Neural Comput & Applic (2016) 27:495–513
DOI 10.1007/s00521-015-1870-7

ORIGINAL ARTICLE

Multi-Verse Optimizer: a nature-inspired algorithm for global optimization

Seyedali Mirjalili · Seyed Mohammad Mirjalili · Abdolreza Hatamlou

Received: 6 October 2014 / Accepted: 22 February 2015 / Published online: 17 March 2015

Abstract This paper proposes a novel nature-inspired algorithm called Multi-Verse Optimizer (MVO). The main inspirations of this algorithm are based on three concepts in cosmology: white hole, black hole, and wormhole. The mathematical models of these three concepts are developed to perform exploration, exploitation, and local search, respectively. The MVO algorithm is first benchmarked on 19 challenging test problems. It is then applied to five real engineering problems to further confirm its performance. To validate the results, MVO is compared with four well-known algorithms: Grey Wolf Optimizer, Particle Swarm Optimization, Genetic Algorithm, and Gravitational Search Algorithm. The results prove that the proposed algorithm is able to provide very competitive results and outperforms the best algorithms in the literature on the majority of the test beds. The results of the real case studies also demonstrate the potential of MVO in solving real problems with unknown search spaces. Note that the source codes of the proposed MVO algorithm are publicly available at http://www.alimirjalili.com/MVO.html.

Keywords Optimization · Meta-heuristic · Algorithm · Benchmark · Genetic Algorithm · Particle Swarm Optimization · Heuristic

1 Introduction

Nature has been the main inspiration for the majority of the population-based stochastic optimization techniques. As the name of such techniques implies, they perform optimization randomly. The optimization process is usually started by creating a set of random solutions. These initial solutions are then combined, moved, or evolved over a predefined number of steps called iterations or generations. This is almost the main framework of all population-based algorithms. What makes an algorithm different from others in this field is the mechanism of combining, moving, or evolving the solutions during optimization.

For instance, Genetic Algorithms (GAs) [1] utilize the survival of the fitter individuals in nature in order to select the best solutions and then combine them based on the reproduction of chromosomes. Particle Swarm Optimization (PSO) [2] was inspired by social and individual thinking of birds when flying, so it obliges the candidate solutions to move around a search space with respect to their own personal best position obtained so far as well as the best position that the swarm found so far. Gravitational Search Algorithm (GSA) [3] uses the Newtonian laws of motion in order to move its search agents towards the promising regions of a search space.

Electronic supplementary material The online version of this article (doi:10.1007/s00521-015-1870-7) contains supplementary material, which is available to authorized users.

S. Mirjalili (✉)
School of Information and Communication Technology, Griffith University, Nathan Campus, Brisbane, QLD 4111, Australia
e-mail: seyedali.mirjalili@griffithuni.edu.au

S. Mirjalili
Queensland Institute of Business and Technology, Mt Gravatt, Brisbane, QLD 4122, Australia

S. M. Mirjalili
Zharfa Pajohesh System (ZPS) Co., Unit 5, NO. 30, West 208 St., Third Sq. Tehranpars, P.O. Box 1653745696, Tehran, Iran

A. Hatamlou
Department of Computer Science, Khoy Branch, Islamic Azad University, Khoy, Iran

Another common concepts among different population-based algorithms are exploration and exploitation. The former refers to the phase that an algorithm tries to discover different promising regions of a search space globally. Generally speaking, abrupt changes in candidate solutions are fruitful at this stage. In contrary, the latter concept is the convergence ability an algorithm around the obtained promising solutions in the exploration phase. A proper balance between exploration and exploitation can guarantee proceeding towards the global optimum.

Recently, there has been a growing interest in proposing new algorithms or improving the current ones in this field. A significant number of practical applications also accompanies the theoretical works. The reason of this remarkable popularity might be originated from the so-called No Free Lunch (NFL) theorem for optimization [4]. This theorem has been proved logically that there is no optimization technique for solving all optimization problems. The NFL theorem, obviously, makes this area of research open, in which researchers are allowed to improve/adapt the current algorithms for solving different problems or propose new algorithms for providing competitive results compared to the current algorithms.

In this work, a novel stochastic population-based algorithm is proposed called Multi-Verse Optimizer (MVO). As its name implies, MVO is inspired by the theory of multi-verse in physics. Three main concepts of the multi-verse theory (white hole, black hole, and wormhole) are mathematically modelled to construct the MVO. The rest of the paper is organized as follows.

Section 2 provides the literature review of the stochastic optimization techniques. Section 3 discusses the concepts of multi-verse theory and proposes the MVO algorithm. The test beds and results are demonstrated in Sect. 4. The real engineering problems are solved and discussed at the end of Sect. 4 as well. Eventually, Sect. 5 concludes the work and suggests some directions for future studies.

2 Related works

Generally speaking, stochastic optimization techniques can be divided into two main categories: single-solution-based versus population-based. The former class of algorithms starts the optimization process with a single random solution and improves it over a pre-defined number of generations. Simulate annealing (SA) [5], local searches [6, 7], and hill climbing [8] belong to this class of algorithms. The advantages of single-solution-based algorithms are: simplicity and low number of function evaluation. However, the main disadvantage is the high probability of entrapment in local optima. In addition, since at every run a single solution is involved, there is no information sharing, and the algorithm should deal with lots of issues such as local optima, isolation of optima, deceptiveness, bias of the search space, and premature convergence with only one candidate solution.

In contrast to single-solution-based algorithms, population-based algorithms initiate the optimization process with a set of random solutions and improve them over the course of iterations. This set of solutions is sometimes called candidate solutions' set. PSO, GA, Ant Colony Optimization (ACO) [9, 10], Artificial Bee Colonies (ABC) [11], and GSA [11] are some of the most popular algorithms in this class. The main advantage of the population-based algorithms is that there can be information exchange between the candidate solutions. Therefore, they can handle the issues such as local optima, isolation of optima, deceptiveness, bias of the search space, and premature convergence easier and faster. Another advantage is the less probability of entrapment in local solutions compared to the single-solution-based algorithms. The disadvantages of these algorithms are: less simplicity and the need for high number of function evaluation at each iteration.

The literature shows that the population-based algorithms have become a reliable alternative to single-solution-based algorithms due to the above-mentioned advantages. The application of these methods can also be found in a wide range of fields, emphasizing the merits of these techniques. Generally speaking, the design process of an algorithm starts with an inspiration. The inspiration could be from behaviour of creatures, natural phenomena, or social events. After the inspiration, different potential mathematical models are generated to design the algorithm. The best combination of mathematical models is then found by conducting experiments on various test beds.

The operators of algorithms in this field are usually designed to accomplish two phases: exploration versus exploitation. In the former phase, an algorithm should be equipped with mechanisms to search the search space as extensively as possible. In fact, promising regions of the search space are identified in this phase. In the exploitation phase, however, there should be emphasizes on local search and convergence towards promising areas obtained in the exploration phase. Exploration and exploitation are two conflicting stages with no specific mathematical definition. The exploration phase usually comes before exploitation, but there is a need to re-explore the search space in case of local optima stagnation, which is quite common in real problems with unknown search spaces.

Another challenge when designing an algorithm is the transition between exploration and exploitation. There is no clear rule for an algorithm to realize the most suitable time for transiting from exploration to exploration due to both unknown shape of search spaces and stochastic nature of population-based algorithms. The majority of population-

based algorithms have been tuned adaptively to smoothly transit between exploration and exploitation. For instance, the inertia weight in PSO is mostly decreased linearly from 0.9 to 0.4 in order to reduce the impacts of velocity vectors on particle movements and emphasize exploitation as iterations increase.

The above paragraphs show the challenges that a designer encounters when developing a new meta-heuristic. The following section proposes a novel meta-heuristic based on the concepts of multi-verse theory.

3 Multi-Verse Optimizer

3.1 Inspiration

The big bang theory [12] discusses that our universe starts with a massive explosion. According to this theory, the big bang is the origin of everything in this world, and there was nothing before that. Multi-verse theory is another recent and well-known theory between physicists [13]. It is believed in this theory that there are more than one big bang and each big bang causes the birth of a universe. The term multi-verse stands opposite of universe, which refers to the existence of other universes in addition to the universe that we all are living in [13]. Multiple universes interact and might even collide with each other in the multi-verse theory. The multi-verse theory also suggests that there might be different physical laws in each of the universes.

We chose three main concepts of the multi-verse theory as the inspiration for the MVO algorithm: white holes, black holes, and wormholes. A white hole has never seen in our universe, but physicists think that the big bang can be considered as a white hole and may be the main component for the birth of a universe [14]. It is also argued in the cyclic model of multi-verse theory [15] that big bangs/white holes are created where the collisions between parallel universes occur. Black holes, which have been observed frequently, behave completely in contrast to white wholes. They attract everything including light beams with their extremely high gravitational force [16]. Wormholes are those holes that connect different parts of a universe together. The wormholes in the multi-verse theory act as time/space travel tunnels where objects are able to travel instantly between any corners of a universe (or even from one universe to another) [17]. Conceptual models of these three key components of the multi-verse theory are illustrated in Fig. 1.

Every universe has an inflation rate (eternal inflation) that causes its expansion through space [18]. Inflation speed of a universe is very important in terms of forming stars, planets, asteroids, black holes, white holes, wormholes, physical laws, and suitability for life. It is argued in one of the cyclic multi-verse models [19] that multiple universes interact via white, black, and wormholes to reach a stable situation. This is the exact inspiration of the MVO algorithm, which is conceptually and mathematically modelled in the following subsection.

3.2 MVO algorithm

As discussed in the preceding section, a population-based algorithm divides the search process into two phases: exploration versus exploitation. We utilize the concepts of white hole and black hole in order to explore search spaces by MVO. In contrast, the wormholes assist MVO in exploiting the search spaces. We assume that each solution is analogous to a universe and each variable in the solution is an object in that universe. In addition, we assign each solution an inflation rate, which is proportional to the corresponding fitness function value of the solution. We also use the term time instead of the iteration in this paper since it is a common term in multi-verse theory and cosmology.

During optimization, the following rules are applied to the universes of MVO:

1. The higher inflation rate, the higher probability of having white hole.

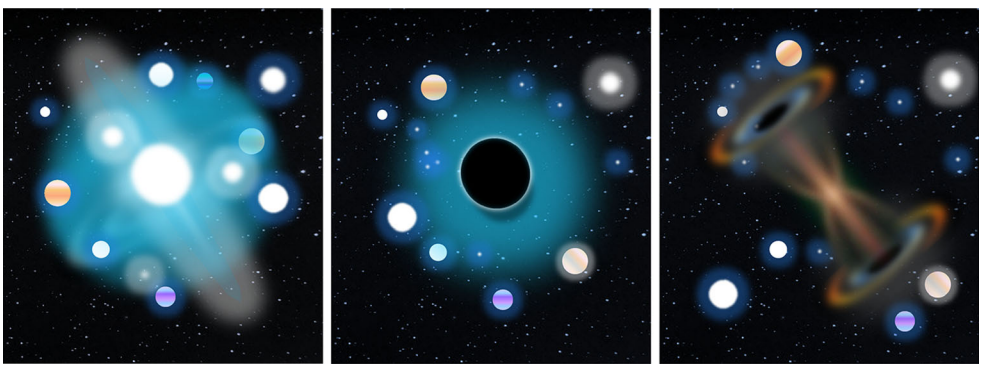

Fig. 1 *White* hole, *black* hole, and wormhole

2. The higher inflation rate, the lower probability of having black holes.
3. Universes with higher inflation rate tend to send objects through white holes.
4. Universes with lower inflation rate tend to receive more objects through black holes.
5. The objects in all universes may face random movement towards the best universe via wormholes regardless of the inflation rate.

The conceptual model of the proposed algorithm is illustrated in Fig. 2.

This figure shows that the objects are allowed to move between different universes through white/black hole tunnels. When a white/black tunnel is established between two universes, the universe with higher inflation rate is considered to have white hole, whereas the universe with less inflation rate is assumed to own black holes. The objects are then transferred from the white holes of the source universe to black holes of the destination universe. This mechanism allows the universes to easily exchange objects. In order to improve the whole inflation rate of the universes, we assume that the universes with high inflation rates are highly probable to have white holes. In contrary, the universes with low inflation rates have a high probability of having black holes. Therefore, there is always high possibility to move objects from a universe with high inflation rate to a universe with low inflation rate. This can guarantee the improvement of the average inflation rates of the whole universes over the iterations.

In order to mathematically model the white/black hole tunnels and exchange the objects of universes, we utilized a roulette wheel mechanism. At every iteration, we sort the universes based of their inflation rates and chose one of them by the roulette wheel to have a white hole. The following steps are done in order to do this.

Assume that

$$U = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^d \\ x_2^1 & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \vdots & \vdots \\ x_n^1 & x_n^2 & \dots & x_n^d \end{bmatrix}$$

where d is the number of parameters (variables) and n is the number of universes (candidate solutions):

$$x_i^j = \begin{cases} x_k^j & r1 < NI(Ui) \\ x_i^j & r1 \geq NI(Ui) \end{cases} \tag{3.1}$$

where x_i^j indicates the jth parameter of ith universe, Ui shows the ith universe, $NI(Ui)$ is normalized inflation rate of the ith universe, $r1$ is a random number in [0, 1], and x_k^j indicates the jth parameter of kth universe selected by a roulette wheel selection mechanism.

The pseudocodes for this part are as follows:

```
SU=Sorted universes
NI=Normalize inflation rate (fitness) of the universes
for each universe indexed by i
        Black_hole_index=i;
        for each object indexed by j
                r1=random([0,1]);
                if r1<NI(Ui)
                        White_hole_index= RouletteWheelSelection(-NI);
                        U(Black_hole_index,j)= SU(White_hole_index,j);
                end if
        end for
end for
```

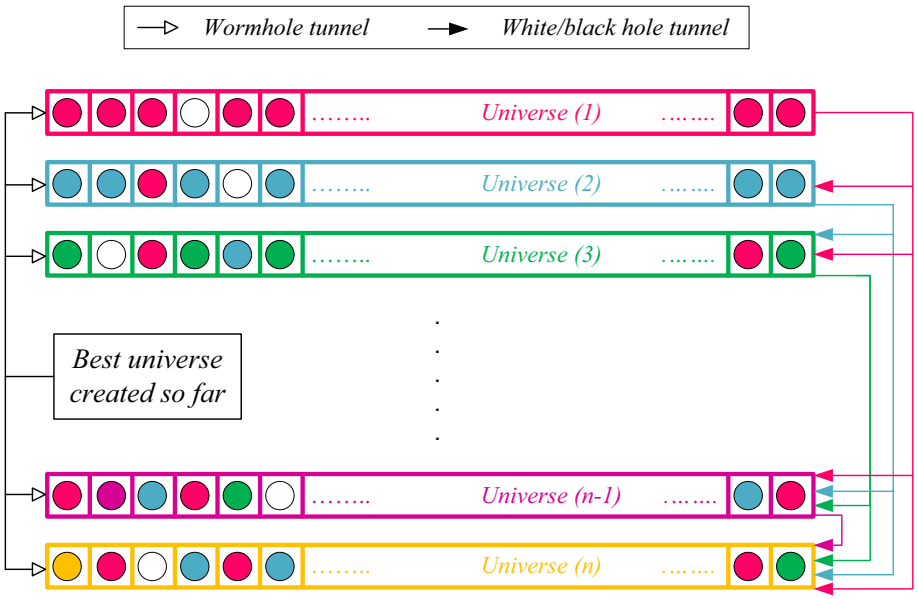

Fig. 2 Conceptual model of the proposed MVO algorithm ($I(U_1) > I(U_2) > \cdots > I(U_{n-1}) > I(U_n)$)

As may be seen in these pseudocodes and Eq. (3.1), the selection and determination of white holes are done by the roulette wheel, which is based on the normalized inflation rate. The less inflation rate, the higher probability of sending objects though white/black hole tunnels. Please note that $-NI$ should be changed to NI for the maximization problems. The exploration can be guaranteed using this mechanism since the universes are required to exchange objects and face abrupt changes in order to explore the search space.

With the above mechanism, the universes keep exchanging objects without perturbations. In order to maintain the diversity of universes and perform exploitation, we consider that each universe has wormholes to transport its objects through space randomly. In Fig. 2, white points represent transferred objects through the wormholes. It may be observed that the wormholes randomly change the objects of the universes without consideration of their inflation rates. In order to provide local changes for each universe and have high probability of improving the inflation rate using wormholes, we assume that wormhole tunnels are always established between a universe and the best universe formed so far. The formulation of this mechanism is as follows:

$$x_i^j = \begin{cases} \begin{cases} X_j + \text{TDR} \times \left((ub_j - lb_j) \times r4 + lb_j \right) & r3 < 0.5 \\ X_j - \text{TDR} \times \left((ub_j - lb_j) \times r4 + lb_j \right) & r3 \geq 0.5 \end{cases} & r2 < \text{WEP} \\ x_i^j & r2 \geq \text{WEP} \end{cases} \tag{3.2}$$

where X_j indicates the jth parameter of best universe formed so far, TDR is a coefficient, WEP is another coefficient, lb_j shows the lower bound of jth variable, ub_j is the upper bound of jth variable, x_i^j indicates the jth parameter of ith universe, and $r2$, $r3$, $r4$ are random numbers in [0, 1].

The pseudocodes are as follows (assuming that *ub* and *lb* indicate upper bound and lower bound of the variables):

It may be inferred from the pseudocodes and mathematical formulation that there are two main coefficients herein: wormhole existence probability (WEP) and travelling distance rate (TDR). The former coefficient is for defining the probability of wormhole's existence in universes. It is required to increase linearly over the iterations in order to emphasize exploitation as the progress of optimization process. Travelling distance rate is also a factor to define the distance rate (variation) that an object can be teleported by a wormhole around the best universe obtained so far. In contrast to WEP, TDR is increased over the iterations to have more precise exploitation/local search around the best obtained universe. Wormhole existence and travelling distance rates are illustrated in Fig. 3. The adaptive formula for both coefficients are as follows:

$$\text{WEP} = \min + l \times \left(\frac{\max - \min}{L} \right) \tag{3.3}$$

where min is the minimum (0.2 in this paper), max is the maximum (1 in this paper), l indicates the current iteration, and L shows the maximum iterations.

$$\text{TDR} = 1 - \frac{l^{1/p}}{L^{1/p}} \tag{3.4}$$

where p (in this paper equals 6) defines the exploitation accuracy over the iterations. The higher p, the sooner and more accurate exploitation/local search.

Note that WEP and TDR can be considered as constants as well, but we recommend adaptive values according to the results of this paper.

After all, the general steps and pseudocodes of the MVO algorithm are provided in the "Appendix 1".

In the MVO algorithm, the optimization process starts with creating a set of random universes. At each iteration, objects in the universes with high inflation rates tend to move to the universes with low inflation rates via white/black holes. Meanwhile, every single universe faces

```
for each universe indexed by i
  for each object indexed by j
        r2=random([0,1]);
        if r2<Wormhole_existance_probability
              r3= random([0,1]);
              r4= random([0,1]);
              if r3<0.5
                    U(i,j)=Best_universe(j) + Travelling_distance_rate * (( ub(j) - lb(j)) *
                     r4 + lb(j));
              else
                    U(i,j)=Best_universe(j) - Travelling_distance_rate * (( ub(j) - lb(j)) *
                    r4 + lb(j));
              end if
        end if
        end for
end for
```

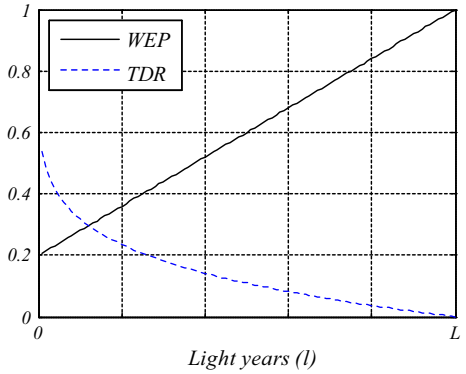

Fig. 3 Wormhole existence probability (WEP) versus travelling distance rate (TDR)

random teleportations in its objects through wormholes towards the best universe. This process is iterated until the satisfaction of an end criterion (a pre-defined maximum number of iterations, for instance).

The computational complexity of the proposed algorithms depends on number of iterations, number of universes, roulette wheel mechanism, and universe sorting mechanism. Sorting universe is done in every iteration, and we employ the Quicksort algorithm, which has the complexity of $O(n \log n)$ and $O(n^2)$ in the best and worst case, respectively. The roulette wheel selection is run for every variable in every universe over the iterations and is of $O(n)$ or $O(\log n)$ based on the implementation. Therefore, the overall computational complexity is as follows:

$$O(\text{MVO}) = O(l(O(\text{Quick sort}) + n \times d \times (O(\text{roulette wheel})))) \tag{3.5}$$

$$O(\text{MVO}) = O\left(l\left(n^2 + n \times d \times \log n\right)\right) \tag{3.6}$$

where n is the number of universes, l is the maximum number of iterations, and d is the number of objects.

To see how the proposed algorithm theoretically has the potential to solve optimization problems, some observations are as follows:

- White holes are more possible to be created in the universes with high inflation rates, so they can send objects to other universes and assist them to improve their inflation rates.
- Black holes are more likely to be appeared in the universes with low inflation rates so they have higher probability to receive objects from other universes. This again increases the chance of improving inflation rates for the universes with low inflation rates.
- White/black hole tunnels tend to transport objects from universes with high inflation rates to those with low inflation rates, so the overall/average inflation rate of all universes is improved over the course of iterations.
- Wormholes tend to appear randomly in any universe regardless of inflation rate, so the diversity of universes can be maintained over the course of iterations.
- While/black hole tunnels require universes to abruptly change, so this can guarantee exploration of the search space.
- Abrupt changes also assist resolving local optima stagnations.
- Wormholes randomly re-span some of the variables of universes around the best solution obtained so far over the course of iterations, so this can guarantee exploitation around the most promising region of the search space.
- Adaptive WEP values smoothly increase the existence probability of wormholes in universes. Therefore, exploitation is emphasized during optimization process.
- Adaptive TDR values decrease the travelling distance of variables around the best universe, a mechanism for increasing the accuracy of local search over the iterations.
- The convergence of the proposed algorithm is guaranteed by emphasizing exploitation/local search proportional to the number of iterations.

The following section investigates these theoretical claims in practice.

4 Results and discussion

We selected 19 benchmark functions in the literature as test beds for comparison [20–23]. To improve the difficultly of the test functions $F1$–$F13$, we randomly shift the optima of test functions at every run. The rest of the test functions, $F14$–$F19$, are six composite benchmark functions provided by CEC 2005 special session [24, 25]. It is worth mentioning here that the dimension of the test functions is also considered 50. The test functions are listed in Tables 1, 2, and 3. The shape of the composite functions is also illustrated in Fig. 4.

The employed test functions can be divided into three groups: unimodal, multi-modal, and composite. Unimodal test functions have one global optimum, so they are suitable for benchmarking the exploitation of algorithms. The multi-modal test functions, however, have a global optimum as well as multiple local optima with number exponentially increasing with the dimension. Eventually, composite test functions are considered as the combination of the first two groups with random rotation, shift, and biasing. The composite test functions are more similar to the real search spaces and good for benchmarking the exploration and exploitation of algorithms simultaneously.

In order to have a fair comparison, the algorithm is run 30 times. We also conduct Wilcoxon's nonparametric

Table 1 Unimodal benchmark functions

Function	Dim	Range	$f_{\min}$
$F_1(x) = \sum_{i=1}^{n} x_i^2$	50	[−100, 100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	50	[−10, 10]	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	50	[−100, 100]	0
$F_4(x) = \max_i \{ \lvert x_i \rvert, 1 \le i \le n \}$	50	[−100, 100]	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$	50	[−30, 30]	0
$F_6(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$	50	[−100, 100]	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1)$	50	[−1.28, 1.28]	0

Table 2 Multi-modal benchmark functions

Function	Dim	Range	$f_{\min}$
$F_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert} \right)$	50	[−500, 500]	
$F_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$	50	[−5.12, 5.12]	0
$F_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$	50	[−32, 32]	0
$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	50	[−600, 600]	0
$F_{12}(x) = \frac{\pi}{n} \left\{ 10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 \left[1 + 10\sin^2(\pi y_{i+1}) \right] + (y_n - 1)^2 \right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	50	[−50, 50]	0
$F_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2 \left[1 + \sin^2(3\pi x_i + 1) \right] + (x_n - 1)^2 \left[1 + \sin^2(2\pi x_n) \right] \right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	50	[−50, 50]	0

statistical test over this 30 runs in order to draw a statistically meaningful conclusion. Such statistical test must be done due to the stochastic nature of meta-heuristics [26, 27]. Any p values <0.05 evidence the statistical significant superiority of the results. After all, the statistical results are provided in Tables 4, 5, and 6. Note that the number of universes is 30 and the maximum number of iterations is equal to 500. WEP is increased linearly from 0.2 to 1, and TDR is decreased from 0.6 to 0 using Eqs. (3.3) and (3.4), respectively.

For the verification of the results, the MVO algorithm is compared to PSO [2] as the best algorithm among SI-based techniques, GA [1] as the best evolutionary algorithms, GSA [3] as one of the best and recent physics-based algorithms, and Grey Wolf Optimizer (GWO) [28] as one of the most recent algorithms.

4.1 Unimodal test functions and exploitation

The results of Table 4 show that the proposed algorithm is able to provide very competitive results on the unimodal test functions. The p values in Table 7 also prove that the superiority is significant in the majority of the cases. This testifies that the proposed algorithm has a high exploitation ability, which is due to the integrated adaptive WEP/TDR constants and wormholes combined that assist MVO to provide high exploitation.

4.2 Multi-modal test functions and exploration analysis

After proving the exploitation of MVO, we are going to discuss the exploration of this algorithm. The results of multi-modal test function can be seen in Table 5. The results of this table evidence that the proposed algorithm is again able to provide very promising performance. It may be observed that this algorithm outperforms others on $F8$, $F11$, $F12$, and $F13$. The p values of Table 7 suggest that the superiority of the MVO algorithm is statistically significant in the majority of times. In addition, p values show that the MVO algorithm rejects the null hypothesis on $F9$ and $F10$, proving that the results of this algorithm are very competitive.

Table 3 Composite benchmark functions

Function	Dim	Range	$f_{\min}$
$F_{14}(CF1)$	20	[−5, 5]	0
$f_1,f_2,f_3,\ldots,f_{10}$ = Sphere function			
$[\sigma_1, \sigma_2, \sigma_3,\ldots, \sigma_{10}] = [1, 1, 1,\ldots, 1]$			
$[\lambda_1, \lambda_2, \lambda_3\ldots, \lambda_{10}] = [5/100, 5/100, 5/100, \ldots, 5/100]$			
$F_{15}(CF2)$	20	[−5, 5]	0
$f_1,f_2,f_3,\ldots,f_{10}$ = Griewank's function			
$[\sigma_1, \sigma_2, \sigma_3,\ldots, \sigma_{10}] = [1, 1, 1,\ldots, 1]$			
$[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [5/100, 5/100, 5/100, \ldots, 5/100]$			
$F_{16}(CF3)$	20	[−5, 5]	0
$f_1,f_2,f_3,\ldots,f_{10}$ = Griewank's function			
$[\sigma_1, \sigma_2, \sigma_3,\ldots, \sigma_{10}] = [1, 1, 1,\ldots, 1]$			
$[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [1, 1, 1, \ldots, 1]$			
$F_{17}(CF4)$	20	[−5, 5]	0
f_1,f_2 = Ackley's function			
f_3,f_4 = Rastrigin's function			
f_5,f_6 = Weierstrass function			
f_7,f_8 = Griewank's function			
f_9,f_{10} = Sphere function			
$[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1,\ldots, 1]$			
$[\lambda_1, \lambda_2, \lambda_3,\ldots,\lambda_{10}] = [5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]$			
$F_{18}(CF5)$	20	[−5, 5]	0
f_1,f_2 = Rastrigin's function			
f_3,f_4 = Weierstrass function			
f_5,f_6 = Griewank's function			
f_7,f_8 = Ackley's function			
f_9,f_{10} = Sphere function			
$[\sigma_1, \sigma_2, \sigma_3,\ldots, \sigma_{10}] = [1, 1, 1,\ldots, 1]$			
$[\lambda_1, \lambda_2, \lambda_3,\ldots,\lambda_{10}] = [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$			
$F_{19}(CF6)$	20	[−5, 5]	0
f_1,f_2 = Rastrigin's function			
f_3,f_4 = Weierstrass function			
f_5,f_6 = Griewank's function			
f_7,f_8 = Ackley's function			
f_9,f_{10} = Sphere function			
$[\sigma_1, \sigma_2, \sigma_3,\ldots, \sigma_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$			
$[\lambda_1, \lambda_2, \lambda_3,\ldots,\lambda_{10}] = [0.1 \times 1/5, 0.2 \times 1/5, 0.3 \times 5/0.5, 0.4 \times 5/0.5, 0.5 \times 5/100,$			
$0.6 \times 5/100, 0.7 \times 5/32, 0.8 \times 5/32, 0.9 \times 5/100, 1 \times 5/100]$			

Superior exploration of the proposed algorithm is due to the white and black holes that allow universes to exchange different objects. Travelling an object from a universe to the other causes sudden changes and promotes exploration. This is similar to the crossover operator in GA that highly emphasizes exploration of the search spaces.

Exploration always comes with local optima avoidance. In fact, stagnation in local solutions can be resolved by promoting exploration. In addition to the discussion provided in the preceding paragraph, local optima avoidance of MVO also can be reasoned by the employed black/white hole concepts. Abrupt changes of universe are very useful in case of local optima stagnation.

4.3 Composite test functions and exploration/exploitation analysis

This subsection discusses the balance between exploration and exploitation, which is an essential feature of stochastic

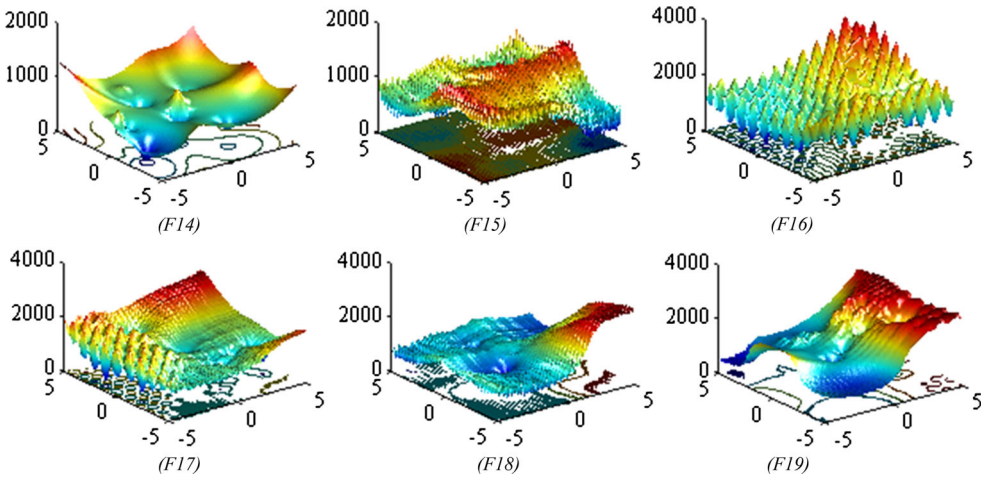

Fig. 4 Search space of composite benchmark functions

Table 4 Results of unimodal benchmark functions

F	MVO		GWO		GSA		PSO		GA	
	Ave	SD	Ave	SD	Ave	SD	Ave	SD	Ave	SD
*F*1	2.08583	0.648651	2319.19	1237.109	2983.667	903.3827	3.552364	2.853733	27,187.58	2745.82
*F*2	15.92479	44.7459	14.43166	5.923015	10.96518	10.54968	8.716272	4.929157	68.6618	6.062311
F3	453.2002	177.0973	7278.133	2143.116	113,740.4	78,786.15	2380.963	1183.351	48,530.91	8249.75
*F*4	3.123005	1.582907	13.09729	11.3469	32.2563	6.226765	21.5169	6.71628	62.99326	2.535643
*F*5	1272.13	1479.477	3,425,462	3,304,309	7582.498	7314.818	1132.486	1357.967	65,361,620	29,714,021
*F*6	2.29495	0.630813	5009.442	3028.875	74,617.45	8231.224	86.62074	147.3067	49,574.1	8545.149
*F*7	0.051991	0.029606	0.408082	0.119544	21.16092	12.1566	0.577434	0.318544	18.72524	4.935256

Table 5 Results of multi-modal benchmark functions

F	MVO		GWO		GSA		PSO		GA	
	Ave	SD	Ave	SD	Ave	SD	Ave	SD	Ave	SD
*F*8	−11,720.2	937.1975	−10,739.5	1162.793	−4638.41	805.0488	−6727.59	1352.882	−10,698.6	602.3045
*F*9	118.046	39.34364	89.13475	37.95765	128.0103	26.90054	99.83202	24.62872	273.2519	29.55218
*F*10	4.074904	5.501546	9.452571	3.467608	1.654073	1.583499	4.295044	1.308386	18.59657	0.351737
*F*11	0.938733	0.059535	22.51942	26.68168	1021.705	82.95486	624.3092	105.3874	353.3655	77.26729
*F*12	2.459953	0.791886	3,200,008	6,746,208	741,596.9	624,375.5	13.38384	8.969122	2.21e+08	1.1e+08
*F*13	0.222672	0.086407	7,815,082	16,475,640	6,670,046	5,719,826	21.11298	12.83179	4.49e+08	2.26e+08

algorithms for solving real problems. The composite test functions are suitable for examining the balance of exploration and exploitation in this regard. As presented in Table 6, the results of the algorithms on composite test functions are very close. This is due to the difficulty of this set of test functions. The results prove that the MVO algorithm provides highly competitive results on the composite test functions as well. This evidences that the MVO algorithm properly balances exploration and exploration during optimization. This is firstly originated from the employed adaptive WEP that assists to smoothly transit between exploration and exploitation phases.

4.4 Convergence analysis

To confirm the convergence of the proposed algorithm, we provide the search histories, trajectories in the first dimension, average fitness of all universes, and convergence curves in Fig. 7 similarly to [28, 29]. From search history point of view, second columns of Fig. 5 show that the

Table 6 Results of composite benchmark functions

F	MVO		GWO		GSA		PSO		GA	
	Ave	SD	Ave	SD	Ave	SD	Ave	SD	Ave	SD
*F*14	10.00017	31.62288	0.00057	0.00044	30.09736	95.1762	20	63.2455	6.41e−05	9.25e−05
*F*15	30.00705	48.30615	30.0135	67.4900	72.92938	62.39415	60	84.3274	12.6899	20.2671
*F*16	50.00061	52.70461	20.0020	42.1642	199.5152	168.6623	20	42.1637	10.0435	31.7419
*F*17	190.3	128.6659	291.347	145.615	420.8455	60.07176	170	48.3045	116.353	80.5314
*F*18	160.5312	158.2887	60.0005	51.6393	266.2412	147.4886	30	48.3045	21.3184	41.5775
*F*19	440.005	51.64	380.031	103.286	421.0165	25.80179	400.330	115.792	216.693	160.751

Table 7 *p* values of the Wilcoxon rank-sum test over 10 runs ($p \geq 0.05$ have been underlined)

F	MVO	GWO	GSA	PSO	GA
F1	N/A	0.002827	0.000183	0.185877	0.000183
F2	0.009108	0.053903	0.909722	N/A	0.000183
F3	N/A	0.000183	0.000183	0.000183	0.000183
F4	N/A	0.140465	0.000183	0.000183	0.000183
F5	0.677585	0.10411	0.005795	N/A	0.000183
F6	N/A	0.000183	0.000182	0.007284	0.000183
F7	N/A	0.000183	0.000183	0.000183	0.000183
F8	N/A	0.053903	0.000183	0.000183	0.021134
F9	0.121225	N/A	0.002827	0.185877	0.000183
F10	0.121225	0.001315	N/A	0.002827	0.000183
F11	N/A	0.005795	0.000183	0.000183	0.000183
F12	N/A	0.025748	0.000183	0.00033	0.000183
F13	N/A	0.075662	0.000183	0.000183	0.000183
F14	0.064022	0.001315	0.002827	0.001745	N/A
F15	0.472676	0.677585	0.037635	0.466409	N/A
F16	0.064022	0.185877	0.003611	0.01385	N/A
F17	0.088973	0.007285	0.000183	0.969146	N/A
F18	0.021134	0.384673	0.00058	0.048538	N/A
F19	0.01133	0.088973	0.001706	0.206046	N/A

universes tend to scatter around the best solutions in the search space over the course of iterations. In order to see the changes in the search agents' variables, the trajectories of the first universe in the first dimension are depicted in the fourth column of Fig. 5 as well. It is evident from the trajectories that universes faced abrupt changes in the initial steps of iterations. These sudden changes are decreased gradually over the course of iterations. According to Berg et al. [30], this behaviour can guarantee that a population-based algorithm eventually converges to a point and search locally in a search space.

Another evidence of convergence and improvement in universes can be concluded from the average fitness of all universes in Fig. 5. The fifth column of this figure shows descending trends in the average fitness over the iterations. This evidences that the model proposed for MVO algorithm can successfully improve the fitness of all universes and guarantee finding better solutions as iteration increases. This can be discussed and reasoned according to the white and black hole concepts of the proposed algorithm. Since the variables of universes tend to move from a fit universe to a less fit universe, similarly to the evolution concepts in GA, the whole universes and their fitness average tend to be improved over the course of iterations. In addition, we save the best universe and move it to the next iteration, so the best universe formed so far is always available to improve the fitness of other universes.

This is similar to the elitism concepts in GA and other evolutionary algorithms. Eventually, the convergence curves in Fig. 5 (last column) show that MVO algorithm finds a better solution iteration by iteration.

4.5 Constrained optimization and classical engineering problems

In this subsection, a set of five constrained real engineering problems is solved in order to further investigate the performance of the proposed MVO algorithm. The following problems have several inequality constraints, so the MVO algorithm should be able to handle them during optimization. Since search agents of the MVO algorithm are fitness independent, the mechanism of this algorithm does not need to be altered for handling constraints. We utilize the simplest constraint handling method, death penalty functions, where search agents are assigned big objective function values if they violate any of the constraints. Assigning big fitness values to the violated universes requires them to have high number of black holes and less number of white holes. Therefore, they mostly tend to receive objects rather than sending them.

4.5.1 Welded beam design using MVO

As its name implies, this problem deals with designing a welded beam with the lowest fabrication cost. The overall

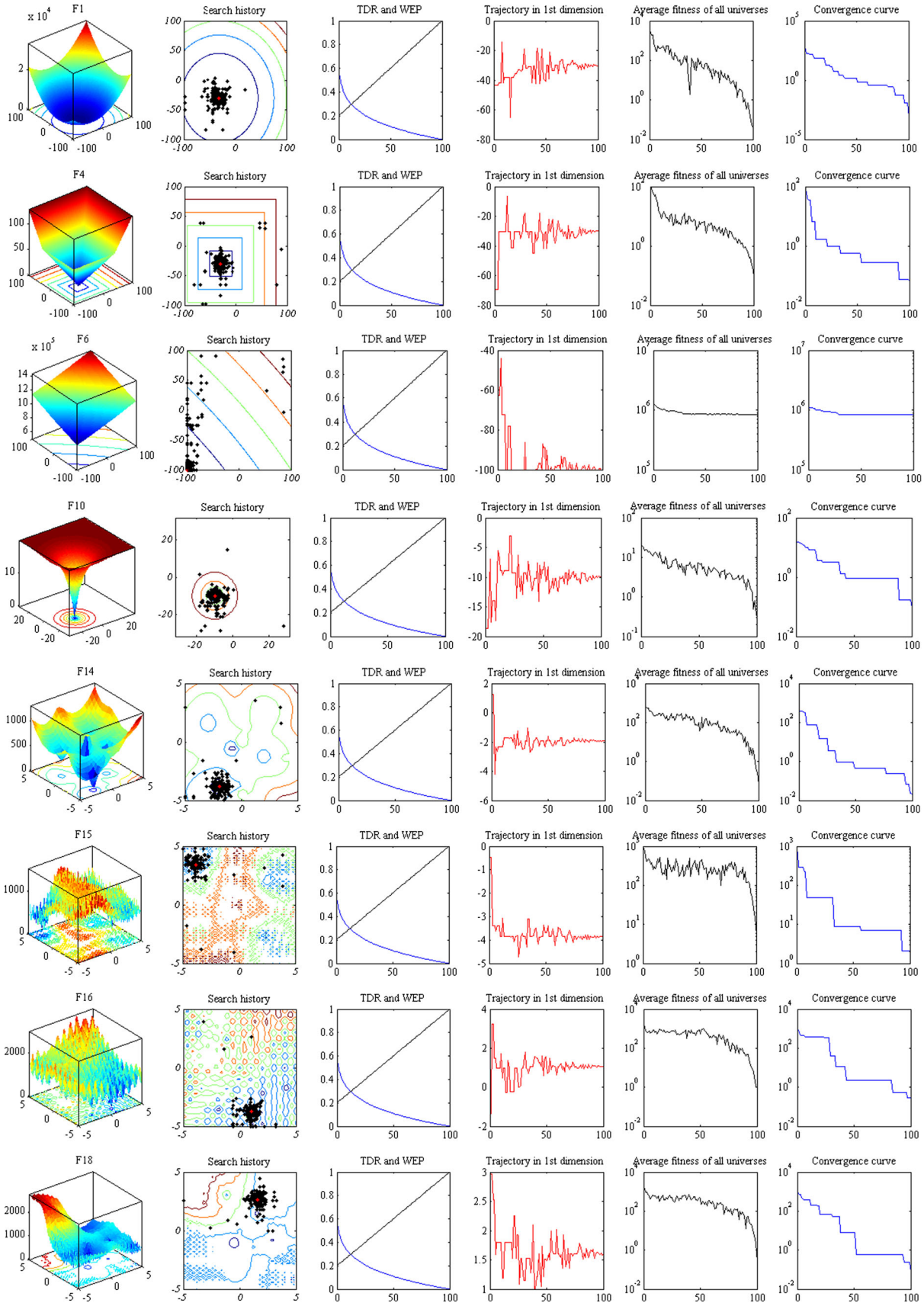

Fig. 5 Search history, trajectory in first dimension, average fitness of all universes and convergence

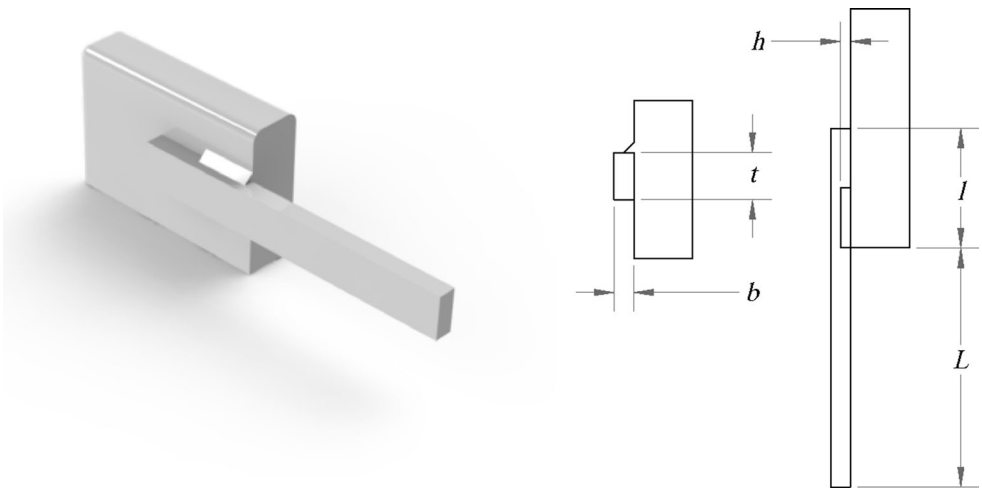

Fig. 6 Design parameters of the welded beam design problem

Table 8 Comparison results of the welded beam design problem

Algorithm	Optimal values for variables				Optimal cost
	h	*l*	*t*	*b*	
MVO	0.205463	3.473193	9.044502	0.205695	1.72645
GSA	0.182129	3.856979	10.0000	0.202376	1.87995
CPSO	0.202369	3.544214	9.048210	0.205723	1.72802
GA (Coello)	N/A	N/A	N/A	N/A	1.8245
GA (Deb)	N/A	N/A	N/A	N/A	2.3800
GA (Deb)	0.2489	6.1730	8.1789	0.2533	2.4331
HS (Lee and Geem)	0.2442	6.2231	8.2915	0.2443	2.3807
Random	0.4575	4.7313	5.0853	0.6600	4.1185
Simplex	0.2792	5.6256	7.7512	0.2796	2.5307
David	0.2434	6.2552	8.2915	0.2444	2.3841
Approx	0.2444	6.2189	8.2915	0.2444	2.3815

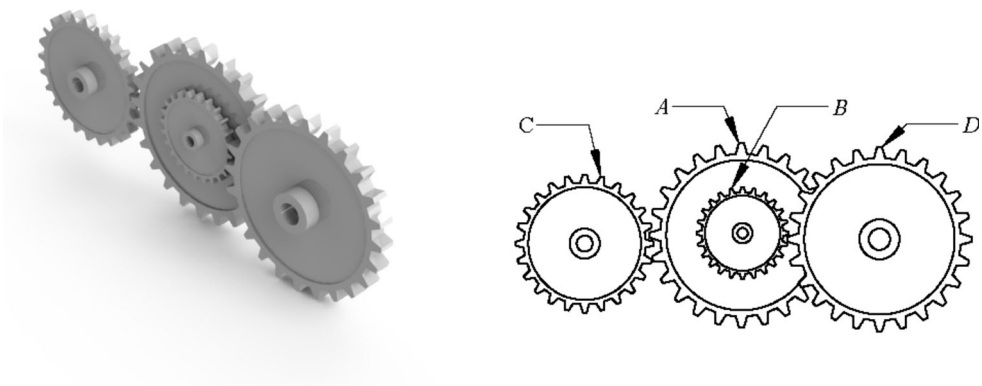

Fig. 7 Gear train design problem

system and structural parameters are illustrated in Fig. 6 [31].

This figure shows that there are four design parameters for this problem to be optimized: thickness of weld (h), length of attached part of bar (l), the height of the bar (t), and thickness of the bar (b). The obtained design should not violate seven constraints when a load applies on the top of the bar. Some of the constraints are as follows: side constraints, shear stress (τ), end deflection of the beam (δ), buckling load on the bar (P_c), and bending stress in the beam (θ). Full details for this problem can be found in the "Appendix 2".

Table 9 Comparison results of the gear train design problem

Algorithm	Optimal values for variables				Optimal gear ratio
	n_A	n_B	n_C	n_D	
MVO	43	16	19	49	2.7009e−012
ABC [39]	49	16	19	43	2.7009e−012
MBA [39]	43	16	19	49	2.7009e−012
Kannan and Kramer [40]	33	15	13	41	2.1469e−08
Deb and Goyal [41]	49	16	19	43	2.7019e−012
CS [42]	43	16	19	49	2.7009e−012
ISA [38]	N/A	N/A	N/A	N/A	2.7009e−012

Fig. 8 Three-bar truss design problem

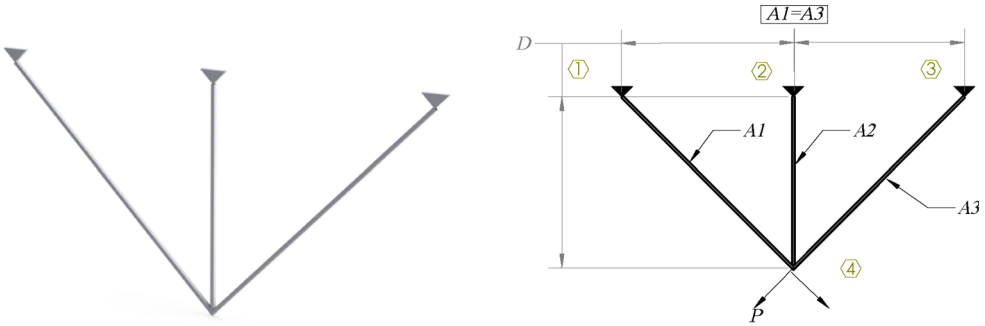

Table 10 Comparison results of the three-bar truss design problem

Algorithm	Optimal values for variables		Optimal weight
	x_1	x_2	
MVO	0.78860276	0.40845307	263.8958499
DEDS [43]	0.78867513	0.40824828	263.8958434
PSO-DE [44]	0.7886751	0.4082482	263.8958433
MBA [39]	0.7885650	0.4085597	263.8958522
Ray and Sain [45]	0.795	0.395	264.3
Tsa [46]	0.788	0.408	263.68
CS [42]	0.78867	0.40902	263.9716

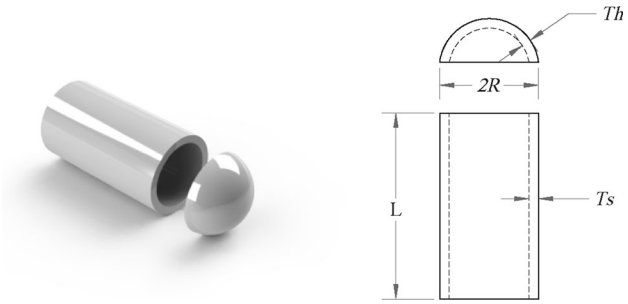

Fig. 9 Pressure vessel design problem

This problem has been very popular in the literature. Some of the works that solved this problem are as follows: GA [31–33], CPSO [34], HS [35], Richardson's random method, Simplex method, Davidon–Fletcher–Powell, and Griffith and Stewart's successive linear approximation [36]. We compare our algorithm with these works and provide the results in Table 8.

The results of the algorithms on welded beam design problem show that the MVO algorithm is able to find a design with the minimum cost. The superiority of the results compared to mathematical approaches is due to the local optima avoidance of this algorithm. The proposed algorithm also outperforms PSO, GA, and GSA, which shows that this algorithm is able to effectively search for optimal solutions in real search spaces as well.

4.5.2 *Gear train design using MVO*

As may be seen in Fig. 7 and "Appendix 2", this problem has four parameters, and the objective is the minimization of gear ratio [37]. The gear ratio is calculated as follows [38]:

$$\text{Gear ratio} = \frac{\text{angular velocity of output shaft}}{\text{angular velocity of input shaft}}$$

The parameters of this problem are discrete with the increment size of 1 since they define the teeth of the gears (n_A, n_B, n_C, n_D). There constraints are only limited the variable ranges.

We solve this problem with MVO and provide the results in Table 9. This table shows that the MVO algorithm

Table 11 Comparison results for pressure vessel design problem

Algorithm	Optimal values for variables				Optimum cost
	T_s	T_h	R	L	
MVO	0.8125	0.4375	42.0907382	176.738690	6060.8066
GSA	1.1250	0.6250	55.9886598	84.4542025	8538.8359
PSO (He and Wang)	0.8125	0.4375	42.091266	176.746500	6061.0777
GA (Coello)	0.8125	0.4345	40.323900	200.000000	6288.7445
GA (Coello and Montes)	0.8125	0.4375	42.097398	176.654050	6059.9463
GA (Deb and Gene)	0.9375	0.5000	48.329000	112.679000	6410.3811
ES (Montes and Coello)	0.8125	0.4375	42.098087	176.640518	6059.7456
DE (Huang et al.)	0.8125	0.4375	42.098411	176.637690	6059.7340
ACO (Kaveh and Talataheri)	0.8125	0.4375	42.103624	176.572656	6059.0888
Lagrangian Multiplier (Kannan)	1.1250	0.6250	58.291000	43.6900000	7198.0428
Branch-bound (Sandgren)	1.1250	0.6250	47.700000	117.701000	8129.1036

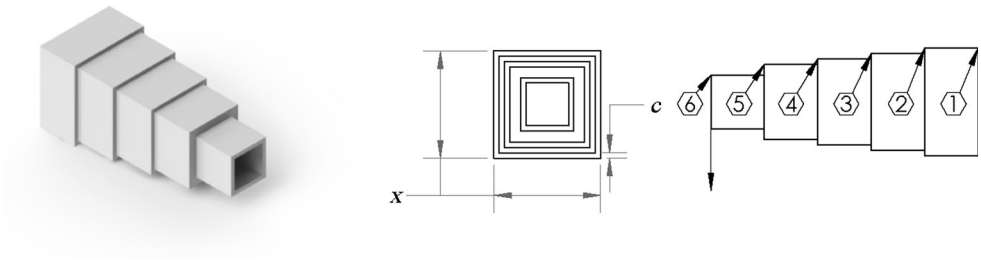

Fig. 10 Cantilever beam design problem

Table 12 Comparison results for cantilever design problem

Algorithm	Optimal values for variables					Optimum weight
	x_1	x_2	x_3	x_4	x_5	
MVO	6.023940221548	5.30601123355	4.4950113234	3.4960223242	2.15272617	1.3399595
MMA [55]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA_I [55]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA_II [55]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
CS [42]	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
SOS [56]	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996

provides similar results compared to other algorithms. The obtained number of gears of MVO and MBA [39] is similar. These results prove that the proposed algorithm is also able to provide very competitive results on the problem with discrete parameters.

4.5.3 Three-bar truss design using MVO

Generally speaking, truss design problems are very popular in the literature of meta-heuristics [39, 42]. The objective is to design a truss with the minimum weight that does not violate the constraints. The most important issue in designing truss is constraints that include stress, deflection, and buckling constraints. Figure 8 shows the structural parameters of this problem.

As can be seen in the full details of this problem in the "Appendix 2", the objective function is very simple, but it is subject to several challenging constraints. We employ this test function as a specific test bed with emphasize on constraints. The results of the MVO algorithm are compared with other algorithms in Table 10.

The results of this table show that MVO provides very close results to DEDS and PSO-DE. This evidences that the proposed algorithm is able to effectively optimize challenging constrained problems as well.

4.5.4 Pressure vessel design using MVO

The pressure vessel design problem is another popular engineering test problem in the literature of meta-heuristics

[42, 47]. This objective is the minimization of the fabrication cost of a vessel. As shown in Fig. 9, one side of the vessel is flat, whereas the other side has hemispherical shape. The structural parameters to be optimizers are (see Fig. 9 and "Appendix 2") thickness of the shell (T_s), thickness of the head (T_h), inner radius (R), and length of the cylindrical section without considering the head (L).

We found an optimal design for this problem using MVO and report the relevant results in Table 11. We compare the results to PSO [48], GA [49–51], ES [52], DE [53], and ACO [54]. Mathematical methods used are augmented Lagrangian multiplier [40] and branch-and-bound [37].

The results of this table show that the proposed MVO algorithm outperforms all other algorithms.

4.5.5 Cantilever beam design using MVO

The structural optimization problem is to minimize the weight of a cantilever beam with hollow square blocks. There are five squares of which the first block is fixed and the fifth one burdens a vertical load. As may be seen in Fig. 10 and "Appendix 2", five parameters define the shape of cross section of the cubes.

We also solve this problem by the MVO algorithm and present the results in the Table 12. The results of this algorithm for this problem are consistent to those of other real problems, in which the MVO algorithm outperforms all of other algorithms.

To sum up, the results of the employed five real engineering design problems prove that the MVO algorithm can be very effective in optimizing real problems. In addition, the performance of the proposed algorithm is verified in solving constrained problems. The reason of the superior results of MVO on constrained problems is that this algorithm efficiently balances exploration and exploitation. For one, the concepts of white/black holes promote exploration, which can cover both feasible and infeasible regions of the search space. For another, increasing the existence of wormholes emphasizes exploitation and local search around the best solution obtained so far.

5 Conclusion

This work proposed a novel optimization technique inspired from multi-verse theory in physics. Three conceptual models of white, black, and wormholes were proposed to design the MVO algorithm. A comprehensive comparative study was conducted on 19 test functions and five real engineering problems to investigate the efficiency of the proposed algorithm in solving optimization problems. The exploration, exploitation, and convergence of MVO were benchmarked and discussed on the test functions. The findings demonstrated the value of the proposed algorithm:

- White holes are more possible to be created in the universes with high inflation rates, so they send objects to other universes and assist them to improve their inflation rates.
- Black holes are more likely to be appeared in the universes with low inflation rates, so they have higher probability to receive objects from other universes. This again increases the chance of improving inflation rate for the universes with low inflation rates.
- White/black hole tunnels tend to transport objects from universes with high inflation rates to those with low inflation rates, so the overall/average inflation rate of all universes is improved over the course of iterations.
- Wormholes tend to appear in any universe randomly regardless of the inflation rate, so the diversity of universe is maintained over the iterations.
- While/black hole tunnels require universes to abruptly change, causing exploration of the search space.
- Abrupt changes also assist resolving local optima stagnations.
- Wormholes randomly re-span some of the variables of universes around the best solution obtained so far over the course of iterations, so this guarantees exploitation around the most promising region of the search space.
- Adaptive WEP values smoothly increase the existence probability of wormholes in universes. Therefore, exploitation is emphasized during optimization process.
- Adaptive TDR values decrease the travelling distance of variables around the best universe, a mechanism that increases the accuracy of local search over the iterations.
- The convergence of the proposed algorithm is guaranteed by emphasizing exploitation/local search proportional to the number of iterations.

For future work, we are planning to design the multi-objective and binary version of the MVO algorithm.

Appendix 1

```
Create random universes (U)
Initialize WER, TDR, and Best_universe
SU=Sorted universes
NI=Normalize the inflation rate (fitnesses) of the universes
while the end criterion is not satisfied
    Evaluate the fitness of all universes
    for each universe indexed by i
        Update WEP and TDR
        Black_hole_index=i;
        for each object indexed by j
            r1=random([0,1]);
            if r1<NI(U_i)
                White_hole_index= RouletteWheelSelection(-NI);
                U(Black_hole_index,j)=SU(White_hole_index,j);

            end if
            r2=random([0,1]);
            if r2<Wormhole_existance_probability
                r3= random([0,1]);
                r4= random([0,1]);
                if r3<0.5
                    U(i,j)=Best_universe(j) + Travelling_distance_rate * ((ub(j)
                    - lb(j)) * r4 + lb(j));
                else
                    U(i,j)=Best_universe(j) - Travelling_distance_rate * ((ub(j)
                    - lb(j)) * r4 + lb(j));
                end if
            end if
        end for
    end for
end while
```

Appendix 2

Welded beam design problem

$$
\begin{aligned}
\text{Consider} \quad & \vec{x} = [x_1\, x_2\, x_3\, x_4] = [h\, l\, t\, b],\\
\text{Minimize} \quad & f(\vec{x}) = 1.10471 x_1^2 x_2 \\
& \quad + 0.04811 x_3 x_4 (14.0 + x_2),\\
\text{Subject to} \quad & g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \le 0,\\
& g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\max} \le 0,\\
& g_3(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \le 0\\
& g_4(\vec{x}) = x_1 - x_4 \le 0,\\
& g_5(\vec{x}) = P - P_c(\vec{x}) \le 0,\\
& g_6(\vec{x}) = 0.125 - x_1 \le 0\\
& g_7(\vec{x}) = 1.10471 x_1^2 \\
& \quad + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \le 0\\
\text{Variable range} \quad & 0.1 \le x_1 \le 2,\\
& 0.1 \le x_2 \le 10,\\
& 0.1 \le x_3 \le 10,\\
& 0.1 \le x_4 \le 2\\
\text{where} \quad & \tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},\\
& \tau' = \frac{P}{\sqrt{2} x_1 x_2}, \quad \tau'' = \frac{\mathrm{MR}}{J},\\
& M = P\left(L + \frac{x_2}{2}\right),\\
& R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},\\
& J = 2\left\{\sqrt{2} x_1 x_2 \left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\},\\
& \sigma(\vec{x}) = \frac{6\mathrm{PL}}{x_4 x_3^2}, \quad \delta(\vec{x}) = \frac{6\mathrm{PL}^3}{E x_3^2 x_4}\\
& P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),\\
& P = 6000 \text{ lb},\ L = 14 \text{ in.},\ \delta_{\max} = 0.25 \text{ in.},\\
& \quad E = 30 \times 1^6 \text{ psi}, \quad G = 12 \times 10^6 \text{ psi},\\
& \tau_{\max} = 13600 \text{ psi},\ \ \sigma_{\max} = 30000 \text{ psi}
\end{aligned}
$$

Gear train design problem

$$
\begin{aligned}
\text{Consider} \quad & \vec{x} = [x_1\, x_2\, x_3\, x_4] = [n_A\, n_B\, n_C\, n_D],\\
\text{Minimize} \quad & f(\vec{x}) = \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4}\right)^2,\\
\text{Variable range} \quad & 12 \le x_1, x_2, x_3, x_4 \le 60,
\end{aligned}
$$

Three-bar truss design problem

$$
\begin{aligned}
\text{Consider} \quad & \vec{x} = [x_1\, x_2] = [A_1\, A_2],\\
\text{Minimize} \quad & f(\vec{x}) = \left(2\sqrt{2} x_1 + x_2\right) * l,\\
\text{Subject to} \quad & g_1(\vec{x}) = \frac{\sqrt{2} x_1 + x_2}{\sqrt{2} x_1^2 + 2 x_1 x_2} P - \sigma \le 0,\\
& g_2(\vec{x}) = \frac{x_2}{\sqrt{2} x_1^2 + 2 x_1 x_2} P - \sigma \le 0,\\
& g_3(\vec{x}) = \frac{1}{\sqrt{2} x_2 + x_1} P - \sigma \le 0,\\
\text{Variable range} \quad & 0 \le x_1, x_2 \le 1,\\
\text{where} \quad & l = 100 \text{ cm}, P = 2 \text{ KN/cm}^2,\\
& \sigma = 2 \text{ KN/cm}^2
\end{aligned}
$$

Pressure vessel design problem

$$
\begin{aligned}
\text{Consider} \quad & \vec{x} = [x_1\, x_2\, x_3\, x_4] = [T_s\, T_h\, R\, L],\\
\text{Minimize} \quad & f(\vec{x}) = 0.6224 x_1 x_3 x_4 + 1.7781 x_2 x_3^2 \\
& \quad + 3.1661 x_1^2 x_4 + 19.84 x_1^2 x_3,\\
\text{Subject to} \quad & g_1(\vec{x}) = -x_1 + 0.0193 x_3 \le 0,\\
& g_2(\vec{x}) = -x_3 + 0.00954 x_3 \le 0,\\
& g_3(\vec{x}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 \\
& \quad + 1296000 \le 0,\\
& g_4(\vec{x}) = x_4 - 240 \le 0,\\
\text{Variable range} \quad & 0 \le x_1 \le 99,\\
& 0 \le x_2 \le 99,\\
& 10 \le x_3 \le 200,\\
& 10 \le x_4 \le 200
\end{aligned}
$$

Cantilever beam design

$$
\begin{aligned}
\text{Consider} \quad & \vec{x} = [x_1 x_2 x_3 x_4 x_5]\\
\text{Minimize} \quad & f(\vec{x}) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5),\\
\text{Subject to} \quad & g(\vec{x}) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \le 0,\\
\text{Variable range} \quad & 0.01 \le x_1, x_2, x_3, x_4, x_5 \le 100,
\end{aligned}
$$

References

1. John H (1992) Holland, adaptation in natural and artificial systems. MIT Press, Cambridge
2. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, pp 1942–1948
3. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248
4. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1:67–82
5. Kirkpatrick S (1984) Optimization by simulated annealing: quantitative studies. J Stat Phys 34:975–986
6. Hoos HH, Stützle T (2004) Stochastic local search: foundations and applications. Elsevier, Amsterdam
7. Johnson DS, Papadimitriou CH, Yannakakis M (1988) How easy is local search? J Comput Syst Sci 37:79–100
8. Mitchell M, Holland JH, Forrest S (1993) When will a genetic algorithm outperform hill climbing? In: NIPS, pp 51–58
9. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1:28–39
10. Dorigo M, Stützle T (2003) The ant colony optimization metaheuristic: algorithms, applications, and advances. In: Glover F, Kochenberger GA (eds) Handbook of metaheuristics. International series in operations research & management science, vol 57. Springer, US, pp 250–285
11. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39:459–471
12. Khoury J, Ovrut BA, Seiberg N, Steinhardt PJ, Turok N (2002) From big crunch to big bang. Phys Rev D 65:086007
13. Tegmark M (2004) Parallel universes. In: Barrow JD, Davies PCW, Harper CL Jr (eds) Science and ultimate reality: Quantum theory, cosmology, and complexity. Cambridge University Press, pp 459–491
14. Eardley DM (1974) Death of white holes in the early Universe. Phys Rev Lett 33:442
15. Steinhardt PJ, Turok N (2002) A cyclic model of the universe. Science 296:1436–1439
16. Davies PC (1978) Thermodynamics of black holes. Rep Prog Phys 41:1313
17. Morris MS, Thorne KS (1988) Wormholes in spacetime and their use for interstellar travel: a tool for teaching general relativity. Am J Phys 56:395–412
18. Guth AH (2007) Eternal inflation and its implications. J Phys A Math Theor 40:6811
19. Steinhardt PJ, Turok N (2005) The cyclic model simplified. New Astron Rev 49:43–57
20. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3:82–102
21. Digalakis J, Margaritis K (2001) On benchmarking functions for genetic algorithms. Int J Comput Math 77:481–506
22. Molga M, Smutnicki C (2005) Test functions for optimization needs. http://www.robertmarks.org/Classes/ENGR5358/Papers/functions.pdf
23. Yang X-S (2010) Test problems in optimization. arXiv preprint: arXiv:1008.0549
24. Liang J, Suganthan P, Deb K (2005) Novel composition test functions for numerical global optimization. In: Proceedings 2005 IEEE swarm intelligence symposium, 2005. SIS 2005, pp 68–75
25. Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. In: KanGAL report, vol 2005
26. 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 1:3–18
27. Mirjalili S, Lewis A (2013) S-shaped versus V-shaped transfer functions for binary particle swarm optimization. Swarm Evol Comput 9:1–14
28. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey Wolf Optimizer. Adv Eng Softw 69:46–61
29. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98. doi:10.1016/j.advengsoft.2015.01.010
30. van den Bergh F, Engelbrecht A (2006) A study of particle swarm optimization particle trajectories. Inf Sci 176:937–971
31. Carlos A, Coello C (2000) Constraint-handling using an evolutionary multiobjective optimization technique. Civil Eng Syst 17:319–346
32. Deb K (1991) Optimal design of a welded beam via genetic algorithms. AIAA J 29:2013–2015
33. Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 186:311–338
34. Krohling RA, dos Santos Coelho L (2006) Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems. IEEE Trans Syst Man Cybern Part B Cybern 36:1407–1416
35. Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194:3902–3933
36. Ragsdell K, Phillips D (1976) Optimal design of a class of welded structures using geometric programming. ASME J Eng Ind 98:1021–1025
37. Sandgren E (1990) Nonlinear integer and discrete programming in mechanical design optimization. J Mech Des 112(2):223–229. doi:10.1115/1.2912596
38. Gandomi AH (2014) Interior search algorithm (ISA): a novel approach for global optimization. ISA Trans 53(4):1168–1183. doi:10.1016/j.isatra.2014.03.018
39. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13:2592–2612
40. Kannan B, Kramer SN (1994) An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 116:405–411
41. Deb K, Goyal M (1996) A combined genetic adaptive search (GeneAS) for engineering design. Comput Sci Inform 26:30–45
42. Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29:17–35
43. Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. Inf Sci 178:3043–3074
44. Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Appl Soft Comput 10:629–640
45. Ray T, Saini P (2001) Engineering design optimization using a swarm with an intelligent information sharing among individuals. Eng Optim 33:735–748
46. Tsai J-F (2005) Global optimization of nonlinear fractional programming problems in engineering design. Eng Optim 37:399–409
47. Mahdavi M, Fesanghary M, Damangir E (2007) An improved harmony search algorithm for solving optimization problems. Appl Math Comput 188:1567–1579

48. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20:89–99
49. Coello Coello CA (2000) Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 41:113–127
50. Coello Coello CA, Mezura Montes E (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inform 16:193–203
51. Deb K, Gene AS (1997) A robust optimal design technique for mechanical component design. Presented at the D. Dasgupta, Z. Michalewicz (eds) Evolutionary algorithms in engineering applications, Berlin
52. Mezura-Montes E, Coello CAC (2008) An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 37:443–473
53. Li L, Huang Z, Liu F, Wu Q (2007) A heuristic particle swarm optimizer for optimization of pin connected structures. Comput Struct 85:340–349
54. Kaveh A, Talatahari S (2010) An improved ant colony optimization for constrained engineering design problems. Eng Comput Int J Comput Aided Eng 27:155–182
55. Chickermane H, Gea HC (1996) Structural optimization using a new local approximation method. Int J Numer Methods Eng 39(5):829–846
56. Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112