JOURNAL OF EXPERIMENTAL & THEORETICAL ARTIFICIAL INTELLIGENCE, 2018
https://doi.org/10.1080/0952813X.2018.1430858

Check for updates

A new chaotic multi-verse optimization algorithm for solving engineering optimization problems

Gehad Ismail Sayed[a,c], Ashraf Darwish[b,c] and Aboul Ella Hassanien[a,c]

[a]Faculty of Computers and Information, Cairo University, Cairo, Egypt; [b]Faculty of Science, Helwan University, Cairo, Egypt; [c]Scientific Research Group in Egypt, Egypt

ABSTRACT

Multi-verse optimization algorithm (MVO) is one of the recent meta-heuristic optimization algorithms. The main inspiration of this algorithm came from multi-verse theory in physics. However, MVO like most optimization algorithms suffers from low convergence rate and entrapment in local optima. In this paper, a new chaotic multi-verse optimization algorithm (CMVO) is proposed to overcome these problems. The proposed CMVO is applied on 13 benchmark functions and 7 well-known design problems in the engineering and mechanical field; namely, three-bar trust, speed reduce design, pressure vessel problem, spring design, welded beam, rolling element-bearing and multiple disc clutch brake. In the current study, a modified feasible-based mechanism is employed to handle constraints. In this mechanism, four rules were used to handle the specific constraint problem through maintaining a balance between feasible and infeasible solutions. Moreover, 10 well-known chaotic maps are used to improve the performance of MVO. The experimental results showed that CMVO outperforms other meta-heuristic optimization algorithms on most of the optimization problems. Also, the results reveal that sine chaotic map is the most appropriate map to significantly boost MVO's performance.

ARTICLE HISTORY
Received 5 December 2016
Accepted 14 January 2018

KEYWORDS
Nature-inspired algorithm; Multi-verse optimization algorithm; Chaotic theory; Constrained optimization problems

1. Introduction

In the last decades, evolutionary algorithms (EAs) including optimization problems, have received a great attraction for different applications. EAs inspired by creating a relationship between the power of natural evolutionary mechanisms and the nature of solving the problem. These algorithms mimic social behaviour or natural phenomena to obtain better solutions. The most popular EAs algorithms are particle swarm optimization (PSO) (Eberhart & Kennedy, 1995, October 4–6) and genetic algorithm (GA) (Holland, 1992). Meta-heuristic optimization algorithms have proven their efficiency for solving optimization problems using iterations and stochastic behaviour. Authors in Sayed and Hassanien (2017) used a modified version of moth-flame optimization to solve the feature selection optimization problem. Also, they applied their algorithm to breast cancer histology images. Ghasemi, Ghavidel, Aghaei, Gitizadeh and Falah (2014) presents a Chaotic invasive weed optimization algorithm (CIWO). The proposed CIWO is evaluated for optimal settings of optimal power flow (OPF) control variables of the OPF problem with non-smooth and non-convex generator fuel cost curves. The experimental results on the standard IEEE 30-bus test system with different objective functions show promising results of CIWO algorithm. In Sayed (2016), authors used grey wolf optimisation algorithm for automatic liver segmentation from abdominal CT liver images. The experimental results show the efficiency of their proposed CAD system. It obtains overall accuracy 96% for healthy liver

CONTACT Gehad Ismail Sayed gehad.ismail@egyptscience.net, GehadIsmail_FCI@yahoo.com

extraction and 97% for liver disease classification. Another application proposed in Sayed, Darwish, Hassanien, and Pan (2017b). In this paper, whale optimization algorithm is applied to automatic breast cancer diagnosis. The common idea behind all these algorithms is that given a population of objects or individuals, the environmental pressure causes natural selection, which causes a rise in the fitness of the population. Evolutionary algorithms implement their structures in different ways according to population initialisation, candidate evaluation, termination condition, selecting method, recombination and mutation (Özkaynak, 2015). Despite the different structures of EAs, they usually generate random population. Then, they evaluate it through the iterations. EAs are similar in dividing the search space into two main phases: exploitation and exploration. In many cases, EAs get stuck in local minima. This is due to improper balancing between exploitation and exploration and the stochastic nature of EAs. Several studies are presented in the literature to overcome these problems and to improve the performance of EAs. One of the most common mathematical methods recently applied to boost the performance of EAs is called chaos theory.

Chaos theory studies the system's behaviour, which is highly sensitive to their initial conditions. They can generate a varied range of numbers instead of random numbers. The chaotic system's behaviour seems to be random; so, chaotic systems can be adopted as for the needed randomness by the evolutionary algorithms. Chaos theory has provided effectiveness in different fields of sciences, such as chaos control (Zhang & Zhang, 2008), synchronization (Zhu, Li, & Yu, 2008), optimization research (Zhang, Li, Zhou, & Wei, 2013; Abdullah, Enayatifa, & Lee, 2012) and so on. In recent years, the applications of chaos in various disciplines including optimization research problems have received more attention. Based on chaos theory, chaotic optimization algorithm (COA) (Wang & Liu, 2010) utilises the nature of chaotic sequence, including the quasi-stochastic property and ergodicity. Currently, chaotic systems are an active area of research in the last few years and have been applied in different sciences research and engineering fields (Chaoshun, Xueli, & Ruhai, 2015; Montes & Coello, 2008). Chaotic sequences have been used previously for parameters tuning in meta-heuristic optimization algorithms and to avoid being trapped in local optimums. Some of these algorithms are PSO (Gandomi, Yun, Yang, & Talatahari, 2013), GA (Gharooni-fard, Moein, Deldari, & Morvaridi, 2010), ant and bee colony optimization (Gong & Wang, 2009), crow search algorithm (Sayed, Hassanien, & Azar, 2017c), harmony search (Alatas, 2010) and firefly algorithm (Gandomi, Yang, Talatahari, & Alavi, 2013). Authors in Kaveh, Sheikholeslami, Talatahari, and Keshvari-Ilkhichi (2014) employed chaos theory in two phases, in the first phase they integrate chaotic maps to control the parameter values of PSO and in the second phase, they utilised the local search of PSO to avoid being trapped in local optimums by embedding chaotic maps. Authors in Sayed et al. (2017c) used a chaotic version of crow search algorithm for feature selection problem. The experimental results show that their algorithm obtained very competitive results on 20 benchmark data-sets.

Multi-verse optimizer (MVO) is one of the recent bio-inspired optimization algorithms proposed in Mirjalili, Mirjalili, and Hatamlou (2016). The main inspiration of MVO is taken from multi-verse theory in physics. This inspiration is based on three main concepts in cosmology. These concepts are a wormhole, white hole and black hole. Recently, several applications have been proposed using MVO. The inventor of this algorithm, Mirjalili et al. (2016), showed that MVO obtained very competitive results compared with other meta-heuristic optimization algorithms. Also, authors in Faris et al. (2016) employed the MVO for training the multi-layer perceptions neural network. Their approach was evaluated and benchmarked using nine different bio-medical data-sets selected from the UCI machine learning repository. The obtained results are compared to five meta-heuristic algorithms: Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Differential Evolution (DE) Algorithm and Cuckoo Search (CS). The experimental results showed that MVO is very competitive and overtakes the other training algorithms using the adopted data-sets in terms of convergence speed and improved local optima avoidance. However, the performance of MVO algorithm can be further enhanced in terms of accuracy and convergence speed. Fewer studies have been proposed to enhance the performance of MVO. Hu, Li, Zhou, Zhu, and Xu (2016) proposed Lévy-flight multi-verse optimization (LFMVO) algorithm to improve the performance of MVO. The proposed LFMVO used Lévy-flight strategy in the searching mechanism of MVO. LFMVO is applied to solve 23 benchmark functions and an NP-complete problem of test schedule for the network on chip (NoC). The experimental results show that LFMVO is competitive than MVO in terms of quality and convergence speed. A quantum version of MVO is applied in Sayed, Darwish, and Hassanien (2017a). The results on 50 benchmark global optimization problems show that it's very competitive algorithm compared with other popular and recent meta-heuristic algorithms.

Currently, there are few works for boosting the performance of MVO. This work presents a new method of hybridising the chaos theory with MVO algorithm called chaotic multi-verse optimization algorithm (CMVO). Ten chaotic maps integrated on the searching iterations of MVO to boost the performance of MVO. The experimental results on 7 constrained problems and 13 benchmarks show that CMVO is very competitive over MVO and other meta-heuristic algorithms.

The outline of the paper is as follows. In Section 2, the basics of chaos theory, chaotic maps and MVO algorithm are presented. In Section 3, the proposed CMVO algorithm is proposed in details. Section 4 presents experimental results and analysis. Finally, concluding remarks and future work are presented in section 5.

2. Basics and background

2.1. Multi-verse optimization algorithm

2.1.1. Inspiration

In this section, one of the recent stochastic population-based algorithms is employed, namely MVO, which proposed in Mirjalili et al. (2016). The main inspiration of MVO came from multi-verse in physics theory. The main components of the multi-verse theory are a black hole, wormhole and a white hole. They are mathematically modelled to construct the MVO. The multi-verse theory is recent in physics. In this theory, physicists believe that there is more than one big bang, and each one of them causes a universe' birth. The multi-verse approach stands opposite of a universe. In other words, it refers to the other universe's existence than our universe, where we all are living in according to multi-verse theory. In MVO, there are three main multi-verse's concepts theories. These three concepts are used in the inspiration process of MVO algorithm. They are a black hole, wormhole and white hole. Each concept has its function in the inspiration process. Physicists believe that the big bang is referred to a white hole and they think that it is the main element of a universe's birth. In contrast to white holes, black holes behave completely different from white holes. With their extremely high gravitational force, they can attract everything including light beams. In the inspiration process, every universe cab causes its expansion through space using its inflation rate. Inflation speed of a universe is important in forming stars, white holes, planets, asteroids, physical laws, black holes and the suitability of life. This scenario represents the exact the MVO inspiration. In the next, the mathematical model of MVO is presented.

2.1.2. Mathematical model of MVO

MVO uses the black and white hole concepts for exploring the search spaces, while it uses wormholes for exploiting the search spaces. Like all evolutionary algorithms, it starts the optimization process by initiating a population of solutions and tries to enhance these solutions over a predefined number of iterations. In this algorithm, the individual's improvement in each population can be performed based on one of the theories about the existence of multiple universes. In these theories, each solution of an optimization problem is considered as a universe and each object in the universe is considered as a variable of a given problem. During the optimization process, MVO follows the following steps.

(1) The higher inflation rate is proportional to the higher probability of the white hole.
(2) The higher inflation rate is proportional to the lower probability of the black holes.
(3) Objects in the universe with higher inflation rate move from a white hole to a black hole.
(4) Objects in the universe 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 regardless of the inflation rate.

In MVO, wormhole tunnels are always established between a universe and the best universe formed so far. The formulation of this mechanism is as follows: Mirjalili et al. (2016).

$$Y_i^j = \begin{cases} \begin{cases} Y_j + \mathrm{TDR} \times ((upb_j - lbp_j) \times R_4 + lbp_j) & R3 < 0.5 \\ Y_j - \mathrm{TDR} \times ((upb_j - lbp_j) \times R_4 + lbp_j) & R3 \geq 0.5 \end{cases} & R2 < \mathrm{WEP} \\ Y_i^j & R2 \geq \mathrm{WEP} \end{cases} \tag{1}$$

where Y_j denotes the *jth* object of the best universe, u_i^j denotes the *jth* parameter of the *ith* universe, lbp_j is the lower bound at *jth* iteration while ubp_j is the upper bound at *jth* iteration and R_2, R_3, R_4 are random parameters in range [0, 1], wormhole existence probability (WEP) and travelling distance rate (TDR) are constant parameters, where WEP is linearly increased through the iteration, while TDR is decreased through the iterations. The adaptive formula for both coefficients is as follows:

$$\mathrm{WEP} = \mathrm{WEP}_{\min} + \mathrm{iter} \times \left(\frac{\mathrm{WEP}_{\max} - WEP_{\min}}{\mathrm{Max}_{iter}} \right) \tag{2}$$

where $\mathrm{WEP}_{\min}$ is the minimum value of WEP, $\mathrm{WEP}_{\max}$ is the maximum value of WEP, iter denotes the current iteration and $\max_{\mathrm{iter}}$ indicates the maximum number of iterations.

$$\mathrm{TDR} = 1 - \frac{\mathrm{iter}^{\mathrm{AC}}}{\mathrm{Max}_{\mathrm{iter}}^{\mathrm{AC}}} \tag{3}$$

where *AC* is one divided by the exploitation accuracy over the iterations. The higher value of *AC*, the more accurate exploitation.

At the beginning of the algorithm, the algorithm starts by initiating a set of random universes (solutions in the search space) of the optimization problem. During each generation/iteration, objects with high inflation rates in the universes move to the universes with low inflation rates via black/white holes. For more details about MVO algorithm, the authors refer to Mirjalili et al. (2016).

2.2. Chaotic theory

One of the mathematical approaches that have been employed recently to improve the exploration and exploitation is chaos. Chaos theory is concerned with the study of chaotic dynamical systems that are highly sensitive to initial conditions. Chaos is the phenomenon that occurs in a deterministic nonlinear dynamic system that it is extremely sensitive to the initial condition. It is mathematically defined as a semi-randomness behaviour generated by nonlinear deterministic systems. Therefore, a chaotic movement can travel all states without any repetition within the certain range. Chaotic algorithms have many advantages such as easy implementation and special capacity in order to avoid being trapped in local optima, chaos-based search algorithms have aroused intense interest (Wang, Zheng, & Lin, 2001). In the literature, chaos theory is may be described as the butterfly effect. Therefore, chaotic systems have the properties; namely, sensitivity to primary condition, randomness and deterministic. Using these properties of chaotic systems there are some studies in the literature were proposed for maintaining population diversity and to avoid the local optimum.

In most of the meta-heuristic algorithms with stochastic behaviour, there are randomness components. The randomness can be obtained using some uniform or probability distributions or Gaussian Gandomi and Yangb (2014). Chaos has similar properties of randomness, but with better dynamical and statistical properties (Schuster 1998). Such the dynamical mixing is needed to ensure that the diversity of the generated solutions enough to potentially reach the optimal solution. Benefiting from the ergodicity and stochastically properties of chaos, it has been employed in several optimization problems. Chaos strategy has been applied to the searching process of optimization algorithms to avoid stagnation in a local optimum and to improve the quality of searching for the global optimum.

Chaotic maps have been employed in Yang and Chen (2002) to manipulate the mutation probability in order to increase the exploitation of genetic algorithm (GA). An improved logistic map, namely a double-bottom map, used in Yang, Tsai, Chuang, and Yang (2012) with PSO algorithm for production optimization problems. Eight chaotic maps for parameter adaptation are used in Alatas, Akin, and Ozer (2009). The experimental results showed that the proposed algorithm improved the solution quality and sometimes improved the global search capacity. The biogeography-based optimization (BBO) is an optimization algorithm which, inspired by biogeography and has been applied in the literature in some application such as ecosystems. Chaotic maps are used in Saremi, Mirjalili, and Lewis (2014) and showed that the chaotic maps can improve the performance of BBO. Ten chaotic maps are integrated with BBO algorithm with the objective to improve the performance of chaos theory in terms of both exploration and exploitation of the BBO algorithm.

A chaotic map with n dimension is a discrete-time dynamical system that can be expressed as in Equation 4 Chaoshun et al. (2015). By defining the initial state of $cp_q^{(0)}$, a chaotic sequence can be evaluated by running the system function, where chaotic sequence can be defined in the form of$\{cp_q^{(k)}, k = 0, 1, 2, \}$. In this paper, 10 distinguished non-invertible 1-D maps are Abdullah et al. (2012) employed in the searching mechanism of MVO as shown in Table 1. In this table, q denotes the index of the chaotic sequence, and p_q is the $q - th$ number in the chaotic sequence. The initial point p_0 is set to 0.7 for all chaos maps. As the initial values for chaotic map may have a great influence of the fluctuation pattern of chaotic maps. We set the initial values as in Saremi et al. (2014).

$$cp_q^{(k+1)} = f(cp_q^{(k)}),\ i = 1, 2, 3, \ldots, n \tag{4}$$

3. The proposed chaotic multi-verse optimization algorithm

As it presented in Equation (2), the main parameter which affecting the balancing between exploration and exploitation is *TDR*. The value of *TDR* in standard MVO is set from 0.6 to 0. This value effects on the variables' travelling distance around the best universe. It is highly influenced by the accuracy of local search (exploitation) through the iterations. Also, this value is vitally important in the updating the universe's position in MVO (Mirjalili et al., 2016). In this paper, the *AC* parameter which used for calculating the *TDR* is replaced with chaotic variables. Such chaos combined with MVO can be defined as the CMVO. Ten different chaotic maps are employed for tuning *AC* parameter. The mathematical formula of these maps is defined in Section 2.2. These maps can significantly improve the performance and convergence rate to the optimal solution of MVO as will be demonstrated later in the following section.

The MVO algorithm combined with chaotic sequences is described as follows:

$$TDR_j = 1 - \frac{iter^{C_j}}{Max_{iter}^{C_j}} \tag{5}$$

where C_j is the obtained value of chaotic map of $j - th$ iteration.

In this paper, the modified version of MVO-based chaos theory is applied to solve benchmark constrained optimization problems. In order to convert the original version of MVO algorithm, which known as the continuous version to solve constrained problems, the modified feasible-based mechanism is employed. This mechanism is based on using the following four heuristic rules (Deb, 2000). This mechanism is based on a simple comparison method, where the objective function is separated from constraint violation. Thus, feasible and infeasible individuals are evaluated with different criteria. These rules are used to maintain the balance between feasible and infeasible individuals and to select the best solution. These rules are:

(1) Rule 1: Any feasible solution is preferred to any infeasible solution.
(2) Rule 2: Any infeasible solution has a slight violation of the constraints which starting from 0.01 at the first iteration and end with 0.001 at last iteration is considered as a feasible solution.
(3) Rule 3: Among two feasible solutions, the one having the better objective function value is preferred.
(4) Rule 4: Among two infeasible solutions, the one having the smaller sum of constraint violation is preferred.

According to the rule number three, the search space is guided to feasible solution with good solutions. The search space based on the rule number one and four is oriented to feasible regions instead of infeasible regions. Almost all of global optimization problems, the global minimum locate on or near to the boundary of feasible space. However, by employing the rule number two, the universes can reach to the optimal solution with higher probability (Zahara & Kao, 2009a). The detailed description is defined as follows:

(1) **Parameters initialization**
At the beginning, CMVO starts with setting the adjustable parameters including defining the search space boundary and number of decision variables and randomly initialised universes positions within the defined boundary. Each position represents a solution in the search space. The initial parameters settings for WEP_{max} is 1, $WEP_{\min}$ is 0.2 with the maximum number of iterations is 1000 and the number of search agents is 50.
(2) **Fitness function**
At each iteration, each search agent position is evaluated using specified fitness function $f(\vec{z})$. The optimal universe position is the one which satisfies the four constrained rules with better fitness score. Also, the best universe position with the best inflation rates moves towards the other universes using black/white holes. Moreover, each universe faces random teleportation using wormholes towards the best universe.
(3) **Positions updating**
The updating universe positions of CMVO defined in Equations (5) and (1).
(4) **Termination criteria**
The optimization process terminates when it reaches the maximum number of iterations or when the optimal solution is found. In our case, we used the maximum number of iterations as the termination criteria.

The pseudo-code of CMVO and its corresponding flowchart are shown in Algorithm **??** and Figure 1, respectively. In addition, the source code of standard MVO, the authors refer to http://www.mathworks.com/matlabcentral/fileexchange/50112-multi-verse-optimizer--mvo.

4. Experimental results and discussion

In this section, several experiments on different versions of CMVO were conducted where the optimal chaotic map is determined. In addition, the optimal chaotic map with CMVO is chosen for further evaluation with other meta-heuristic optimization algorithms. The performance of the proposed CMVO is evaluated for solving 7 constrained and 13 unconstrained optimization problems. In this paper, different evaluation criteria are employed. These criteria are divided to statistical measurements, including mean, worst and best fitness value, standard deviation (SD), number of function evaluations (NFEs) and computational time in seconds (Avg. Time). The Avg. Time measurement is calculated by taking the average processing time for the independent runs. In order to estimate NFEs, we first assigned a large limit for the number of function evaluations, which is calculated by multiplying the total number of iterations by the population size. Then, the algorithm stops running (termination criterion) when there is no progress after 100 iterations. Finally, the final obtained NFEs is reported. In this paper, we took the average of NFEs for the total number of independent runs.

Table 1. The 10 adapted chaotic maps.

No.	Chaotic map	Mathematical equation	Range
CMVO1	Chebyshev (Wang & Liu, 2010)	$p_{q+1} = cos(q\cos^{-1}(p_q))$	$(-1, 1)$
CMVO2	Circle (Zheng, 1994)	$p_{q+1} = mod(p_q + d - (\frac{c}{2\pi})sin(2\pi p_q), 1)$, $c = 0.5$ and $d = 0.2$	(0,1)
CMVO3	Guass/mouse (Peitgen, Jurgens, & Saupe, 1992)	$p_{q+1} = \begin{cases} 1, & p_q = 0 \\ \frac{1}{mod(p_q,1)}, & \text{otherwise} \end{cases}$	(0,1)
CMVO4	Iterative (May, 1976)	$p_{q+1} = \sin(\frac{c\pi}{p_q})$, $c = 0.7$	$(-1, 1)$
CMVO5	Logistic (May, 1976)	$p_{q+1} = cp_q(1 - p_q)$, a=4	(0,1)
CMVO6	Piecewise (Saremi et al., 2014)	$p_{q+1} = \begin{cases} \frac{p_q}{l}, & 0 \le p_q < l \\ \frac{p_q - l}{0.5 - l}, & l \le p_q < 0.5 \\ \frac{1 - l - p_q}{0.5 - l}, & 0.5 \le p_q < 1 - l \\ \frac{1 - p_q}{l}, & 1 - l \le p_q < 1 \end{cases}, l = 0.4$	(0,1)
CMVO7	Sine (Özkaynak, 2015)	$p_{q+1} = \frac{c}{4}\sin(\pi p_q), c = 4$	(0,1)
CMVO8	Singer (Saremi et al., 2014)	$p_{q+1} = \mu(7.86p_q - 23.31p_q^2 + 28.75p_q^3 - 13.302875p_q^4)$, $\mu = 1.07$	(0,1)
CMVO9	Sinusoidal May (1976)	$p_{q+1} = cp_q^2 sin(\pi p_q)$, $c = 2.3$	(0,1)
CMVO10	Tent (Peitgen et al., 1992)	$p_{q+1} = \begin{cases} \frac{p_q}{0.7}, & p_q < 0.7 \\ \frac{10}{3}(1 - p_q), & p_q \ge 0.7 \end{cases}$	(0,1)

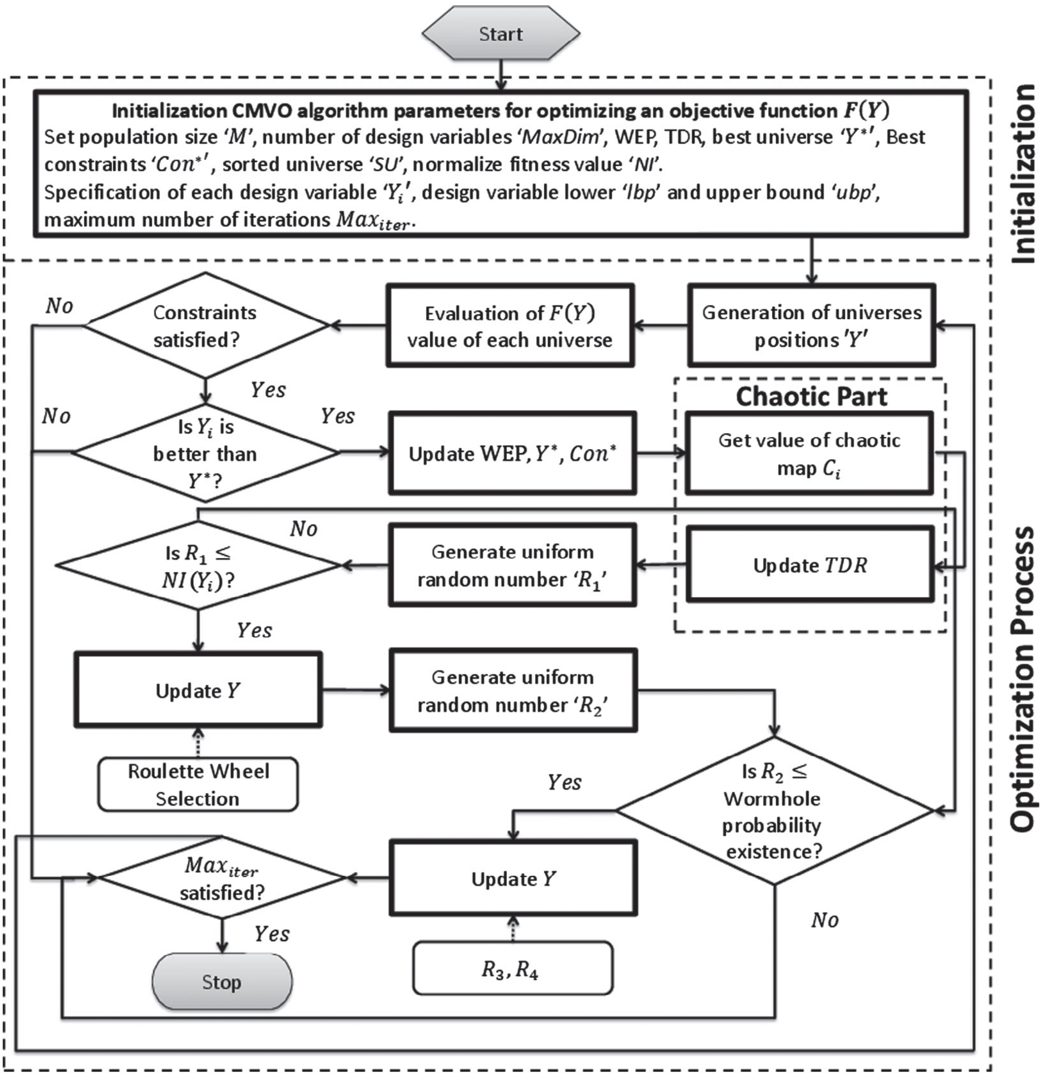

Figure 1. CMVO algorithmbased engineering optimization design procedure.

Two main experiments were conducted to evaluate proposed CMVO. The aim of the first experiment is to determine the optimal chaotic map using seven engineering and mechanical problems and to evaluate the performance of the selected chaotic map with CMVO with other well-known optimization algorithms. The second experiment aims to evaluate the performance of CMVO with a selected chaotic map on 13 benchmark functions with different characteristics. All these experiments are performed on the same PC with Core i3 and RAM 2 GB on OS Windows 7. Also, all the obtained results from the proposed CMVO were executed for 30 independent runs. Moreover, for the maximization problems, they were transformed into minimization ones using $-f(\vec{z})$ and all the equality constraints were transformed into inequality ones using $|h_z| - \varepsilon \leq 0$ where $\varepsilon = 2.2E - 16$.

Table 2. Comparison of statistical results obtained using MVO and different version of CMVO for three-bar truss design problem.

	Worst	Mean	Best	SD	Avg time	NFEs
MVO	263.897613	263.896671	263.895953	6.34E−04	4.643413	46854
CMVO1	263.895908	263.895857	263.895844	2.83E−05	2.357987	26380
CMVO2	263.896015	263.895924	263.895850	6.62E−05	2.417036	25735
CMVO3	263.895004	263.914729	263.895847	4.22E−02	4.592660	32996
CMVO4	263.895959	263.895868	263.895843	5.10E−05	2.460820	24152
CMVO5	263.895846	263.895845	263.895843	1.28E−06	3.601140	26933
CMVO6	263.895997	263.895833	263.895852	5.65E-05	2.815927	25210
CMVO7	263.895848	263.895846	263.895843	2.29E−06	2.198608	27154
CMVO8	263.895958	263.895819	263.895856	2.57E−05	3.791624	33521
CMVO9	263.896460	263.896008	263.895850	2.54E−04	3.057749	30444
CMVO10	263.896039	263.895939	263.895851	5.83E−05	2.550444	25810

4.1. The performance of CMVO for engineering and mechanical design optimization problems experiment

The main objectives of this experiment are: (1) to evaluate the performance of embedding chaotic maps in searching iteration of MVO and (2) to determine the optimal chaotic map for the adopted benchmark engineering and mechanical design optimization problems, (3) to compare the best obtained results from CMVO with an optimal chaotic map with other meta-heuristic algorithms. In all the experiments, the top results obtained by algorithms for each optimization problem are underlined. For constrained optimization problems, the best fitness value is select as the corresponding to the best algorithm. In addition, for global optimization problem, the best values for each adopted measurement are underlined as well.

4.1.1. MVO vs. different version of CMVO experiment

The detailed description of the adopted engineering and mechanical design optimization problems are presented in the Appendix 1. Table 2 compares the obtained results of MVO with different versions of CMVO for solving the three-bar truss design problem in terms of statistical measurements, average time (Avg. Time) and the number of fitness evaluation. As it can be observed from this table, CMVO with iterative, logistic and sine chaotic maps are obtained the best scores. In addition, it can be observed that CMVO with the logistic and sine chaotic map obtained the best mean fitness value and standard deviation. This can prove the high performance and great stability of these two adopted chaotic maps. Moreover, it can be observed that CMVO with iterative chaotic map reaches to the optimal solution with a small number of function evaluations. In addition, it can be observed that generation of random parameters for the original MVO took much time compared with replacing these parameters with calculated chaos sequence. All the proposed CMVO algorithms took less time compared with original MVO. So, the proposed CMVO, especially with logistic, sine and iterative chaotic maps, can be effectively and efficiently design the three-bar truss design problem.

The statistical results of speed reducer design problem of MVO were compared with the proposed CMVO algorithms and are given in Table 3. As it can be observed from this table, CMVO again with sine and logistic maps reached the optimal solution compared with the original MVO and the rest of other versions of CMVO. In addition, it can be observed that all the proposed CMVO algorithms require less number of function evaluations to reach the optimal solution. Moreover, it can be observed that these two maps have higher stability with minimum processing time, as they obtained the lowest standard deviation. In addition, CMVO with Gauss chaotic map took the function evaluations to solve this problem, as it took only 37,638 function evaluations.

Table 4 compares the statistical results of MVO algorithm and CMVO algorithms for solving pressure vessel problem. As it can be seen from this table, CMVO with the logistic chaotic map obtained the best solution, while CMVO with sine and singer chaotic maps are in second place. CMVO with Chebyshev, piecewise and tent chaotic maps couldn't improve the performance of MVO, as their best fitness values

Table 3. Comparison of statistical results obtained using MVO and different version of CMVO for speed reducer design problem.

	Worst	Mean	Best	SD	Avg Time	NFEs
MVO	2996.219546	2995.123587	2994.471226	2.55	5.954004	50000
CMVO1	2994.472125	2994.471651	2994.471346	3.23E-04	5.235454	47807
CMVO2	2996.227884	2994.946402	2994.472108	7.33E-01	5.765875	49001
CMVO3	2994.472522	2994.471800	2994.471226	5.29E-04	5.005046	37700
CMVO4	2994.489360	2994.477091	2994.471945	7.15E-03	5.189546	47921
CMVO5	2994.471634	2994.471254	2994.471068	2.30E-04	5.087251	46606
CMVO6	2997.888202	2995.166219	2994.472617	1.52	5.235545	49054
CMVO7	2994.471567	2994.471358	2994.471068	2.30E-04	5.245437	47083
CMVO8	2996.808129	2995.449859	2994.488117	1.20	5.422354	49122
CMVO9	2996.512844	2996.541878	2994.475755	3.13E+01	5.235178	50000
CMVO10	2996.364095	2995.218825	2994.473150	1.02	5.325459	49045

Table 4. Comparison of statistical results obtained using MVO and different version of CMVO for pressure vessel problem.

	Worst	Mean	Best	SD	Avg time	NFEs
MVO	7201.8922	6326.3745	6059.9528	6.12E+02	4.854449	50000
CMVO1	6591.2105	6422.5200	6060.8123	3.13E+02	4.545441	50000
CMVO2	6711.0477	6168.3766	6059.7554	4.35E+02	4.535587	50000
CMVO3	7131.9759	6598.5650	6059.7454	7.01E+02	4.558754	50000
CMVO4	6947.7612	6638.2945	6059.8944	5.32E+02	4.612780	50000
CMVO5	6415.6213	6492.1968	6059.7207	3.40E+02	4.544986	50,000
CMVO6	6829.4803	6328.2817	6059.9540	6.01E+02	4.566210	50000
CMVO7	6547.6712	6281.6724	6059.7208	5.51E+02	4.545445	50000
CMVO8	7123.2319	6326.0121	6059.7208	7.84E+02	4.877540	50000
CMVO9	7214.1254	6487.5723	6059.8359	6.75E+02	4.505455	50000
CMVO10	7204.5129	6353.1765	6060.1134	6.41E+02	4.428544	50000

Table 5. Comparison of statistical results obtained using MVO and different version of CMVO for tension/compression spring design problem.

	Worst	Mean	Best	SD	Avg time	NFEs
MVO	0.0227750	0.0167897	0.0126982	3.82E−03	4.6949491	49901
CMVO1	0.0171921	0.0146043	0.0126651	2.16E−03	4.6230239	49990
CMVO2	0.0176200	0.0143211	0.0127034	1.95E−03	4.6232871	49991
CMVO3	0.0177232	0.0164397	0.0126660	1.77E−03	4.6344961	50000
CMVO4	0.0170181	0.0144848	0.0126652	1.56E−03	4.7609962	49994
CMVO5	0.0176311	0.0151012	0.0130458	2.13E−03	4.6877411	49985
CMVO6	0.0173048	0.0143315	0.0126650	1.72E−03	4.6290109	49859
CMVO7	0.0154764	0.0139167	0.0126650	1.31E−03	4.7313686	49680
CMVO8	0.0162262	0.0132037	0.0126651	1.33E−03	4.6102704	49936
CMVO9	0.0192389	0.0146430	0.0126653	2.64E−03	4.6424574	49905
CMVO10	0.0176647	0.0150193	0.0126724	1.64E−03	4.6324804	50000

are worse than the obtained one from MVO. In addition, it can be observed that all the proposed algorithms iterate to the maximum number of iteration to solve this problem.

Tension/compression spring design is one of the engineering problems. The obtained results for this problem are presented in Table 5. In this table, CMVO with piecewise and sine chaotic maps obtained the best score. Moreover, it can be observed the high stability of CMVO algorithms compared with the original MVO. This can prove the superiority of embedding chaotic maps in searching iterations of MVO. Also, it can be observed that CMVO with tent and Gauss chaotic maps took a larger number of function evaluations compared with MVO.

A comparison between MVO with CMVO algorithms for solving welded beam problem is proposed in Table 6. As it can be seen from this table, CMVO with sine and piecewise chaotic maps overtake MVO.

Table 6. Comparison of statistical results obtained using MVO and different version of CMVO for welded beam problem.

	Worst	Mean	Best	SD	Avg time	NFEs
MVO	1.963254	1.800613	1.724855	8.15E−02	4.695001	50000
CMVO1	1.857987	1.769055	1.724854	5.88E−02	4.644243	50000
CMVO2	1.917060	1.777848	1.725094	6.98E−02	4.606168	50000
CMVO3	4.983510	3.330243	2.012418	1.22	4.776203	50000
CMVO4	1.796269	1.749781	1.724860	2.95E−02	4.647799	50000
CMVO5	1.832774	1.770342	1.724855	4.38E−02	4.605032	50000
CMVO6	1.785743	1.749125	1.724852	2.58E−02	4.623860	50,000
CMVO7	1.823003	1.749254	1.724852	3.97E−02	4.597628	50,000
CMVO8	1.856951	1.749839	1.724931	4.82E−02	4.693158	50000
CMVO9	1.839914	1.749088	1.724859	4.19E−02	4.714233	50000
CMVO10	1.837646	1.759558	1.724862	4.74E−02	4.701263	50000

Table 7. Comparison of statistical results obtained using MVO and different version of CMVO for rolling element bearing maximization problem.

	Worst	Mean	Best	SD	Avg time	NFEs
MVO	83230.0150	84325.455565	85594.335698	1.27E+03	3.4916	50,000
CMVO1	81255.6544	85271.544513	85661.126865	1.45E+02	3.5450	50,000
CMVO2	84154.5392	84542.544539	85682.539204	8.95E+02	3.5207	50,000
CMVO3	83588.4845	85305.577374	85685.373523	1.35E+02	3.6077	50,000
CMVO4	85607.0461	85620.046165	85686.477101	1.06E+02	3.6161	50,000
CMVO5	83751.4454	85212.481178	85682.811777	4.20E+02	3.7734	50,000
CMVO6	82608.1454	85368.554120	85678.119594	6.46E+01	3.5420	50,000
CMVO7	84545.7545	85556.745454	85679.783659	1.23E+02	3.5965	50,000
CMVO8	84544.4545	85060.254410	85682.097902	3.77E+02	3.5051	50,000
CMVO9	85006.4484	85307.845453	85676.829009	1.24E+02	3.5341	50,000
CMVO10	85044.4444	85452.545435	85683.352578	2.06E+02	3.5578	50,000

Table 8. Comparison of statistical results obtained using MVO and different version of CMVO for multiple disc clutch brake problem.

	Worst	Mean	Best	SD	Avg Time	NFEs
MVO	0.40120	0.351241	0.313656	2.31E-05	3.350355	50,000
CMVO1	0.32316	0.320104	0.313656	3.22E−07	3.234352	50,000
CMVO2	0.37442	0.323544	0.313656	2.45E−04	3.258809	50,000
CMVO3	0.40134	0.324521	0.313656	2.47E−03	3.289344	50,000
CMVO4	0.33454	0.318545	0.313656	5.78E−07	3.281233	50,000
CMVO5	0.35474	0.316440	0.313656	3.55E−07	3.299792	50,000
CMVO6	0.35487	0.323112	0.313656	6.45E−06	3.326158	50,000
CMVO7	0.33745	0.313944	0.313656	2.54E−07	3.341179	50,000
CMVO8	0.35612	0.318754	0.313656	2.64E−06	3.343347	50,000
CMVO9	0.35154	0.319822	0.313656	2.98E−06	3.391360	50,000
CMVO10	0.35414	0.328745	0.313656	3.75E−05	3.404880	50,000

In addition, it can be observed that CMVO with the piecewise chaotic map has the highest stability, while MVO has the lowest stability. This is due to the random parameters embedded in the searching mechanism of MVO. Rolling element bearing problem is one of the maximization problems. Table 7 shows the statistical, average computational time and the number of function evaluation results for rolling element bearing maximization problem. As it can be observed from this table, CMVO with the iterative chaotic map obtained the best solution which is the maximum a rolling element bearing. CMVO with Gauss/mouse is in second place. Also, it can be observed that all CMVO versions have higher stability than MVO.

Table 8 compares the obtained results from MVO with other versions of CMVO for multiple disc clutch brake problem. As it can be observed from Table 8, all CMVO algorithms and MVO reached to

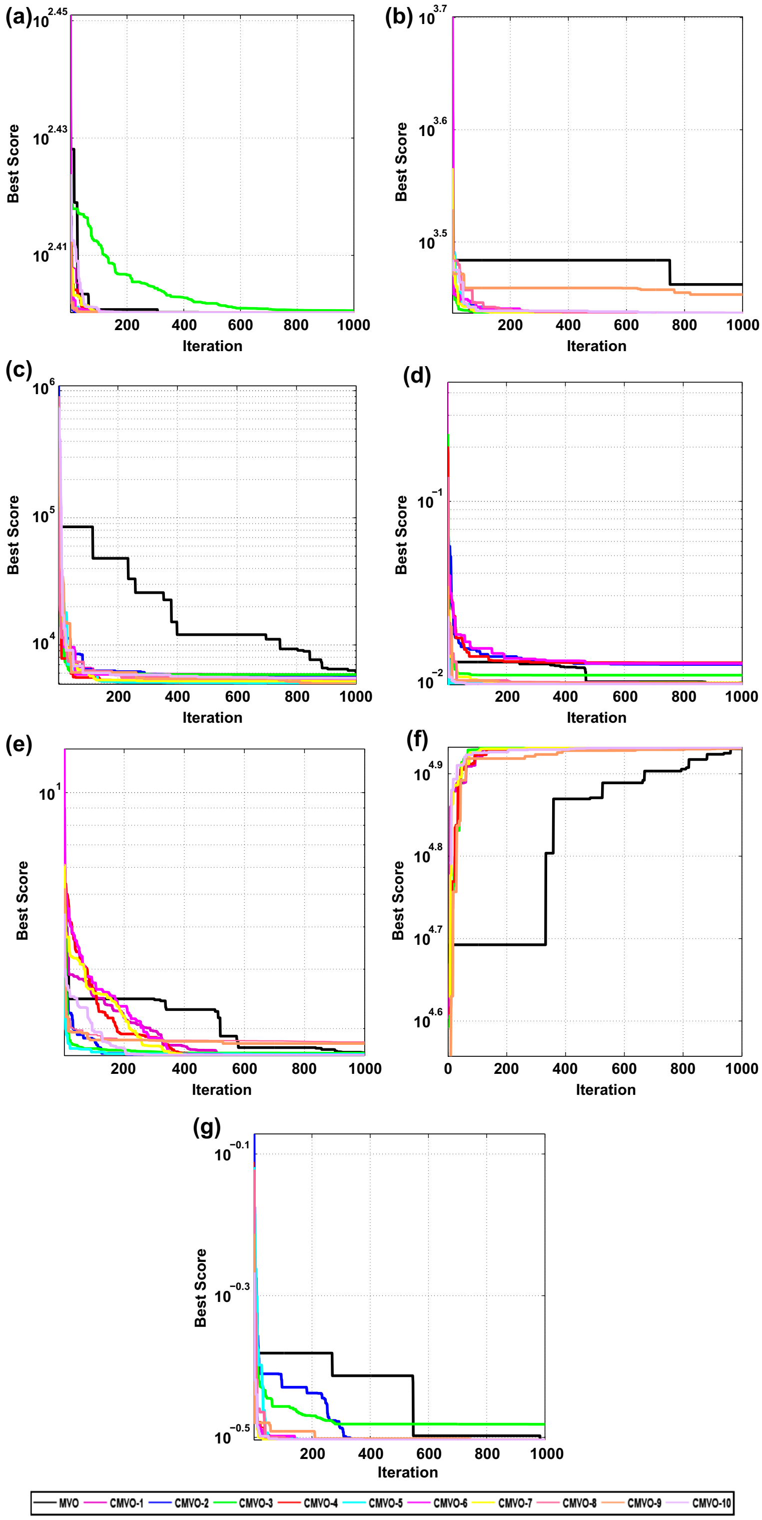

Figure 2. Convergence curve of: (a) Three-bar truss design problem, (b) Speed reducer design problem, (c) Pressure vessel design problem, (d) Tension/compression spring design problem, (e) Welded beam design problem, (f) Rolling element bearing design problem and (g) Multiple disc clutch brake design problem.

Table 9. Comparison of best obtained results using various optimization algorithms for three-bar truss design problem.

	DEDS	PSO-DE	WCA	MBA	MVO	CMVO
z_1	0.788675	0.788675	0.788651	0.788565	0.788676	0.788675
z_2	0.408248	0.408248	0.408316	0.4085597	0.408453	0.408248
$h_1(\vec{z})$	1.77E−08	−5.29E−11	0	−5.29E−11	−2.85E−08	−5.97E−10
$h_2(\vec{z})$	−1.464101	−1.463747	−1.464024	−1.463747	−1.465499	−1.464402
$h_3(\vec{z})$	−0.535898	−0.536252	−0.535975	−0.536252	−0.534214	−0.535598
$f(\vec{z})$	263.89584	263.89584	263.89584	263.89585	263.89595	263.89584

the same solution. However, CMVO with sine chaotic map reached to the optimal solution faster than the other algorithms, as it obtains the lowest standard deviation. This can indicate the high stability and good performance of embedding sine chaotic map in searching iterations of MVO algorithm. From the obtained results, it can be concluded that CMVO can significantly boost the performance of MVO.

For further comparison and evaluation of CMVO with different chaotic maps, the graphical representation of the convergence curve of MVO and CMVO versions are analysed as well. The convergence curves of CMVO with different chaotic maps on seven benchmark optimization engineering design problems are shown in Figure 2. As it can be seen, embedding chaotic maps in the searching mechanism of MVO outperform the original MVO. That's mean, CMVO algorithms reach the near optimum point in early iterations. In addition, it can be observed that CMVO algorithms have lower fluctuations in the curves. That can prove the high stability of CMVO algorithms. These observations are consistent with the obtained results from Tables 2–8.

From the obtained results for seven constrained problems, it can be concluded that CMVO with sine chaotic map obtained the optimal solutions in most cases. CMVO with piecewise and logistics chaotic map are in second place. Therefore, we choose CMVO with sine chaotic map as the optimal chaotic map. As it significantly boosts the performance of MVO. Also, this map will be evaluated and compared with other meta-heuristic algorithms at the following section.

4.1.2. CMVO vs. other meta-heuristic algorithms experiment

The main objective of this experiment is to compare the performance of CMVO with sine chaotic map with other meta-heuristic algorithms proposed in the literature.

The best solution obtained for three-bar truss design problem along with their corresponding constraint values are reported in Table 9, where 'NA' denotes not available. In this table, the performance of the proposed CMVO is compared with dynamic stochastic selection (DEDS) (Zhang, Luo, & Wang, 2008), PSO-DE (Liu & Cai, 2010), water cycle algorithm (WCA) (Eskandar, Sadollah, Bahreininejad, & Hamdi, 2012), mine blast algorithm (MBA) (Sadollaha, Bahreininejada, Eskandarb, & Hamdi, 2013) and original MVO. As it can be seen from this table, CMVO is a competitive algorithm. It obtained the minimum value with violating the constraints.

The algorithms applied to solve speed reducer design problem for comparison are DEDS, differential evolution with level comparison (DELC) (Eskandar et al., 2012), hybrid evolutionary algorithm and adaptive constraint handling technique (HEAA) (Wang, Cai, Zhou, & Fan, 2009), MDE, PSO-DE, WCA, MBA and MVO. Their best solutions obtained by these algorithms and CMVO are shown in Table 10. As it can be observed that the WCA, DELC, DEDS, MVO and CMVO outperformed other considered optimization engines as shown in Table 10. Moreover, it can be observed that HEAA violates the constraint that z_3 should be an integer value defined between 17 and 28.

The GA based on using of dominance-based tour tournament selection (GA-TTS) (Coello & Mezura, 2002), hybrid particle swarm optimization(HPSO) (He & Wang, 2007), CPSO, hybrid Nelder–mead simplex search and particle swarm optimization (NM–PSO) (Zahara & Kao, 2009b), co-evolutionary differential evolution (CDE) (Huang, Wang, & He, 2007), PSO, Gaussian quantum-behaved particle swarm optimization (G-QPSO), GSA, WCA and MBA (Coelho, 2010) algorithms were applied to the pressure vessel optimization problem. Their optimal results along with their constraints are shown

Table 10. Comparison of best obtained results using various optimization algorithms for speed reducer design problem.

	DEDS	DELC	HEAA	MDE	PSO-DE
z_1	3.5	3.5	3.500022	3.50001	3.5
z_2	0.7	0.7	0.7	0.7	0.7
z_3	17	17	17.000012	17	17
z_4	7.3	7.3	7.300427	7.300156	7.3
z_5	7.715319	7.715319	7.715377	7.800027	7.8
z_6	3.350214	3.350214	3.35023	3.350221	3.350214
z_7	5.285665	5.286654	5.28663	5.286685	5.286683
$h_1(\vec{z})$	−0.073915	0.073915	−0.073923	−0.07391528	NA
$h_2(\vec{z})$	−0.197999	−1.98E−01	−0.198006	−0.197998527	NA
$h_3(\vec{z})$	−0.499172	−0.49917	0.499095	−0.499172248	NA
$h_4(\vec{z})$	−0.904644	−0.90464	0.904643	−0.904643905	NA
$h_5(\vec{z})$	−0.000001	−1.33E−15	0.000014	−1.33E−15	NA
$h_6(\vec{z})$	0	0	0.000005	0	NA
$h_7(\vec{z})$	−0.7025	−0.7025	0.702500	−0.7025	NA
$h_8(\vec{z})$	0	0	0.000006	0	NA
$h_9(\vec{z})$	−0.583333	−0.58333	0.58333	−0.795833333	NA
$h_{10}(\vec{z})$	−0.051326	−0.051326	0.05137	−0.051325754	NA
$h_{11}(\vec{z})$	0	−5.55E−16	0.000006	−5.56E−16	NA
$f(\vec{z})$	2994.471	2994.471	2.994.499	2996.357	2996.342

	WCA	MBA	MVO	CMVO
z_1	3.5	3.5	3.5	3.5
z_2	0.7	0.7	0.7	0.7
z_3	17	17	17	17
z_4	7.3	7.3	7.3	7.3
z_5	7.715319	7.715772	7.715319	7.715319
z_6	3.350214	3.350218	3.350214	3.350214
z_7	5.286654	5.286654	5.286654	5.286654
$h_1(\vec{z})$	NA	NA	−0.073915	−0.073915
$h_2(\vec{z})$	NA	NA	−0.197998	−1.98E−01
$h_3(\vec{z})$	NA	NA	−0.499171	-4.99E−01
$h_4(\vec{z})$	NA	NA	−0.904644	−9.05E-01
$h_5(\vec{z})$	NA	NA	−1.30E−15	−5.96E−07
$h_6(\vec{z})$	NA	NA	0	0
$h_7(\vec{z})$	NA	NA	−0.7025	−0.7025
$h_8(\vec{z})$	NA	NA	0	0
$h_9(\vec{z})$	NA	NA	−0.795833	−0.795833
$h_{10}(\vec{z})$	NA	NA	−0.051325	−5.09E−02
$h_{11}(\vec{z})$	NA	NA	−5.18E−08	−1.30E−06
$f(\vec{z})$	2994.471	2994.482	2994.471	2994.471

in Table 11. As it can be observed, NM-PSO, WCA and MBA achieved the minimum values. However, their solutions do not satisfy that z_1 and z_2 should be discrete values not continuous which are integer multiples of 0.0625 inch. CMVO and G-QPSO are in second place, as they obtained the optimal results compared with the other constrained algorithms. The results obtained by CMVO and G-QPSO achieve the design with the best solution without violating any constraints.

Table 12 compares the best solutions for tension/compression spring design problem obtained by the proposed CMVO and DEDS, CPSO, GA-TTS, HEAA, NM-PSO, MBA, DELC, WCA and MVO. As it can be seen, NM-PSO obtained the optimal solution. However, NM-PSO violates two constraints. CMVO is in second place after NM-PSO. Welded beam problem obtained by GA-TTS, CPSO, CAEP, HGA, NM-PSO, WCA, gravitational search algorithm (GSA) (Rashedi, Nezamabadi-Pour, & Saryazdi, 2009), HPSO, MVO and CMVO are reported in Table 13. As it can be seen, NM-PSO overtakes the other algorithms. However, this algorithm violates the third constraint, where the value should be less than or equals to zero. CMVO and HPSO obtained the optimal results without violating any constraints. The optimal solution for rolling element bearing problem obtained by the proposed CMVO and multi-objective genetic algorithm (M-GA) (Gupta, Tiwari, & Shivashankar, 2007), TLBO, MBA, WCA and MVO are shown

Table 11. Comparison of best obtained results using various optimization algorithms for pressure vessel problem.

	GA-TTS	CDE	CPSO	HPSO	NM-PSO	G-QPSO
z_1	0.8125	0.8125	0.8125	0.8125	0.8036	0.8125
z_2	0.4375	0.4375	0.4375	0.4375	0.3972	0.4375
z_3	42.0974	42.0984	42.0913	42.0984	41.6392	42.0984
z_4	176.654	176.6376	176.765	176.6366	182.412	176.6372
$h_1(\vec{z})$	−2.01E−03	−6.67E−07	−1.37E−06	−8.80E−07	3.65E−05	−8.79E−07
$h_2(\vec{z})$	−3.58E−02	−3.58E−02	−3.59E−04	−3.58E−02	3.79E−05	−3.58E−02
$h_3(\vec{z})$	−24.7593	−3.705123	−118.7687	3.1226	−1.5914	−0.2179
$h_4(\vec{z})$	-63.346	-63.3623	-63.2535	-63.3634	−57.5879	−63.3628
$f(\vec{z})$	6059.9463	6059.734	6061.0777	6059.7143	5930.3137	6059.7208
	GSA	MBA	WCA	MVO	CMVO	
z_1	1.125	0.7802	0.7781	0.8125	0.8125	
z_2	0.625 0.3856	0.3846	0.4375	0.4375		
z_3	55.9887	40.4292	40.3196	42.098	42.0984	
z_4	84.4542	198.4964	−200	176.6502	176.6372	
$h_1(\vec{z})$	NA	0	−2.95E−11	−8.60E−06	−8.80E−07	
$h_2(\vec{z})$	NA	0	−7.15E−11	−3.58E−02	−0.03588	
$h_3(\vec{z})$	NA	−8.64E+01	−1.35E−06	−43.8867	−0.21798	
$h_4(\vec{z})$	NA	−4.15E+01	−40	−63.3499	−63.3627	
$f(\vec{z})$	8538.8359	5889.3216	<u>5885.3327</u>	6059.9528	6059.7208	

Table 12. Comparison of best obtained results using various optimization algorithms for tension/compression spring design problem.

	DEDS	GA-TTS	CPSO	HEAA	NM-PSO	DELC
z_1	0.051689	0.051989	0.051728	0.051689	0.05162	0.051689
z_2	0.356717	0.363965	0.357644	0.356729	0.355498	0.356717
z_3	11.288965	10.890522	11.244543	11.288293	11.333272	11.288965
$h_1(\vec{z})$	1.45E−09	−1.26E−03	−8.25E−04	3.96E−10	1.01E−03	−3.40E−09
$h_2(\vec{z})$	−1.19E−09	−2.54E−05	−2.52E−05	3.59E−10	9.94E−04	2.44E−09
$h_3(\vec{z})$	−4.053785	−4.061337	−4.051306	−4.053808	−4.061859	-4.053785
$h_4(\vec{z})$	−0.727728	−0.722697	−0.727085	−0.72772	−0.728588	−0.727728
$f(\vec{z})$	0.012665	0.012681	0.012674	0.012665	<u>0.01263</u>	0.012665
	MBA	WCA	MVO	CMVO		
z_1	0.051656	0.05168	0.051689	0.051689		
z_2	0.35594	0.356522	0.3567409	0.356716		
z_3	11.344665	11.30041	11.288291	11.289012		
$h_1(\vec{z})$	0	1.65E−13	−0.0007813	−2.95E−09		
$h_2(\vec{z})$	0	−7.90E−14	−0.0143359	−1.94E−09		
$h_3(\vec{z})$	−4.052248	−4.053399	−3.053784	−4.053784		
$h_4(\vec{z})$	−0.728268	−0.727864	−0.727728	−0.727729		
$f(\vec{z})$	0.012665	0.012665	0.0126982	0.012665		

in Table 14. From this table, CMVO show great performance, as it obtains the best solution which is 85679.783659. MVO is in second place and WCA is in third place. The problem of the multiple disc clutch brake was also solved previously using NSGA-II (Deb & Srinivasan, 2008), TLBO and WCA. The comparison of best solutions along with their constraints is reported in Table 15. As it can be observed, CMVO, MVO, TLBO and WCA give the same optimal solution for the multiple disc clutch brake problem.

4.2. The performance of CMVO for unconstrained benchmark optimization functions experiment

In this experiment, the performance of the proposed CMVO with sine chaotic map is compared with MVO, MVO, moth flame optimization (MFO) (Mirjalili, 2015) and artificial bee colony (ABC) (Karaboga & Basturk, 2007) in terms of mean fitness value and standard deviation (std) as shown in Table 17. Thirty

Table 13. Comparison of best obtained results using various optimization algorithms for welded beam problem.

	GA-TTS	CPSO	CAEP	HGA	NM-PSO	WCA
$z_1(h)$	0.20599	0.20237	0.20573	0.20573	0.20583	0.20573
$z_2(l)$	3.4712	3.5442	3.4705	3.4705	3.468338	3.4705
$z_3(t)$	9.02022	9.04821	9.03662	9.03662	9.03662	9.03662
$z_4(b)$	0.2065	0.20572	0.20573	0.20573	0.20573	0.20573
$h_1(\vec{z})$	−0.10305	−13.65555	1.988676	1.988676	−0.02525	−0.034128
$h_2(\vec{z})$	−0.23175	−78.81408	4.481548	4.481548	−0.053122	−3.49E−05
$h_3(\vec{z})$	−5.0E−04	−3.35E−03	0	0	0.0001	−1.19E−06
$h_4(\vec{z})$	−3.43044	−3.424572	−3.433213	−3.433213	−3.433169	−3.43298
$h_5(\vec{z})$	−0.08099	−0.077369	−0.0507	−0.0807	−0.08083	−0.08072
$h_6(\vec{z})$	−0.23551	−0.235595	−0.235538	−0.235538	−0.23554	−0.23554
$h_7(\vec{z})$	−58.6469	−4.472858	2.603347	2.603347	−0.031555	−0.013503
$f(\vec{z})$	1.72823	1.728024	1.724852	1.724852	1.724717	1.724856
	GSA	HPSO	MVO	CMVO		
$z_1(h)$	0.1821	0.20573	0.20573	0.20573		
$z_2(l)$	0.8570	3.4705	3.4705	3.4705		
$z_3(t)$	10	9.03662	9.036662	9.03662		
$z_4(b)$	0.2024	0.20573	0.20574	0.20573		
$h_1(\vec{z})$	NA	−0.025399	−0.019144	−0.04788		
$h_2(\vec{z})$	NA	−0.053122	−0.53122	−0.30792		
$h_3(\vec{z})$	NA	0	−0.00662	−7.7E−04		
$h_4(\vec{z})$	NA	−3.432981	−3.4144	−3.43298		
$h_5(\vec{z})$	NA	−0.08073	−0.08073	−0.08073		
$h_6(\vec{z})$	NA	−0.23554	−0.23554	−0.23553		
$h_7(\vec{z})$	NA	−0.031555	−0.032454	−0.03155		
$f(\vec{z})$	1.880	1.724852	1.724855	1.724852		

Table 14. Comparison of best obtained results using various optimization algorithms for rolling element bearing maximization problem.

	GA4	TLBO	MBA	WCA	MVO	CMVO
z_1	125.7170	125.7191	125.7153	125.7211	125.5893	125.6959
z_2	21.42300	21.42559	21.42330	21.42330	21.43685	21.44271
z_3	11.00000	11.00000	11.00000	11.00103	10.99363	11.00165
z_4	0.515000	0.515000	0.515000	0.515000	0.515000	0.515000
z_5	0.515000	0.515000	0.515000	0.515000	0.515000	0.515000
z_6	0.415900	0.424266	0.488805	0.401514	0.466970	0.448186
z_7	0.651000	0.633948	0.627829	0.659047	0.671195	0.642065
z_8	0.300043	0.300000	0.300149	0.300032	0.300000	0.300189
z_9	0.022300	0.068858	0.097305	0.040045	0.032447	0.020780
z_{10}	0.751000	0.799498	0.646095	0.600000	0.616751	0.669704
$h_2(\vec{z})$	13.73299	13.15257	−8.630183	14.74059	10.185804	11.51243
$h_3(\vec{z})$	2.724000	1.525200	−1.101429	3.286749	4.109907	2.059158
$h_4(\vec{z})$	3.606000	0.719056	−2.040448	3.423300	2.934324	1.351586
$h_5(\vec{z})$	0.717000	16.49544	−0.715337	0.721167	−0.589363	−0.695913
$h_6(\vec{z})$	4.857899	0	−23.61100	9.290112	−7.522279	−4.499024
$h_7(\vec{z})$	0.003050	0	−0.000480	0.000087	−0.055835	0.006184
$h_8(\vec{z})$	0.000007	2.559363	0	0	0	0
$h_9(\vec{z})$	0.000007	0	0	0	0	0
$h_{10}(\vec{z})$	0.000005	0	0	0	0	0
$f(\vec{z})$	81843.3000	81859.7400	85535.6911	85538.4800	85594.3356	85679.7836

independent runs with the number of iterations equal to 1000 and the number of search agents equal to 50 with the same initialization method of the search agents are used for all adopted algorithms in this experiment. The reason behind this, we want to make almost a fair comparison of all competitor algorithms. In this table, the best results are underlined. In this table, two different categories of benchmark problems with different characteristics are used in the evaluation. These categories are

Table 15. Comparison of best obtained results using various optimization algorithms for multiple disc clutch brake problem.

	NSGA-II	TLBO	WCA	MVO	CMVO
z_1	70	70	70	70	70
z_2	90	90	90	90	90
z_3	1.5	1	1	1	1
z_4	1000	810	910	910	910
z_5	3	3	3	3	3
$h_1(\vec{z})$	0	0	0	0	0
$h_2(\vec{z})$	22	24	24	24	24
$h_3(\vec{z})$	0.900500	0.919427	0.909480	0.909480	0.909480
$h_4(\vec{z})$	9.790600	9830.371000	0.909480	0.909480	0.909480
$h_5(\vec{z})$	7.894700	7894.696500	7.894696	7.894696	7.894696
$h_6(\vec{z})$	3.352700	0.702013	2.231421	2.231421	2.231421
$h_7(\vec{z})$	60.625000	37706.250000	12.768749	12.768749	12.768749
$h_8(\vec{z})$	11.647300	14.297860	12.768578	12.768578	12.768578
$f(\vec{z})$	0.470400	<u>0.313656</u>	<u>0.313656</u>	<u>0.313656</u>	<u>0.313656</u>

Table 16. Properties of benchmark functions, Dim denotes dimensions, Opt denotes optimum point.

No.	Name	Range	Opt	Dim	Modality
$F1$	Sphere	$[-100, 100]$	0	50	Unimodal
$F2$	Schwefel 2.22	$[-10, 10]$	0	50	Unimodal
$F3$	Rotated Hyper−Ellipsoid	$[-100, 100]$	0	50	Unimodal
$F4$	Schwefel 2.21	$[-100,100]$	0	50	Unimodal
$F5$	Rosenbrock's valley	$[-30, 30]$	0	50	Unimodal
$F6$	Step−2	$[-100, 100]$	0	50	Unimodal
$F7$	Quartic with noise	$[-1.28, 1.28]$	0	50	Unimodal
$F8$	Schwefel	$[-500, 500]$	-418.9829×50	50	Multimodal
$F9$	Rastrign	$[-5.12, 5.12]$	0	50	Multimodal
$F10$	Ackely	$[-32, 32]$	0	50	Multimodal
$F11$	Griewank	$[-600, 600]$	0	50	Multimodal
$F12$	Penalty 1	$[-50, 50]$	0	50	Multimodal
$F13$	Penalty 2	$[-50, 50]$	0	50	Multimodal

seven unimodal and six multimodal. The first category, namely unimodal test functions has only one global optimum and no local optima. These functions are used to evaluate the exploitation and the convergence rate of the algorithm. However, the second category, namely multimodal test functions has multiple global optimum and multiple local optima. The characteristic of this category is used to evaluate the capability of the algorithm in avoiding the local optima and the explorative ability of an algorithm. Table 16 shows the properties of the used functions. In the table, Dim indicates the number of dimensions and Opt indicates the optimum value of the function. More information to the adopted benchmark functions with references can be found in (Digalakis & Margaritis, 2001; Molga & Smutnicki, 2005; Jamil & Yang, 2013). As it can be observed from Table 17 the CMVO overtakes the other algorithms for both unimodal and multimodal test function. Also, it can be noticed that the modified version of MVO using chaos computing outperforms the standard version of MVO for the most of benchmark functions. This can prove the ability to embed chaotic maps in the searching methodology of MVO in enhancing the performance for both exploration and exploitation.

Moreover, the results of Wilcoxon's rank sum test for CMVO vs. MVO, CMVO vs. ABC and CMVO vs. MFO are presented in Table (17), which gives p values stating whether a statistical difference is significant or not. The smaller obtained values of p, the greater the difference between the provided algorithms. The CMVO algorithm shows similar performance to MVO for only F_5, and shows great improvement over MFO and ABC with significance level $\alpha = 0.05$.

These overall optimization results report that embedding chaotic maps in the searching iterations of MVO improves the performance of MVO. Furthermore, the results show that sine chaotic is the

Table 17. Statistical and *P*-values of the Wilcoxon ranksum test results for 13 benchmark functions.

	ABC			MFO		
	Avg	Std	*P*-value	Avg	Std	*P*-value
F1	1.6E+04	2.8E+03	1.23E−05	5.1E+03	5.2E+03	1.23E−05
F2	2.1E+09	1.1E+10	1.73E−06	9.8E+01	3.0E+01	3.18E−06
F3	2.2E+05	4.3E+04	1.73E−06	2.8E+04	1.3E+04	1.73E−06
F4	8.8E+01	3.4E+00	1.34E−09	7.6E+01	3.7E+00	1.98E−09
F5	1.8E+08	4.0E+07	1.73E−06	1.9E+07	3.4E+07	1.73E−06
F6	6.4E+03	2.1E+03	1.83E−09	3.7E+03	4.9E+03	1.00E−11
F7	1.1E+02	2.6E+01	1.76E−09	5.2E+01	4.5E+01	1.98E−06
F8	−1.1E+118	4.7E+18	1.73E−06	−1.4E+04	1.4E+03	2.70E−02
F9	5.7E+02	1.8E+01	1.44E−07	3.4E+02	4.5E+01	2.35E−06
F10	1.8E+01	1.0E+00	1.36E−01	1.9E+01	1.3E+00	1.66E−02
F11	5.8E+01	1.9E+01	2.40E−06	4.4E+01	5.7E+01	1.54E−06
F12	5.5E+08	1.2E+08	2.98E−11	1.7E+04	8.3E+04	2.62E−09
F13	8.5E+08	1.9E+08	1.78E−09	9.6E+07	1.8E+08	1.92E−11

	CMVO		MVO		
	Avg	Std	Avg	Std	*P*-value
F1	1.1E−02	2.1E−03	9.9E+00	2.5E+00	1.23E−05
F2	2.3E+01	2.3E+01	3.3E+02	1.8E+02	7.27E−03
F3	6.4E+02	2.5E+02	1.7E+03	5.8E+02	1.92E−06
F4	7.1E+00	3.0E+00	3.3E+01	4.9E+00	1.73E−06
F5	2.2E+02	3.2E+02	4.5E+02	5.3E+02	1.02E−01
F6	1.1E−02	2.4E−03	2.3E+00	5.5E−01	1.73E−06
F7	1.8E−01	4.6E−02	6.7E−02	1.8E−02	2.46E−02
F8	−1.3E+04	8.4E+02	−1.2E+04	8.5E+02	9.63E−04
F9	2.2E+02	5.2E+01	2.6E+02	2.6E+01	1.96E−03
F10	1.8E+01	4.8E−01	2.4E+00	5.4E−01	1.25E−05
F11	3.6E−02	1.2E−02	9.4E−01	6.6E−02	1.63E−04
F12	3.9E+00	1.4E+400	1.7E+01	3.9E+00	1.73E−06
F13	6.1E−01	4.3E−01	2.9E+01	2.7E+01	6.16E−04

optimal chaotic to significantly boost the performance of MVO. This finding is similar to the same finding reported in the following Talatahari, Azar, Sheikholeslami, and Gandomi (2012a), Talatahari, Kaveh, and Sheikholeslami (2012b) and Talatahari, Kaveh, and Sheikholeslami (2012) with different meta-heuristic algorithms. Moreover, the results show the capability of CMVO algorithms in handling various combinatorial optimization problems under lower computational efforts. CMVO can reach to the optimal or near optimal solutions better than to most of the previously reported results. Thus, it can be concluded that the proposed CMVO algorithm is an attractive alternative optimiser for handling constrained and unconstrained problems.

5. Conclusions and future work

This paper presented a new hybrid algorithm called CMVO based on chaotic theory and MVO. Ten chaotic maps were used to improve the performance and convergence speed of MVO algorithm. Chaotic maps regulate exploration–exploitation trade-off in a more balanced manner by creating noise in the search region. The proposed CMVO algorithm was successfully evaluated on 7 constrained and 13 unconstrained benchmark optimization problems. The simulation results are compared with other popular optimization algorithms. The results show the superiority of the proposed CMVO in solving numerous constrained and unconstrained problems. In addition, the obtained results show that proposed CMVO obtains better solutions compared with other optimization algorithms. Moreover, the experimental results show that the adjusted variable using sine map can significantly boost the performance of MVO.

Further work on large-scale optimization problems can be considered as the future scope of this paper. CMVO can be tested in other various real-world applications such as wireless sensor networks, and business optimization problems, etc. A self-adaptive version of CMVO algorithm can also be developed in future.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Gehad Ismail Sayed http://orcid.org/0000-0001-9007-916X

References

Abdullah, A., Enayatifa, R., & Lee, M. (2012). A hybrid genetic algorithm and chaotic function model for image encryption. *Journal of Electronics and Communication, 66*, 806–816.

Alatas, B. (2010). Chaotic harmony search algorithms. *Applied Mathematics and Computation, 216*, 2687–2699.

Alatas, B., Akin, E., & Ozer, A. (2009). Chaos embedded particle swarm optimization algorithms. *Journal of Chaos Soliton & Fractals, 4*, 1715–1734.

Changsen, W. (1991). *Analysis of Rolling Element Bearings*. London: Mechanical Engineering Publications Ltd..

Chaoshun, L., Xueli, A., & Ruhai, L. (2015). A chaos embedded gsa-svm hybrid system for classification. *Neural Computing and Applications, 26*(3), 713–721.

Coelho, L. (2010). Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Systems with Applications, 37*, 1676–1683.

Coello, C., & Mezura, E. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics, 16*, 193–203.

Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering, 186*(2), 311–338.

Deb, K., & Srinivasan, A. (2008). *Innovization: Discovery of innovative design principles through multiobjective evolutionary optimization* (pp. 243–262). Berlin: Springer.

Digalakis, J., & Margaritis, K. (2001). On benchmarking functions for genetic algorithms. *International Journal of Computer Mathematics, 77*, 481–506.

Eberhart, R., & Kennedy, J. (1995, October 4–6). *IEEE MHS'95. Proceedings of the sixth international symposium on micro machine and human science – a new optimizer using particle swarm theory* (pp. 39–43). Nagoya, Japan.

Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm - a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers and Structures, 110*, 151–166.

Faris, H., Aljarah, I., & Mirjalili, S. (2016). Training feedforward neural networks using multi-verse optimizer for binary classification problems. *Applied Intelligence, 45*(2), 322–332.

Gandomi, A., & Yangb, X. (2014). Chaotic bat algorithm. *Journal of Computational Science, 5*, 224–232.

Gandomi, A., Yang, X., Talatahari, S., & Alavi, A. (2013). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation, 18*(1), 89–98.

Gandomi, A., Yun, G., Yang, X., & Talatahari, S. (2013). Chaos-enhanced accelerated particle swarm algorithm. *Communications in Nonlinear Science and Numerical Simulation, 18*(2), 327–340.

Gharooni-fard, G., Moein, F., Deldari, H., & Morvaridi, A. (2010). Scheduling of scientific workflows using a chaos-genetic algorithm. *Procedia Computer Science, 1*, 1445–1454.

Ghasemi, M., Ghavidel, S., Aghaei, J., Gitizadeh, M., & Falah, H. (2014). Application of chaos-based chaotic invasive weed optimization techniques for environmental opf problems in the power system. *Chaos, Solitons & Fractals, 69*, 271–284.

Gong, W., & Wang, S. (2009). IEEE 2009 Fourth International Conference on Internet Computing for Science and Engineering (ICICSE). In *Chaos ant colony optimization and application* (pp. 301–303), Harbin, TBD, China.

Gupta, S., Tiwari, R., & Shivashankar, B. (2007). Multi-objective design optimization of rolling bearings using genetic algorithm. *Mechanism and Machine Theory, 42*, 1418–1443.

He, Q. & Wang, L. (2007). A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied Mathamtic Computation, 186*, 1407–1422.

Holland, J. H. (1992). *Adaptation in natural and artificial systems*. Cambridge, MA, USA: MIT Press.

Huang, F., Wang, L., & He, Q. (2007). An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation, 186*(1), 340–356.

Hu, C., Li, Z., Zhou, T., Zhu, A., & Xu, C. (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), 1–22.

Jamil, M., & Yang, X. (2013). A literature survey of benchmark functions for global optimization problems. *International Journal of Mathematical Modelling and Numerical Optimisation, 4*(2), 150–194.
Kannan, B. K., & Kramer, S. N. (1994). An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design, 116*, 318–320.
Karaboga, D., & Basturk, B. (2007). *Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems*, 12th International Fuzzy Systems Association World Congress (Vol. 4529, pp. 789–798). Mexico
Kaveh, A., Sheikholeslami, R., Talatahari, S., & Keshvari-Ilkhichi, M. (2014). Chaotic swarming of particles: a new method for size optimization of truss structures. *Advances in Engineering Software, 67*, 136–147.
Liu, H., Cai, Z., & Wang, Y. (2010). Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing, 10*, 629–640.
May, R. (1976). Mathematical models with very complicated dynamics. *Nature, 261*, 459–467.
Mirjalili, S. M. (2015). *Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm*, Knowledge-based systems (Vol. 89, pp. 228–249). Elsevier.
Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications, 27*(2), 1–19.
Molga, M., & Smutnicki, A. (2005). Test functions for optimization needs. unpublished paper.
Montes, E. & Coello, C. (2008). An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *International Journal of General Systems, 37*, 443–473.
Özkaynak, F. (2015). A novel method to improve the performance of chaos based evolutionary algorithms. *Optik, 126*, 5434–5438.
Peitgen, H., Jurgens, H., & Saupe, D. (1992). *Chaos and fractals*. Berlin, Germany: Springer-Verlag.
Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). Gsa: A gravitational search algorithm. *Information Science, 179*, 2232–2248.
Sadollaha, A., Bahreininejada, A., Eskandarb, H., & Hamdi, M. (2013). Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing, 13*(5), 2592–2612.
Saremi, S., Mirjalili, S., & Lewis, A. (2014). Biogeography-based optimization with chaos. *Neural Computing and Applications, 25*(5), 1077–1097.
Sayed, Gehad Ismail, Hassanien, A.E., & Schaefer, G. (2016). An automated computer-aided diagnosis system for abdominal CT liver images. *Procedia Computer Science, 90*, 68–73.
Sayed, G., Darwish, A., & Hassanien, A. (2017a). Quantum multiverse optimization algorithm for optimization problems. *Neural Computing and Applications,* 1–18.
Sayed, G., Darwish, A., Hassanien, A., & Pan, J. (2017b). *Breast cancer diagnosis approach based on meta-heuristic optimization algorithm inspired by the bubble-net hunting strategy of whales* (pp. 306–13). Cham: Springer.
Sayed, G., & Hassanien, A. (2017). Moth-flame swarm optimization with neutrosophic sets for automatic mitosis detection in breast cancer histology images. *Applied Intelligence, 47*(2), 397–408.
Sayed, G., Hassanien, A., & Azar, A. (2017c). Feature selection via a novel chaotic crow search algorithm. *Neural Computing and Applications,* 1–18. doi:10.1007/s00521-017-2988-6.
Schuster, H. (1998). *Deterministic chaos: An introduction*. 2nd Revised ed., D-6940 Weinheim: Physick-Verlag GmnH, Federal Republic of Germany.
Talatahari, S., Azar, B., Sheikholeslami, R., & Gandomi, A. (2012a). Imperialist competitive algorithm combined with chaos for global optimization. *Communications in Nonlinear Science and Numerical Simulation, 17*(3), 1312–1321.
Talatahari, S., Kaveh, A., & Sheikholeslami, R. (2012). Engineering design optimization using chaotic enhanced charged system search algorithms. *Acta Mechanica, 223*(10), 2269–2285.
Talatahari, S., Kaveh, A., & Sheikholeslami, R. (2012b). Chaotic imperialist competitive algorithm for optimum design of truss structures. *Structural and Multidisciplinary Optimization, 46*(3), 355–367.
Wang, Y., Cai, Z., Zhou, Y., & Fan, Z. (2009). Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint handling technique. *Structural and Multidisciplinary Optimization, 37*, 395–413.
Wang, N., & Liu, L. (2010). Genetic algorithm in chaos. *OR Transactions, 5*, 1–10.
Wang, L., Zheng, D., & Lin, Q. (2001). Survey on chaotic optimization methods. *Computing Technology and Automation, 20*, 1–5.
Yang, L., & Chen, T. (2002). Application of chaos in genetic algorithms. *Communications in Theoretical Physics, 38*, 168–172.
Yang, C., Tsai, S., Chuang, L., & Yang, C. (2012). An improved particle swarm optimization with double-bottom chaotic maps for numerical optimization. *Journal of Applied Mathematics and computation, 219*, 1–20.
Zahara, E., & Kao, Y. (2009a). Hybrid nelder-mead simplex search and particle swarm optimization for constrained engineering design problems. *Expert System with Application, 36*, 3880–3886.
Zahara, E., & Kao, Y. (2009b). Hybrid nelder-mead simplex search and particle swarm optimization for constrained engineering design problems. *Expert Systems with Applications, 36*, 3880–3886.
Zhang, Q., Li, Z., Zhou, C. J., & Wei, X. P. (2013). Bayesian network structure learning based on the chaotic particle swarm optimization algorithm. *Genetics and Molecular Research, 12*(4), 4468–4479.
Zhang, M., Luo, W., & Wang, X. (2008). Differential evolution with dynamic stochastic selection for constrained optimization. *Information Science, 178*, 3043–3074.

Zhang, L., & Zhang, C. J. (2008). Hopf bifurcation analysis of some hyperchaotic systems with time-delay controllers. *Kybernetika, 44*(1), 35–42.
Zheng, W. (1994). Kneading plane of the circle map. *Chaos, Solitons & Fractals, 4*(7), 1221–1233.
Zhu, Z. L., Li, S. P., & Yu, H. (2008). A new approach to generalized chaos synchronization based on the stability of the error system. *Kybernetika, 44*(4), 492–500.

Appendix 1. Details of the 7 constrained design benchmark optimization problems

A.1. The design of welded beam problem

Welded beam design is a minimization problem of the cost under some constraints where there are four variables in this problem which are the length of bar attached to the weld (l), weld thickness (h), the height of the bar (t) and the thickness of the bar (b). Moreover, there are some constraints in this problem which are bending stress in the beam (α), the beam deflection (β), buckling load on the bar (BL), the end deflection of the beam (δ) and side constraints. This design problem can then formulated as follows.

Minimise the function (A1)

$$f(\vec{z}) = 1.10471 z_1^2 z_2 + 0.04811 z_3 z_4 (14.0 + z_2), \vec{z} = [z_1 z_2 z_3] = [hltb],$$

Subject to,

$$h_1(\vec{z}) = \delta(\vec{z}) - \delta_{\max} \leq 0,$$
$$h_2(\vec{z}) = \alpha(\vec{z}) - \alpha_{\max} \leq 0,$$
$$h_3(\vec{z}) = z_1 - z_4 \leq 0,$$
$$h_4(\vec{z}) = 0.10471 z_1^2 + 0.04811 z_3 z_4 (14.0 + z_2) - 5.0 \leq 0,$$
$$h_5(\vec{z}) = 0.125 - z_1 \leq 0,$$
$$h_6(\vec{z}) = \delta(\vec{z}) - \delta_{\max} \leq 0,$$
$$h_7(\vec{z}) = B - BL(\vec{z}) \leq 0$$

where,

$$0.1 \leq z_1 \leq 2,$$
$$0.1 \leq z_2 \leq 10,$$
$$0.1 \leq z_3 \leq 10,$$
$$0.1 \leq z_4 \leq 20$$

and

$$\delta(\vec{z}) = \sqrt{\delta'^2 + 2\delta'\delta^n \frac{z_2}{2R} + \delta^{n2}}$$
$$\delta' = \frac{B}{\sqrt{2} z_1 z_2}$$
$$\delta^n = \frac{MR}{J}$$
$$M = B\left(L + \frac{z_2}{2}\right)$$
$$R = \sqrt{\frac{z_2^2}{4} + (\frac{z_1 + z_3}{2})^2}$$
$$J = 2\left(\sqrt{2} z_1 z_2 \left[\frac{z_2^2}{12}\left(\frac{z_1 + z_3}{2}\right)^2\right]\right)$$
$$\alpha(\vec{z}) = \frac{6PL}{z_4 z_3^2}, \beta(\vec{z}) = \frac{4PL^3}{E z_3^2 z + z_4}$$
$$BL(\vec{z}) = \frac{4.013E\sqrt{\frac{z_3^2 z_4^6}{36}}}{L^2}\left(1 - \frac{z_3}{2L}\sqrt{\frac{E}{4G}}\right)$$
$$P = 6000 lb,\ L = 14 in.,\ \beta_{\max} = 0,\ 25 in.,\ E = 30 \times 10^6 psi,$$

$$G = 12 \times 10^6 psi,\ \delta_{\max} = 13600psi,\ \alpha_{\max} = 30000psi$$

A.2. Design of pressure vessel problem

The main objective of the problem of pressure vessel design is to minimise the total cost of the welding and forming of pressure vessel problem. In this problem, there are four variables which are thickness of the head (T_h), thickness of the shell (T_s), inner radius (R) and length of the cylindrical section of the vessel (L). Authors in Kannan and Kramer (1994) described the Design of Pressure vessel problem and used the following notations. $z_1,\ z_2$ are discrete values which are integer multiples of 0.0625 inch.

Minimise the function

$$f(\vec{z}) = 0.6224z_1z_2z_3z_4 + 1.7781z_2z_3^2 + 3.1661z_1^2z_4 + 19.84z_1^2z_3, (\vec{z}) \qquad = [z_1z_2z_3] = [T_sT_hRL], \quad \text{(A2)}$$

Subject to,

$$h_1(\vec{z}) = -z_1 + 0.0193z_3 \leq 0,$$
$$h_2(\vec{z}) = -z_2 + 0.00954z_3 \leq 0,$$
$$h_3(\vec{z}) = -\Pi * z_3{}^2 - z_4 - 4/3\Pi * z_3{}^3 + 1296.000 \leq 0,$$
$$h_4(\vec{z}) = z_4 - 240 \leq 0,$$

where,

$$0 \leq z_1 \leq 100,$$
$$0 \leq z_2 \leq 100,$$
$$10 \leq z_3 \leq 200,$$
$$10 \leq z_4 \leq 200$$

A.3. Speed reducer design problem

The weight of speed reducer in this optimization problem needs to be minimised under some constraints on the surface stress, bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stresses in the shafts. In this problem, there are seven variables are $z_1, z_2, \ldots, z_7$ and they are defined as the face width, module of teeth, number of the teeth in pinion, length of the first shaft between bearings, length of the second shaft between bearings and the diameter of the first and second shafts. Eskandar et al. (2012). All these variables are continuous except the third one which is integer. This problem can be mathematically described as follows.

Minimize the function

$$f(\vec{z}) = 0.7854z_1z_2^2(3.3333z_3^2 + 14.9334z_3 - 43.0934) - 1.508z_1(z_6^2 + z_7^2) + 7.4777(z_6^2 + z_7^2) + 0.7854(z_4z_6^2 + z_5z_7^2) \quad \text{(A3)}$$

Subject to,

$$h_1(\vec{z}) = \frac{27}{z_1z_2^2z_3} - 1 \leq 0,$$
$$h_2(\vec{z}) = \frac{397.5}{z_1z_2^2z_3^2} - 1 \leq 0,$$
$$h_3(\vec{z}) = \frac{1.93z_4^3}{z_3z_2z_6^4} - 1 \leq 0,$$
$$h_4(\vec{z}) = \frac{1.93z_5^3}{z_2z_3z_7^4} - 1 \leq 0,$$
$$h_5(\vec{z}) = \frac{((\frac{745z_4}{z_2z_3})^2 + 16.9 \times 10^6)^{1/2}}{110.0z_6^3} - 1 \leq 0,$$

$$h_6(\vec{z}) = \frac{((\frac{745z_5}{z_2z_3})^2 + 157.5 \times 10^6)^{1/2}}{85.0z_7^3} - 1 \leq 0,$$
$$h_7(\vec{z}) = \frac{z_2z_3}{40} - 1 \leq 0,$$
$$h_8(\vec{z}) = \frac{5z_2}{z_1} - 1 \leq 0,$$
$$h_9(\vec{z}) = \frac{z_1}{12z_2} - 1 \leq 0,$$
$$h_{10}(\vec{z}) = \frac{1.5z_6 + 1.9}{z_4} - 1 \leq 0,$$
$$h_{11}(\vec{z}) = \frac{1.1z_7 + 1.9}{z_5} - 1 \leq 0$$

where,

$$2.6 \leq z_1 \leq 3.6,$$
$$0.7 \leq z_2 \leq 0.8,$$
$$17 \leq z_3 \leq 28,$$
$$7.3 \leq z_4 \leq 8.3,$$
$$7.8 \leq z_5 \leq 8.3,$$
$$2.9 \leq z_6 \leq 3.9,$$
$$5.0 \leq z_7 \leq 5.5$$

A.4. The multiple disc clutch brake problem

This problem is minimization problem in which the mass of multiple disc clutch brake should be minimised. In this problem, there are five variables which are $z_1, z_2, \ldots, z_5$, where these variables represent the inner radius, outer radius and thickness of the disc, actuating force and the number of friction surfaces. This problem can be described mathematically as follows.

Minimise the function

$$f(\vec{z}) = \pi(r_0^2 - r_i^2)t(Z+1)p, \tag{A4}$$

Subject to,

$$h_1(\vec{z}) = r_0 - r_i - \triangle r \geq 0,$$
$$h_2(\vec{z}) = l_{\max} - (Z+1)(t+\delta) \geq 0,$$
$$h_3(\vec{z}) = p_{\max} - p_{rz} \geq 0,$$
$$h_4(\vec{z}) = p_{\max}v_{st\ max} - p_r z v_{st} \geq 0,$$
$$h_5(\vec{z}) = v_{st\ max} - v_{st} \geq 0,$$
$$h_6(\vec{z}) = T_{\max} - T \geq 0,$$
$$h_7(\vec{z}) = M_h - sM_s \geq 0,$$
$$h_8(\vec{z}) = T \geq 0,$$

where,

$$\mu = 0.5,\ I_z = 55kg\ mm^2,\ n = 250rpm,\ \triangle r = 20mm,\ M_s = 40Nm,\ p_{\max} = 1MPa,$$
$$T_{\max} = 15s,\ F_{\max} = 1000N,\ s = 1.5,\ M_f = 3Nm,\ v_{sr\ max} = 10$$
$$F_{\max} = 1000,\ l_{\max} = 30mm,\ r_{i\ min} = 60,\ \frac{m}{s},\ r_{o\ min} = 90,\ r_{o\ max} = 110,$$
$$r_{i\ max} = 80,\ t_{\max} = 3,\ F_{\min} = 600,\ Z_{\min} = 2,\ Z_{\max} = 9,\ t_{\min} = 1.5,$$

and all variables are discrete
and have the following values.

$$z_1 = 60, 61, \ldots, 80; \quad z_2 = 90, 91, \ldots, 110; \quad z_3 = 1, 1.5, \ldots, 3;$$
$$z_4 = 600, 610, \ldots, 1000, and\ z_5 = 2, 3, \ldots, 9.$$

A.5. The rolling element bearing problem

Rolling element bearings are critical components of rotating machinery and are confronted with vulnerability due to unsteady loads and speeds, and the corrosive effects of the working environment. In the literature, there are different objective functions for rolling element bearings have been proposed. The most important of these objective functions are representing the requirement of the longest fatigue life *FL* subject to any other applied load *L*. Therefore, the rolling element bearings problem maximum fatigue life and can be represented by the following equation.

$$FL = \left(\frac{C}{L}\right)^n \tag{A5}$$

where n is a constant and C is the dynamic load rating (dynamic capacity).
In this design problem, there are five variables which are diameter of the balls D_b, mean diameter D_m, number of balls N_b, curvature radius coefficient of inner raceway groove$f_i = \frac{r_i}{D_b}$ and curvature radius coefficient of outer raceway groove $f_o = \frac{r_o}{D_b}$, where r_i and r_o represent the inner and outer ring groove curvature radii, respectively. The dynamic load capacity can be used to represent the objective function Changsen (1991) as follows.

Maximise the function

$$F(\vec{z}) = \begin{cases} \max\,(f_c Z^{\frac{2}{3}} D_b^{1.8}), & D_b \le 25.4mm \\ \max\,(3.647 f_c Z^{\frac{2}{3}} D_b^{1.4}), & D_b > 25.4mm \end{cases} \tag{A6}$$

with

$$\vec{z} = (D_b\, D_m\, Z\, f_i\, f_o)$$

$$f_c = 37.91\left(1 + \left(1.04\left[\frac{1-\alpha}{1+\alpha}\right]^{1.72}\left[\frac{f_i(2f_o - l)}{f_o(2f_i - 1)}\right]^{0.41}\right)^{\frac{10}{3}}\right)^{-0.3}$$
$$\left[\frac{\alpha^{0.3}(1-\alpha)^{1.39}}{(1+\alpha)^{\frac{1}{3}}}\right]\left[\frac{2f_i}{2f_i - 1}\right]^{0.41}$$

where $a = D_b cos?(\frac{\beta}{D_m})$ is not an independent parameter, and therefore it does not appear in the vector of design parameters. β is defined as contact angle that depends upon the type of bearing. Subject to,

$$h_1(\vec{z}) = \frac{2\pi - 2cos^{-1}\left[\frac{\left(\frac{D-d}{2} - 3\frac{T}{4})^2 + \left(\frac{D}{2} - \frac{T}{4} - D_b\right)^2 - \left(\frac{d}{2} + \frac{T}{4}\right)^2}{2\left(\frac{D-d}{2} - 3\left(\frac{T}{4}\right)\left(\frac{D}{2} - \frac{T}{4} - D_b\right)\right.}\right]}{2sin^{-1}\frac{D_b}{D_m}} - Z + 1 \ge 0,$$
$$h_2(\vec{z}) = 2D_b - K_{Dmin}(D - d) \ge 00,$$
$$h_3(\vec{z}) = K_{Dmax}(D - d) - 2D_b \ge 0,$$
$$h_4(\vec{z}) = -D_m + (0.5 + e)(D + d) \ge 0,$$
$$h_5(\vec{z}) = D_m - 0.5(D + d) \ge 0,$$
$$h_6(\vec{z}) = f_0 \ge 0.515,$$
$$h_7(\vec{z}) = 0.5(D - D_m - D_b) - \varepsilon D_b \ge 0,$$
$$h_8(\vec{z}) = \zeta B_w - D_b \ge 0,$$
$$h_9(\vec{z}) = f_i \ge 0.515$$

where,

$$\alpha = \frac{D_b}{D_m},\ T = D - d - 2D_b,\ D = 160,\ d = 90,\ B_w = 30,\ r_i = r_0 = 11.033,$$
$$0.5(D + d) \le Dm \le 0.6(D + d),\ 0.15(D - d) \le D_b \le 0.45(D - d),$$

$$4 \le \vec{z} \le 500515 \le f_i \; and \; f_0 \le 0.6, \; 0.4 \le k_{Dmin} \le 0.5,$$
$$0.6 \le k_{Dmax} \le 0.7, \; 0.3 \le \varepsilon \le 0.4, \; 0.2 \le e \le 0.1, \; 0.6 \le \zeta \le 0.85$$

A.6. The three-bar trust design problem

This is a minimization problem. This problem can be described mathematically as follows.

Minimise the function

$$f(\vec{z}) = (2\sqrt{2}z_1 + z_2) \times l, \tag{A7}$$

Subject to,

$$h_1(\vec{z}) = \frac{\sqrt{2}z_1 + z_2}{\sqrt{2}z_1^2 + 2z_1z_2}p - \alpha \le 0,$$
$$h_2(\vec{z}) = \frac{z_2}{\sqrt{2}z_1^2 + 2z_1z_2}p - \alpha \le 0,$$
$$h_3(\vec{z}) = \frac{1}{\sqrt{2}z_2 + z_1}p - \alpha \le 0,$$

where,

$$0 \le z_i \le 1, i = 1, 2,$$
$$l = 100cm, \; p = 2kN/cm^2 \; \alpha = 2kN/cm^2.$$

A.7. Compression spring design problem

This is a minimization problem which contain three main variables are coil diameter (CD), wire diameter (WD) and the number of active coils (NAC) under some restrictions such as minimum deflection, surge frequency and shear stress. This problem can be mathematically formulated as follows.

Minimise the function

$$f(\vec{z}) = (z_3 + 2)z_2z_1^2, \vec{z} = [z_1z_2z_3] = (CD)(WD)(NAC),$$

Subject to,

$$h_1(\vec{z}) = 1 - \frac{z_2^3z_3}{717854z_1^4} \le 0,$$
$$h_2(\vec{z}) = \frac{4z_2^2 - z_1z_2}{12566(z_2z_1^3 - z_1^4)} + \frac{1}{5108z_1^2} - 1 \le 0,$$
$$h_3(\vec{z}) = 1 - \frac{140.45z_1}{z_2^2z_3} \le 0,$$
$$h_4(\vec{z}) = \frac{z_1 + z_2}{1.5} - 1 \le 0$$

where,

$$0.05 \le z_1 \le 2,$$
$$0.25 \le z_2 \le 1.30,$$
$$2 \le z_3 \le 15$$