Soft Computing (2022) 26:4991–5023
https://doi.org/10.1007/s00500-022-06903-5

FOUNDATIONS

Anti-coronavirus optimization algorithm

Hojjat Emami[1]

Accepted: 4 February 2022 / Published online: 14 March 2022

Abstract
This paper introduces a new swarm intelligence strategy, anti-coronavirus optimization (ACVO) algorithm. This algorithm is a multi-agent strategy, in which each agent is a person that tries to stay healthy and slow down the spread of COVID-19 by observing the containment protocols. The algorithm composed of three main steps: social distancing, quarantine, and isolation. In the social distancing phase, the algorithm attempts to maintain a safe physical distance between people and limit close contacts. In the quarantine phase, the algorithm quarantines the suspected people to prevent the spread of disease. Some people who have not followed the health protocols and infected by the virus should be taken care of to get a full recovery. In the isolation phase, the algorithm cared for the infected people to recover their health. The algorithm iteratively applies these operators on the population to find the fittest and healthiest person. The proposed algorithm is evaluated on standard multi-variable single-objective optimization problems and compared with several counterpart algorithms. The results show the superiority of ACVO on most test problems compared with its counterparts.

Keywords Algorithms · Swarm intelligence · Coronavirus · Anti-coronavirus optimization algorithm · Numerical optimization

1 Introduction

In recent years, optimization meta-heuristics have become increasingly one of the most interesting topics in computer science (Krause and Cordeiro 2013). A meta-heuristic algorithm is a black-box optimizer system that gets a set of problems' variables that need to be tuned and even some constraints in the form of limitations. The optimizer modifies the variables by running an updating process until reaching the optimum value of an objective function. The output is a near-optimal solution with the maximum/ minimum value of the objective function. Meta-heuristics are efficient, fairly simple, flexible, derivation free, and need limited prior knowledge about problems (Mirjalili and Lewis 2016). They have drawn inspiration from a biological, physical, or social phenomenon. For example, the inspiration source of the artificial bee colony (ABC) algorithm (Karaboga and Basturk 2007) is the cooperation of honey bees in finding food sources; and the motivation of the chaotic presidential election (CPE) algorithm (Emami and Derakhshan 2015) is the people's democratic behavior in the presidential election.

✉ Hojjat Emami
emami@ubonab.ac.ir

[1] Department of Computer Engineering, University of Bonab, Bonab, Iran

In general, the main objective of all meta-heuristics is to obtain the best solutions with the minimum computational complexity in a reasonable time. Meta-heuristics should provide a proper trade-off between the exploration (diversification) and exploitation (intensification). The objective of diversification is to disperse the search agents around the solution space to efficiently explore the promising areas, while intensification ensures the algorithm searches around the current best solutions (Yang and Deb 2009). The main difference between meta-heuristics is the way they adopted to achieve the best solutions. According to the search strategy and source of inspiration, meta-heuristics are classified into different groups. The most studied meta-heuristics are evolutionary and swarm intelligence algorithms. Several thorough surveys of recent meta-heuristics are given in Krause and Cordeiro (2013); Abdel-Basset et al. (2018); Boussaïd et al. (2013).

Evolutionary algorithms are motivated by Darwin's evolutionary theory. They work with a collection of solutions and generate increasingly better solutions over time using reproduction operators. The success of evolutionary algorithms is due to their ability in modeling the best features in nature, particularly biological systems evolved over millions of years. Adaptation to the environment and selection of the fittest individuals are two important characteristics of the evolutionary algorithms (Emami and Derakhshan 2015).

These algorithms can handle the problems with many variables. In the beginning, the mainstream of evolutionary algorithms was the genetic algorithm (GA) (Haupt and Haupt 2004). In recent years, new algorithms have emerged and achieved encouraging results. The GA algorithm is inspired by natural selection and genetic variation. GA is a stochastic search approach, which works with a randomly initialized set of chromosomes. The quality of chromosomes is evaluated using a fitness function. Chromosomes are combined to generate offspring individuals through selection, crossover, and mutation. The least-fit chromosomes are replaced with offspring if they achieve better fitness. This process iterates several times until the termination conditions are met. Some distinguished evolutionary algorithms are differential evolution (DE) (Storn and Price 1997), genetic programming (GP) (Yao et al. 1999), gene expression programming (GEP) (Ferreira 2001), covariance matrix adaptation evolution strategy (CMA-ES) (Igel et al. 2007), self-adaptive differential evolution (Qin and Suganthan 2005), evolution strategy (ES) (Taylor et al. 2008), biogeography-based optimization (BBO) (Simon 2008), granular agent evolutionary (GAE) (Pan and Jiao 2011), cultural evolution algorithm (CEA) (Kuo and Lin 2013), backtracking search optimization algorithm (BSA) (Civicioglu 2013), symbiotic organisms search (SOS) (Cheng and Prayogo 2014), strawberry plant algorithm (SBA) (Yamato and Fujiwara 2019), forest optimization algorithm (FOA) (Ghaemia and Feizi-Derakhshi 2014), and runner-root algorithm (RRA) (Merrikh-Bayat 2015).

Swarm intelligence algorithms mimic the intelligent behavior of insects or groups of animals in nature to solve real-world problems. These algorithms mainly depend on the decentralization idea in which the individuals search the solution space by extracting the useful information from a group of individuals and the environment (Abdel-Basset et al. 2018). Particle swarm optimization (PSO) (Kennedy and Eberhart 1995) is a popular and oldest swarm intelligence algorithm, which is widely used in different disciplines. PSO models the cooperation of a flock of migrating birds attempting to reach destination. In PSO, each search agent in solution space is called a particle. The particles explore the solution space by sharing their own local experience and the best-known experience of the others. This process iterates so that the particles move toward the best solution. Some of the well-known swarm intelligence algorithms are artificial bee colony (ABC) (Karaboga and Basturk 2007), cuckoo search algorithm (CSA) (Yang and Deb 2009), ant colony optimization (ACO) (Dorigo et al. 2006), firefly algorithm (FA) (Yang 2009), bat algorithm (BA) (Yang 2010), bacterial foraging optimization (BFO) (Passino and Ohio 2010), teaching learning-based optimization (TLBO) (Rao et al. 2011), krill herd (KH) (Gandomia and Alavi 2012), dolphin echolocation (DEL) (Kaveh and Farhoudi 2013), fruit fly optimization algorithm (FOA) (Pan 2012), grey wolf optimization (GWO) (Mirjalili et al. 2014), animal migration optimization (AMO) (Li et al. 2014), social spider optimization (SSO) (Yu and Li 2015), group counseling optimization (GCO) (Eita and Fahmy 2014), lion optimization algorithm (LOA) (Yazdani and Jolai 2016), whale optimization algorithm (WOA) (Mirjalili and Lewis 2016), grasshopper optimization algorithm (GOA) (Saremi et al. 2017), and salp swarm algorithm (SSA) (Mirjalili et al. 2017).

A few recent algorithms are inspired by the behavior of viruses that attack a living cell. An interesting algorithm in this area is bacterial foraging optimization (BFO) algorithm (Passino and Ohio 2010) that models the foraging behavior of bacteria over a landscape of nutrients. This algorithm consists of five phases that include population chemotaxis, swarming, reproduction, elimination, and dispersal. Virus optimization algorithm (VOA) (Liang and Cuevas Juarez 2020) models the interaction between the harmless viruses and the immune system. The viruses are classified into two groups: common and strong. Common-type viruses are responsible to explore promising areas of solution space. Strong viruses reproduce at a higher rate to exploit points within target areas found by common viruses. In (Liang and Cuevas Juarez 2020), a self-adaptive virus optimization algorithm (SaVOA) is introduced to solve the parameter initialization issue of VOA. Magnetotactic bacteria optimization algorithm (MBOA) (Mo and Xu 2013) models the moment of magnetosomes and energy management and in magnetotactic bacteria. MBOA starts by calculating the magnetic field of cells. Then, the distance and interaction energy between cells is computed. Finally, moments of magnetosomes are tuned by integrating with cells to reach the minimum magnetostatic energy.

A particular algorithm may provide very well outcomes on a set of problems, but not on others. This issue is consistent with the no free lunch (NFL) theorem, which dictates that there is no algorithm best suited for solving all problems (Wolpert and Macready 1997). Therefore, introducing new meta-heuristics or enhancing the performance of existing ones is an active research field. In the literature, there is no research, which models the containment measures needed to control the transmission of coronavirus disease 2019 (COVID-19). This issue and the NFL theorem motivated our attempt to mathematically model the COVID-19 mitigation protocols, and propose a new swarm intelligence strategy, which is called anti-coronavirus optimization (ACVO) algorithm. ACVO simulates the protocols recommended by the world health organization (WHO) to slow down the transmission of COVID-19 diseases and improve public health.

COVID-19 is a recently discovered infectious disease that spread throughout the world (Guan 2020). It is announced as a serious pandemic by the WHO that needs international concern (Maier and Brockmann 2020; Wu et al. 2020). So far, no effective vaccine or drug has been identified for

COVID-19. Only some mitigation policies have been taken by the WHO and governments to COVID-19. These policies mainly include social distancing, quarantine, isolation, personal hygiene, wearing plastic screens and face masks in community, restricting the movement of people, and avoiding public gatherings in large groups (Anderson et al. 2020). The social distancing policy aims to keep a safe space between individuals. In the quarantine phase, people with suspected symptoms are separated from others and quarantined. In the isolation step, treatment is carried out for people having the disease to return them to normal conditions. These policies hopefully improve the health of people. The ACVO attempts to solve optimization problems by modeling three main mitigation policies including social distancing, quarantine, and isolation. The objective is to find the fittest and healthiest person in the population, which corresponds to a near-optimal solution to a given optimization problem.

Overall, the main contributions of this paper include:

- Introducing an anti-coronavirus optimization (ACVO) algorithm inspired by the measures recommended to mitigate the spread of COVID-19.
- Investigating the ability of ACVO in solving numerical and engineering optimization problems, and compare it with counterpart algorithms. The results show that the ACVO achieved superior results compared with its counterparts.

The organization of the remaining parts is as follows. Section 2 presents the inspiration source and the mathematical model of the proposed ACVO. Section 3 discusses the effectiveness of the ACVO algorithm on a variety of test optimization problems. Section 4 makes some conclusions and provides some future research directions.

2 Anti-coronavirus optimization (ACVO) algorithm

This section presents the inspiration source of the proposed ACVO algorithm and its mathematical model. Then, a simple numerical example is given to illustrate the functioning of ACVO on a continuous optimization problem.

2.1 Inspiration source

COVID-19 is an emerging contagious disease caused by SARS-CoV-2 virus. It infects people of all ages, especially those with preexisting medical conditions. According to the report published by WHO, the reproductive number (R_0) of COVID-19 is about 2.5 (Liu et al. 2020). It means that each infected person is infecting about 1 to 3 people on average. As shown in Fig. 1, the number of coronavirus cases is growing exponentially.

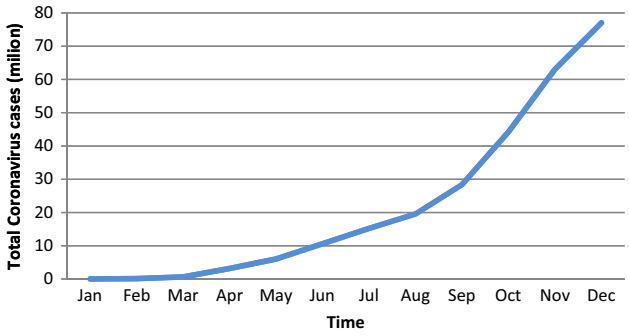


Fig. 1 The total number of confirmed cases of COVID-19 from 1 January 2020 to 20 December 2020 (The data for draw charts are driven from "coronavirus worldwide graphs" [https://www.worldometers.info, last access 20 Dec 2020]

There are two main approaches to spread the coronavirus: breathing and close contact. In the former approach, the virus spreads by respiratory droplets generated when an infected person sneezes or coughs. The droplets generated in the respiratory system are large and heavy that cannot be suspended in the air for a long time, and deposit about 1 to 1.5 meters.[1] The transmission capacity of COVID-19 is less than 1.8 meters (6ft). In the close contacting approach, the virus can transmit from infected individuals to healthy people and infect them through close communication and physical contact. In the other form of close contact, people may be infected by touching the objects or surfaces that contain the virus, and then touching their mouth, nose or eyes.

To date, there is no effective treatment for COVID-19 disease. The WHO recommends that containment measures including social distancing, quarantine, and isolation can significantly diminish the spread of COVID-19 by removing infectious individuals from the transmission process (Maier and Brockmann 2020), (Anderson et al. 2020).

Social distancing means reducing closeness and frequency of contact between individuals to slow down the spread of a contagious disease. The safe distance recommended by the authorities varies. WHO adopted a 1m (3.3ft) physical distancing policy.[2] China, Singapore, Lithuania, and Hong Kong recommended that 1m (3.3ft) is safe. The USA adopted 1.8m (6ft), and Canada recommended 2m (6.6ft) physical distancing. Because of the nature of COVID-19, if the physical distance between people is more than a safe space, the probability of spreading the virus through breathing and close

[1] Coronavirus disease (COVID-19) advice for the public: WHO, [https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public, last access 20 Dec 2020].

[2] Social Distancing, [https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/social-distancing.html, last access 20 Dec 2020].

(a)

(b)

Fig. 2 The effect of social distancing on the spread of the COVID-19; **a** reducing social exposure by 50%; **b** reducing social exposure by 75%

contacting decreases. As the outbreak slowdowns, safe distance decreases, because other methods can be adopted to mitigate virus transmission, for example, face masks. Recent studies show that social distancing is one of the best tools to decrease the spread of the COVID-19 (Anderson et al. 2020), (Matrajt and Leung 2020). Figure 2 shows the impact of social distancing on the spread of the diseases. Fig. 2a, b shows the number of infected people when they reduce social exposure by 50 and 75%, respectively. Without social distancing, 406 people will be infected in 30 days when R_0 is 2.5.[3]

The quarantine is another efficient approach to curb the spread of the disease. A person should be quarantined if she/he was within 6 feet of someone who has COVID-19 for a total of 15 minutes or more, drops of sneezing or coughing of an infected individual falls on the person, had direct physical contact with the person, provided care at home to someone who is sick with COVID-19, and shared eating or drinking utensils. In the quarantine phase, the movement of people whose disease has not been definitely diagnosed but may have potentially been exposed to COVID-19 is restricted. Individuals in quarantine should separate themselves from the population, stay at home, monitor their health, and follow health directions. Health experts recommended that quarantine should last for 2–14 days because the time between exposure to the virus and the onset of symptoms is between 2 and 14 days. More recent estimations found that the mean incubation period was 4.2 days on average; however, it varies greatly among infected individuals (Sanche et al. 2020).

The infected individuals that have a confirmed medical diagnosis are isolated from the healthy population to protect the public from exposure to the virus (Swanson and Jeanes 2007). In isolation, infected people who are at higher risk for severe illness should be monitored and cared for. There are various forms of isolation that periodically revise according to outbreak time and environmental conditions. Some popular methods include contact isolation, respiratory isolation, reverse isolation, self-isolation, and strict isolation. Researchers are working hard to quickly find an effective treatment for COVID-19. A few methods are currently being investigated in trials as a potential therapy for COVID-19. One of the potential methods is convalescent plasma therapy (Ye 2020), in which the plasma from people who have recovered transfer into the infected ones. The main principle of this method is based on the fact that people who have recovered from the disease have antibodies to the virus in their blood. It is important to notice that isolation is different from quarantine. Isolation separates infected individuals from healthy individuals, while quarantine separates and restricts the movement of individuals who are exposed to a virus to check if they become sick. The empirical studies show that the proportion of individuals with COVID-19 that need to be isolated and admitted to the intensive care unit (ICU) was 6% (Li 2020). The mean length of hospitalization among the recovered individuals often was in the range of 10–13 days (Li 2020; Wang 2020). For asymptomatic cases, discharging patients from isolation is 10 days after symptom onset, and for symptomatic cases, discharging from isolation

[3] https://www.visualcapitalist.com/the-math-behind-social-distancing/.

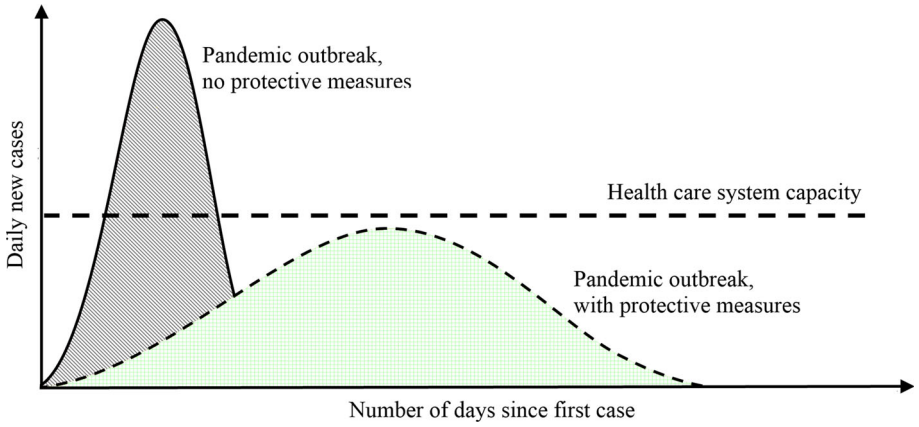


Fig. 3 Curve of COVID-19 pandemic outbreak with/ without protective measures

is 10 days after symptom onset plus 3 additional days without the disease's symptoms.[4]

Figure 3 shows the impact of adopting the containment protocols to slow down the spread of COVID-19. Without pandemic containment measures, the virus can spread exponentially (Maier and Brockmann 2020). Following the mitigation measures, many governments have been able to control the coronavirus and push effective reproductive number (R_e) to below 1 (Sanche et al. 2020).

2.2 Mathematical model

Anti-coronavirus optimization (ACVO) algorithm is a mathematical modeling of the containment measures developed to mitigate the spread of the COVID-19. The working principle of ACVO is shown in Fig. 4. ACVO starts its work with the initialization of parameters and a population of solutions. Then, the population is updated using three operators: social distancing, quarantine, and isolation. If termination conditions are met, the algorithm terminates, and reports the best solution; otherwise continues the population updating process. In modeling the algorithm, the following assumptions are considered:

- The basis of this algorithm is to keep all members of society healthy, recover infected people and improve their health.
- There is no concept of death in this algorithm. No one dies in this algorithm.
- It is assumed that everyone will be healthy at the end of the pandemic.
- Most of the parameters are configured according to the values recommended by WHO and other organizations involved in disease control.

[4] WHO, clinical management of COVID-19 (Interim Guidance) [https://www.who.int/publications-detail/clinical-management-of-covid-19, last access 20 December 2020].

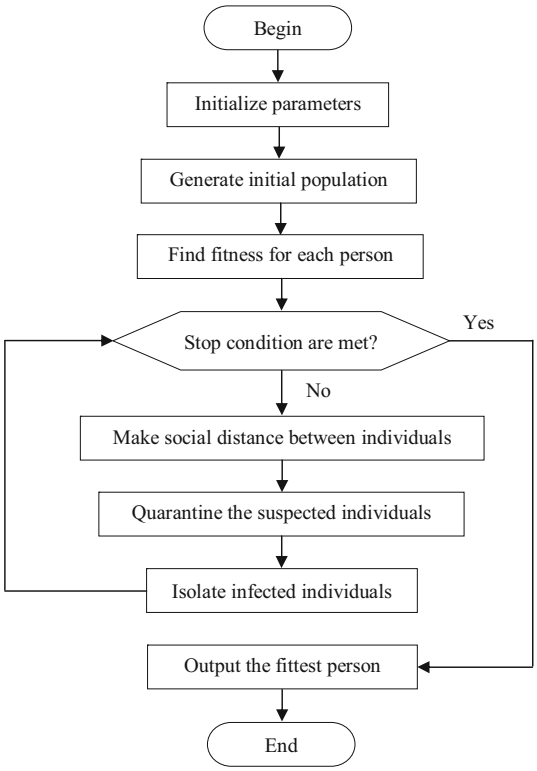


Fig. 4 Flowchart of the ACVO algorithm

2.2.1 Population initialization

The ACVO in each generation works with a randomly distributed population P, defined as follows:

$$P = [P_1, P_2, ..., P_N] \tag{1}$$

Each solution $P_i \in P$ in the population is called a person, which shows a point in a multidimensional space. It is encoded as a set of real-valued variables and an integer variable indicating the health status of the person.

$$P_i = [p_{i1}, p_{i2}, ..., p_{iD}, s] \tag{2}$$

where p_{ij} consists of a possible value for jth variable, and s is the health status of the person that is defined as

$$s = \begin{cases} 1 & \text{if } P_i \text{ is healthy} \\ 0 & \text{if } P_i \text{ is in quarantine} \\ -1 & \text{if } P_i \text{ is in isolation} \end{cases} \tag{3}$$

each variable p_{ij} is initialized as follows:

$$p_{ij} = (p_{\max j} - p_{\min j}) \times r + p_{\min j} \tag{4}$$

where $r \in [0, 1]$ is a uniformly distributed random number, $p_{\min j}$ and $p_{\max j}$ are lower and upper bounds of P_i in jth dimension, respectively.

The individuals pass to a fitness function f to assess their quality. In ACVO, it is assumed that individuals' health is proportionate to their fitness. The fitness function assigns higher fitness to healthy individuals and less fitness to weak and infected individuals.

2.2.2 Social distancing

This operator simulates the social distancing policy. It is obvious that not all people engage in social distancing at any given time, but only a proportion of the population follows this policy. As the disease spreads, the importance of social distancing becomes more apparent and people are forced to be more observant. To mathematically model this issue, at each iteration, m number of people are selected to engage in social distancing to decrease the inter-personal contacts. The algorithm attempts to guide the individuals into a safe and promising area in the solution space. Here, it is assumed that the best individual of the population is in the promising and optimal place. So, the algorithm tries to lead population toward the best individual. The algorithm updates the position of each selected person P_i as follows:

$$P_i^{t+1} = P_i^t + \Delta_1^t + \Delta_2^t \tag{5}$$

where P_i^t denotes the position of ith person, and t is the current iteration. Δ_1 controls the local distancing between P_i and any other individual in its vicinity. Δ_2 controls the global distancing between P_i and the best individual P^*, which guides P_i into a promising position outside the current area. The objective of Δ_1 and Δ_2 is to move the individual P_i into a promising area in solution space. Δ_1 is defined as follows:

$$\Delta_1^t = \alpha_{ij}^t \times sd_{ij}^t \times U(-1, 1) \tag{6}$$

where $U(-1, +1)$ is a uniform random number generator that generates +1 or -1. The function $U(-1, +1)$ is considered to make individual P_i move better in different directions. sd_{ij}^t shows the minimum physical distance that P_i and each other individual P_j in the population should be observed to prevent the infection. sd_{ij}^t is defined as

$$sd_{ij}^t = \begin{cases} \Delta - d_{ij}^t & \text{if}(d_{ij}^t < \Delta) \\ d_{ij}^t & \text{if}(d_{ij}^t \geq \Delta) \end{cases} \quad \text{where} \quad d_{ij}^t = \left\| P_i^t - P_j^t \right\| \tag{7}$$

d_{ij}^t indicates the current distance between P_i and P_j. Δ is the pre-defined safe physical distance between people at iteration t according to the mitigation protocols. Since the safe physical distance may differ in various conditions and from time to time, Δ can be customized according to the epidemic conditions. α_{ij}^t is the infection effect of person P_j on P_i, defined as

$$\alpha_{ij}^t = e^{-\left(\frac{d_{ij}^t}{\Delta}\right)} \tag{8}$$

The larger α_{ij}^t is, the stronger the infection imposed on the individual P_i. As the algorithm proceeds, the physical distance between individuals increases; hence, the disease transmission between individuals decreases, and the outbreak slowdowns.

Δ_2^t is defined as follows:

$$\Delta_2^t = \beta_{ij}^t \times V \times (P^* - P_i^t) \tag{9}$$

where β_{ij}^t is the infection effect of P^* on P_i. V is the step size, which adjusts the amount of movement of P_i toward P^* with different distance steps. P^* is the best solution found until current iteration. V is computed by Levy distribution (Al-Betar et al. 2012) as follows:

$$V \sim \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, \quad s \gg s_0 > 0, \quad s_0 = 0.1 \tag{10}$$

$\Gamma(\lambda)$ denotes the standard gamma distribution, where λ is set to be 1.5. The variable s is defined as

$$s = \frac{U}{|V|^{1/\lambda}}, \qquad U \sim N(0, \delta^2), \quad V \sim N(0, 1) \tag{11}$$

where U and V are Gaussian distributions, $N(0, \delta^2)$ indicates the normal distribution with mean 0 and variance δ^2, and $N(0, 1)$ is the standard normal distribution.

$$\beta_{ij}^t = e^{-\left(\frac{\left\| P^* - P_i^t \right\|}{\Delta}\right)} \tag{12}$$

The idea behind Eq. (9) is to direct the other individuals toward the best individual P^*, which is the leader in following the virus mitigation protocols.

2.2.3 Quarantine

The quarantine operator simulates the actions that each suspected person with COVID-19 should take during the quarantine phase. In the ACVO, the people suspected of having the disease are those who obtain low fitness during the optimization process. To determine the suspected people, first, the individuals are sorted based on their fitness in ascending order. Here, it is assumed that the problem is maximization, and the fitness value indicates the health condition of a person. In minimization problems, the individuals will be sorted in descending order, because the infected ones will have low fitness. Then, the q number of the weakest individuals are selected to form the quarantine list $Q = \{P_1, P_2, ..., P_q\}$. This list is composed of suspected individuals that should be quarantined to determine if they were infected or not. The parameter q at each iteration t is computed as

$$q^t = \left\lceil \left((1 - (1 - (\lambda^t)^2) m^t \right) R_0 \right\rceil \tag{13}$$

where q^t is the effective reproductive number at iteration t. It counts the number of suspected individuals that may have potentially been exposed to diseases. m^t is the fraction of the population that takes part in social distancing to mitigate their interpersonal contacts to a fraction of λ of normal conditions. R_0 is the basic reproductive number. In a population where everyone is equally susceptible to a disease, R_0 indicates the average number of secondary infected cases caused by one primary infected person. At the time of writing this paper, the R_0 of the COVID -19 is estimated to be 1.4–2.5 (Anderson et al. 2020). If we assume R_0 to be 2.5, and $m = 25\%$ of the population take part in social distancing to decrease their interpersonal relations to $\lambda = 50\%$ of normal contacts, then $q^t \approx 3$. Under this condition, a single infected person would generate an average of 3 new secondary infected cases. If $q > 1$, the number of infected individuals will increase, and where $q < 1$ the number of infected cases decreases. To successfully eliminate COVID-19 from the population, q needs to be less than 1. To realize this goal, the algorithm should increase the value of m, and simultaneously decrease λ as it proceeds. The following equations are proposed to control the values of m and λ:

$$\lambda^t = 1 - \frac{t}{T} \tag{14}$$

$$m^t = \frac{t}{T} \tag{15}$$

where T is the maximum number of generations.

The algorithm randomly selects k_1 variables from each suspected person $P_i \in Q$, and updates them by Eq. (17). To select the variables, the algorithm takes as input the solution dimension D and k_1 and then generates k_1 pseudo-random integers, $\{r_1, r_2, ..., r_{k_1}\}$ using the discrete uniform distribution on the interval $[1, D]$. Each random number r_j shows the position of a variable $p_{ij} \in P_i$ that to be selected and updated. k_1 is calculated as

$$k_1 = \lceil r_1 \times D \rceil \tag{16}$$

where r_1 is a small uniform random number. The selected variables are updated as follows:

$$p_{ik}^{t+1} = p_{ik}^t + U(-1, +1) \times rand \tag{17}$$

$rand$ is a random number in the range $[0, 1]$. The suspected individuals should quarantine for q_d days and their variables are updated with Eq. (17). After the end of the quarantine period, if the fitness of P_i is greater than or equal to its fitness on the first day of quarantine, then P_i is recognized as healthy and returns to the population; otherwise, it must be isolated and hospitalized. The following equation formulates this issue:

$$\begin{cases} P^{t+1} = \{P^t\} \cup \{P_i^t\} \text{ if} (f_i^{q_e} \geq f_i^{q_s}) \\ I^{t+1} = \{I^t\} \cup \{P_i^t\} \qquad else \end{cases} \tag{18}$$

where P is the population, and I is the isolation list. $f_i^{q_s}$ and $f_i^{q_e}$ show the fitness of individual P_i at the first and last day of quarantine, respectively. q_s and q_e are the iteration number at the first and last day of the quarantine, respectively. q_e is equal to $q_s + q_d$, where q_d is the duration of quarantine. q_d is initialized according to the WHO recommendations.

2.2.4 Isolation

The isolation operator simulates the health measures taken to treat infected individuals and get back them to normal conditions. In the current version of ACVO, the convalescent plasma therapy is modeled to treat infected individuals. In this way, some characteristics of the fittest healthy person are injected into the infected individuals. To simulate this process, the algorithm randomly selects k_2 variables from each isolated person $P_i \in I$, and updates them by Eq. (20). k_2 is computed as

$$k_2 = \lceil r_2 \times D \rceil \tag{19}$$

where r_2 is a uniform random number, regenerated every iteration. Since the variable r_2 regenerates at every iteration and varies for each solution P_i, the number of variables

selected from each solution is different from other solutions. The selected variables are updated using the following equation:

$$p_{ij}^{t+1} = \frac{1}{2}\left(p_{ij}^{t} + (\gamma \times p_j^{*t})\right) \tag{20}$$

where p_{ij}^{t} shows the jth element of P_i at iteration t, p_j^{*t} shows the jth element of fittest individual P^*. γ is a scaling factor, which measures the improvement effect of p_j^{*t} on p_{ij}^{t}. Eq. (20) simply simulates the convalescent plasma therapy by combining the variables of the best-fit individual P^* with the corresponding components of the infected person P_i. In the early days of hospitalization, more care is taken, while in the last days, drugs and care are reduced. Therefore, it makes sense to reduce the coefficient γ over isolation time. γ is defined as follows:

$$\gamma^{v} = 1 - \frac{v}{h_d} \tag{21}$$

v ranges from 1 to h_d, where h_d is the maximum duration of isolation. The infected individuals are isolated for K_2 days. After the end of the hospitalization period, if the person's fitness is greater than or equal to his/her fitness on the first day of isolation, the person is recognized as healthy and returns to the population; otherwise, care and hospitalization should continue. The following equation formulates this issue:

$$\begin{cases} P^{t+1} = \{P^t\} \cup \{P_i^t\} \; if (f_i^{h_e} \geq f_i^{q_s}) \\ I^{t+1} = \{I^t\} \cup \{P_i^t\} \qquad else \end{cases} \tag{22}$$

$f_i^{h_e}$ is the fitness of individual P_i at the last day of hospitalization, and $f_i^{q_s}$ is the fitness of P_i at the first day of quarantine. h_e is the iteration number at the last day of the isolation. h_e is equal to $h_s + h_d$, where h_s is the iteration number at the first day of the isolation. h_d is configured according to the WHO recommendations.

2.2.5 Selecting the best solution

Until termination conditions are not met three operators—social distancing, quarantine, and isolation—are applied to update the population. Finally, an individual that obtained the best fitness will introduce as the optimal solution for the problem. The algorithm terminates when the status of all individuals to be healthy (s=1), or the maximum number of function evaluations (NFEs) reaches to a predefined value.

Algorithm 1 summarizes the pseudo-code of the ACVO. In Sect. 3, we will present intuitive reasoning of why the ACVO algorithm is an efficient optimization algorithm.

Algorithm 1: Pseudo code of the ACVO algorithm

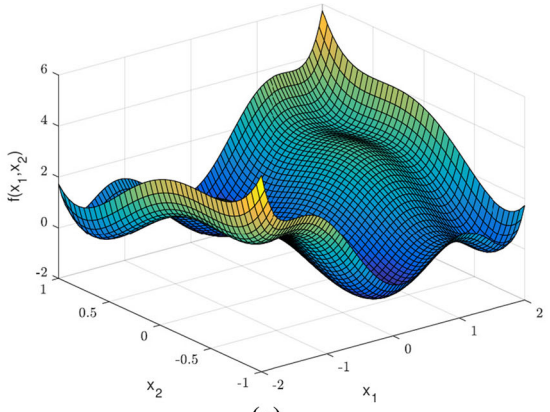
```
Initialize algorithm's parameters;
Create an initial population by Eq. (4);
Calculate the fitness of each individual;
while (t < MaxFES) do
    for (i = 1:m) do
        for (j = i+1:m) do
            if (d_ij^t < Δ) then
                Compute sd_ij by Eq. (7);
                Calculate local distance by Eq. (6);
                Calculate global distance by Eq. (9);
                Update position of P_i by Eq. (5);
            end
        end
    end
    Identify suspected people to form list Q;
    for i=1:|Q| do
        for j=1:q_d do
            Update each suspected individual P_i ∈ Q by Eq. (17);
            Evaluate each individual P_i;
            Update q^t by Eq. (13);
        end
        if (f_i^{q_e} ≥ f_i^{q_s}) then
            P^{t+1} = {P^t} ∪ {P_i^t};
        else
            I^{t+1} = {I^t} ∪ {P_i^t};
        end
    end
    for i=1:|I| do
        for (j=1:h_q) do
            Update each isolated individual P_i ∈ I by Eq. (20);
            Evaluate each isolated individual P_i;
            Update coefficient γ by Eq. (21);
        end
        if (f_i^{h_e} ≥ f_i^{q_s}) then
            P^{t+1} = {P^t} ∪ {P_i^t};
        else
            I^{t+1} = {I^t} ∪ {P_i^t};
        end
    end
    Update the current fittest individual P*;
end
return P*;
```

2.3 A numerical example

The following numerical problem is used to illustrate how the ACVO algorithm updates the population and finds optimal solutions.

$$f(\mathbf{x}) = f(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \quad -5 \leq x_1, x_2 \leq 5$$

This function is called six-hump camel. Its global minimum is -1.0316 at $(x_1, x_2) = (0.0898, -0.7126)$. Fig. 5a shows the 3D plot of this function. Figure 5b–f shows the



(a)

Fig. 5 A numerical example to illustrate the functioning of ACVO

t	x_1	x_2	$f(x)$	s
Initial	-1.45E+00	1.91E+00	3.76E+01	1
	-2.84E+00	1.09E+00	6.82E+01	1
	-2.86E+00	-3.15E-01	7.56E+01	1
	-2.00E+00	-3.13E+00	3.53E+02	1
	-3.71E+00	3.23E+00	9.06E+02	1
	-3.17E+00	-3.81E+00	9.65E+02	1
	6.41E-01	4.22E+00	1.20E+03	1
	3.64E+00	4.33E+00	1.80E+03	1

(b)

t	x_1	x_2	$f(x)$	s
1	-1.14E+00	1.11E+00	2.28E+00	1
	-2.34E+00	-1.59E-02	1.39E+01	1
	-1.48E+00	1.69E+00	2.08E+01	1
	-1.45E+00	1.91E+00	3.76E+01	1
	-2.84E+00	1.09E+00	6.82E+01	1
	-2.86E+00	-3.15E-01	7.56E+01	1
	-2.10E+00	2.86E+00	3.47E+02	0
	-2.00E+00	4.11E+00	3.53E+02	0

(c)

t	x_1	x_2	$f(x)$	s
2	-1.13E+00	9.81E-01	1.12E+00	1
	-1.14E+00	1.11E+00	2.28E+00	1
	-1.83E+00	-2.14E-01	2.58E+00	1
	-2.34E+00	-1.59E-02	1.39E+01	1
	-1.40E+00	-8.03E-01	1.76E+01	0
	-1.48E+00	1.62E+00	2.08E+01	0
	2.76E+00	2.86E+00	3.47E+02	0
	-2.00E+00	-2.39E+00	3.53E+02	0

(d)

t	x_1	x_2	$f(x)$	s
3	-8.07E-01	8.31E-01	2.80E-01	1
	-1.13E+00	9.81E-01	1.12E+00	1
	-2.00E+00	-2.39E+00	1.16E+02	1
	2.76E+00	2.86E+00	2.98E+02	1
	-1.31E+00	3.81E-01	1.36E+00	0
	3.50E+00	-8.03E-01	1.76E+01	0
	-1.48E+00	-2.15E+00	2.08E+01	0
	-1.14E+00	-7.47E-01	2.28E+00	0

(e)

t	x_1	x_2	$f(x)$	s
4	-7.35E-01	7.69E-01	6.91E-02	1
	-8.07E-01	8.31E-01	2.80E-01	1
	-1.13E+00	-2.66E+00	1.12E+00	0
	-1.31E+00	-2.90E+00	1.36E+00	0
	3.40E-01	-7.47E-01	2.28E+00	0
	-1.29E+00	6.08E-01	2.56E+00	0
	-1.63E+00	-2.15E+00	7.24E+01	-1
	3.50E+00	-6.47E-01	3.47E+02	-1

(f)

t	x_1	x_2	$f(x)$	s
5	3.40E-01	-7.47E-01	-8.06E-01	1
	-7.35E-01	7.69E-01	6.91E-02	1
	-8.07E-01	8.31E-01	2.80E-01	1
	-1.63E+00	-1.34E+00	9.91E+00	1
	-1.13E+00	4.64E+00	1.12E+00	0
	-1.29E+00	2.86E+00	2.56E+00	0
	-1.44E+00	-2.90E+00	2.57E+02	-1
	3.50E+00	-3.37E-01	3.49E+02	-1

(g)

population updating mechanism of ACVO in five generations on test function. Without loss of generality, for simplicity and space saving, it is assumed that individuals reduced their close contact to $\lambda = 0.75$, and the quarantine and isolation period are set to be 2 and 3 days, respectively. As shown in Fig. 5, ACVO improves the individuals at each iteration by social distancing, quarantine, and isolation operators. Individuals are gradually moving toward the optimal solution.

3 Experiment and analysis

To evaluate the performance of the proposed ACVO algorithm, seven well-established optimization algorithms are adopted to make a comparison. These algorithms are WOA (Mirjalili and Lewis 2016), ABC (Karaboga and Basturk 2007), SADE (Qin and Suganthan 2005), PSOGSA (Mirjalili and Hashim 2010), TLBO (Rao et al. 2012), HHO (Heidari et al. 2019), and HGSA (Wang et al. 2019). The ABC, TLBO,

WOA, and HHO are popular swarm intelligence algorithms, and the SADE and HGSA are evolutionary algorithms that have been introduced recently and obtained the best results on numerical optimization problems.

3.1 Benchmark problems

Tables 1 and 2 summarize the characteristics of CEC2018 (Suganthan et al. 2018) and CEC2019 (Price et al. 2018) test functions. The test suite CEC2018 contains 28 single objective minimization problems: three unimodal (U), seven multimodal (M), ten hybrid (H), and eight composition (C) functions. CEC2019 test set includes 10 single objective optimization functions. In test set CEC2019, functions CEC04 to CEC10 are shifted and rotated, whereas benchmark functions CEC01 to CEC03 are not. These problems are more complex, difficult, and have many local optima. Therefore, they are suitable to test the algorithms' accuracy, reliability, convergence rate, and the ability to avoid the local optima. Detailed information about these problems is given in (Suganthan et al. 2018; Price et al. 2018). To clearly show the scalability of algorithms, the CEC2018 functions with three dimensions 10, 30, and 50 are considered to assess their performance on low, medium, and large scale. The search domain for all CEC2018 test problems is $[-100, 100]^D$, where D indicates the dimension of test problems. In Tables 1, 2, F_{min} stands for the global optimum of benchmark problem.

3.2 Experimental setup

All algorithms are implemented in MATLAB language on a Laptop machine with 8GB RAM and 2.2GHz Intel(R) Core (TM) i7 CPU. Following the guidance provided in Suganthan et al. (2018), the algorithms executed 30 times independently, each time with a different population to obtain statistical results for each benchmark function. The maximum number of fitness function evaluations is initialized as $10,000 \times D$, and the population size of all the algorithms is 50.

Table 1 Characteristics of the IEEE-CEC 2018 test suite

No.	Problem	Type	F_{min}
f_1	Shifted and Rotated Bent Cigar Function	U	100
f_2	Shifted and Rotated Sum of Different Power Function	U	200
f_3	Shifted and Rotated Zakharov Function	U	300
f_4	Shifted and Rotated Rosenbrock's Function	M	400
f_5	Shifted and Rotated Rastrigin's Function	M	500
f_6	Shifted and Rotated Expanded Scaffer's F6 Function	M	600
f_7	Shifted and Rotated Lunacek Bi_Rastrigin Function	M	700
f_8	Shifted and Rotated Non-Continuous Rastrigin's Function	M	800
f_9	Shifted and Rotated Levy Function	M	900
f_{10}	Shifted and Rotated Schwefel's Function	M	1000
f_{11}	Hybrid Function 1 (K=3)*	H	1100
f_{12}	Hybrid Function 2 (K=3)	H	1200
f_{13}	Hybrid Function 3 (K=3)	H	1300
f_{14}	Hybrid Function 4 (K=4)	H	1400
f_{15}	Hybrid Function 5 (K=4)	H	1500
f_{16}	Hybrid Function 6 (K=4)	H	1600
f_{17}	Hybrid Function 6 (K=5)	H	1700
f_{18}	Hybrid Function 6 (K=5)	H	1800
f_{19}	Hybrid Function 6 (K=5)	H	1900
f_{20}	Hybrid Function 6 (K=6)	H	2000
f_{21}	Composition Function 1 (K=3)	C	2100
f_{22}	Composition Function 2 (K=3)	C	2200
f_{23}	Composition Function 3 (K=4)	C	2300
f_{24}	Composition Function 4 (K=4)	C	2400
f_{25}	Composition Function 5 (K=5)	C	2500
f_{26}	Composition Function 6 (K=5)	C	2600
f_{27}	Composition Function 7 (K=6)	C	2700
f_{28}	Composition Function 8 (K=6)	C	2800

*K indicates the number of components in function f_i

Table 2 Characteristics of the IEEE-CEC 2019 test suite

No.	Problem	D	Search Range	F_{min}
CEC01	Storn's Chebyshev Polynomial Fitting Problem	9	[−8192, 8192]	1
CEC02	Inverse Hilbert Matrix Problem	16	[−16384, 16384]	1
CEC03	Lennard-Jones Minimum Energy Cluster	18	[−4, 4]	1
CEC04	Rastrigin's Function	10	[−100,100]	1
CEC05	Griewangk's Function	10	[−100,100]	1
CEC06	Weierstrass Function	10	[−100,100]	1
CEC07	Modified Schwefel's Function	10	[−100,100]	1
CEC08	Expanded Schaffer's F6 Function	10	[−100,100]	1
CEC09	Happy Cat Function	10	[−100,100]	1
CEC10	Ackley Function	10	[−100,100]	1

Table 3 Parameter tuning of ACVO and comparison algorithms

Algorithm	Parameters
SADE (Qin and Suganthan 2005)	$p = 0.05$, $C = 0.1$, $CR = N(CR_m; 0.1)$, $F = N(0.5; 0.3)$
ABC (Karaboga and Basturk 2007)	$limit = N \times D$
PSOGSA (Mirjalili and Hashim 2010)	$K \in [n, 2]$, $w_1(t) = 0.5$, $w_2(t) = 1.5$, $\alpha = 20$, $G_0 = 100$
HHO (Heidari et al. 2019)	$\beta = 1.5$, $E_0 \in [-1, 1]$
WOA (Mirjalili and Lewis 2016)	$a \in [0, 2]$, $a_2 \in [-1, -2]$
TLBO (Rao et al. 2012)	$T = 1, 2$
HGSA (Wang et al. 2019)	$K \in [n, 2]$, $L = 100$, $G_0 = 100$, $w_1(t) = 1 - t^6/T^6$, $w_2(t) = t^6/T^6$
ACVO	$R_0 = 2.5$, $\Delta = 2$, $q_d = 5$, $h_d = 10$, $r_1, r_2 \in [0, 0.5]$

Other specific parameters of algorithms are configured following the recommended settings in the original works and summarized in Table 3.

3.3 Results

Tables 4, 5, and 6, respectively, summarize the statistical results on 10, 30, and 50D functions. The results are shown in $mean \pm std$ format, where $mean$ is the mean cost, and std is the standard deviation obtained in all simulation runs. In the tables, the best values obtained by the algorithms on each function is represented in boldface. Results demonstrate that ACVO obtains the best results on the most functions due to its high strength in finding the optimal solutions.

Table 7 shows the final and average ranks of the algorithms computed by the Friedman test (Derrac et al. 2011) according to the mean cost values. The first rank belongs to the ACVO algorithm in all dimensions. The second rank belongs to SADE in 10 and 30D functions, and HGSA in 50D functions. The third rank belongs to HHO, HGSA, and SADE, respectively, in 10, 30, and 50D functions. It can be observed that the ACVO is a competitive algorithm for numerical optimization problems with different scale.

To provide more accurate conclusions, Table 8 shows a multi-problem based pairwise statistical analysis that is computed by the Wilcoxon signed-rank test (Derrac et al. 2011) at a significance level of $\alpha = 0.05$. The computation of p-values is performed according to the *mean* values generated through 30 runs. The results show that the ACVO performs the best among counterpart algorithms on test problems with three dimensions 10, 30 and 50. This highlights that ACVO can efficiently optimize the test problems with diverse dimensions. The p-values obtained in D= 30 and 50 show that ACVO is an effective method in optimizing medium and large dimension problems.

Figure 6 shows the algorithms' aggregate performance on test problems. The horizontal axis shows the kth rank of algorithms. The vertical axis represents the number of functions on which the algorithm gained the kth rank based on the Wilcoxon signed-rank test. As shown in Fig 6, the ACVO algorithm is the top-1 performer among most functions. This shows that the solution quality of the ACVO is better than its counterparts. The second rank belongs to SADE on 10 and 30D functions, and HGSA on 50D functions. The third rank on 10, 30 and 50D functions, respectively, belongs to HHO, HGSA and SADE.

Tables 9 and 10 list the unadjusted and adjusted p-values (Derrac et al. 2011) obtained by counterpart algorithms in comparison with the ACVO on benchmarks. The computation is performed according to the *mean* costvalues. The results demonstrate that ACVO performs well on most func-

Table 4 Results obtained by the ACVO and comparison algorithms on test functions, when $D = 10$

Algorithm	f_1	f_2	f_3	f_4	f_5	f_6	f_7
SADE (Qin and Suganthan 2005)	1.19E+02 ± 7.09E+01	3.87E+04 ± 2.11E+03	**3.00E+02 ± 0.00E+00**	4.00E+02 ± 9.43E-13	5.29E+02 ± 1.05E+01	6.01E+02 ± 1.24E+00	7.27E+02 ± 6.30E+00
ABC (Karaboga and Basturk 2007)	1.10E+02 ± 2.50E+01	4.96E+02 ± 3.75E+02	3.22E+02 ± 1.77E+00	4.74E+02 ± 1.64E+01	**5.09E+02 ± 6.95E+00**	6.01E+02 ± 6.41E+00	7.91E+02 ± 8.88E+00
PSOGSA (Mirjalili and Hashim 2010)	1.83E+03 ± 2.06E+02	8.87E+04 ± 4.76E+05	3.00E+02 ± 1.83E-14	4.03E+02 ± 5.56E+00	5.22E+02 ± 7.55E+00	6.02E+02 ± 3.48E+00	7.34E+02 ± 8.29E+00
TLBO (Rao et al. 2012)	3.68E+02 ± 2.17E+02	4.56E+02 ± 3.13E+03	3.70E+03 ± 2.62E+02	4.94E+02 ± 1.97E+02	5.87E+02 ± 1.42E+01	6.37E+02 ± 1.87E+01	7.56E+02 ± 6.82E+01
WOA (Mirjalili and Lewis 2016)	2.40E+03 ± 1.62E+02	4.17E+03 ± 3.17E+03	1.98E+03 ± 1.99E+03	4.30E+02 ± 1.99E+01	5.32E+02 ± 3.17E+00	6.11E+02 ± 4.39E+00	7.16E+02 ± 1.39E+00
HHO (Heidari et al. 2019)	1.03E+02 ± 1.81E+02	**3.65E+02 ± 1.16E+02**	**3.00E+02 ± 5.12E+00**	4.27E+02 ± 4.42E+01	5.73E+02 ± 2.23E+01	6.01E+02 ± 1.51E+01	7.77E+02 ± 1.91E+01
HGSA (Wang et al. 2019)	1.80E+03 ± 6.09E-04	4.72E+02 ± 6.85E+02	3.00E+02 ± 1.18E-08	4.00E+02 ± 4.33E-02	5.18E+02 ± 3.84E+00	6.00E+02 ± 1.77E-01	**7.12E+02 ± 1.04E+00**
ACVO	**1.02E+02 ± 2.40E+00**	3.22E+03 ± 1.17E+03	**3.00E+02 ± 0.00E+00**	**4.00E+02 ± 2.61E-15**	5.16E+02 ± 1.61E+00	**6.00E+02 ± 0.00E+00**	7.23E+02 ± 1.96E+00
	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
SADE (Qin and Suganthan 2005)	8.20E+02 ± 1.07E+01	**9.00E+02 ± 0.00E+00**	**1.65E+03 ± 1.07E+02**	1.12E+03 ± 2.24E+01	1.49E+03 ± 1.63E+02	**1.34E+03 ± 1.56E+01**	1.46E+03 ± 3.10E+01
ABC (Karaboga and Basturk 2007)	8.46E+02 ± 6.52E+00	1.20E+03 ± 6.86E+01	1.92E+03 ± 2.27E+02	1.17E+03 ± 8.92E+01	1.87E+03 ± 8.22E+03	6.24E+03 ± 5.42E+03	1.54E+03 ± 1.37E+01
PSOGSA (Mirjalili and Hashim 2010)	8.21E+02 ± 7.90E+00	1.08E+03 ± 3.93E+02	1.72E+03 ± 3.48E+02	1.12E+03 ± 1.41E+01	1.72E+04 ± 1.26E+04	9.06E+03 ± 6.93E+03	2.05E+03 ± 4.73E+02
TLBO (Rao et al. 2012)	8.72E+02 ± 1.20E+01	1.09E+03 ± 2.85E+02	2.33E+03 ± 4.71E+02	1.26E+03 ± 2.60E+02	1.91E+03 ± 3.23E+03	2.29E+03 ± 2.13E+05	1.80E+03 ± 3.48E+02
WOA (Mirjalili and Lewis 2016)	8.23E+02 ± 2.53E+00	**9.00E+02 ± 0.00E+00**	1.88E+03 ± 1.13E+02	1.16E+03 ± 1.88E+01	1.52E+03 ± 2.77E+02	1.65E+03 ± 1.06E+03	1.40E+03 ± 1.53E+03
HHO (Heidari et al. 2019)	**8.01E+02 ± 1.83E+01**	9.08E+02 ± 1.78E+02	1.80E+03 ± 1.08E+02	1.22E+03 ± 5.43E+01	1.69E+03 ± 5.97E+02	1.95E+03 ± 9.12E+01	1.51E+03 ± 3.34E+01
HGSA (Wang et al. 2019)	8.17E+02 ± 2.65E+00	9.00E+02 ± 1.45E-09	2.26E+03 ± 2.02E+02	1.12E+03 ± 7.38E+00	8.70E+03 ± 3.22E+03	6.73E+03 ± 1.56E+03	4.39E+03 ± 9.68E+02
ACVO	8.23E+02 ± 2.07E+01	**9.00E+02 ± 0.00E+00**	1.74E+03 ± 2.83E+02	**1.10E+03 ± 7.24E-01**	**1.43E+03 ± 7.66E+01**	2.30E+03 ± 2.78E+00	**1.40E+03 ± 8.32E-01**

Table 4 continued

	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}
SADE (Qin and Suganthan 2005)	1.50E+03 ± 8.66E+01	1.74E+03 ± 1.57E+02	1.77E+03 ± 2.38E+01	1.80E+03 ± 8.34E+01	2.54E+03 ± 3.19E+01	**2.01E+03 ± 6.49E+01**	2.33E+03 ± 1.07E+01
ABC (Karaboga and Basturk 2007)	2.20E+03 ± 3.66E+02	1.61E+03 ± 6.23E+01	1.80E+03 ± 2.44E+01	9.49E+03 ± 7.07E+02	2.73E+03 ± 7.82E+00	2.14E+03 ± 6.83E+00	2.27E+03 ± 7.62E+01
PSOGSA (Mirjalili and Hashim 2010)	4.72E+03 ± 4.83E+03	1.78E+03 ± 1.23E+02	1.77E+03 ± 4.88E+01	5.54E+03 ± 3.79E+03	5.56E+03 ± 3.57E+03	2.10E+03 ± 5.90E+01	2.29E+03 ± 5.18E+01
TLBO (Rao et al. 2012)	1.84E+03 ± 2.92E+03	1.97E+03 ± 5.86E+01	1.85E+03 ± 2.88E+01	2.24E+06 ± 1.93E+06	2.23E+03 ± 9.50E+02	2.14E+03 ± 1.54E+01	**2.23E+03 ± 1.95E+01**
WOA (Mirjalili and Lewis 2016)	2.81E+03 ± 9.73E+02	1.61E+03 ± 1.31E+02	1.77E+03 ± 2.47E+01	5.78E+04 ± 1.76E+04	2.73E+03 ± 1.36E+03	2.10E+03 ± 6.59E+01	2.30E+03 ± 5.69E+01
HHO (Heidari et al. 2019)	2.98E+03 ± 1.94E+03	1.76E+03 ± 1.01E+02	1.78E+03 ± 1.67E+01	1.82E+03 ± 8.55E+01	**2.06E+03 ± 1.43E+02**	2.02E+03 ± 4.19E+01	2.26E+03 ± 6.64E+01
HGSA (Wang et al. 2019)	3.56E+03 ± 1.29E+03	1.98E+03 ± 7.42E+01	1.76E+03 ± 1.28E+01	5.48E+03 ± 2.34E+03	5.86E+03 ± 1.87E+03	2.15E+03 ± 4.40E+00	2.31E+03 ± 4.33E+01
ACVO	**1.50E+03 ± 2.25E+00**	**1.60E+03 ± 2.57E-01**	**1.71E+03 ± 8.03E+00**	**1.80E+03 ± 1.09E+00**	2.63E+03 ± 6.10E-02	2.07E+03 ± 1.40E-01	2.31E+03 ± 3.26E+00
	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}	f_{28}
SADE (Qin and Suganthan 2005)	2.30E+03 ± 1.41E+00	2.63E+03 ± 1.25E+01	2.76E+03 ± 1.73E+01	2.94E+03 ± 3.03E+01	2.92E+03 ± 6.54E+02	3.12E+03 ± 4.64E+01	3.33E+03 ± 1.31E+01
ABC (Karaboga and Basturk 2007)	2.38E+03 ± 8.80E+01	2.65E+03 ± 5.02E+00	2.75E+03 ± 6.46E+01	2.99E+03 ± 1.63E+01	3.12E+03 ± 8.88E+01	3.10E+03 ± 3.58E+00	3.27E+03 ± 4.16E+01
PSOGSA [58]	2.30E+03 ± 6.14E+00	2.63E+03 ± 1.57E+01	2.72E+03 ± 8.71E+01	**2.92E+03 ± 2.41E+01**	2.96E+03 ± 2.22E+02	3.12E+03 ± 2.44E+01	3.30E+03 ± 1.85E+02
TLBO (Rao et al. 2012)	2.45E+03 ± 1.42E+02	2.69E+03 ± 7.14E+00	2.73E+03 ± 3.41E+01	2.97E+03 ± 8.26E+01	2.96E+03 ± 2.22E+02	3.15E+03 ± 2.72E+01	3.31E+03 ± 8.29E+01
WOA (Mirjalili and Lewis 2016)	2.33E+03 ± 7.15E+00	2.64E+03 ± 5.40E+00	2.77E+03 ± 6.57E+00	2.94E+03 ± 1.91E+01	3.38E+03 ± 5.09E+02	**3.10E+03 ± 6.36E-01**	3.37E+03 ± 8.35E+01
HHO (Heidari et al. 2019)	2.30E+03 ± 1.96E+01	2.64E+03 ± 2.19E+01	2.78E+03 ± 2.56E+01	2.94E+03 ± 2.36E+01	3.48E+03 ± 6.81E+02	3.12E+03 ± 2.53E+01	3.27E+03 ± 9.86E+01
HGSA (Wang et al. 2019)	**2.30E+03 ± 5.23E-02**	2.64E+03 ± 7.89E+00	**2.52E+03 ± 6.20E+01**	2.94E+03 ± 8.28E+00	**2.82E+03 ± 4.07E+01**	3.10E+03 ± 4.54E+00	3.38E+03 ± 2.14E+01
ACVO	2.30E+03 ± 3.09E-01	**2.61E+03 ± 3.38E+00**	2.73E+03 ± 4.51E+00	2.92E+03 ± 2.45E+01	2.90E+03 ± 0.00E+00	3.11E+03 ± 7.15E-01	**3.16E+03 ± 1.39E+01**

Bold values indicate the best results generated by the algorithms

Table 5 Results obtained by the ACVO and comparison algorithms on test functions, when $D = 30$

Algorithm	f_1	f_2	f_3	f_4	f_5	f_6	f_7
SADE (Qin and Suganthan 2005)	1.96E+03 ± 1.75E+03	4.11E+02 ± 3.30E+01	3.05E+02 ± 3.45E+00	4.36E+02 ± 3.10E+01	6.78E+02±4.68E+01	6.28E+02 ± 7.54E+00	9.00E+02±5.24E+01
ABC (Karaboga and Basturk 2007)	3.08E+03 ± 1.91E+01	4.25E+02 ± 1.10E+01	3.00E+02 ± 1.25E+02	**4.14E+02 ± 5.23E+01**	**5.11E+02 ± 2.48E+01**	6.03E+02 ± 2.45E+00	7.50E+02 ± 1.94E+01
PSOGSA (Mirjalili and Hashim 2010)	2.74E+03± 8.83E+02	4.12E+03 ± 3.26E+03	3.56E+03 ± 7.87E+03	1.04E+03 ± 5.05E+02	6.46E+02 ± 3.40E+01	6.24E+02 ± 8.94E+00	9.72E+02 ± 6.32E+01
TLBO (Rao et al. 2012)	2.80E+03 ± 2.00E+02	4.38E+03 ± 1.60E+01	3.15E+02 ± 1.50E+01	4.35E+02 ± 3.92E+02	5.98E+02 ± 3.93E+01	6.03E+02 ± 4.77E-01	7.95E+02 ± 9.37E+00
WOA (Mirjalili and Lewis 2016)	1.66E+04 ± 2.00E+03	4.17E+03 ± 3.60E+01	3.84E+02 ± 1.33E+02	4.60E+02 ± 6.78E+01	5.75E+02 ± 2.68E+00	6.53E+02 ± 7.12E+00	1.39E+03 ± 9.90E+01
HHO (Heidari et al. 2019)	4.76E+04 ± 1.88E+03	4.55E+03 ± 5.00E+01	3.57E+02 ± 1.90E+01	4.41E+02 ± 2.43E+01	5.75E+02 ± 7.43E+00	6.00E+02 ± 4.01E-01	8.59E+02 ± 1.50E+01
HGSA (Wang et al. 2019)	**1.63E+03 ± 9.80E+01**	**2.68E+03 ± 2.50E+03**	4.36E+04±5.49E+03	5.19E+02 ± 2.63E+00	6.53E+02 ± 1.28E+01	6.08E+02 ± 4.54E+00	**7.41E+02 ± 3.01E+00**
ACVO	1.71E+03 ± 1.69E+03	4.67E+03 ± 2.17E+01	**3.00E+02 ± 2.84E-14**	4.66E+02±4.23E+01	5.50E+02 ± 8.23E+00	**6.00E+02 ± 1.69E-03**	7.65E+02 ±1.27E+01
	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
SADE (Qin and Suganthan 2005)	**9.51E+02±3.60E+01**	9.42E+02±3.25E+02	6.43E+03±8.01E+02	1.16E+03 ± 1.48E+01	2.85E+04 ± 1.82E+04	2.45E+04±7.37E+03	1.80E+03 ± 1.07E+02
ABC (Karaboga and Basturk 2007)	8.69E+02±2.71E+01	3.16E+03±7.43E+02	5.09E+03±6.45E+01	1.74E+03 ± 2.45E+02	3.53E+06 ± 5.96E+07	2.41E+04 ±7.30E+04	2.36E+03 ± 6.59E+02
PSOGSA (Mirjalili and Hashim 2010)	9.36E+02±3.25E+01	4.54E+03±1.67E+03	4.70E+03±6.23E+02	1.49E+03 ± 3.14E+02	6.00E+07 ± 1.48E+08	2.39E+07 ± 7.46E+07	9.87E+04 ± 2.69E+05
TLBO (Rao et al. 2012)	8.62E+02±5.52E+00	9.39E+02±2.31E+02	5.22E+03±1.08E+02	1.16E+03 ± 7.14E+02	2.13E+04 ± 2.03E+04	2.97E+04 ± 1.12E+03	**1.47E+03±1.70E+03**
WOA (Mirjalili and Lewis 2016)	8.56E+02±3.25E+00	9.02E+02±7.52E+00	5.21E+03±1.46E+01	1.17E+03 ± 2.96E+01	1.48E+06 ± 4.14E+05	3.02E+04 ± 2.70E+03	1.52E+03 ± 5.74E+02
HHO (Heidari et al. 2019)	9.17E+02±5.30E+00	9.91E+02±3.36E+01	6.02E+03±1.30E+02	1.30E+03±3.29E+01	3.89E+04 ± 2.61E+04	1.60E+04 ± 5.20E+04	6.20E+03 ± 9.24E+02
HGSA (Wang et al. 2019)	9.00E+02±9.03E+00	**9.00E+02±9.67E-14**	**4.21E+03 ± 2.93E+02**	1.20E+03 ± 2.98E+01	1.29E+05 ± 8.15E+04	**1.46E+04 ± 5.32E+03**	6.72E+03 ± 3.05E+03
ACVO	8.65E+02±2.38E+00	9.07E+02±2.49E+00	5.02E+03±5.53E+02	**1.15E+03±3.13E+01**	**2.10E+04 ± 1.99E+04**	2.33E+04 ± 1.82E+04	1.49E+03 ± 3.89E+01
	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}
SADE (Qin and Suganthan 2005)	4.31E+03±3.08E+03	2.79E+03±3.38E+02	2.49E+03±2.23E+02	6.90E+03 ± 6.52E+03	**3.77E+03 ± 5.94E+03**	2.82E+03 ± 3.23E+02	2.35E+03 ± 9.69E+00
ABC (Karaboga and Basturk 2007)	4.58E+03±8.64E+02	2.47E+03±2.37E+02	2.18E+03±1.26E+02	9.07E+04 ± 5.20E+03	1.08E+04 ± 5.61E+03	2.16E+03 ± 2.40E+02	2.50E+03 ± 1.60E+01
PSOGSA (Mirjalili and Hashim 2010)	5.31E+05±2.82E+06	3.05E+03±4.59E+02	2.27E+03±2.29E+02	3.07E+05 ± 3.07E+05	1.43E+04 ± 1.33E+04	2.57E+03 ± 2.35E+02	2.43E+03 ± 3.53E+01
TLBO (Rao et al. 2012)	2.93E+03±9.50E+02	4.76E+03±1.86E+02	3.19E+03±2.92E+02	4.54E+03 ± 9.10E+03	1.45E+04 ± 4.21E+04	2.91E+03 ± 1.71E+02	**2.30E+03 ± 1.21E+02**
WOA (Mirjalili and Lewis 2016)	1.04E+04±3.36E+02	2.86E+03±9.53E+01	2.63E+03±1.29E+02	4.75E+03±8.95E+03	2.01E+04 ± 9.98E+03	2.85E+03 ± 1.17E+02	2.56E+03 ± 1.18E+01
HHO (Heidari et al. 2019)	**1.93E+03±2.08E+02**	2.36E+03±4.45E+02	1.76E+03±3.02E+02	4.79E+04 ± 1.27E+03	2.81E+04 ± 8.57E+03	2.74E+03 ± 1.26E+02	2.37E+03 ± 3.40E+01
HGSA (Wang et al. 2019)	2.20E+03±7.21E+02	2.83E+03±2.32E+02	2.77E+03±1.99E+02	6.16E+04 ± 1.47E+04	5.42E+03±1.25E+03	2.86E+03 ± 2.24E+02	2.41E+03 ± 5.90E+01
ACVO	1.94E+03±2.42E+02	**2.18E+03±1.87E+02**	**1.75E+03±2.18E+01**	**4.38E+03 ± 1.54E+03**	3.96E+03±2.27E+03	**2.10E+03 ± 6.96E+01**	2.39E+03 ± 4.56E+01
	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}	f_{28}
SADE (Qin and Suganthan 2005)	2.31E+03±2.67E+01	2.86E+03±5.38E+01	3.01E+03±8.30E+01	2.89E+03 ± 2.36E+00	3.92E+03±6.42E+02	3.24E+03±1.37E+01	3.18E+03±4.39E+01
ABC (Karaboga and Basturk 2007)	6.35E+03±3.20E+03	2.75E+03±2.90E+01	3.05E+03±1.26E+01	3.02E+03 ± 2.83E+01	5.22E+03 ± 1.42E+02	3.58E+03 ± 2.48E+01	3.51E+03 ± 6.67E+01
PSOGSA (Mirjalili and Hashim 2010)	4.68E+03±1.91E+03	2.93E+03±8.75E+01	3.21E+03±1.43E+02	3.02E+03 ± 7.53E+01	5.70E+03 ± 1.30E+03	3.52E+03 ± 1.36E+02	3.52E+03 ± 2.00E+02
TLBO (Rao et al. 2012)	2.30E+03±1.36E+02	2.84E+03±4.38E+01	3.54E+03±6.41E+01	2.90E+03 ± 2.12E+01	4.06E+03 ± 3.08E+03	3.30E+03 ± 9.19E+01	3.77E+03 ± 2.24E+03
WOA (Mirjalili and Lewis 2016)	2.31E+03±2.91E+02	2.98E+03±1.32E+01	3.14E+03±2.64E+01	**2.88E+03 ± 1.58E+01**	7.54E+03 ± 2.54E+02	**3.22E+03 ± 3.03E+01**	4.49E+03±2.77E+02
HHO (Heidari et al. 2019)	3.16E+03±4.41E+02	2.75E+03±7.29E+01	3.18E+03±6.17E+01	2.90E+03 ± 2.02E+01	6.67E+03 ± 9.72E+02	3.30E+03 ± 3.96E+01	3.28E+03 ± 9.94E+00
HGSA (Wang et al. 2019)	2.30E+03±3.91E-09	2.76E+03±1.33E+02	2.92E+03±3.58E+01	2.89E+03 ± 7.59E+00	**2.85E+03 ± 5.07E+01**	3.25E+03 ± 2.08E+01	**3.11E+03 ± 2.82E+01**
ACVO	**2.30E+03±0.00E+00**	**2.70E+03±5.14E+00**	**2.86E+03±1.59E+01**	2.89E+03 ± 1.06E-01	5.29E+03 ± 1.51E+03	3.23E+03 ± 1.19E+01	3.55E+03 ± 5.96E+01

Bold values indicate the best results generated by the algorithms

Table 6 Results obtained by the ACVO and comparison algorithms on test functions, when $D = 50$

Algorithm	f_1	f_2	f_3	f_4	f_5	f_6	f_7
SADE (Qin and Suganthan 2005)	1.76E+03±1.25E+02	1.01E+03±4.23E+02	3.11E+03±4.80E+02	5.02E+02 ± 7.52E+01	8.02E+02 ± 2.29E+01	**6.05E+02±4.60E+01**	7.70E+02±1.63E+01
ABC (Karaboga and Basturk 2007)	2.10E+03±1.13E+02	9.56E+02±6.50E+05	2.44E+03±3.10E+04	7.67E+02 ± 2.86E+01	8.80E+02 ± 3.18E+01	6.35E+02 ± 5.72E+01	8.00E+02 ± 2.40E+01
PSOGSA (Mirjalili and Hashim 2010)	4.66E+03±2.39E+03	1.69E+09±4.33E+09	3.67E+04±6.07E+04	3.70E+03 ± 2.22E+03	7.87E+02 ± 7.88E+01	6.35E+02 ± 7.66E+00	1.46E+03 ± 1.51E+02
TLBO (Rao et al. 2012)	**1.34E+03±5.51E+02**	**3.15E+02±4.16E+03**	**3.00E+02 ± 0.00E+00**	5.19E+02 ± 9.53E+02	1.09E+03±4.26E+01	6.29E+02 ± 4.79E+00	1.40E+03±1.29E+02
WOA (Mirjalili and Lewis 2016)	3.59E+03±3.53E+03	9.12E+02±3.67E+05	8.39E+04±7.37E+03	2.13E+03 ± 4.90E+02	9.36E+02 ± 2.14E+01	6.40E+02 ± 2.43E+00	1.38E+03 ± 7.82E+01
HHO (Heidari et al. 2019)	1.74E+03±3.66E+05	1.14E+03±3.60E+03	5.83E+04±3.35E+04	6.27E+02 ± 4.58E+01	8.47E+02 ± 4.76E+01	6.21E+02 ± 9.34E+00	1.61E+03 ± 7.57E+01
HGSA (Wang et al. 2019)	2.10E+03±9.10E+02	9.42E+02±1.17E+03	1.19E+05±1.15E+04	6.02E+02 ± 2.91E+01	7.68E+02± 1.36E+01	6.25E+02 ± 3.97E+00	**7.70E+02 ± 3.44E+00**
ACVO	1.95E+03±1.42E+02	1.02E+03±7.63E+05	**3.00E+02±0.00E+00**	**4.73E+02 ± 6.40E+01**	**6.93E+02 ± 9.77E+01**	6.23E+02 ± 5.46E-02	7.88E+02 ± 2.27E+01
	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
SADE (Qin and Suganthan 2005)	**1.01E+03±2.71E+01**	1.15E+03±2.37E+02	9.60E+03±3.30E+03	**1.22E+03±5.63E+01**	7..71E+05±1.19E+05	1.80E+04±1.47E+04	1.18E+04±1.05E+04
ABC (Karaboga and Basturk 2007)	1.16E+03±4.04E+01	9.59E+03±2.31E+02	9.34E+03±2.28E+02	2.56E+03±2.71E+01	8.06E+05±9.99E+04	1.23E+04 ± 7.37E+03	7.11E+04 ± 7.78E+03
PSOGSA (Mirjalili and Hashim 2010)	1.09E+03±6.46E+01	1.26E+04±3.13E+03	7.67E+03±1.80E+03	5.20E+03±4.61E+03	1.19E+05±2.30E+04	2.36E+08 ± 7.44E+08	3.33E+06 ± 5.41E+06
TLBO (Rao et al. 2012)	1.10E+03±1.79E+01	9.00E+02±1.06E+02	7.76E+03±3.81E+02	1.52E+03±1.40E+02	9.25E+05±1.77E+04	3.92E+04 ± 1.72E+03	2.05E+06± 3.49E+05
WOA (Mirjalili and Lewis 2016)	1.25E+03±3.22E+01	1.47E+04±4.12E+03	1.30E+04±5.81E+02	3.87E+03±6.93E+02	3.60E+06±6.85E+05	8.74E+08 ± 1.75E+08	9.82E+05 ± 3.39E+05
HHO (Heidari et al. 2019)	1.04E+03±6.12E+01	1.57E+04±2.58E+03	6.87E+03±1.22E+02	1.48E+03±7.04E+01	1.81E+06±1.02E+04	1.26E+05 ± 7.25E+04	2.19E+05 ± 1.59E+04
HGSA (Wang et al. 2019)	1.09E+03±2.04E+01	9.00E+02±1.01E-13	6.83E+03±5.55E+02	1.23E+03±1.92E+01	7.11E+05±3.35E+05	**1.90E+03 ± 5.38E+02**	2.71E+04 ± 3.76E+04
ACVO	1.01E+03±1.93E+02	**9.00E+02±3.14E+00**	**6.80E+03±4.31E+02**	1.29E+03±2.05E+01	**5.60E+04±3.11E+04**	2.02E+03 ± 1.74E+03	**6.11E+03 ± 2.34E+03**
	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}
SADE (Qin and Suganthan 2005)	8.89E+03±2.69E+03	3.57E+03±1.90E+02	3.13E+03±2.78E+02	2.57E+05±1.67E+05	1.74E+04±7.44E+03	3.63E+03 ±1.68E+02	2.64E+03 ± 5.59E+02
ABC (Karaboga and Basturk 2007)	3.63E+04±1.50E+04	7.47E+03±6.57E+02	9.91E+03±2.47E+03	5.88E+05±2.28E+04	1.94E+05±1.50E+04	3.91E+03 ±2.91E+02	2.57E+03 ± 1.81E+02
PSOGSA (Mirjalili and Hashim 2010)	8.14E+04±1.09E+04	3.39E+03±7.44E+02	**2.24E+03±1.33E+02**	2.14E+05±1.42E+03	1.21E+04±1.06E+04	3.32E+03 ± 4.38E+02	2.62E+03 ± 1.02E+02
TLBO (Rao et al. 2012)	1.30E+04±4.47E+03	5.62E+03±1.19E+02	2.24E+03±2.53E+02	2.14E+05±1.02E+04	9.60E+03±1.56E+04	3.31E+03 ±1.25E+02	2.86E+03 ± 5.26E+01
WOA (Mirjalili and Lewis 2016)	2.01E+08±2.59E+08	4.19E+03±3.89E+02	3.44E+03±2.42E+02	5.52E+05±4.09E+06	8.26E+05±8.74E+06	**3.18E+03 ±1.53E+02**	2.75E+03 ± 1.35E+01
HHO (Heidari et al. 2019)	7.08E+04±2.89E+04	4.07E+03±4.61E+02	3.76E+03±4.71E+02	3.21E+05±2.85E+04	1.73E+06±1.83E+06	3.53E+03 ± 3.52E+02	2.85E+03 ± 1.15E+02
HGSA (Wang et al. 2019)	8.61E+03±1.64E+03	3.55E+03±3.50E+02	3.44E+03±3.14E+02	1.74E+05±7.74E+04	1.69E+04±3.36E+03	3.38E+03 ± 2.75E+02	2.56E+03 ± 3.22E+01
ACVO	**8.60E+03±1.36E+03**	**3.09E+03±2.33E+02**	3.56E+03±4.16E+02	**1.10E+05±1.05E+04**	**9.52E+03±9.17E+03**	3.35E+03 ±1.68E+02	**2.45E+03 ± 3.01E+02**
	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}	f_{28}
SADE (Qin and Suganthan 2005)	7.27E+03 ± 4.12E+03	3.33E+03 ± 2.04E+01	3.21E+03 ± 2.05E+02	3.18E+03 ± 2.21E+01	5.45E+03 ± 5.41E+02	**3.48E+03±5.17E+01**	3.30E+03±2.26E+01
ABC (Karaboga and Basturk 2007)	1.35E+04 ± 3.29E+03	4.20E+03 ± 6.53E+01	4.24E+03 ± 3.90E+02	2.11E+04 ± 9.98E+03	1.39E+04 ± 7.04E+03	5.03E+03 ± 1.02E+03	9.85E+03 ± 5.00E+03
PSOGSA (Mirjalili and Hashim 2010)	8.86E+03 ± 1.52E+03	3.53E+03 ± 2.19E+02	3.81E+03 ± 2.44E+02	4.75E+03 ± 1.06E+03	9.89E+03 ± 1.49E+03	4.44E+03 ± 4.31E+02	6.22E+03 ± 1.10E+03
TLBO (Rao et al. 2012)	1.03E+04 ± 2.12E+02	3.49E+03 ± 5.68E+01	3.63E+03 ± 5.23E+01	5.66E+03 ± 1.03E+02	1.18E+04 ± 7.14E+02	4.24E+03 ± 1.20E+02	7.33E+03 ± 5.37E+02
WOA (Mirjalili and Lewis 2016)	1.51E+04 ± 4.93E+02	**3.23E+03 ± 3.50E+01**	3.37E+03 ± 2.19E+01	4.28E+03±4.63E+02	9.06E+03 ± 4.05E+02	3.68E+03 ± 8.84E+01	4.45E+03 ± 4.34E+02
HHO (Heidari et al. 2019)	1.21E+04 ± 1.61E+03	3.54E+03 ± 1.50E+02	3.60E+03 ± 1.23E+02	3.11E+03 ± 3.12E+01	1.32E+04 ±1.12E+03	4.71E+03 ± 4.32E+02	3.36E+03 ± 1.10E+01
HGSA (Wang et al. 2019)	1.01E+04± 4.22E+02	3.30E+03 ± 1.80E+02	3.29E+03 ± 4.71E+01	3.08E+03 ± 1.89E+01	**2.90E+03 ± 7.51E-13**	4.02E+03±3.01E+02	3.31E+03 ± 1.56E+01
ACVO	**7.23E+03 ± 2.79E+03**	3.26E+03 ± 1.93E+03	**3.02E+03±1.86E+01**	**3.06E+03 ± 3.05E+01**	3.22E+03 ± 1.64E+02	4.01E+03 ± 1.13E+02	**3.29E+03 ± 1.89E+01**

Bold values indicate the best results generated by the algorithms

Table 7 Comparison of the algorithms in terms of the average and final ranks computed by Friedman test

Algorithm	10D		30D		50D	
	Average rank	Final rank	Average rank	Final rank	Average rank	Final rank
SADE (Qin and Suganthan 2005)	3.589	2	3.828	2	3.304	3
ABC (Karaboga and Basturk 2007)	5.321	7	4.638	5	6.142	8
PSOGSA (Mirjalili and Hashim 2010)	4.893	6	6.327	8	5.357	5
HHO (Heidari et al. 2019)	4.321	3	4.603	4	4.625	4
WOA (Mirjalili and Lewis 2016)	4.839	5	5.172	7	5.875	7
TLBO (Rao et al. 2012)	6.107	8	4.724	6	5.518	6
HGSA (Wang et al. 2019)	4.393	4	3.948	3	3.143	2
ACVO	**2.536**	**1**	**2.759**	**1**	**2.036**	**1**

Bold values indicate the best results generated by the algorithms

Table 8 Statistical results obtained by the Wilcoxon signed rank test between ACVO and comparison algorithms in D= 10, 30 and 50 dimensions, and $\alpha = 0.05$

ACVO vs.	$D = 10$			$D = 30$			$D = 50$		
	$T+$	$T-$	p-value	$T+$	$T-$	p-value	$T+$	$T-$	p-value
SADE (Qin and Suganthan 2005)	241	84	3.49E−02	299	107	2.85E−02	297	109	3.24E−02
ABC (Karaboga and Basturk 2007)	357	49	4.40E−04	319.5	86.5	8.04E−03	396	10	1.00E−05
PSOGSA (Mirjalili and Hashim 2010)	348	30	1.40E−04	421.5	13.5	1.00E−05	388	18	1.00E−05
TLBO (Rao et al. 2012)	356	50	5.00E−04	349.5	56.5	8.40E−04	387	19	1.00E−05
WOA (Mirjalili and Lewis 2016)	335	43	4.40E−04	362	44	3.00E−04	381	25	1.00E−05
HHO (Heidari et al. 2019)	279	99	3.08E−02	330.5	55.5	3.74E−03	261	90	3.00E−02
HGSA (Wang et al. 2019)	231	69	2.09Ev02	292.5	113.5	4.14E-02	227	73	2.78E−02

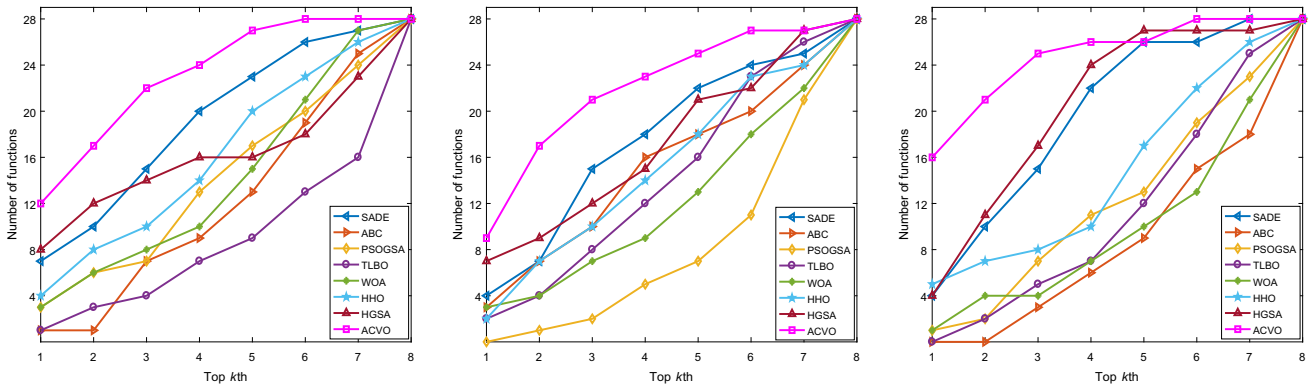


Fig. 6 The aggregate performance of algorithms on test problems with **a** 10, **b** 30, and **c** 50 dimensions

tions, matching or exceeding the best result reported by its counterparts.

The contrast estimation based on medians for algorithms over test problems is summarized in Table 11. The objective of contrast estimation is to show the global performance of the algorithms by computing the magnitudes of the differences among their performances (Derrac et al. 2011). This test shows how well one algorithm compared to another one. The results highlight the ACVO as the best performing algorithm. ACVO obtains low error rates compared with other algorithms.

Figure 8 represents the convergence behavior of comparison algorithms on test functions f_4, f_{10}, f_{22}, and f_{28} as representatives from four groups of unimodal, multimodal, hybrid and composite functions. Figure 7 represents the 3D plot of these functions. As shown in Fig. 8, on f_4 function, the ACVO and SADE algorithms outperformed counterparts in terms of average best-so-far solution and convergence speed. On f_{10}, ACVO, SADE, and HGSA obtained similar perfor-

Table 9 Unadjusted p-values of comparison algorithms where the ACVO is the control algorithm

Dimension	Statistics	Algorithm						
		SADE (Qin and Suganthan 2005)	ABC (Karaboga and Basturk 2007)	PSOGSA (Mirjalili and Hashim 2010)	TLBO (Rao et al. 2012)	WOA (Mirjalili and Lewis 2016)	HHO (Heidari et al. 2019)	HGSA (Wang et al. 2019)
10	p-value	1.08E-01	2.09E-05	3.17E-04	4.88E-08	4.34E-04	6.38E-03	4.56E-03
	p_{Bonf}	7.53E-01	1.46E-04	2.22E-03	3.42E-07	3.03E-03	4.46E-02	3.19E-02
	p_{Holm}	1.08E-01	1.25E-04	1.59E-03	3.42E-07	1.73E-03	1.37E-02	1.37E-02
	$p_{Hochberg}$	1.08E-01	1.25E-04	1.59E-03	3.42E-07	1.73E-03	1.28E-02	1.28E-02
30	p-value	9.66E-02	3.48E-03	2.89E-08	2.25E-03	1.75E-04	4.13E-03	6.44E-02
	p_{Bonf}	6.76E-01	2.44E-02	2.02E-07	1.57E-02	1.23E-03	2.89E-02	4.51E-01
	p_{Holm}	1.29E-01	1.39E-02	2.02E-07	1.12E-02	1.05E-03	1.39E-02	1.29E-01
	$p_{Hochberg}$	9.66E-02	1.24E-02	2.02E-07	1.12E-02	1.05E-03	1.24E-02	9.66E-02
50	p-value	5.28E-02	3.52E-10	3.90E-07	1.04E-07	4.50E-09	7.65E-05	9.08E-02
	p_{Bonf}	3.69E-01	2.47E-09	2.73E-06	7.30E-07	3.15E-08	5.35E-04	6.36E-01
	p_{Holm}	1.06E-01	2.47E-09	1.56E-06	5.22E-07	2.70E-08	2.29E-04	1.06E-01
	$p_{Hochberg}$	9.08E-02	2.47E-09	1.56E-06	5.22E-07	2.70E-08	2.29E-04	9.08E-02

Table 10 Adjusted p-values of comparison algorithms where the ACVO is the control algorithm

Dimension	Statistics	Algorithm SADE (Qin and Suganthan 2005)	ABC (Karaboga and Basturk 2007)	PSOGSA (Mirjalili and Hashim 2010)	TLBO (Rao et al. 2012)	WOA (Mirjalili and Lewis 2016)	HHO (Heidari et al. 2019)	HGSA (Wang et al. 2019)
10	p-value	6.75E-01	4.79E-03	4.89E-04	1.03E-05	2.21E-03	1.29E-01	4.29E-03
	p_{Bonf}	4.72E+00	3.35E-02	3.42E-03	7.22E-05	1.54E-02	9.00E-01	3.00E-02
	p_{Holm}	6.75E-01	1.72E-02	2.93E-03	7.22E-05	1.10E-02	2.57E-01	1.72E-02
	$p_{Hochberg}$	6.75E-01	1.44E-02	2.93E-03	7.22E-05	1.10E-02	2.57E-01	1.44E-02
30	p-value	1.37E-01	9.40E-03	1.28E-08	6.05E-03	1.64E-03	5.04E-02	1.70E-02
	p_{Bonf}	9.56E-01	6.58E-02	8.99E-08	4.24E-02	1.15E-02	3.53E-01	1.19E-01
	p_{Holm}	1.37E-01	3.76E-02	8.99E-08	3.03E-02	9.87E-03	1.01E-01	5.10E-02
	$p_{Hochberg}$	1.37E-01	3.76E-02	8.99E-08	3.03E-02	9.87E-03	1.01E-01	5.10E-02
50	p-value	1.70E-01	5.38E-10	4.21E-07	5.45E-06	4.98E-09	2.01E-03	7.50E-02
	p_{Bonf}	1.19E+00	3.77E-09	2.95E-06	3.82E-05	3.49E-08	1.41E-02	5.25E-01
	p_{Holm}	1.70E-01	3.77E-09	2.11E-06	2.18E-05	2.99E-08	6.03E-03	1.50E-01
	$p_{Hochberg}$	1.70E-01	3.77E-09	2.11E-06	2.18E-05	2.99E-08	6.03E-03	1.50E-01

mance; however, the ACVO algorithm is a little better. The ACVO outperformed other ones on f_{22} function. The ACVO, SADE, and HHO algorithms performed better than others on f_{28} function. Overall, ACVO shows a steady convergence behavior, matching or exceeding the best results obtained by other algorithms. It is not trapped in local optima due to its efficient exploration and exploitation of solution space.

Figure 9 shows the distribution of solutions obtained by comparison algorithms on f_4, f_{10}, f_{22}, and f_{28} functions with 10, 30, and 50 dimensions. From Fig. 9c, f, i, and l, we observe that ACVO obtains the best solutions with minimum deviation from the global solution. Other algorithms generate solutions with larger distribution due to their inconsistent convergence behavior. This justifies that the ACVO is an efficient method to optimize high dimension problems compared with other algorithms. In Fig. 9a, b, d, and e, the SADE algorithm obtains the best results indicating that it an efficient method to optimize low and medium scale unimodal and multimodal problems. The HGSA obtains the best results on medium scale hybrid and composite functions f_{22} and f_{28}. This is shown in Fig. 9h, k. For 30 dimension functions, ACVO obtains moderate results; however, its performance is close to HGSA and SADE algorithms. Except for function f_{10}, ACVO finds the best solutions on 10 dimension functions compared with other algorithms. This indicates that ACVO is a powerful method in optimizing low scale functions. Overall, we can conclude that the ACVO algorithm is an efficient method for solving problems with different dimensions. It is important to note that the algorithms should be tested on a variety of functions to comment more confidently on their performance in finding optimal solutions.

To further analyze the search efficiency of the ACVO algorithm, it is evaluated on recent CEC2019 test functions (Price et al. 2018). The statistical results are summarized in Table 12. The results confirm that the ACVO obtains better performance than its counterparts except in CEC03, CEC08, and CEC10. The *Mean* results generated for the CEC03, CEC08, and CEC10 functions where the ACVO is not the ideal is still competitive to the other algorithms. An investigation on the *Std* values shows that the values generated by ACVO is comparable to other algorithms.

3.4 Scalability analysis

To show the search performance of the proposed ACVO algorithm and counterpart algorithms in solving high dimension problems, we test the algorithms on benchmark functions of different sizes. We performed a series of tests on 100, 500, and 1000 dimension scalable unimodal and multimodal problems. Table 13 shows the characteristics of used unimodal and multimodal functions.

Tables 14, 15, and 16 summarize the statistical results generated by algorithms. The results confirm that ACVO generates better results compared with its counterparts. The proposed ACVO algorithm generates the best mean results in 8, 5, and 9 test problems in 100, 500, and 1000 dimensions, respectively. If we consider the test problems that ACVO generates the similar mean results compared with counterparts, the success numbers will increase. The mean results of the problems where ACVO is not the best performing algorithm are competitive to the best performance. This issue proves the superior scalability of the ACVO algorithm

3.5 ACVO for engineering problems

To investigate the performance of ACVO in solving engineering applications, it is evaluated on seven real-world engineering problems. These problems are drawn from CEC2011 competition (Das and Suganthan 2012) and related literature (Emami 2020, 2021), and include the frequency-modulated sound waves (FMSW), static economic load dispatch (SELD), transmission network expansion planning (TNEP), spread spectrum radar poly phase code design (SSRPCD), speed reducer design (SRD), welded beam design (WBD), and rolling element bearing design (REBD). The characteristics of FMSW, SELD, TNEP, and SSRPCD problems are summarized in Table 17.

3.5.1 Frequency-modulated sound waves (FMSW)

The objective of FMSW is to estimate a sound that has the minimum difference with a target sound. FMSW is a multimodal problem with six dimensions that need to be optimized to minimize the difference between the target sound and the estimated sound. The mathematical formulation and constraints of FMSW are formulated as follows (Das and Suganthan 2012):

$$\begin{aligned} &\text{minimize } f(\mathbf{X}) = \sum_{t=0}^{100} (y(t) - y_0(t))^2 \\ &\text{where} \\ &y_t(t) = a_1.\sin(\omega_1.t.\theta + a_2.\sin(\omega_2.t.\theta + a_3.\sin(\omega_3.t.\theta))) \\ &y_g(t) = 1.\sin(5.t.\theta + 1.5.\sin(4.8.t.\theta + 2.\sin(4.9.t.\theta))) \end{aligned} \tag{23}$$

where $\theta = 2\pi/100$, y_t is the target sound, and y_g is the generated sound by the algorithm.

3.5.2 Spread spectrum radar polyphase code design (SSRPCD)

The objective of SSRPCD is to create a radar system using the polyphase codes (Wang et al. 2019). SSRPCD is a nonlinear problem with many local optima. It falls in the category of continuous min-max global optimization problems and defined as follows Das and Suganthan (2012):

Table 11 Contrast estimation results for the comparison algorithms

Dimension	Algorithm	SADE (Qin and Suganthan 2005)	ABC (Karaboga and Basturk 2007)	PSOGSA (Mirjalili and Hashim 2010)	TLBO (Rao et al. 2012)	WOA (Mirjalili and Lewis 2016)	HHO (Heidari et al. 2019)	HGSA (Wang et al. 2019)	ACVO
10	SADE (Qin and Suganthan 2005)	0	2.15E-05	1.59E-05	4.13E-05	1.47E-05	8.26E-06	1.31E-05	−3.82E-06
	ABC (Karaboga and Basturk 2007)	−2.15E-05	0	−5.61E-06	1.98E-05	−6.83E-06	−1.32E-05	−8.37E-06	−2.53E-05
	PSOGSA (Mirjalili and Hashim 2010)	−1.59E-05	5.61E-06	0	2.55E-05	−1.21E-06	−7.60E-06	−2.75E-06	−1.97E-05
	TLBO (Rao et al. 2012)	−4.13E-05	−1.98E-05	−2.55E-05	0	−2.67E-05	−3.31E-05	−2.82E-05	−4.51E-05
	WOA (Mirjalili and Lewis 2016)	−1.47E-05	6.83E-06	1.21E-06	2.67E-05	0	−6.39E-06	−1.54E-06	−1.85E-05
	HHO (Heidari et al. 2019)	−8.26E-06	1.32E-05	7.60E-06	3.31E-05	6.39E-06	0	4.85E-06	−1.21E-05
	HGSA (Wang et al. 2019)	−1.31E-05	8.37E-06	2.75E-06	2.82E-05	1.54E-06	−4.85E-06	0	−1.69E-05
	ACVO	3.82E-06	2.53E-05	1.97E-05	4.51E-05	1.85E-05	1.21E-05	1.69E-05	0
30	SADE (Qin and Suganthan 2005)	0	9.80E-06	4.02E-05	1.16E-05	1.71E-05	3.49E-06	6.85E-07	−1.83E-05
	ABC (Karaboga and Basturk 2007)	−9.80E-06	0	3.04E-05	1.83E-06	7.26E-06	−6.31E-06	−9.11E-06	−2.81E-05
	PSOGSA (Mirjalili and Hashim 2010)	−4.02E-05	−3.04E-05	0	−2.85E-05	−2.31E-05	−3.67E-05	−3.95E-05	−5.85E-05
	TLBO (Rao et al. 2012)	−1.16E-05	−1.83E-06	2.85E-05	0	5.44E-06	−8.13E-06	−1.09E-05	−3.00E-05
	WOA (Mirjalili and Lewis 2016)	−1.71E-05	−7.26E-06	2.31E-05	−5.44E-06	0	−1.36E-05	−1.64E-05	−3.54E-05
	HHO (Heidari et al. 2019)	−3.49E-06	6.31E-06	3.67E-05	8.13E-06	1.36E-05	0	−2.81E-06	−2.18E-05
	HGSA (Wang et al. 2019)	−6.85E-07	9.11E-06	3.95E-05	1.09E-05	1.64E-05	2.81E-06	0	−1.90E-05
	ACVO	1.83E-05	2.81E-05	5.85E-05	3.00E-05	3.54E-05	2.18E-05	1.90E-05	0
50	SADE (Qin and Suganthan 2005)	0	7.59E-05	6.07E-05	5.32E-05	6.86E-05	3.98E-05	1.50E-05	−1.81E-05
	ABC (Karaboga and Basturk 2007)	−7.59E-05	0	−1.52E-05	-2.28E-05	−7.39E-06	−3.62E-05	-6.09E-05	−9.40E-05
	PSOGSA (Mirjalili and Hashim 2010)	−6.07E-05	1.52E-05	0	−7.55E-06	7.85E-06	v2.09E-05	−4.57E-05	−7.88E-05
	TLBO (Rao et al. 2012)	−5.32E-05	2.28E-05	7.55E-06	0	1.54E-05	−1.34E-05	−3.81E-05	−7.13E-05
	WOA (Mirjalili and Lewis 2016)	−6.86E-05	7.39E-06	−7.85E-06	−1.54E-05	0	−2.88E-05	−5.35E-05	−8.67E-05
	HHO (Heidari et al. 2019)	−3.98E-05	3.62E-05	2.09E-05	1.34E-05	2.88E-05	0	−2.47E-05	−5.79E-05
	HGSA (Wang et al. 2019)	−1.50E-05	6.09E-05	4.57E-05	3.81E-05	5.35E-05	2.47E-05	0	−3.31E-05
	ACVO	1.81E-05	9.40E-05	7.88E-05	7.13E-05	8.67E-05	5.79E-05	3.31E-05	0

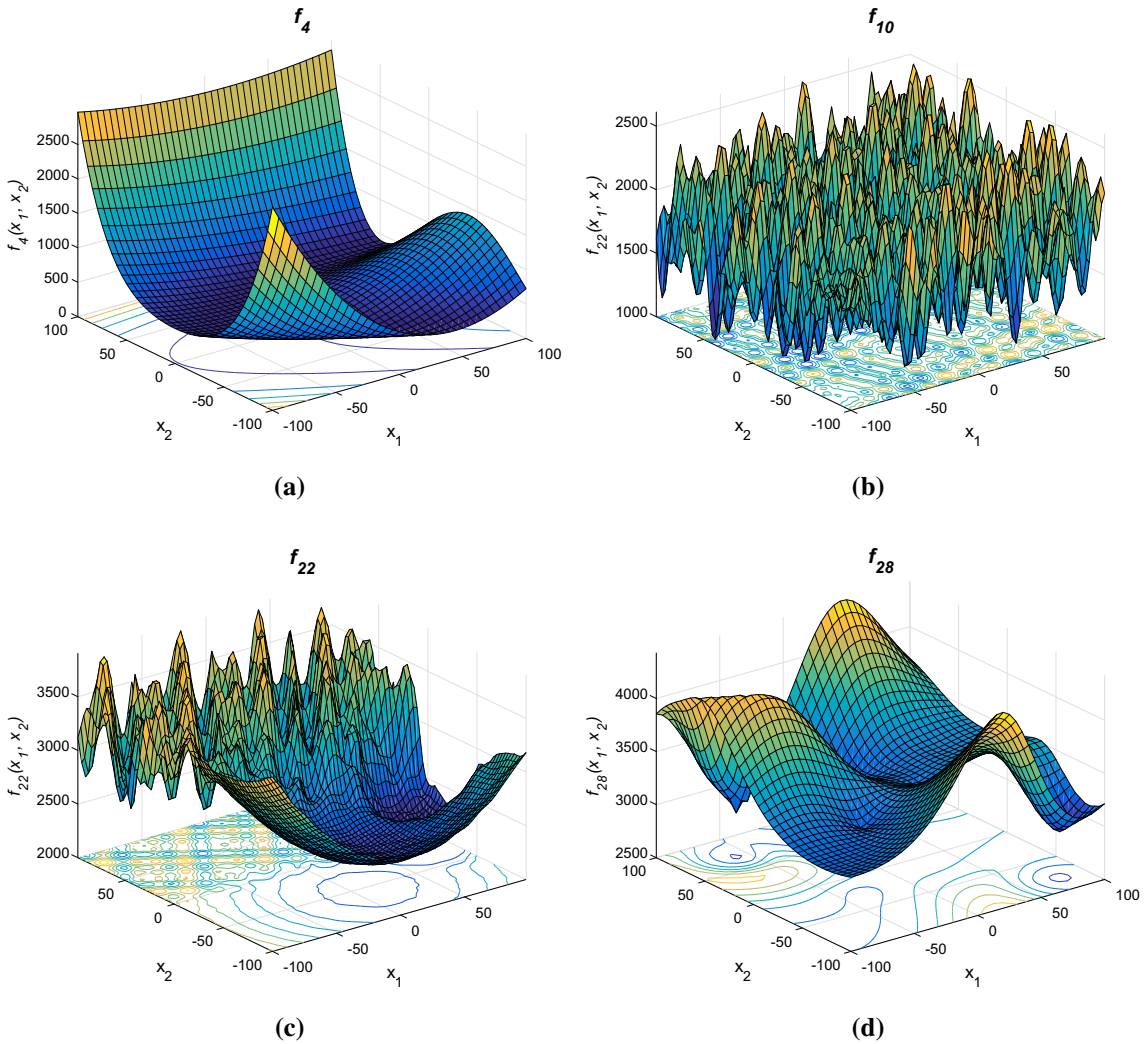


Fig. 7 Plots of f_4, f_{10}, f_{22}, and f_{28} test problems

$$\min_{x \in X} f(x) = \max\{\phi_1(x), ..., \phi_{2m}(x)\}$$
$$X = \{(x_1, ..., x_n) \in R^n | 0 \le x_j \le 2\pi,\ j = 1, 2, ..., n\}$$
subject to

$$\begin{aligned}
\phi_{2i-1}(x) &= \sum_{j=i}^{n} \cos\Big(\sum_{k=|2i-j-1|+1}^{j} x_k\Big), \quad i = 1, ..., n \\
\phi_{2i}(x) &= 0.5 + \sum_{j=i+1}^{n} \cos\Big(\sum_{k=|2i-j|+1}^{j} x_k\Big), \quad i = 1, ..., n-1 \\
\phi_{m+i}(x) &= -\phi_i(x), \quad i = 1, ..., m
\end{aligned} \tag{24}$$

where n is the number of variables and $m = 2n - 1$.

3.5.3 Transmission network expansion planning (TNEP)

An optimal design of TNEP is essential for planning the power systems efficiently and economically. The objective of the TNEP is to cope with the problem of finding a set of transmission lines that must be built in a way that no overloads are generated during the planning and achieve the minimum cost of the expansion plan. The TNEP without security constraints is formulated as follows Das and Suganthan (2012):

$$\begin{aligned}
&\text{minimize } f = \sum_{l \in \Omega} c_l n_l + W_1 \sum_{ol} (abs(f_l) - \bar{f}_l) + W_2 (n_l - \bar{n}_l) \\
&\text{subject to} \\
&\quad Sf + g = d \\
&\quad f_l - \gamma_l (n_l^0 + n_l)(\Delta\theta_l) = 0, \quad \text{for } l \in 1, 2, ...nl \\
&\quad |f_l| \le (n_l^0 + n_l)\bar{f}_l, \quad \text{for } l \in 1, 2, ...nl \\
&\quad 0 \le n_l \le \bar{n}_l
\end{aligned} \tag{25}$$

nl: total number of lines in the circuit
Ω: set of all right-of-ways
$\bar{n}_l$: the maximum number of circuits that can be added in lth right-of-way
$\bar{f}_l$: the maximum allowed real power flow in the circuit in lth right-of-way
f_l: total real power flow by the circuit in lth right-of-way
ol: the set of overloaded lines

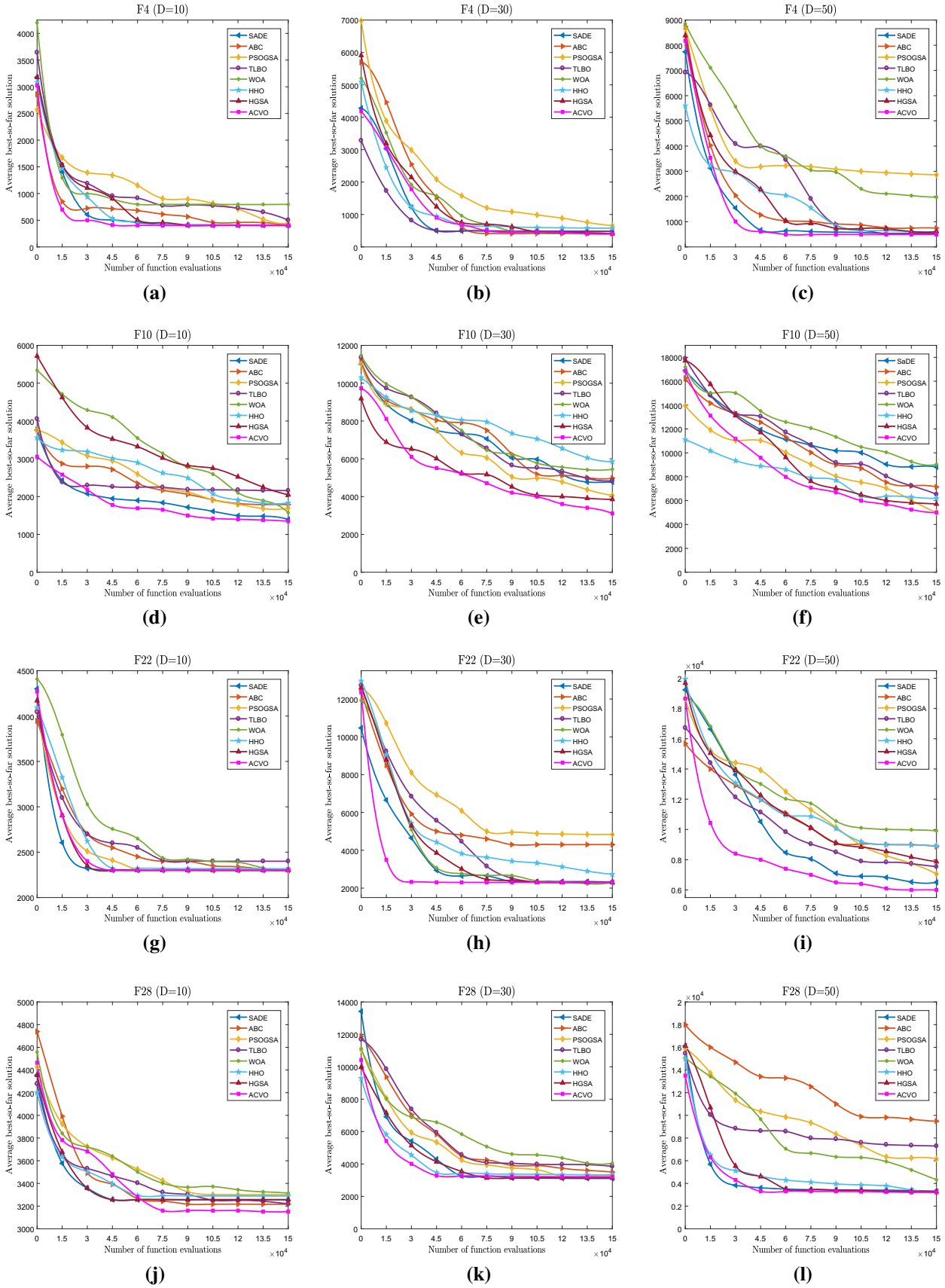


Fig. 8 Convergence plots of ACVO and counterpart algorithms on f_4, f_{10}, f_{22}, and f_{28} test functions with 10, 30 and 50 dimensions

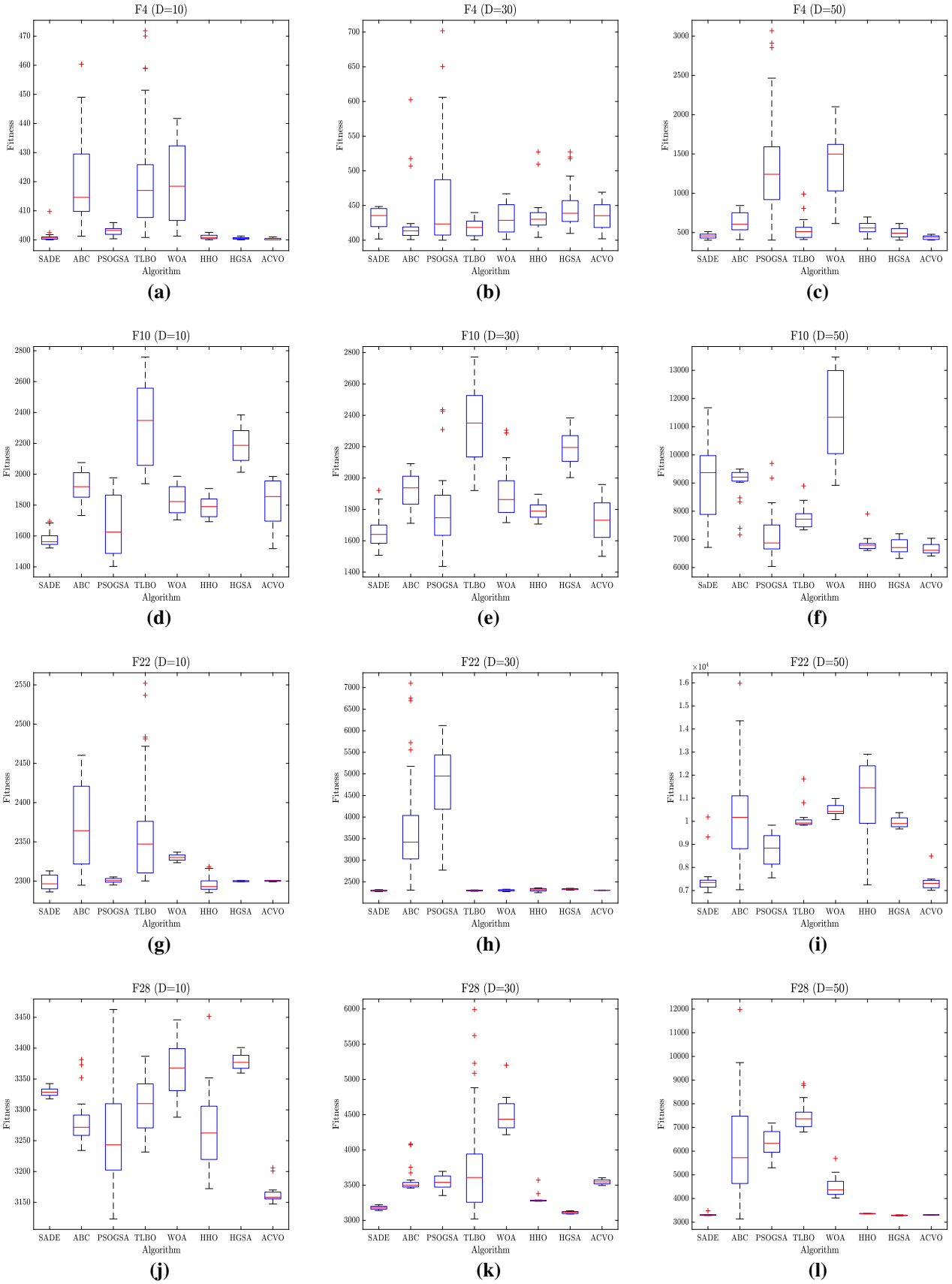


Fig. 9 The box-and-whisker plots of solutions reported by ACVO and counterpart algorithms on f_4, f_{10}, f_{22}, and f_{28} test functions with 10, 30, and 50 dimensions

Table 12 Results obtained by comparison algorithms on CEC2019 test functions

Algorithm	CEC01	CEC02	CEC03	CEC04	CEC05
SADE (Qin and Suganthan 2005)	5.93E+07 ± 9.44E+05	1.74E+01 ± 7.41E-03	1.27E+01 ± 1.24E-05	2.19E+02 ± 2.75E+01	1.83E+00 ± 1.93E-02
ABC (Karaboga and Basturk 2007)	6.79E+08 ± 4.47E+06	3.27E+02 ± 2.22E+02	1.27E+01 ± 1.31E-05	8.28E+01 ± 5.60E+00	1.46E+00 ± 1.09E-01
PSOGSA (Mirjalili and Hashim 2010)	5.93E+07 ± 9.44E+05	2.13E+01 ± 3.25E-03	1.27E+01 ± 3.28E-05	2.19E+02 ± 2.75E+01	1.83E+00 ± 1.93E-02
TLBO (Rao et al. 2012)	6.79E+09 ± 1.44E+09	6.91E+02 ± 1.57E+02	1.27E+01 ± 5.03E-05	3.47E+02 ± 1.85E+02	1.99E+00 ± 5.84E-02
WOA (Mirjalili and Lewis 2016)	7.58E+05 ± 2.87E+05	1.80E+01 ± 3.74E-01	1.27E+01 ± 4.19E-04	1.60E+04 ± 9.37E+03	5.36E+00 ± 8.22E-01
HHO (Heidari et al. 2019)	8.82E+08 ± 8.06E+07	1.83E+01 ± 5.34E-05	1.27E+01 ± 4.56E-11	1.38E+02 ± 1.45E+01	1.61E+00 ± 2.33E-01
HGSA (Wang et al. 2019)	2.97E+05 ± 5.33E+04	1.74E+01 ± 1.09E-02	1.27E+01 ± 3.77E-07	2.16E+02 ± 6.03E+01	1.80E+00 ± 5.33E-03
ACVO	**2.17E+05 ± 3.20E+05**	**1.73E+01 ± 2.51E-15**	1.27E+01 ± 0.00E+00	**3.95E+01 ± 4.14E+00**	**1.12E+00 ± 4.75E-02**
Algorithm	CEC06	CEC07	CEC08	CEC09	CEC10
SADE (Qin and Suganthan 2005)	1.02E+01 ± 3.51E-01	2.48E+02 ± 2.15E+02	4.65E+00 ± 3.41E-01	5.31E+00 ± 6.83E-01	2.03E+01 ± 5.23E-02
ABC (Karaboga and Basturk 2007)	8.00E+00 ± 5.61E-01	1.96E+02 ± 1.69E+02	6.32E+00 ± 3.45E-01	4.40E+00 ± 1.77E+00	**1.89E+01 ± 1.82E-01**
PSOGSA (Mirjalili and Hashim 2010)	1.02E+01 ± 3.51E-01	2.51E+02 ± 1.17E+02	4.65E+00 ± 3.41E-01	5.48E+00 ± 9.53E-01	2.14E+01 ± 2.54E-02
TLBO (Rao et al. 2012)	1.00E+01 ± 8.34E-02	5.96E+02 ± 3.15E+01	**4.50E+00 ± 2.59E-01**	4.68E+01 ± 1.45E+01	2.05E+01 ± 3.35E-02
WOA (Mirjalili and Lewis 2016)	1.12E+01 ± 2.34E-01	1.11E+03 ± 1.20E+02	6.44E+00 ± 4.33E-01	2.14E+01 ± 2.66E+02	3.94E+00 ± 4.02E-01
HHO (Heidari et al. 2019)	6.54E+00 ± 9.51E-01	3.44E+02 ± 1.64E+02	5.36E+00 ± 1.76E-01	4.91E+00 ± 6.70E-01	2.01E+01 ± 8.40E-02
HGSA (Wang et al. 2019)	1.04E+01 ± 7.75E-01	4.45E+02 ± 1.02E+02	4.92E+00 ± 3.98E-01	4.67E+00 ± 2.23E-01	2.03E+01 ± 6.38E-02
ACVO	**1.00E+00 ± 1.05E-13**	**1.90E+02 ± 1.35E+02**	5.24E+00 ± 7.25E-01	**2.34E+00 ± 3.43E-05**	2.00E+01 ± 3.68E-02

Bold values indicate the best results generated by the algorithms

$\Delta\theta_l$: phase angle difference in the lth right-of-way
n_l^0: the number of circuits in the base case
n_l: the number of circuits added in lth right-of-way
c_l: cost of line added in the lth right-of-way
S: branch-node incidence transposed matrix of the power system
f: the vector with elements f_l
γ_l: susceptance of the circuit that can be added to lth right-of-way

3.5.4 Static economic load dispatch (SELD)

SELD aims to minimize the fuel cost of generating units through finding the optimal thermal generation schedule provided that four constraints are satisfied. The constraints are prohibited operating zones, generator operation, ramp rate limits, and load demand. SELD is a non-differentiable, complex, and nonlinear problem with multiple local minimums. This problem mathematically is expressed as follows Das and Suganthan (2012):

Table 13 Characteristics of scalable unimodal and multimodal test functions

No.	Problem	Search range	F_{min}
F_1	Sphere	[−100 100]	0
F_2	Sum Squares	[−10, 10]	0
F_3	Quartic	[−1.28, 1.28]	0
F_4	Dixon Price	[−10, 10]	0
F_5	Zakharov	[−5.12, 5.12]	0
F_6	Rosenbrock	[−30, 30]	0
F_7	Rastrigin	[−5.12 5.12]	0
F_8	Griewank	[−600, 600]	0
F_9	Shubert	[−10, 10]	−186.7309
F_{10}	Penalized	[−50, 50]	0

Table 14 Results obtained by comparison algorithms on test functions with 100 dimensions

Algorithm	F_1	F_2	F_3	F_4	F_5
SADE (Qin and Suganthan 2005)	5.97E-05±2.15E-06	5.17E-26±6.20E-32	4.77E-02±3.15E-02	6.49E-07±3.80E-09	4.89E-46±9.88E-53
ABC (Karaboga and Basturk 2007)	5.30E-04±2.70E-05	8.76E-18±67-16	6.02E-01±8.70E-04	2.15E-03±3.45E-04	9.10E-37±5.37E-32
PSOGSA cite[53]	2.45E-15±1.47E-17	2.55E-74±3.46E-79	4.90E-02±4.26E-02	3.57E-06±6.15E-06	2.55E-08±7.35E-11
TLBO (Heidari et al. 2019)	6.54E-26±4.50E-19	3.01E-55±2.22E-48	5.14E-03±9.14E-02	5.80E-05±1.73E-05	7.39E-60±8.97E-52
WOA (Mirjalili and Lewis 2016)	3.54E-74±1.24E-28	2.09E-66±3.75E-36	6.20E-03±2.36E-03	1.25E-05±4.77E-06	1.98E-52±6.18E-48
HHO (Rao et al. 2012)	3.20E-77±6.34E-80	6.01E-67±4.07E-69	9.96E-04±7.60E-03	9.02E-07±7.03E-08	9.18E-67±3.80E-64
HGSA (Wang et al. 2019)	3.17E-80±2.97E-67	2.08E-81±6.21E-90	5.05E-04±6.50E-05	2.47E-08±3.02E-07	3.80E-76±3.16E-77
ACVO	**2.80E-83±2.97E-90**	**6.99E-84±3.57E-86**	**3.05E-04±8.50E-04**	**2.34E-08±1.62E-06**	**6.28E-77±6.48E-80**
	F_6	F_7	F_8	F_9	F_{10}
SADE (Qin and Suganthan 2005)	3.34E-08±3.18E-09	6.14E-53±4.13E-60	0.00E+00±0.00E+00	−1.87E+02±4.06E-03	8.12E-04±7.01E-05
ABC (Karaboga and Basturk 2007)	6.47E-10±3.47E-09	1.88E-25±4.12E-26	3.20E-74±5.47E-86	−1.77E+02±3.08E-03	4.27E-03±3.02E-04
PSOGSA (Mirjalili and Hashim 2010)	2.65E-06±1.68E-05	7.13E-56±2.18E-57	0.00E+00±0.00E+00	−1.87E+02±2.55E-04	2.97E-05±1.97E-06
TLBO (Heidari et al. 2019)	1.05E-01±9.74E-02	4.36E-27±3.18E-31	0.00E+00±0.00E+00	−1.87E+02±2.94E-03	5.01E-04±9.57E-05
WOA (Mirjalili and Lewis 2016)	**3.48E-12±9.71E-02**	2.97E+67±5.20E-74	0.00E+00±0.00E+00	−1.87E+02±6.65E-05	3.95E-05±5.89E-09
HHO (Rao et al. 2012)	1.28E-07±3.97E-04	6.03E-55±8.13E-57	0.00E+00±0.00E+00	−1.87E+02±3.71E-10	1.46E-05±1.11E-05
HGSA (Wang et al. 2019)	1.77E-09±6.97E-08	7.11E-75±3.17E-78	0.00E+00±0.00E+00	−1.87E+02±5.22E-07	1.51E-21±6.07E-19
ACVO	6.35E-12±7.38E-11	**9.04E-76±5.00E-74**	0.00E+00±0.00E+00	**−1.87E+02±5.67E-08**	**1.00E-08±3.40E-10**

Bold values indicate the best results generated by the algorithms

Table 15 Results obtained by comparison algorithms on test functions with 500 dimensions

Algorithm	F_1	F_2	F_3	F_4	F_5
SADE (Qin and Suganthan 2005)	3.60E-59±5.40E-70	3.60E-56±1.50E-49	2.36E-03±4.50E-03	9.24E-01±8.75E-02	2.50E-41±2.79E-44
ABC (Karaboga and Basturk 2007)	3.50E-17±1.90E-23	7.79E-48±3.00E-34	1.50E-03±4.80E-05	9.90E-01±1.45E-01	4.20E-20±3.60E-35
PSOGSA (Mirjalili and Hashim 2010)	1.56E-23±4.58E-24	3.60E-27±7.40E-25	1.50E-03±1.94E-04	3.57E-01±1.24E-04	3.60E-03±6.58E-06
TLBO (Heidari et al. 2019)	3.56E-60±5.14E-09	9.71E-22±9.37E-15	4.65E-02±5.13E-04	**3.48E-01±3.14E-05**	4.37E-35±4.00E-10
WOA (Mirjalili and Lewis 2016)	6.33E-67±6.05E-54	3.57E-32±1.96E-33	9.00E-04±8.70E-04	9.53E-01±3.25E-02	1.59E-48±9.87E-20
HHO (Rao et al. 2012)	5.77E-83±3.72E-96	4.59E-40±2.97E-34	4.79E-03±2.15E-01	3.49E+00±8.74E-05	6.78E-13±4.56E-02
HGSA (Wang et al. 2019)	**2.90E-85±1.36E-81**	3.17E-53±3.85E-49	8.55E-03±3.62E-04	6.35E-01±4.70E-02	6.00E-23±7.80E-10
ACVO	6.34E-79±9.54E-80	**6.11E-81±1.28E-79**	**1.30E-04±1.28E-05**	8.54E-01±1.55E-02	**5.65E-75±2.74E-91**
	F_6	F_7	F_8	F_9	F_{10}
SADE (Qin and Suganthan 2005)	**1.51E-01±1.16E-02**	3.56E-12±8.00E-17	5.50E-22±7.00E-15	−1.80E+02±3.15E-01	3.26E-04±2.36E-05
ABC (Karaboga and Basturk 2007)	5.79E-01±2.59E-02	4.50E-13±5.31E-09	5.80E-06±6.00E-06	−1.59E+02±2.48E-01	5.40E-04±4.78E-03
PSOGSA (Mirjalili and Hashim 2010)	5.71E+00±9.46E-02	4.58E-28±4.57E-27	0.00E+00±0.00E+00	−1.87E+02±4.86E-04	1.00E+00±2.00E-03
TLBO (Heidari et al. 2019)	8.74E-01±4.20E-03	3.29E-45±7.12E-60	0.00E+00±0.00E+00	−1.77E+02±7.90E+01	5.90E-03±4.58E-02
WOA (Mirjalili and Lewis 2016)	3.64E-01±2.50E-05	0.00E+00±0.00E+00	0.00E+00±0.00E+00	−1.87E+02±3.54E-12	5.00E-03±8.00E-04
HHO (Rao et al. 2012)	1.57E-01±3.54E-03	5.43E-09±1.20E-06	6.46E-08±8.76E-06	−1.87E+02±4.73E-14	1.10E+00±4.00E-04
HGSA (Wang et al. 2019)	9.36E-01±4.00E-02	6.74E-25±5.76E-32	8.80E-14±3.20E-13	−1.87E+02±2.47E+00	7.64E-06±2.14E-04
ACVO	1.53E-01±5.10E-02	**0.00E+00±0.00E+00**	0.00E+00±0.00E+00	−1.87E+02±5.26E-04	**5.16E-06±2.65E-04**

Bold values indicate the best results generated by the algorithms

Table 16 Results obtained by comparison algorithms on test functions with 1000 dimensions

Algorithm	F_1	F_2	F_3	F_4	F_5
SADE (Qin and Suganthan 2005)	2.07E-32±3.30E-44	2.90E-42±1.45E-43	1.62E+00±1.14E+00	3.26E+03±4.55E+01	8.00E+04±6.74E+02
ABC (Karaboga and Basturk 2007)	9.65E-23±8.60E-35	1.70E-17±5.40E-11	2.50E+00±1.13E-01	1.49E+03±3.62E+02	9.09E+03±2.87E+02
PSOGSA (Mirjalili and Hashim 2010)	1.96E-25±1.35E-23	3.29E-28±6.00E-33	8.40E-01±4.50E-02	8.34E-01±5.78E-03	3.18E+03±5.32E+02
TLBO (Heidari et al. 2019)	8.09E-17±39.64-16	3.67E-64±3.45E-81	2.15E-01±8.47E-02	3.50E-01±1.90E-02	6.61E-04±1.25E-04
WOA (Mirjalili and Lewis 2016)	1.09E-66±5.56E-51	9.02E-55±5.36E-80	2.40E-03±6.62E-03	3.55E+00±4.00E-03	8.49E+03±9.43E+02
HHO (Rao et al. 2012)	9.90E-13±7.89E-08	7.90E-16±6.88E-06	9.23E-02±2.27E-01	3.50E-02±6.33E-01	5.72E-03±9.21E-03
HGSA (Wang et al. 2019)	5.16E-31±1.18E-26	2.94E-51±4.61E-32	3.44E-03±8.55E-03	6.70E-02±4.50E-02	4.10E-19±6.57E-15
ACVO	**3.88E-74±8.55E-70**	**2.54E-76±6.91E-86**	**9.24E-04±5.00E-04**	**3.20E-03±2.75E-02**	**6.03E-45±1.04E-39**
	F_6	F_7	F_8	F_9	F_{10}
SADE (Qin and Suganthan 2005)	3.97E+08±1.51E+06	2.53E-06±1.10E-10	5.64E-01±1.47E-04	−1.68E+02±2.43E+01	1.93E+00±8.51E-01
ABC (Karaboga and Basturk 2007)	9.82E+08±4.33E+07	1.16E-10±7.10E-05	2.57E-01±9.80E-03	−1.80E+02±8.87E+00	7.55E+06±8.02E+03
PSOGSA (Mirjalili and Hashim 2010)	4.98E+02±3.00E-01	3.50E-10±8.90E-09	0.00E+00±0.00E+00	−1.87E+02±8.46E-02	3.29E+00±1.07E-01
TLBO (Heidari et al. 2019)	9.33E-01±2.73E-01	6.50E-16±7.10E-21	0.00E+00±0.00E+00	1.81E+02±2.05E-01	6.31E+00±1.26E-01
WOA (Mirjalili and Lewis 2016)	9.05E+02±3.88E-01	3.50E-14±1.15E-13	0.00E+00±0.00E+00	−1.87E+02±2.94E-06	7.12E-01±3.01E-03
HHO (Rao et al. 2012)	4.94E+02±3.76E-02	7.39E-14±6.50E-15	6.46E-08±8.76E-06	−1.87E+02±2.04E-11	8.05E-01±4.42E-01
HGSA (Wang et al. 2019)	5.33E-01±2.78E+01	5.50E-25±3.24E-29	8.80E-14±3.20E-13	1.73E+02±2.65E+01	3.77E-05±7.00E-03
ACVO	**4.51E-01±4.02E-01**	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	−1.66E+02±3.65E+01	**1.97E-05±2.75E-05**

Bold values indicate the best results generated by the algorithms

$$\begin{aligned}
&\text{minimize } f = \sum_{i=1}^{N_G} a_i P_i^2 + b_i P_i + c_i \\
&\text{subject to} \\
&\quad \sum_{i=1}^{N_G} P_i = P_D + P_L \\
&\quad P_i^{\min} \le P_i \le P_i^{\max} \\
&\quad \max(P_i^{\min}, UR_i - P) \le P_i \le \min(P_i^{\max}, P_i^{t-1} - DR_i) \\
&\quad P_i \le \breve{P}^{pz} \text{ and } P_i \ge \hat{P}^{pz}
\end{aligned} \tag{26}$$

Table 18 summarizes the best, mean, worst, and standard deviation of objective function values reported by ACVO and counterparts over 30 independent runs. In simulations, the algorithms independently execute 30 times, and the statistical results are reported. At each run, the initial population is formed using uniform random initialization within the predetermined search boundary of each test problem. In terms of *best* values, the ACVO performed better or had a similar performance to other algorithms in four problems. From the *mean* and *std* values reported by ACVO, one can conclude that it has a steady behavior in finding optimal solutions. The worst values found by ACVO are close to the mean values. This issue justifies that the ACVO can avoid local optimum points. The results given in Table 18 show that the ACVO can be applied to various practical and engineering problems.

3.5.5 Speed reducer design (SRD)

The speed reducer design problem is a popular benchmark in the structural optimization field. A graphical view of a speed reducer is shown in Fig. 10. The goal is to design a gearbox with the minimum weight, which is located between the propeller and engine in a light aircraft. In designing a speed reducer, four constraints should be considered that are stresses in the shafts, surface stress, bending stress of the gear teeth, and the transverse deflection of the shafts (Askari et al. 2020). The following equations show the mathematical definition of the problem.

$$\begin{aligned}
&\text{minimize} \quad f(\mathbf{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - \\
&\qquad 1.508x_1(x_6^2+x_7^2) + 7.4777(x_6^3+x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \\
&\text{subject to} \quad g_1(\mathbf{x}) = \frac{27}{x_1x_2^2x_3} - 1 \le 0, \quad g_2(\mathbf{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0, \\
&\qquad g_3(\mathbf{x}) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \le 0, \quad g_4(\mathbf{x}) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \le 0, \\
&\qquad g_5(\mathbf{x}) = \frac{1}{110x_6^3}\sqrt{\left(\frac{745x_4}{110x_2x_3}\right)^2 + 16.9 \times 10^6} - 1 \le 0 \\
&\qquad g_6(\mathbf{x}) = \frac{1}{85x_7^3}\sqrt{\left(\frac{745x_5}{110x_2x_3}\right)^2 + 157.5 \times 10^6} - 1 \le 0 \\
&\qquad g_7(\mathbf{x}) = \frac{x_2x_3}{40} - 1 \le 0, \quad g_8(\mathbf{x}) = \frac{5x_2}{x_1} - 1 \le 0 \\
&\qquad g_9(\mathbf{x}) = \frac{x_1}{12x_2} - 1 \le 0, \quad g_{10}(\mathbf{x}) = \frac{15x_6+1.9}{x_4} - 1 \le 0 \\
&\qquad g_{11}(\mathbf{x}) = \frac{1.1x_7+1.9}{x_5} - 1 \le 0 \\
&\text{where} \quad 2.6 \le x_1 \le 3.6, 0.7 \le x_2 \le 0.8, 17 \le x_3 \le 28, \\
&\qquad 7.3 \le x_4 \le 8.3, 7.3 \le x_5 \le 8.3, 2.9 \le x_6 \le 3.9, \\
&\qquad 5.0 \le x_7 \le 5.5
\end{aligned} \tag{27}$$

As shown in Table 19, the results show that ACVO obtains the best results. With a slight difference from ACVO, HHO attains the second rank. HGSA obtains the third rank, and the last rank belongs to ABC. The results suggest that ACVO is a proper choice to design the speed reducer.

3.5.6 Welded beam design (WBD)

As shown in Fig. 11, the objective of the EBD is to decrease the manufacturing cost considering constraints on bending stress (σ), buckling load (P_c), shear stress (τ), deflection (δ), and side constraints. The problem is formulated as follows:

$$\begin{aligned}
&\text{minimize} \quad f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \\
&\text{subject to} \quad c_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \le 0, \\
&\qquad c_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\max} \le 0, \\
&\qquad c_3(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \le 0, \\
&\qquad c_4(\vec{x}) = x_1 - x_4 \le 0, \\
&\qquad c_5(\vec{x}) = P - P_c(\vec{x}) \le 0, \\
&\qquad c_6(\vec{x}) = 0.125 - x_1 \le 0, \\
&\qquad c_7(\vec{x}) = 1.10471x_1^2 + 0.0811x_3x_4(14.0 + x_2) - 5.0 \le 0
\end{aligned} \tag{28}$$

$$\begin{aligned}
&\vec{x} = [x_1, x_2, x_3, x_4] = [h, l, t, b] \quad \text{where} \quad 0.1 \le x_1 \le 2, \\
&\qquad 0.1 \le x_2 \le 10, \\
&\qquad 0.1 \le x_3 \le 10, \\
&\qquad 0.1 \le x_4 \le 2
\end{aligned} \tag{29}$$

where $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$ are the constraints of the WBD problem, b is the width of the beam, t is the height of the beam, l is the length of the weld, and h is the thickness of weld. The objective is to find optimal values for variables $\{h, l, t, b\}$. The other variables are defined below:

$$\begin{aligned}
&\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \\
&\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, \mathrm{M} = \mathrm{P}(\mathrm{L}+\tfrac{x_2}{2}), \\
&R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2}, \\
&J = 2\{\sqrt{2}x_1x_2[\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2]\}, \\
&\sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}, \delta(\vec{x}) = \frac{6PL^3}{ex_3^2x_4}, \\
&P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \\
&P = 6000lb, \mathrm{L} = 14\mathrm{in}, \delta_{\max} = 0.25in, \\
&\mathrm{E} = 30 \times 10^6 psi, \mathrm{G} = 12 \times 10^6 psi, \\
&\tau_{\max} = 13600psi, \sigma_{\max} = 30000psi
\end{aligned} \tag{30}$$

Table 20 reports the results generated by the comparison algorithms on the WBD problem. The results prove that the ACVO attains better results compared with counterpart algorithms.

3.5.7 Rolling element bearing design (REBD)

A graphical representation of a REBD problem is shown in Fig. 12. This maximization problem consists of 9 design con-

Table 17 Characteristics of FMSW, SELD, TNEP, and SSRPCD engineering problems

Problem	Dimension	Constraints	Bounds
FMSW	6	Bound constrained	[-6.4, 6.35]
SSRPCD	20	Bound constrained	$[0, 2\pi]$
TNEP	7	Equality and inequality constraints	[0, 15]
SELD	13	Inequality constraints	[0, 680; 0, 360; 0, 360; 60, 180; 60, 180; 60, 180; 60, 180; 60, 180; 60, 180; 40, 120; 40, 120; 55, 120; 55, 120]

straints and 10 geometric variables to handle the geometric-based and assembly restrictions. The mathematical definition of the problem is as follows (Emami 2021):

maximize

$$C_d = f_c Z^{2/3} D_b^{1.8} \quad \text{if}(D \le 25.4mm)$$
$$C_d = 3.647 f_c Z^{2/3} D_b^{1.4} \quad \text{if}(D > 25.4mm)$$

subject to

$$g_1(\mathbf{z}) = \frac{\phi_0}{2\sin^{-1}(D_b/D_m)} - Z + 1 \le 0, \quad g_2(\mathbf{z}) = 2D_b - K_{D\min}(D-d) > 0,$$
$$g_3(\mathbf{z}) = K_{D\max}(D-d) - 2D_b \ge 0, \quad g_4(\mathbf{z}) = \xi B_w - D_b \le 0,$$
$$g_5(\mathbf{z}) = D_{\mathrm{m}} - 0.5(D+d) \ge 0, \quad g_6(\mathbf{z}) = (0.5+e)(D+d) - D_m \ge 0,$$
$$g_7(\mathbf{z}) = 0.5(D - D_m - D_b) - \varepsilon D_b \ge 0 \quad g_8(\mathbf{z}) = f_i \ge 0.515,$$
$$g_9(\mathbf{z}) = f_o \ge 0.515$$

where

$$f_c = 37.91\left[1 + \left\{1.04\left(\frac{1-\gamma}{1+\gamma}\right)^{1.72}\left(\frac{f_i(2f_0-1)}{f_o(2f_i-1)}\right)^{0.41}\right\}^{10/3}\right]^{-0.3} \times \left[\frac{\gamma^{0.3}(1-\gamma)^{1.39}}{(1-\gamma)^{1/3}}\right]\left[\frac{2f_i}{2f_i-1}\right]^{0.41} \tag{31}$$
$$x = \left[\{(D-2)/2 - 3(T/4)\}^2 + \{D/2 - T/4 - D_b\}^2 - \{d/2 + T/4\}^2\right]$$
$$y = 2\{(D-d)/2 - 3(T/4)\}\{(D/d) - T/4 - D_b\}$$
$$\phi_o = 2\prod -\cos^{-1}\left(\frac{x}{y}\right)$$
$$\gamma = \frac{D_b}{D_m}, f_i = \frac{r_i}{D_b}, f_o = \frac{r_i}{D_b}, T = D - d - 2D_b$$
$$D = 160, d = 90$$
$$B_w = 30, r_i = r_o = 11.0330.5(\mathrm{D} + \mathrm{d}) \le \mathrm{D}_m \le 0.6(D+d),$$
$$0.15(\mathrm{D} - \mathrm{d}) \le \mathrm{D}_b \le 0.45(D-d), 4 \le Z \le 50, 0.515 \le f_i \text{and} f_o \le 0.6,$$
$$0.4 \le K_{D\min} \le 0.5,$$
$$0.6 \le K_{D\max} \le 0.7, 0.3 \le e \le 0.4, 0.02 \le \mathrm{e} \le 0.1,$$
$$0.6 \le \xi \le 0.85$$

The results generated by ACVO and counterpart algorithms on REBD problems are summarized in Table 21. The results prove that the ACVO shows the best design and attains better performances compared with other algorithms. HGSA takes the second rank, and TLBO obtains the third rank in terms of C_d parameter.

3.6 Computational complexity

The time complexity of the ACVO depends on three main steps: initialization, fitness calculation, and updating of individuals. To be very precise, the time complexity of each phase in ACVO is computed as follows:

The parameter initialization phase needs the time complexity $O(1)$.

The population initialization phase needs $O(N)$, where N is the population size.

Computing the fitness of all individuals costs $O(N)$.

Table 18 Results of ACVO and counterpart algorithms on four real-world engineering problems

Problem	Statistics	SADE (Qin and Suganthan 2005)	ABC (Karaboga and Basturk 2007)	PSOGSA (Mirjalili and Hashim 2010)	TLBO (Rao et al. 2012)	WOA (Mirjalili and Lewis 2016)	HHO (Heidari et al. 2019)	HGSA (Wang et al. 2019)	ACVO
FMSW	Best	**0.00E+00**	5.99E-03	**0.00E+00**	**0.00E+00**	6.20E-02	7.97E-05	3.55E-13	**0.00E+00**
	Mean	**1.49E+00**	2.99E+00	3.18E+00	2.60E+00	2.71E+00	2.32E+00	2.11E+00	**1.49E+00**
	Std	3.88E+00	4.51E+00	4.73E+00	4.87E+00	**1.80E+00**	3.15E+00	4.05E+00	3.88E+00
	Worst	**4.14E+00**	5.35E+00	8.51E+00	6.19E+00	5.97E+00	5.47E+00	6.52E+00	5.68E+00
SSRPCD	Best	5.05E-01	7.32E-01	6.05E- 01	6.21E-01	6.27E-01	5.48E-01	5.00E-01	**5.00E-01**
	Mean	8.05E-01	1.04E+00	1.14E+00	1.10E+00	1.37E+00	1.13E+00	**7.05E-01**	7.48E-01
	Std	3.80E-01	**1.00E-01**	2.42E-01	3.33E-01	3.57E-01	3.10E-01	1.12E-01	1.42E-01
	Worst	1.28E+00	1.61E+00	1.60E+00	1.67E+00	1.91E+00	1.64E+00	**8.93E-01**	1.03E+00
TNEP	Best	**2.20E+02**	**2.20E+02**	**2.20E + 02**	2.25E+02	2.26E+02	**2.20E+02**	**2.20E + 02**	**2.20E+02**
	Mean	**2.20E+00**	2.31E+00	2.34E + 02	2.62E+02	2.45E+02	2.21E+02	**2.20E + 02**	**2.20E+02**
	Std	**0.00E+00**	1.56E+01	3.81E + 01	2.18E+01	5.83E+01	1.40E+00	**0.00E + 00**	**0.00E+00**
	Worst	**2.20E+02**	2.35E+02	3.51E + 02	2.69E+02	2.75E+02	2.22E+02	**2.20E + 02**	**2.20E+02**
SELD	Best	1.87E+04	1.90E+04	1.94E+04	1.88E+04	1.89E+04	1.86E+04	1.89E+04	**1.85E+04**
	Mean	1.92E+04	1.95E+04	1.94E+04	1.94E+04	1.94E+04	1.92E+04	1.91E+04	**1.90E+04**
	Std	3.48E+02	2.53E+02	1.60E+02	3.81E+02	2.66E+02	3.30E+02	**1.14E+02**	3.75E+02
	Worst	1.98E+04	1.99E+04	1.96E+04	1.97E+04	1.98E+04	**1.92E+04**	1.94E+04	1.95E+04

Bold values indicate the best results generated by the algorithms

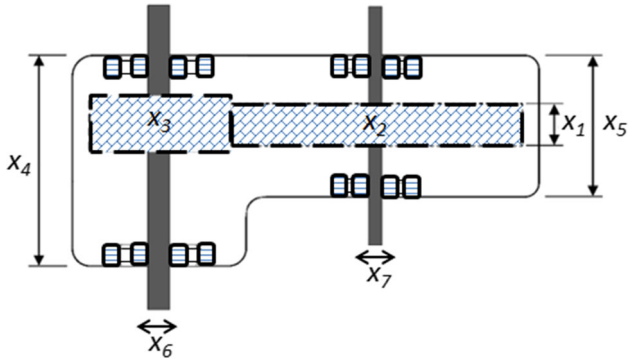


Fig. 10 A graphical view of a speed reducer (Askari et al. 2020)

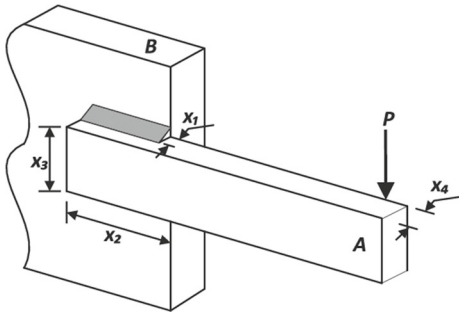


Fig. 11 Welded beam design problem (Emami 2020)

Controlling the boundary of individuals' variables costs $O(n)$.
The time complexity of social distancing phase is $O(n^2)$.
Quarantine phase needs the time complexity $O(|Q|)$, where $|Q| < N$.
Isolation phase is of order $O(|I|)$, where $|I| < N$.

The overall time complexity of ACVO in each generation in the worst case is as follows:

$$O(1) + O(N) + O(N) + O(N) + O(N^2) + O(|Q|) + O(|I|) = O(1) + 3O(N) + O(N^2) + O(|Q|) + O(|I|) \approx O(N^2) \quad (32)$$

Thus, the overall time complexity of ACVO is $O(N^2)$. The time complexity of SADE, ABC, PSOGSA, TLBO, WOA, HHO, and HGSA is $O(N^2)$ in the worst case. The time complexity of ACVO is the same as its counterparts. This shows

Table 19 Comparison of results generated by algorithms on speed reducer design problem

Algorithm	Problem parameters							Optimal cost
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
SADE (Qin and Suganthan 2005)	3.5000	0.7	17	7.3000	8.300000	3.350215	5.286859	3007.4368
ABC (Karaboga and Basturk 2007)	3.5000	0.7	17	8.3000	8.300000	3.352207	5.286859	3016.7705
PSOGSA (Mirjalili and Hashim 2010)	3.5000	0.7	17	8.3000	7.715381	3.352210	5.286659	3003.8110
TLBO (Rao et al. 2012)	3.5000	0.7	17	7.3000	8.015279	3.350215	5.286758	3001.1214
WOA (Mirjalili and Lewis 2016)	3.5005	0.7	17	7.3000	7.766471	3.352840	5.286887	2996.6212
HHO (Heidari et al. 2019)	3.5000	0.7	17	7.3065	7.715439	3.350227	5.286655	2994.5342
HGSA (Wang et al. 2019)	3.5000	0.7	17	7.3000	7.721984	3.350215	5.286657	2994.6192
ACVO	3.5000	0.7	17	7.3000	7.715335	3.350215	5.286655	**2994.4718**

Table 20 Results obtained by the algorithms on welded beam design problem

Algorithm	Problem parameters				Optimum cost
	h	l	t	b	
SaDE (Qin and Suganthan 2005)	3.06E-01	3.02E+00	6.33E+00	4.19E-01	2.48E+00
ABC (Karaboga and Basturk 2007)	2.79E-01	2.74E+00	7.79E+00	2.79E-01	1.99E+00
PSOGSA (Mirjalili and Hashim 2010)	2.40E-01	3.09E+00	8.36E+00	2.40E-01	1.85E+00
TLBO (Rao et al. 2012)	2.28E-01	3.20E+00	8.57E+00	2.28E-01	1.81E+00
WOA (Mirjalili and Lewis 2016)	2.04E-01	3.43E+00	9.25E+00	2.05E-01	1.74E+00
HHO (Heidari et al. 2019)	2.04E-01	3.53E+00	9.03E+00	2.06E-01	**1.73E+00**
HGSA (Wang et al. 2019)	2.11E-01	3.40E+00	8.90E+00	2.12E-01	1.75E+00
ACVO	2.05E-01	3.48E+00	9.04E+00	2.06E-01	**1.73E+00**

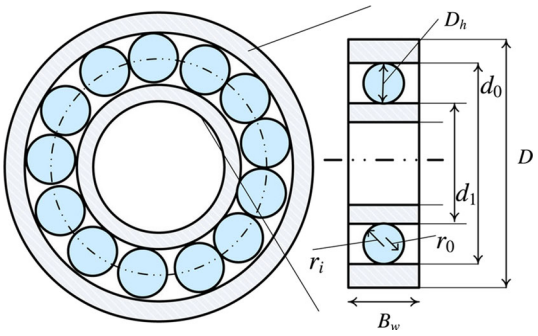


Fig. 12 Rolling element bearing problem (Emami 2021)

that the ACVO is computationally efficient in comparison with other algorithms.

We performed a test on 1000 dimensions scalable problems F_1-F_{10} to show the execution time consumed by algorithms to reach the global optimum. As shown in Fig. 13, it is clear that ACVO consumes less execution time than other algorithms in most benchmark functions.

4 Conclusion

In this paper, a novel anti-coronavirus optimization (ACVO) algorithm is introduced to solve single-objective optimization problems. This algorithm simulates the protocols recommended by the world health organization to prevent the spreading of coronavirus. It consists of three main operators including social distancing, quarantine, and isolation. These operators hopefully guide the individuals in the population to converge to the global optimum point in solution space. ACVO is relatively easy to implement and customize for various real-world applications. The effectiveness of the proposed algorithm is compared with several state-of-the-art methods on a set of 28 test functions, which covers a wide variety of different problems. The results demonstrate that ACVO obtains outstanding performance in solving single-objective optimization problems. There remain several works to further improve the performance of ACVO. New operators such as vaccination can be embedded in the algorithm to enhance its search power. A multi-objective version of the algorithm should be investigated for solving problems with multiple objectives. The parameters of the algorithm need to be further analyzed to be tuned with optimal values. The

Table 21 Comparison of results for rolling element bearing design problem

Algorithms	SADE	ABC	PSOGSA	TLBO	WOA	HGSA	HHO	ACVO
D_m	125	127.393727	125.0085325	125.6830297	125.1007341	125.7080059	125.305428	125.7095901
D_b	21.42330094	20.37891402	21.11263796	21.42330091	21.42330001	21.42330054	21.41961507	21.42329966
Z	10.94309732	11.55679004	11.06226678	10.99797096	10.95119042	10.99997761	10.96902407	11.00010435
f_i	0.515	0.515	0.515	0.515	0.515	0.515	0.515	0.515
f_o	0.515	0.53271438	0.519599291	0.515	0.515	0.515	0.515088575	0.515
K_{Dmin}	0.5	0.491963843	0.404876429	0.454384155	0.4	0.5	0.4	0.483526982
K_{Dmax}	0.69522976	0.647640879	0.603250134	0.624561334	0.7	0.7	0.646640076	0.618218965
ε	0.3	0.3	0.300000011	0.300036133	0.314216016	0.300304057	0.3	0.300275339
e	0.023301906	0.020595704	0.1	0.02	0.02	0.027109778	0.1	0.02
ξ	0.626576428	0.645102744	0.703712695	0.607771799	0.6	0.6	0.70453363	0.647881738
C_d	85220.6809	80900.6816	83650.9164	85521.743	85265.167	85532.7227	85336.7584	**85533.4103**

Bold values indicate the best results generated by the algorithms

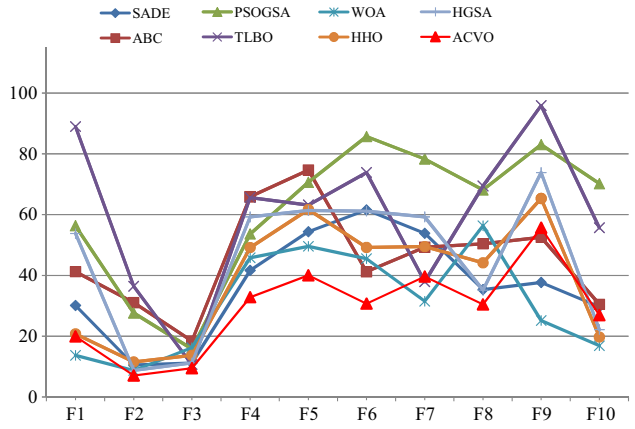


Fig. 13 Comparison of the execution time of algorithms in 1000 dimensions unimodal and multimodal functions

algorithm should be applied to various engineering problems to assess its disadvantages and potentials.

Author Contributions This paper presents a new swarm intelligence method, namely anti coronavirus optimization algorithm. This algorithm obtains better performance compared with other optimization algorithms.

Funding There is no funding information.

Availability of data and material There are no data for distribution.

Code Availability The source code is submitted as a electronic supplementary material.

Declarations

Conflict of Interest H. Emami declares that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Abdel-Basset M, Abdel-Fatah L, Sangaiah AK (2018) Metaheuristic algorithms: a comprehensive review. Elsevier Inc

Al-Betar MA, Awadallah MA, Doush IA, Hammouri AI, Mafarja M, Alyasseri ZA (2012) Flower pollination algorithm for global optimization, In: International conference on unconventional computing and natural computation, Springer, Berlin, 240–249

Anderson RM, Heesterbeek H, Klinkenberg D, Hollingsworth TD (2020) How will country-based mitigation measures influence the course of the COVID-19 epidemic? Lancet 395(10228):931–934

Askari Q, Younas I, Saeed M (2020) Political optimizer: a novel socio-inspired meta-heuristic for global optimization. Knowl-Based Syst 195:105709

Askari Q, Saeed M, Younas I (2020) Heap-based optimizer inspired by corporate rank hierarchy for global optimization. Expert Syst Appl 161:113702

Boussaïd I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. Inf Sci 237:82–117

Cheng MY, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112

Civicioglu P (2013) Backtracking search optimization algorithm for numerical optimization problems. Appl Math Comput 219(15):8121–8144

Das S, Suganthan PN (2012) Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Jadavpur University, Nanyang Technological University, Kolkata

Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18

Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1:28–39

Eita MA, Fahmy MM (2014) Group counseling optimization. Appl Soft Comput J 22:585–604

Emami H (2020) Seasons optimization algorithm. Eng Comput 123456789:1–21

Emami H (2021) (2021) Stock exchange trading optimization algorithm: a human-inspired method for global optimization. J Supercomput. https://doi.org/10.1007/s11227-021-03943-w

Emami H, Derakhshan F (2015) Election algorithm: a new socio-politically inspired strategy. AI Commun 28(3):591–603

Ferreira C (2001) Gene expression programming: a new adaptive algorithm for solving problems. Complex Syst 13(2):87–129

Gandomia AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17(12):4831–4845

Ghaemia M, Feizi-Derakhshi M-R (2014) Forest optimization algorithm. Expert Syst Appl 41(15):6676–6687

Guan W et al (2020) Clinical characteristics of coronavirus disease 2019 in China. N Engl J Med 382(18):1708–1720

Haupt RL, Haupt RL (2004) a J. Wiley, Algorithms Practical Genetic Algorithms, Second Edi. A JOHN WILEY & SONS, INC., PUBLICATION

Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. Futur Gener Comput Syst 97:849–872

Igel C, Hansen N, Roth S (2007) Covariance matrix adaptation for multi-objective optimization. Evol Comput 15(1):1–28

Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39(3):459–471

Kaveh A, Farhoudi N (2013) A new optimization method: dolphin echolocation. Adv Eng Softw 59:53–70

Kennedy J, Eberhart R (1995) Particle swarm optimization, Proceedings of ICNN'95 - International Conference on Neural Networks, 1942–1948

Krause J, Cordeiro J (2013) A survey of swarm algorithms applied to discrete optimization problems, In: Swarm intelligence and bio-inspired computation, Elsevier, pp. 169–191

Kuo HC, Lin CH (2013) Cultural evolution algorithm for global optimizations and its applications. J Appl Res Technol 11(4):510–522

Li R et al (2020) Substantial undocumented infection facilitates the rapid dissemination of novel coronavirus (SARS-CoV-2). Science (80-.) 368(6490):489–493

Li X, Zhang J, Yin M (2014) Animal migration optimization: an optimization algorithm inspired by animal migration behavior. Neural Comput Appl 24(7–8):1867–1877

Liang YC, Cuevas Juarez JR (2020) A self-adaptive virus optimization algorithm for continuous optimization problems. Soft Comput 24(17):13147–13166

Liang YC, CuevasJuarez JR (2016) A novel metaheuristic for continuous optimization problems: virus optimization algorithm. Eng Optim 48(1):73–93

Liu Y, Gayle AA, Wilder-Smith A, Rocklöv J (2020) The reproductive number of COVID-19 is higher compared to SARS coronavirus. J Travel Med 27(2):1–4

Maier BF, Brockmann D (2020) Effective containment explains subexponential growth in recent confirmed COVID-19 cases in China. Science (80) 368(6492):742–746

Matrajt L, Leung T (2020) Evaluating the effectiveness of social distancing interventions to delay or flatten the epidemic curve of coronavirus disease. Emerg Infect Dis 26(8):1740–1748

Merrikh-Bayat F (2015) The runner-root algorithm: a metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature. Appl Soft Comput J 33:292–303

Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67

Mirjalili S, Mohammad S, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61

Mirjalili S, Gandomi AH, Zahra S, Saremi S (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:1–29

Mirjalili S, Hashim SZM (2010) A new hybrid PSOGSA algorithm for function optimization, In: Proceedings of ICCIA 2010 - 2010 international conference on computer and information application, 1, 374–377

Mo H, Xu L (2013) Magnetotactic bacteria optimization algorithm for multimodal optimization, In: Proceedings of the 2013 IEEE symposium swarm intelligence. SIS 2013 - 2013 IEEE symposium series on computational intelligence SSCI 2013, pp. 240–247

Pan WT (2012) A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowl-Based Syst 26:69–74

Pan X, Jiao L (2011) A granular agent evolutionary algorithm for classification. Appl Soft Comput J 11(3):3093–3105

Passino KM, Ohio T (2010) Bacterial foraging optimization, 1, 1–16

Price KV, Awad NH, Ali MZ, Suganthan PN (Nov. 2018) The 100-digit challenge: problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization, School Elect. Electron. Eng., Nanyang Technol. Univ., Singapore, Tech. Rep

Qin AK, Suganthan PN (2005) Self-adaptive differential evolution algorithm for numerical optimization. IEEE Trans Evol Comput 1(3):1785–1791

Rao RV, Savsani VJ, Vakharia DP (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. CAD Comput Aided Des 43(3):303–315

Rao RV, Savsani VJ, Vakharia DP (2012) Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems. Inf Sci 183(1):1–15

Sanche S, Lin YT, Xu C, Romero-Severson E, Hengartner N, Ke R (2020) Research high contagiousness and rapid spread of severe acute respiratory syndrome coronavirus 2. Emerg Infect Dis 26(7):1470–1477

Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47

Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713

Storn R, Price K (1997) Differential evolution- a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359

Suganthan P, Ali M, Wu G, Mallipeddi R (2018) Special session & competitions on real-parameter single objective optimization, In: Proceedings of the IEEE congress on evolutionary computation (CEC), Rio de Janeiro, Brazil, Rep., Jul 2018

Swanson J, Jeanes A (2007) Title community: a pragmatic approach. Clin Focus 16(6):282–289

Taylor P, Mezura-montes E, Coello CAC (2008) An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 37(4):37–41

Wang D et al (2020) Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirus-infected pneumonia in Wuhan, China. JAMA - J Am Med Assoc 323(11):1061–1069

Wang Y, Yu Y, Gao S, Pan H, Yang G (2019) A hierarchical gravitational search algorithm with an effective gravitational constant. Swarm Evol Comput 46(February):118–139

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82

Wu JT, Leung K, Leung GM (2020) Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: a modelling study. Lancet 395(10225):689–697

Yamato M, Fujiwara A (2019) A strawberry optimization algorithm for the multi-objective knapsack problem. Bull Netw, Comput, Syst, Softw 8(2):129–132

Yang XS (2009) "Firefly algorithms for multimodal optimization," Lecture notes in computer science (including Subser. lecture notes in artifficial intelligence lecture notes bioinformatics), 5792 LNCS, 169–178

Yang XS (2010) A new metaheuristic bat-inspired algorithm, In: Nature inspired cooperative strategies for optimization (NICSO 2010), 65–74

Yang X, Deb S (2009) Cuckoo Search via Levy Flights, In 2009 world congress on nature & biologically inspired computing (NaBIC 2009), pp. 210–214

Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3(2):82–102

Yazdani M, Jolai F (2016) Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. J Comput Des Eng 3(1):24–36

Ye M et al (2020) Treatment with convalescent plasma for COVID-19 patients in Wuhan, China, J Med Virol, 0–1

Yu JJQ, Li VOK (2015) A social spider algorithm for global optimization. Appl Soft Comput J 30:614–627