Noname manuscript No.
(will be inserted by the editor)

Spiral Water Cycle Algorithm for Solving Multi-Objective Optimization and Truss Optimization problems

Heba F. Eid · **Laura Garcia-Hernandez** · **Ajith Abraham**

Received: date / Accepted: date

Abstract —- This paper addresses multi-objective optimization and the Truss Optimization Problem employing a novel meta-heuristic that is based on the real-world water cycle behaviour in rivers, rainfalls, streams, etc. This meta-heuristic is called Multi-Objective Water Cycle Algorithm (MOWCA) which is receiving great attention from researchers due to the good performance in handling optimization problems in different fields. Additionally, the hyperbolic spiral movement is integrated into the basic MOWCA to guide the agents throughout the search space. Consequently, under this hyperbolic spiral movement, the exploitation ability of the proposed MOSWCA is promoted. To assess the robustness and coherence of the MOSWCA, the performance of the proposed MOSWCA is analysed on some multi-objective optimisation benchmark functions; and three truss structure optimization problems. The results obtained by the MOSWCA of all test problems were compared with various multi-objective meta-heuristic algorithms reported in the literature. From the empirical results, it is evident that the suggested approach reaches an excellent performance when solving multi-objective optimization and the Truss Optimization Problems.

Keywords Water Cycle Algorithm · Multi-Objective Optimization · Truss Optimization

Heba F. Eid
Faculty of Science, Al-Azhar University,Cairo, Egypt
E-mail: eid.heba@gmail.com,heba.fathy@azhar.edu.eg

Laura Garcia
Area of Project Engineering, University of Cordoba, Spain
E-mail: ir1gahel@uco.es

Ajith Abraham
Machine Intelligence Research Labs, MIR Labs,USA
E-mail: abraham.ajith@gmail.com

1 Introduction

Most of the real-world problems in engineering and other research fields are dynamic optimization problems, which are modeled by a set of nonlinear equations. To simulate real-world engineering problems, several objectives need to be investigated simultaneously for obtaining the optimum design. In recent decades, solving real-world engineering problems via multi-objective optimization algorithms has become an attractive research area for many researchers and scientists [1] [2] [3] [4].

Multi-objective optimization deals with finding solutions for problems with more than single objective [5] [6]. Those several objectives which are to be optimized simultaneously. The main challenge in multi-objective optimization is the proper addressing of multiple objectives, which often have conflicting nature. Generally, a multi-objective optimization problem can be stated as:

$$\begin{array}{ll} \text{min / max:} & F(x) = \{f_1(x), f_2(x), \dots, f_m(x)\} \\ \text{subject to:} & g_i(x) \leq 0, \quad i = 1, 2, \dots, k \\ & h_i(x) = 0, \quad i = 1, 2, \dots, p \\ & LB \leq x_i \leq UB, \quad i = 1, 2, \dots, n \end{array} \tag{1}$$

where F is an m-dimensional objective vector; x is an n-dimensional decision vector; g and h represent the inequality and equality constraints respectively; and [LB, UB] are the boundaries of the i^{th} variable.

In multi-objective optimization problems, since the objectives are usually conflicting, there does not exist one global solution which optimizes all the objectives simultaneously. Alternately, there exists a set of tradeoff solutions which is defined as Pareto optimal solutions or non-dominated solutions, Pareto dominance, and Pareto front [7] [8] [9] [10] [11] . Whereby, the main goal of the multi-objective optimization is to find as many of non-dominated solutions as possible.Recently, a significant number of multi-objective optimizers have been developed in the literature trying to solve multi-objective optimization problems such as : Nondominated Sorting Genetic Algorithm [12], Nondominated Sorting Genetic Algorithm version 2 (NSGA-II) [13], MultiObjective Particle Swarm Optimization (MOPSO) [14],Pareto archive evolution strategy (PAES) [15], Multi-objective Harmony Search (MOHS) [16], Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [17] and Multi-objective water cycle algorithm (MOWCA) [18].

Presently, engineering structural optimization has attracted substantial attention [19] [20] [21].Truss design optimization is one of the main problems in the field of structural engineering [22]. Truss design structures are carried out in such a way that their topology, sizes and shape are optimized simultaneously [23] [24]. Truss optimization is a multi-objective optimization problem; for which, the main objective function is to minimize the weight of the structure (mass), subject to some constraints on stresses, frequencies and displacements. Meanwhile, several types of research have considered structural strength. For which, structural compliance can be used to measure structural reliability and strength [25] [26] [27].

In this paper, a multi-objective spiral water cycle algorithm (MOSWCA) is proposed. For which a hyperbolic spiral movement is integrated into the basic MOWCA to guide the agents throughout the search space. Consequently, under this hyperbolic spiral movement, the exploitation ability of the proposed MOSWCA is promoted. To assess the robustness and coherence of the MOSWCA, the performance of the proposed MOSWCA is investigated on some multi-objective optimisation benchmark functions; and three truss structure optimization problems. The results obtained by the MOSWCA of all test problems were compared with various multi-objective meta-heuristic algorithms reported in the literature. The rest of the paper is organized as follows: In section 2, a brief description of the multi-objective water cycle algorithm (MOWCA) is presented. In section 3, the proposed MOSWCA is elaborated. In section 4, the efficiency of the proposed MOSWCA algorithm investigated two well known multi-objective optimisation benchmark. Three Truss optimization test problems are provided and the comparative study of MOSWCA against various state-of-the-art optimization algorithms are presented in section 5 and 6, respectively. Finally, in section 7 the main findings of this study are discussed.

2 Water Cycle Algorithm

WCA is a meta-heuristic algorithm which is derived by the observation of the water cycle process in nature and proposed by Eskandar et al [28]. WCA simulates the flow of streams and rivers, rainfall, confluence, and evaporation.

For which, an initial population of variables is randomly generated by the rainfall process. Then, the initial population is divided in terms of having the least cost into three grades; sea (best solution), river (near to the current best) and stream.

$$Totalpopulation = \begin{bmatrix} \text{Sea} \\ \text{River }_1 \\ \vdots \\ \text{Stream }_{N_{\mathrm{sr}+1}} \\ \vdots \\ \text{Stream }_{N_{\mathrm{pop}}} \end{bmatrix} \tag{2}$$

where N and N_{pop} are the number of design variables (problem dimension) and the total number of the population respectively.

$$N_{\mathrm{sr}} = NumberOfRivers + 1 \tag{3}$$

$$N_{\mathrm{streams}} = N_{\mathrm{pop}} - N_{\mathrm{sr}} \tag{4}$$

N_{sr} represents the total number of sea and rivers; and N_{streams} indicates the number of streams which indirectly or directly flow to sea and rivers.

The cost of a raindrop is attain by the evaluation of the cost function

$$\text{cost}_i = f\left(x_1^i, x_2^i, \ldots, x_{N_{\text{rs}}}^i\right) \quad i = 1, 2, 3, \ldots, N_{\text{pop}} \tag{5}$$

In order to simulate the flow of the streams to the rivers, and the streams and the rivers to the sea in nature, WCA uses the following position updating equation:

$$X_{\text{Stream}}^{i+1} = X_{\text{Stream}}^{i} + \text{ rand } \times C \times \left(X_{\text{River}}^{i} - X_{\text{Stream}}^{i}\right) \tag{6}$$

$$X_{\text{Stream}}^{i+1} = X_{\text{Stream}}^{i} + \text{rand} \times C \times \left(X_{\text{Sea}}^{i} - X_{\text{Stream}}^{i}\right) \tag{7}$$

$$X_{\text{River}}^{i+1} = X_{\text{River}}^{i} + \text{rand} \times C \times \left(X_{\text{Sea}}^{i} - X_{\text{Rirer}}^{i}\right) \tag{8}$$

where rand is a uniformly distributed random number within the range of $[0, 1]$ and C is a constant value between 1 and 2.

One of the most important characteristics of the meta-heuristic algorithms is randomization. In WCA, to increase randomization, the raining and evaporation process are considered.

Raining and evaporation take place when the distance between a river or any stream and the sea is less than parameter $d_{\max}$

$$\left|X_{\text{Sea}}^{i} - X_{\text{River}}^{i}\right| < d_{\max} \quad i = 1, 2, 3, \ldots, N_{\text{sr}} - 1 \ (9)$$

A large value for $d_{\max}$ reduces the search and leads to focus more on exploration, while for exploitation a small $d_{\max}$ value motivate the search intensity near the sea. Therefore, to make a proper trade-off between exploitation and exploration in the WCA, the value of $d_{\max}$ adaptively decreases linearly using the following equation:

$$d_{\max}^{i+1} = d_{\max}^{i} - \frac{d_{\max}^{i}}{\text{max iteration}} \tag{10}$$

After fulfilling the evaporation condition, the raining procedure is performed. In the raining procedure, the new solutions (scattered streams) are generate by the formula:

$$X_{\text{Stream}}^{\text{new}} = LB + \text{ rand } \times (UB - LB) \tag{11}$$

where UB and LB are the upper and lower bounds of the given problem, respectively.

3 Proposed Multi-Objective Spiral WCA (MOSWCA)

For handling multi-objective optimization problems, the WCA is equipped with multi-objective operators; such as archiving the mechanism based on non-dominated sorting and crowing distance operator [29]. Developing an efficient archive is a vital step for MOSWOA. Hence, it affects the convergence capability of MOSWOA and maintains a good spread of non-dominated solutions. Therefore, over the course of iterations, the crowding-distance for all non-dominated solutions are calculated; and archive of non-dominated solutions with higher crowding-distance values are preserved. Afterwards, the preserved values which considered to be good representatives of the entire Pareto front sets are designated as the sea and rivers. The archive is updated at each iteration, and any dominated solutions are eliminated from the archive. Moreover, the number of non-dominated solutions may be overloaded; therefore, the crowding distance operator is applied again to eliminate the overload non-dominated solutions having the lowest crowding distance values among the Pareto archive members.

One of the main operators of the MOSWCA is the position updating process. The proposed MOSWCA aims to employ a hyperbolic spiral to simulate the flow of the streams to the rivers, and the streams and the rivers to the sea. The hyperbolic spiral interpreted as polar coordinates by: $r = a/\theta,\ \theta \neq 0$, where, r is the radius and θ is the azimuthal angle in a polar coordinate system, while, a is a real number constant. Figure 1 illustrates the hyperbolic spiral, whereby, as θ increases, the spiral winds around the origin moves closer to it. Subsequently, the hyperbolic spiral updating procedure of the proposed MOSWCA allows the streams and rivers to update their position anywhere around the sea; which, increases the exploitation ability of the MOSWCA. The hyperbolic spiral updating position of the proposed MOSWCA is given by the following equations:

$$X^{i+1}_{\text{Stream}} = X^{i}_{\text{Stream}} + |X^{i}_{\text{River}} - X^{i}_{\text{Stream}}| \cdot \frac{\cos(2\pi l)}{1} \tag{12}$$

$$X^{i+1}_{\text{Stream}} = X^{i}_{\text{Stream}} + |X^{i}_{\text{Sea}} - X^{i}_{\text{Stream}}| \cdot \frac{\cos(2\pi l)}{1} \tag{13}$$

$$X^{i+1}_{\text{River}} = X^{i}_{\text{River}} + |X^{i}_{\text{Sea}} - X^{i}_{\text{Rirer}}| \cdot \frac{\cos(2\pi l)}{1} \tag{14}$$

The parameter l is a random uniform number in the range [-1, 1], where, the values of parameter l are reduced over the course of iterations.

$$l = (a-1) \times \text{rand} + 1 \tag{15}$$

$$a = -1 - \left(\frac{t}{\max iteration}\right) \tag{16}$$

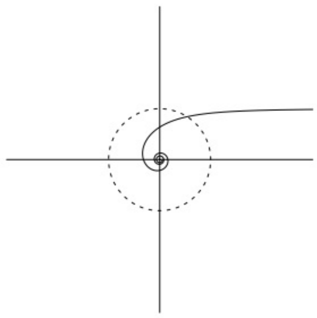

Fig. 1 The hyperbolic spiral

The pseudo code of the proposed MOSWCA algorithm is presented in Algorithm 1

4 Empirical Evaluation

With the aim of investigating the performance and capabilities of the proposed MOSWCA algorithms; Matlab R2015b was used for implementation purposes. All of the experiments were carried out on Intel(R), Core i7- 4910MQ CPU 2.90GHz and 16GB RAM.

4.1 MOSWCA statistical analysis

In order to assess the performance of the proposed MOSWCA, variety multi-objective test problems from the Zitzler-Deb-Thiele (ZDT) [30] and Deb, Thiele, Laumanns and Zitzler (DTLZ) [29] benchmark are solved within various runs.

ZDT benchmark considers problems with bi-objective, which is the most common usage of Pareto optimization, especially in real engineering applications. Its main focus is on the convergence of the obtained solutions towards the Pareto front. Whereby, the characteristics of ZDT problems are : "ZDT1 is convex, ZDT3 is nonconvex and disconnected, ZDT4 is convex and multimodal, and ZDT6 is nonconvex and non-uniform distribution".

While the DTLZ benchmark is scalable for any number of objectives. Scalability is a desirable property which makes DTLZ test functions suitable for testing the optimization capability of the proposed MOSWCA. For all DTLZ problems, the Pareto front is located in the first orthant of the objective space and its shape is either a curve, a sphere, or a simplex. For this work, two cases are considered for DTLZ problems, bi-objective and tri-objective functions. While, the characteristics of DTLZ problems are : "DTLZ1 is convex, DTLZ4 is nonconvex, and DTLZ7 is disconnected".

Algorithm 1 Pseudocode of MOSWCA Algorithm

Input:
Population total number N_{pop}
total number of sea and rivers N_{sr}
Number of optimization iterations Max_Iter, archive size M and d_{max}
Output:
MOSWCA Pareto front

Initialize the MOSWCA population positions randomly.
for i=1:N_{pop} **do**
  Create streams X_{stream}
  Calculate multi-objective costs $f(X_{stream})$
end for
Find the non-dominated solutions among the feasible solutions and initialized the Pareto archive A with them
Calculate the crowding-distance for each member $\in$ A
sort A members
Sea $\leftarrow A(firstmember) Rivers \leftarrow A$(N$_{sr}$ - 1 members)
Stream $\leftarrow A$(N$_{pop}$ - N_{sr} members)
while $t \leq Max\_Iter$ **do**
  for i=1:N_{pop} **do**
    Update the position of X_{stream} using eq. 12 and 13
    Calculate generated stream multi-objective costs $f(X_{stream})$
    if $f(X_{stream}) < f(X_{river})$ **then**
      River.position= new stream.position
      if $f(X_{stream}) < f(X_{sea})$ **then**
        Sea.position= new stream.position
      end if
    end if
  end for
  for i=1:$N_{sr} - 1$ **do**
    Update the position of X_{river} using eq. 14
    Calculate generated river multi-objective cost $f(X_{river})$
    if $f(X_{river}) < f(X_{sea})$ **then**
      Sea.position= River.position
    end if
  end for
  for i=1:$N_{sr} - 1$ **do**
    if $|River - Sea| < d_{max}$ or $rand < 0.1$ **then**
      Generate new X_{stream} using eq. 11
    end if
  end for
  Decrease d_{max} using eq. 10
  Find the new non-dominated solutions among the feasible solutions
  Update A with respect to the obtained non-dominated solutions
  if $A > M$ **then**
    Calculate the crowding-distance value for A members
    Maintain A with M lowest crowding-distance value
  else
    Calculate the crowding-distance value for A members and select new Sea and Rivers
  end if
  t=t+1
end while
return X_{sea}

For comparing the MOWCA and MOSWCA algorithms two performance criteria are considered: Generational Distance (GD) [31] and Inverted Generational Distance (IGD) [32] [33]. GD is a quantifying measure of convergence between the optimal Pareto front and obtained Pareto front. The mathematical representation of the GD performance metric is as follows:

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{17}$$

Where, n is the number of obtained Pareto front and d_i indicates the Euclidean distance between the i^{th} obtained optimal Pareto front and the nearest true optimal Pareto front in the reference set.

IGD measures the approximate distance from the Pareto front to the solution set in the objective space. IGD don't miss any part of the true Pareto set on comparison, the mathematical formula of IGD is given by:

$$IGD = \frac{\sqrt{\sum_{i=1}^{n'} (d'_i)^2}}{n'} \tag{18}$$

Where, n' is the number of true optimal Pareto solutions and d' represented the Euclidean distance between the i^{th} true optimal Pareto solution and the nearest obtained optimal Pareto solution in the reference set.

The internal parameters of the MOWCA and MOSWCA are chosen as: population size (Npop) was 50 and the total number of iterations (max_it) was 100, Nsr = 4 and dmax = 1.0E-16 on all of the simulations, while the size of Pareto archive (number of nondominated solutions) is set to be equal to Npop. To make an unbiased comparison, the results are obtained over 30 independent runs on each test function; with entirely random initial conditions. Table 1 shows the statistical optimization results of MOWCA against proposed MOSWCA based on GD measures including the best solution (Min), average solution (Av.), worst solution (Max), and standard deviation (std).

From the results in Table 1, the proposed MOSWCA outperformed the MOWCA algorithm by having the smallest value of GD in term of "min, max and av." results for the test functions. Moreover, the proposed MOSWCA was able to find the Pareto optimal with less std. than MOWCA for all test functions, which indicates the robustness of MOSWCA.

To further verify the proposed MOSWCA, more comparisons have been carried out using NSGA-II, [13], MOEA/D [17], Multi-Objective Multi-Verse Optimizer (MOMVO) [34], MultiObjective Particle Swarm Optimization (MOPSO) [14] and Multi-objective Whale Optimization Algorithm (MOWOA) [35]. The obtained optimization GD and IGD results are given in table 2 and table3, respectively. The best attained statistical results (average and standard deviation) are highlighted in bold. Looking at Table 2, the proposed MOSWCA surpassed the other compared algorithm for seven test functions in term of GD measure. While, compared to the MOMVO algorithm, MOSWCA obtained the second best results for test functions ZDT1 and DTLZ 2. Moreover, from the results reported in table3; the MOSWCA is clearly superior to all compared optimizers on all multi-objective problems in term of IGD measure; except for DTLZ 4 problem; where it shows the second best after MOMVO. Consequently, the results shown in table 2 and 3 validate that the proposed MOSWCA is an effective algorithm for solving challenging multi-objective problems.

Table 1 GD Statistical results of MOWCA and MOSWCA algorithm

Problem	MOWCA				MOSWCA			
	Av.	Min	Max	Std	Av.	Min	Max	Std
M=2								
ZDT1	4.13e-3	2.68e-3	5.69e-3	7.91e-4	3.3928e-3	1.1277e-3	3.9476e-3	3.9746e-4
ZDT3	1.30e-3	7.39e-4	1.81e-3	3.43e-4	1.0027e-3	1.4297e-4	1.1303e-3	2.8496e-4
ZDT4	2.19e-3	1.12e-3	5.23e-3	1.32e-4	1.8923e-3	1.0123e-3	4.621e-3	1.317e-5
ZDT6	1.03e-2	3.23e-4	0.018	1.01e-2	1.0136e-4	7.1478e-5	2.3941e-3	8.9343e-3
DTLZ 2	1.78e-4	1.65e-4	1.87e-4	6.70e-6	1.6558e-4	1.0251e-4	1.8917e-4	7.0951e-7
DTLZ 4	2.06e-4	1.76e-4	2.57e-4	2.56e-5	1.6378e-4	1.1558e-4	4.0566e-4	1.6708e-6
DTLZ 7	2.65e-5	1.27e-5	4.99e-5	1.62e-5	2.4445e-5	1.2705e-5	5.8618e-5	1.7838e-9
M=3								
DTLZ 2	1.93e-3	1.39e-3	2.25e-3	2.87e-4	7.2617e-4	4.0393e-4	6.4955e-3	2.3188e-4
DTLZ 4	1.79e-3	4.69e-4	3.66e-3	9.57e-4	1.205e-3	1.0594e-3	3.6524e-3	1.3323e-4
DTLZ 7	2.65e-3	3.31e-4	1.81e-2	5.47e-3	1.3801e-3	4.7418e-4	1.8618e-2	4.2705e-4

Table 2 GD Comparison results obtained for MOSWCA and different multi-objective optimization algorithms

Problem	MOSWCA		NSGA-II		MOEA/D		MOMVO		MOPSO	
	Av.	std	Av.	Std	Av.	Std	Av.	Std	Av.	Std
M=2										
ZDT1	3.3928e-3	3.9746e-4	0.503569	0.052127	0.00413	0.00008	**1.73e-3**	**0.00005**	0.756200	0.145703
ZDT3	**1.0027e-3**	**2.8496e-4**	0.502427	0.047587	0.005129	0.00011	5.076e-3	0.00009	0.794325	0.070546
ZDT4	**1.8923e-3**	**1.317e-5**	0.485384	0.052186	0.003971	0.00009	2.463e-3	0.00006	1.081437	0.195925
ZDT6	**1.0136e-4**	8.9343e-3	2.151897	2.285011	0.003322	0.00024	6.52e-4	**0.00002**	3.707637	0.849501
DTLZ 2	**1.6558e-4**	7.0951e-7	0.006888	0.000576	0.00035	2.8e-11	2.6 e-4	**4.8e-13**	8.7 e-3	0.0037
DTLZ 4	**1.6378e-4**	**1.6708e-6**	0.0435	2.53e-04	0.0258	6.66e-4	0.0019	4.32e-5	0.0053	0.0042
DTLZ 7	2.4445e-5	1.7838e-9	0.008157	0.000749	N/A	N/A	7.98 e-3	5.34e-8	N/A	N/A
M=3										
DTLZ 2	7.2617e-4	2.3188e-4	0.056743	0.006245	0.00096	8.7e-11	**5.6 e-4**	**4.6e-12**	N/A	N/A
DTLZ 4	**1.205e-3**	**1.3323e-4**	0.029916	0.027569	0.0457	1.06e-3	1.85e-2	2.14e-4	N/A	N/A
DTLZ 7	**1.3801e-3**	4.2705e-4	0.059228	0.021574	N/A	N/A	1.8e-03	**8.17e-5**	N/A	N/A

Table 3 IGD Comparison results obtained for MOSWCA and different multi-objective optimization algorithms

Problem	MOSWCA		NSGA-II		MOEA/D		MOWOA		MOMVO	
	Av.	std	Av.	Std	Av.	Std	Av.	Std	Av.	Std
M=2										
ZDT1	**4.871e-4**	8.6233e-4	4.8419e-3	1.73e-4	3.9754e-3	**1.34e-5**	5.2523e-3	3.58e-4	5.04e-4	6.0e-5
ZDT3	**1.1329e-3**	**1.3194e-5**	0.15447	9.07e-2	0.22553	8.04e-2	5.9617e-3	2.62e-4	9.057e-3	6.0e-5
ZDT4	**5.3125e-4**	7.317e-3	0.23946	0.185	0.48762	0.226	4.6998e-3	2.06e-4	4.13e-4	**4.0e-5**
ZDT6	**3.7691e-5**	0.0080768	9.4342e-2	5.07e-2	8.4125e-2	4.60e-2	4.3985e-3	3.32e-4	4.05e-4	**4.0e-5**
DTLZ 2	**4.0977e-5**	0.00028952	1.8e-04	6.7e-11	1.7e-04	1.3e-15	N/A	N/A	1.5e-4	**2.2e-16**
DTLZ 4	**1.1373e-4**	0.0020657	0.02000	1.41e-4	0.03373	1.57e-4	N/A	N/A	1.8e-4	**2.50e-05**
M=3										
DTLZ 2	**2.0771e-4**	0.0015103	6.8912e-2	2.87e-3	5.4861e-2	1.68e-4	7.4635e-2	3.73e-3	7.5e-4	**3.4e-12**
DTLZ 4	3.7107e-4	0.0011265	1.1785e-1	1.34e-1	4.7370e-1	3.41e-1	7.0870e-2	2.33e-3	**1.9e-4**	**3.77e-06**
DTLZ 7	**1.031e-3**	**1.0572e-4**	1.0219e-1	4.94e-2	1.7481e-1	1.21e-1	8.2654e-2	5.57e-3	1.1e-3	1.8e-4

5 Truss optimization test problems

The performance of the proposed MOSWCA is examined by the various benchmark of truss structure problems. Three truss structure problems were used, the problems are named according to the numbers of ground elements of the trusses as 10-bar, 25-bar and 200-bar problems. The 10-bar and 25-bar, being of a small scale to discuss the accuracy of the obtained optimal solutions; while the 200-bar, a larger structure, is used to demonstrate the proposed algorithm efficiency.

The bi-objective problem for trusses optimization is design as follow:

$$\begin{aligned} &\text{min:} \quad \{f_1(x), f_2(x)\} \\ &\text{subject to} \quad |\sigma_{\max}| \leq \sigma_{allow} \\ &\qquad\qquad\quad u_{\max} \leq u_{allow} \end{aligned} \tag{19}$$

Where $f_1(x)$ is structural mass function of the truss, $f_2(x)$ is structural compliance function, $\sigma_{\max}$ and $\sigma_{\ allow}$ are the maximum stress and allowable stress respectively. While $u_{\max}$ and $u_{\ allow}$ are the maximum nodal and allowable nodal displacement of the truss structure due to applied forces respectively.

5.1 Two-dimensions 10 bar Truss problem

The 2-D 10-bar truss structure is shown in Figure 2. The mechanical properties of the 10-bar is given by: "the modulus of elasticity $E = 68.95GPa$, the density material $\rho = 2,768kg/m^3$ and the allowable stresses $\sigma = 172MPa$. While the lower and upper bounds for all cross-sectional areas are 0.65 and 225.75 cm2 respectively".

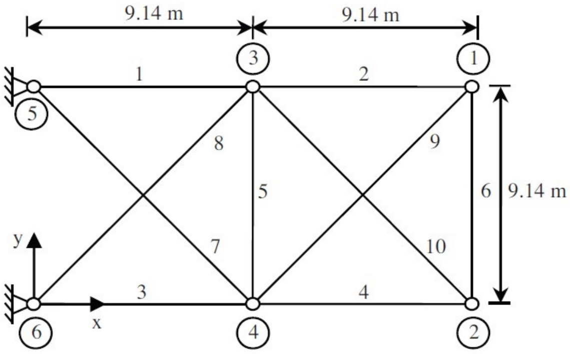

Fig. 2 2-D 10-bar Truss structure

5.2 Three-dimensions 25 bar Truss problem

The 3-D 25-bar truss structure is illustrated in Figure 3. The structure material properties are: "$E = 68.95GPa$ and $\rho = 2.768kg/m3$. Meanwhile, the Cross-sectional areas of members are to be adopted among the range from $6.5e5m^2$ to $2.26e2m^2$ and the allowable stress $\sigma = 275.8MPa$".

5.3 Two-dimensions 200 bar Truss problem

The 2-D 200-bar truss structure is shown in Figure 4. The 200-bar truss contains 77 nodes. The 200-bar material parameters are as follow: "material density and modulus of elasticity are $\rho = 7,833kg/m^3$ and $E = 206.9GPa$, respectively. The lower bound of cross-sectional areas is $0.645cm^2$ and the upper bound is $132.73cm^2$. While stress limitations of 68.95 MPa are adopted for the truss structure members.

6 Results and Discussion

In this section, 30 independent runs of the 10-bar, 25-bar and 200-bar truss bi-objective optimization problems were performed. MOSWCA algorithm parameters values for solving the truss optimization problems were kept the same as follows: Npop=50, Nsr=4 , dmax = 1e-16, the maximum number of iterations= 100 and archive size = Npop.

The performance assessment of the results obtained by the proposed MOSWCA are carried out by using the hypervolume indicator (HV). The HV is able to measures both front advancement and extension simultaneously; whereas the higher value of HV indicates the better Pareto front. Moreover, the non-dominated fronts obtained from the proposed MOSWCA are compared to various established and recent optimizers including non dominated sorting genetic algorithm II (NSGA-II) [13], Multi-Objective Evolutionary Algorithm

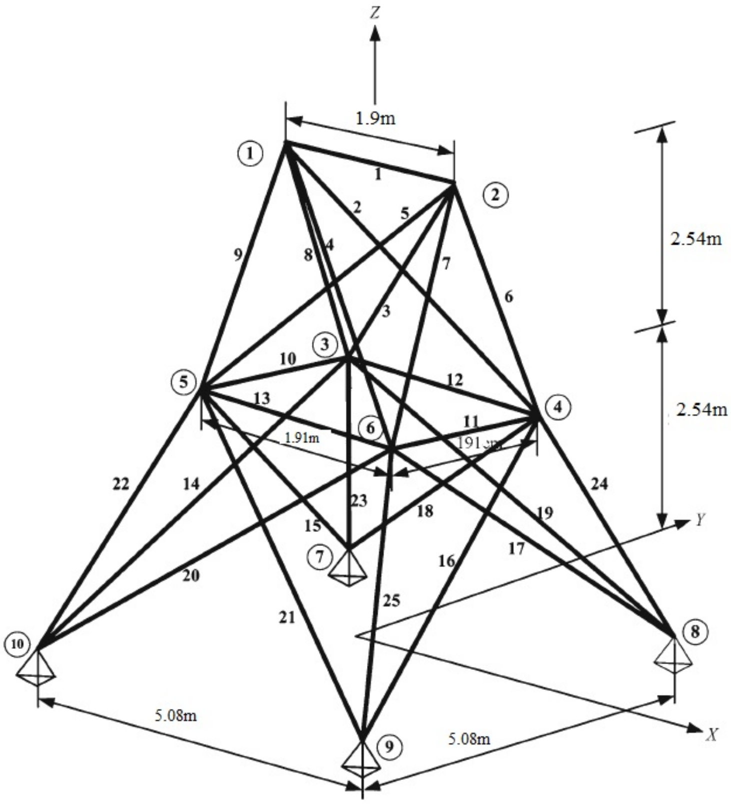

Fig. 3 3-D 25-bar Truss structure

based on Decomposition (MOEA/D) [17], Multi-Objective Dragonfly Algorithm (MODA) [36] and MOWCA. The initial parameters of NSGA-II, MOEA/D and MODA are identical to the values reported in their original papers cited above.

The comparative HV results, mean and standard deviation, are reported in Table 4. Based on table 4, for the 10-bar truss problem, the proposed MOSWCA provides the highest mean hypervolume, which indicates that MOSWCA obtains a Pareto front with better front extension. The second best algorithm is NSGA-II, while the worst is MOWCA. Moreover, MOSWCA obtained the less standard deviation value; which indicates the algorithm consistency. Likewise, for 25-bar truss problem, the proposed MOSWCA gives the best mean value, while the second-best still is NSGA-II and the worst is MOWCA. In addition, proposed MOSWCA is the most consistent algorithm based on its standard deviation value. For the 200-bar truss problem, the best algorithm is the proposed MOSWCA and the second best is MOWCA according to the mean HV, whereas the worst is MODA. Nevertheless, MOEA/D obtains the

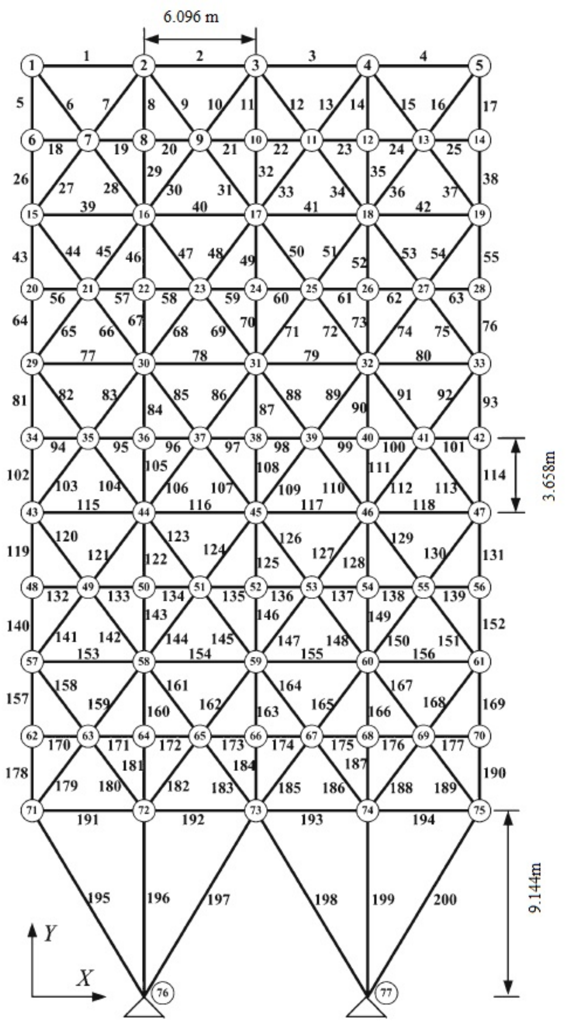

Fig. 4 2-D 200-bar Truss structure

best standard deviation value, while MOSWCA obtains the third best. The Pareto fronts comparison of the best runs provided by the five optimizers for 10-bar, 25-bar and 200-bar truss problems are shown in Fig 5-7. From the figures, it is clear that the proposed MOSWCA dominates other algorithms.

7 Conclusion

This article has employed a recent meta-heuristic called Multi-Objective Water Cycle Algorithm (MOWCA) for addressing multi-objective optimization

Table 4 Hypervolume Statistical results of the truss optimization problems

Algorithm		10-bar	25-bar	200-bar
NSGA-II	mean	5.1556e8	2.6201e8	1.1545e10
	Std	1.9359e7	3.3619e7	1.4968e8
MOEA/D	mean	3.5143e8	2.0815e8	5.8565e9
	Std	2.2596e7	2.9949e6	**5.4340e7**
MODA	mean	3.5202e8	2.2629e8	3.5795e9
	Std	1.0239e7	5.1117e6	1.0423e8
MOWCA	mean	2.3588e8	1.7289e8	1.1978e10
	Std	3.3475e7	2.4759e7	3.0614e8
MOSWCA	mean	**5.4766e8**	**3.3225e8**	**2.0706e10**
	Std	**2.8862e6**	**7.6102e5**	1.2700e8

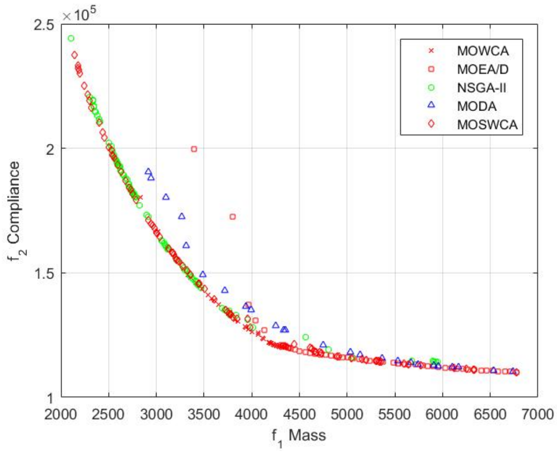

Fig. 5 10-bar space truss Pareto front

and the Truss Optimization Problem. This meta-heuristic simulates the real-world behaviour of the water cycle process in rivers, rainfalls, streams, etc. Besides, the hyperbolic spiral movement is integrated into the basic MOWCA to guide the agents throughout the search space. Consequently, under this hyperbolic spiral movement, the exploitation ability of the proposed MOSWCA is promoted. To assess the robustness and coherence of the MOSWCA, the performance of the proposed MOSWCA has been analysed on some multi-objective optimization benchmark functions; and three truss structure optimization problems. The results obtained by the MOSWCA of all test problems were compared with alternative multi-objective meta-heuristic algorithms reported in the literature. From the results and comparison it is evident that the suggested approach reaches an excellent performance when solving multi-objective optimization and the Truss Optimization Problem. As future work, an enhanced version of the MOWCA can be applied and studied to multi-

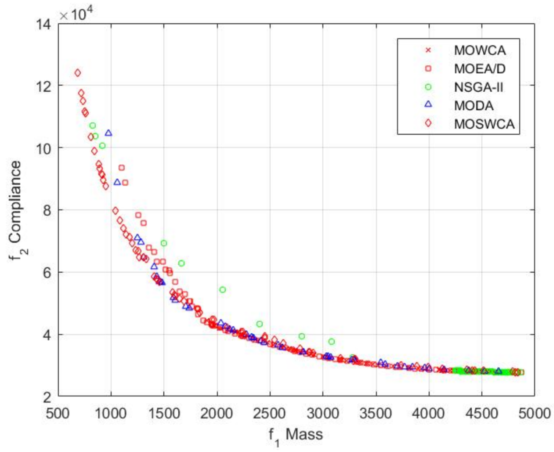

Fig. 6 25-bar space truss Pareto front

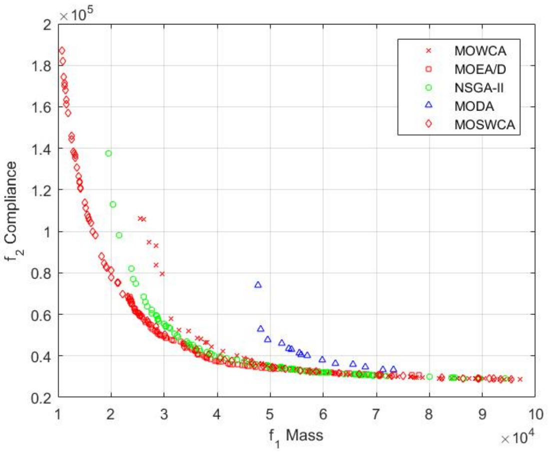

Fig. 7 200-bar space truss Pareto front

objective optimization and the Truss Optimization Problem by combining MOWCA with an additional optimization method.

References

1. S. Mirjalili, P. Jangir, S. Saremi, Applied Intelligent **46**, 79–95 (2017)
2. A.N. nal, G. Kayakutlu, Complex & Intelligent Systems **2020**. URL https://doi.org/10.1007/s40747-020-00159-y
3. Kaveh, K. Laknejadi, IJST, Transactions of Civil Engineering **35**, 137 (2011)
4. S. Gholizadeh, H. Asadi, A. Baghchevan, Int. J. Optim. Civil Eng **4**(3), 415 (2014)
5. K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms* (John Wiley and Sons, Dordrecht, 2001)

6. C. Cac, G.B. Lamont, V.V. DA, *Evolutionary algorithms for solving multi-objective problems* (Springer, 2007)
7. D. Yagyasen, M. Darbari, P.K. Shukla, V.K. Singh, IERI Procedia **5**, 81–86 (2013)
8. P.A.N. Bosman, D. Thierens, IEEE Trans Evol Comput **7**, 174–188 (2003)
9. S. Jiang, S. Yang, (International Conference on Parallel Problem Solving from Nature, Cham, 2016), p. 984–993
10. P. Ngatchou, A. Zarei, A. El-Sharkawi, in *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems* (IEEE, 2005), p. 84–91
11. Y. Hua, Y. Jin, H.K. et al., Complex & Intelligent Systems **6**, 275–285 (2020)
12. N. Srinivas, K. Deb, Evol Comput **2**, 221 (1994)
13. K. Deb, A. Pratap, A. S, M. T, IEEE Trans Evol Comput **6**, 182 (2002)
14. C.A.C. Coello, G.T. Pulido, M.S. Lechuga, IEEE Trans Evol Comput **8**, 256 (2004)
15. J.D. Knowles, C. DW., Evol Comput **8**, 149–72 (2000)
16. F. Mohamed, E. Santawy, A.A. N, Computing Science **1**, 9 (2012)
17. Q. Zhang, H. Li, IEEE Trans Evol Comput **11**, 712 (2007)
18. A. Sadollah, H. Eskandar, A. Bahreininejad, et al., Soft Comput **19**, 2587–2603 (2015)
19. X. Guo, G.D. Cheng, Acta Mech Sin **20**, 526–533 (2004)
20. W.H. Zhang, G.M. Dai, F.W. Wang, S.P. Sun, H. Bassir, Acta Mech Sin **2**, 77–89 (2007)
21. Y.K. Sui, J.Z. Du, Y.Q. Guo, Acta Mech Sin **22**, 611–619 (2006)
22. P. Chadchotsak, KKU Engineering Journal **82**, 32 (2005)
23. N. Noiluplao, B. S., Hindawi Publishing Corporation Mathematical Problems in Engineering pp. 1–9 (2013)
24. N. Pholdee, S. Bureerat, Computers and Structures **106**, 115 (2012)
25. D. Greiner, P. Hajela, Struct Multidisc Optim **45**, 589 (2012)
26. T. Techasen, K. Wansasueb, N. Panagant, et al., Engineering with Computers **35**, 721–740 (2019)
27. V. Ho-Huu, D. Duong-Gia, V.D.T. et al, Expert Syst Appl **102**, 262 –272 (2018)
28. H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Computers and Structures **111**, 151 (2012)
29. K. Deb, L. Thiele, M. Laumanns, E. Zitzler, in *In: Evolutionary Multiobjective Optimization, Advanced Information and Knowledge Processing* (Springer, London, 2005), p. 105–145
30. E. Zitzler, K. Deb, L. Thiele, Evol. Comput **8**, 173 (2000)
31. D.A.V. Veldhuizen, G.B. Lamont, Citeseer **1998**
32. M.R. Sierra, C.A.C. Coello, in *Third Int. Conf. on Evolutionary Multi-Criterion Optimization* (Guanajuato, 2005), p. 505–519
33. C.A.C. Coello, M. Reyes Sierraa, in *In: MICAI 2004: Advances in Artificial Intelligence* (Springer, Berlin, Heidelberg ,, 2004), p. 688–697
34. S. Mirjalili, P. Jangir, S.Z. Mirjalili, S. Saremi, I.N. Trivedi, *Optimization of problems with multiple objectives using the multi-verse optimization algorithm* (Knowledge-based Systems, 2017). DOI http://dx.doi.org/10.1016/j.knosys.2017.07.018
35. F.A. Siddiqi, M. Rahman, Springer pp. 431–446 (2020)
36. S. Mirjalili, Neural Computing & Application **27**, 1053–1073 (2016)