new metaheuristic algorithms AIDO, ITGO, RFOA, SCA, and CSA under equal conditions for optimizing CEC2014 functions. In general, the AIDO



**Fig. 27** Convergence characteristic of the CMOA algorithm for different populations for function number 23



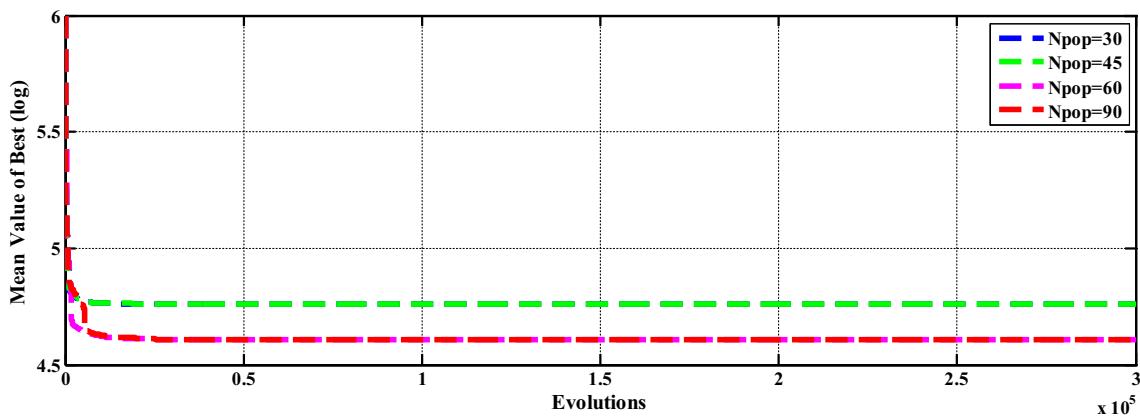
**Fig. 28** Convergence characteristic of the CMOA algorithm for different populations for function number 24



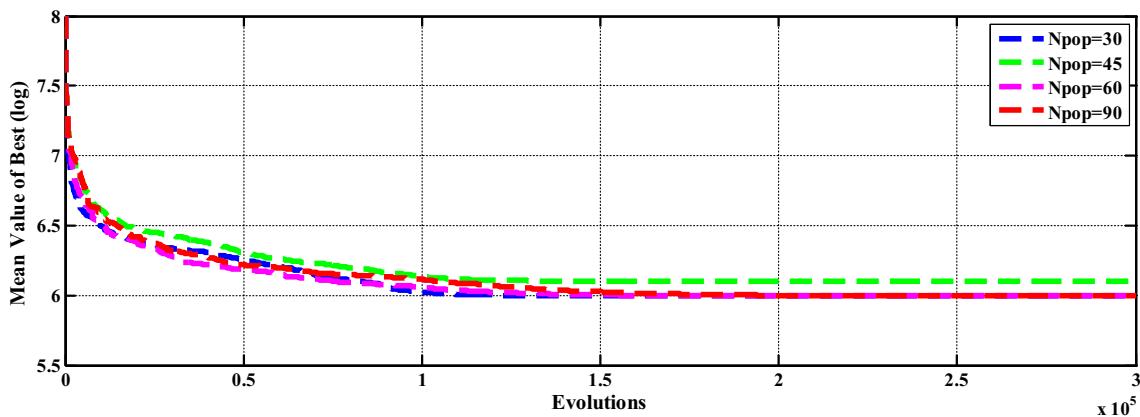
**Fig. 29** Convergence characteristic of the CMOA algorithm for different populations for function number 25

algorithm in two functions and the CSA algorithm in only one function were able to surpass the CMOA, while other algorithms have much less power in optimization than the proposed algorithm, which indicates the superiority and high value of CMOA as an emerging optimization algorithm.

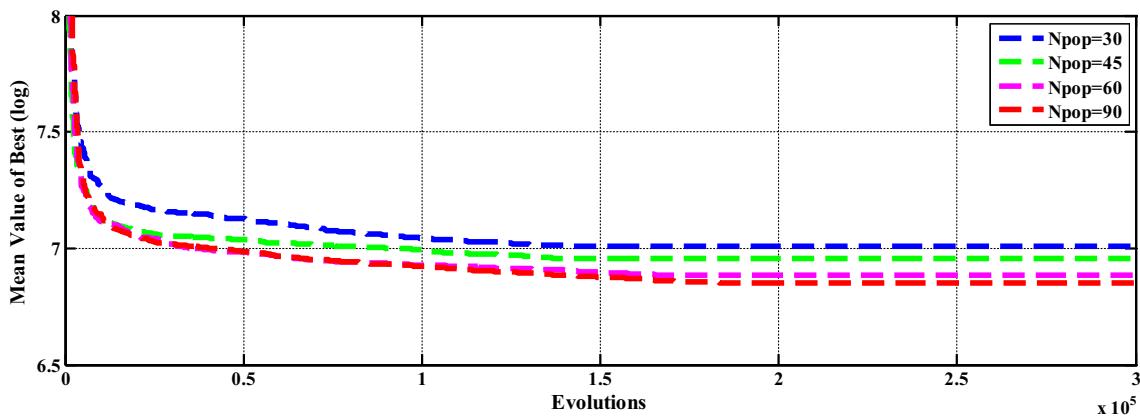
Convergence characteristics of the optimization algorithms used to optimize functions 1, 4, 7, 9, 11, 14, 19, 22, 27, and 30 are shown in Figs. 38, 39, 40, 41, 42, 43, 44, 45, 46 and 47. According to the results of these functions, the CMOA algorithm has enough power to escape the local optimal solution, and as can be seen from these diagrams, in the first iterations, the solution obtained by the CMOA



**Fig. 30** Convergence characteristic of the CMOA algorithm for different populations for function number 26



**Fig. 31** Convergence characteristic of the CMOA algorithm for different populations for function number 27

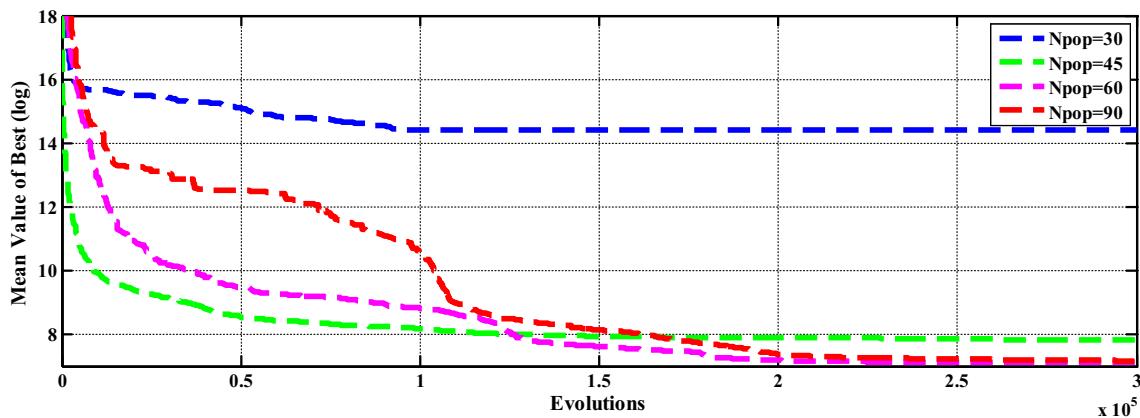


**Fig. 32** Convergence characteristic of the CMOA algorithm for different populations for function number 28

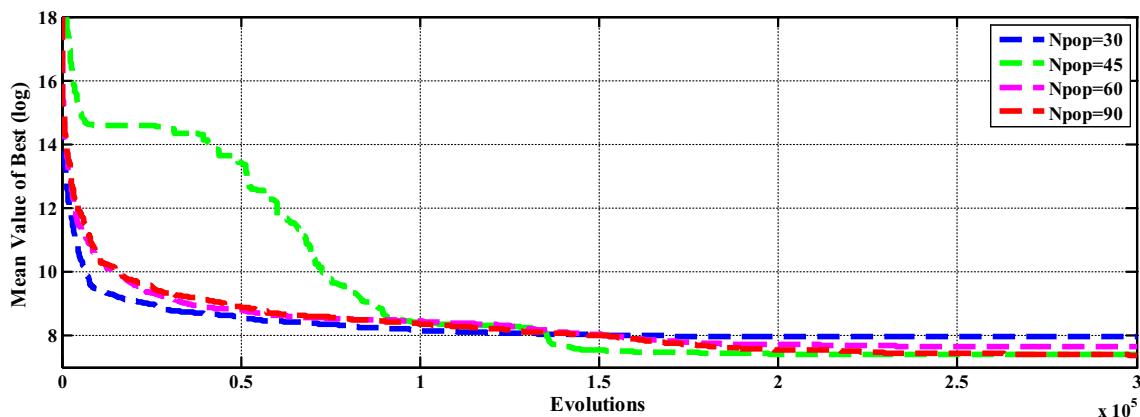
has a lower and better curve than those of other algorithms in most optimization functions, indicating a special advantage of CMOA.

The CMOA algorithm uses five motion and jump vectors, one of which is probably selected in each period. Thus, it has a strategy for all test functions and in all

iterations of the algorithm for better optimization, and also this algorithm is very suitable for adaptation.

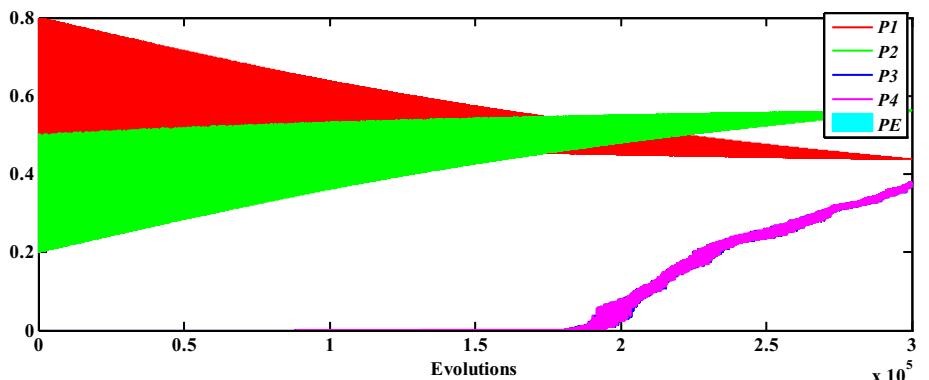


**Fig. 33** Convergence characteristic of the CMOA algorithm for different populations for function number 29



**Fig. 34** Convergence characteristic of the CMOA algorithm for different populations for function number 30

**Fig. 35** Characterization of changes in possible values with PE for test function 1 in one random run



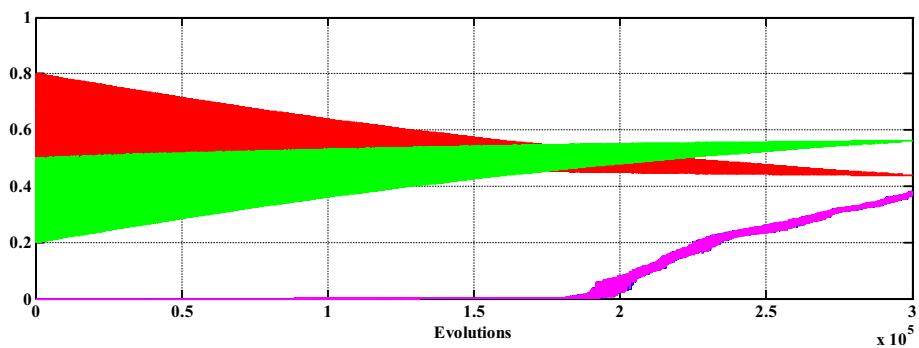
#### 4.4 A comparative study of CMOA with the results of the current studies

In this section, to better demonstrate the proposed algorithm, the optimal results obtained in the previous section are compared with those of new methods in terms of original algorithms and the improved algorithm to examine and analyze the efficiency of the proposed algorithm. A comparative study is provided.

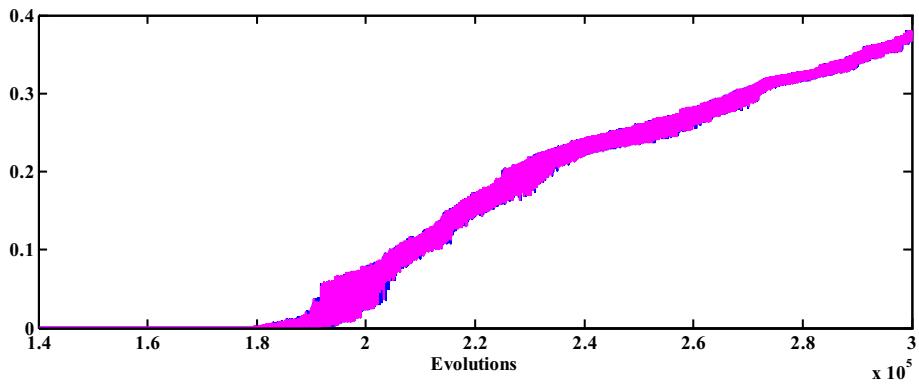
##### 4.4.1 A comparative study of CMOA with the results of the basic algorithms

In this subsection, we first compare and numerically analyze the results of the proposed algorithm with those of several original algorithms described in recent studies. These algorithms include CS (cuckoo search) [74, 112], SOS (symbiotic organisms search) [75, 113], GSA [40, 75], GWO (grey wolf optimizer) [76, 97], WOA (whale

**Fig. 36** Characterization of changes in possible values without PE for test function 1 in one random run



**Fig. 37** Characterization of changes in possible values P3 and P4 for test function 1 in one random run



optimization algorithm) [77, 114], MFO (moth-flame optimization) [77, 78], PSO [77, 79], Jaya [80], and CMA-ES (derandomized evolution strategy with covariance matrix adaptation) [81, 115]. The maximum number of fitness evaluations is 300,000 for any algorithm for the test functions of CEC2014. The results of all algorithms are given in Table 9. In this table, the sign – means the result of the algorithm is worse than the proposed algorithm, the sign + means the result of the algorithm is better than our proposed algorithm, and the sign = means the results of the two algorithms are similar. From this table, it is clear that the proposed CMOA algorithm has achieved the best result in most functions compared to other algorithms, and also CMOA is superior over other algorithms for most functions and obtained the rank 1 which is a satisfying outcome.

#### 4.4.2 A comparative study of CMOA with the results of the modern improved algorithms

Here we first compare the results of the proposed algorithm with the results of several modern modified algorithms in recent studies and analyze the results. These algorithms include RW-GWO (random walk GWO) [82], mTLBO (modified teaching–learning-based optimization) [83, 84], MG-SCA (memory guided sine cosine algorithm) [77], TOGPEAe (improved grey prediction evolution algorithm based on topological opposition-based learning) [85], m-SCA (modified sine cosine algorithm) [86], EEO

(enhanced equilibrium optimization) [116], and OB-L-EO (opposition-based Laplacian equilibrium optimizer) [121].

The results of all modern algorithms and the proposed algorithm are given in Table 10. The maximum number of fitness evaluations is 300,000 for any algorithm for the test functions of CEC2014. It is clear from this table that the proposed CMOA algorithm has achieved the best results in most functions compared to other modern and modified algorithms, and also that CMOA is superior to other modern and modified algorithms for most of the mathematical functions studied. And it has achieved rank 1 among all modern and modified algorithms, which is a suitable result compared to modern algorithms.

#### 4.5 A study of CMOA with modern algorithms to the engineering optimization

In this section, to better understand the performance of the proposed algorithm on solving real problems, three well-known engineering optimization problems including welded beam problem, three-bar truss optimal design, and pressure vessel problem have been studied and analyzed. The population size of the algorithm is assumed to be 60 for all three problems, and the number of iterations of the algorithm is assumed to be 400, 400, and 5000, respectively.

**Table 6** Mean statistical results of the optimization algorithms

Function		AIDO	SCA	RFOA	ITGO	CSA	CMOA
F1	Unimodal	499,649	525,045,166	337,806,247	144,657,018	163,730,949	<b>110,735</b>
		2	6	5	3	4	<b>1</b>
F2		3.27e – 01	70,563,554,127	52,033,889,368	20,659,641,208	146,845	<b>0.0326</b>
		2	6	5	4	3	<b>1</b>
F3		502	100,546	81,205	44,923	15,802	<b>102.02</b>
		2	6	5	4	3	<b>1</b>
F4	Simple multimodal	11.83	7943	5126	2470	241	<b>0.1851</b>
		2	6	5	4	3	<b>1</b>
F5		<b>20.86</b>	21.02	20.92	20.96	20.93	20.94
		<b>1</b>	6	3	5	4	2
F6		17.66	39.90	36.45	33.81	36.83	<b>14.64</b>
		2	6	4	3	5	<b>1</b>
F7		0.00984	566.9	345.57	248.04	0.2019	<b>0.00615</b>
		2	6	5	4	3	<b>1</b>
F8		79.02	272.8	247.22	227.13	183.24	<b>59.03</b>
		2	6	5	4	3	<b>1</b>
F9		79.60	325.6	284.85	256.55	188.08	<b>54.06</b>
		2	6	5	4	3	<b>1</b>
F10		2874	6660.7	5885.86	5420	6681	<b>2281</b>
		2	5	4	3	6	<b>1</b>
F11		4099	6940	5805	5348	6904	<b>3500</b>
		2	6	4	3	5	<b>1</b>
F12		2.096	3.008	2.049	2.13	2.40	<b>1.97</b>
		3	6	2	5	4	<b>1</b>
F13		0.3962	6.789	5.085	3.627	0.4985	<b>0.2232</b>
		2	6	5	4	3	<b>1</b>
F14		0.4231	214.978	165.69	73.58	0.2964	<b>0.219</b>
		3	6	5	4	2	<b>1</b>
F15		6.759	211,686	276,641	9476	16.983	<b>4.77</b>
		2	5	6	4	3	<b>1</b>
F16		11.65	13.40	13.073	12.93	12.97	<b>10.90</b>
		2	6	5	3	4	<b>1</b>
F17	Hybrid	199,512	15,275,617	1,911,491	284,650	4,973,547	<b>20,043</b>
		2	6	4	3	5	<b>1</b>
F18		17,633	586,507,558	64,092,460	4626	34,690,739	<b>155</b>
		3	6	5	2	4	<b>1</b>
F19		18.85	279.18	150.09	106.28	13.13	<b>5.31</b>
		3	6	5	4	2	<b>1</b>
F20		<b>327.51</b>	214,841	44,473	10,519	3580	581
		<b>1</b>	6	5	4	3	2
F21		12,270	817,230	214,054	19,602	997,041	<b>5740</b>
		2	5	4	3	6	<b>1</b>
F22		449	1099	961.55	699.92	383	<b>117</b>
		3	6	5	4	2	<b>1</b>

**Table 6** (continued)

Function		AIDO	SCA	RFOA	ITGO	CSA	CMOA
F23	Composition	<b>315.24</b>	733.58	464.47	406.38	<b>315.24</b>	<b>315.24</b>
		<b>1</b>	4	3	2	<b>1</b>	<b>1</b>
F24		243.81	314.57	299.81	273.98	<b>221.50</b>	224
		3	6	5	4	<b>1</b>	2
F25		210.44	253.66	240.49	225.20	233.2	<b>206</b>
		2	6	5	3	4	<b>1</b>
F26		100.51	125.5	105.82	103.92	100.37	<b>100.34</b>
		3	6	5	4	2	<b>1</b>
F27		650	1241.08	1225.5	942.26	1083.94	<b>401.06</b>
		2	6	5	3	4	<b>1</b>
F28		1840.9	3012.8	3142	2364	1622.9	<b>975</b>
		3	5	6	4	2	<b>1</b>
F29		12,736,831	68,713,664	34,184,381	23,459,365	110,797	<b>1243</b>
		3	6	5	4	2	<b>1</b>
F30		7767	633,463	363,515	99,071	60,343	<b>2124</b>
		2	6	5	4	3	<b>1</b>
Nb/ Nw/Mean		3/0/2.2	0/26/5.8	0/2/5.0	0/0/3.667	2/2/3.0	<b>27/0/1.1</b>

Results of Coronavirus Metamorphosis Optimization Algorithm (CMOA)

#### 4.5.1 Optimal design of a welded beam

The optimal design of a welded beam problem minimizes the cost of a welded beam.  $x_1$  or  $h$  (thickness of the weld),  $x_2$  or  $l$  (length of the attached part of the bar),  $x_3$  or  $t$  (height of the bar), and  $x_4$  or  $b$  (the thickness of the bar) are the decision variables of the problem (refer to Fig. 48).

The objective function of the problem is a nonlinear function with two linear and five nonlinear inequality constraints, namely, shear stress ( $\tau$ ), bending stress in the beam ( $\sigma$ ), buckling load on the bar ( $P_b$ ), end deflection of the beam ( $\delta$ ) [87]:

Minimize:

$$f(X) = 1.10471x_2x_1^2 + 0.04811x_3x_4(14 + x_2) \quad (12)$$

Subject to:

$$g_1(X) = \tau(x) - \tau_{\max} \leq 0, \quad (13)$$

$$g_2(X) = \sigma(x) - \sigma_{\max} \leq 0, \quad (14)$$

$$g_3(X) = x_1 - x_4 \leq 0, \quad (15)$$

$$g_4(X) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0. \quad (16)$$

$$g_5(X) = 0.125 - x_1 \leq 0, \quad (17)$$

$$g_6(X) = \delta(x) - \delta_{\max} \leq 0, \quad (18)$$

$$g_7(X) = P - P_c(x) \leq 0, \quad (19)$$

Subject to:

In this equation:

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2} \quad (20)$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{MR}{J}, \quad (21-22)$$

$$M = P \left( L + \frac{x_2}{2} \right), \quad R = \sqrt{\frac{x_2^2}{4} + \left( \frac{x_1 + x_3}{2} \right)^2}, \quad \delta(x) = \frac{4PL^3}{Ex_3^3x_4} \quad (23-25)$$

$$J = 2 \left[ \sqrt{2}x_1x_2 \left\{ \frac{x_2^2}{12} + \left( \frac{x_1 + x_3}{2} \right)^2 \right\} \right], \quad \sigma(x) = \frac{6PL}{x_4x_3^2}, \quad (26-27)$$

$$P_c(x) = \frac{4.013E \sqrt{\frac{x_4^6x_3^2}{36}}}{L^2} \left( 1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right), \quad (28)$$

$$P = 6000 \text{ lb}; \quad L = 14 \text{ in}; \quad E = 30e6 \text{ psi} \quad G = 12e6 \text{ psi}, \\ \tau_{\max} = 13,000 \text{ psi}, \quad \sigma_{\max} = 30,000 \text{ psi}, \\ \delta_{\max} = 0.25 \text{ in}, \quad 0.1 \leq x_1 \leq 2, \quad 0.1 \leq x_2 \leq 10, \\ 0.1 \leq x_3 \leq 10, \quad 0.1 \leq x_4 \leq 2.$$

Simulation results obtained by the CMOA algorithm in the optimal design of the welded beam problem are tabulated in Table 11. The table also compares the CMOA results with standard algorithms including Sinusoidal FFA (firefly algorithm) [88], EPSO (extraordinary PSO) [89], SFO (sailfish optimizer) [27], EO (equilibrium optimizer) [90], FSA (derivative-free filter simulated annealing method) [91], UPSO (unified PSO) [92], SBM (a socio-behavioral simulation model) [93], TEO (thermal exchange optimization) [42], WCA (water cycle algorithm) [94], BA

**Table 7** Best statistical results of the optimization algorithms

Function		AIDO	SCA	RFOA	ITGO	CSA	CMOA
F1	Unimodal	59,959	402,650,398	142,350,379	64,653,556	100,179,515	<b>43,922</b>
		2	6	5	3	4	<b>1</b>
F2		5.68e – 3	605,838,411,818	39,030,060,235	10,137,059,094	76,770	<b>9.27e – 05</b>
		2	6	5	4	3	<b>1</b>
F3		13.7	824,148	50,011	31,947	11,413	<b>6.95</b>
		2	6	5	4	3	<b>1</b>
F4	Simple multimodal	7.014e – 02	5773	3943	1144	130	<b>0.00741</b>
		2	6	5	4	3	<b>1</b>
F5		<b>20.73</b>	20.93	20.88	20.89	20.87	20.84
		<b>1</b>	6	4	5	3	2
F6		12.59	37.61	32.0	29.90	34.87	<b>9.77</b>
		2	6	4	3	5	<b>1</b>
F7		1.24e – 05	383.5	175.44	199.68	0.0927	<b>0.0</b>
		2	6	4	5	3	<b>1</b>
F8		53.77	240.58	217.2	168.35	167.35	<b>47.75</b>
		2	6	5	4	3	<b>1</b>
F9		57.75	272	258	195	170	<b>46.76</b>
		2	6	5	4	3	<b>1</b>
F10		17,305	5915	4474	4868	6529	<b>2013</b>
		6	4	2	3	5	<b>1</b>
F11		27,835	64,175	4659	4990	6365	<b>2954</b>
		5	6	2	3	4	<b>1</b>
F12		0.9655	2.0765	1.64	0.814	2.24	<b>0.4481</b>
		3	5	4	2	6	<b>1</b>
F13		0.2885	5.8355	4.3053	3.313	0.436	<b>0.1167</b>
		2	6	5	4	3	<b>1</b>
F14		0.2303	1665	111.08	48.98	0.260	<b>0.124</b>
		2	6	5	4	3	<b>1</b>
F15		4.24	15,211	35,422	4846	15.20	<b>3.50</b>
		2	5	6	4	3	<b>1</b>
F16		10.81	12.87	12.89	11.93	12.65	<b>10.45</b>
		2	5	6	3	4	<b>1</b>
F17	Hybrid	30,004	4,659,794	452,660	45,943	2,793,309	<b>5272</b>
		2	6	4	3	5	<b>1</b>
F18		4508	7,160,712	65,361	292.25	23,736,909	<b>13.50</b>
		3	5	4	2	6	<b>1</b>
F19		7.37	192.92	96.08	46.25	12.52	<b>3.64</b>
		2	6	5	4	3	<b>1</b>
F20		388	9692	5412	1194	1968	<b>238</b>
		2	6	5	3	4	<b>1</b>
F21		5623	120,835	20,907	10,315	325,403	<b>1074</b>
		2	5	4	3	6	<b>1</b>
F22		288	909	773	215	197	<b>25.99</b>
		4	6	5	3	2	<b>1</b>

**Table 7** (continued)

Function		AIDO	SCA	RFOA	ITGO	CSA	CMOA
F23	Composition	<b>315.24</b>	492.67	413.3	376.58	<b>315.24</b>	<b>315.24</b>
		<b>1</b>	4	3	2	<b>1</b>	<b>1</b>
F24		226	290	278	266	<b>214</b>	222
		3	6	5	4	<b>1</b>	2
F25		208.4	234.6	225.1	213.75	228.06	<b>205.08</b>
		2	6	4	3	5	<b>1</b>
F26		100.45	105.75	105.28	103.05	100.40	<b>100.30</b>
		3	6	5	4	2	<b>1</b>
F27		401.4	848.945	625.62	533.6	1041.95	<b>400.78</b>
		2	5	4	3	6	<b>1</b>
F28		1247	2503	1679	1317	1015	<b>937</b>
		3	6	5	4	2	<b>1</b>
F29		1525	10,547.775	496.767	26,261	35,236	<b>1111</b>
		2	6	5	3	4	<b>1</b>
F30		1400	45,359	8516	19,136	24,971	<b>935</b>
		2	6	3	4	5	<b>1</b>
Nb/ Nw/Mean		2/1/2.4	0/24/5.667	0/2/4.33	0/0/3.467	2/3/3.667	28/0/1.067

Results of Coronavirus Metamorphosis Optimization Algorithm (CMOA)

(bat algorithm) [95], PFA (pathfinder algorithm) [26], MRFO (manta ray foraging optimization) [96], GWO (grey wolf optimizer) [97] and HGSO (henry gas solubility optimization) [43].

According to Table 11, the value of the fitness function obtained by CMOA is 1.724852. Table 12 provides the best solutions achieved for the problem under study. Figure 49 depicts the performance of CMOA.

#### 4.5.2 Optimal design of a three-bar truss

This problem minimizes the volume of a statistically loaded three-bar truss subject to bar's stress ( $\sigma$ ) (refer to Fig. 50). The nonlinear objective function of the problem under study is subject to three constraints and two design parameters [87], i.e., the cross-sectional areas,  $A_1(x_1)$  and  $A_2(x_2)$ , is defined as follows:

Minimize:

$$f(X) = H * \left(2\sqrt{2}x_1 + x_2\right). \quad (29)$$

Subject to:

$$g_1(X) = P * \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} - \sigma \leq 0, \quad (30)$$

$$g_2(X) = P * \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} - \sigma \leq 0, \quad (31)$$

$$g_3(X) = P * \frac{1}{\sqrt{2}x_2 + x_1} - \sigma \leq 0, \quad (32)$$

$$0 \leq x_i \leq 1, i = 1, 2$$

$$H = 100 \text{ cm}, P = 2 \text{ kN/cm}^2, \sigma = 2 \text{ kN/cm}^2.$$

Table 13 gives the simulation results of the CMOA algorithm for the three-bar truss optimal design problem. A comparison is also made between the performance of the CMOA with some other standard algorithms.

As is seen in Table 13, the CMOA provides more suitable results than its counterparts. The standard deviation (Std.) of the solution is 6.25E-05, showing that the CMOA gives the most reliable and robust solution for the problem considered here. Additionally, the best solutions achieved for the problem are provided in Table 14. Figure 51 depicts the performance of CMOA.

#### 4.5.3 Optimal design of a pressure vessel

The aim of the optimal design of a pressure vessel problem is to find the minimum cost of a pressure vessel subject to material, shaping, and welding costs. The problem has two discrete ( $x_1$  and  $x_2$ ) and two continuous ( $x_3$  and  $x_4$ ) decision variables (Fig. 52), where  $x_1$  or  $T_s$  is the thickness of the shell,  $x_2$  or  $T_h$  is the thickness of the head,  $x_3$  or  $R$  is the inner radius, and  $x_4$  or  $L$  represents the length of the cylindrical section of the vessel. The nonlinear objective function of the problem subject to linear and one nonlinear inequality constraints is mathematically expressed as follows [87]:

Minimize:

$$f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (33)$$

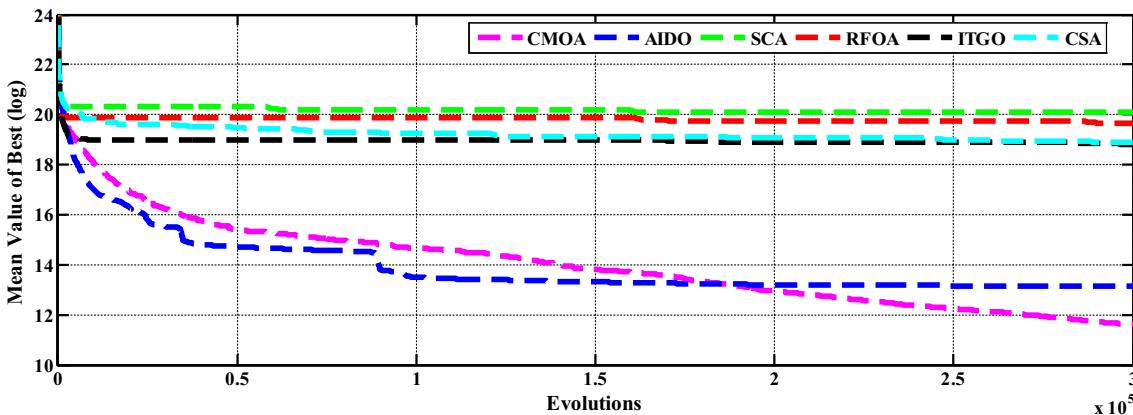
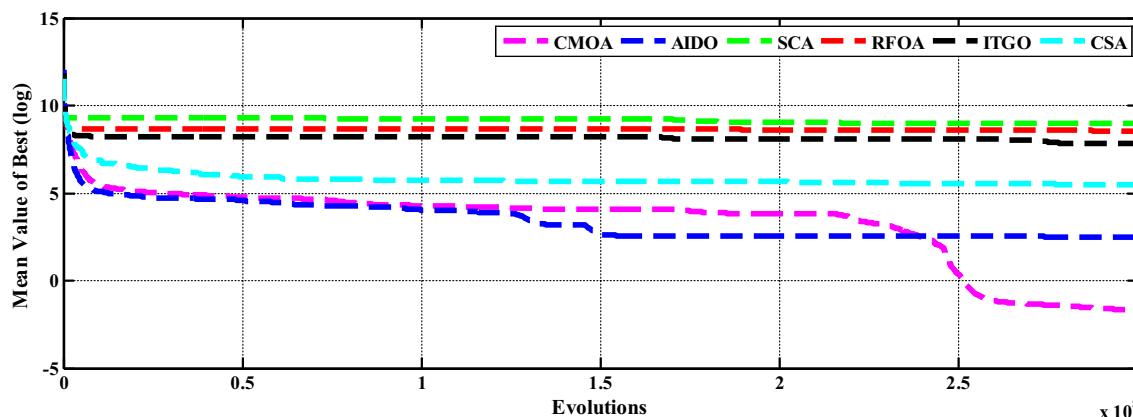
**Table 8** Standard deviation (STD) statistical results of the optimization algorithms

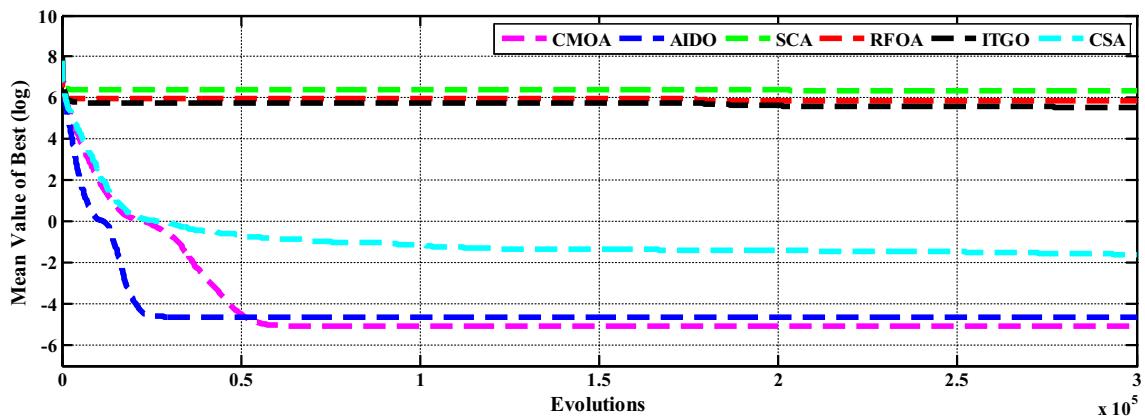
Function		AIDO	SCA	RFOA	ITGO	CSA	CMOA
F1	Unimodal	573,026	97,383,302	183,166,435	97,049,024	59,308,454	<b>60,015</b>
		2	4	6	5	3	<b>1</b>
F2		4.841e – 01	7,977,848,257	7,567,700,892	79,921,772,496	45,352	<b>0.0632</b>
		2	5	4	6	3	<b>1</b>
F3		725	18,679	25,890	13,698	4213	<b>103</b>
		2	5	6	4	3	<b>1</b>
F4	Simple multimodal	25.3	1815	1288	891	66	<b>0.24</b>
		2	6	5	4	3	<b>1</b>
F5		0.127	0.0733	0.0905	0.0872	0.0588	<b>0.056</b>
		6	3	5	4	2	<b>1</b>
F6		4.63	11.846	3.919	4.76	<b>1.338</b>	3.66
		3	6	4	5	<b>1</b>	2
F7		0.0138	135.35	94.626	62.023	0.14	<b>0.0103</b>
		2	6	5	4	3	<b>1</b>
F8		21.71	19.6	23.78	31.59	<b>10.62</b>	13.18
		4	3	5	6	<b>1</b>	2
F9		20.67	48.98	24.26	35.36	12.35	<b>6.50</b>
		3	6	4	5	2	<b>1</b>
F10		881	387.7	728	490	377	<b>295</b>
		6	3	5	4	2	<b>1</b>
F11		974	429	618	<b>373</b>	3677	553
		5	2	4	<b>1</b>	6	3
F12		0.975	1.57	0.97	0.92	1.94	<b>0.819</b>
		4	5	3	2	6	<b>1</b>
F13		0.594	1.088	0.751	0.2615	0.626	<b>0.0663</b>
		3	6	5	2	4	<b>1</b>
F14		0.2499	34.76	40.38	22.73	<b>0.026</b>	0.0511
		3	5	6	4	<b>1</b>	2
F15		2.285	299,552	216,539	3795	10.46	<b>1.56</b>
		2	6	5	4	3	<b>1</b>
F16		0.5944	0.9459	1.11	0.6543	0.82	<b>0.4402</b>
		2	5	6	3	4	<b>1</b>
F17	Hybrid	471,462	9,786,998	1,316,854	410,461	1,714,664	<b>13,500</b>
		3	6	4	2	5	<b>1</b>
F18		8673	704,313,457	93,059,286	10,449	10,704,904	<b>143</b>
		2	6	5	3	4	<b>1</b>
F19		24.36	85.05	68.43	49.90	2.655	<b>0.932</b>
		3	6	5	4	2	<b>1</b>
F20		435	250,229	39,345	8195	1543	<b>311</b>
		2	6	5	4	3	<b>1</b>
F21		7310	590,072	236,329	11,696	479,149	<b>4098</b>
		2	6	4	3	5	<b>1</b>
F22		177	217	181	294	154	<b>61.2</b>
		3	5	4	6	2	<b>1</b>

**Table 8** (continued)

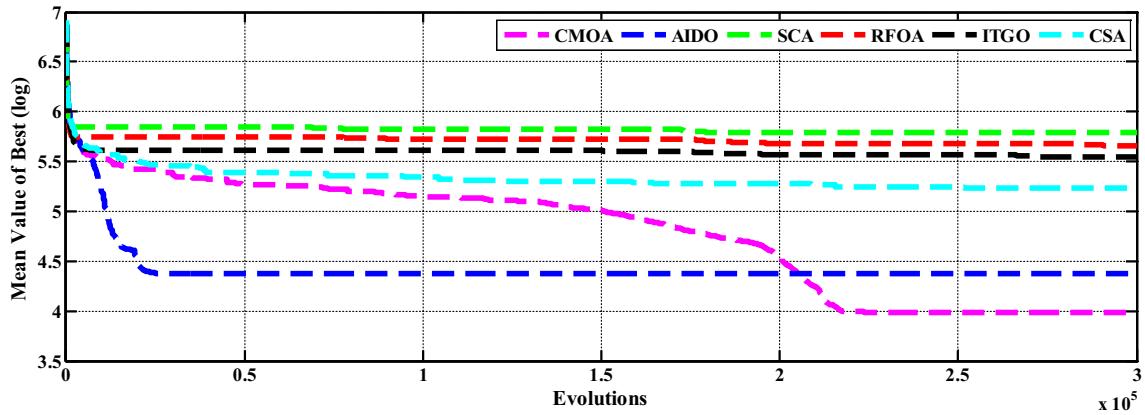
Function		AIDO	SCA	RFOA	ITGO	CSA	CMOA
F23	Composition	<b>0.0</b>	200.6	34.46	25.53	1.87e – 13	<b>0.0</b>
		<b>1</b>	5	4	3	2	<b>1</b>
F24		11.28	15.09	20.95	8.466	5.33	<b>1.44</b>
		4	5	6	3	2	<b>1</b>
F25		6.57	14.16	10.93	7.38	5.58	<b>1.39</b>
		3	6	5	4	2	<b>1</b>
F26		0.2024	45.98	0.6898	1.068	0.069	<b>0.05083</b>
		3	6	4	5	2	<b>1</b>
F27		196	211.74	308.32	407.70	41.35	<b>0.404</b>
		3	4	5	6	2	<b>1</b>
F28		293	400	808	763	673	<b>27.16</b>
		2	3	6	5	4	<b>1</b>
F29		10,074,130	66,013,620	28,041,483	28,703,030	51,298	<b>152</b>
		3	6	4	5	2	<b>1</b>
F30		10,360	550,541	4,520,165	116,696	38,202	<b>1360</b>
		2	6	5	4	3	<b>1</b>
Nb/ Nw/Mean		1/2/2.9	0/16/5.067	0/6/4.8	1/4/4	3/2/2.9	<b>26/0/1.167</b>

Results of Coronavirus Metamorphosis Optimization Algorithm (CMOA)

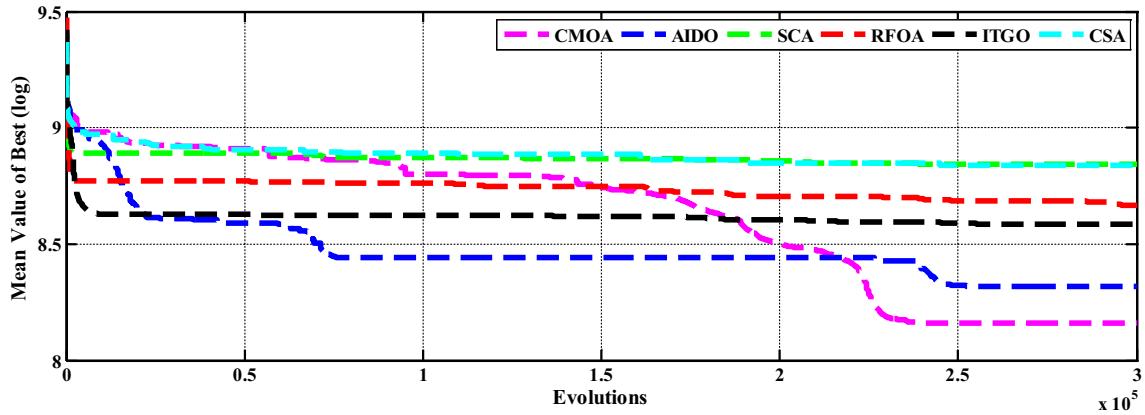
**Fig. 38** Convergence characteristic of algorithms for function number 1**Fig. 39** Convergence characteristic of algorithms for function number 4



**Fig. 40** Convergence characteristic of algorithms for function number 7



**Fig. 41** Convergence characteristic of algorithms for function number 9



**Fig. 42** Convergence characteristic of algorithms for function number 11

Subject to:

$$g_1(X) = -x_1 + 0.0193x_3 \leq 0, \quad (34)$$

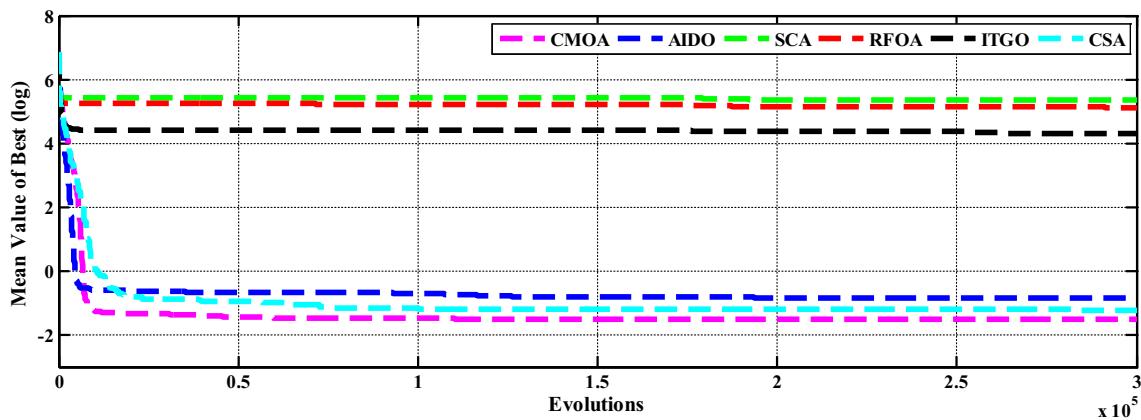
$$g_2(X) = -x_2 + 0.00954x_3 \leq 0, \quad (35)$$

$$g_3(X) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leq 0, \quad (36)$$

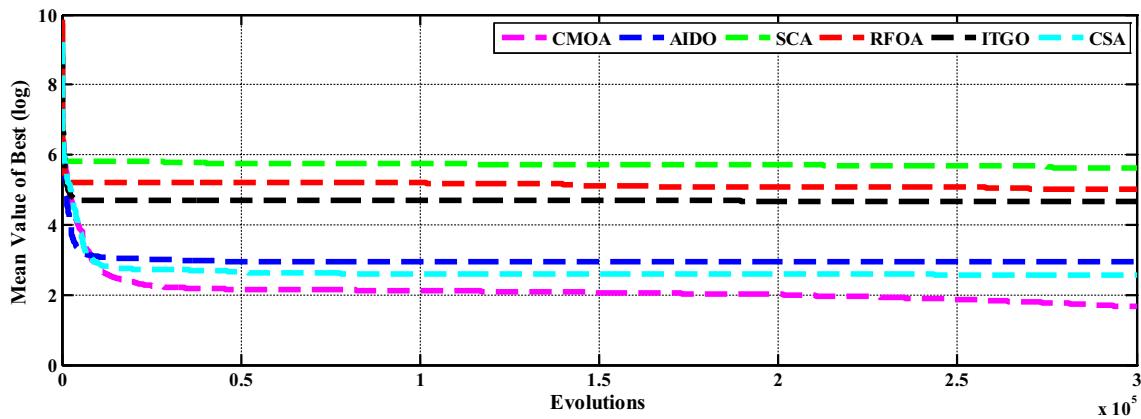
$$g_4(X) = x_4 - 240 \leq 0, \quad (37)$$

$$0 \leq x_i \leq 100, \quad i = 1, 2 \text{ and } 10 \leq x_i \leq 200, \quad i = 3, 4$$

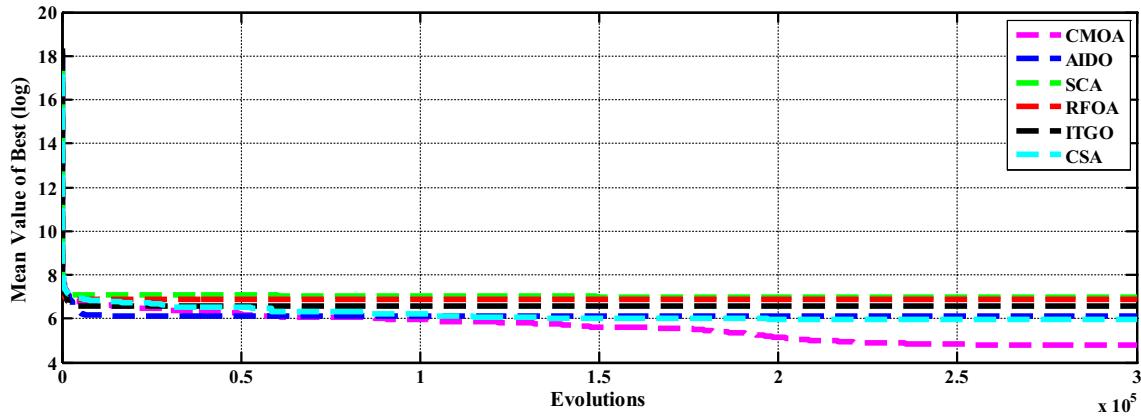
Simulation results achieved by applying the CMOA algorithm to the problem are provided in Table 15, where a



**Fig. 43** Convergence characteristic of algorithms for function number 14



**Fig. 44** Convergence characteristic of algorithms for function number 19



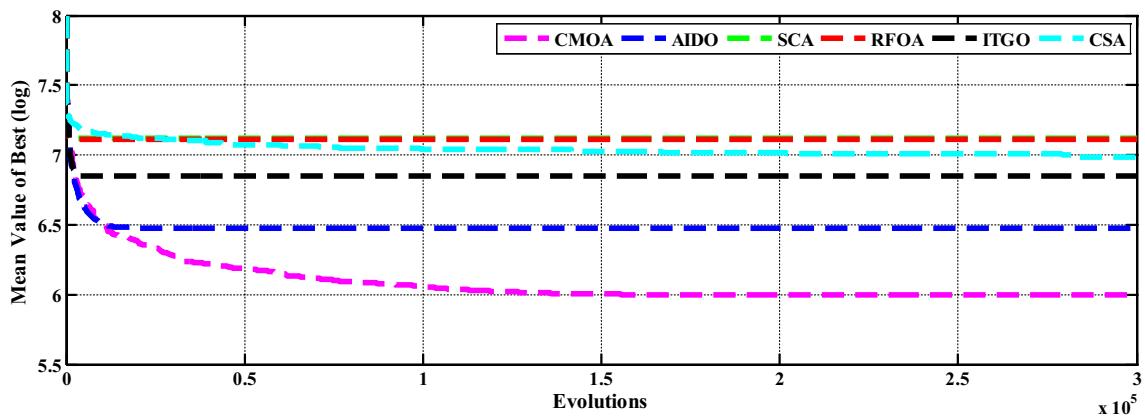
**Fig. 45** Convergence characteristic of algorithms for function number 22

comparison has also been made among the proposed algorithm and some other well-known algorithms.

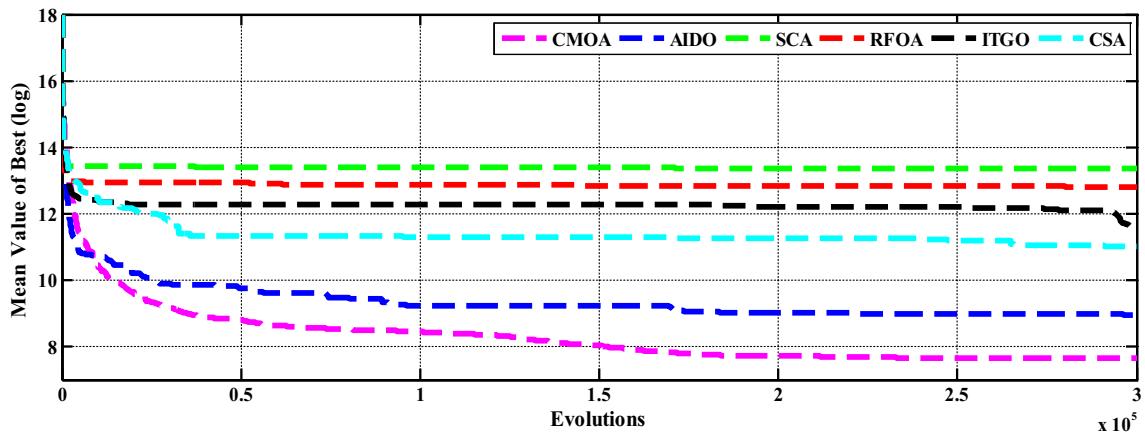
It is observed from Table 15 that the value of the fitness function obtained using the CMOA is 6059.71433. According to this Table 16, the best solutions to the pressure vessel optimal design problem belong to the CMOA. Figure 53 depicts the performance of CMOA.

#### 4.6 Discussion and future trends

This paper tests the application of CMOA on several popular problems to validate its capability. In each of the iterations of the CMOA, the best and random candidate solutions found with searching trends of various mutation and movement integrated strategies are used to generate the



**Fig. 46** Convergence characteristic of algorithms for function number 27



**Fig. 47** Convergence characteristic of algorithms for function number 30

population. The purpose of the strategies is to keep sufficient diversity during the search process to prevent the population being trapped in sub-optimal solutions. A comparison is also made between the performance of this algorithm with some other optimization tools including AIDO, ITGO, RFOA, SCA, CSA, CS, SOS, GWO, WOA, MFO, PSO, Jaya, CMA-ES, GSA, RW-GWO, mTLBO, MG-SCA, TOGPEAe, m-SCA, EEO and OB-L-EO. As it was observed from the results, the CMOA provides more suitable and optimized solutions than its competitors for the problems studied. The CMOA preserves the diversity of the population and prevents trapping in local optima. Moreover, the search behavior of CMOA is superior to its counterparts when applied to the problems under study and succeeds to achieve the best or the second-best solutions, validating its generality and robust structure. The CMOA is also applied to three engineering problems, showing its high potential in solving such practical problems and effectiveness in finding global optima. Several statistical indicators are also tested using the CMOA, which demonstrates its efficiency compared to the rest of the

methods. This is also highlighted that the CMOA is a stable and reliable method when employed for expert systems.

The CMOA as a meta-heuristic algorithm may not find the optimal solution (based on NFL theorem, [53]), and it is a shortcoming. Many real-world problems like handwritten word or digit recognition, human activity recognition, facial emotion recognition, script recognition, graphology applications, sleep deprivation detection, etc. can adopt the CMOA as a solver. Other classifiers such as neural networks or random forest can also be investigated to extend this discussion.

Also, future studies will hopefully focus on the implementation of CMOA on complicated real problems. The studies can also pay attention to the development of constrained and multi-objective CMOA, while they can even take a look at other combined solutions so that the quality and convergence speed of the CMOA can be enhanced.

**Table 9** Mean statistical results of the optimization algorithms

Function	CS [74, 112]	SOS [75, 113]		GWO [76, 97]		WOA [77, 114]		MFO [77, 78]		PSO [77, 79]		Jaya [80]		CMA-ES [81, 115]		GSA [40, 75]			
		mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
F1	3.50E + 07	6.95E + 07	6.74E + 07	3.16E + 07	7.59E + 07	8.19E + 07	8.47E + 07	<b>9.42E + 04</b>	2.25E + 07	<b>7.89E + 04</b>	2.66E + 07	1.11E + 05	6.00E + 06	6.00E + 04					
	2.49E + 07	2.92E + 07	4.75E + 07	1.38E + 07	9.77E + 07	8.27E + 09	−,9	−,10	+ ,1	−,3								2	
F2	1.95E + 07	3.61E + 09	2.17E + 09	3.10E + 06	1.36E + 10	2.81E + 04	7.55E + 09	2.55E + 10	7.85E + 03	1.18E + 09	3.85E + 09	1.66E + 03	7.85E + 03	3.26E − 02	1.66E + 03	6.32E − 02			
	5.49E + 07	7.76E + 08	1.94E + 09	2.12E + 06	8.42E + 09	5.30E + 02	−,3	−,8	−,10	−,2								1	
F3	3.10E + 04	2.38E + 04	3.02E + 04	3.52E + 04	8.99E + 04	<b>2.09E + 01</b>	8.10E + 04	1.45E + 04	7.32E + 04	1.24E + 04	5.66E + 03	4.03E + 03	1.03E + 02	1.02E + 02	4.03E + 03	1.03E + 02			
	1.36E + 04	1.35E + 04	8.01E + 03	2.18E + 04	4.98E + 04	<b>2.77E + 01</b>	+ ,1	−,9	−,3	−,8								2	
F4	2.03E + 02	4.10E + 02	2.17E + 02	1.95E + 02	1.14E + 03	9.01E + 03	5.69E + 02	2.52E + 03	3.13E + 02	1.29E + 02	5.36E + 02	2.85E + 01	2.85E + 01	1.85E − 01	2.85E + 01	2.40E − 01			
	6.69E + 01	6.58E + 01	5.54E + 01	4.74E + 01	1.13E + 03	1.85E + 02	−,7	−,9	−,5	−,5								1	
F5	<b>2.00E + 01</b>	2.06E + 01	2.09E + 01	2.04E + 01	2.19E + 02	2.04E + 01	2.19E + 02	2.09E + 01	<b>2.00E + 01</b>	4.71E − 02	<b>2.63E − 05</b>	3.70E − 04	5.60E − 02	2.09E + 01	2.09E + 01	2.09E + 01	2.09E + 01		
	<b>2.28E − 03</b>	6.32E − 02	6.98E − 02	1.64E − 01	1.75E − 01	6.01E + 03	−,5	−,10	+ ,1	−,7	−,9	−,9	−,5	+ ,1	+ ,1	+ ,1			
F6	3.23E + 01	2.33E + 01	<b>1.30E + 01</b>	3.60E + 01	2.40E + 01	6.78E + 03	3.48E + 01	4.09E + 01	2.17E + 01	1.77E + 00	2.13E + 00	1.59E + 00	3.66E + 00	1.46E + 01	1.46E + 01	1.46E + 01	1.46E + 01		
	3.27E + 00	1.55E + 00	<b>2.30E + 00</b>	4.05E + 00	3.33E + 00	2.46E + 00	−,5	−,10	−,7	−,7	−,9	−,9	−,3	2					
F7	1.79E + 00	3.33E + 01	2.16E + 01	9.99E − 01	1.17E + 02	1.61E + 00	2.59E + 01	2.31E + 02	0.00E + 00	5.83E + 00	2.83E + 01	0.00E + 01	0.00E + 01	6.15E − 03	6.15E − 03	6.15E − 03	6.15E − 03		
	2.19E + 00	6.92E + 00	1.78E + 01	7.38E − 02	6.91E + 01	2.67E + 01	−,2	−,4	−,7	−,7	−,10	−,10	−,3	2					
F8	1.71E + 02	8.16E + 01	7.95E + 01	1.93E + 02	1.43E + 02	1.10E + 03	2.29E + 02	2.83E + 02	1.40E + 02	1.34E + 01	2.21E + 01	9.91E + 00	9.91E + 00	1.32E + 01	1.32E + 01	1.32E + 01			
	3.46E + 01	6.84E + 00	1.90E + 01	4.43E + 01	3.81E + 01	1.24E + 01	−,5	−,10	−,8	−,8	−,9	−,9	−,4	1					
F9	2.80E + 02	1.69E + 02	9.86E + 01	2.26E + 02	2.23E + 02	2.30E + 06	2.64E + 02	3.28E + 02	1.60E + 02	1.85E + 01	7.65E + 01	1.14E + 01	1.14E + 01	<b>5.41E + 01</b>	<b>5.41E + 01</b>	<b>5.41E + 01</b>			
	5.16E + 01	1.75E + 01	2.50E + 01	5.27E + 01	6.06E + 01	8.06E + 07	−,5	−,10	−,7	−,7	−,9	−,9	−,3	1					
F10	2.66E + 03	1.38E + 03	2.22E + 03	4.04E + 03	3.47E + 03	<b>4.43E + 01</b>	5.59E + 03	2.61E + 02	4.35E + 02	1.06E + 02	3.14E + 02	3.14E + 02	3.14E + 02	2.95E + 02	2.95E + 02	2.95E + 02			
	5.34E + 02	1.96E + 02	4.04E + 02	6.57E + 02	8.85E + 02	<b>7.82E + 03</b>	+ ,1	−,10	−,4	−,7	−,7	−,7	−,7	3					
F11	4.13E + 03	4.48E + 03	2.83E + 03	4.82E + 03	4.15E + 03	6.50E + 05	6.91E + 03	<b>1.69E + 02</b>	3.16E + 02	<b>1.98E + 02</b>	4.12E + 03	4.12E + 03	3.50E + 03	3.50E + 03	5.53E + 02	5.53E + 02	5.53E + 02		
	5.35E + 02	4.10E + 02	8.26E + 02	9.06E + 02	6.90E + 02	6.35E + 02	−,6	−,10	−,9	+ ,1	−,4	−,4	−,4	3					

Table 9 (continued)

Function	CS [74, 112]	SOS [75, 113]	GWO [76, 97]	WOA [77, 114]	MFO [77, 78]	PSO [77, 79]	Jaya [80]	CMA-FS [81, 115]	GSA [40, 75]	CMOA
	mean std	mean std	mean std	mean std	mean std	mean std	mean std	mean std	mean std	mean std
	winner, rank	winner, rank	winner, rank	winner, rank	winner, rank	winner, rank	winner, rank	winner, rank	winner, rank	winner, rank
F12	5.11E – 01 2.56E – 01	8.29E – 01 1.44E – 01	1.48E + 00 1.22E + 00	1.63E + 00 4.22E – 01	4.33E – 01 2.64E – 01	3.73E + 02 2.76E + 02	2.44E + 00 2.63E – 01	3.03E – 01 2.18E + 00	<b>9.10E – 04</b> <b>1.28E – 03</b>	1.97E + 00 8.19E – 01
	+ ,4	+ ,5	+ ,6	+ ,7	+ ,3	- ,10	- ,9	+ ,2	+ ,1	8
F13	4.81E – 01 1.17E – 01	7.79E – 01 1.48E – 01	5.57E – 01 4.72E – 01	2.21E + 00 1.26E – 01	2.14E + 02 1.34E + 00	1.80E + 00 1.58E + 02	5.51E + 00 3.59E – 01	3.31E – 01 3.07E – 01	<b>2.23E – 01</b> <b>3.14E – 02</b>	<b>6.63E – 02</b>
	- ,3	- ,6	- ,5	- ,4	- ,8	- ,10	- ,7	- ,9	- ,2	1
F14	3.08E – 01 5.64E – 02	9.76E + 00 3.87E + 00	4.43E + 00 6.96E + 00	2.55E – 01 4.84E – 02	3.54E + 01 2.47E + 01	9.05E + 02 1.92E + 03	1.23E + 01 1.82E + 00	7.53E + 01 8.08E + 00	2.30E – 01 2.19E – 02	<b>2.19E – 01</b> <b>5.11E – 02</b>
	- ,4	- ,6	- ,5	- ,3	- ,8	- ,10	- ,7	- ,9	- ,2	1
F15	9.80E + 01 3.02E + 01	2.50E + 02 2.05E + 02	2.21E + 02 5.57E + 02	7.00E + 01 1.99E + 01	2.23E + 05 5.77E + 05	1.04E + 07 6.54E + 04	8.39E + 01 7.03E + 01	1.02E + 04 3.24E + 04	1.05E + 01 1.69E + 00	<b>4.77E + 00</b> <b>1.56E + 00</b>
	- ,5	- ,7	- ,6	- ,3	- ,9	- ,10	- ,4	- ,8	- ,2	1
F16	1.27E + 01 5.01E – 01	1.13E + 01 3.70E – 01	<b>1.09E + 01</b> <b>6.90E – 01</b>	1.25E + 01 4.48E – 01	1.27E + 01 5.33E – 01	2.49E + 07 2.21E + 09	1.30E + 01 1.70E – 01	1.38E + 01 5.31E – 01	1.37E + 01 1.73E – 01	<b>1.09E + 01</b> <b>4.40E – 01</b>
	- ,4	- ,2	= ,1	- ,3	- ,4	- ,8	- ,5	- ,7	- ,6	1
F17	1.48E + 06 1.21E + 06	5.68E + 06 3.70E + 06	1.90E + 06 2.19E + 06	3.56E + 06 2.52E + 06	3.39E + 06 4.07E + 06	1.26E + 04 3.30E + 02	4.69E + 06 1.36E + 06	<b>5.49E + 03</b> <b>3.62E + 03</b>	8.55E + 05 1.74E + 05	2.00E + 04 1.35E + 04
	- ,5	- ,10	- ,6	- ,8	- ,7	+ ,2	- ,9	+ ,1	- ,4	3
F18	7.67E + 03 6.70E + 03	4.95E + 05 2.26E + 05	7.19E + 06 1.84E + 07	1.50E + 04 5.01E + 04	5.19E + 06 3.61E + 07	<b>4.81E – 02</b> <b>1.43E + 00</b>	2.97E + 07 3.19E + 07	1.52E + 09 3.93E + 08	3.90E + 02 1.25E + 02	1.55E + 02 1.43E + 02
	- ,4	- ,6	- ,8	- ,5	- ,7	+ ,1	- ,9	- ,10	- ,3	2
F19	5.33E + 01 3.63E + 01	3.80E + 01 2.93E + 01	3.89E + 01 2.29E + 01	5.07E + 01 3.83E + 01	7.36E + 01 5.32E + 01	2.91E + 01 1.48E + 01	3.85E + 01 1.91E + 01	2.98E + 02 4.25E + 01	1.32E + 02 2.75E + 01	<b>5.31E + 00</b> <b>9.32E – 01</b>
	- ,7	- ,3	- ,5	- ,6	- ,8	- ,2	- ,4	- ,10	- ,9	1
F20	3.93E + 04 2.20E + 04	1.59E + 04 1.09E + 04	1.50E + 04 9.68E + 03	2.29E + 04 1.95E + 04	5.67E + 04 4.34E + 04	<b>1.34E + 01</b> <b>4.16E + 02</b>	1.16E + 04 3.70E + 03	4.61E + 03 3.88E + 03	8.23E + 04 1.35E + 04	5.81E + 02 3.11E + 02
	- ,8	- ,6	- ,5	- ,7	- ,9	+ ,1	- ,4	- ,3	- ,10	2
F21	3.54E + 05 3.48E + 05	7.86E + 05 6.12E + 05	1.30E + 06 2.49E + 06	1.17E + 06 9.72E + 05	7.83E + 05 1.18E + 06	<b>3.88E + 02</b> <b>2.69E – 01</b>	9.02E + 05 3.08E + 05	6.86E + 03 2.76E + 03	1.79E + 05 3.19E + 04	5.74E + 03 4.10E + 03
	- ,5	- ,7	- ,10	- ,9	- ,6	+ ,1	- ,8	- ,3	- ,4	2
F22	9.47E + 02 3.31E + 02	5.36E + 02 1.74E + 02	3.75E + 02 1.59E + 02	8.13E + 02 2.70E + 02	8.67E + 02 2.29E + 02	<b>7.35E – 01</b> <b>1.19E + 01</b>	6.45E + 02 1.38E + 02	1.61E + 03 2.92E + 02	9.39E + 02 1.80E + 02	1.17E + 02 6.12E + 01
	- ,9	- ,4	- ,3	- ,6	- ,7	+ ,1	- ,5	- ,10	- ,8	2

**Table 9** (continued)

Function	CS [74, 112]	SOS [75, 113]	GWO [76, 97]	WOA [77, 114]	MFO [77, 78]	PSO [77, 79]	Jaya [80]	CMA-FS [81, 115]	GSA [40, 75]	CMOA mean std winner, rank
	mean std winner, rank	mean std winner, rank	mean std winner, rank	mean std winner, rank	mean std winner, rank	mean std winner, rank	mean std winner, rank	mean std winner, rank	mean std winner, rank	mean std winner, rank
F23	3.29E + 02 7.51E + 00	3.32E + 02 4.05E + 00	3.35E + 02 1.16E + 01	3.35E + 02 9.16E + 00	3.71E + 02 3.98E + 01	2.94E + 03 3.05E – 01	3.57E + 02 6.69E + 00	5.79E + 02 4.94E + 01	2.00E + 02 3.35E – 09	3.15E + 02 0.00E + 00
–,4	–,5	–,6	–,6	–,6	–,8	+ ,2	–,7	–,9	+ ,1	3
F24	2.78E + 02 3.11E + 01	2.66E + 02 4.05E + 00	<b>2.00E + 02</b> <b>8.20E – 04</b>	2.06E + 02 5.12E + 00	2.76E + 02 2.73E + 01	1.05E + 06 4.44E + 07	2.61E + 02 4.74E + 00	2.12E + 02 7.49E + 00	<b>2.00E + 02</b> <b>5.04E – 03</b>	2.24E + 02 1.44E + 00
–,8	–,6	–,1	+ ,2	+ ,2	–,7	–,9	–,5	+ ,3	+ ,1	4
F25	2.23E + 02 9.39E + 00	2.16E + 02 2.60E + 00	2.10E + 02 5.11E + 00	2.16E + 02 1.60E + 01	2.14E + 02 7.65E + 00	<b>1.64E + 01</b> <b>1.52E + 04</b>	2.23E + 02 5.18E + 00	2.12E + 02 2.97E + 00	2.00E + 02 6.29E – 11	2.06E + 02 1.39E + 00
–,8	–,7	–,4	–,7	–,7	–,6	+ ,1	–,8	–,5	+ ,2	3
F26	<b>1.00E + 02</b> <b>1.63E – 01</b>	1.01E + 02 1.22E – 01	<b>1.00E + 02</b> 4.98E + 01	<b>1.10E – 01</b>	1.03E + 02 1.50E + 00	2.83E + 05 1.99E + 02	1.01E + 02 1.70e – 01	1.25E + 02 5.51E + 01	1.76E + 02 3.32E + 01	<b>1.00E + 02</b> <b>5.08E – 02</b>
=,1	–,2	–,5	=,1	–,3	–,3	–,7	–,2	–,4	–,6	1
F27	4.27E + 02 1.96E + 01	5.03E + 02 1.48E + 02	6.53E + 02 1.30E + 02	1.08E + 03 3.71E + 02	9.21E + 02 2.23E + 02	<b>2.28E + 01</b> <b>2.15E + 00</b>	1.08E + 03 1.96E + 02	1.07E + 03 2.30E + 02	8.45E + 02 5.65E + 02	4.01E + 02 4.04E – 01
–,3	–,4	–,5	–,9	–,9	–,7	+ ,1	–,9	–,8	–,6	2
F28	3.49E + 03 5.48E + 02	1.33E + 03 1.36E + 02	1.11E + 03 2.23E + 02	2.38E + 03 4.88E + 02	1.12E + 03 1.57E + 02	<b>2.59E + 00</b> <b>5.98E + 01</b>	1.21E + 03 1.70E + 02	2.79E + 03 5.92E + 02	7.74E + 02 3.86E + 02	9.75E + 02 2.72E + 01
–,10	–,7	–,4	–,8	–,8	–,5	+ ,1	–,6	–,9	+ ,2	3
F29	5.44E + 05 2.61E + 06	1.11E + 05 1.14E + 05	9.12E + 05 1.84E + 06	4.85E + 06 4.82E + 06	3.06E + 06 3.62E + 06	<b>2.71E + 02</b> 5.04E + 02	1.57E + 06 3.06E + 06	3.52E + 04 5.34E + 03	<b>2.00E + 02</b> <b>3.57E – 03</b>	1.24E + 03 1.52E + 02
–,6	–,5	–,7	–,10	–,9	+ ,2	–,8	–,8	–,4	+ ,1	3
F30	2.49E + 04 2.26E + 04	3.73E + 04 2.12E + 04	5.66E + 04 5.95E + 04	8.38E + 04 6.11E + 04	5.89E + 04 5.40E + 04	1.18E + 07 8.87E + 04	1.56E + 04 6.41E + 03	6.48E + 05 1.31E + 05	<b>2.12E + 03</b> <b>1.36E + 03</b>	1.24E + 03 1.36E + 04
Nb/ Nw/ Mean	3/2/5/2000	0/1/5/4000	3/1/4/6000	1/2/5/7333	0/1/6/7667	9/14/5/4000	0/4/6/8000	4/6/6/3000	6/1/3/4667	<b>11/0/66</b>
+ / – / =	Final rank	2/27/1	2/28/0	5/23/2	3/26/1	2/28/0	12/18/0	0/29/1	6/24/0	–
										1

Results of Coronavirus Metamorphosis Optimization Algorithm (CMOA)

**Table 10** Mean statistical results of the optimization algorithms

Function	RW-GWO [82]	mTLBO [83, 84]	MG-SCA [77] Mean	TOGPEAe [85]	m-SCA [86] Mean	EEO [116]	OB-L-EO [121]	CMOA Mean	
	Mean	Mean	Std	Mean	Std	Mean	Mean	Std	
	Std	Std	Winner, rank	Std	Winner, rank	Std	Std	Winner, rank	
	Winner, rank	Winner, rank		Winner, rank	Winner, rank		Winner, rank		
F1	Unimodal	8.02E + 06 3.31E + 06 − ,3	6.46E + 07 4.03E + 07 − ,7	2.92E + 07 2.07E + 07 − ,6	6.54E + 06 3.49E + 06 − ,2	2.258E + 07 6.353E + 06 − ,5	7.76E + 07 5.63E + 07 − ,8	8.15E + 06 3.60E + 06 − ,4	<b>1.11E + 05</b> <b>6.00E + 04</b> <b>1</b>
F2		2.23E + 05 5.51E + 05 − ,3	1.18E + 09 1.52E + 09 − ,7	2.26E + 09 1.69E + 09 − ,8	1.79E + 07 2.26E + 07 − ,5	8.109E + 07 5.590E + 07 − ,6	1.13E + 06 1.07E + 06 − ,4	2.65E + 04 1.26E + 04 − ,2	<b>3.26E − 02</b> <b>6.32E − 02</b> <b>1</b>
F3		3.16E + 02 4.34E + 02 − ,3	<b>4.68E + 00</b> <b>1.73E + 01</b> + ,1	1.77E + 04 6.63E + 03 − ,7	5.47E + 03 4.07E + 03 − ,6	2.587E + 04 6.433E + 03 − ,8	3.15E + 03 4.90E + 03 − ,4	3.43E + 03 2.02E + 03 − ,5	1.02E + 02 1.03E + 02 2
F4	Simple multimodal	3.41E + 01 1.80E + 01 − ,2	3.85E + 02 1.17E + 02 − ,8	2.76E + 02 6.55E + 01 − ,7	1.44E + 02 4.37E + 01 − ,4	1.834E + 02 2.583E + 01 − ,5	2.17E + 02 6.23E + 01 − ,6	1.08E + 02 2.96E + 01 − ,3	<b>1.85E − 01</b> <b>2.40E − 01</b> <b>1</b>
F5		2.05E + 01 7.46E − 02 + ,2	2.09E + 01 5.60E − 02 = ,3	<b>2.04E + 01</b> <b>1.44E − 01</b> + ,1	2.05E + 01 3.56E − 01 + ,2	2.090E + 01 7.070E − 02 = ,3	2.10E + 01 8.74E − 02 − ,4	2.09E + 01 4.76E − 02 = ,3	2.09E + 01 5.60E − 02 3
F6		<b>9.84E + 00</b> <b>3.49E + 00</b> + ,1	2.42E + 01 2.27E + 00 − ,7	1.94E + 01 2.89E + 00 − ,4	2.16E + 01 3.46E + 00 − ,6	1.464E + 01 3.352E + 00 = ,2	1.79E + 01 5.31E + 00 − ,3	2.00E + 01 8.49E + 00 − ,5	1.46E + 01 3.66E + 00 2
F7		2.53E − 01 1.43E − 01 − ,3	6.00E + 01 2.36E + 01 − ,8	1.99E + 01 1.18E + 01 − ,7	1.40E + 00 3.12E − 01 − ,4	2.007E + 00 4.617E − 01 − ,5	1.90E + 01 1.51E + 01 − ,6	8.97E − 02 3.38E − 02 − ,2	<b>6.15E − 03</b> <b>1.03E − 02</b> <b>1</b>
F8		<b>4.38E + 01</b> <b>8.48E + 00</b> + ,1	1.16E + 02 2.46E + 01 − ,8	1.07E + 02 2.14E + 01 − ,6	5.89E + 01 1.80E + 01 + ,2	1.096E + 02 1.240E + 01 − ,7	8.32E + 01 2.04E + 01 − ,5	6.62E + 01 1.50E + 01 − ,4	5.90E + 01 1.32E + 01 3
F9		6.33E + 01 1.30E + 01 − ,2	1.25E + 02 2.89E + 01 − ,6	1.39E + 02 2.56E + 01 − ,8	7.22E + 01 2.17E + 01 − ,3	1.308E + 02 9.809E + 00 − ,7	1.09E + 02 2.48E + 01 − ,4	1.24E + 02 1.52E + 01 − ,5	<b>5.41E + 01</b> <b>6.50E + 00</b> <b>1</b>
F10		<b>9.61E + 02</b> <b>2.72E + 02</b> + ,1	2.87E + 03 6.57E + 02 − ,6	2.82E + 03 6.83E + 02 − ,5	3.79E + 03 1.91E + 03 − ,8	3.580E + 03 4.722E + 02 − ,7	2.73E + 03 2.99E + 02 − ,3	2.75E + 03 9.01E + 02 − ,4	2.28E + 03 2.95E + 02 2
F11		<b>2.68E + 03</b> <b>3.68E + 02</b> + ,1	3.28E + 03 5.48E + 02 + ,3	3.30E + 03 6.26E + 02 + ,4	4.53E + 03 1.45E + 03 − ,6	4.934E + 03 4.707E + 02 − ,8	2.70E + 03 7.55E + 02 + ,2	4.56E + 03 1.71E + 02 − ,7	3.50E + 03 5.53E + 02 5
F12		<b>5.45E − 01</b> <b>1.66E − 01</b> + ,1	2.50E + 00 2.66E − 01 − ,8	6.33E − 01 3.36E − 01 + ,2	7.74E − 01 1.10E + 00 + ,3	1.756E + 00 2.886E − 01 + ,4	2.19E + 00 1.09E + 00 − ,7	2.08E + 00 3.96E − 01 − ,6	1.97E + 00 8.19E − 01 5
F13		2.80E − 01 6.30E − 02 − ,3	1.77E + 00 9.83E − 01 − ,8	5.51E − 01 8.94E − 02 − ,7	4.93E − 01 1.15E − 01 − ,6	3.856E − 01 5.918E − 02 − ,4	3.96E − 01 1.22E − 01 − ,5	2.72E − 01 3.48E − 02 − ,2	<b>2.23E − 01</b> <b>6.63E − 02</b> <b>1</b>
F14		4.23E − 01 2.15E − 01 − ,5	2.04E + 01 8.51E + 00 − ,8	2.34E + 00 3.31E + 00 − ,7	2.63E − 01 5.02E − 02 − ,2	2.649E − 01 2.939E − 02 − ,3	1.18E + 00 1.54E + 00 − ,6	3.11E − 01 1.88E − 01 − ,4	<b>2.19E − 01</b> <b>5.11E − 02</b> <b>1</b>
F15		8.81E + 00 1.51E + 00 − ,2	1.20E + 03 1.46E + 03 − ,8	8.72E + 01 1.01E + 02 − ,7	2.43E + 01 7.54E + 00 − ,6	1.512E + 01 1.470E + 00 − ,4	1.78E + 01 8.66E + 00 − ,5	1.43E + 01 8.83E − 01 − ,3	<b>4.77E + 00</b> <b>1.56E + 00</b> <b>1</b>

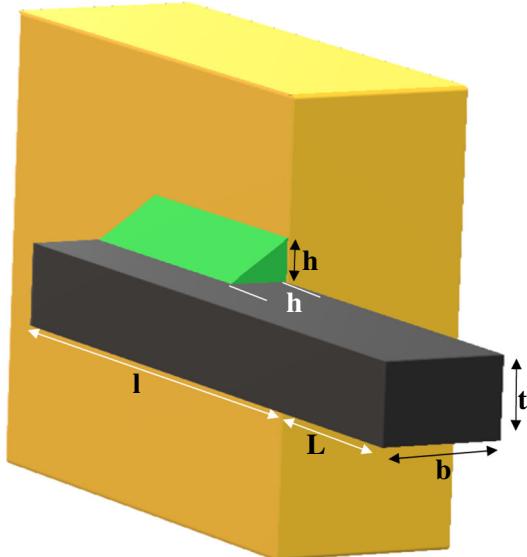
**Table 10** (continued)

Function	RW-GWO [82]	mTLBO [83, 84]	MG-SCA [77]	TOGPEAe [85]	m-SCA [86]	EEO [116]	OB-L-EO [121]	CMOA Mean Std
	Mean	Mean	Mean	Mean	Mean	Mean	Mean	Mean
	Std	Std	Std	Std	Std	Std	Std	Std
	Winner, rank	Winner, rank	Winner, rank	Winner, rank	Winner, rank	Winner, rank	Winner, rank	Winner, rank
F16	<b>1.03E + 01</b>	1.11E + 01	1.16E + 01	1.23E + 01	1.203E + 01	1.12E + 01	1.06E + 01	1.09E + 01
	<b>6.11E - 01</b>	7.48E - 01	6.91E - 01	6.54E - 01	2.893E - 01	9.10E - 01	2.14E - 01	4.40E - 01
	+ ,1	- ,4	- ,6	- ,8	- ,7	- ,5	+ ,2	3
F17	Hybrid	5.71E + 05	6.56E + 05	9.56E + 05	7.12E + 04	5.993E + 05	2.21E + 05	3.90E + 04
	4.10E + 05	1.24E + 06	7.62E + 05	5.59E + 04	3.592E + 05	1.94E + 05	1.48E + 04	<b>2.00E + 04</b>
	- ,5	- ,7	- ,8	- ,3	- ,6	- ,4	- ,2	<b>1.35E + 04</b>
F18		6.52E + 03	1.92E + 04	1.48E + 05	3.07E + 03	1.611E + 05	2.43E + 05	8.83E + 03
	4.62E + 03	8.10E + 04	9.00E + 05	3.05E + 03	8.252E + 04	5.24E + 05	1.85E + 03	<b>1.55E + 02</b>
	- ,3	- ,5	- ,6	- ,2	- ,7	- ,8	- ,4	<b>1.43E + 02</b>
F19		1.14E + 01	8.02E + 01	2.28E + 01	1.60E + 01	1.934E + 01	2.08E + 01	7.14E + 00
	2.03E + 00	3.50E + 01	1.43E + 01	1.80E + 01	5.953E + 00	2.95E + 01	8.73E - 01	<b>5.31E + 00</b>
	- ,3	- ,8	- ,7	- ,4	- ,5	- ,6	- ,2	<b>9.32E - 01</b>
F20		6.27E + 02	<b>2.72E + 02</b>	4.24E + 03	3.31E + 03	1.217E + 04	1.77E + 04	4.38E + 02
	1.12E + 03	<b>1.21E + 02</b>	3.82E + 03	3.67E + 03	3.703E + 03	1.10E + 04	1.01E + 02	5.81E + 02
	- ,4	+ ,1	- ,6	- ,5	- ,7	- ,8	+ ,2	3
F21		2.58E + 05	3.18E + 04	2.35E + 05	1.73E + 04	1.096E + 05	2.43E + 05	1.61E + 05
	1.76E + 05	2.86E + 04	2.39E + 05	1.74E + 04	5.655E + 04	2.50E + 05	7.63E + 04	<b>5.74E + 03</b>
	- ,8	- ,3	- ,6	- ,2	- ,4	- ,7	- ,5	<b>4.10E + 03</b>
F22		2.08E + 02	5.32E + 02	3.39E + 02	6.87E + 02	2.570E + 02	2.00E + 02	1.90E + 02
	1.29E + 02	2.09E + 02	1.78E + 02	1.65E + 02	5.706E + 01	1.37E + 02	1.88E + 01	<b>1.17E + 02</b>
	- ,4	- ,7	- ,6	- ,8	- ,5	- ,3	- ,2	<b>6.12E + 01</b>
F23	Composition	<b>3.15E + 02</b>	3.53E + 02	3.29E + 02	3.17E + 02	3.211E + 02	3.27E + 02	3.25E + 02
	<b>2.77E - 01</b>	1.98E + 01	4.03E + 00	1.93E + 00	1.584E + 00	5.54E + 00	1.39E - 04	<b>3.15E + 02</b>
	= ,1	- ,7	- ,6	- ,2	- ,3	- ,5	- ,4	<b>0.00E + 00</b>
F24		<b>2.00E + 02</b>	<b>2.00E + 02</b>	<b>2.00E + 02</b>	2.48E + 02	<b>2.001E + 02</b>	<b>2.00E + 02</b>	2.29E + 02
	<b>3.04E - 03</b>	<b>7.93E - 04</b>	<b>1.56E - 03</b>	8.19E + 00	<b>4.292E - 02</b>	<b>8.85E - 04</b>	1.20E + 01	1.44E + 00
	+ ,1	+ ,1	+ ,1	- ,4	+ ,1	+ ,1	- ,3	2
F25		2.04E + 02	2.04E + 02	2.11E + 02	2.09E + 02	<b>2.010E + 02</b>	2.10E + 02	2.12E + 02
	1.18E + 00	7.51E + 00	2.82E + 00	7.21E + 00	<b>3.194E + 00</b>	6.18E + 00	3.16E - 01	2.06E + 02
	+ ,2	+ ,2	- ,6	- ,4	+ ,1	- ,5	- ,7	3
F26		<b>1.00E + 02</b>	1.41E + 02	1.01E + 02	1.04E + 02	<b>1.004E + 02</b>	1.20E + 02	<b>1.00E + 02</b>
	<b>7.36E - 02</b>	4.93E + 01	1.53E - 01	1.83E + 01	<b>5.401E - 02</b>	4.45E + 01	<b>3.36E - 02</b>	<b>5.08E - 02</b>
	= ,1	- ,6	- ,2	- ,3	= ,1	- ,5	= ,1	1
F27		4.09E + 02	9.40E + 02	8.19E + 02	9.26E + 02	4.328E + 02	4.22E + 02	4.19E + 02
	6.09E + 00	2.48E + 02	9.17E + 01	8.92E + 01	2.008E + 01	1.19E + 02	7.37E + 01	<b>4.01E + 02</b>
	- ,2	- ,8	- ,6	- ,7	- ,5	- ,4	- ,3	<b>4.04E - 01</b>
F28		<b>4.34E + 02</b>	2.36E + 03	9.68E + 02	1.58E + 03	1.054E + 03	1.15E + 03	1.49E + 03
	<b>8.45E + 00</b>	5.34E + 02	1.06E + 02	2.22E + 02	2.735E + 02	3.41E + 02	1.45E + 01	9.75E + 02
	+ ,1	- ,8	+ ,2	- ,7	- ,4	- ,5	- ,6	3
F29		<b>2.14E + 02</b>	5.45E + 06	1.19E + 06	2.09E + 03	4.443E + 04	2.40E + 06	2.95E + 03
	<b>2.37E + 00</b>	1.21E + 07	3.25E + 06	5.156E + 02	1.769E + 04	4.01E + 06	9.06E + 02	1.24E + 03
	+ ,1	- ,8	- ,6	- ,3	- ,5	- ,7	- ,4	2
F30		<b>6.69E + 02</b>	8.61E + 04	1.92E + 04	1.09E + 04	3.484E + 04	4.26E + 04	2.37E + 03
	<b>2.14E + 02</b>	6.81E + 04	8.25E + 03	6.80E + 03	1.054E + 04	2.70E + 04	8.45E + 02	2.12E + 03
	+ ,1	- ,8	- ,5	- ,4	- ,6	- ,7	- ,3	2

**Table 10** (continued)

Function	RW-GWO [82]	mTLBO [83, 84]	MG-SCA [77]	TOGPEAe [85]	m-SCA [86]	EEO [116]	OB-L-EO [121]	CMOA Mean Std
	Mean	Mean	Mean	Mean	Mean	Mean	Mean	Winner, rank
	Std	Std	Std	Std	Std	Std	Std	Winner, rank
	Winner, rank	Winner, rank	Winner, rank	Winner, rank	Winner, rank	Winner, rank	Winner, rank	Winner, rank
Nb/Nw/Mean	12/1/2.3667	3/15/5.9667	2/3/5.5000	0/4/4.3667	3/2/4.8333	1/4/5.0667	1/1/3.6333	<b>16/0/1.8667</b>
+ / - / =	12/16/2	5/24/1	5/25/0	3/27/0	3/24/3	2/28/0	2/26/2	-
Final rank	2	8	7	4	5	6	3	<b>1</b>

Results of Coronavirus Metamorphosis Optimization Algorithm (CMOA)

**Fig. 48** Optimal design of a welded beam**Table 11** Best statistical results obtained by various algorithms for the welded beam design problem

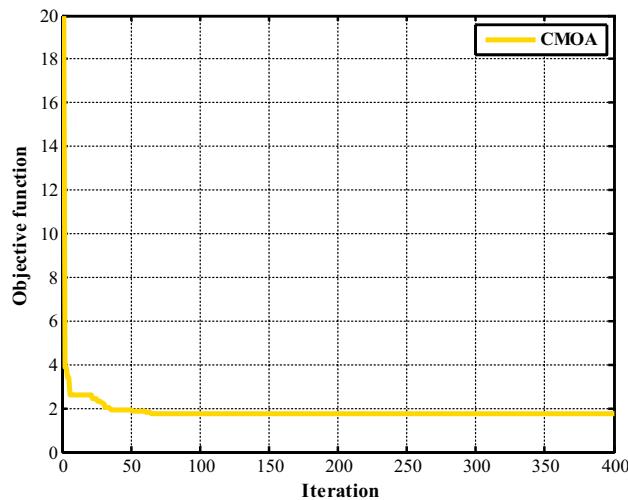
Methods	Best	Mean	Worst	Std
SFFA [88]	1.724868	1.724953	1.724868	4.6E – 05
EPSO [89]	1.7248530	1.7282190	1.7472200	5.62E – 03
SFO [27]	1.73231	N.A	N.A	N.A
EO [90]	1.724853	1.726482	1.736725	3.257E – 03
FSA [91]	2.3811	2.4041	2.4889	N.A
UPSO [92]	1.92199	2.83721	N.A	6.83E – 01
SBM [93]	2.4426	2.5215	2.6315	N.A
TEO [42]	1.725284	1.768040	1.931161	5.81661E – 02
WCA[94]	1.724856	1.726427	1.744697	4.29E – 03
BA [95]	1.7312065	1.8786560	2.3455793	2.677989E – 01
PFA [26]	1.7248530	N.A	N.A	N.A
MRFO [96]	<b>1.7248523</b>	1.7248547	1.7248648	3.832E – 06
GWO [97]	1.72624	N.A	N.A	N.A
HGSO [43]	1.7260	1.7265	1.7325	7.66E – 03
CMOA	1.724852	1.7248544	1.7248646	1.97E – 06

Results of Coronavirus Metamorphosis Optimization Algorithm (CMOA)

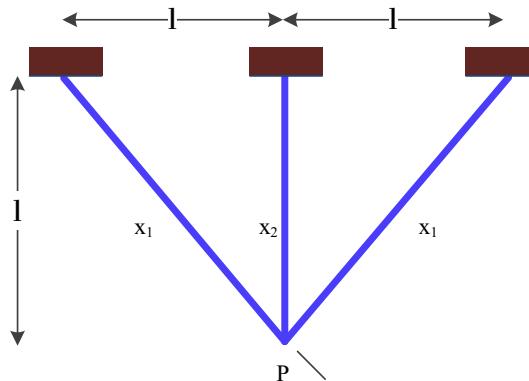
**Table 12** The best solutions achieved for the welded beam optimal design problem

Design variables	CMOA
$x_1$	0.20572964
$x_2$	3.470488666
$x_3$	9.03662391
$x_4$	0.20572964
$g_1(X)$	– 2.2653E – 07
$g_2(X)$	– 3.19327E – 07
$g_3(X)$	0.0
$g_4(X)$	– 3.43298
$g_5(X)$	– 0.08073
$g_6(X)$	– 0.23554
$g_7(X)$	– 1.1055E – 06
Best	<b>1.724852</b>

Results of Coronavirus Metamorphosis Optimization Algorithm (CMOA)



**Fig. 49** Convergence characteristic of the welded beam optimal design problem

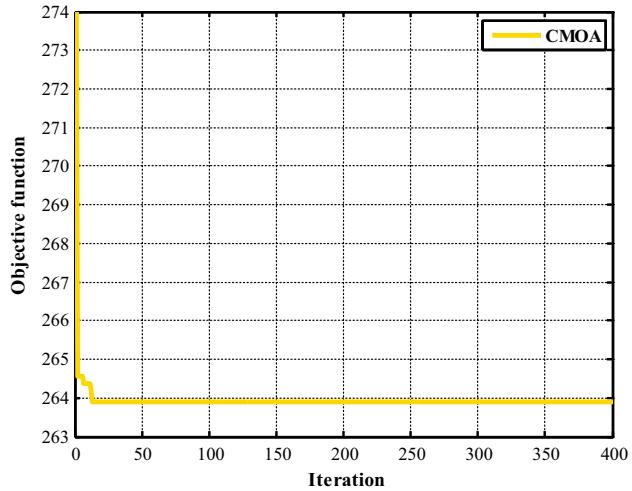


**Fig. 50** Depiction of the three-bar truss design

**Table 14** The best solutions for the optimal design problem of the three-bar truss structure

Design variables	CMOA
$x_1$	0.78871700
$x_2$	0.40812990
$g_1(X)$	- 7.50879E - 09
$g_2(X)$	- 1.4642362
$g_3(X)$	- 0.535764
Best	<b>263.89584</b>

Results of Coronavirus Metamorphosis Optimization Algorithm (CMOA)



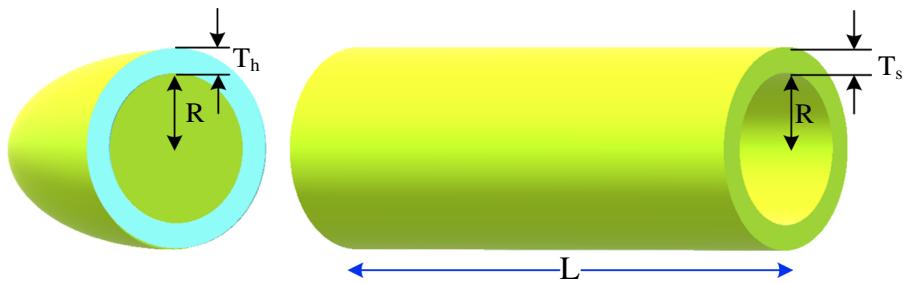
**Fig. 51** Convergence characteristic of the three-bar truss design problem

**Table 13** Best statistical results of different algorithms for the three-bar truss problem

Methods	Best	Mean	Worst	Std
MBA (mine blast algorithm) [4]	263.895852	263.897996	263.915983	3.93E - 03
Rao-3 [101]	<b>263.895843</b>	263.897243	263.899650	1.0390E - 03
Rao-2 [101]	263.895846	263.897469	263.899734	1.0976E - 03
Rao-1 [101]	<b>263.895841</b>	263.896207	263.897166	3.7109E - 04
RL-BA (adaptive reinforcement learning-based BA) [98]	<b>263.89584</b>	263.9003	263.924700	6.06E - 03
SFO [27]	263.89592128	N.A	N.A	N.A
DSS-MDE (DE with dynamic stochastic selection) [100]	263.8958434	263.8981518	263.95226	9.19E - 03
CS (Cuckoo search) [99]	263.97156	264.0669	N.A	9.0E - 05
CMOA	<b>263.89584</b>	263.896198	263.897157	6.25E - 05

Results of Coronavirus Metamorphosis Optimization Algorithm (CMOA)

**Fig. 52** The pressure vessel design problem



**Table 15** Best statistical results of different algorithms for the pressure vessel

Methods	Best	Mean	Worst	Std
CVI-PSO (PSO with a new constraint-handling mechanism) [102]	<b>6059.7143</b>	6292.1231	6820.4101	288.4550
CDE (co-evolutionary DE) [103]	6059.7340	6085.2303	6371.0455	43.013
QPSO (quantum-behaved PSO) [104]	6059.7209	6440.3786	8017.2816	479.2671
DHOA (deer hunting optimization algorithm) [105]	6103.842	N.A	N.A	N.A
UPSO [92]	6154.70	8016.37	9387.77	745.869
CB-ABC (crossover-based artificial bee colony (ABC)) [110]	<b>6059.714335</b>	6126.623676	N.A	1.14E + 02
QS (queuing search) [50]	<b>6059.714</b>	6060.947	6090.526	N.A
CSA (crow search algorithm) [87]	6059.71436343	6342.49910551	7332.8416211	384.94541634
BFOA (bacterial foraging algorithm) [107]	6060.460	6074.625	N.A	156
SFFA [88]	6059.71441707	6061.25542415	6090.52625202	6.88963799
PVS (passing vehicle search) [106]	<b>6059.714</b>	6065.877	6090.526	N.A
BIANCA (genetic algorithms (GA) with automatic dynamic penalization method (ADP)) [111]	6059.9384	6182.0022	6447.3251	122.3256
GA3 (GA with a self-adaptive penalty approach) [108]	6288.7445	6293.8432	6308.4970	7.4133
Rao-3 [101]	<b>6059.714334</b>	6061.883052	6090.526202	7.810982
Rao-2 [101]	<b>6059.714334</b>	6062.055668	6090.526202	7.171409
Rao-1 [101]	<b>6059.714334</b>	6069.230694	6093.903548	10.451664
ABC [109]	6059.714339	6245.308144	N.A	2.05E + 02
<b>CMOA</b>	<b>6059.71433</b>	<b>6060.55627</b>	<b>6090.5262</b>	<b>4.093</b>

Results of Coronavirus Metamorphosis Optimization Algorithm (CMOA)

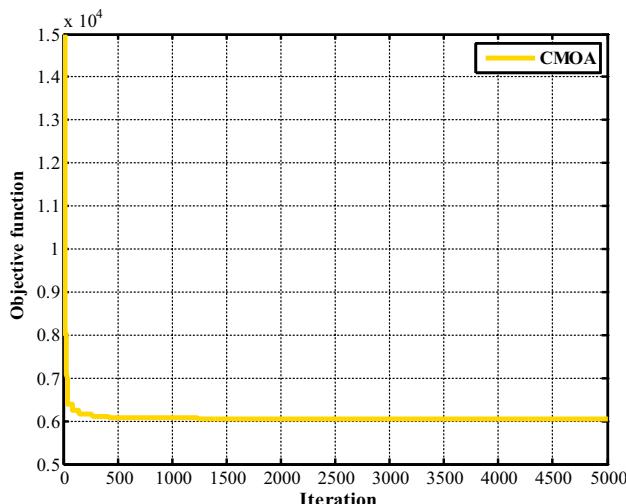
**Table 16** The best solutions for the pressure vessel optimal design problem

Design variables	CMOA
$x_1$	0.8125
$x_2$	0.4375
$x_3$	42.09845
$x_4$	176.6366
$g_1(X)$	- 1.130E - 10
$g_2(X)$	- 0.035881
$g_3(X)$	- 2.78875E - 05
$g_4(X)$	- 63.3634
Best	<b>6059.71433</b>

Results of Coronavirus Metamorphosis Optimization Algorithm (CMOA)

## 5 Conclusion

The world we live in includes so complex problems that both academia and industry are in dire need of a robust and appropriate algorithm to find the best possible solution to these functions. In this study, we designed and proposed a new metaheuristic algorithm called CMOA, genetically inspired by a newly emerging viral disease of the world, coronavirus disease 2019 (COVID-19), taking into account several hypotheses. The proposed algorithm has several synchronous jump and motion vectors and therefore has a suitable optimization power over a wide range. To evaluate the power and performance of the proposed CMOA algorithm, it was run on modern functions and CEC2014 standard and compared with some novel optimization methods. The results obtained in this comparative study well indicate the robustness and effectiveness of the



**Fig. 53** Convergence characteristic of the pressure vessel design problem

proposed CMOA algorithm as an effective and noteworthy algorithm.

The present research tested three popular engineering problems to investigate the efficacy of the CMOA. According to the obtained results, CMOA can improve the results strongly when compared with other methods.

## Declarations

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

## References

- Gogna A, Tayal A (2013) Metaheuristics: review and application. *J Exp Theor Artif Intell* 25:503–526
- Darwish A, Hassani AE, Das S (2020) A survey of swarm and evolutionary computing approaches for deep learning. *Artif Intell Rev* 53:1767–1812
- Khajehzadeh M, Taha MR, El-Shafie A, Eslami M (2011) A survey on meta-heuristic global optimization algorithms. *Res J Appl Sci Eng Technol* 3:569–578
- Saddollah A, Bahreininejad A, Eskandar H et al (2013) Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Appl Soft Comput* 13:2592–2612
- Petrowski JDA, Taillard PSE (2006) Metaheuristics for hard optimization. Springer
- Holland JH (1992) Genetic algorithms. *Scientific american* 267:66–73
- Akyol S, Alatas B (2017) Plant intelligence based metaheuristic optimization algorithms. *Artif Intell Rev* 47:417–462. <https://doi.org/10.1007/s10462-016-9486-6>
- Jaderyan M, Khotanlou H (2016) Virulence optimization algorithm. *Appl Soft Comput* 43:596–618
- Salmani MH, Eshghi K (2017) A metaheuristic algorithm based on chemotherapy science: CSA. *J Optim*. <https://doi.org/10.1155/2017/308204>
- Jaddi NS, Alvankarian J, Abdullah S (2017) Kidney-inspired algorithm for optimization problems. *Commun Nonlinear Sci Numer Simul* 42:358–369
- Huang G (2016) Artificial infectious disease optimization: A SEIQR epidemic dynamic model-based function optimization algorithm. *Swarm Evol Comput* 27:31–67. <https://doi.org/10.1016/j.swevo.2015.09.007>
- Ghasemi M, Taghizadeh M, Ghavidel S, Abbasian A (2016) Colonial competitive differential evolution: an experimental study for optimal economic load dispatch. *Appl Soft Comput* 40:342–363
- Cheraghaliour A, Hajiaghaei-Keshteli M, Paydar MM (2018) Tree Growth Algorithm (TGA): a novel approach for solving optimization problems. *Eng Appl Artif Intell* 72:393–414
- Tang D, Dong S, Jiang Y et al (2015) ITGO: invasive tumor growth optimization algorithm. *Appl Soft Comput* 36:670–698. <https://doi.org/10.1016/j.asoc.2015.07.045>
- Mehraban AR, Lucas C (2006) A novel numerical optimization algorithm inspired from weed colonization. *Eco Inform* 1:355–366
- Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: 2007 IEEE Congress on Evolutionary Computation. pp 4661–4667
- Rao R (2016) Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int J Ind Eng Comput* 7:19–34
- Dhiman G, Kumar V (2018) Emperor penguin optimizer: a bio-inspired algorithm for engineering problems. *Knowl Based Syst* 159:20–50
- Baykasoglu A, Akpinar S (2017) Weighted Superposition Attraction (WSA): a swarm intelligence algorithm for optimization problems—Part 1: unconstrained optimization. *Appl Soft Comput* 56:520–540
- Zhou Y, Wang Y, Chen X et al (2017) A novel path planning algorithm based on plant growth mechanism. *Soft Comput* 21:435–445
- Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. *Knowl-Based Syst* 96:120–133
- 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). IEEE, pp 1470–1477
- Heidari AA, Mirjalili S, Faris H et al (2019) Harris hawks optimization: Algorithm and applications. *Futur Gener Comput Syst* 97:849–872
- Zaldivar D, Morales B, Rodríguez A et al (2018) A novel bio-inspired optimization model based on Yellow Saddle Goatfish behavior. *Biosystems* 174:1–21
- Jahani E, Chizari M (2018) Tackling global optimization problems with a novel algorithm—Mouth Brooding Fish algorithm. *Appl Soft Comput* 62:987–1002
- Yapıcı H, Çetinkaya N (2019) A new meta-heuristic optimizer: pathfinder algorithm. *Appl Soft Comput* 78:545–568
- Shadravan S, Naji HR, Bardsiri VK (2019) The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Eng Appl Artif Intell* 80:20–34
- Yazdani M, Jolai F (2016) Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. *Journal of computational design and engineering* 3:24–36
- Cheng L, Wu X, Wang Y (2018) Artificial flora (AF) optimization algorithm. *Appl Sci* 8:329

30. Feng Z, Niu W, Liu S (2021) Cooperation search algorithm: A novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems. *Appl Soft Comput* 98:106734
31. Jain M, Singh V, Rani A (2019) A novel nature-inspired algorithm for optimization: Squirrel search algorithm. *Swarm Evol Comput* 44:148–175
32. Ghasemi M, Akbari E, Rahimnejad A et al (2019) Phasor particle swarm optimization: a simple and efficient variant of PSO. *Soft Comput* 23:9701–9718
33. Zhang Q, Wang R, Yang J et al (2017) Collective decision optimization algorithm: a new heuristic optimization method. *Neurocomputing* 221:123–137
34. Kallioras NA, Lagaros ND, Avtzis DN (2018) Pity beetle algorithm—A new metaheuristic inspired by the behavior of bark beetles. *Adv Eng Softw* 121:147–166
35. Ghasemi M, Ghavidel S, Aghaei J et al (2018) CFA optimizer: A new and powerful algorithm inspired by Franklin's and Coulomb's laws theory for solving the economic load dispatch problems. *International Transactions on Electrical Energy Systems* 28:e2536
36. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing science 220:671–680
37. Punnathanam V, Kotcha P (2016) Yin-Yang-pair Optimization: A novel lightweight optimization algorithm. *Eng Appl Artif Intell* 54:62–79
38. Abedinpourshotorban H, Shamsuddin SM, Beheshti Z, Jawawi DNA (2016) Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. *Swarm Evol Comput* 26:8–22
39. Alatas B (2011) ACROA: artificial chemical reaction optimization algorithm for global optimization. *Expert Syst Appl* 38:13170–13180
40. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. *Inf Sci* 179:2232–2248
41. Zhao W, Wang L, Zhang Z (2019) A novel atom search optimization for dispersion coefficient estimation in groundwater. *Futur Gener Comput Syst* 91:601–610
42. Kaveh A, Dadras A (2017) A novel meta-heuristic optimization algorithm: thermal exchange optimization. *Adv Eng Softw* 110:69–84
43. Hashim FA, Houssein EH, Mabrouk MS, et al (2019) Henry gas solubility optimization: A novel physics-based algorithm. *Futur Gener Comput Syst* 101:646–667
44. Kaveh A, Bakhshpoori T (2016) Water evaporation optimization: a novel physically inspired optimization algorithm. *Comput Struct* 167:69–85
45. Patel VK, Savsani VJ (2015) Heat transfer search (HTS): a novel optimization algorithm. *Inf Sci* 324:217–246
46. Vommi VB, Vemula R (2018) A very optimistic method of minimization (VOMMI) for unconstrained problems. *Inf Sci* 454:255–274
47. Moghdani R, Salimifard K (2018) Volleyball premier league algorithm. *Appl Soft Comput* 64:161–185
48. Hudaib AA, Fakhouri HN (2018) Supernova optimizer: a novel natural inspired meta-heuristic. *Mod Appl Sci* 12:32–50
49. Kaboli SHA, Selvaraj J, Rahim NA (2017) Rain-fall optimization algorithm: A population based algorithm for solving constrained optimization problems. *Journal of Computational Science* 19:31–42
50. Zhang J, Xiao M, Gao L, Pan Q (2018) Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems. *Appl Math Model* 63:464–490
51. Muthiah-Nakarajan V, Noel MM (2016) Galactic Swarm Optimization: A new global optimization metaheuristic inspired by galactic motion. *Appl Soft Comput* 38:771–787
52. Ghasemi M, Davoudkhani IF, Akbari E et al (2020) A novel and effective optimization algorithm for global optimization and its engineering applications: Turbulent Flow of Water-based Optimization (TFWO). *Eng Appl Artif Intell* 92:103666
53. Lau SKP, Lee P, Tsang AKL et al (2011) Molecular epidemiology of human coronavirus OC43 reveals evolution of different genotypes over time and recent emergence of a novel genotype due to natural recombination. *J Virol* 85:11325–11337
54. Wu Z, McGoogan JM (2020) Characteristics of and important lessons from the coronavirus disease 2019 (COVID-19) outbreak in China: summary of a report of 72 314 cases from the Chinese Center for Disease Control and Prevention. *JAMA* 323:1239–1242
55. Cowling BJ, Ali ST, Ng TWY et al (2020) Impact assessment of non-pharmaceutical interventions against coronavirus disease 2019 and influenza in Hong Kong: an observational study. *The Lancet Public Health* 5:e279–e288
56. Sun K, Chen J, Viboud C (2020) Early epidemiological analysis of the coronavirus disease 2019 outbreak based on crowd-sourced data: a population-level observational study. *The Lancet Digital Health* 2:e201–e208
57. Anderson RM, Heesterbeek H, Klinkenberg D, Hollingsworth TD (2020) How will country-based mitigation measures influence the course of the COVID-19 epidemic? *The lancet* 395:931–934
58. Graham RL, Donaldson EF, Baric RS (2013) A decade after SARS: strategies for controlling emerging coronaviruses. *Nat Rev Microbiol* 11:836–848
59. Liang JJ, Qu BY, Suganthan PN (2013) Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore 635:490
60. Dehghani M, Mardaneh M, Guerrero JM et al (2020) Football game based optimization: An application to solve energy commitment problem. *Int J Intell Eng Syst* 13:514–523
61. Ghasemi M, Rahimnejad A, Hemmati R et al (2021) Wild Geese Algorithm: A novel algorithm for large scale optimization based on the natural life and death of wild geese. *Array* 11:100074
62. Abualigah L, Yousri D, Abd Elaziz M et al (2021) Aquila optimizer: a novel meta-heuristic optimization algorithm. *Comput Ind Eng* 157:107250
63. Yang Y, Chen H, Heidari AA, Gandomi AH (2021) Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Syst Appl* 177:114864
64. Potap D, Woźniak M (2021) Red fox optimization algorithm. *Expert Syst Appl* 166:114107
65. Abualigah L, Diabat A, Mirjalili S et al (2021) The arithmetic optimization algorithm. *Comput Methods Appl Mech Eng* 376:113609
66. Al-Betar MA, Alyasseri ZAA, Awadallah MA, Abu Doush I (2021) Coronavirus herd immunity optimizer (CHIO). *Neural Comput Appl* 33:5011–5042
67. Martínez-Alvarez F, Asencio-Cortés G, Torres JF et al (2020) Coronavirus optimization algorithm: a bioinspired metaheuristic based on the COVID-19 propagation model. *Big data* 8:308–322
68. Faria NR, Mellan TA, Whittaker C et al (2021) Genomics and epidemiology of the P. 1 SARS-CoV-2 lineage in Manaus. *Brazil Science* 372:815–821
69. Fontanet A, Autran B, Lina B et al (2021) SARS-CoV-2 variants and ending the COVID-19 pandemic. *The Lancet* 397:952–954
70. Gu H, Chen Q, Yang G et al (2020) Adaptation of SARS-CoV-2 in BALB/c mice for testing vaccine efficacy. *Science* 369:1603–1607

71. Chen Y, Guo Y, Pan Y, Zhao ZJ (2020) Structure analysis of the receptor binding of 2019-nCoV. *Biochem Biophys Res Commun* 525:135–140
72. Korber B, Fischer WM, Gnanakaran S et al (2020) Tracking changes in SARS-CoV-2 spike: evidence that D614G increases infectivity of the COVID-19 virus. *Cell* 182:812–827
73. Faria NR, Claro IM, Candido D et al (2021) Genomic characterisation of an emergent SARS-CoV-2 lineage in Manaus: preliminary findings. *Virological* 372:815–821
74. Sree Ranjini KS, Murugan S (2017) Memory based hybrid dragonfly algorithm for numerical optimization problems. *Expert Syst Appl* 83:63–78
75. Li MD, Zhao H, Weng XW, Han T (2016) A novel nature-inspired algorithm for optimization: Virus colony search. *Adv Eng Softw* 92:65–88
76. Gupta S, Deep K (2020) A memory-based grey wolf optimizer for global optimization tasks. *Appl Soft Comput* 93:106367
77. Gupta S, Deep K, Engelbrecht AP (2020) A memory guided sine cosine algorithm for global optimization. *Eng Appl Artif Intell* 93:103718
78. Mirjalili S (2015) Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl-Based Syst* 89:228–249
79. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360). IEEE, pp 69–73
80. Iacca G, dos Santos Junior VC, de Melo VV (2021) An improved Jaya optimization algorithm with Lévy flight. *Expert Syst Appl* 165:113902
81. Long W, Cai S, Jiao J, Tang M (2020) An efficient and robust grey wolf optimizer algorithm for large-scale numerical optimization. *Soft Comput* 24:997–1026
82. Gupta S, Deep K (2019) A novel random walk grey wolf optimizer. *Swarm Evol Comput* 44:101–112
83. Satapathy SC, Naik A (2014) Modified Teaching–Learning-Based Optimization algorithm for global numerical optimization—A comparative study. *Swarm Evol Comput* 16:28–37
84. Chen X, Xu B, Yu K, Du W (2018) Teaching-learning-based optimization with learning enthusiasm mechanism and its application in chemical engineering. *J Appl Math* 2018:19
85. Dai C, Hu Z, Li Z et al (2020) An improved grey prediction evolution algorithm based on topological opposition-based learning. *IEEE Access* 8:30745–30762
86. Gupta S, Deep K (2019) A hybrid self-adaptive sine cosine algorithm with opposition based learning. *Expert Syst Appl* 119:210–230
87. Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Comput Struct* 169:1–12
88. Baykasoglu A, Ozsoydan FB (2015) Adaptive firefly algorithm with chaos for mechanical design optimization problems. *Appl Soft Comput* 36:152–164
89. Ngo TT, Sadollah A, Kim JH (2016) A cooperative particle swarm optimizer with stochastic movements for computationally expensive numerical optimization problems. *Journal of Computational Science* 13:68–82
90. Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S (2020) Equilibrium optimizer: A novel optimization algorithm. *Knowl-Based Syst* 191:105190
91. Hedar A-R, Fukushima M (2006) Derivative-free filter simulated annealing method for constrained continuous global optimization. *J Global Optim* 35:521–549
92. Parsopoulos KE, Vrahatis MN (2005) Unified particle swarm optimization for solving constrained engineering optimization problems. In: International conference on natural computation. Springer, pp 582–591
93. Akhtar S, Tai K, Ray T (2002) A socio-behavioural simulation model for engineering design optimization. *Eng Optim* 34:341–354
94. 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
95. Gandomi AH, Yang X-S, Alavi AH, Talatahari S (2013) Bat algorithm for constrained optimization tasks. *Neural Comput Appl* 22:1239–1255
96. Zhao W, Zhang Z, Wang L (2020) Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Eng Appl Artif Intell* 87:103300
97. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
98. Meng X-B, Li H-X, Gao X-Z (2019) An adaptive reinforcement learning-based bat algorithm for structural design problems. *International Journal of Bio-Inspired Computation* 14:114–124
99. Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with Computers* 29:17–35
100. Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. *Inf Sci* 178:3043–3074
101. Rao RV, Pawar RB (2020) Self-adaptive multi-population Rao algorithms for engineering design optimization. *Appl Artif Intell* 34:187–250
102. Mazhoud I, Hadj-Hamou K, Bigeon J, Joyeux P (2013) Particle swarm optimization for solving engineering problems: a new constraint-handling mechanism. *Eng Appl Artif Intell* 26:1263–1273
103. Huang F, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. *Appl Math Comput* 186:340–356
104. dos Santos CL (2010) Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Syst Appl* 37:1676–1683
105. Brammya G, Praveena S, Ninu Preetha NS et al (2019) Deer hunting optimization algorithm: a new nature-inspired metaheuristic paradigm. *Comput J*. <https://doi.org/10.1093/comjnl/bxy133>
106. Savsani P, Savsani V (2016) Passing vehicle search (PVS): A novel metaheuristic algorithm. *Appl Math Model* 40:3951–3978
107. Mezura-Montes E, Hernández-Ocana B (2008) Bacterial foraging for engineering design problems: preliminary results. In: Memorias del 4o Congreso Nacional de Computación Evolutiva (COMCEV'2008). Centro de Investigación en Matemáticas Guanajuato, México
108. Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. *Comput Ind* 41:113–127
109. Akay B, Karaboga D (2012) Artificial bee colony algorithm for large-scale problems and engineering design optimization. *J Intell Manuf* 23:1001–1014
110. Brajevic I (2015) Crossover-based artificial bee colony algorithm for constrained optimization problems. *Neural Comput Appl* 26:1587–1601
111. Montemurro M, Vincenti A, Vannucci P (2013) The automatic dynamic penalisation method (ADP) for handling constraints with genetic algorithms. *Comput Methods Appl Mech Eng* 256:70–87
112. Mlakar U, Fister I Jr, Fister I (2016) Hybrid self-adaptive cuckoo search for global optimization. *Swarm Evol Comput* 29:47–72

113. Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. *Comput Struct* 139:98–112
114. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
115. Hansen N, Müller SD, Koumoutsakos P (2003) Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evol Comput* 11:1–18
116. Ates A (2021) Enhanced equilibrium optimization method with fractional order chaotic and application engineering. *Neural Comput Appl* 33:9849–9876
117. Nguyen TT, Nguyen TT, Duong MQ (2022) An improved equilibrium optimizer for optimal placement of photovoltaic systems in radial distribution power networks. *Neural Comput Appl* 34:6119–6148
118. Tabak A (2021) A novel fractional order PID plus derivative ( $\text{PI}^{\lambda} \text{D}^{\mu} \text{D}^{\mu} 2$ ) controller for AVR system using equilibrium optimizer. *COMPEL Int J Comput Math Electric Electron Eng.* <https://doi.org/10.1108/COMPEL-02-2021-0044>
119. Mahmoodjanloo M, Tavakkoli-Moghaddama R, Baboli A, Bozorgi-Amiri A (2021) Distributed job-shop rescheduling problem considering reconfigurability of machines: a self-adaptive hybrid equilibrium optimiser. *Int J Prod Res.* <https://doi.org/10.1080/00207543.2021.1946193>
120. Houssein EH, Dirar M, Abualigah L, Mohamed WM (2022) An efficient equilibrium optimizer with support vector regression for stock market prediction. *Neural Comput Appl* 34:3165–3200
121. Dinkar SK, Deep K, Mirjalili S, Thapliyal S (2021) Opposition-based Laplacian equilibrium optimizer with application in image segmentation using multilevel thresholding. *Expert Syst Appl* 174:114766

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.