Water Flow-Like Optimization Algorithm for Multi-objective Continuous Optimization Problem

Feng-Cheng Yang and Bryan Ni

Abstract This paper presents a WFA for Multi-objective Continuous Optimization Problems. Namely WFA4MC. In order to prove WFA4MC performances precisely, this research proposes Correctness and Coverness to measure non-dominated solutions in ZDT functions. Besides, the Generational Distance is used in the comparison with other heuristic algorithms. The result showed that based on the same limit of the number of objective function calls, the WFA4MC outperform than others.

Keywords Correctness • Multi-objective continuous optimization problem • ZDT functions

1 Introduction

Multiple-Objective continuous optimization problem has been applied in widely many fields such as science, engineering and management. However, solving such problem is regarded as a challenge because of the high dimensionality and complex objectives. Some heuristic-based methods capable with dealing such difficulties have been developed such as AMOPSO [1], NSGAII [2] and MOEPSO [3]. Such kind of methods has constant number of solution agent. For these methods, once the solutions tend to converge during the searching process, redundant computation is arisen.

This paper proposed a novice approach, i.e. Water Flow-like Algorithm for Multi-objective Continuous Optimization Problems (WFA4MC). This approach tactically adjusts the solution agent number for efficient search and is based

F.-C. Yang (✉) • B. Ni
Institute of Industrial Engineering, National Taiwan University, Taipei, Taiwan
e-mail: d00546005@ntu.edu.tw

E. Qi et al. (eds.), *Proceedings of 2013 4th International Asia Conference on Industrial Engineering and Management Innovation (IEMI2013)*,
DOI 10.1007/978-3-642-40060-5_46, © Springer-Verlag Berlin Heidelberg 2014

on Water Flow-like Algorithm (WFA) proposed by Yang and Wang [4]. WFA mimics the water flow traversing on the terrain with regard to several phenomena such as flow moving, flow splitting, evaporation and precipitation. Single water corresponds to a solution agent and the terrain is related to the solution space. A single moving water flow represents searching behavior of a solution agent. Flow splitting increases the solution agent number to intensify the searching direction and computation. While flow merging decreases the solution agent number to remove the resemble solutions. Evaporation decreases the solution agent number for the removal of weak solution. When the air is saturated with water, precipitation starts. Additional solution agents are generated for random search.

Evaluation of non-dominant solutions in a proper manner is another issue. Generational distance proposed by Zitzler et al. [5] is commonly in use for measuring the Euclidean distance between current obtained non-dominant solution and the true optimal p solution.

However, Generation distance does not provide the information that how well the non-dominant solutions are not dominant by Pareto-front solution Also it cannot detect the solutions narrowly distributed in Pareto front. This paper further proposed two more performance indices, i.e. Correctness and Coverness. Correctness evaluates how many current obtained non-dominant solutions are not dominant from true optimal non-dominant solutions. Coverness quantifies how well the current-obtained non-dominant solutions cover the true optimal non-dominant solutions.

2 Multi-objective Continuous Optimization Problem

Assume that a solution to the multi-objective optimization problem consists of p indexed real numbers and is defined as a p-dimensional vector $\mathbf{x} = [x_1 x_2 \cdots x_p]$. Each variable is bounded by given lower and upper limits, such that $\underline{x}_k \le x_k \le \overline{x}_k$; $x_k, \underline{x}_k, \overline{x}_k \in \mathbb{R}$, $\forall\, k \in \{1,2, \ldots ,p\}$. Suppose that there are g objective functions defined for the problem: $f_j(\mathbf{x}) : \mathbb{R}^p \to \mathbb{R}; j = 1, 2, \cdots, g$. Objective values of the solution $\mathbf{x}$ can be denoted by a g-dimensional vector $\mathbf{f} = [f_1 f_2 \cdots f_g] \equiv [f_1(\mathbf{x}) f_2(\mathbf{x}) \cdots f_g(\mathbf{x})]$. The goal of this optimization problem is to either maximize or minimize these objective values.

Instead of providing an optimal solution, a set of non-dominated solutions are usually presented to the user for a multi-objective optimization problem. Let the set be $\widehat{X}$. Without loss of generality, assume that the goal of the optimization problem is to minimize all objective values. Therefore, for a pair of non-dominated solutions $\mathbf{x}^i$ and $\mathbf{x}^{i'}$, $\mathbf{x}^i, \mathbf{x}^{i'} \in \widehat{X}$, there is at least one component j such that $f^i_j < f^{i'}_j$, where $\mathbf{f}^i = [f^i_1 f^i_2 \cdots f^i_g] \equiv [f_1(\mathbf{x}^i)\ f_2(\mathbf{x}^i) \cdots f_g(\mathbf{x}^i)]$ and $\mathbf{f}^{i'} = \left[f^{i'}_1 f^{i'}_2 \cdots f^{i'}_g\right] \equiv \left[f_1\left(\mathbf{x}^{i'}\right)\ f_2\left(\mathbf{x}^{i'}\right) \cdots f_g\left(\mathbf{x}^{i'}\right)\right]$ are their objective vectors respectively. In other words, non-dominated solutions should mutually have at least one objective value superior than that of others. Let operation Dominated$\left(\mathrm{X}^i, \mathrm{X}^{i'}\right)$ be used to test whether the first solution, $\mathbf{x}^i$, is dominated by the second one, $\mathbf{x}^{i'}$. Then

$$\text{Dominated}\left(\mathrm{X}^i, \mathrm{X}^{i'}\right) = \begin{cases} \text{false, if } \exists j \in \{1, 2, \ldots, g\} \ni f_j^i < f_j^{i'} \\ \text{true, otherwise} \end{cases} \tag{1}$$

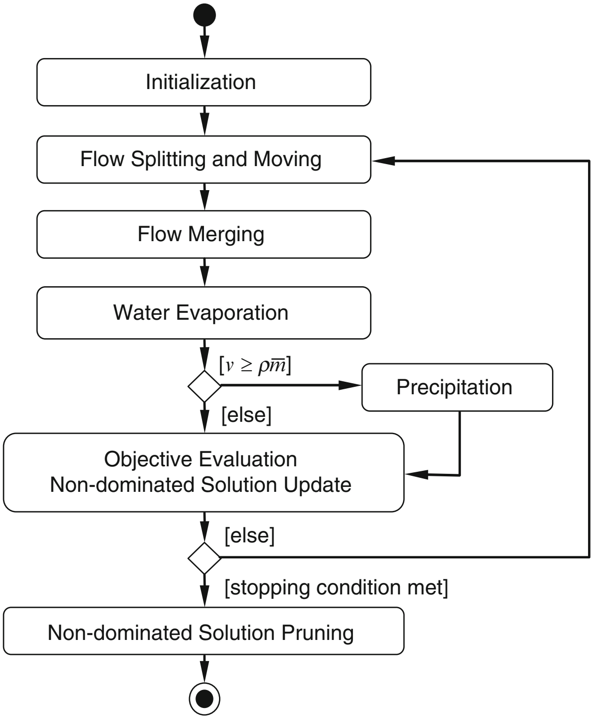
When a candidate solution $\mathbf{x}^i$ is considered for joining the non-dominated solution set $\widehat{X}$, it will be subject to this domination test against all solutions in the set. Once the solution is not dominated by any solution in the set, it is recognized as a new member of the set.

Although, it seems that we can provide as many as non-dominated solutions to the user for decision making, people prefers a finite number of solutions for easier application. Therefore, suppose $\overline{s}$ is the required number of non-dominated solution in $\widehat{X}$ and s is the number of solutions in $\widehat{X}$. When $s > \overline{s}$, $s - \overline{s}$ "inferior" solutions should be expelled from $\widehat{X}$. Moreover, the objective vectors $\mathbf{f}$ of the solutions in $\widehat{X}$ are expected to cover the objective space as large as possible to provide a wider spectrum for decision making. This means the $\overline{s}$ non-dominated solutions should be separated far from each other on the objective space. A solution with smaller distances to its neighbors on the solution space is then regarded as inferior one, which is a candidate for solution pruning. We propose a pruning method that sorts the solutions in $\widehat{X}$ with a jamming distance. The $s - \overline{s}$ solutions with smaller jamming distances are pruned away; details will be given later.

3 WFA Model for the Continuous Multi-objective Optimization Problems

The developed computation model based on the water flow like algorithm consists of a set of water flows represented by their locations, $X = \{\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^n\}$, where n is the number of flows of the current flow set X. The location of flow i is $\mathbf{x}^i = [x_1^i \ x_2^i \cdots x_p^i]$, representing a solution to the optimization problem. For easier presentation, hereafter we alternatively use index i or location $\mathbf{x}^i$ to refer to a flow. The set of water flows is then recursively subject to a serial of water flow operations to yield a new set of flows to enhance the objective values of the solutions represented by their locations. Note that the set of flows X is updated both in the number of flows and their locations, subjecting to water splitting and merging operations. The flow splitting operation on a flow depends on the objective enhancement of the current location with respective to its previous location. Therefore, a virtual parent flow with location $\tilde{\mathbf{x}}^i = \left[\tilde{x}_1^i \tilde{x}_2^i \cdots \tilde{x}_p^i\right]$ is defined for flow i. Chronically, $\tilde{\mathbf{x}}^i$ can be regarded as the previous location of flow i. When more than one sub flow is split, these sub flows share the same parent. Let $\mathbf{f}^i = [f_1^i f_2^i \cdots f_g^i]$ and $\tilde{\mathbf{f}}^i = \left[\tilde{f}_1^i \tilde{f}_2^i \cdots \tilde{f}_g^i\right]$ be the vectors of objective values of flow i and its parent, respectively. In addition to the location and objective values, a mass property m^i is

Fig. 1 The computation flow of the proposed WFA for multi-objective optimization problems



defined for flow i, which is used to determine whether a flow survives to the next flowing iteration and to regulate the precipitation operation.

For final optimal solution presentation, a non-dominated solution set is constructed, which is denoted by $\widehat{X} = \left\{\widehat{\mathbf{x}}^1, \widehat{\mathbf{x}}^2, \cdots, \widehat{\mathbf{x}}^s\right\}$, where s is number of non-dominated solutions found. Each solution in $\widehat{X}$ is cloned from the set of flows in X after exclusive domination tests.

Figure 1 illustrates the computation flow of the presented WFA for Multi-objective Continuous Optimization Problems. Notice that several stopping conditions can be provided and specified by the user to stop the solution evolution. In our model, iteration limits, elapsed CPU execution time, and number of objective evaluations are provided to stop the solution evolution.

3.1 Initialization

Initially, only one flow is defined in the flow set; i.e., $n = 1$ and $X = \{\mathbf{x}^1\}$. Let $m^1 = \overline{m}$, where $\overline{m}$ is a user specified mass that will regulate flow splitting. Components of location $\mathbf{x}^1$ are randomly set by

$$x_k^1 \leftarrow\sim U\left(0,1\right)\left(\overline{x}_k - \underline{x}_k\right) + \underline{x}_k, k = 1, 2, \cdots, p$$

where $\sim U(0,1)$ is a uniformly distributed random real value bounded within [0,1]. Since the initial flow is not split from any other flow, we randomly assign the location of its parent flow $\tilde{\mathbf{x}}^1$ as

$$\tilde{x}_k^1 \leftarrow \sim U(0,1)\left(\overline{x}_k - \underline{x}_k\right) + \underline{x}_k, k = 1, 2, \cdots, p$$

The objective vectors $\mathbf{f}^1 = [f_1^1 f_2^1 \cdots f_g^1]$ and $\tilde{\mathbf{f}}^1 = \left[\tilde{f}_1^1 \tilde{f}_2^1 \cdots \tilde{f}_g^1\right]$ of both locations (solutions) $\mathbf{x}^1$ and $\tilde{\mathbf{x}}^1$ are then evaluated and solution domination test is conducted to constitute the initial non-dominated solution set. Therefore,

$$\widehat{X} = \begin{cases} \left\{\mathbf{x}^1\right\}, \text{if Dominated}\left(\tilde{\mathbf{x}}^1, \mathbf{x}^1\right) = \text{true} \\ \left\{\tilde{\mathbf{x}}^1\right\}, \text{else if Dominated}\left(\mathbf{x}^1, \tilde{\mathbf{x}}^1\right) = \text{true} \\ \left\{\tilde{\mathbf{x}}^1, \mathbf{x}^1\right\}, \text{otherwise} \end{cases} \tag{2}$$

The amount of objective improvement influences the flow moving and flow splitting, both in the location change and the number of sub flows split. To evaluate the amount of objective value improvement, the largest and smallest values of each objective value obtained so far are updated and recorded during the water flowing operation. Let $\underline{f}_j$ and $\overline{f}_j$ be the smallest and largest values of objective values $f_j(\mathbf{x})$ obtained so far. Therefore, initially

$$\underline{f}_j = \min\left(f_j^1, \tilde{f}_j^1\right); j = 1, 2, \cdots, g \tag{3}$$

and

$$\overline{f}_j = \max\left(f_j^1, \tilde{f}_j^1\right); j = 1, 2, \cdots, g \tag{4}$$

The j-th objective value of water flow i can be normalized as

$$f'^i_j = \frac{f_j^i - \underline{f}_j}{\overline{f}_j - \underline{f}_j} \tag{5}$$

In addition, since there are g objective values, a weighted sum of objective value F^i is defined for flow i as

$$F^i = \sum\nolimits_{j=1}^{g} w_j f'^i_j = \sum\nolimits_{j=1}^{g} w_j \frac{f_j^i - \underline{f}_j}{\overline{f}_j - \underline{f}_j} \tag{6}$$

where w_j is a user specified weight for objective function $f_j(\cdot)$. Similarly, the weighted objective value of the parent flow of flow i is

$$\tilde{F}^i = \sum\nolimits_{j=1}^{g} w_j \frac{\tilde{f}_j^i - \underline{f}_j}{\overline{f}_j - \underline{f}_j} \tag{7}$$

We let the amount of objective improvement determine the number of sub flows in the operation of flow splitting. To help evaluating the improvement, an improvement threshold θ is defined in our WFA model. When the objective improvement of a flow exceeds the threshold, no flow splitting is executed. The threshold is defined as the average of the difference between weighted objective values of all flows and their parents; i.e.,

$$\theta = \frac{\Delta F}{N} \tag{8}$$

where N is the total number of flows constructed so far and ΔF is cumulated objective differences of these flows. Initially

$$\Delta F = \left|F^1 - \tilde{F}^1\right| \text{ and } N = 1 \text{ since } X = \left\{\mathbf{x}^1\right\} \tag{9}$$

3.2 Flow Splitting and Moving

Flow splitting and moving operation is sequentially conducted for each flow in the current flow set X. Firstly, the objective value improvement for each flow is evaluated to determine the number of sub flows. Once the number is determined, the location advancements of all sub flows follow to create sub flows. The number of sub flows split from a flow is restricted by a split limit $\overline{n}$, whose value is set by the user to avoid an exponentially increased number of flows. In general, a smaller improvement amount of objective value is designated to split more sub flows for a wider solution exploration. On the other hand, a larger amount should bind the same flow for neighborhood exploitation. Therefore, we define the number of sub flows split from flow i as

$$n^i = \begin{cases} 1, \text{if } \left|\tilde{F}^i - F^i\right| > \theta \\ \text{Round}\left(\text{Pow}\left(\overline{n}, \left(1 - \frac{\left|\tilde{F}^i - F^i\right|}{\theta}\right)\right)\right), \text{otherwise} \end{cases} \tag{10}$$

Assume that $\mathbf{x}_1^i, \mathbf{x}_2^i, \cdots, \mathbf{x}_{n^i}^i$ are locations of the n^i sub flows split from flow i and

$$\mathbf{x}_q^i = \left[x_{q1}^i \;\; x_{q2}^i \cdots x_{qp}^i\right], q = 1, 2, \cdots, n^i \tag{11}$$

Where

$$\underline{x}_k \leq x_{qk}^i \leq \overline{x}_k, k = 1, 2, \cdots, p \tag{12}$$

For a multi-objective optimization problem, the solution evolution is a forwarding process toward non-dominated solutions. Locations of the sub flows split from flow i are stochastically determined by the corresponding variables of flow i, parent flow of flow i, and a randomly selected flow from the non-dominated set. For each sub flow, a referential non-dominated solution is randomly selected first. Let location vector $\widehat{\mathbf{x}}^{i'}$ be the selected non-dominated solution for a sub flow of flow i, where $i' = \text{RandomInteger}(1, m)$.

Note that RandomInteger(*min*, *max*) returns a random integer that follows uniform distribution between integers *min* and *max*.

The value of the k-th variable x^i_{qk} is then set stochastically with respective to a target value ξ that is selected from the k-th component of either $\mathbf{x}^i$, $\tilde{\mathbf{x}}^i$, or $\widehat{\mathbf{x}}^{i'}$. If flow i has a clone in the non-dominated solution set $\widehat{X}$, ξ is directly selected from $\mathbf{x}^i$. Otherwise, if the parent flow is a non-dominated solution ξ is selected from $\tilde{\mathbf{x}}^i$, since flow i is inferior to its parent. On the other hand, if both $\mathbf{x}^i$ and $\tilde{\mathbf{x}}^i$ are not non-dominated solutions, $\widehat{\mathbf{x}}^{i'}$ is targeted. Therefore,

$$\xi = \begin{cases} x^i_k, \text{if } \mathbf{x}^i \in \widehat{X} \\ \tilde{x}^i_k, \text{else if } \tilde{\mathbf{x}}^i \in \widehat{X} \\ \widehat{x}^{i'}_k, \text{othewise} \end{cases} \tag{13}$$

Value of x^i_{qk} is then stochastically set to the target ξ or a random value bounded by $\left[\underline{\xi}, \overline{\xi}\right]$ with an equal probability. Bounds

$$\underline{\xi} = \begin{cases} \frac{1}{2}\left(\underline{x}_k + \xi\right), \text{if } \xi = \widehat{x}^{i'}_k \\ \max\left(\underline{x}_k, \xi - \left|\xi - \widehat{x}^{i'}_k\right|\right), \text{otherwise} \end{cases} \tag{14}$$

$$\overline{\xi} = \begin{cases} \frac{1}{2}\left(\overline{x}_k + \xi\right), \text{if } \xi = \widehat{x}^{i'}_k \\ \min\left(\overline{x}_k, \xi + \left|\xi - \widehat{x}^{i'}_k\right|\right), \text{otherwise} \end{cases} \tag{15}$$

Note that bounds $\underline{\xi}$ and $\overline{\xi}$ are constrained by the given coordinate bounds $\underline{x}_k$ and $\overline{x}_k$, as well as the target value ξ and $\widehat{x}^{i'}_k$. The reduced bounds are then used to generate location values centering around ξ. Therefore,

$$x^i_{qk} = \begin{cases} \xi, \text{if } \sim U(0, 1) \le 0.5 \\ \text{RandomReal}\left(\underline{\xi}, \overline{\xi}\right), \text{otherwise} \end{cases} \tag{16}$$

Where RandomReal(*min*,*max*) returns a uniformly distributed random real value bounded by *min* and *max*. Note that the location vectors of all sub flows are sequentially determined while the component values are determined one after the other.

The last step in flow splitting is to distribute the mass to sub flows and assign their parents. Two types of mass distribution are provided. Each type is dedicated to a type of flow merging operation. Our WFA for multi-objective optimization problems presents two types of flow merging: one is based on the closeness of flow locations and the other is on the similarity of objective values.

The first type of mass distribution evenly distributes the mass to all the sub flows. Let m_q^i be the mass of the q-th sub flow split form flow i; then

$$m_q^i = \frac{1}{n^i} m^i, q = 1, 2, \cdots, n^i \tag{17}$$

The second type of mass distribution requires that the objective values of all sub flows must be evaluated first. Assume that the objective vector of the q-th sub flow split form flow i is

$$\mathbf{f}_q^i \equiv \left[f_1\left(\mathbf{x}_q^i\right) \cdots f_g\left(\mathbf{x}_q^i\right) \right] = \left[f_{q1}^i \; f_{q2}^i \cdots f_{qg}^i \right] \tag{18}$$

and the weighted sum of objective value is

$$F_q^i = \sum\nolimits_{j=1}^{g} w_j \frac{f_{qj}^i - \underline{f}_j}{\overline{f}_j - \underline{f}_j}, q = 1, 2, \cdots, n^i \tag{19}$$

Notice that the largest and smallest objective values, $\overline{f}_j$ and $\underline{f}_j$, are updated before evaluating the objective values of all sub flows. The amount of mass received by a sub flow is proportional to the amount of objective improvement from its parent; e.g., $F^i - F_q^i$ for the q-th sub flow. Notice that the amount of improvement might be negative. We let the sub flow with the largest improvement receive twice the mass of the one with the lowest. Therefore for sub flows split from flow i, the largest and smallest objective improvements are determined first by

$$\overline{\delta} = \max_{q=1,2,\cdots,n^i} \left\{ F^i - F_q^i \right\} \tag{20}$$

and

$$\underline{\delta} = \min_{q=1,2,\cdots,n^i} \left\{ F^i - F_q^i \right\} \tag{21}$$

The relative portion of mass splitting for the q-th sub flow is

$$\rho_q = \left(F^i - F_q^i - \underline{\delta}\right) + \left(\overline{\delta} - \underline{\delta}\right) \tag{22}$$

The mass received by the q-th sub flow form flow i is

$$m_q^i = \frac{\rho_q}{\sum_{q'=1}^{n^i} \rho_{q'}} m^i, q = 1, 2, \cdots, n^i \tag{23}$$

After the mass is distributed, the location, objective values, and weighted objective value of flow i are then assigned as parent attributes of the split sub flows; i.e.,

$$\left.\begin{array}{l} \tilde{\mathbf{x}}_q^i \leftarrow \mathbf{x}^i \\ \tilde{\mathbf{f}}_q^i \leftarrow \mathbf{f}^i \\ \tilde{F}_q^i \leftarrow F^i \end{array}\right\} q = 1, 2, \cdots, n^i$$

Notice that if $n^i = 1$, flow i is regarded as being transformed into its single sub flow that is subject to location moving, objective evaluation, and parent change, etc.

Finally, a new set of flows constructed from sub flows is yielded as

$$X \leftarrow \bigcup_{i=1}^{n} \left\{\mathbf{x}_1^i, \mathbf{x}_2^i, \cdots, \mathbf{x}_{n^i}^i\right\}$$

and the new number of flows is

$$n \leftarrow \sum_{i=1}^{n} n^i$$

3.3 Flow Merging

The flow merging operation conducted in the WFA is to eliminate duplicated solution agents to avoid unnecessary solution searches and evaluations. The location-based merging approach is frequently adopted in general optimization problems, either continuous or discrete ones. In addition to the location-based approach, we propose an objective-based merging approach for our multi-objective optimization problems, where a set of non-dominated solutions with objective values dispersedly distributed in the objective space is preferred.

Two flows will be merged in the location-based merging operation when the locations of two flows are close enough. A Location Closeness Factor β_L, $0 < \beta_L < 1$, is defined to facilitate the merging operation. Ifthe normalized component variable differences of two flows are all smaller than the factor, they are eligible for merging. Let LocationMergeable($\mathbf{x}^i$, $\mathbf{x}^{i'}$) be used to identify whether flow i can be merged with i';

$$\text{LocalMergeable}\left(x^i, x^{i'}\right) = \begin{cases} \text{false, if } \exists k \in \{1, 2, \cdots, p\} \ni \frac{\left|x_k^i - x_k^{i'}\right|}{\overline{x}_k - \underline{x}_k} > \beta_L \\ \text{true, otherwise} \end{cases} \tag{24}$$

On the other hand, when objectives of two flows on the objective space are close enough, they can be merged to increase the spread of non-dominated solutions on the objective space. Let β_O, $0 < \beta_O < 1$ be the *Objective Closeness Factor*, which is used for objective-based merging operation. In addition, let ObjectiveMergeable ($\mathbf{x}^i$, $\mathbf{x}^{i'}$) be used to identify whether flow i can be merged with i'; then

$$\text{ObjectiveMergeable}\left(\mathbf{x}^i, \mathbf{x}^{i'}\right) = \begin{cases} \text{false, if } \exists j \in \{1, 2, \cdots, g\} \ni \frac{\left|f_j^i - f_j^{i'}\right|}{\overline{f}_j - \underline{f}_j} > \beta_O \\ \text{true, otherwise} \end{cases} \tag{25}$$

Note that if the number of objective functions is far smaller than the number of variables in a problem, an objective-based merging test is simpler than the location-based test. However, the newly generated sub flows are subject to objective value evaluations prior to the test.

The merging operation sequentially loops through each flow i in the current flow set X to establish a flow sub set A^i that rounds in mergeable flows indexed after i. If flow i', $i < i'$, is rounded in A^i, which means flow i' is mergeable with flow i, $A^{i'}$ is set to null. On the other hand if flow i' is not merged by any preceding flow, $A^{i'}$ contains at least flow i' itself; i.e., $A^{i'} \neq \varnothing$ and $\mathbf{x}^{i'} \in A^{i'}$. To identify whether flow i can merge flow i', operation LocalMergeable(x^i,$x^{i'}$) or ObjectiveMergeable(x^i,$x^{i'}$) is applied depending on which merging mode is chosen. Suppose that flow i is not null and $A^i = \left\{\mathbf{x}^i\right\} \cup \left\{\mathbf{x}'^i_1, \mathbf{x}'^i_2, \cdots, \mathbf{x}'^i_{n'^i}\right\}$; then, there are n'^i flows that are mergeable with flow i. The merging operation simply randomly selects one flow from A^i to represent the merged flow and assigns its mass properties with the aggregated mass of all flows in A^i. Let RandomElement(A) be a random operation that returns one element of the given set A; i.e. RandomElement(A) $\in A$. Then the merged flow of a non-null flow set A^i is $\mathbf{x}'^i =$ RandomElement(A^i). In addition, the mass of the selected flow is replaced with the aggregated mass; i.e., $m'^i = \sum_{\forall i, \mathbf{x}^i \in A^i} m^i$. When all none-null mergeable flow sets are merged into single flows the flow set is further updated as

$$X \leftarrow \left\{\mathbf{x}'^i \,\middle|\, A^i \neq \varnothing \wedge \mathbf{x}'^i = \text{RandomElement}\left(A^i\right)\right\}$$

and the number of flows is as $n \leftarrow Count(A^i \neq \varnothing)$.

3.4 Water Evaporation

Evaporation operation simulates the natural behavior of water evaporating to the atmosphere. The water flow-like algorithm conducts this operation to store a certain amount of water up in the air for chances of water flow regeneration in water precipitation.

An *Evaporation Factor* α is initially set by the user, whose value is restricted by $0 \le \alpha < 0.5$. Let the mass of the cumulated water in the air be v. In the evaporation operation, the evaporated mass of water is added to v by

$$v \leftarrow v + \alpha \sum\nolimits_{i=1}^{n} m^i$$

and the mass of each flow in X after evaporation is updated as

$$m^i \leftarrow (1-\alpha)\, m^i; i = 1, 2, \cdots, n$$

Notice that the mass of a flow indicates the liveliness of the flow while it traverses on the ground. Once it is reduced to a minimal amount, it is regarded as a dried-out flow. The total number of flows is restricted by a user specified limit. Let the maximal number of flows allowed be $\bar{z}$; the minimal amount of mass is then $\overline{m}/\bar{z}$, where $\overline{m}$ is the user specified initial flow mass. A dried-out flow is removed from the flow set and its leftover mass is added to v; i.e.,

$$v \leftarrow v + \sum\nolimits_{\forall i, m^i \le \overline{m}/\bar{z}} m^i$$
$$X \leftarrow X - \left\{ \mathbf{x}^i \middle| m^i \le \overline{m}/\bar{z} \right\}$$

3.5 Precipitation

Precipitation operation simulates the natural rainfall that results in water flows covering a wide range of the territory. The water flow-like algorithm conducts this operation to generate new solution agents for exploring new solution space. When the mass of the evaporated water exceeds a user specified saturation amount the precipitation operation is conducted automatically. The limit is a fraction of the initial mass: $\rho\overline{\mathrm{m}}$; where ρ is the *Precipitation Factor* specified by the user and $0 < \rho \le 0.5$. In other worlds, the precipitation operation is conducted when $v \ge \rho\overline{m}$.

The number of precipitation flows generated is set to the number of ground flows in X; i.e., n. However the total number of flows allowed is restricted by $\bar{z}$. On the other hand, since a minimal mass $\overline{m}/\bar{z}$ is required, at most $\lfloor v\bar{z}/\overline{m} \rfloor$ flows can be generated. Therefore, the number of precipitation flows is

$$n'' = \min\left(n,\ \ \bar{z} - n,\ \ \lfloor v\bar{z}/\overline{m} \rfloor\right) \tag{26}$$

Let $X'' = \left\{ \mathbf{x}''^1, \mathbf{x}''^2, \cdots, \mathbf{x}''^{n''} \right\}$ be the set of generated precipitation flows. Evaporated mass v is then evenly distributed to generated precipitation flows. Let $m\prime\prime^i$ be the mass of precipitation flow $\mathbf{x}\prime\prime^i$, then $m''^i = {}^{v}\!\big/_{n''}$. The location of each precipitation flow is directly duplicated from a randomly selected ground flow and some components are perturbed with random distances. In assigning the location of flow $\mathbf{x}\prime\prime^i$, a ground flow i' in X is randomly selected first. Then a random distance δ_k^i for each component k, restricted by the length between lower and upper bounds, is stochastically calculated. If a forward advancement or backward retraction of distance δ_k^i from $x_k^{i'}$ exceeds the bounds, no perturbation is applied. Otherwise, either forward or backward perturbation is stochastically applied. Components of the location are set by

$$x''^i_k = \left\{ \begin{array}{l} x_k^{i'}, \text{if} \left(x_k^{i'} + \delta_k^i > \overline{x}_k \vee x_k^{i'} - \delta_k^i < \underline{x}_k \right) \\ x_k^{i'} + \delta_k^i, \text{else if} \sim U(0,1) \geq 0.5 \\ x_k^{i'} - \delta_k^i, \text{otherwise} \end{array} ; k = 1, 2, \cdots, p \right\}; \quad (27)$$
$$i = 1, 2, \cdots, n''$$

Where

$$i' = \text{RandomInteger}\,(1, n),$$

and

$$\delta_k^i = \sim U(0,1)\left(\overline{x}_k - \underline{x}_k\right).$$

The parent flow of a precipitation flow is cloned from the selected ground flow as $\tilde{\mathbf{x}}''^i \leftarrow \tilde{\mathbf{x}}^{i'}$. Finally, the flow set X is augmented by precipitation flows as $X \leftarrow X \cup X\prime\prime$ and number of flows is updated as $n \leftarrow n + n\prime\prime$.

3.6 Water Flow Evaluation and Non-dominated Solution Set Update

The final operation performed in an iteration of water flow-like optimization algorithm is to evaluate the objectives of the solutions represented by the flows. Therefore, objective vector for each flow i in X is evaluated as

$$\mathbf{f}^i = \left[f_1\left(\mathbf{x}^i\right)\ f_2\left(\mathbf{x}^i\right) \cdots f_g\left(\mathbf{x}^i\right) \right] \equiv \left[f_j^i \right]_{1 \times g}, i = 1, 2, \cdots, n$$

and the maximum and minimum objectives are updated as

$$\left.\begin{array}{l} \underline{f}_j \leftarrow \min\left(\underline{f}_j, f_j^i\right) \\ \overline{f}_j \leftarrow \max\left(\overline{f}_j, f_j^i\right) \end{array}\right\} j = 1, 2, \cdots, g$$

Moreover, the cumulated objective differences ΔF and the cumulated number of flows generated N are both updated as

$$\Delta F \leftarrow \Delta F + \sum\nolimits_{i=1}^{n} \left|F^i - \tilde{F}^i\right| \text{ and } N \leftarrow N + n$$

Notice that we have maintained the non-dominated solution set $\widehat{X}$ that will be presented to the user as solutions to the problem. Once the objectives of all flows are calculated, each flow is subject to a serial of domination tests for being added to $\widehat{X}$. To be added to $\widehat{X}$ the solution of a flow must not be dominated by any solution in both X and $\widehat{X}$. Therefore $\widehat{X}$ is augmented by new non-dominated solutions as

$$\widehat{X} \leftarrow \widehat{X} \cup \left\{\mathbf{x}^i\right\}, \text{if} \left(\underset{\forall \mathbf{x}^{i'} \in X, i \neq i'}{\wedge} \neg\text{Dominated}\left(\mathbf{x}^i, \mathbf{x}^{i'}\right)\right)$$
$$\wedge \left(\underset{\forall \mathbf{x}^{i'} \in \widehat{X}}{\wedge} \neg\text{Dominated}\left(\mathbf{x}^i, \widehat{\mathbf{x}}^{i'}\right)\right); \quad i = 1, 2, \cdots n.$$

3.7 Non-dominated Solution Set Pruning

Before presenting the final non-dominated solution set $\widehat{X}$ to the user, an extra pruning operation is required if the number of solutions s $\widehat{X}$ in exceeds the user specified maximum number $\overline{s}$. In the operation, the same procedure of removing the worst solution is recursively executed $s - \overline{s}$ times to prune $\widehat{X}$ to have exactly $\overline{s}$ non-dominated solutions left. The worst solution is the one that has the smallest *jamming distance* among the solutions. The jamming distance τ^i of a solution $\widehat{\mathbf{x}}^i$ in $\widehat{X}$ is the smallest aggregated normalized distance to other solutions on the objective space. A normalized objective distance is the objective value difference between two solutions divided by the bounding length of the objective value. Let r_j be the bounding length of objective j, then

$$r_j = \max_{i'=1,\cdots,s}\left\{\widehat{f}_j^{i'}\right\} - \min_{i'=1,\cdots,s}\left\{\widehat{f}_j^{i'}\right\} \tag{28}$$

The jamming distance of each non-dominated solution i in $\widehat{X}$ is

$$\tau^i = \min_{i'=1,\cdots,s;i\neq i'} \left\{ \sum\nolimits_{j=1}^{g} \left(\frac{1}{r_j} \left| \widehat{f}_j^i - \widehat{f}_j^{i'} \right| \right) \right\} \tag{29}$$

In the repeated procedure, all r_j are updated by the objective values of all of the current solutions in $\widehat{X}$ first. Then the jamming distances of all solutions are recalculated. The solution $\widehat{\mathbf{x}}^{i^*}$ with the smallest jamming distance is then removed form $\widehat{X}$; i.e., $\widehat{X} \leftarrow \widehat{X} - \left\{ \widehat{\mathbf{x}}^{i^*} \right\}$, where $i^* = \arg \min_{i=1,\cdots,s} \tau^i$. Note that instead of executing the procedure only one time to removing $s - \overline{s}$ inferior solutions directly, $s - \overline{s}$ recursive recalculations of all r_j and τ^i are conducted to remove totally $s - \overline{s}$ solutions from the objective space. The recursive removal of the worst solution has a better ability to yield an evenly dispersed solution set.

4 Performance Index

Generational distance is a common way to evaluate the performance of non-dominant solution. In addition, this paper proposed two more performance indices are proposed to evaluate the non-dominant solution.

4.1 Generational Distance

Generational Distance is proposed by Veldhuize [5]. It is defined as the Euclidean distance between current computed non-dominant solution and the known global optimal solution. Smaller Generational Distance means the current obtained non-dominant solution is closer to the optimal Pareto-front solution.

4.2 Correctness

Correctness index is to measure that how many percentages of current obtained non-dominant solutions belong to the true optimal Pareto front. An optimizer which yields the solution with high Correctness indicates that such optimizer is credible. Correctness index is calculated as follows

$$C = \frac{\left|\widehat{X}'\right|}{\left|\widehat{X}\right|} \times 100\% = \frac{1}{\overline{s}} \cdot \left|\widehat{X}'\right| \times 100\% \tag{30}$$

Where

$$\widehat{X}\text{'}\left\{\widehat{X}^{i}\,\middle|\,\widehat{X}^{i}\in\widehat{X}\text{ and}\forall\bigwedge_{\forall\widehat{X}^{i}\in\widehat{X}}\neg\text{Dominated}\left(\widehat{X}^{i},\widehat{X}^{i'}\right)\right\}\tag{31}$$

4.3 Coverness

Coverness is devised to quantify how well the non-dominant solutions spread. Define the Coverness

$$\ell=\frac{u}{\left|\overset{\frown}{X}\right|}\tag{32}$$

Where u represents number of Pareto front solution covered by non-dominant solution. Define ε as user specified Radius for Pareto Neighborhood. Here Radius for Pareto Neighborhood is suggested to set as

$$\varepsilon=\frac{1}{\left|\overset{\frown}{X}\right|}\tag{33}$$

Define $\mathbb{Q}=\left\{Q^{1},Q^{2},\dots Q^{\overset{\frown}{n}}\right\}$ as the sets for identifying if Pareto front solution covered by non-dominant solution. Where $\overset{\frown}{n}$ is the number of Pareto front solution. Initially $Q^{i'}=0,i'=1,2,\dots,\overset{\frown}{n}$. Each element of the set is computed as

$$Q^{i*}=\begin{cases}1, & \text{if}\ \ i^{*}=\arg\ \ \min\left\{K_{j}^{i'}\,\middle|\,K_{j}^{i'}<\varepsilon\right\},\forall Q_{i'}=0\\0, & \text{otherwise}\end{cases}\tag{34}$$

where

$$K_{j}^{i'}=\sum_{j=1}^{g}\left(\frac{\left|\overset{\frown}{f}{}_{j}^{i'}-\widehat{f}_{j}^{i}\right|}{f_{j}{}^{(u)}-f_{j}{}^{(l)}}\right)^{2}\tag{35}$$

and

$$u=\sum_{i'=1}^{\widehat{n}}Q^{i'}\tag{36}$$

Coverness ℓ can be obtained then. Large ℓ value means non-dominant solutions have wide spreading and able to cover most of Pareto-front solution. Notice that while using Coverness, $\overset{\frown}{n} = s$ is necessary condition.

5 Simulation Study

Simulation study in this paper is conducted in twofold. Firstly, find out the best of flow merging method in WFA. Secondly, compare non-dominant solution of WFA among other proposed methods. Recommended Parameter settings for WFA are shown as Table 1. High saturation rate setting is suggested to avoid redundant search caused by precipitation overly.

Table 2 presents the numerical result for simulation study2. Benchmark problem proposed by Zizler et al. [6] is in use for this experiment. We can obtain that OM method has better performance based on Coverness index. It indicates that OM method generates the wider range of non-dominant solution. However, two of other indices show that LM method has better overall performance.

In Table 3, generation distance shows that WFA4MC outperforms other methods in the benchmark ZDT3, ZDT4 and ZDT6. Though WFA4MC is not an overwhelming MOO method, it is still an option for solving MOO problem.

Table 1 WFA4MC parameter settings for different benchmark

	Benchmark				
Parameter	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
$\overline{m}$	50	50	50	50	50
$\overline{n}$	5	5	5	5	5
$\overline{z}$	15	15	15	15	15
ρ	0.9	0.9	0.9	0.9	0.9
α	0.1	0.1	0.1	0.1	0.1

Table 2 Performance LM and OM methods

Average	Generational distance		Coverness		Correctness		Obj. recalling number
Benchmark	LM	OM	LM	OM	LM	OM	
ZDT1	**0.000508**	0.00072	**65.5**	64.7	**0.973**	0.971333	30, 000
ZDT2	**0.000427**	0.00049	**98.3**	96.2	0.9545	**0.965333**	30, 000
ZDT3	**0.000501**	0.0005	40.4	**44.5**	0.8531	**0.861724**	30, 000
ZDT4	**0.000438**	0.00048	**98.4**	97	**0.979**	0.965423	250, 000
ZDT6	**0.000991**	0.00107	**0.45**	0.37	0.9036	**0.918946**	30, 000

Table 3 Benchmark problems on different methods

	WFA4MC	WFA4MC			
Benchmark	(LM)	(OM)	AMOPSO	NSGAII	MOEPSO
ZDT1	0.0004	0.0004	0.0215	0.0243	**0.0002**
ZDT2	9.71E-05	9.80E-05	0.0083	0.0214	**9.43E-05**
ZDT3	**0.0002**	**0.0002**	0.0058	0.0126	0.0005
ZDT4	**0.0002**	**0.0002**	0.0224	0.0217	0.0185
ZDT6	**0.0009**	0.001	0.0167	0.0195	0.0177

6 Conclusion

In this paper, a multi-objective continuous problem solver WFA4MC based on WFA is proposed. WFA mimics the phenomenon of water flow in the nature, consisting of flow splitting, flow merging and precipitation.

In addition, other than generational distance, this paper proposed two novice criteria for non-dominant solution evaluation such as correctness and Coverness index. Firstly, correctness calculates the percentage that current non-dominant solutions are not dominated from the true non-dominant solutions. Secondly, Coverness quantifies how well the current non-dominant covers the true Pareto front. Three total indices are applied to evaluate two of the flow merging mechanism, OM and LM. Simulation result shows that though LM can gain smaller generational distance, OM has higher correctness.

Comparing with AMOPSO, NSGAII and MOEPSO, Simulation study based on generational distance index shows that our proposed method outperforms other methods in several benchmarks.

References

1. Chen D-b, Zou F, Wang J-t (2011) A multi-objective endocrine PSO algorithm and application. Appl Soft Comput 11:4508–4520
2. Deb K, Pratap A, Agarwal S, MeyM T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:182–197
3. Tripathi PK, Bandyopadhyay S, Pal SK (2007) Adaptive multi-objective particle swarm optimization algorithm. IEEE Congr Evol Comput 8:2281–2288
4. Yang FC, Wang YP (2007) Water flow-like algorithm for object grouping problems. J Chin Inst Ind Eng 24(6):475–488
5. Veldhuizen DAV (1999) Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Air Force Institute of Technology, Wright Patterson AFB, USA
6. Zitzler E, Deb K, Thiele L (2002) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8:173–195