

Water cycle algorithm for solving multi-objective optimization problems

Ali Sadollah · Hadi Eskandar · Ardeshir Bahreininejad · Joong Hoon Kim

Published online: 6 September 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract In this paper, the water cycle algorithm (WCA), a recently developed metaheuristic method is proposed for solving multi-objective optimization problems (MOPs). The fundamental concept of the WCA is inspired by the observation of water cycle process, and movement of rivers and streams to the sea in the real world. Several benchmark functions have been used to evaluate the performance of the WCA optimizer for the MOPs. The obtained optimization results based on the considered test functions and comparisons with other well-known methods illustrate and clarify the robustness and efficiency of the WCA and its exploratory capability for solving the MOPs.

Keywords Multi-objective optimization · Water cycle algorithm · Pareto-optimal solutions · Benchmark function · Metaheuristics

1 Introduction

Many problems in engineering and other fields of research can be considered as optimization problems aimed at finding an optimum design solution. To simulate a real-life problem

Communicated by V. Loia.

A. Sadollah · J. H. Kim
School of Civil, Environmental and Architectural Engineering,
Korea University, Seoul 136-713, South Korea

H. Eskandar
Faculty of Mechanical Engineering, University of Semnan,
Semnan, Iran

A. Bahreininejad (✉)
Faculty of Engineering, University of Malaya,
50603 Kuala Lumpur, Malaysia
e-mail: bahreininejad@um.edu.my; bahreininejad@gmail.com

for a real-life situation, a designer should investigate numerous objectives for obtaining the optimum design, while the given problem approaches its real-life nature as the number of objectives increases. The real-life optimization situations may involve solving various objectives, simultaneously.

As the number of objectives increases, the given problem approaches its real-life nature. Nowadays, researchers prefer to conduct their problems in real-life situations considering various objectives, simultaneously.

In contrast to an ordinary optimization problem (having only a single objective), multi-objective problems (MOPs) do not have a single solution (Glover and Kochenberger 2003). Depending on the designer's decision, an optimum design solution may be extracted from the set of Pareto front solutions (Deb 2001; Coello et al. 2002).

Among optimization algorithms, metaheuristic methods have shown their potential for finding the near-optimal solution to the numerical real-valued test problems (Osman and Laporte 1996; Blum and Andrea 2003). Over the last decades, numerous algorithms have been widely used to solve MOPs (Wang et al. 2012), since MOPs are widely observed in the domains of science and engineering (Lin and Chen 2013).

Numerous metaheuristic algorithms can provide designers the flexible means for solving optimization problems. Such methods are usually based on mathematical rules and models to imitate natural phenomena or real-life events to conduct search on the domain space and optimize given problems.

The concepts of metaheuristic algorithms are inspired by various events in nature such as natural selection and evolution processes used in genetic algorithms (GAs) (Holland 1975) or animal behavior and their search abilities for finding food such as particle swarm optimization (PSO) (Kennedy and Eberhart 1995), and social-human evolution such as imperialist competitive algorithm (Atashpaz and Lucas 2007).

Considering multi-objective approaches, a new version of the non-dominated sorting genetic algorithm (NSGA) (Srinivas and Deb 1995) alleviated the shortcomings (i.e., high computational effort, non-elitist approach, and specifying sharing parameters) of the NSGA. The improved method is known as NSGA-II (Deb et al. 2002a). Knowles and Corne (2000) came up with a new method called the Pareto archived evolution strategy (PAES). The PAES employs a local search approach for creating new generations using the population information from its selection process.

For handling several objective functions, the PSO was not exempted from the eyes of researchers and was considered in literature as a multi-objective optimizer (Coello and Lechuga 2002; Mostaghim and Teich 2003; Sierra and Coello 2005). For instance, Kaveh and Laknejadi (2011) combined the concept of the PSO with their developed method, the charge system search (CSS), for solving MOPs so called the CSS-MOPSO.

In addition, recently, many optimizers have been proposed in literature for tackling MOPs trying to improve and enhance the exploratory capabilities of non-dominated solutions (Zitzler and Thiele 1999; Zitzler et al. 2001; Gao and Wang 2010; Pradhan and Panda 2012; Wang et al. 2012; Mahmoodabadi et al. 2013).

In this paper, a recently proposed metaheuristic method which is based on the water cycle process has been used to tackle MOPs. The idea of water cycle algorithm (WCA) was first suggested by Eskandar et al. (2012) and the application and validation of the WCA was carried out for constrained optimization problems (Eskandar et al. 2012). The main purpose of this paper is to show the potential and performance of WCA for solving multi-objective functions.

The remaining of this paper is organized as follows: definitions of standard MOPs are given in Sect. 2. In addition, performance criteria used to have a quantitative assessment of MOPs are described in Sect. 2. In Sect. 3, detailed descriptions of the WCA and multi-objective water cycle algorithm (MOWCA) and their concepts are introduced. Section 4 represents the comparisons of the obtained statistical optimization results using the MOWCA with other optimizers for reported problems in form of tables and figures. Numerical examples and benchmark functions accompanied with their mathematical formulations considered in this paper are provided in Appendix A. Finally, conclusions are drawn in Sect. 5.

2 Multi-objective problems

The nature of many real-life problems are considered as a form of MOP. In many fields of science and engineering, multiple objective functions should be considered and optimized, simultaneously. Therefore, a MOP can be formulated

as follows:

$$F(X) = [f_1(X), f_2(X), \dots, f_N(X)]^T, \quad (1)$$

where $X = [x_1, x_2, x_3, \dots]$ is a vector of design variables. The simplest approach for the MOPs is to use weighting factor for each function and add them together based on the following equation (Haupt and Haupt 2004):

$$F = \sum_{n=1}^N w_n f_n, \quad (2)$$

where N is the number of objective functions, and w_n and f_n are weighting factors and objective functions, respectively. The major drawback of aforementioned technique (Eq. 2) is selecting a suitable value for the weighting factors (w_n). Different values of w_n give different optimal solutions for the same f_n .

However, the Pareto front approach can be used as an alternative approach to solve the MOPs. In the MOPs, there is usually a set of solution which is defined as Pareto optimal solutions or non-dominated solutions (Coello 2000). The main purpose of the multi-objective optimization is to find as many of non-dominated solutions as possible. The non-dominated solutions are defined as follows (Wang et al. 2012):

- (a) Pareto dominance: $U = (u_1, u_2, u_3, \dots, u_n) < V = (v_1, v_2, v_3, \dots, v_n)$ if and only if U is partially less than V in the objective space which it means:

$$\begin{cases} f_i(U) \leq f_i(V) & \forall i \\ f_i(U) < f_i(V) & \exists i \end{cases} \quad i = 1, 2, 3, \dots, N, \quad (3)$$

where N is the number of objective functions.

- (b) Pareto optimal solution: vector U is said to be a Pareto optimal solution if and only if any other solutions cannot be detected to dominate U . A set of Pareto optimal solution is called Pareto optimal front (PF_{optimal}).

Figure 1 illustrates the concept of Pareto optimal optimization technique for bi-objective problems. As can be seen in Fig. 1, solutions A and B are considered as non-dominated solutions. The reason is they are not dominated by each other for given objectives.

To clarify further, the obtained solution A has the minimum value for the f_1 compared with solution B . However, the obtained value for solution A for the f_2 is higher than solution B (see Fig. 1). In contracts, solution C is dominated by solutions A and B in terms of the minimum values for both objective functions (f_1 and f_2) as shown in Fig. 1. The solution C is called dominated solution and solutions A and B are known as Pareto optimal solutions (non-dominated solutions).

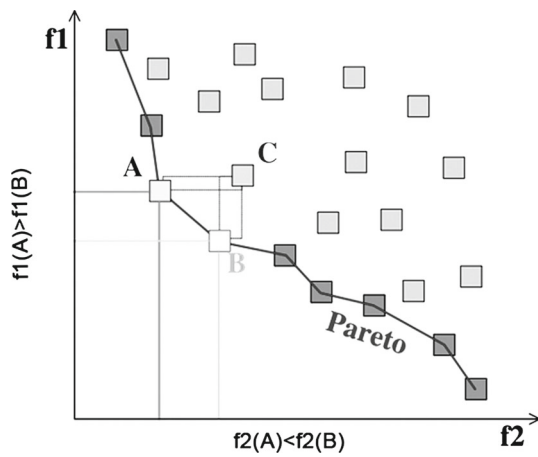


Fig. 1 Optimal Pareto solutions (A and B) for the two-dimensional domain

2.1 Performance metrics

In order to have an accurate evaluation for the proposed MOWCA to solve MOPs, three factors are usually taken into consideration (Zitzler et al. 2000). These three criteria are given in the following subsections.

2.1.1 Generational distance metric

Generational distance (GD) metric is defined as a criterion for the convergence between the Pareto optimal front (PF_{optimal}) and generated (calculated) Pareto front (PF_g). In fact, it is a Euclidian distance between the resulting non-dominated solution and PF_{optimal} (Kaveh and Laknejadi 2011).

Based on this definition, each algorithm with the minimum GD can have the best performance among others. This evaluation factor is defined in form of mathematical formulation, however, there are different variants of GD reported in the literature (Kaveh and Laknejadi 2011; Coello 2004):

$$GD1 = \left(\frac{1}{n_{pf}} \sum_{i=1}^{n_{pf}} d_i^2 \right)^{1/2}, \tag{4}$$

$$GD2 = \frac{1}{n_{pf}} \left(\sum_{i=1}^{n_{pf}} d_i^2 \right)^{1/2}, \tag{5}$$

where n_{pf} is number of member in PF_g and d is the Euclidean distance between member i th in PF_g and nearest member in PF_{optimal}. Meanwhile, the Euclidean distance (d) is obtained based on the following equation:

$$d(p, q) = d(q, p) = \left[\sum_{i=1}^n (f_{iq} - f_{ip})^2 \right]^{1/2}, \tag{6}$$

where $q = (f_{1q}, f_{2q}, f_{3q}, \dots, f_{nq})$ is a point on PF_g and $P = (f_{1p}, f_{2p}, f_{3p}, \dots, f_{np})$ is the nearest member to q in

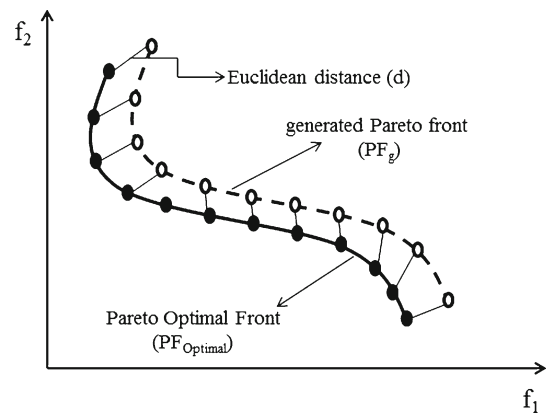


Fig. 2 Schematic view of GD criterion for the MOPs

PF_{optimal}. Figure 2 shows schematic view of this performance meter for two-dimensional space. The best obtained value for the GD metric is equal to zero which means the PF_g can exactly cover the PF_{optimal}.

2.1.2 Metric of spacing

Metric of spacing (S) gives an overview about the distribution of non-dominated solutions along the generated Pareto front (Kaveh and Laknejadi 2011). In other words, the main objective of this criterion is to demonstrate and clarify distribution of the non-dominated solutions in the objective space. Similar to the GD performance metric, the S metric is suggested by researchers having different formulations as given follows (Kaveh and Laknejadi 2011; Coello 2004):

$$S1 = \frac{\left[\frac{1}{n_{pf}} \sum_{i=1}^{n_{pf}} (d_i - \bar{d})^2 \right]^{1/2}}{\bar{d}}, \tag{7}$$

$$S2 = \left[\frac{1}{n_{pf} - 1} \sum_{i=1}^{n_{pf}} (d_i - \bar{d})^2 \right]^{1/2}, \tag{8}$$

where \bar{d} is the mean value of all d_i . The smallest value of S shows the best uniform distribution on PF_g. If all non-dominated solutions are uniformly distributed in the PF_g, then, the values of d_i and \bar{d} are the same, therefore, the value of S metric equals to zero.

3 Multi-objective water cycle algorithm

3.1 Water cycle algorithm

The WCA mimics the flow of rivers and streams towards the sea and derived by the observation of water cycle process. Let us assume that there are some rain or precipitation phenomena. An initial population of design variables (population of

streams) is randomly generated after raining process. The best individual (i.e., the best stream), classified in terms of having the minimum cost function (for minimization problem), is chosen as the sea (Eskandar et al. 2012).

Then, a number of good streams (i.e., cost function values close to the current best record) are chosen as rivers, while all other streams flow to the rivers and sea. In an N dimensional optimization problem, a stream is an array of $1 \times N$. This array is defined as follows:

$$\text{A Stream Candidate} = [x_1, x_2, x_3, \dots, x_N], \tag{9}$$

where N is the number of design variables (problem dimension). To start the optimization algorithm, an initial population representing a matrix of streams of size $N_{\text{pop}} \times N$ is generated. Hence, the matrix of initial population, which is generated randomly, is given as (rows and column are the number of population and the number of design variables, respectively):

$$\begin{aligned} \text{Total Population} &= \begin{bmatrix} \text{Sea} \\ \text{River}_1 \\ \text{River}_2 \\ \text{River}_3 \\ \vdots \\ \text{Stream}_{N_{\text{sr}}+1} \\ \text{Stream}_{N_{\text{sr}}+2} \\ \text{Stream}_{N_{\text{sr}}+3} \\ \vdots \\ \text{Stream}_{N_{\text{pop}}} \end{bmatrix} \\ &= \begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & \dots & x_N^1 \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{N_{\text{pop}}} & x_2^{N_{\text{pop}}} & x_3^{N_{\text{pop}}} & \dots & x_N^{N_{\text{pop}}} \end{bmatrix}, \tag{10} \end{aligned}$$

where N_{pop} and N are the total number of population and the number of design variables, respectively. Each of the decision variable values (x_1, x_2, \dots, x_N) can be represented as floating point number (real values) or as a predefined set for continuous and discrete problems, respectively. The cost of a stream is obtained by the evaluation of cost function (C) given as follows:

$$C_i = \text{Cost}_i = f(x_1^i, x_2^i, \dots, x_N^i) \quad i = 1, 2, 3, \dots, N_{\text{pop}}. \tag{11}$$

At the first step, N_{pop} streams are created. A number of N_{sr} from the best individuals (minimum values) are selected as a sea and rivers. The stream which has the minimum value among others is considered as the sea. In fact, N_{sr} is the

summation of number of rivers (which is defined by user) and a single sea (Eq. 12). The rest of the population (i.e., streams flow to the rivers or may directly flow to the sea) is calculated using the following equation:

$$N_{\text{sr}} = \text{Number of Rivers} + \underbrace{1}_{\text{Sea}}, \tag{12}$$

$$N_{\text{Stream}} = N_{\text{pop}} - N_{\text{sr}}. \tag{13}$$

Equation (14) shows the population of streams which flow to the rivers or sea. Indeed, Eq. (14) is part of Eq. (10) (i.e., total individual in population):

$$\begin{aligned} \text{Population of Streams} &= \begin{bmatrix} \text{Stream}_1 \\ \text{Stream}_2 \\ \text{Stream}_3 \\ \vdots \\ \text{Stream}_{N_{\text{Stream}}} \end{bmatrix} \\ &= \begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & \dots & x_N^1 \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{N_{\text{Stream}}} & x_2^{N_{\text{Stream}}} & x_3^{N_{\text{Stream}}} & \dots & x_N^{N_{\text{Stream}}} \end{bmatrix}. \tag{14} \end{aligned}$$

Depending on flow magnitude, each river absorbs water from streams. The amount of water entering a river and/or the sea, hence, varies from stream to stream. In addition, rivers flow to the sea which is the most downhill location. The designated streams for each rivers and sea are calculated using the following equation (Eskandar et al. 2012):

$$NS_n = \text{round} \left\{ \left| \frac{\text{Cost}_n}{\sum_{i=1}^{N_{\text{sr}}} \text{Cost}_i} \right| \times N_{\text{Stream}} \right\}, \quad n = 1, 2, \dots, N_{\text{sr}}, \tag{15}$$

where NS_n is the number of streams which flow to the specific rivers and sea. As it happens in nature, streams are created from the raindrops and join each other to generate new rivers. Some stream may even flow directly to the sea. All rivers and streams end up in the sea that corresponds to the current best solution.

Let us assume that there are N_{pop} streams of which $N_{\text{sr}} - 1$ are selected as rivers and one is selected as the sea. Figure 3a shows the schematic view of a stream flowing towards a specific river along their connecting line.

The distance X between the stream and the river may be randomly updated as the following relation:

$$X \in (0, C \times d), \quad C > 1, \tag{16}$$

where $1 < C < 2$ and the best value for C may be chosen as 2; d is the current distance between stream and river. The

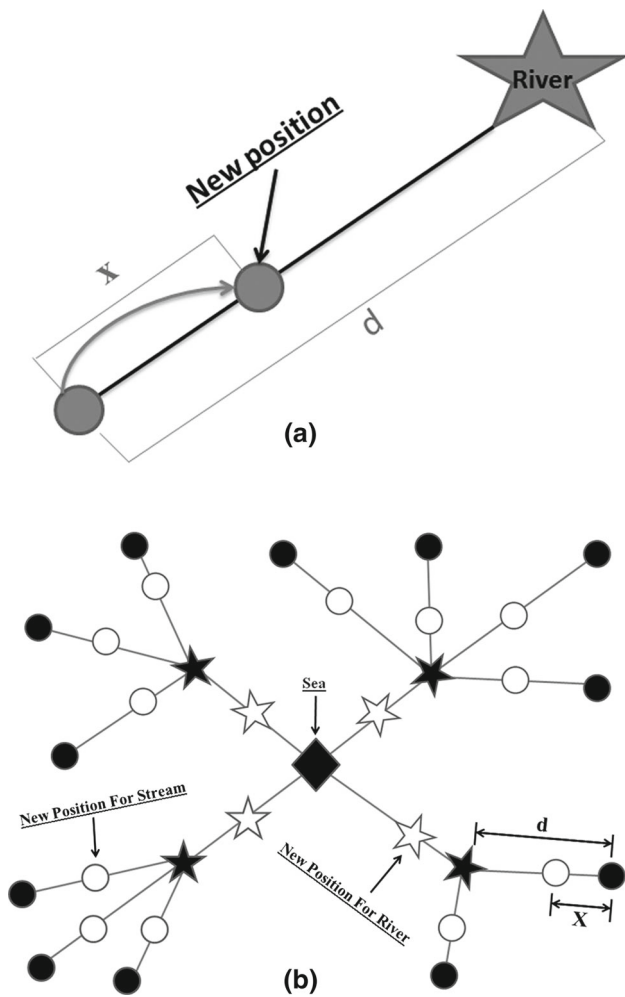


Fig. 3 a Schematic description of the stream’s flow to a specific river; b schematic of the WCA optimization process

value of X in relation (16) corresponds to a random number (uniformly distributed or determined from any appropriate distribution) between 0 and $(C \times d)$.

Setting $C > 1$ allows streams to flow in different directions towards rivers. This concept may also be used to describe rivers flowing to the sea. Therefore, as the exploitation phase in the WCA, the new position for streams and rivers have been suggested as follows (Eskandar et al. 2012):

$$\vec{X}_{Stream}^{i+1} = \vec{X}_{Stream}^i + \text{rand} \times C \times (\vec{X}_{River}^i - \vec{X}_{Stream}^i), \tag{17}$$

$$\vec{X}_{Stream}^{i+1} = \vec{X}_{Stream}^i + \text{rand} \times C \times (\vec{X}_{Sea}^i - \vec{X}_{Stream}^i), \tag{18}$$

$$\vec{X}_{River}^{i+1} = \vec{X}_{River}^i + \text{rand} \times C \times (\vec{X}_{Sea}^i - \vec{X}_{River}^i), \tag{19}$$

where rand is an uniformly distributed random number between zero and one. Equations (17) and (18) are for streams

which flow to their corresponding rivers and sea, respectively. Notations having vector sign correspond to vector values, otherwise the rest of notations and parameters are considered as scalar values. If the solution given by a stream is better than its connecting river, the positions of river and stream are exchanged (i.e., the stream becomes a river and the river becomes a stream). A similar exchange can be performed for a river and the sea.

The evaporation process operator also is introduced to avoid premature (immature) convergence to local optima (exploitation phase). Basically, evaporation causes sea water to evaporate as rivers/streams flow to the sea. This leads to new precipitations. Therefore, we have to check if the river/stream is close enough to the sea to make the evaporation process occur. For that purpose, the following criterion is utilized for evaporation condition:

$$\begin{aligned} & \text{if } \|\vec{X}_{Sea}^i - \vec{X}_{River}^i\| < d_{\max} \text{ or } \text{rand} < 0.1 \\ & \quad i = 1, 2, 3, \dots, N_{sr} - 1, \\ & \text{Perform raining process } u \text{ sin } g \text{ Eq. (20)} \\ & \text{end} \end{aligned}$$

where d_{\max} is a small number close to zero. After evaporation, the raining process is applied and new streams are formed in the different locations (similar to mutation in the GAs). To further clarify, if evaporation condition is satisfied for any rivers, the corresponding river together with its streams will be removed (i.e., evaporated). Afterward, the new streams which are equal to the number of previous streams and a river will be generated in new positions using Eq. (20). Hence, in the new generated sub-population, the best stream will act as a new river and other streams move toward their new river.

Indeed, the evaporation operator is responsible for the exploration phase in the WCA. The following equation is used to specify the new locations of the newly formed streams:

$$\vec{X}_{Stream}^{new} = L\vec{B} + \text{rand} \times (U\vec{B} - L\vec{B}), \tag{20}$$

where LB and UB are lower and upper bounds defined by the given problem, respectively. Similarly, the best newly formed stream is considered as a river flowing to the sea. The rest of new streams are assumed to flow into the rivers or may directly flow into the sea.

A large value for d_{\max} prevents extra searches and small values encourage the search intensity near the sea. Therefore, d_{\max} controls the search intensity near the sea (i.e., best obtained solution). The value of d_{\max} adaptively decreases as follows:

$$d_{\max}^{i+1} = d_{\max}^i - \frac{d_{\max}^i}{\text{Max Iteration}} \tag{21}$$

Infiltration and transpiration are considered as two important steps in the water cycle process seen in nature. Infiltration is an important process where rain water is absorbed into the

Table 1 Pseudo-code of the WCA

```

• Set user parameter of the WCA:  $N_{pop}$ ,  $N_{sr}$ , and Maximum_Iteration.
• Determine the number of streams (individuals) which flow to the rivers and sea using Eqs. (12) and (13).
• Randomly create initial population of streams.
• Define the intensity of flow (How many streams flow to their corresponding rivers and sea) using Eq. (15).
while ( $t < \text{Maximum\_Iteration}$ ) or (any stopping condition)
  for  $i = 1 : \text{Population Size } (N_{pop})$ 
    Stream flows to its corresponding rivers and sea using Eqs. (17) and (18).
    Calculate the objective function of the generated stream
    if  $F_{\text{New\_Stream}} < F_{\text{river}}$ 
      River = New_Stream;
    if  $F_{\text{New\_Stream}} < F_{\text{Sea}}$ 
      Sea = New_Stream;
    end if
  end if
  River flows to the sea using Eq. (19).
  Calculate the objective function of the generated river
  if  $F_{\text{New\_River}} < F_{\text{Sea}}$ 
    Sea = New_River;
  end if
end for
for  $i = 1 : \text{number of rivers } (N_{sr})$ 
  if ( $\text{norm}(\text{Sea} - \text{River}) < d_{\text{max}}$ ) or ( $\text{rand} < 0.1$ )
    New streams are created using Eq. (20).
  end if
end for
Reduce the  $d_{\text{max}}$  using Eq. (21).
end while
Postprocess results and visualization

```

ground, through the soil and underlying rock layers. For the transpiration step, as plants absorb water from the soil, the water moves from the roots through the stems to the leaves. Once the water reaches the leaves, some of it evaporates from the leaves adding to the amount of water vapor in the air.

However, in the standard WCA (in its current version), the loss of waters using groundwater or plant absorption was not considered. In fact, these two steps (i.e., infiltration and transpiration steps in water cycle processes) are not included in the standard WCA.

The development of the WCA optimization process is illustrated by Fig. 3b where circles, stars, and the diamond correspond to streams, rivers, and sea, respectively. The white (empty) shapes denote the new positions taken by streams and rivers. In addition, Table 1 shows the pseudo-code and step-by-step processes of the WCA in detail.

3.1.1 Similarities and differences with other optimizers

In this subsection, similarities and differences of WCA with other optimization techniques are highlighted. The PSO (Kennedy and Eberhart 1995) and ICA (Atashpaz and Lucas

2007) as two common metaheuristic optimizers are selected for comparison purposes with the WCA. Indeed, every metaheuristic algorithm has its own approach and methodology in finding global optimum solution.

As a similarity among the WCA, PSO, and ICA, we can say that all methods are categorized as population-based metaheuristic algorithms; population of particles in the PSO, population of countries in the ICA, and population of streams in the WCA. As for the ICA, the WCA utilizes the concept of grouping for individuals using different strategy.

Except this similarity, their concepts, parameters and operators are different with each other. The PSO's concept is based on the movement of particles (e.g., fishes, birds, etc.) and their personal and best individual experiences (Kennedy and Eberhart 1995). The WCA's notions are derived by the water cycle process in nature and the observation of how streams and rivers flow to the sea, while the ICA is inspired by the imperialistic competition and social-political phenomenon in the globe.

The updating formulations for the positions of rivers and streams differ from the updating formulations used in the PSO and ICA. The WCA does not use the concept of moving

directly to the best solution (global best) as used in the PSO. In fact, the WCA utilizes the concept of moving indirectly from streams to the rivers and from rivers to the sea (i.e., the temporal obtained optimum solution).

In contrast, in the ICA, colony's countries move toward their relevant imperialist country; however, the imperialist countries do not have any moment toward the best solution (i.e., best imperialist country).

In the WCA, rivers [a number of best selected solutions except the best one (sea), (Eq. 12)] act as guidance points for guiding other individuals in the population (streams) towards better positions (see Fig. 3b) and to avoid the search in inappropriate regions (see Eq. 17).

It is worth pointing out that rivers, themselves, move towards the sea (i.e., best obtained solution). They are not fixed points (see Eq. 19) unlike the imperialist countries in the ICA. In fact, this procedure (moving streams to the rivers and, then moving rivers to the sea) leads to indirect movements towards the best solution by the WCA. In fact, the third movement (moving rivers to the sea, Eq. 19) does not define in the ICA (Atashpaz and Lucas 2007).

On other hand, in the PSO, individuals (particles) based on their personal and best experiences attempt to find the best solution as the searching approach is moving directly towards the best optimal solution. In addition, in the WCA, a number of near-best to best selected solutions (rivers + sea) attract other individuals of population (streams) based on their goodness of the function values (i.e., intensity of flow) using Eq. (15). However, in the classical PSO, this process is not used.

Another difference among the WCA, PSO, and ICA is the existence of evaporation condition and raining process in the WCA which corresponds to the exploration phase. The evaporation condition and raining process provide an escape mechanism for the WCA to avoid getting trapped in local optima, while in the PSO, the exploration mechanism (formulation) is different.

In the PSO, inertia weight (w) (i.e., a user parameter) in the updating equation (movement equation) is responsible for the exploration phase and reduces at each iteration, while in the ICA, based on the revolution probability (i.e., defined by user), revolution phase is in charge of exploration task. Table 2 summarizes the differences of three reported optimizers in terms of applied strategies.

3.2 Proposed MOWCA

In order to convert the WCA as an efficient multi-objective optimization algorithm, it is crucially important to define predominant features of WCA in a correct way (i.e., sea and rivers). In standard optimization problems by WCA, only one objective function should be minimized and in this condition,

a number of best obtained solutions in the population are selected as a sea (best obtained solution) and rivers.

Nevertheless, for MOPs, there is more than one function to be minimized (or maximized). Therefore, modifications required for the standard WCA for selecting sea and rivers in the multi-objective space. To select the most efficient (best) solutions in the population as a sea and rivers, crowding-distance mechanism is used. The concept of crowding-distance mechanism was first defined by Deb et al. (2002a).

This parameter is a criterion to show distribution of non-dominated solutions around a particular non-dominated solution. Figure 4 illustrates how to calculate crowding-distance for point i which is the average side length of the cuboid (Deb et al. 2002a). Lower value for crowding-distance indicates more distribution of the solutions in a specific region. In MOPs, this parameter is calculated in objective space. Hence, to compute this parameter for each non-dominated solution, all non-dominated solutions should be sorted in term of values for one of the objective functions.

Selection of the sea and rivers from the obtained population as the best guide solution for other solutions at each iteration is a vital step in the MOWCA. This affects both the convergence capability of the MOWCA as well as maintaining a good distribution of non-dominated solutions. Therefore, for all iterations, crowding-distance for all non-dominated solutions should be calculated to determine which solutions have the highest crowding-distance values.

Afterwards, the obtained non-dominated solutions are designated as sea and rivers and also, the intensity of flow for rivers and sea are calculated based on the crowding-distance values. In this situation, most likely, some non-dominated solutions creates around sea and rivers at next iterations and their value of crowding-distance amends and reduces.

Moreover, it is significantly important to save the non-dominated solutions in an archive to generate the Pareto front sets. This archive is updated at each iteration and dominated solutions are eliminated from the archive and all non-dominated solutions are added to the Pareto archive.

However, the size of Pareto archive (number of non-dominated solutions in the archive) is variable in the literature. Therefore, whenever the number of members in the Pareto archive increases the Pareto archive size, the crowding distance is applied again in order to eliminate as many non-dominated solutions as necessary which have the lowest crowding-distance values among the Pareto archive members.

3.3 Steps and flowchart of MOWCA

The steps of the MOWCA are summarized as follows:

Step 1: Choose the initial parameters of MOWCA: N_{sr} , d_{max} , N_{pop} , Max_Iteration, and Pareto archive size.

Table 2 Differences among three optimization methods in terms of their approaches for finding global optimum solution

Strategy	ICA	PSO	WCA
Population	Countries (imperialists and colonies)	Particles (e.g., fishes, birds)	Streams (including sea and rivers)
User parameter	- N_{pop}^2 - N_{imp} (number of imperialist countries) - ζ (colonies mean cost coefficient) - μ (revolution rate) - P (revolution probability)	- N_{pop} - w (inertia weight) - c_1 (personal learning constant) - c_2 (global learning constant) - V_{max} (maximum velocity)	- N_{pop} - N_{sr} (number of rivers + sea) - d_{max} (maximum allowable distance between river and sea) Evaporation condition: if $norm(x_{sea}^i - x_{river}^i) < d_{max}$ Perform raining using Eq. (20) end if
Global search	Revolution: random changes occur in the characteristics of some colonies	Inertia weight (first term of the movement equation): $w \times v_i^t$	
Local search	Assimilation strategy: moving colonies towards their imperialist country (imperialist countries are fixed point)	The second and third terms of the movement equation: $c_1 r_1 (pBes \vec{T}_i - \vec{X}_i) +$ $c_2 r_2 (gBes \vec{T}_i - \vec{X}_i)$	Moving streams to the rivers and rivers to the streams (Eqs. 17–19)
Selection	if $F_{Col}(k^1) < F_{Imp}(n^1)$ Imp. (n) = Col. (k); end if - Calculate the total power of an empire (TC_n) - Find the weakest colony - Assign the weakest colony of the weakest empire to an empire with high probability and eliminate an empire with no colony	if $F_{New}^3 < F_{Old}$ Accept New_Particle if $F_{New} < F_{Best}$ Accept New_Particle end if end if	if $F_{Stream} < F_{River}$ River = Stream; if $F_{Stream} < F_{Sea}$ Stream = Sea; end if end if if $F_{River} < F_{Sea}$ River = Sea; end if

¹ k and n are indexes for colonies and imperialists, respectively, ² N_{pop} total number of population, ³ F_{*} value of objective function

Step 2: Generate random initial population and form the initial streams, rivers, and sea using Eqs. (10), (12), and (13).

Step 3: Calculate the value of multi-objective functions for each stream using Eq. (11).

Step 4: Determine the non-dominated solutions in the initial population and save them in the Pareto archive.

Step 5: Calculate crowding-distance for each Pareto archive member.

Step 6: Select a sea and rivers based on the crowding-distance value.

Step 7: Determine the intensity of the flow for rivers and sea based on the crowding distance values using Eq. (15).

Step 8: Streams flow into the rivers using Eq. (17).

Step 9: Streams flow into the sea using Eq. (18).

Step 10: Exchange positions of river and sea with a stream which gives the best solution.

Step 11: Rivers flow into the sea using Eq. (19).

Step 12: Similar to Step 10, if a river finds better solution than the sea, the position of river is exchanged with the sea.

Step 13: Check the evaporation condition using the pseudo-code given in Subsect. 3.1.

Step 14: If the evaporation condition is satisfied, the raining process occurs using Eq. (20).

Step 15: Reduce the value of d_{max} which is a user-defined parameter using Eq. (21).

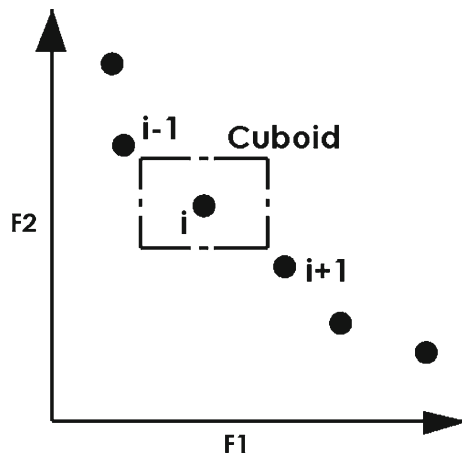


Fig. 4 Schematic view of crowding-distance calculation

Step 16: Determine the new non-dominated solutions in the population and save them in the Pareto archive.

Step 17: Eliminate any dominated solutions in the Pareto archive.

Step 18: If the number of member in the Pareto archive is more than the determined Pareto archive size, go to the Step 19, otherwise, go to the Step 20.

Step 19: Calculate the crowding-distance value for each Pareto archive member and remove as many members as necessary with the lowest crowding-distance value.

Step 20: Calculate the crowding-distance value for each Pareto archive member to select new sea and rivers.

Step 21: Check the convergence criteria. If the stopping criterion is satisfied, the algorithm will be stopped, otherwise return to the Step 8.

4 Optimization results and discussions

In this section, 12 MOPs are considered for validating the performance of the proposed MOWCA. These benchmark problems are selected from a set of significant past studies in this area (Fonseca and Fleming 1993; Deb 2002; Freschi and Repetto 2006; Gao and Wang 2010; Kaveh and Laknejadi 2011). The natures of mentioned problems include various types of objective functions (quadratic, cubic, polynomial, and nonlinear) having different number of design variables. Mathematical formulations of all considered MOPs accompanied with their optimal Pareto front are listed in Appendix A.

The proposed MOWCA was coded in MATLAB and the task of optimization was executed using 30 independent runs. For all benchmark problems, the initial parameters for the MOWCA (N_{total} , N_{sr} , and d_{max}) were selected as 50, 10, and $1e-5$, respectively.

Additionally, the maximum number of iterations varies for each problem in order to have fair comparisons. In fact, the

maximum number of function evaluations (NFEs) is taken as the stopping condition, similar to the other methods in this paper.

Meanwhile, based on the previous studies (Deb et al. 2002a; Freschi and Repetto 2006; Gao and Wang 2010; Kaveh and Laknejadi 2011; Pradhan and Panda 2012), the Pareto archive size is set to 100 for all reported MOPs.

Moreover, Eqs. (4) and (7) are used to calculate the performance parameters (i.e., the GD1 and S1) for test problems 2, 3, 6, 7, 8, and 9. Similarly, Eqs. (5) and (8) (i.e., GD2 and S2) are utilized for computing the aforementioned parameters for test problems 1, 4, 5, and 10. The comparison set adopted for our study is composed of state-of-the-art techniques covering a wide range of techniques such as the NSGA-II, PAES, MOPSO, charge system search and particle swarm optimization (CSS-MOPSO), and immune system multi-objective optimization algorithm (ISMOA) (Knowles and Corne 2000; Deb et al. 2002a; Deb et al. 2002b; Zhang et al. 2009; Kaveh and Laknejadi 2011).

For quantitative and qualitative evaluations, the final statistical results for these algorithms are evaluated based on the values obtained for the performance parameters (i.e., GD and S) and the generated plot for the Pareto front using the MOWCA. Table 3 shows the statistical optimization results including the best, mean, worst, standard deviation (SD), and NFEs used for all of the reported MOPs in this paper using the MOWCA.

For the DTLZ problems, two cases are considered for bi-objective and three-objective functions in this paper. From Table 3, for the DTLZ problems two sets of results are provided. The first and second rows of Table 3 correspond to bi-objective and three-functions, respectively, for DTLZ series.

Tables 4 and 5 represent the obtained statistical results for the GD as performance metric for different optimizers for the MOPs given in Appendix A. Looking at Table 4, it can be inferred that the MOWCA has the advantage of having the smallest value of GD for the DEB, POL, KUR, ZDT3, ZDT4, ZDT6, and VNT functions, while the MOPSO and CSS-MOPSO (Kaveh and Laknejadi 2011) indicate better GD for the FON and ZDT1 functions, respectively (see Table 4).

Judging by Table 5, more comparisons have been carried out using the MOWCA, rank-based multi objective artificial physics optimization (RMOAPO), simple multi objective artificial physics optimization (SMOAPO), and multi objective particle swarm optimization (MOPSO) (Wang and Zeng 2013). In addition, the obtained optimization results given in Table 5 are based on 10,000 function evaluations.

Looking at Table 5, (similar to Table 4), the MOWCA surpassed other reported optimizers obtaining better statistical results for the GD. The best attained statistical results (i.e., mean and SD) are highlighted in bold as shown in Tables 4

Table 3 Statistical optimization results obtained by the MOWCA for all reported MOPs given in Appendix A

MOPs	Best solution		Mean solution		Worst solution		SD ^a		NFEs
	GD	S	GD	S	GD	S	GD	S	
DEB	1.49e-5	0.0016	5.75e-5	2.01e-3	9.10e-5	0.0025	2.03e-5	2.09e-4	4,000
FON	4.69e-3	0.26	5.6e-3	0.35	6.61e-3	0.44	4.84e-4	4.60e-2	10,000
POL	1.71e-2	0.11	2.06e-2	0.17	2.62e-2	0.26	2.07e-3	0.10	10,000
KUR	1.65e-3	0.055	2.65e-3	8.73e-2	0.043	0.27	6.53e-4	5.26e-2	12,000
VNT	2.63e-3	0.014	3.09e-3	2.70e-2	3.61e-3	0.114	2.88e-4	2.13e-2	4,000
ZDT1	2.68e-3	0.202	4.13e-3	0.258	5.69e-3	0.354	7.91e-4	3.42e-2	10,000
ZDT3	7.39e-4	0.465	1.30e-3	0.589	1.81e-3	0.679	3.43e-4	7.35e-2	20,000
ZDT4	1.12e-3	0.132	2.19e-3	0.167	5.23e-3	0.21	1.32e-4	4.57e-3	25,000
ZDT6	3.23e-4	0.84	1.03e-2	0.717	0.018	5.98	1.01e-2	1.11	10,000
DTLZ 2	1.65e-4	1.45e-3	1.78e-4	1.95e-3	1.87e-4	2.44e-3	6.70e-6	3.73e-4	10,000
	1.39e-3	0.0221	1.93e-3	0.026	2.25e-3	0.028	2.87e-4	2.14e-3	5,000
DTLZ 4	1.76e-4	1.25e-3	2.06e-4	2.19e-3	2.57e-4	3.03e-3	2.56e-5	4.88e-4	10,000
	4.69e-4	3.00e-3	1.79e-3	0.027	3.66e-3	0.086	9.57e-4	2.293-2	5,000
DTLZ 7	1.27e-5	1.23e-4	2.65e-5	3.44e-4	4.99e-5	7.18e-4	1.62e-5	2.70e-4	10,000
	3.31e-4	3.82e-4	2.65e-3	0.015	1.81e-2	0.036	5.47e-3	1.21e-2	5,000

^a Standard deviation

Table 4 Mean and SD for the GD criterion

MOPs	MOWCA		NSGA-II		PAES		MOPSO		CSS-MOPSO	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
DEB	0.000057	0.000020	0.023046	0.045429	0.163484	0.441303	0.000118	0.000025	N/A	N/A
FON	0.005600	0.000484	0.007174	0.000301	0.028305	0.007129	0.004059	0.004059	0.004942	0.004942
POL	0.020698	0.002076	0.020980	0.002283	0.085626	0.100294	0.022610	0.022610	0.022928	0.022928
KUR	0.002655	0.000653	0.029255	0.027170	0.549140	0.030744	0.008450	0.000051	N/A	N/A
VNT	0.003091	0.000288	0.005870	0.017800	0.078100	0.044500	N/A	N/A	N/A	N/A
ZDT1	0.004138	0.000791	0.003731	0.000342	0.004932	0.006013	0.096400	0.096400	0.003048	0.003048
ZDT3	0.001304	0.000343	0.005031	0.000162	0.082004	0.107889	0.068005	0.068005	0.004781	0.004781
ZDT4	0.002195	0.000132	0.003559	0.000589	0.450385	0.063752	0.228776	0.070243	0.003462	0.000380
ZDT6	0.010373	0.010144	0.024666	0.024338	0.011739	0.016882	0.019078	0.019078	0.026345	0.026345

N/A not available

Table 5 Mean and SD for the GD criterion for the DTLZ series

MOPs	MOWCA		RMOAPO		SMOAPO		MOPSO		
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	
M = 2									
DTLZ 2	1.78e-4	6.70e-6	0.0212	0.0043	0.0433	0.0053	0.0469	0.0051	
DTLZ 4	2.06e-4	2.56e-5	0.0194	0.0036	0.0231	0.0039	0.0386	0.0047	
DTLZ 7	2.65e-5	1.62e-5	N/A	N/A	N/A	N/A	N/A	N/A	

and 5. In fact, the MOWCA offers the best performance obtaining the lowest GD for the most MOPs (10 out of 12 in Tables 4 and 5) in this paper and has been placed in first rank for the GD. On the contrary, the PAES and NSGA-II (Deb et al. 2002a) have the worst statistical results in terms of the GD (see Table 4).

In order to have more comparisons with the optimization results obtained by the MOWCA, the VNT and DEB functions were solved using other optimizers given in the literature. The statistical optimization results found by the MOWCA are given in Table 3 for the GD. For the VNT function, vector immune system (VIS) (Freschi and Repetto

Table 6 Mean and SD for the S metric

MOPs	MOWCA		NSGA-II		PAES		MOPSO		CSS-MOPSO	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
DEB	0.002016	0.000209	0.00369	0.003372	1.114617	4.434594	0.010392	0.002782	N/A	N/A
FON	0.352934	0.046011	0.356864	0.041145	2.914900	0.994202	0.787028	0.098926	0.175515	0.019036
POL	0.170715	0.105542	0.491536	0.056932	4.076686	2.125581	1.089849	0.464838	0.813929	0.608955
KUR	0.087356	0.052608	0.036136	0.010977	0.197532	0.064114	0.09747	0.01675	N/A	N/A
VNT	0.027078	0.002138	0.042400	0.009190	0.059400	0.060800	N/A	N/A	N/A	N/A
ZDT1	0.258370	0.034299	0.503569	0.052127	3.765871	1.367000	0.756200	0.145703	0.199631	0.048466
ZDT3	0.589864	0.073595	0.502427	0.047587	2.044142	1.500228	0.794325	0.070546	0.301347	0.042321
DT4	0.167352	0.004576	0.485384	0.052186	2.799198	2.015591	1.081437	0.195925	0.175199	0.031783
ZDT6	0.717644	1.113447	2.151897	2.285011	3.499715	2.217825	3.707637	0.849501	3.179233	1.351519

Table 7 Mean and SD for the S metric for the DTLZ problems

MOPs	MOWCA		RMOAPO		SMOAPO		MOPSO		
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	
M = 2									
DTLZ 2	0.0019	3.73e-4	0.0046	0.0021	0.0073	0.0031	0.0087	0.0037	
DTLZ 4	0.0022	4.88e-4	0.0029	0.0036	0.0038	0.0039	0.0053	0.0042	
DTLZ 7	3.44e-4	2.70e-4	N/A	N/A	N/A	N/A	N/A	N/A	

Table 8 Mean and SD for the S metric for the DTLZ problems

MOPs	MOWCA (5,000)		ISMOA (25,000)		NSGA-II (25,000)	
	Mean	SD	Mean	SD	Mean	SD
M = 3						
DTLZ 2	0.026295	0.002137	0.045306	0.003490	0.056743	0.006245
DTLZ 4	0.026709	0.022928	0.039188	0.018554	0.029916	0.027569
DTLZ 7	0.015340	0.012134	0.056493	0.006398	0.059228	0.021574
M = 2						
DTLZ 2	0.004079	0.000728	0.004143	0.000238	0.006888	0.000576
DTLZ 4	0.003814	0.001011	0.004265	0.000348	0.005764	0.003064
DTLZ 7	0.000734	0.000598	0.005211	0.000391	0.008157	0.000749

Values in parenthesis mean the NFEs

2006) has obtained its mean GD value of 0.0033 and SD of 0.00171, while multi objective immune system algorithm (MISA) (Coello and Cruz Cortés 2005) has attained the values of 0.00338 and 0.00215 for the aforementioned evaluators, respectively.

Recently, multi-objective cat swarm optimization (MOCSSO) (Pradhan and Panda 2012) was investigated for solving the Deb benchmark problem using the same conditions and offered the mean and SD values of 0.000769 and 0.000057, respectively. In summary, it can be seen from Table 3 that the MOWCA offers superiority over the VIS, MISA, and MOCSSO in terms of mean and SD values for the GD metric.

Accordingly, in Tables 6, 7, and 8, the metric of spacing (S) is presented for reported MOPs. In order to perform a fair comparison with corresponding optimizers, the used NFEs for Tables 6 and 7 are set to 10,000, while for Table 8 the NFEs is chosen as 5,000.

By observing Table 6, the MOWCA, (as for the CSS-MOPSO for some cases), obtained the best optimization results with respect to the average metric of spacing for the most MOPs in this paper. Also, the PAES Knowles and Corne (2000) has the weakest performance of all (see Table 6). Comparing with other optimizers, including the MOPSO, SMOAPO, and RMOAPO, using 10,000 NFEs (see Table 7 for bi-objective DTLZ problem) for the DTLZ series, the

MOWCA could find a wide variety of solutions having uniform spread and the smallest value for the S metric.

In addition, in Table 8 (for three-objective DTLZ problem), using different optimizers and different NFEs, the obtained statistical results are compared. As can be seen in Table 8, in terms of S metric, the MOWCA has been placed in the first rank offering the minimum value for the metric of spacing.

However, the SD values obtained by the ISMOA (Zhang et al. 2009) are slightly better than those by the MOWCA. The maximum NFEs for the MOWCA is set to 5,000 which is five times fewer than the 25,000 NFEs considered for the NSGA-II and ISMOA. Hence, better SD obtained by the ISMOA can be easily justified by NFEs against the MOWCA. To be more precise, the ISMOA required more time for stability of its solutions (Zhang et al. 2009).

However, the ISMOA and NSGA-II could not find non-dominated solutions with well distribution compared to the MOWCA (Deb et al. 2002a; Zhang et al. 2009). The best obtained statistical results are highlighted in bold in Tables 6, 7, and 8. Moreover, the VIS (Freschi and Repetto 2006), MISA (Coello and Cruz Cortés 2005), and MOCSO (Pradhan and Panda 2012) were tackled to solve the Deb and VNT benchmark functions. The values for mean and SD using the MOCSO for the Deb test problem are 0.009 and 0.0007, respectively, for the metric of spacing.

Likewise, for the VNT function, the mean S and its SD values obtained by the VIS were 0.0589 and 0.00950, respectively; whereas, the MISA reached values of 0.0710 and 0.00962 for the aforesaid parameters (Coello and Cruz Cortés 2005). From the assessments, it can be seen from Tables 6, 7, and 8 that compared with the VIS, MISA, and MOCSO, the MOWCA has the advantages of having smaller statistical values for metric of spacing.

It is worth pointing out that the MOWCA offers acceptable statistical results for all performance parameters, while for the NSGA-II (Deb et al. 2002a) the results for the GD metric are considerably less accurate compared with the MOWCA. In general, a suitable optimization algorithm should offer reasonable statistical results for all existing evaluators given in the literature.

Figure 5 demonstrates the comparisons between the exact and computed Pareto fronts using the proposed optimizer for the MOPs given in Table 3. Further, Figs. 6 and 7 show the final non-dominated solutions obtained by the MOWCA and their optimal Pareto fronts for the DTLZ series problems (i.e., DTLZ 2, 4, and 7) having two and three objective functions, respectively. It is clear that the considered performance metrics for the given MOPs using the MOWCA have smaller values of the GD and S metrics, as shown in Figs. 5, 6, and 7.

5 Conclusions

This paper presented a proposed optimization technique for solving MOPs called MOWCA. The basic concepts of the WCA are inspired by observation of the water cycle process in real world. In this paper, the MOWCA was used for solving a number of well-known MOPs (i.e., 12 problems). The efficiency and performance of the MOWCA were carried out using two popular criteria (i.e., metric of generational distance and spacing). The obtained statistical results from performance metrics apparently reveal that the MOWCA was able to offer solutions close to the full optimal Pareto front in addition to providing superior quality of solutions in comparison with other state of the art algorithms considered in this paper. In general, MOWCA offers competitive solutions compared with other population based algorithms based on the reported numerical results in this research. In fact, although the robustness and exploratory capability of the MOWCA depends on the nature and complexity of the problems, the obtained optimization results show that the MOWCA can be considered suitable and efficient alternative method, having comparable degree of accuracy to find the optimal Pareto fronts for different scales of MOPs.

Acknowledgments This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (NRF-2013R1A2A1A01013886).

Appendix A: Mathematical formulation of studied MOPs

This Appendix represents the MOPs used in this paper to conduct a qualitative assessment for performance and efficiency of the MOWCA.

- (1) *Test problem 1-DEB*: Deb’s function is a problem with two design variables. This problem is defined as follows (Deb 2002):

$$DEB : \min \begin{cases} f_1(X) = x_1 \\ f_2(X) = g(X) \times h(X) \end{cases}, \tag{22}$$

where

$$g(X) = 11 + x_2^2 - 10 \cos(2\pi x_2), \tag{23}$$

$$h(X) = \begin{cases} 1 - \sqrt{f_1/g} & \text{if } f_1(X) \leq g(X) \\ 0 & \text{Otherwise} \end{cases}$$

$$\text{where } 0 \leq x_1 \leq 1, -30 \leq x_2 \leq 30. \tag{24}$$

The Pareto optimal front for this problem is convex and defined as $x_1 \in [0, 1], x_2 = 0$.

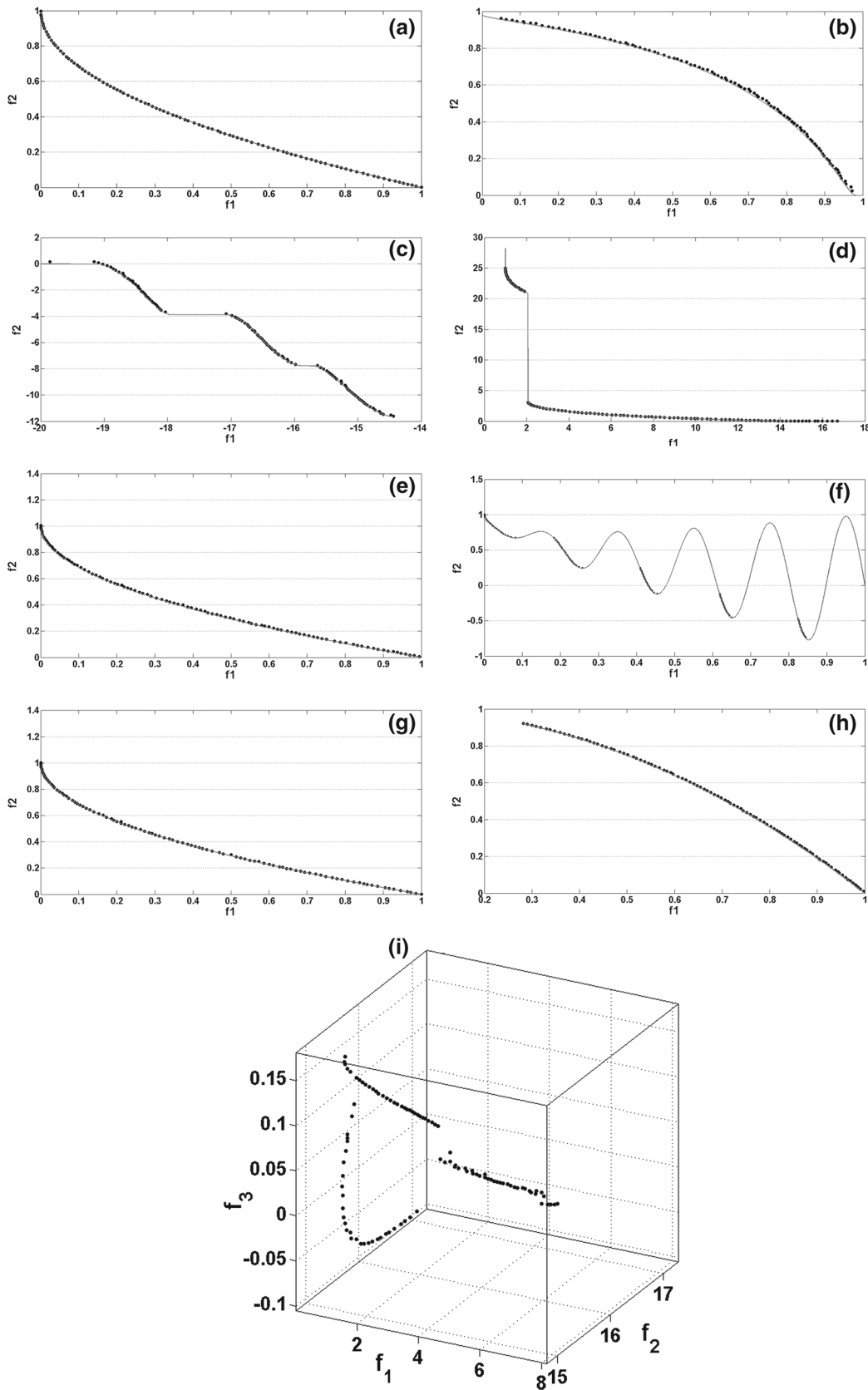


Fig. 5 Comparisons of optimal Pareto fronts and generated Pareto front using the MOWCA for: **a** DEB, **b** FON, **c** KUR, **d** POL, **e** ZDT1, **f** ZDT3, **g** ZDT4, **h** ZDT6, and **i** VNT (solid lines and dot points represent the optimal and generated (obtained) Pareto fronts, respectively)

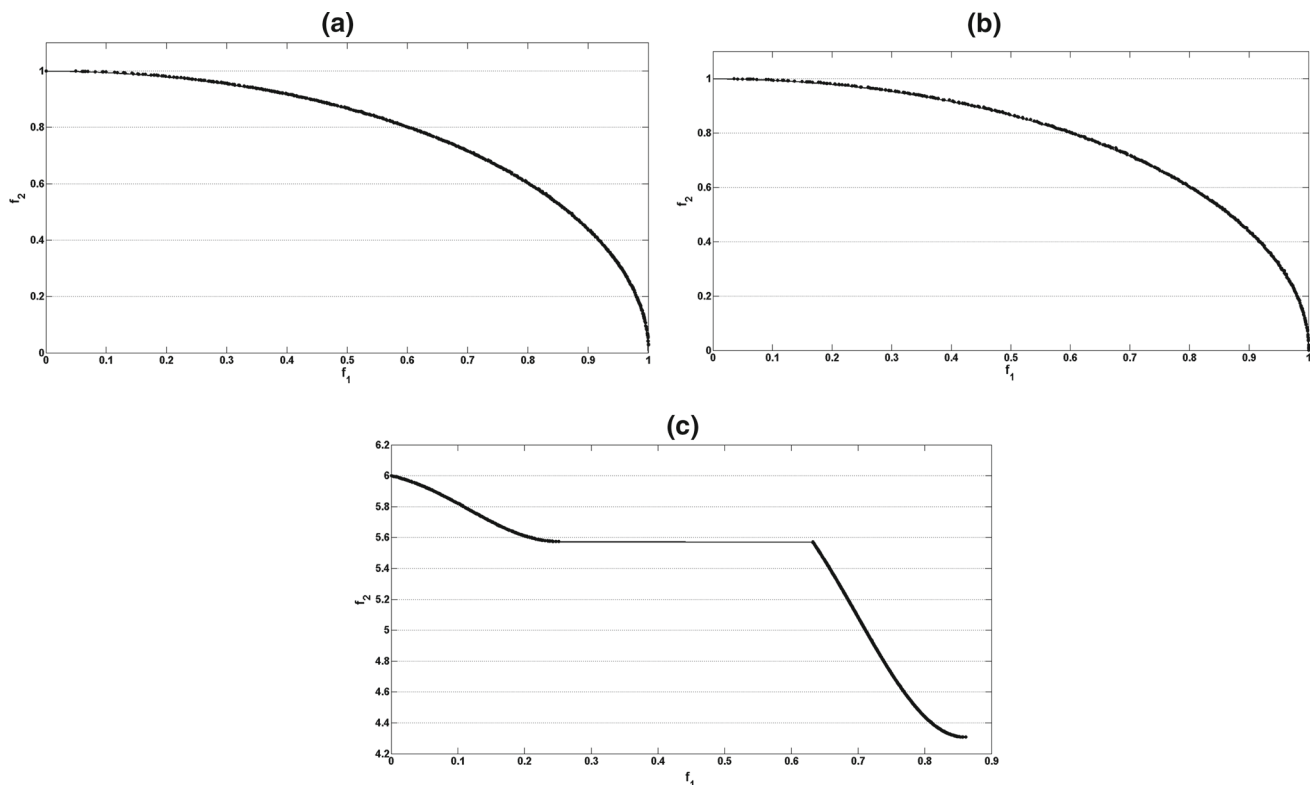


Fig. 6 Comparisons of optimal Pareto fronts and generated Pareto front using the MOWCA for the bi-objective function: **a** DTLZ 2, **b** DTLZ 4, and **c** DTLZ 7 (solid lines and dot points represent the optimal and generated (obtained) Pareto fronts, respectively)

(2) *Test problem 2–FON*: Fonseca and Fleming’s function (FON) is an eight-design variable function suggested as follows (Fonseca and Fleming 1993):

$$\begin{aligned}
 \text{FON : min } & \begin{cases} f_1(X) = 1 - \exp\left(-\sum_{i=1}^8 \left(x_i - \frac{1}{\sqrt{8}}\right)^2\right) \\ f_2(X) = 1 - \exp\left(-\sum_{i=1}^8 \left(x_i + \frac{1}{\sqrt{8}}\right)^2\right) \end{cases} \\
 \text{where } & -2 < x_i < 2, \quad i = 1, 2, 3, \dots, 8, \quad (25)
 \end{aligned}$$

The optimal Pareto front for this bi-objective problem is $x_i^* = [-1/\sqrt{8}, 1/\sqrt{8}]$ for $i = 1, 2, 3, \dots, 8$.

(3) *Test problem 3–POL*: This function was first introduced by Poloni (1997) which has been widely analyzed in the literature (Deb et al. 2002a; Kaveh and Laknejadi 2011). The mathematical formulation proposed by Poloni (1997) is given as follows:

$$\begin{aligned}
 \text{POL : min } & \begin{cases} f_1(X) = [1 + (A_1 - B_1)^2 + (A_2 - B_2)^2] \\ f_2(X) = [(x_1 + 3)^2 + (x_2 + 1)^2] \\ A_1 = 0.5 \sin 1 - 2 \cos 1 + \sin 2 - 1.5 \cos 2 \\ A_2 = 1.5 \sin 1 - \cos 1 + 2 \sin 2 - 0.5 \cos 2 \\ B_1 = 0.5 \sin(x_1) - 2 \cos(x_1) + \sin(x_2) - 1.5 \cos(x_2) \\ B_2 = 1.5 \sin(x_1) - \cos(x_1) + 2 \sin(x_2) - 0.5 \cos(x_2) \end{cases} \\
 \times & \begin{cases} \\ \\ \\ \\ \\ \end{cases} \\
 \text{where } & -\pi < x_1, x_2 < \pi. \quad (26)
 \end{aligned}$$

The Pareto optimal front for the POL function is non-convex and discontinuous.

(4) *Test problem 4–KUR*: This problem, presented by Kursawe (1991), has three design variables having non-convex and discontinuous Pareto optimal front. The KUR’s mathematical formulation is as follows (Kursawe 1991):

$$\begin{aligned}
 \text{KUR : min } & \begin{cases} f_1(X) = \sum_{i=1}^{n-1} \left(-10 \exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right)\right) \\ f_2(X) = \sum_{i=1}^n \left(|x_i|^{0.8} + 0.5 \sin x_i^3\right) \end{cases} \\
 \text{where } & -5 < x_i < 5, \quad i = 1, 2, 3. \quad (27)
 \end{aligned}$$

(5) *Test problem 5–VNT*: This problem is a three-dimensional problem in objective space suggested by Viennet et al. (1995). This problem has previously been investigated by many researchers (Freschi and Repetto 2006; Gao and Wang 2010) and is formulated as follows (Viennet et al. 1995):

$$\begin{aligned}
 \text{VNT = min } & \begin{cases} f(X) = 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2) \\ f(X) = (3x_1 - 2x_2 + 4)^2/8 + (x_1 - x_2 + 1)^2/27 + 15 \\ f(X) = (x_1^2 + x_2^2 + 1)^{-1} - 1.1 \exp(-x_1^2 - x_2^2) \end{cases} \\
 \text{where } & -3 \leq x_1, x_2 \leq 3. \quad (28)
 \end{aligned}$$

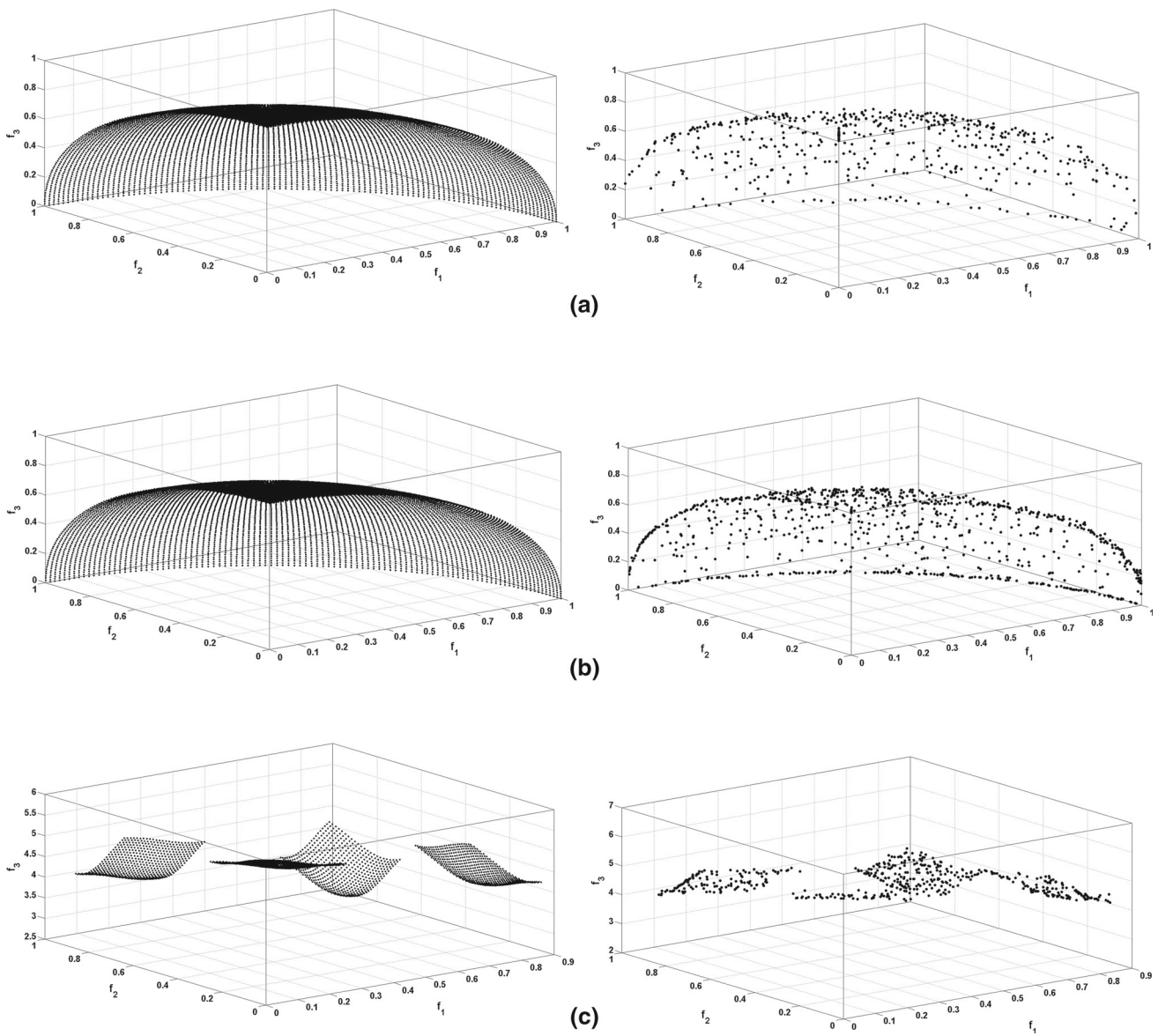


Fig. 7 Comparisons of optimal Pareto fronts and generated Pareto front using the MOWCA for the three-objective function: **a** DTLZ 2, **b** DTLZ 4, and **c** DTLZ 7 (*left* and *right* sides represent the optimal and generated (obtained) Pareto fronts, respectively)

It is worth mentioning that the discontinuous Pareto optimal set and having several local Pareto fronts are considered to be challenging features of the VNT problem (Gao and Wang 2010).

(6) *Test problem 6–ZDT1*: The ZDT1 function, was suggested by Zitzler et al. (2000), and has been extensively investigated (Deb et al. 2002a; Kaveh and Laknejadi 2011). This problem is described as follows:

$$\text{ZDT1} : \min \begin{cases} f_1(X) = x_1 \\ f_2(X) = g(X)[1 - \sqrt{x_1/g(X)}] \\ g(X) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n - 1) \end{cases} \quad (29)$$

where $0 < x_i < 1, i = 1, 2, 3, \dots, 30$.

The ZDT1 problem has 30 design variables and its Pareto optimal front is convex and defined as $x_1 \in [0, 1], x_i = 0$, for $i = 2, \dots, 30$.

Table 9 Mathematical formulations of the DTLZ series problems

MOPs	Objective function
DTLZ 2	$f_1(x) = (1 + g(x_M)) \cos(x_1\pi/2) \cos(x_2\pi/2) \dots \cos(x_{M-2}\pi/2) \cos(x_{M-1}\pi/2)$
	$f_2(x) = (1 + g(x_M)) \cos(x_1\pi/2) \cos(x_2\pi/2) \dots \cos(x_{M-2}\pi/2) \sin(x_{M-1}\pi/2)$
	\vdots
	$f_{M-1}(x) = (1 + g(x_M)) \cos(x_1\pi/2) \sin(x_2\pi/2)$ $f_M(x) = (1 + g(x_M)) \sin(x_1\pi/2)$ $g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2$
DTLZ 4	$f_1(x) = (1 + g(x_M)) \cos(x_1^a\pi/2) \cos(x_2^a\pi/2) \dots \cos(x_{M-2}^a\pi/2) \cos(x_{M-1}^a\pi/2)$
	$f_2(x) = (1 + g(x_M)) \cos(x_1^a\pi/2) \cos(x_2^a\pi/2) \dots \cos(x_{M-2}^a\pi/2) \sin(x_{M-1}^a\pi/2)$
	\vdots
	$f_{M-1}(x) = (1 + g(x_M)) \cos(x_1^a\pi/2) \sin(x_2^a\pi/2)$ $f_M(x) = (1 + g(x_M)) \sin(x_1^a\pi/2)$ $g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2$
DTLZ 7	$f_1(x) = x_1$ $f_2(x) = x_2$ \vdots $f_{M-1}(x) = x_{M-1}$
	$f_M(x) = (1 + g(x_M))h(f_1, f_2, \dots, f_{M-1}, g)$
	$g(x_M) = 1 + \frac{9}{ x_M } \sum_{x_i \in x_M} x_i$
	$h(f_1, f_2, \dots, f_{M-1}, g) = M - \sum_{i=1}^{M-1} \left[\frac{f_i}{1+g(x_M)} (1 + \sin(3\pi f_i)) \right]$

(7) *Test problem 7–ZDT3*: The ZDT3 problem, suggested by Zitzler et al. (2000), has 30 design variables with a non-convex and discontinuous Pareto optimal front. The mathematical formulation of the ZDT3 problem is as follows (Zitzler et al. 2000):

$$\begin{aligned} \text{ZDT3 : min} \\ \begin{cases} f_1(X) = x_1 \\ f_2(X) = g(X)[1 - \sqrt{x_1/g(X)} - \frac{x_1}{g(X)} \sin(10\pi x_1)] \\ g(X) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n - 1) \end{cases} \end{aligned} \tag{30}$$

where $0 < x_i < 1, i = 1, 2, 3, \dots, 30$.

The Pareto optimal front is defined as $x_1 \in [0, 1], x_i = 0$, for $i = 2, 3, 4, \dots, 30$.

(8) *Test problem 8–ZDT4*: The ZDT4 problem, proposed by Zitzler et al. (2000), has 10 design variables having several local Pareto fronts. This problem is given as follows:

$$\begin{aligned} \text{ZDT4 : min} \\ \begin{cases} f_1(X) = x_1 \\ f_2(X) = g(X)[1 - \sqrt{x_1/g(X)}] \\ g(X) = 1 + 10(n - 1) + \sum_{i=2}^n [x_i^2 - 10 \cos(4\pi x_i)] \end{cases} \end{aligned} \tag{31}$$

where $x_1 \in [0, 1]$ and $x_i \in [-5, 5]$, for $i = 2, 3, \dots, 10$. In addition, the optimal Pareto front is defined as $x_1 \in [0, 1], x_i = 0$, for $i = 2, 3, 4, \dots, 10$.

(9) *Test problem 9–ZDT6*: The ZDT6, introduced by Zitzler et al. (2000), has 10 design variables with a non-convex Pareto optimal front. The mathematical formulation of this problem is given as follows:

$$\begin{aligned} \text{ZDT6 : min} \\ \begin{cases} f_1(X) = 1 - \exp(-4x_1) \sin^6(6\pi x_1) \\ f_2(X) = g(X)[1 - (f_1(X)/g(X))^2] \\ g(X) = 1 + 9 \left[\left(\sum_{i=2}^n x_i \right) / (n - 1) \right]^{0.25} \end{cases} \end{aligned} \tag{32}$$

where $0 < x_i < 1, i = 1, 2, 3, \dots, 10$.

The Pareto optimal front is characterized as $x_1 \in [0, 1], x_i = 0$ for i from 2 to 10 for test problem 9.

(10) *Test problem 10–DTLZ series*

The DTLZ series, proposed by Deb et al. (2002b), known as scalable problem, are considered in this paper. The mathematical formulation of DTLZ problems are given in Table 9. The Pareto optimal front for all DTLZ problems in this paper is defined as $x_i = 0.5$ in which $i \in x_M$. The design space for considered DTLZ problems is between zero and one.

Two and three objective functions ($M = 2$ and 3) are considered for the DTLZ series given in Table 9. The DTLZ series are minimization problems and the number of design

variables for these problems is calculated as follows:

$$n = M + |x_M| - 1, \quad (33)$$

where n and M are the number of design variables and number of objective functions, respectively. Also, $|x_M|$ is set to 10 for all considered problems in this paper.

References

- Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. *IEEE Congress on Evolutionary Computation*, Singapore, pp 4661–4667
- Blum C, Andrea R (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput Surv* 35(3):268–308
- Coello CAC, Lechuga MS (2002) MOPSO: A proposal for multiple objective particle swarm optimization. In: *Proceedings of the congress on evolutionary computation (CEC'2002)*, Honolulu, vol 1, pp 1051–1056
- Coello CAC (2000) An updated survey of GA-based multi-objective optimization techniques. *ACM Comput Surv* 32(2):109–143
- Coello CAC, Veldhuizen DAV, Lamont G (2002) *Evolutionary algorithms for solving multi-objective problems.*, Genetic Algorithms and Evolutionary Computation Kluwer, Dordrecht
- Coello CAC (2004) Handling multiple objectives with particle swarm optimization. *IEEE T Evolut comput* 8(3):256–279
- Coello CAC, Cruz Cortés N (2005) Solving multiobjective optimization problems using an artificial immune system. *Genet Program Evol M* 6:163–190
- Deb K (2001) *Multi-objective optimization using evolutionary algorithms.* Wiley, New York
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002a) A fast and elitist multi objective genetic algorithm: NSGA-II. *IEEE Trans Evolut Comput* 6(2):182–197
- Deb K (2002) Multi-objective genetic algorithms: problem difficulties and construction of test problems. *Evol Comput* 7:205–230
- Deb K, Thiele L, Laumanns M, Zitzler E (2002) Scalable multi-objective optimization test problems. In: *Proceedings of IEEE Conference on Evolutionary Computation*, pp 825–830
- Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput Struct* 110–111:151–166
- Fonseca CM, Fleming PJ (1993) Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: Forrest S (ed) *Proceedings of the fifth international conference on genetic algorithms.* Morgan Kaufman, San Mateo, pp 416–423
- Freschi F, Repetto M (2006) VIS: an artificial immune network for multi-objective optimization. *Eng Optim* 38(8):975–996
- Gao J, Wang J (2010) WBMOAIS: a novel artificial immune system for multiobjective optimization. *Comput Oper Res* 37:50–61
- Glover FW, Kochenberger GA (2003) *Handbook of metaheuristics.* Kluwer, Dordrecht
- Haupt RL, Haupt SE (2004) *Practical genetic algorithms*, 2nd edn. John Wiley, New York
- Holland J (1975) *Adaptation in natural and artificial systems.* University of Michigan Press, Ann Arbor
- Kaveh A, Laknejadi K (2011) A novel hybrid charge system search and particle swarm optimization method for multi-objective optimization. *Expert Syst Appl* 38(12):15475–15488
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of the IEEE international conference on neural networks.* Perth, Australia, pp 1942–1948
- Knowles JD, Corne DW (2000) Approximating the nondominated front using the Pareto archived evolution strategy. *Evol Comput* 8(2):149–172
- Kursawe F (1991) A variant of evolution strategies for vector optimization. In: *Lecture Notes in Computer Science.* In: *Proceedings of the Parallel Problem Solving From Nature, PPSN I*, vol 496, pp 193–197
- Lin Q, Chen J (2013) A novel micro-population immune multiobjective optimization algorithm. *Comput Oper Res* 40(6):1590–1601
- Mahmoodabadi MJ, Adljooy Safaie A (2013) A novel combination of particle swarm optimization and genetic algorithm for pareto optimal design of a five-degree of freedom vehicle vibration model. *Appl Soft Comput* 13:2577–2591
- Mostaghim S, Teich J (2003) Strategies for finding good local guides in multi objective particle swarm optimization (MOPSO). In: *Proceedings of the IEEE swarm intelligence symposium*, pp 26–33
- Osman IH, Laporte G (1996) Metaheuristics: a bibliography. *Ann Oper Res* 63:513–623
- Poloni C (1997) *Hybrid GA for multiobjective aerodynamic shape optimization in genetic algorithms.*, Engineering and Computer Science Wiley, New York
- Pradhan PM, Panda G (2012) Solving multiobjective problems using cat swarm optimization. *Expert Syst Appl* 39:2956–2964
- Sierra MR, Coello CAC (2005) Improving PSO-based multi objective optimization using crowding, mutation and e-dominance. In: *Proceedings of evolutionary multi-criterion optimization conference.* Guanajuato, Mexico, pp 505–519
- Srinivas N, Deb K (1995) Multi objective function optimization using nondominated sorting genetic algorithms. *Evol Comput* 2(3):221–248
- Viennet R, Fontiex C, Marc I (1995) New multicriteria optimization method based on the use of a diploid genetic algorithm: example of an industrial problem. In: *Proceedings of Artificial Evolution.* Brest, France, pp 120–127
- Wang L, Zhong X, Liu M (2012) A novel group search optimizer for multi-objective optimization. *Expert Syst Appl* 39:2939–2946
- Wang L, Zhong X, Liu M (2012) A novel group search optimizer for multi-objective optimization. *Expert Syst Appl* 39(3):2939–2946
- Wang Y, Zeng J (2013) A multi-objective artificial physics optimization algorithm based on ranks of individuals. *Soft Comput* 17:939–952
- Zhang B, Ren W, Zhao L, Deng X (2009) Immune system multiobjective optimization algorithm for DTLZ problems. In: *Fifth international conference on natural computation*, pp 603–609
- Zitzler E, Thiele L (1999) Multi objective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans Evolut Comput* 3(4):257–271
- Zitzler E, Deb K, Thiele L (2000) Comparison of multi-objective evolutionary algorithms: empirical results. *Evol Comput* 8(2):173–195
- Zitzler E, Laumanns M, Thiele L (2001) SPEA2: Improving the strength Pareto evolutionary algorithm. *Swiss Federal Institute Technology, Zurich, Switzerland, TIK Report*, vol 103, pp 1–21