Opposition-based learning multi-verse optimizer with disruption operator for optimization problems

Mohammad Shehab · Laith Abualigah

Accepted: 25 August 2022 / Published online: 12 September 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract
Multi-verse optimizer (MVO) algorithm is one of the recent metaheuristic algorithms used to solve various problems in different fields. However, MVO suffers from a lack of diversity which may trapping of local minima, and premature convergence. This paper introduces two steps of improving the basic MVO algorithm. The first step is using opposition-based learning (OBL) in MVO, called OMVO. The OBL aids to speed up the searching and improving the learning technique for selecting a better generation of candidate solutions of basic MVO. The second stage, called OMVOD, combines the disturbance operator (DO) and OMVO to improve the consistency of the chosen solution by providing a chance to solve the given problem with a high fitness value and increase diversity. To test the performance of the proposed models, fifteen CEC 2015 benchmark functions problems, thirty CEC 2017 benchmark functions problems and seven CEC 2011 real-world problems were used in both phases of the enhancement. The second step, known as OMVOD, incorporates the disruption operator (DO) and OMVO to improve the accuracy of the chosen solution by giving a chance to solve the given problem with a high fitness value while also increasing variety. Fifteen CEC 2015 benchmark functions problems, thirty CEC 2017 benchmark functions problems and seven CEC 2011 real-world problems were used in both phases of the upgrade to assess the accuracy of the proposed models.

Keywords Multi-verse optimizer · Opposition-based learning · Disruption operator · CEC2015 and CEC2017 benchmark functions problems · CEC2011 real-world problems

1 Introduction

Population-based algorithms and local search-based algorithms are examples of metaheuristic algorithms (Shehab et al. 2019; Altabeeb et al. 2021). Particle swarm optimization (PSO) Kennedy and Eberhart (1995) and ant colony optimization (ACO) Dorigo et al. (2006) are examples of population-based algorithms that work on a group of solutions at a time (Hassan et al. 2021). At each iteration, the best solutions were often used to add one or more new solutions. Hill climbing is one of the most popular local search-based algorithms (Abualigah et al. 2019). While local search-based algorithms focus on one solution and look to improve it using neighborhood solutions (Shehab et al. 2017a), hill climbing is one of the most common local search-based algorithms (Koziel and Yang 2011). The only disadvantage is that they choose to concentrate on exploitation over discovery, which raises the chances of being trapped in local optima (Abualigah et al. 2017). These approaches are suitable for finding promising areas in the search area. However, they are not so good at leveraging the search space field that is searched (Shehab et al. 2017b). Population-based approaches are classified as evolutionary computing and swarm intelligence (Abdelmadjid et al. 2013; Şahin and Abualigah 2021). Both approaches are focused on the natural biological evolution of natural creatures’ social interaction behavior. Particle swarm optimization is an example of a swarm-based algorithm (PSO) Kennedy and Eberhart (1995).

This work highlights a recent stochastic population-based algorithm, called multi-verse optimizer (MVO) Mirjalili et al. (2016). MVO is bio-inspired from the multi-verse theory...
in physics, where it includes unique concepts models such as white, black, and wormholes. These concepts increase the ability of the exploration and exploitation search mechanisms, as well as the diversity. Table 1 presents a brief description of the comparison between MVO and popular metaheuristic algorithms, though MOV suffers from slow searching, local minima and premature convergence (Shehab et al. 2017; Mirjalili et al. 2017).

Therefore, there are several methods applied to solve these drawbacks including the enhanced MVO proposed in ADHIM et al. (2019); the authors improved the basic MVO by introducing a new version, called EMVO, to achieve high accuracy and efficiency of the requirement prioritization. EMVO is based on exchanging the information between the current solutions. EMVO starts by calculating the warm hole probability. Then, it calculates the rate of the traveling space. However, EMVO suffers from minimum simplicity, so it needs the maximum number of functions. The results illustrated that the EMVO achieved better accuracy of the solutions comparing with different existing techniques.

Jangir et al. (2017) proposed a new technique, that is, hybridization particle swarm optimization (PSO) and multiverse optimizer (MVO), called HPSO-MVO, to determine the optimal reactive power dispatch (ORPD). The aim of HPSO-MVO is taking advantage of PSO to enhance the exploitation search, as well as the MVO to improve the exploration search in the uncertain environment (Shehab et al. 2016). The speed and locations of the particles are updated based on the position of the universe in each repetition. The experiment results show that the HPSO-MVO is outperformed the original MVO and PSO algorithm.

Geng et al. (2019) proposed an improved multi-objective multi-verse optimizer (IMOMVO) to find the optimal solution of the reentrant hybrid flow shop scheduling problem (RHFSP) with maximum tardiness, objectives of makespan and idle energy consumption. IMOMVO includes Latin hypercube sampling (LHS) for the initialization of the population. Then, it updated the position of candidate solutions based on Levy flight. Moreover, it utilized logical self-mapping to chaotic local search. The authors proved the performance of IMOMVO using a set of benchmark functions and comparing it with various techniques such as MOMVO, MOPSO, NSGA-II and MOALO.

Hu et al. (2016) employed the Levy flights with MVO, namely, LFMVO to find the optimal solution of the numerical and engineering optimization problems. The aim of utilizing the levy flights is to enhance the exploration search in the large-scale search space. The experiment results showed that the convergence speed and the quality of the solutions of LFMVO are outperformed various algorithms in the literature.

However, the drawback of these proposed methods is that they take into account just one factor to be enhanced (i.e., diversity, local minima and premature convergence). Also, most of the approaches that utilized an enhanced MVO algorithm contain more parameters that must be tuned. Therefore, the weaknesses of MVO still didn’t solve, perfectly. Thus, an alternative enhanced method for the MVO algorithm should be introduced to cover all limitations, completely.

This paper highlights the two main weaknesses recognized in the performance trajectory of the basic version of the MVO: diversity of the population, trapping of local minima and premature convergence. Because of these weaknesses, MVO requires further refinements, to be enhanced the procedure search process of the basic MVO and combinations with other techniques. The following points are summarized the main contributions of this work.

- A new version including MVO and opposition-based learning is developed to improve the premature convergence problem by optimizing the initial population.
- Applied the disruption operator to enhance the diversity of the population and the exploration ability.
- Evaluating the performance of the proposed algorithm by comparing it with other published methods in the literature using CEC 2015 benchmark function problems, CEC 2017 benchmark function problems and CEC 2011 real-world problems.

The following is how the sections of this paper are structured. Section 2 displays some similar works. The opposition-based learning (OBL) technique, the interruption operator (DO) and the multi-verse optimizer (MVO) algorithm are all introduced in Section 3. After that, in Sect. 4, the proposed methods are listed. The experimental findings and discussions are presented in Sect. 5. Section 6 concludes with recommendations for the future.

## 2 Related works

To improve the efficiency of the simple Krill Herd (KH) algorithm, Wang et al. (2016) used opposition-based learning (OBL), position clamping (PC) and Cauchy mutation (CM). OBL speeds up the method’s convergence, while PC and heavy-tailed CM aid KH in escaping local optima. The results show that among various OKH variants, the KH with OBL, CM and PC operators has the best score.

Li and Wang (2021) used dynamic topology and biogeography-based optimization to solve computational optimization problems based on learning to improve elephant herding optimization (EHO). When compared to other algorithms in the literature, the proposed algorithm proved to be more effective. It also outperformed the competition in the simple traveling salesman dilemma (TSP).
The Cuckoo search (CS) algorithm has been improved in version (Li et al. 2019, 2020, 2021, 2020). The developers in Li et al. (2019) used I-PKL-CS or self-adaptive intelligence learning. By combining the power of human knowledge learning and community knowledge learning with a threshold statistics learning approach, the proposed algorithm achieves a strong balance between discovery and manipulation. I-PKL-CS is a competitive new form of the algorithm, according to the findings.

The authors used the hierarchical step size cuckoo search algorithm and the Q-Learning step size and genetic operator in Li et al. (2020). (DMQL-CS). The DMQL-CS algorithm uses a step size management technique to analyze the individual multi-step evolution effect and learn the individual optimum step size by estimating the Q function value.

Using the opposition-based learning (OBL) strategy and OBSSA, Li et al. Hussien (2021) improved the shortcomings of the Salp Swarm Algorithm (SSA), which may fail to resist local optima and have a sluggish convergence curve. The proposed algorithm was divided into two stages: the first stage used OBL to improve the initialization stage. The second stage used OBL in the population updating phase in each iteration. The findings show that OBSSA can compete with other algorithms in the literature in terms of efficiency.

Using the opposition-based learning (OBL) strategy, dubbed OBSSA, Hussien (2021) improved the shortcomings of the Salp Swarm Algorithm (SSA), which may fail to resist local optima and have a sluggish convergence curve. The proposed algorithm was divided into two stages: the first stage used OBL to improve the initialization stage. The second stage used OBL in the population updating phase in each iteration. The findings show that OBSSA can compete with other algorithms in the literature in terms of efficiency.

### Table 1 Differences between MVO, CSA, GA, PSO, HS and TS

| Properties         | Algorithm | MVO                                                                 | CAS                                                                 | GA          | PSO   | HS     | TS         |
|--------------------|-----------|----------------------------------------------------------------------|----------------------------------------------------------------------|-------------|-------|--------|------------|
| **Parameter**      |           | Mirjalili (2015)                                                     | Yang and Deb (2009)                                                  | Holland (1975) | Kennedy (2010) | Geem et al. (2001) | Glover (1977) |
| **Complexity**     |           | $O(n\log n)$ Li et al. (2016)                                       | $O(nD_{max})$ Salgotra et al. (2018)                                 | $O(m^2)$ Adeec (2000) | $O(nm^2)$ Wang et al. (2007) | $O(HM\times M + HM \times \log (HM\times S))$ Wang et al. (2013) | $O(nm^2)$ Reeves (1993) |
| **Convergence**    |           | Smooth convergence with fast rate Li et al. (2016)                   | Slow convergence rate Shehab et al. (2018)                           | Fast convergence Wright (1991) | Quickly converge Liu et al. (2011) | Suffer from premature convergence Guo et al. (2013) | Rapidly converged Zhang and Sun (2002) |
| **Strength**       |           | Balance between exploration and exploitation Hussien et al. (2020)  | Balance between intensification and diversification Abualigah et al. (2020a) | Deal with the complex fitness landscape Hussien et al. (2020) | Don’t have overlapping and mutation calculation Abualigah et al. (2020b) | Increases the diversity of the new solutions Assiri et al. (2020) | Avoid trapped at local optimum Kulturel-Konak et al. (2003) |
| **Weaknesses**     |           | Relaxed convergence Jain and Saxena (2019)                          | Trapped in a local optimum Shehab et al. (2019)                      | Evaluation is relatively expensive Zingg et al. (2008) | Suffers from partial optimism Bai (2010) | Get stuck on local optima Milad (2013) | Needs huge memory resources Kulturel-Konak et al. (2003) |
evaluation strategy reduced the number of function evaluations while also speeding up convergence. As a result, the TOB-DCS could strike a good balance between discovery and extraction, as shown by the results.

3 Preliminaries

This section presents a brief introduction for the opposition-based Learning (OBL) strategy, disruption operator (DO) and the multi-verse optimizer (MVO) algorithm.

3.1 Opposition-based learning

Opposition-based learning (OBL) mechanism represented is finding the opposite solution for each current solution in the population. After that, it is using their fitness function to select the best solution (ElAziz et al. 2017).

3.1.1 Opposite number

The basics of the OBL method determined by assuming \( x \in [a, b] \) (a and b refer to the upper boundary and the lower boundary of the problem, respectively) and \( a, b \in R \). Thus, the opposite point of \( x \) indicated by \( x_{op} \) can be acquired as Tizhoosh (2005):

\[
x_{op} = a + b - x
\]

(1)

The following subsection shows determining the opposite numbers with a higher dimension.

3.1.2 Opposite vector in d-dimensional

In case, opposite points d-dimensional space need to be computed as \( x_{op} = (x_{op1}, x_{op2}, ..., x_{opd}) \) for the point \( x = (x_1, x_2, ..., x_d) \in R \). The calculation process is shown in following equation (Rahnamayan et al. 2008):

\[
x_{op}^i = a_j + b_j + x^i
\]

(2)

where \( a_j \) and \( b_j \) refer to the lower and the upper boundaries of the element \( x^i \in x \), respectively. Finally, in the optimization method, the current solution \( x \) is chosen if \( f(x) \) is better than \( f(x_{op}) \); otherwise, \( x \) is chosen. So, the population of the solutions is updated based on the best values of \( x \) and \( x_{op} \).

3.2 Disruption operator

The disturbance operator \( (D_{op}) \) was inspired by the astrophysics effect, which it summarizes in “When the total mass (m) of a swarm of gravitationally bound particles is too close to a large object (M), the swarm will be broken apart.” Similarly, when gravitational forces pull together a rigid body, it approaches a much larger entity harwit2006astrophysical.

The full use of the \( D_{op} \) is defined as the following equation (Liu et al. 2014):

\[
D_{op} = \begin{cases} 
Dis_{i,j} \times \Psi \left( \frac{1}{\gamma}, \frac{1}{2} \right) & \text{if } Dis_{i,best} \geq 1 \\
1 + Dis_{i,best} \times \Psi \left( -\frac{10^{-16}}{2}, \frac{10^{-16}}{2} \right) & \text{otherwise}
\end{cases}
\]

(3)

where \( Dis_{i,j} \) indicates the Euclidean distance between two components, \( i'h \) and \( j'h \). The \( \Psi(\ a, \ b \) ) refers to a random number that is uniformly distributed located in \([a, b]\). Consequently, when the two components are too close together, then \( D_{op} < 1 \), so the components will converge toward the origin.

3.3 Multi-verse optimizer (MVO)

MVO is one of the efficient metaheuristic algorithms proposed by Mirjalili et al. (2016) in 2016. MVO is inspired by the big bang theory that led to the birth of a universe. The algorithm and their mathematical models have been established based on three main concepts: white hole, black hole and wormhole. As many proposed metaheuristic algorithms, MVO focused on the exploration process using the white hole and black hole concepts, as well as the exploitation using the wormhole concept. The following steps describe the main operations of the MVO algorithm.

1. Population Initialization

The population is a subset of solutions, in MVO each candidate solution is called a Universe where represent as a vector of actual elements. For instance, \( [x_1^1, x_2^2, ..., x_d^d] \) where \( i \) refers to the number solutions in the population and \( d \) refers to the problem’s dimension (Shehab et al. 2020). The following matrix shows a group of solutions in a certain population \( P \):

\[
P_i = \begin{bmatrix} 
x_1^1 & x_2^1 & ... & x_d^1 \\
x_1^2 & x_2^2 & ... & x_d^2 \\
\vdots & \vdots & \ddots & \vdots \\
x_1^n & x_2^n & ... & x_d^n 
\end{bmatrix}
\]

(4)

where \( n \) refers to the number of solutions in the population \( P_i \).

2. Exploration

As mentioned previously, the white and black hole controls the exploration process in MVO. The mechanism of exchanging between the universes starts by calculating the fitness value which is called inflation rate for each
Opposition-based learning multi-verse... 11673

universe. After that, the white hole (i.e., solution $x_k$) is selected using the fitness proportionate selection scheme. Equation 5 describes the exchange technique between the chosen universe $x_k$ and the black hole (i.e., solution $x_i$).

$$
x_i^j(t + 1) = \begin{cases} 
x_i^j(t) & r1 < N1(U_i) 
ex_j^k(t) & r1 \geq N1(U_i) 
\end{cases} \quad (5)
$$

where $j$ refers to the number of the elements in the universe, $r1$ is a random number locates between [0,1] and $N1(U_i)$ indicates the normalize fitness value of the $i$th universe. Equation 6 presents the normalization method for fitness value.

$$
A_i = \frac{A_{oi}}{\sqrt{\sum^n_{j=1}(A_{oj})^2}} \quad (6)
$$

where $A_i$ and $A_{oi}$ refer to the $i$th element of the vector after standardization and before standardization, respectively. $n$ indicates the number of elements in the universe.

It is worth to mention that the MVO algorithm doesn’t utilize the normalization in the fitness proportionate selection scheme. The normalization is used the fitness value based on the decision in Eq. 5. Consequently, since the value of $r1$ locates between [0,1], the normalization of the fitness value should be located in the same range (i.e., [0,1]).

3. Exploitation

The wormhole in the MVO algorithm is responsible for the exploitation process in the search space. Equation 7 represents the mechanism of the wormhole’s utilization.

$$
x_i^j(t + 1) = \begin{cases} 
x_i^j(t) + TD R (r4(ub_j - lb_j) + lb_j) & r3 < 0.5 
x_i^j(t) + TD R (r4(ub_j - lb_j) + lb_j) & r3 \geq 0.5 
x_i^j(t) & r2 < WEP \end{cases} \quad (7)
$$

where $X_j$ indicates the $j$th solution of the best universe; $lb_j$ and $ub_j$ refer to the lower bound and upper bound of the $j$th element, respectively. $r2$, $r3$ and $r4$ are random numbers that fall in the ranges of [0, 1]. The transmission processes from discovery to exploitation are regulated by the traveling distance rate (TDR) and the wormhole presence likelihood (WEP). The WEP and TDR change their values over the direction of the iterations, as seen in Eqs. 8 and 9.

$$
WEP = min + l \times \left( \frac{\text{max} - \text{min}}{L} \right) \quad (8)
$$

$$
TDR = 1 - \left( \frac{l}{L} \right)^p \quad (9)
$$

where $l$ refers to the current iteration, $L$ refers to the maximum number of iterations, $min$ is the minimum possible values of WEP where the common value is 0 and max is the maximum possible values of WEP where the common value is 2. $p$ indicates the exploitation intensifier (Fig. 1).

It can be noticed that in Eq. 8, the WEP’s value increases in each iteration. Thus, that enhanced the exploitation process, while in Eq. 9, the value of TDR reducing which affects the local search process in each iteration. The pseudo-code of the MVO algorithm is shown in Algorithm 1.

**Algorithm 1 Pseudo-code of MVO algorithm**

Input: Population size and number of iterations ($L$).

Output: The best universe and its inflation rate.

Define: $SU$ = Sorted universes, $NI$ = Normalized inflation rate, Black hole_index = $i$, $r_1$, $r_2$, $r_3$, $r_4$ = Rand([0,1])

Initialize all random universes $x_i$ ($i = 1, 2, \ldots, n$), WEP, TDR, and best universe.

while (end condition is not met) do
  Calculate the fitness of universes.
  for (each Universej) do
    Update WEP and TDR
    for (each Objectj) do
      if $r_1 < NI(U_i)$ then
        White hole_index = RouletteWheelSelection($-NI$)
        $U$(Black hole_index, $j$) = $SU$(White hole_index, $j$)
      end
      if $r_2 < WEP$ then
        if $r_3 < 0.5$ then
          $U(i, j) = \text{Best universe}(j) + \text{TDR} \times ((ab(j) - lb(j)) \times r_4 + lb(j))$
        else
          $U(i, j) = \text{Best universe}(j) - \text{TDR} \times ((ab(j) - lb(j)) \times r_4 + lb(j))$
        end
      end
  end
Return: The best universe
4 The proposed method

This section presents two new methods for enhancing basic MVO.

4.1 Opposition-based learning and multi-verse optimizer

This section presents the first improvement which combining the basic MVO with the OBL to enhance the ability of MVO in the exploration search and achieve the optimal value, rapidly. The proposed version is called OMVO. The following points show the two main stages of the OMVO:

1. Initial stage:
The OMVO starts by initializing the population using the MVO strategy (i.e., \([x_1^i, x_2^i, \ldots, x_d^i]\)). The OBL is utilized to find the opposite solution for each actual element (solution) in the vector. After that, it calculates the fitness function each solution \([x_1^i, x_2^i, \ldots, x_d^i]\) and their \([\bar{x}_1^i, \bar{x}_2^i, \ldots, \bar{x}_d^i]\). Then, it determines the best N solutions from the union of the two populations (i.e., the actual and the opposite solutions).

2. Updating stage:
This stage presents the updating and evaluating the solutions based on Eq. 6. Then, it determines the best solution and its fitness value. In contrast, the OBL mechanism keeps a group of the updated solutions from the MVO and calculating their opposite fitness values. After that, the best solution will be compared with the OBL solutions and the OMVO will choose the best to update the population in the next iteration. These processes will repeat until achieving the maximum number of iterations (i.e., stop criterion).

4.2 Improving OMVO using disruption operator

This section shows enhancing both the diversity and the convergence rate of the OMVO using the \(D_{op}\) which is called OMVOD. The proposed algorithms are applied to improve its exploration and exploitation capabilities (Shehab et al. 2019). Thus, the disruption operator in OMVOD is depended on the distances between the elements at the given time in the search space. Therefore, the \(D_{op}\) will be applied if the ratio of distances between the element \(i\) and the nearest element, and the ratio of distances between the element \(i\) and the start element is lower than the threshold \(C\), see the following equation.

\[
\frac{R_{i,nbd}}{R_{i,best}} < C \tag{10}
\]

\(R_{i,nbd}\) refers to the Euclidean distance between \(i^{th}\) element and it’s nearest element, \(R_{i,best}\) is refers to the Euclidean distance between \(i^{th}\) element and star element, \(C\) is defined as the following equation.

\[
C = \theta \left(1 - \frac{GenIndex}{MaxGen} \right) \tag{11}
\]

where the \(\theta\) is a parameter that is either a constant or a function of generations; the \(GenIndex\) and \(MaxGen\) indicate the generation index and the maximum number of generations, respectively. The position of every element that satisfies Eq.10 will update according to the following equation:

\[
x_i(new) = x_i(old) \cdot D, \quad D = \begin{cases} R_{i,nbd} \cdot U(-0.5, 0.5) & \text{if } R_{i,best} \geq 1 \\ 1 + \rho \cdot U(-0.5, 0.5) & \text{otherwise} \end{cases} \tag{12}
\]
Table 2  Review of CEC 2015 benchmark function problems

| No. | Type                        | Description                                         | $F^*$ |
|-----|-----------------------------|------------------------------------------------------|-------|
| 1   | Unimodal functions          | Rotated Bent Cigar Function                          | 100   |
| 2   |                            | Rotated Discus Function                              | 200   |
| 3   | Simple multimodal functions | Shifted and Rotated Weierstrass Function             | 300   |
| 4   |                            | Shifted and Rotated Schwefel’s Function              | 400   |
| 5   |                            | Shifted and Rotated Katsuura Function                | 500   |
| 6   |                            | Shifted and Rotated HappyCat Function                | 600   |
| 7   |                            | Shifted and Rotated HGBat Function                   | 700   |
| 8   |                            | Shifted and Rotated Expanded Griewank’s plus Rosenbrock’s Function | 800   |
| 9   |                            | Shifted and Rotated Expanded Scaffer’s F6 Function   | 900   |
| 10  | Hybrid functions            | Hybrid Function 1 (N=3)                              | 1000  |
| 11  |                            | Hybrid Function 2 (N=4)                              | 1100  |
| 12  |                            | Hybrid Function 3 (N=5)                              | 1200  |
| 13  | Composition functions       | Composition Function 1 (N=5)                         | 1300  |
| 14  |                            | Composition Function 2 (N=3)                         | 1400  |
| 15  |                            | Composition Function 3 (N=5)                         | 1500  |

where $U(-0.5, 0.5)$ returns a uniformly distributed pseudo-random number in the interval $[-0.5, 0.5]$. \(X_i = (x^1_i, x^2_i, \ldots, x^d_i)\) is the position of element \(i\) that should be disrupted, and \(\rho\) refers to a small number.

5 Experiments and results

In this section, comprehensive experiments are conducted using three sets of optimization problems to test the performance of the proposed algorithms (Abualigah et al. 2021a, b). These problems are divided into three main parts; Part 1: Experiments on CEC 2015, which contains fifteen benchmark functions (Shehab et al. 2017b). Part 2: Experiments on CEC 2017, which contains thirteen benchmark functions (Shehab et al. 2018). And Part 3: Experiments on CEC 2011 real-world problem, which includes seven real-world optimization problems (Shehab et al. 2020).

The values of MVO’s parameters are the same in Mirjalili et al. (2016). The maximum number of iterations is 1000, and the number of solutions $N$ is set to thirteen with a dimension space value equal to the dimension of the given problem. Thus, all of the proposed algorithms were run thirteen times over each optimization problem for statistical analysis purposes. Furthermore, both experimental groups run on MATLAB 2016 running on Windows 7 on a PC with a Core 2 Duo processor and 16 GB of RAM. The below are the organizations and analyses of proposed approaches for these research problems:

5.1 Part 1: Experiments on CEC 2015

In this section, the proposed algorithms are tested. The findings of other published approaches in the literature are compared using CEC 2015 benchmark function problems. Table 2 contains the specifics of the CEC 2015 benchmark feature issues.

These benchmark functions are divided into four categories. The first is the Unimodal functions (from F1 to F2 in Table 2) used to evaluate the proposed algorithms’ optimization accuracy and convergence speed (Shehab et al. 2019). In the given search field, these functions have a single final solution. Meanwhile, the Simple Multimodal Functions (from F3 to F9 in Table 2) test the efficiency of the proposed algorithms in evading the exploitation optimum and providing the global solution (Mohammad Abualigah et al. 2020). The third form is Hybrid Functions (from F10 to F12 in Table 2), which evaluates the proposed algorithms’ optimization efficiency and convergence acceleration. Finally, the Composition Functions (from F13 to F15 in Table 2) are the fourth kind. They are used to evaluate the optimization efficiency and convergence acceleration of the proposed algorithms.

Four versions of the proposed MVO are investigated using CEC 2015 benchmark function problems as shown in Tables 3-6 and Figure 2 (from A to O). These experiments are conducted to find the best vision from the given algorithms, which are Basic Multi-Verse Optimizer called BMVO, multi-verse optimizer with disruption operator called MVOD, Opposition based Multi-Verse Optimizer called OMVO and Opposition-based learning Multi-Verse Optimizer with disruption operator called OMVOD.
Fig. 2 Convergence behavior of the proposed algorithms on CEC 2015 test functions
Fig. 2 continued
When the dimension space is set to 10, the efficiency of the proposed algorithms (BMVO, MVOD, OMVO and OMVOD) is shown in Table 3. The findings in Table 3 show that the proposed algorithm (OMVOD) outperformed other proposed methods (MVOD, OMVO) and the simple MVO in the vast majority of instances (BMVO). The suggested algorithm (OMVOD) generated the best results (F1 and F2). For four of the seven functions, the proposed algorithm (OMVOD) generated the best results in the second group (F4, F5, F8 and F9). In the third group, the proposed algorithm (OMVOD) outperformed the competition in three scenarios (all given functions). Finally, OMVOD achieved the best performance in the fourth group (F13 to F15). These research problems were used to assess discovery and extraction potential. We deduced from their findings that MVO’s modification accomplished its primary goal of improving its efficiency.

The efficiency of the proposed algorithms (BMVO, MVOD, OMVO, and OMVOD) is shown in Table 4 when the dimension space is expanded to 30. The findings in Table 4 show that the proposed algorithm (OMVOD) outperformed other proposed methods (MVOD, OMVO) and the simple MVO in the vast majority of instances (BMVO). The suggested algorithm (OMVOD) generated the best results (F1 and F2). For seven out of seven functions, the proposed algorithm (OMVOD) generated the best results in the second group (all functions). In the third group, the proposed algorithm (OMVOD) outperformed the competition in two of three scenarios (F11 and F12).

OMVOD achieved the best performance in the fourth group (F13 to F15). These optimization problems were used to assess discovery and exploitation using an expanded dimension space (30). We deduced from their findings that the MVO modification, especially the proposed algorithm (OMVOD), achieved its primary goal of improving efficiency. The convergence behavior of the proposed algorithms on CEC 2015 test functions is seen in Figure 2(A-O). Furthermore, the findings seen in the sub-figures demonstrated
Table 3  Performance of the proposed algorithms on CEC 2015 benchmark function problems, Dim=10

| Function No. | BMVO | MVOD | OMVO | OMVOD |
|--------------|------|------|------|-------|
| F1           | 3.4522E+07 | 2.8945E+08 | 2.9862E+05 | 4.9786E+09 |
| F2           | 3.9453E+03 | 1.2487E+00 | 3.2561E+03 | 6.124E+00 |
| F3           | 3.2253E+02 | 1.2487E+00 | 3.3614E+02 | 6.2603E+02 |
| F4           | 3.5453E+03 | 3.2112E+02 | 2.9856E+03 | 2.6203E+02 |
| F5           | 5.2513E+03 | 5.9796E-01 | 5.6480E+02 | 9.3430E+01 |
| F6           | 6.3352E+03 | 5.8077E+00 | 4.0541E+02 | 3.5349E+01 |
| F7           | 7.4744E+03 | 5.8077E+00 | 2.3422E+03 | 2.6453E+03 |
| F8           | 7.6545E+02 | 2.3842E-01 | 5.5482E+02 | 4.7453E-01 |
| F9           | 5.3672E+06 | 6.5045E+05 | 1.1345E+03 | 5.3696E+02 |
| F10          | 1.9960E+03 | 4.7569E+01 | 6.2587E+02 | 5.4141E+02 |
| F11          | 3.5984E+03 | 2.9519E+01 | 2.2345E+03 | 4.6565E+02 |
| F12          | 2.6687E+04 | 9.1615E+01 | 1.3689E+02 | 4.3856E+02 |
| F13          | 2.7981E+04 | 7.4514E+00 | 1.4324E+02 | 2.2525E+02 |

Table 4  Performance of the proposed algorithms on CEC 2015 benchmark function problems, Dim=30

| Function No. | BMVO | MVOD | OMVO | OMVOD |
|--------------|------|------|------|-------|
| F1           | 2.4586E+06 | 5.8475E+02 | 9.2586E+04 | 1.6482E+05 |
| F2           | 5.3468E+05 | 2.6484E+04 | 3.9830E+05 | 1.4687E+05 |
| F3           | 3.5560E+02 | 2.9654E+00 | 3.2781E+02 | 2.9972E+00 |
| F4           | 8.4856E+03 | 4.4676E+02 | 6.5743E+03 | 4.4676E+02 |
| F5           | 5.7467E+02 | 5.8678E+00 | 5.5778E+02 | 1.8747E+00 |
| F6           | 6.1345E+02 | 3.5678E+01 | 3.3795E+02 | 5.3696E+01 |
| F7           | 7.1586E+02 | 1.5464E+01 | 6.6526E+02 | 2.6543E+01 |
| F8           | 2.3578E+04 | 1.7532E+06 | 1.3689E+02 | 4.3856E+05 |
| F9           | 3.4574E+02 | 2.4924E+01 | 9.3971E+01 | 1.1254E+01 |
| F10          | 2.4568E+05 | 1.8076E+05 | 3.1458E+04 | 1.4582E+05 |
| F11          | 1.5485E+05 | 2.4856E+01 | 2.5969E+04 | 3.8589E+02 |
| F12          | 2.4585E+06 | 3.3621E+02 | 4.9514E+04 | 6.4328E+04 |
| F13          | 1.8641E+05 | 2.1153E+02 | 2.1236E+03 | 4.7545E+02 |
| F14          | 2.8504E+03 | 2.2665E+01 | 1.3854E+03 | 1.4580E+02 |
| F15          | 3.5640E+03 | 4.4751E+01 | 2.8547E+03 | 4.6582E+02 |

that the planned OMVOD conference to the best solution prevents premature convergence.

The statistic rank test, Friedman rank test, for the proposed algorithms on CEC 2015 test functions with 10 and 30 dimension space (Dim) is shown in Table 5. The findings indicate that, in general, the suggested algorithms generated comparable results. Furthermore, when the dimension is set to 10, the proposed OMVOD received the highest mean rank value (17), followed by OMVO (35), MVOD (41) and BMVO (51). In addition, when the dimension is set to 30, the proposed OMVOD has the best mean rank value (16), followed by OMVO (35), MVOD (41) and BMVO (47). Furthermore, when the dimension value is set at 10, the proposed OMVOD has the highest final ranking value (ranked first), followed by OMVO (ranked second), MVOD (ranked third) and BMVO (ranked fourth) (ranked as the fourth).
Table 5  Friedman rank test for the proposed algorithms on CEC 2015 test functions using 10 and 30 Dim

| Function No. | Dim | Proposed Algorithms |
|--------------|-----|----------------------|
|              |     | BMVO | MVOD | OMVO | OMVOD |
| F1           | 10  | 4    | 3    | 2    | 1     |
| F1           | 30  | 4    | 3    | 2    | 1     |
| F2           | 10  | 4    | 3    | 2    | 1     |
| F2           | 30  | 4    | 3    | 2    | 1     |
| F3           | 10  | 3    | 4    | 2    | 1     |
| F3           | 30  | 4    | 3    | 2    | 1     |
| F4           | 10  | 4    | 3    | 2    | 1     |
| F4           | 30  | 4    | 3    | 2    | 1     |
| F5           | 10  | 2    | 4    | 3    | 1     |
| F5           | 30  | 4    | 3    | 2    | 1     |
| F6           | 10  | 4    | 3    | 1    | 2     |
| F6           | 30  | 4    | 2    | 3    | 1     |
| F7           | 10  | 4    | 1    | 3    | 2     |
| F7           | 30  | 4    | 2    | 3    | 1     |
| F8           | 10  | 4    | 2    | 3    | 1     |
| F8           | 30  | 4    | 2    | 3    | 1     |
| F9           | 10  | 4    | 2    | 3    | 1     |
| F9           | 30  | 4    | 3    | 2    | 1     |
| F10          | 10  | 4    | 3    | 2    | 1     |
| F10          | 30  | 3    | 4    | 1    | 2     |
| F11          | 10  | 4    | 3    | 2    | 1     |
| F11          | 30  | 4    | 3    | 2    | 1     |
| F12          | 10  | 4    | 2    | 3    | 1     |
| F12          | 30  | 4    | 3    | 2    | 1     |
| F13          | 10  | 4    | 3    | 2    | 1     |
| F13          | 30  | 3    | 4    | 2    | 1     |
| F14          | 10  | 4    | 3    | 2    | 1     |
| F14          | 30  | 4    | 2    | 3    | 1     |
| F15          | 10  | 4    | 2    | 3    | 1     |
| F15          | 30  | 4    | 2    | 3    | 1     |
| Mean rank    | 10  | 57   | 41   | 35   | 17    |
| Mean rank    | 30  | 58   | 41   | 35   | 16    |
| Final ranking| 10  | 4    | 3    | 2    | 1     |
| Final ranking| 30  | 4    | 3    | 2    | 1     |

Furthermore, when the dimension value is set to 30, the proposed OMVOD has the highest final ranking value (ranked first), followed by OMVO (ranked second), MVOD (ranked third) and BMVO (ranked fourth) (ranked as the fourth), as shown in Table 5. When we merged the opposition-based learning approach into the MVO with a disruption operator, we found that the best proposed variant is opposition-based learning multi-verse optimizer with a disruption operator called OMVOD. The second-best suggested approach, opposition-based learning with MVO (OMVO), showed that the opposition-based learning technique was better than the disruption operator. It gave the MVO more flexibility while maintaining its diversity of solutions.

On CEC 2015 test functions using 30 Dim, Table 6 displays efficiency comparisons of the best proposed algorithm (OMVOD) with other related approaches. The results of the best proposed algorithm are validated and compared using eight well-known published methods (OMVOD). These methods are DE (Ngo et al. 2016), iSRPSO (Tanweer et al. 2015), $\mu + \lambda$-ES (Aydilek 2018), EPSO (Ngo et al. 2016), CMAES-S (Andersson et al. 2015), CMAES-G (Andersson et al. 2015), ISRPSO (Aydilek 2018) and HFPSSO (Aydilek 2018). The proposed OMVOD obtained better results in comparison with other similar methods in almost all cases (it got eleven best cases out of fifteen). Generally, the results show the superiority of the proposed algorithm by getting the best global optimum solution.

The statistic rank test, Friedman rank test, for the proposed algorithms on CEC 2015 test functions using 30 dimension space (Dim) is also shown in Table 6. The findings show that the suggested algorithms outperformed the competition. Furthermore, the proposed OMVOD received the highest mean rank rating (1.60), followed by HFPSSO (3.26), ISRPSO (3.46), iSRPSO (3.73), EPSO (5.20), DE (6.13), $\mu + \lambda$-ES (6.80), CMAES-G (6.80) and CMAES-G (7.86). The proposed OMVOD was ranked first, followed by HFPSSO (ranked second), ISRPSO (ranked third), iSRPSO (ranked fourth), EPSO (ranked fifth), DE (ranked sixth), $\mu + \lambda$-ES (ranked seventh), CMAES-G (ranked eighth), CMAES-G (ranked ninth) and CMAES-G (ranked the ninth).

When we merged the opposition-based learning approach into the MVO with a disruption operator, we found that the best proposed variant is opposition-based learning multi-verse optimizer with a disruption operator called OMVOD. It outperformed other published approaches that used the same benchmark functions in the literature.

5.2 Part 2: Experiments on CEC 2017

In this section, the proposed algorithms are tested, and the findings of other published approaches in the literature are compared using CEC 2017 benchmark function problems. Table 7 shows the specifics of the CEC 2017 benchmark feature issues.

These benchmark functions are divided into four categories. The first is the Unimodal functions (from F1 to F3 in Table 7), which are used to evaluate the proposed algorithms’ optimization accuracy and convergence speed. Furthermore, in the given search field, these functions provide a single final solution. Meanwhile, the Simple Multimodal Functions (from F4 to F10 in Table 7) are used to test the efficiency of the proposed algorithms in evading the exploitation optimum and
Table 6  Performance comparisons of the best proposed algorithm (OMVOD) with other similar methods on CEC 2015 test functions using 30 Dim

| Function No. | Comparative Algorithms | (μ+λ)-ES | EPSO | CMAES-S | CMAES-G | ISRPSO | HFPSO | OMVOD |
|--------------|------------------------|----------|------|---------|---------|--------|-------|-------|
| Rank         |                        |          |      |         |         |        |       |       |
| F1           | 2.3911E+10             | 6.0180E+08 | 3.5775E+10 | 8.4866E+09 | 6.8700E+07 | 1.1080E+08 | 7.1910E+08 | 1.1795E+09 | 2.5515E+03 |
| Rank         | 8                      | 4        | 9    | 7      | 2       | 3      | 5     | 6     | 1      |
| F2           | 1.8254E+05             | 7.7320E+04 | 1.6179E+05 | 6.3748E+04 | 2.3630E+05 | 2.9530E+05 | 7.6860E+04 | 8.5653E+04 | 1.3363E+05 |
| Rank         | 6                      | 1        | 7    | 2      | 8       | 9      | 3     | 4     | 5      |
| F3           | 3.4190E+02             | 2.6030E+02 | 3.4353E+02 | 3.3800E+02 | 6.3390E+02 | 6.5270E+02 | 3.2569E+02 | 3.2638E+02 | 2.2687E+02 |
| Rank         | 7                      | 7        | 6    | 5      | 8       | 9      | 3     | 4     | 1      |
| F4           | 7.9627E+03             | 5.4450E+03 | 7.0557E+03 | 6.6946E+03 | 8.6730E+03 | 1.2040E+04 | 5.8090E+03 | 5.1202E+03 | 5.7343E+03 |
| Rank         | 7                      | 4        | 6    | 5      | 8       | 9      | 3     | 2     | 1      |
| F5           | 5.0431E+02             | 4.4100E+02 | 5.0499E+02 | 5.0430E+02 | 1.0010E+03 | 1.0080E+03 | 5.0424E+02 | 5.0410E+02 | 4.3456E+02 |
| Rank         | 4                      | 7        | 7    | 6      | 8       | 9      | 3     | 2     | 1      |
| F6           | 6.0365E+02             | 6.3980E-01 | 6.0433E+02 | 6.0276E+02 | 1.2010E+03 | 1.2010E+03 | 6.0064E+02 | 6.0076E+02 | 1.0025E+02 |
| Rank         | 6                      | 1        | 7    | 5      | 8       | 8      | 3     | 4     | 2      |
| F7           | 7.5438E+02             | 4.8800E-01 | 7.8216E+02 | 7.2189E+02 | 1.4010E+03 | 1.4010E+03 | 7.0057E+02 | 7.0074E+02 | 5.0052E+02 |
| Rank         | 6                      | 1        | 7    | 5      | 8       | 8      | 3     | 4     | 2      |
| F8           | 7.9963E+05             | 5.1450E+02 | 7.3789E+06 | 1.2746E+05 | 1.7670E+03 | 2.3210E+03 | 1.4262E+03 | 2.6354E+03 | 1.2743E+02 |
| Rank         | 8                      | 2        | 9    | 7      | 4       | 5      | 3     | 6     | 1      |
| F9           | 9.1349E+02             | 1.3540E+01 | 9.1408E+02 | 9.1372E+02 | 1.8270E+03 | 1.8280E+03 | 9.1357E+02 | 9.1337E+02 | 1.0330E+01 |
| Rank         | 9                      | 4        | 2    | 7      | 6       | 8      | 9     | 5     | 3      |
| F10          | 3.8759E+07             | 6.4750E+06 | 9.5323E+07 | 2.6363E+07 | 3.6310E+06 | 1.4730E+07 | 6.8320E+06 | 5.4690E+06 | 2.1114E+04 |
| Rank         | 7                      | 4        | 9    | 8      | 2       | 6      | 5     | 3     | 1      |
| F11          | 1.2870E+03             | 3.1530E+01 | 1.4378E+03 | 1.2288E+03 | 2.2460E+03 | 2.2580E+03 | 1.1509E+03 | 1.1336E+03 | 1.1887E+03 |
| Rank         | 6                      | 1        | 7    | 5      | 8       | 9      | 3     | 2     | 4      |
| F12          | 3.0110E+03             | 7.0880E+02 | 3.8087E+03 | 2.4432E+03 | 3.4540E+03 | 4.0940E+03 | 1.9357E+03 | 1.7752E+03 | 3.4852E+02 |
| Rank         | 6                      | 2        | 7    | 5      | 8       | 9      | 4     | 3     | 1      |
| F13          | 1.9613E+03             | 3.9950E+03 | 2.2208E+03 | 1.8839E+03 | 3.8480E+03 | 3.4260E+03 | 1.6996E+03 | 1.6866E+03 | 1.4568E+03 |
| Rank         | 5                      | 9        | 6    | 4      | 7       | 8      | 3     | 2     | 1      |
| F14          | 1.7479E+03             | 2.6560E+03 | 1.8406E+03 | 1.7016E+03 | 3.2660E+03 | 3.3000E+03 | 1.6655E+03 | 1.6469E+03 | 1.3446E+03 |
| Rank         | 6                      | 7        | 5    | 4      | 8       | 9      | 3     | 2     | 1      |
| F15          | 2.9304E+03             | 9.7610E+03 | 2.9154E+03 | 2.7488E+03 | 4.4270E+03 | 4.8360E+03 | 2.4510E+03 | 2.4467E+03 | 2.2565E+03 |
| Rank         | 6                      | 9        | 5    | 4      | 7       | 8      | 3     | 2     | 1      |
| Mean rank    | 6.13                   | 3.73     | 6.80  | 5.20    | 6.80    | 7.86   | 3.46   | 3.26   | 1.60    |
| Final ranking| 6                     | 4        | 7    | 5      | 7       | 8      | 3     | 2     | 1      |
Table 7: Review of CEC 2017 benchmark function problems

| No. | Type                                | Description                              | F\(i^*\) |
|-----|-------------------------------------|------------------------------------------|-----------|
| 1   | Unimodal functions                  | Shifted and Rotated Bent Cigar Function  | 100       |
| 2   | Shifted and Rotated Sum of Different Power Function | 200       |
| 3   | Shifted and Rotated Zakharov Function | 300       |
| 4   | Simple Multimodal Functions         | Shifted and Rotated Rosenbrock’s Function | 400       |
| 5   | Shifted and Rotated Rastrigin’s Function | 500       |
| 6   | Shifted and Rotated Expanded Scaffer’s F6 Function | 600       |
| 7   | Shifted and Rotated Lunacek Bi-Rastrigin Function | 700       |
| 8   | Shifted and Rotated Non-Continuous Rastrigin’s Function | 800       |
| 9   | Shifted and Rotated Levy Function    |                                           | 900       |
| 10  | Shifted and Rotated Schwefel’s Function |                                           | 1000      |
| 11  | Hybrid functions                    | Hybrid Function 1 (N=3)                  | 1100      |
| 12  | Hybrid Function 2 (N=3)             |                                           | 1200      |
| 13  | Hybrid Function 3 (N=3)             |                                           | 1300      |
| 14  | Hybrid Function 4 (N=4)             |                                           | 1400      |
| 15  | Hybrid Function 5 (N=4)             |                                           | 1500      |
| 16  | Hybrid Function 6 (N=4)             |                                           | 1600      |
| 17  | Hybrid Function 6 (N=5)             |                                           | 1700      |
| 18  | Hybrid Function 6 (N=5)             |                                           | 1800      |
| 19  | Hybrid Function 6 (N=5)             |                                           | 1900      |
| 20  | Hybrid Function 6 (N=6)             |                                           | 2000      |
| 21  | Composition Functions               | Composition Function 1 (N=3)             | 2100      |
| 22  | Composition Function 2 (N=3)        |                                           | 2200      |
| 23  | Composition Function 3 (N=4)        |                                           | 2300      |
| 24  | Composition Function 4 (N=4)        |                                           | 2400      |
| 25  | Composition Function 5 (N=5)        |                                           | 2500      |
| 26  | Composition Function 6 (N=5)        |                                           | 2600      |
| 27  | Composition Function 7 (N=6)        |                                           | 2700      |
| 28  | Composition Function 8 (N=6)        |                                           | 2800      |
| 29  | Composition Function 9 (N=3)        |                                           | 2900      |
| 30  | Composition Function 10 (N=3)       |                                           | 3000      |

providing the global solution. The third form is Hybrid Functions (from F11 to F20 in Table 7), which are used to evaluate the proposed algorithms’ optimization efficiency and convergence acceleration. Finally, the Composition Functions (from F13 to F15 in Table 7) are the fourth kind, and they are used to evaluate the optimization efficiency and convergence acceleration of the proposed algorithms.

When the dimension space is set to 10, Table 8 displays the efficiency of the proposed algorithms (i.e., BMVO, MVOD, OMVO, and OMVOD) on the CEC 2017 benchmark function problems. The findings in Table 8 confirmed that the proposed algorithm (OMVOD) outperformed other proposed methods (MVOD, OMVO) and the simple MVO in the majority of instances (BMVO). The suggested algorithm (OMVOD) generated the best results (F1 and F3).

In all cases (seven out of seven functions), the suggested algorithm (OMVOD) generated the best results in the second group (F4 to F10). The suggested algorithm (OMVOD) outperformed the competition in three out of three cases (all functions from F11 to F20). Finally, overall situations, OMVOD received the best performance in the fourth group (F21 to F30). These research problems were used to assess discovery and extraction potential. The performance of the proposed algorithms (i.e., BMVO, MVOD, OMVO and OMVOD) on CEC 2017 benchmark function problems is shown in Table 9 when the dimension space is set to 30.

The results shown in Table 9 are a continuation of the earlier results. The findings showed that the proposed algorithm (OMVOD) outperformed other proposed methods (MVOD, OMVO) and the simple MVO in every case (BMVO). We inferred from the results that adding the opposite dependent
learning method to the MVO with the disruption operator achieved its key goals to improve its efficiency.

On CEC 2017 benchmark function problems, Table 9 shows the efficiency of the proposed algorithms (i.e., BMVO, MVOD, OMVO and OMVOD) when the dimension space is set to 30. The results shown in Table 9 suggested that the proposed algorithm (OMVOD) outperformed other proposed methods (MVOD, OMVO) and the simple MVO (BMVO) in all situations (BMVO). In any case, in the first range, the suggested algorithm (OMVOD) produced the best results (F1-F3). In all cases (seven out of seven functions), the suggested algorithm (OMVOD) generated the best results in the second group (F4-F10). The suggested algorithm (OMVOD) outperformed the competition in three out of three cases (all functions from F11 to F20). Finally, overall situations, OMVOD received the best performance in the fourth group (F21 to F30). These research problems were used to assess discovery and extraction potential. The performance of the proposed algorithms (i.e., BMVO, MVOD, OMVO and OMVOD) on CEC 2017 benchmark function problems is seen in Table 9 when the dimension space is set to 30. According to the Friedman ranking test, the findings in Table 10 are devoted to the results seen earlier. Where the dimensions are equal to 10 and 30, the results showed that the proposed algorithm (OMVOD) outperformed other proposed methods (MVOD, OMVO) and the simple MVO (BMVO) in both situations. Therefore, we inferred from the results that adding the opposite dependent learning method to the MVO
with the disruption operator achieved its key goals to improve its efficiency.

Figure 3 depicts the distribution effects of the suggested processes. The distribution results for multiple functions from all groups are presented. The distributions of the obtained results by the proposed approach (OMVOD) are much better than the other comparative approaches, as seen in all sub-figures. Furthermore, all OMVOD’s results have a high density near the optimum solutions, demonstrating that OMVOD’s efficiency with these modifications is superior.

On CEC 2017 test functions using 30 Dim, Table 11 displays efficiency comparisons of the best proposed algorithm (OMVOD) with other related approaches. The results of the best proposed algorithm are validated and compared using eight well-known published methods (OMVOD). These methods are Firefly Algorithm (FA) Aydilek (2018), Particle Swarm Optimization (PSO) Ngo et al. (2016), Gravitational Search Algorithm (GSA) Yazdani et al. (2014), Glowworm Swarm Optimization (GSO) Abedinpourshotorban et al. (2016), Hybrid Firefly and Particle Swarm Optimization (FFPSO) Aydilek (2018), Dynamic Virtual Bats Algorithm (DVBA) Topal and Altun (2016), Hybrid Particle Swarm Optimization Firefly algorithm (HPSOFF) Aydilek (2018), MVO Mirjalili et al. (2016) and OMVOD.

In every scenario, the proposed OMVOD produced better results than other related approaches (it got fourteen best cases out of thirteen). In general, the results suggest that the proposed algorithm is better for obtaining the best global
Table 10 Friedman rank test for the proposed algorithms on CEC 2017 test functions using 10 and 30 Dim

| Function No. | Dim | Proposed Algorithms | BMVO | MVOD | OMVO | OMVOD |
|--------------|-----|---------------------|------|------|------|-------|
| F1           | 10  | 3                   | 2    | 1    |
| F1           | 30  | 4                   | 3    | 2    |
| F2           | 10  | 4                   | 3    | 2    |
| F2           | 30  | 4                   | 3    | 2    |
| F3           | 10  | 3                   | 4    | 2    |
| F3           | 30  | 4                   | 3    | 2    |
| F4           | 10  | 3                   | 4    | 2    |
| F4           | 30  | 4                   | 3    | 2    |
| F5           | 10  | 4                   | 3    | 2    |
| F5           | 30  | 4                   | 3    | 2    |
| F6           | 10  | 4                   | 3    | 2    |
| F6           | 30  | 4                   | 2    | 3    |
| F7           | 10  | 4                   | 3    | 2    |
| F7           | 30  | 4                   | 3    | 2    |
| F8           | 10  | 4                   | 2    | 3    |
| F8           | 30  | 4                   | 3    | 2    |
| F9           | 10  | 4                   | 3    | 2    |
| F9           | 30  | 4                   | 3    | 2    |
| F10          | 10  | 4                   | 3    | 2    |
| F10          | 30  | 4                   | 3    | 2    |
| F11          | 10  | 4                   | 3    | 2    |
| F11          | 30  | 4                   | 3    | 2    |
| F12          | 10  | 4                   | 3    | 2    |
| F12          | 30  | 4                   | 3    | 2    |
| F13          | 10  | 4                   | 3    | 2    |
| F13          | 30  | 4                   | 3    | 2    |
| F14          | 10  | 4                   | 3    | 2    |
| F14          | 30  | 4                   | 3    | 2    |
| F15          | 10  | 4                   | 3    | 2    |
| F15          | 30  | 4                   | 3    | 2    |
| F16          | 10  | 4                   | 3    | 2    |
| F16          | 30  | 4                   | 3    | 2    |
| F17          | 10  | 4                   | 3    | 2    |
| F17          | 30  | 4                   | 3    | 2    |
| F18          | 10  | 4                   | 3    | 2    |
| F18          | 30  | 4                   | 3    | 2    |
| F19          | 10  | 4                   | 3    | 2    |
| F19          | 30  | 4                   | 3    | 2    |
| F20          | 10  | 4                   | 3    | 2    |
| F20          | 30  | 4                   | 3    | 2    |
| F21          | 10  | 4                   | 3    | 2    |
| F21          | 30  | 4                   | 3    | 2    |
| F22          | 10  | 4                   | 3    | 2    |
| F22          | 30  | 4                   | 3    | 2    |

Table 10 continued

| Function No. | Dim | Proposed Algorithms | BMVO | MVOD | OMVO | OMVOD |
|--------------|-----|---------------------|------|------|------|-------|
| F23          | 10  | 4                   | 3    | 2    |
| F23          | 30  | 4                   | 2    | 2    |
| F24          | 10  | 4                   | 3    | 2    |
| F24          | 30  | 4                   | 3    | 2    |
| F25          | 10  | 4                   | 3    | 2    |
| F25          | 30  | 4                   | 3    | 2    |
| F26          | 10  | 4                   | 3    | 2    |
| F26          | 30  | 4                   | 3    | 2    |
| F27          | 10  | 4                   | 3    | 2    |
| F27          | 30  | 4                   | 3    | 2    |
| F28          | 10  | 4                   | 3    | 2    |
| F28          | 30  | 4                   | 3    | 2    |
| F29          | 10  | 4                   | 3    | 2    |
| F29          | 30  | 4                   | 3    | 2    |
| F30          | 10  | 4                   | 3    | 2    |
| F30          | 30  | 4                   | 3    | 2    |

Mean rank

| Dim | Proposed Algorithms | BMVO | MVOD | OMVO | OMVOD |
|-----|---------------------|------|------|------|-------|
| 10  | 3.93                | 3.00 | 2.06 | 1.00 |
| 30  | 4.00                | 2.83 | 2.16 | 1.00 |

Final ranking

| Dim | Proposed Algorithms | BMVO | MVOD | OMVO | OMVOD |
|-----|---------------------|------|------|------|-------|
| 10  | 4                   | 3    | 2    |
| 30  | 4                   | 3    | 2    |

optimum solution. The statistic rank test, Friedman rank test, for the proposed algorithms on CEC 2017 test functions using 30 dimension space value is also seen in Table 11. The findings show that the suggested algorithms outperformed the competition. In addition, the suggested OMVOD received the highest mean rank value (2.13), followed by DVBA (3.10), GSA (3.46), GSO (4.23), PSO (4.63), HPSOFF (4.86), FA (6.46), FFPSO (8.00) and MVO (8.03). The proposed OMVOD received the highest overall ranking rating (ranking first), followed by DVBA (ranking second), GSA (ranking third), GSO (ranking fourth), PSO (ranking fifth), HPSOFF (ranking sixth), FA (ranking seventh), FFPSO (ranking eighth) and MVO (ranking ninth). When we merged the opposition-based learning approach into the MVO with a disruption operator, we found that the best-proposed variant is opposition-based learning multi-verse optimizer with a disruption operator called OMVOD. When contrasted to other published approaches in the literature that used the same benchmark functions, OMVOD produced better performance.

5.3 Part 3: Experiments on CEC 2011 real-world problems

Additionally, for the standard benchmark functions presented in the sections earlier, seven more real-world problems
Fig. 3 Convergence behavior of the proposed algorithms on CEC 2017 test functions
(RWPs) have also been utilized to verify the effectiveness of the proposed method (OMVOD) Wang (2018), which are chosen from CEC 2011 real-world problems. These problems are as follows:

- RWP01: optimal control of a non-linear stirred tank reactor
- RWP02: spread spectrum radar Polly phase code design
- RWP03: large-scale transmission pricing problem
- RWP04: dynamic economic dispatch (DED) problem
- RWP05: hydrothermal scheduling problem
- RWP06: hydrothermal scheduling problem
- RWP07: hydrothermal scheduling problem

Seven real-world problems were used to evaluate the proposed method’s success and equate it to other related approaches from Wang (2018). To test the efficiency of many algorithms, these problems can be handled as continuous or discrete constrained problems. Five complementary approaches are used in this article to compare the findings of the proposed process. The population size is set to 30 for the experiments in this section. The maximum number of iterations is set to 1000. In Table 12, the results of 50 runs on seven RWPs are presented.

The proposed OMVOD algorithm outperformed the competition on six out of seven RWPs, as seen in Table 12 (except RWP06). The BBO algorithm, on the other hand, just got the best solution on one out of seven RWPs. Furthermore, the suggested OMVOD algorithm obtained the mean best results on six out of seven RWPs in terms of mean best results (except RWP06). The BBO algorithm, on the other hand, had the best results on one out of seven RWPs. The consequence of the outcomes of these two scenarios is continuity. On five out of seven RWPs, the proposed OMVOD algorithm got the worst scores (except RWO02 and RWP06). The BBO algorithm, on the other hand, got the worst best results on one of seven RWPs (RWO02), while the ABC got the worst best results on one of seven RWPs (RWP06). Finally, on five out of seven RWPs, the proposed OMVOD algorithm obtained the best STD results (except RWO01 and RWP06). The BBO algorithm, on the other hand, had the better STD results on two out of seven RWPs. In general, as compared to other related approaches, the proposed OMVOD algorithm generated the best overall results by 22 times. We concluded that the proposed method (OMVOD) could solve the optimization problems from different disciplines efficiently, reflecting the effect of the opposite-based learning method and disruption operator in developing the optimization process to reach the optimal solution effectively.

6 Conclusions and future directions

Using the multi-verse optimizer (MVO) algorithm, this paper introduced new alternative approaches. Two key stages are included in the proposed methods: To improve its exploitation quest, the basic MVO is combined with opposition-based learning (OBL) in the first phase, resulting in OMVO. The disruption operator (DO) is examined in the second phase to increase the discovery search of the OMVO while preserving the variety of the solutions, which is referred to as OMVOD.

Twenty-five CEC 2015 benchmark functions problems, thirty CEC 2017 benchmark functions problems and seven CEC 2011 real-world problems are included in the experiments. The proposed algorithms’ findings are comparable to those of many other algorithms that have been published in the literature. The best, average, and standard deviation of the fitness values is used to determine the efficacy of each algorithm. Compared to other related algorithms, the findings revealed that the OMVOD version is almost the best
### Performance comparisons of the best proposed algorithm (OMVOD) with other similar methods on CEC 2017 test functions using 30 Dim

| No. | Function | Comparative Algorithms | FA (Aydilek *2018*) | PSO (Ngo et al. *2016*) | GSA (Yazdani et al. *2014*) | GSO (Abedinpourshotorban et al. *2016*) | FFFPSO (Aydilek *2018*) | DVBA (Topal and Altun *2016*) | HPSOFF (Aydilek *2018*) | MVO (Mirmajili et al. *2016*) | OMVOD |
|-----|----------|------------------------|---------------------|------------------------|------------------|-------------------|------------------------|------------------------|------------------------|------------------------|--------|
| F1  | 3,3061E+10 | 5,5573E+09 | 3,1400E+08 | 2,2000E+08 | 1,1348E+11 | 1,0500E+05 | 6,0544E+09 | 3,3661E+11 | 1,5448E+07 | 2 |
| F2  | 1,8379E+39 | 3,9099E+32 | 1,5500E+10 | 1,4400E+10 | 1,0418E+57 | 2,9100E+04 | 3,1775E+34 | 2,4874E+49 | 3,5418E+27 | 4 |
| F3  | 2,1556E+05 | 1,3257E+05 | 8,6400E+04 | 9,9600E+04 | 6,7723E+09 | 1,5900E+04 | 1,3635E+05 | 2,8455E+09 | 1,4688E+04 | 1 |
| F4  | 6,8240E+03 | 9,0860E+02 | 1,3900E+03 | 1,8400E+03 | 4,4924E+04 | 1,8800E+02 | 1,2583E+03 | 3,5378E+04 | 1,5499E+02 | 1 |
| F5  | 8,7976E+02 | 8,0387E+02 | 2,0000E+01 | 2,0400E+01 | 1,1284E+03 | 2,0100E+01 | 8,1763E+02 | 5,2582E+03 | 7,5688E+01 | 1 |
| F6  | 6,7297E+02 | 6,5628E+02 | 3,0400E+01 | 1,2600E+02 | 7,2989E+02 | 2,3400E+02 | 6,7252E+02 | 6,8453E+02 | 2,0256E+02 | 3 |
| F7  | 1,7985E+03 | 1,1273E+03 | 1,6300E+03 | 9,6200E+01 | 2,9927E+03 | 1,0700E+03 | 1,1241E+03 | 1,7985E+03 | 1,0634E+03 | 1 |
| F8  | 1,1591E+03 | 1,0857E+03 | 1,4600E+03 | 4,1700E+02 | 1,3608E+03 | 3,0000E+01 | 1,1756E+03 | 1,6548E+04 | 1,2425E+02 | 2 |
| F9  | 1,3035E+04 | 6,6038E+03 | 1,6400E+03 | 8,0900E+02 | 3,0274E+04 | 2,1400E+03 | 9,0601E+03 | 1,0143E+04 | 2,4584E+02 | 1 |
| F10 | 9,4664E+03 | 9,2920E+03 | 3,9100E+03 | 7,6600E+03 | 1,0715E+04 | 7,9000E+03 | 8,9099E+03 | 9,8757E+04 | 6,2946E+02 | 1 |
| F11 | 1,1963E+04 | 3,6404E+03 | 4,3800E+03 | 1,7600E+04 | 3,4403E+04 | 3,9300E+02 | 4,5469E+03 | 3,4589E+04 | 3,3465E+02 | 3 |
| F12 | 3,0340E+09 | 6,4666E+08 | 2,0300E+02 | 1,1400E+00 | 2,7727E+10 | 9,8000E+01 | 3,3040E+08 | 3,9450E+09 | 7,9259E+05 | 1 |
| F13 | 1,1545E+09 | 1,9336E+08 | 3,7300E+04 | 5,5400E+04 | 2,7922E+10 | 5,1000E+04 | 2,7671E+07 | 4,1745E+09 | 2,0185E+03 | 4 |
| F14 | 1,8933E+06 | 1,1117E+06 | 7,2600E+05 | 5,5900E+05 | 5,6035E+07 | 2,0000E+04 | 1,1120E+06 | 1,3452E+07 | 4,4369E+04 | 2 |
| F15 | 1,0725E+08 | 3,9118E+07 | 1,3700E+03 | 2,9800E+03 | 7,1411E+09 | 2,4900E+03 | 1,5676E+06 | 1,7255E+08 | 2,3512E+03 | 3 |

**Note:** rankings are calculated as per the table's format and values.
| No. Ranking | Comparative Algorithms | Function | FA Aydilek (2018) | PSO Ngo et al. (2016) | GSA Yazdani et al. (2014) | GSO Abedinpourshotorban et al. (2016) | FPSO Aydilek (2018) | DVBA Topal and Altun (2016) | HPSOFF Aydilek (2018) | MVO Mirjalili et al. (2016) | OMVOD Our proposed |
|-------------|------------------------|----------|---------------------|---------------------|----------------------|-----------------------------|---------------------|------------------------|----------------------|------------------------|---------------------|
| F16 Rank 6  | 4.4164E+03 3.9514E+03 | 1.3700E+01 9.4800E+05 | 9.6352E+03 1.2700E+02 | 3.6383E+03 4.4645E+04 | 1.8761E+02 |
| F17 Rank 4  | 3.0044E+03 2.6556E+03 | 2.3000E+07 4.5000E+07 | 3.3347E+04 8.0900E+04 | 2.5254E+03 3.0694E+04 | 1.9954E+02 |
| F18 Rank 7  | 2.9281E+07 7.9043E+06 | 5.3600E+04 8.1600E+06 | 1.0554E+09 7.9000E+04 | 8.5890E+06 2.2581E+09 | 2.5574E+04 |
| F19 Rank 7  | 1.6603E+08 5.1700E+07 | 1.7300E+02 2.3200E+02 | 8.2628E+09 1.8100E+01 | 8.4341E+06 1.6632E+09 | 5.9657E+03 |
| F20 Rank 4  | 3.1162E+03 3.0177E+03 | 2.5400E+05 1.2600E+05 | 3.7975E+03 8.2900E+03 | 2.9083E+03 3.9641E+03 | 1.1485E+03 |
| F21 Rank 4  | 2.6405E+03 2.6009E+03 | 1.0100E+07 1.8800E+07 | 2.8880E+03 8.4500E+04 | 2.5982E+03 2.8705E+03 | 1.3257E+02 |
| F22 Rank 7  | 8.0003E+03 6.8423E+03 | 1.2700E+03 3.0500E+03 | 1.2117E+04 4.7000E+02 | 5.7433E+03 5.3253E+04 | 3.7366E+02 |
| F23 Rank 6  | 3.1542E+03 3.0819E+03 | 2.6600E+02 4.2600E+02 | 4.0749E+03 3.1600E+03 | 3.0969E+03 3.3652E+04 | 2.0124E+03 |
| F24 Rank 7  | 3.0563E+03 3.2139E+03 | 2.1500E+02 2.0300E+02 | 4.4529E+03 2.4900E+02 | 3.2719E+03 5.7656E+04 | 1.1582E+03 |
| F25 Rank 7  | 4.9480E+03 3.2145E+03 | 2.0500E+02 3.3500E+02 | 1.5956E+04 2.1900E+02 | 3.2320E+03 4.9540E+05 | 1.0623E+03 |
| F26 Rank 7  | 8.8041E+03 6.6461E+03 | 1.9000E+02 2.0500E+02 | 1.6163E+04 1.9000E+02 | 7.2368E+03 7.8541E+04 | 2.6540E+02 |
| F27 Rank 8  | 6.6059E+03 3.3573E+03 | 1.7400E+03 3.5500E+03 | 5.8099E+03 5.7000E+03 | 3.4360E+03 3.6456E+04 | 2.6543E+03 |
| F28 Rank 8  | 5.6940E+03 3.6388E+03 | 2.3100E+03 1.9900E+04 | 1.3535E+04 1.5600E+03 | 3.7938E+03 5.6356E+04 | 3.3855E+02 |
| F29 Rank 4  | 5.7051E+03 4.9240E+03 | 3.3000E+07 8.1300E+05 | 9.3924E+03 1.8400E+04 | 4.8406E+03 5.4565E+04 | 1.6749E+03 |
| F30 Rank 7  | 1.7163E+08 4.8514E+07 | 1.7900E+06 7.7300E+05 | 4.1425E+09 1.1900E+05 | 2.5749E+07 1.6452E+09 | 4.1495E+04 |
| Mean rank 6.46 | 4.63 | 3.46 | 4.23 | 8.00 | 3.10 | 4.86 | 8.03 | 2.13 |
| Final ranking 7 | 5 | 3 | 4 | 8 | 2 | 6 | 9 | 1 |
| Problem | Metric | Comparative Algorithms | ABC | BBO | DE | PSO | SGA | OMVOD |
|---------|--------|------------------------|-----|-----|----|-----|-----|-------|
| RWP01   |        |                        | 19.81 | 23.93 | 20.93 | 14.39 | 23.93 | 13.73 |
|         | Best   |                        | 21.43 | 23.93 | 22.93 | 20.47 | 23.93 | 18.55 |
|         | Mean   |                        | 22.91 | 23.93 | 23.93 | 21.83 | 23.93 | 21.14 |
|         | Worst  |                        | 5.1E-01 | 7.2E-15 | 8.4E-1 | 1.78E+00 | 7.2E-15 | 3.5E-10 |
| RWP02   |        |                        | 1.62 | 1.73 | 1.88 | 1.95 | 1.62 | 1.56 |
|         | Best   |                        | 2.34 | 2.13 | 2.44 | 2.42 | 2.17 | 2.04 |
|         | Mean   |                        | 2.88 | 2.49 | 2.77 | 2.89 | 2.90 | 2.50 |
|         | Worst  |                        | 2.4E-01 | 2.0E-01 | 2.2E-01 | 2.0E-01 | 3.0E-01 | 1.9E-01 |
| RWP03   |        |                        | 1.6E-06 | 8.9E-05 | 1.1E-06 | 1.4E-06 | 1.0E-06 | 7.6E-05 |
|         | Best   |                        | 2.0E+06 | 1.2E-06 | 1.3E-06 | 1.7E-06 | 1.3E-06 | 1.1E-06 |
|         | Mean   |                        | 2.0E-06 | 1.5E-06 | 1.5E-06 | 2.0E-06 | 1.6E-06 | 1.3E-06 |
|         | Worst  |                        | 1.7E-05 | 1.0E-05 | 9.3E-04 | 1.3E-05 | 1.3E-05 | 7.3E-04 |
| RWP04   |        |                        | 7.8E-07 | 5.7E-06 | 3.9E-07 | 8.6E-07 | 1.8E-07 | 3.9E-06 |
|         | Best   |                        | 3.3E-08 | 5.2E-07 | 2.8E-08 | 1.2E-08 | 8.8E-07 | 2.4E-07 |
|         | Mean   |                        | 4.0E-08 | 9.2E-07 | 3.3E-08 | 1.6E-08 | 1.5E-08 | 4.6E-07 |
|         | Worst  |                        | 3.2E-07 | 2.0E-07 | 2.4E-07 | 2.4E-07 | 2.3E-07 | 1.4E-07 |
| RWP05   |        |                        | 2.2E-08 | 1.2E-07 | 2.2E-08 | 7.2E-07 | 2.5E-07 | 3.4E-06 |
|         | Best   |                        | 1.5E-08 | 1.6E-07 | 6.2E-07 | 1.1E-08 | 4.0E-07 | 8.6E-06 |
|         | Mean   |                        | 1.9E-08 | 3.6E-07 | 8.5E-07 | 1.3E-08 | 5.9E-07 | 2.4E-07 |
|         | Worst  |                        | 2.1E-07 | 6.7E-06 | 1.1E-07 | 1.0E-07 | 9.1E-06 | 3.7E-06 |
| RWP06   |        |                        | 8.2E-07 | 4.6E-06 | 4.0E-07 | 7.9E-07 | 1.4E-07 | 5.2E-07 |
|         | Best   |                        | 1.5E-08 | 1.7E-07 | 6.2E-07 | 1.1E-08 | 3.7E-07 | 2.6E-07 |
|         | Mean   |                        | 2.0E-07 | 4.0E-07 | 8.5E-07 | 1.3E-08 | 5.5E-07 | 4.1E-07 |
|         | Worst  |                        | 2.2E-07 | 8.5E-06 | 1.1E-07 | 1.3E-07 | 9.3E-06 | 1.1E-07 |
| RWP07   |        |                        | 1.0E-08 | 2.5E-06 | 3.6E-07 | 7.9E-07 | 2.2E-07 | 2.4E-06 |
|         | Best   |                        | 1.5E-08 | 1.4E-07 | 5.9E-07 | 1.1E-08 | 3.8E-07 | 1.2E-07 |
|         | Mean   |                        | 1.9E-08 | 3.2E-07 | 8.1E-07 | 1.3E-08 | 4.1E-07 | 3.2E-07 |
|         | Worst  |                        | 2.1E-07 | 6.5E-06 | 8.9E-06 | 1.2E-07 | 8.9E-06 | 4.3E-06 |
| Obtained best |        |                        | 0 | 0 | 0 | 0 | 0 | 0 |
| Best   |        |                        | 0 | 1 | 0 | 0 | 0 | 6 |
| Mean   |        |                        | 0 | 1 | 0 | 0 | 0 | 6 |
| Worst  |        |                        | 1 | 2 | 0 | 0 | 0 | 5 |
| STD    |        |                        | 0 | 2 | 0 | 0 | 0 | 5 |
| Summation |        |                        | 1 | 6 | 0 | 0 | 0 | 22 |
optimizer in all test problems. In summary, solving real-world problems showed that the proposed OMVOD has a promising potential to be very useful in solving structural design problems with unfamiliar search spaces. Furthermore, the suggested OMVOD encourages discovery and extraction phases to be balanced while maintaining solution diversity. It does, however, suffer from a late convergence flaw.

In future works, we will consider the performance of other algorithms for new hybrid versions. We will also utilize them in the different optimization problems and multi-objective problems to achieve better results. Furthermore, we will also apply the other representative computational intelligence algorithms such as Monarch Butterfly Optimization (MBO), Earthworm Optimization Algorithm (EWA), Elephant Herding Optimization (EHO), Moth Search (MS) Algorithm, Arithmetic Optimization Algorithm (AOA) and Aquila Optimizer (AO).

Author Contributions The authors contributed to each part of this paper equally. The authors read and approved the final manuscript.

Funding The authors have not disclosed any funding.

Data Availability Enquiries about data availability should be directed to the authors.

Declaration Conflict of interest The authors declare that they have no conflicts of interest.

References

Abdelmadjid C, Mohamed S-A, Boussad B (2013) Cfd analysis of the volute geometry effect on the turbulent air flow through the turbocharger compressor. Energy Procedia 36(2):746–755
Abedinpourshotorban H, Shamsuddin SM, Beheshti Z, Jawawi DN (2016) Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. Swarm Evol Comput 26(1):8–22
Abualigah LM, Sawae IM, Khader AT, Rashaideh H, Al-Betar MA, Shehab M (2017) β-hill climbing technique for the text document clustering. New Trends in Information Technology (NTIT)–2017 6(2):60
Abualigah L, Shehab M, Alshinwan M, Alabool H (2019) Salp swarm algorithm: a comprehensive survey. Neural Comput Appl 6(2):1–21
Abualigah L, Gandomi AH, Elaziz MA, Hussien AG, Khasawneh AM, Alshinwan MN, Houssein EH (2020) Nature inspired optimization algorithms for text document clustering: a comprehensive analysis. Algorithms 13(12):345
Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609
Abualigah L, Elaziz MA, Hussien AG, Alsalibi B, Jalali SM, Gandomi AH (2020) Lightning search algorithm: a comprehensive survey. Appl Intell 1–24
Adeec UCEH (2000) Time complexity of genetic algorithms on exponentially scaled problems. Urbana 51(4):61–801
Adhim Khalid K, Hudaib Amjad, Al-Shboul Bashar (2019) Efficient requirement prioritization based on enhanced vector-merge optimizer. J Theor Appl Inf Technol 97(19)
Altubeeb AM, Mohsen AM, Abualigah L, Ghalab A (2021) Solving capacitated vehicle routing problem using cooperative firefly algorithm. Appl Soft Comput 108:107403
Andersson M, Bandaru S, Ng Amos HC, Syberfeldt A (2015) Parameter tuned cma-es on the cec’15 expensive problems. In: 2015 IEEE congress on evolutionary computation (CEC), pp 1950–1957. IEEE
Assiri AS, Hussien AG, Amin M (2020) Ant lion optimization: variants, hybrids, and applications. IEEE Access 8:77746–77764
Aydilek IB (2018) A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems. Appl Soft Comput 66(2):232–249
Bai Q (2010) Analysis of particle swarm optimization algorithm. Comput Inf Sci 3(1):180
Ceylan H, Ceylan H (2009) Harmony search algorithm for transport energy demand modeling. In: Music-inspired harmony search algorithm. Springer, pp 163–172
Dhiman R (2018) Moth-flame optimization technique for optimal coordination of directional overcurrent relay system. PhD thesis
Dorigo M, Birattari M, Stützle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–39
Elaziz MA, Selim IM, Xiong S (2017) Automatic detection of galaxy type from datasets of galaxies image based on image retrieval approach. Sci Rep 7(1):4463
Feng Y, Wang G-G, Dong J, Wang L (2018) Opposition-based learning monarch butterfly optimization with gaussian perturbation for large-scale 0–1 knapsack problem. Comput Electr Eng 67:454–468
Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68
Geng K, Ye C, Cao L, Liu L (2019) Multi-objective reentrant hybrid flowshop scheduling with machines turning on and off control strategy using improved multi-verse optimizer algorithm. Math Probl Eng 27(2)
Glover F (1977) Heuristics for integer programming using surrogate constraints. Decis Sci 8(1):156–166
Guo W, Lim CJ, Bi X, Sokhansanj S, Melin S (2013) Determination of effective thermal conductivity and specific heat capacity of wood pellets. Fuel 103:347–355
Hassan MH, Kamel S, Abualigah L, Eid A (2021) Development and application of slime mould algorithm for optimal economic dispatch. Expert Syst Appl 182:115205
Holland J (1975) Adaptation in natural and artificial systems: an introductory analysis with application to biology. Control Artif Intell 3(1):1–15
Hu C, Zhi L, Tian Z, Aijun Z, Chunpeil X (2016) A multi-verse optimizer with levy flights for numerical optimization and its application in test scheduling for network-on-chip. PLoS ONE 11(12):e0167341
Hussien AG (2021) An enhanced opposition-based salp swarm algorithm for global optimization and engineering problems. J Ambient Intell Humaniz Comput 1–22
Hussien AG, Amin M, Wang M, Liang G, Alsanad A, Gumaei A, Chen H (2020) Crow search algorithm: theory, recent advances, and applications. IEEE Access 8:173548–173565
Hussien AG, Onda D, Houssein EH, AngelA Juan XY (2020) Binary whale optimization algorithm for dimensionality reduction. Mathematics 8(10):1821
Jain P, Saxena A (2019) An opposition theory enabled moth flame optimizer for strategic bidding in uniform spot energy market. Eng Sci Technol Int J 22(4):1047–1067
Jangir P, Parmar SA, Trivedi IN, Bhesdadiya RH (2017) A novel hybrid particle swarm optimizer with multi verse optimizer for global numerical optimization and optimal reactive power dispatch problem. Eng Sci Technol Int J 20(2):570–586
Kennedy J (2010) Particle swarm optimization. Encycl Mach Learn 12(5):760–766
Kennedy J, Eberhart R (1995) Particle swarm optimization (pso). In: Proc. IEEE international conference on neural networks, Perth, Australia, pp 1942–1948
Koziel S, Yang X-S (2011) Computational optimization, methods and algorithms, vol 356. Springer, Berlin
Kulturel-Konak S, Smith AE, Coit DW (2003) Efficiently solving the redundancy allocation problem using tabu search. IIE Trans 35(6):515–526
Laith Abualigah, Dalia Yousri, Abd Elaziz Mohamed, Ewees Ahmed A, Al-quness MA, Gandomi Amir H (2021) A novel meta-heuristic optimization algorithm. Comput Ind Eng 157:107250
Li Z, Zhou Y, Zhang S, Song J (2016) Lévy-flight moth-flame algorithm for function optimization and engineering design problems. Math Probl Eng 16:25–47
Li J, Li Y, Tian S, Zou J (2019) Dynamic cuckoo search algorithm based on Taguchi opposition-based search. Int J Bio Inspir Comput 13(1):59–69
Li J, Xiao D, Lei H, Zhang T, Tian T (2020) Using cuckoo search algorithm with q-learning and genetic operation to solve the problem of logistics distribution center location. Mathematics 8(2):149
Li J, Yang Y-H, Lei H, Wang G-G (2020) Solving logistics distribution center location with improved cuckoo search algorithm. Int J Comput Intell Syst 14(1):676–692
Li J, Xiao D, Zhang T, Liu C, Li Y, Wang G (2021) Multi-swarm cuckoo search algorithm with q-learning model. Comput Intell Syst 14(1):676–692
Li J, Li YX, Tian SS, Xia JL (2019) An improved cuckoo search algorithm for self-adaptive knowledge learning. Neural Comput Appl 1–31
Liu Y, Wang G, Chen H, Dong H, Zhu X, Wang S (2011) An improved particle swarm optimization for feature selection. J Bionic Eng 8(2):191–200
Liu H, Ding G, Wang B (2014) Bare-bones particle swarm optimization with disruption operator. Appl Math Comput 238:106–122
Li W, Wang GG (2021) Elephant herding optimization using dynamic topology and biogeography-based optimization based on learning for numerical optimization. Eng Comput 1–29
Milad A (2013) Harmony search algorithm: strengths and weaknesses. J Comput Eng Inf Technol 2(1):1–7
Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst 89(2):228–249
Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513
Mirjalili S, Jangir P, Mirjalili SZ, Saremii S, Trivedi IN (2017) Optimization of problems with multiple objectives using the multi-verse optimization algorithm. Knowl Based Syst 134:50–71
Mohammad Abualigah L, Al-diabat M, Al Shinwan M, Dhou K, Alsalibi B, Said Hanandeh E, Shehab M (2020) Hybrid harmony search algorithm to solve the feature selection for data mining applications. Recent Adv Hybrid Metaheuristics Data Cluster 19–37
Murata T, Ishibuchi H, Tanaka H (1996) Multi-objective genetic algorithm and its applications to flowshop scheduling. Comput Ind Eng 30(4):957–968
Ngo TT, Sadollah A, Kim JH (2016) A cooperative particle swarm optimizer with stochastic movements for computationally expensive numerical optimization problems. J Comput Sci 13(1):68–82
Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization. Swarm Intell 1(1):33–57
Rahnamayan S, Tizhoosh HR, Salama MMA (2008) Opposition versus randomness in soft computing techniques. Appl Soft Comput 8(2):906–918
Reeves CR (1993) Improving the efficiency of tabu search for machine sequencing problems. J Oper Res Soc 44(4):375–382
Şahin CB, Abualigah L (2021) A novel deep learning-based feature selection model for improving the static analysis of vulnerability detection. Neural Comput Appl 1–19, 2018 New cuckoo search algorithms with enhanced exploration and exploitation properties. Expert Syst Appl 95(7):384–420
Shehab M, Khader AT, Al-Betar M (2016) New selection schemes for particle swarm optimization. IEEJ Trans Electron Inf Syst 136(12):1706–1711
Shehab M, Khader AT, Al-Betar MA (2017) A survey on applications and variants of the cuckoo search algorithm. Appl Soft Comput 61(1):1041–1059
Shehab M, Khader AT, Laouchedi M (2018) A hybrid method based on cuckoo search algorithm for global optimization problems. J Inf Comput Commun Tech 17(3):469–491
Shehab M, Abualigah L, AlHamad H, Alabool H, Alshinwan M, Khasawneh AM (2019) Moth-flame optimization algorithm: variants and applications. Neural Comput Appl 15(2):1–26
Shehab M, Daoud MS, AlMimi HM, Abualigah LM, Khader AT (2019) Hybridising cuckoo search algorithm for extracting the odf maxima in spherical harmonic representation. Int J Bio Inspired Comput 14(3):190–199
Shehab M, Alshuwabbak H, Abualigah L, Nagham AL-M (2020) Enhanced a hybrid moth-flame optimization algorithm using new selection schemes. Eng Comput 23(3):1–26
Shehab M, Khader AT, Al-Betar MA, Abualigah LM (2017) Hybridizing cuckoo search algorithm with hill climbing for numerical optimization problems. In: 2017 8th international conference on information technology (ICIT), pp 36–43. IEEE
Shehab M, Khader AT, Alia MA (2019) Enhancing cuckoo search algorithm by using reinforcement learning for constrained engineering optimization problems. In: 2019 IEEE Jordan international joint conference on electrical engineering and information technology (JEEIT)
Shehab M, Khader AT, Laouchedi M (2017) Modified cuckoo search algorithm for solving global optimization problems. In: International conference of reliable information and communication technology, no 2. Springer, pp 561–570
Tanweer MR, Suresh S, Sundararajan N (2015) Improved srpso algorithm for solving cec 2015 computationally expensive numerical optimization problems. In: 2015 IEEE congress on evolutionary computation (CEC), pp 1943–1949. IEEE
Tizhoosh HR (2005) Opposition-based learning: a new scheme for machine intelligence. In: International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC’06), vol 1, pp 695–701. IEEE
Topal AO, Altun O (2016) A novel meta-heuristic algorithm: dynamic virtual bats algorithm. Inf Sci 354(5):222–235
Wang G-G, Deb S, Gandomi AH, Alavi AH (2016) Opposition-based krill herd algorithm with cauchy mutation and position clamping. Neurocomputing 177:147–157
Wang G-G (2018) Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. Memetic Comput 10(2):151–164
Wang X, Yang J, Teng X, Xia W, Jensen R (2007) Feature selection based on rough sets and particle swarm optimization. Pattern Recogn Lett 28(4):459–471
Wang L, Yang R, Yin X, Niu Q, Pardalos PM, Fei M (2013) An improved adaptive binary harmony search algorithm. Inf Sci 232(4):58–87
Wright AH (1991) Genetic algorithms for real parameter optimization. In: Foundations of genetic algorithms, vol 1. Elsevier, pp 205–218
Yang XS, Deb S (2009) Cuckoo search via lévy flights. In: 2009 World congress on nature & biologically inspired computing (NaBIC), pp 210–214. IEEE
Yazdani S, Nezamabadi-pour H, Kamyab S (2014) A gravitational search algorithm for multimodal optimization. Swarm Evol Comput 14:1–14
Zhang H, Sun G (2002) Feature selection using tabu search method. Pattern Recog 35(3):701–711
Zingg DW, Nemec M, Pulliam TH (2008) A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization. European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique 17(1–2):103–126

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

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