A Population-Based Simulated Water-stream Algorithm for Multi-modal Optimization Problem

Abstract

It is found that the water streams with a hybrid process of downstream and penetration towards the basin are analogous to the process of finding the minimum solution in an optimization problem. Inspired by this natural phenomenon, we will propose a novel population-based simulated water-stream algorithm (SWA) accordingly for multi-modal optimization problems which are quite common in a variety of application areas. It is shown that the SWA can be converged to a global optimum solution with probability one. The SWA featuring a combination of deterministic search and heuristic search generally converges much faster than the existing counterparts with a considerable accuracy enhancement. Experimental results show the efficacy of the proposed algorithm.

1 Introduction

Optimization problems are quite common in a variety of scientific areas. In the last decades, a number of optimization algorithms based on different theories and methodologies have been presented, which can be roughly summarized into two categories. In the first category, supposing the objective function in an optimization problem is differentiable, the classical optimization algorithms include Steepest Decent Method (SDM), Conjugate Gradient Method [Hestenes and Stiefel, 1952], and Quasi-Newton Method [Davidon, 1991]. In general, they are able to converge to a global optimal solution quickly for a convex optimization problem [Boyd and Vandenberghe, 2004], but may almost always lead to a local optimal solution if the objective function is multi-modal. Recently, some optimization algorithms, e.g. tunneling algorithm [Oblow, 2001] and filled function method [Lucidi and Piccialli, 2002], have been developed for single objective optimization problems (SOP) with multi modals. They have been shown to be effective in their application domain, but their computation is generally laborious without guaranteeing to converge to global optimum solution in the multi-dimensional optimization problems. Further, as well as the previously stated algorithms in this category, they can find one solution only in a single run, which is essentially inappropriate for multi-objective optimization problems (MOP), in which a set of Pareto optimal solutions, rather than a single one, is desired.

The other category is population-based heuristic algorithms, e.g. Evolutionary Algorithm [Deb *et al.*, 2002; Zhang and Li, 2007; Wang and Dang, 2007], Particle Swarm Optimization [Van den Bergh and Engelbrecht, 2004], and Ant Colony Algorithm [Dorigo *et al.*, 1996], all of which have achieved promising results in dealing with complex problems. Compared to the methods in the first category, the population-based ones need not hypothesize the objective functions in terms of modality and differentiability, and can find a set of optimal solutions in a single run. Hence, they are much appropriate for non-differentiable multi-modal optimization problems and MOP. Nevertheless, such an algorithm suffers from the curse-of-dimensionality problem as described in [Van den Bergh and Engelbrecht, 2004]. That is, their performance will deteriorate as the dimensionality of the search space increases. Further, the convergence speed of these population-based algorithms is generally much slower in comparison with the methods in the first category.

It is found that the water streams with a hybrid process of downstream and penetration towards the basin are analogous to the process of finding the minimum solution in an optimization problem. Inspired by this natural phenomenon, this paper will propose a novel population-based simulated water-stream algorithm (SWA) accordingly for differentiable multi-modal optimization problems. Specifically, a deterministic search is utilized to simulate the process of downstream, while two heuristic searches are introduced to simulate the necessity and contingency of the water penetration. The necessity means most of the water will penetrate downward. It is simulated by a heuristic search based on adaptive cooperative learning and simplex method, in which the leaning rate is adapted to the norm of the direction of deterministic search. The contingency implies that a small portion of water may penetrate laterally, even upward. It is represented by a heuristic search based on random perturbation. As a result, the SWA featuring a combination of the deterministic search and heuristic search inherits the merits of those methods in the above-stated two categories. That is, the characteristics of the SWA are three-folded at least: (1) It is shown that the SWA can be converged to a global optimum solution with the probability 1; (2) The SWA is suitable for MOP, as well as SOP; (3) The convergence speed of SWA is quite fast. Numerical studies have shown the promising results of SWA in comparison with (1) SDM and EA in four single objective test instances, and (2) MOEA/D, NSGA-II and SDM with different weight vectors in five multi-objective test instances, respectively.

The remainder of this paper is organized as follows: Section 2 gives a detail description of the SWA. Section 3 analyzes the global convergence of SWA. In Section 4, we show the experimental results to compare the SWA with the existing counterparts. Finally, we draw a conclusion in Section 5.

2 Description of SWA

Without loss of generality, an optimization problem can be formulated as the following objective function:

$$\min_{x \in D} f(x) = (f_1(x), f_2(x), \cdots, f_m(x)) \tag{1}$$

where $D \subset R^n$ is the decision space, the vector $x \in D$ is the decision variable and $f_i(x)$ is the ith objective function, $i = 1, \cdots, m$. As $m > 1$, we can transform the MOP into a number of SOPs with the different uniform weight vectors by using Tchebycheff strategy [Deb, 2001].

The SWA is an adaptive learning algorithm, which mainly consists of two operators: downstream and penetration. That is, finding of a minimum solution can be regarded as a hybrid search process of downstream and penetration as described in Figure 1. In the following sub-sections, we will give the detailed description of SWA.

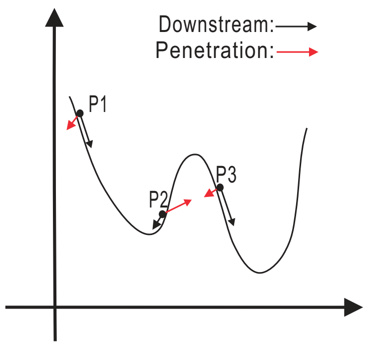

Figure 1: The search procedure of SWA for a solution

2.1 Downstream Operator

The search direction of downstream operator is the descent direction of the objective function. Thus, it can be simulated by a deterministic search approach. In this paper, the SDM algorithm is utilized for this purpose. In the tth iteration of the algorithm, the current solutions are $X^t = \{x_1^t, \cdots, x_N^t\}$, where N is the number of the solutions. For each solution x_i^t, the search direction of downstream operator can be formulated as:

$$p_i^t = \alpha_i^t f'(x_i^t) \tag{2}$$

where the positive scalar α_i^t is called the step length. It is computed by using the line search methods. $f'(x_i^t)$ is the negative gradient of $f(x)$ at point x_i^t. Since the transformed objective function is a non-smooth function for MOP, we need to replace the gradient with the subgradient. The search direction of downstream operator is represented by the black arrow in Figure 1. The enhanced solution is generated as follows:

$$x_i^D = x_i^t + p_i^t, \;\; i = 1, \cdots, N. \tag{3}$$

2.2 Penetration Operator

The water penetration is a unity of necessity and contingency. Necessity means most of water oozes downward. It is similar to the search direction of simplex method of mathematic programming. It is expected that the new solution will be close to the better neighbors but depart from the worse ones. Additionally, it is found that the higher the velocity of the water stream is, the lower its permeability is. Accordingly, a heuristic search based on simplex method and adaptive cooperative learning among the neighbor solutions is utilized to simulate the necessity of water penetration. Contingency refers to a little of water runs around all over the place. Thus, a small portion of solutions are operated by a random perturbation to simulate the contingency of water penetration. The heuristic search direction simulating the water penetration is marked as the red arrow in Figure 1.

Adaptive Cooperative Learning Heuristic Search

Analogous to the phenomenon that the scale of water penetration is within a certain region, a solution can exchanges information with the neighbor ones only. Hence, we define the neighbor structure matrix as $B_{N\times K}$, where K is the number of the neighbors of each solution. The ith column of B, denoted as $B(i)$, contains the indexes of the K neighbors of the ith solution x_i. If $j \in B(i)$, x_j will be a neighbor of x_i.

Hereinafter, we introduce the neighbor structure B for SOP. We first introduce a shift mapping S: $S(1, 2, \cdots, N) = (2, 3, \cdots, N, 1)$ and a permutation $V_0 = (1, 2, \cdots, N)$. Then, the jth row of the neighbor structure matrix B, denoted as V_j, is given by $V_j = S(V_{j-1})$, $j = 1, 2, \cdots, K$. Clearly, the neighbors of each solution just depends upon the index of the solutions in SOP. The neighbor structure of MOP is defined as follows: Let $(\omega^1, \omega^2, \cdots, \omega^N)$ be a set of even spread weight vectors, set $B(i)$ are the index of the K closest weight vectors to ω^i.

Using the idea of simplex method of mathematic programming, the new solution will be close to the better neighbors but depart from the worse ones. Let B_{bi}^t be the indexes of the neighbors which are better than x_i^D and B_{wi}^t be the indexes of the worse ones. Then, a new solution can be generated as:

$$x_i^P = x_i^D + \lambda_i^t (C_1^i - x_i^D) - \lambda_i^t (C_2^i - x_i^D) \tag{4}$$

where C_1^i and C_2^i are the center of the better neighbors and the worse neighbors, respectively, and λ_i^t is the learning rate. Subsequently, Eq.(4) can be modified as:

$$x_i^P = \begin{cases} x_i^D - \lambda_i^t (C_2^i - x_i^D) & \text{if } |B_{bi}^t| = 0 \\ x_i^D + \lambda_i^t (C_1^i - x_i^D) & \text{if } |B_{wi}^t| = 0 \\ x_i^D + \lambda_i^t (C_1^i - C_2^i) & \text{otherwise,} \end{cases} \tag{5}$$

where $|B_{bi}^t|$, $|B_{wi}^t|$ are the size of set B_{bi}^t and B_{wi}^t, respectively.

An adaptive learning rate control strategy is introduced to adjust the learning rate λ_i^t in accordance with the norm of the deterministic search direction p_i^t in Eq.(2). That is,

$$\lambda_i^t = 0.1 r_1 \cdot e^{-\|p_i^t\|} \quad (6)$$

where $\|p_i^t\|$ is the norm of p_i^t, r_1 is a random number generated by a uniform random number generator in $[0, 1]$, denoted as $rand$. In general, when x_i^t is close to the extreme points, $\|p_i^t\|$ is relatively small. Equation (6) implies that the smaller of $\|p_i^t\|$ is, the bigger of the learning rate is. As shown in Figure 1, the position P2 is close to the local minimum. The cooperative learning search direction dominates the search procedure. It facilitates the solution escapes from the local minimum and avoids the premature convergence in the algorithm. However, when x_i^t is not close to the extreme points, Equation (6) leads to the conclusion that the cooperative learning rate is relatively small. Thus, the deterministic search direction dominates the search procedure. The positions P1 and P3 tally with the above situation in Figure 1. Obviously, it is better for searching minimum optimum solution.

Random Perturbation Heuristic Search

A random heuristic search is performed on a small portion of solutions to simulate the contingency of water penetration. Suppose a random search is performed on x_i^D. Every component of x_i^D is perturbed with perturbation probability p. If $x_{i,l}^D$, the lth component of x_i^D, is selected to perturb, it is reassigned a random number in $[lb(l), ub(l)]$. where lb, ub are the lower bound and upper boundary, respectively. Hence, a new solution is generated as:

$$x_{i,l}^P = \begin{cases} lb(l) + r_2(ub(l) - lb(l)) \text{ if } rand < p \\ x_{i,l}^D \quad \text{otherwise} \end{cases} \quad (7)$$

with $l = 1, \cdots, n$. where r_2 is a random number generated from $rand$. Obviously, the random perturbation facilitates maintaining the diversity of solutions and overcoming premature.

2.3 The Summarized Procedure of SWA

The implementation procedure of SWA can be summarized as follows:

The SWA Algorithm

Step 1. **Initialization:** Set the size N of the solutions, the size of the neighbor K, identify the neighbor structure $B(i)$ for $i = 1, \cdots, N$, give the perturbation probability p, the current iteration $t = 0$. Generate N initial solutions $X^0 = \{x_1^0, x_2^0, \cdots, x_N^0\}$.

Step 2. **Update:**

For each x_i^t, $i = 1, \cdots, N$.

Step 2.1. **Downstream Operator:** Apply a specific deterministic optimization approach on x_i^t to produce a solution x_i^D by Eq.(3).

Step 2.2. **Penetration Operator:**

Let r_3 be a random number from $rand$.

If $r_3 > 0.1$, the new solution x_i^P is generated by Eq.(5).

Else

The new solution x_i^P is generated by Equation (7)

Step 2.3. **Update of Solution:**

If x_i^P is better than x_i^t, then $x_i^{t+1} = x_i^P$.

Step 3. **Stopping Criteria:** Let $t = t + 1$. If stopping criteria is satisfied, the algorithm stops and outputs X^t. Otherwise, go to Step 2.

3 Analysis of Global Convergence of SWA

3.1 Overview of Theoretical Results in [Solis and Wets, 1981]

Suppose the global minimizer of the problem in Eq.(1) is not isolated. That is, its sublevel sets

$$D_\beta = \{x \in D \mid f(x) \leq \beta\}$$

for $\beta > \min_{x \in D} f(x)$ is a nonempty and compact set. In the literature, [Solis and Wets, 1981] has provided a criteria to show whether an algorithm is the global search one or not with the two assumptions and one theorem as follows:

A1.$f(H(z,\xi)) \leq f(z)$ and if $\xi \in D$, then $f(H(z,\xi)) \leq f(\xi)$, where H is a function that constructs a solution to the problem. This assumption guarantees that the newly constructed solution will be no worse than the current one.

A2. For any subset A of D with $v(A) > 0$, we have that

$$\prod_{k=0}^{\infty}(1 - \mu_k(A)) = 0,$$

where $v(A)$ is the n-dimensional volume of the set A, $\mu_k(A)$ is the probability of A being generated by μ_k, and μ_k is a probability measure. This assumption implies that any subset A of D with positive measure v, the probability of repeatedly missing the set A must be zero.

Theorem 1: Suppose that f is a measurable function and D is a measurable subset of R^n. If an algorithm satisfies A1 and A2, it converges to a global optimum solution with probability one.

3.2 SWA: A Global Search Algorithm

In this sub-section, we will utilize the result of the paper [Solis and Wets, 1981] to study the convergence characteristics of the SWA.

The SWA simulates the downstream and penetration of the water-stream. More precisely, it can be accomplished by the following processes:

$$X^t \xrightarrow{\text{Down}} X^D \xrightarrow{\text{Penetrate}} X^P \xrightarrow{\text{Update}} X^{t+1}$$

Lemma 1: The SWA satisfies A1.

proof: From the solution update in SWA, we know that function H (as introduced in assumption A1) is defined as:

$$H(g(x_i^t), x_i^t) = \begin{cases} x_i^t & \text{if } f(g(x_i^t)) \geq f(x_i^t) \\ g(x_i^t) & \text{otherwise} \end{cases} \quad (9)$$

where g denotes the function of the operators performed in SWA as defined in Eq.(3), Eq.(5), and Eq.(7). The definition of H in the above clearly complies with A1.

Lemma 2: The SWA satisfies A2.

proof: The penetration process of SWA is heuristic search process. To satisfy A2, the union of the sample space of the solutions must cover D so that

$$D \subseteq \bigcup_{i=1}^{N} M_i^t \quad (10),$$

where M_i^t is the support of the sample space of x_i^D. There are two different definitions for M_i^t.

1. For the solutions updated by Eq.(5), whose index set is denoted as I_1, the shape of M_i^t is defined as follows:

$$M_i^t = x_i^D + \lambda_i^t \cdot \Omega_i^t$$

where

$$\Omega_i^t = \begin{cases} x_i^D - C_2^i & \text{if } |B_{bi}^t| = 0 \\ C_1^i - x_i^D & \text{if } |B_{wi}^t| = 0 \\ C_1^i - C_2^i & \text{otherwise.} \end{cases}$$

M_i^t is a hyper-rectangle parameterized by λ_i^t, with one corner specified by $\lambda_i^t = 0$ and the other by $\lambda_i^t = 0.1e^{-\|p_i^t\|}$. As $0.1e^{-\|p_i^t\|} \cdot \Omega_i^t < 0.5diam(D)$, it is clear that $v(M_i^t \bigcap D) < v(D)$, where $diam(D)$ denotes the length of D. Based on Lemma 1, we know that the SWA algorithm is convergent. Subsequently, the length of M_i^t will tend to 0 as t tends to infinity. Thus, $v(\bigcup_{i \in I_1} M_i^t \bigcap D) < v(D)$, i.e. M_i^t cannot cover D for $i \in I_1$.

2. For the solutions updated by Eq.(7), whose index is denoted as I_2, it is clear that $M_i^t = D$ for $i \in I_2$.

In summary, we have $D \subseteq \bigcup_{i \in I_2} M_i^t \subseteq \bigcup_{i=1}^{N} M_i^t$, which implies that SWA satisfies A2.

*Theorem 2:*The SWA algorithm converges to a global optimal solution with probability one.

proof: The SWA satisfies A1 and A2 by Lemma 1 and Lemma 2. According to Theorem 1, the SWA converges to a global optimum solution with probability one.

4 Simulation Results

We conducted two experiments to evaluate the performance of the SWA in comparison with the existing counterparts. In Experiment 1, we compared the SWA with the SDM and EA [Wang and Dang, 2007] on SOP with the population size $N = 50$. To make a fair comparison, N initial solutions were generated at the beginning of implementing SDM, and then optimized individually. Also, the setting of the control parameters in EA was the same as that in [Wang and Dang, 2007]. In Experiment 2, we compared the SWA with NSGA-II [Deb *et al.*, 2002], MOEA/D [Zhang and Li, 2007], and SDM on multi-modal multiobjective optimization problems. Since the SDM is usually used to solve SOP only, N evenly distributed weight vectors were then generated as described in [Zhang and Li, 2007] so that N solutions can be finally obtained in a single run. Furthermore, we decomposed the MOP into N subproblems by using Tchebycheff strategy, and then optimized them individually by the SDM using the subgradient. Additionally, in NSGA-II and MOEA/D, the crossover and mutation operators with the same control parameters as the ones in [Zhang and Li, 2007] were used to generate new solutions. The other control parameters in MOEA/D, i.e. $T = 20$, $\delta = 0.9$, were the same as the ones in [Zhang and Li, 2007]. In both of the two experiments, the size of the neighbors in SWA was set at 5, while the perturbation probability was set at 0.1.

4.1 Experiment 1

To make the test instances differentiable, this experiment utilized the following four test instances that are the modified ones of those instances in [Wang and Dang, 2007]:

SF1:(Rastrigin's function)

$$f(x) = \sum_{i=1}^{n} \left[x_i^2 - 3\cos(2\pi x_i) + 3\right]$$

where $-5.12 \leq x_i \leq 5.12$. There are about 10^n local optimum in the decision space. Figure 2 depicts the 3D-plot of **SF1**.

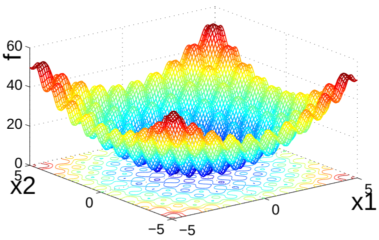

Figure 2: 3D-plot of SF1

SF2:(Ackley's function)

$$f(x) = -20\exp\left(-\sqrt{1 + \frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20e^{-1} + e$$

where $-32 \leq x_i \leq 32$. There are about 64^n local optimal solutions in the decision space as depicted in Figure 3.

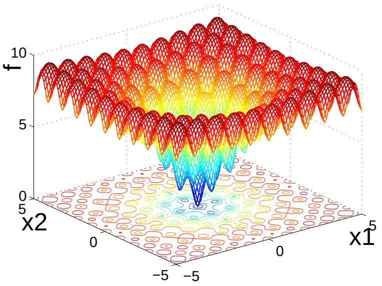

Figure 3: 3D-plot of SF2

In **SF1** and **SF2**, the decision variables are separable and can be solved by using n searches. Hence, we have two rotated multi-modal problems. An orthogonal matrix M should be generated, a new rotated variable is then obtained by $y = M * x$. That is, if $M = \begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ m_{n1} & m_{2n} & \cdots & m_{nn} \end{pmatrix}$, then we have

$$y_i = \sum_{j=1}^{n} m_{ij} x_j, \ i = 1, \cdots, n.$$

When one dimension in x is changed, all dimensions in y will be affected. Hence, the rotated function cannot be solved by just n searches. The orthogonal rotation matrix does not affect the optimal solution of the functions. Subsequently, we have the other two corresponding test instances:

SF3: $f(x) = \sum_{i=1}^{n} \left[y_i^2 - 3\cos(2\pi y_i) + 3 \right]$.

SF4: $f(x) = -20 \exp\left(-\sqrt{1 + \frac{1}{n} \sum_{i=1}^{n} y_i^2} \right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi y_i) \right) + 20e^{-1} + e.$

All the algorithms independently run for 20 times for each test instance. Table 1 presents the results obtained by the algorithms. It can be seen that SWA can successfully find the optimal solutions in all test instances we have tried so far. Furthermore, the EA could find close-to-optimal solutions only with more generations. By contrast, the solutions found by SWA are better than the ones by EA with a much smaller number of function evaluations as shown in Figure 4 and 5. Also, the SWA outperformed the SDM in those multi-modal test instances with a considerable accuracy enhancement.

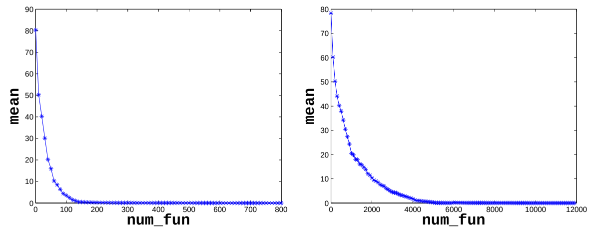

Figure 4: The mean value versus num_fun obtained by SWA and EA, respectively, for **SF1** with $n = 10$

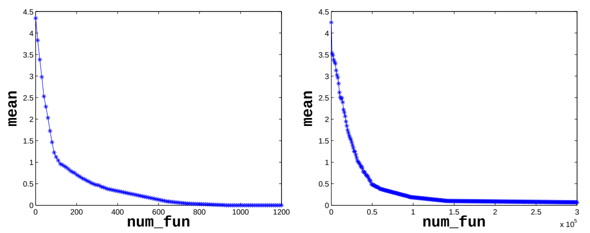

Figure 5: The mean value versus num_fun obtained by SWA and EA, respectively, for **SF2** with $n = 100$

4.2 Experiment 2

In this experiment, five differentiable multiobjective test instances, denoted as MF1-MF5, were constructed based on DTLZ [Deb *et al.*, 2002]:

MF1: $\begin{cases} f_1(x) = g(x) + 1 - x_1 \\ f_2(x) = g(x) + x_1 \end{cases}$

where $g(x) = \sum_{i=2}^{n} \left(x_i^2 - 3\cos(10\pi x_i) + 3 \right)$, $x \in [0,1] \times [-1,1]^{n-1}$.

MF2: $\begin{cases} f_1(x) = g(x) + \cos(0.5\pi x_1) \\ f_2(x) = g(x) + \sin(0.5\pi x_1) \end{cases}$

where $g(x)$ and the decision space are the same as MF1.

MF3: $\begin{cases} f_1(x) = g(x) + 1 - \cos x_1 \\ f_2(x) = g(x) + 1 - \sin x_1 \end{cases}$

where $g(x) = -20 \exp\left(-\sqrt{1 + \frac{1}{n} \sum_{i=2}^{n} 10x_i^2} \right) - \exp\left(\frac{1}{n} \sum_{i=2}^{n} \cos(20\pi x_i) \right) + 20e^{-1} + e$, $x \in [0,1] \times [-1,1]^{n-1}$.

MF4: $\begin{cases} f_1(x) = g(x) + \cos(0.5\pi x_1)\cos(0.5\pi x_2) \\ f_2(x) = g(x) + \cos(0.5\pi x_1)\sin(0.5\pi x_2) \\ f_3(x) = g(x) + \sin(0.5\pi x_1) \end{cases}$

where $g(x) = \sum_{i=3}^{n} \left(x_i^2 - 3\cos(10\pi x_i) + 3 \right)$, $x \in [0,1]^2 \times [-1,1]^{n-2}$.

MF5: $\begin{cases} f_1(x) = g(x) + (1 - \cos(0.5\pi x_1))(1 - \cos(0.5\pi x_2)) \\ f_2(x) = g(x) + (1 - \cos(0.5\pi x_1))(1 - \sin(0.5\pi x_2)) \\ f_3(x) = g(x) + (1 - \sin(0.5\pi x_1)) \end{cases}$

where $g(x) = -20 \exp\left(-\sqrt{1 + \frac{1}{n} \sum_{i=3}^{n} 10x_i^2} \right) - \exp\left(\frac{1}{n} \sum_{i=3}^{n} \cos(20\pi x_i) \right) + 20e^{-1} + e$ $x \in [0,1]^2 \times [-1,1]^{n-2}$. In MF1-MF3, the population size N was set at 100, while $N = 300$ in MF4-MF5.

To evaluate the algorithms' performance, we suppose a set of points Q^* uniformly distribute along the Pareto Front (PF) in object space and Q is an approximate to the PF. Hence, the distance between Q^* and Q can be defined as:

$$IGD(Q^*, Q) = \frac{\sum_{v \in Q^*} d(v, Q)}{|Q^*|},$$

where $d(v, Q)$ is the minimum Euclidean distance from the point v to Q. Obviously, the smaller value of IGD-value, the better performance of algorithm. Accordingly, we utilized the IGD-value to evaluate the performance of all algorithms. We uniformly selected 500 points for 2-objective test instances and $1,000$ points for 3-objective test instances to construct the set Q^* along the PF.

All the algorithms were executed 20 independent runs for each test instance. Table 2 presents the best and mean of IGD-value of the final solutions obtained by each algorithm for each test instance. It can be seen that the SWA outperforms the other algorithms in all test instances we have tried so far.

5 Conclusion

Inspired by the water streams with a hybrid process of downstream and penetration, this paper has proposed a population-based SWA for multi-modal optimization problem. The SWA mainly consists of two operators: downstream and penetration. In the former, the deterministic search is utilized to simulate the process of downstream, while the latter utilizes a heuristic search based on adaptive cooperative learning and simplex method to simulate the necessity of

Instances	n	SWA			EA			SDM		
		num_fun	best	mean	num_fun	best	mean	num_fun	best	mean
SF1	10	800	0	0	12,000	3.481E-12	8.921E-11	800	0.829	3.345
	50	1000	0	0	100,000	8.320E-10	9.221E-7	1000	4.235	10.23
	100	1200	0	0	300,000	4.478E-4	1.235	1200	11.24	34.23
SF2	10	800	0	0	12,000	3.429E-10	4.527E-9	800	0.112	0.357
	50	1000	0	0	100,000	2.453E-5	8.378E-4	1000	0.623	1.534
	100	1200	0	0	300,000	4.213E-3	6.362E-2	1200	1.023	1.589
SF3	10	1200	0	0	12,000	5.563E-8	5.216E-6	1200	1.234	2.247
	50	1500	0	0	100,000	2.367E-5	4.763E-3	1500	3.417	7.372
	100	2000	0	0	300,000	0.357	2.317	2000	5.245	8.251
SF4	10	1200	0	0	12,000	6.472E-10	7.582E-8	1200	0.241	0.379
	50	1500	0	0	100,000	7.271E-7	6.317E-5	1500	0.427	1.003
	100	2000	0	0	300,000	2.468E-3	0.368	2000	0.713	1.421

Table 1: Comparative results obtained by the SWA, EA, and SDM, respectively, where n is the dimension of the search space, num_fun denotes the maximal number of function evaluations in the 20 runs, $best$ refers to the best function values in the 20 runs, and $mean$ is the average value of the best function values in the 20 runs.

Instances	n	SWA			NSGA-II			MOEA/D			SDM		
		num_fun	best	mean	num_fun	best	mean	num_fun	best	mean	num_fun	best	mean
MF1	10	2000	0.0025	0.0025	50,000	0.0045	0.0068	50,000	0.0030	0.0035	2000	1.0245	1.2802
	50	3000	0.0031	0.0033	300,000	0.0052	0.0073	300,000	0.0038	0.0053	3000	2.0219	2.9028
MF2	10	2000	0.0019	0.0020	50,000	0.0036	0.0047	50,000	0.0020	0.0021	2000	0.8039	1.3729
	50	3000	0.0020	0.0020	300,000	0.0038	0.0050	300,000	0.0020	0.0023	3000	0.5390	1.8037
MF3	10	2000	0.0028	0.0035	50,000	0.0030	0.0039	50,000	0.0031	0.0035	2000	0.4290	0.9048
	50	3000	0.0030	0.0038	300,000	0.0041	0.0058	300,000	0.0034	0.0068	3000	0.8053	1.0948
MF4	10	6000	0.0235	0.0257	50,000	0.0325	0.0429	50,000	0.0298	0.0328	6000	0.9048	1.0649
	50	9000	0.0248	0.0263	300,000	0.0498	0.0501	300,000	0.0279	0.0319	9000	0.3079	2.8303
MF5	10	6000	0.0423	0.0487	50,000	0.0523	0.0581	50,000	0.0429	0.0507	6000	0.7838	1.9048
	50	9000	0.0512	0.0584	300,000	0.0529	0.0698	300,000	0.0460	0.0529	9000	0.9684	1.5394

Table 2: The IGD-values of SWA, NSGA-II, MOEA/D and SDM in 20 independent runs for each test instance, where $best$ and $mean$ denote the minimum and average IGD-values of the final solutions in each test instance with 20 runs, respectively.

water penetration. Further, a random perturbation has been used to represent the contingency of water penetration. Consequently, the SWA combines the deterministic search and heuristic search in a single paradigm, resulting in the fast convergence speed. Further, it has been shown that that the SWA converges to a global optimal solution with probability one. Numerical simulations have shown the promising results of the SWA in comparison with the existing counterparts.

References

[Boyd and Vandenberghe, 2004] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, 2004.

[Davidon, 1991] W. Davidon. Variable metric method for minimization. *SIAM Journal on Optimization*, 1(1):1–17, 1991.

[Deb *et al.*, 2002] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comp.*, 6(2):182–197, 2002.

[Deb, 2001] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley, New York, 2001.

[Dorigo *et al.*, 1996] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Systems, Man, and Cybernetics, Part B:*, 26(1):29–41, 1996.

[Hestenes and Stiefel, 1952] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradient for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[Lucidi and Piccialli, 2002] S. Lucidi and V. Piccialli. New classes of globally convexized filled functions for global optimization. *Journal of Global Optimization*, 24:219–236, 2002.

[Oblow, 2001] E.M. Oblow. SPT: a stochastic tunneling algorithm for global optimization. *Journal of Global Optimization*, 20:191–208, 2001.

[Solis and Wets, 1981] Francisco J. Solis and Roger J-B. Wets. Minimization by random search techniques. *Mathematics of Operations Research*, 6:19–30, 1981.

[Van den Bergh and Engelbrecht, 2004] F. Van den Bergh and A. P. Engelbrecht. A cooperative approach to particle swarm optimization. *IEEE, Trans. Evol. Comp.*, 8(3):225–239, 2004.

[Wang and Dang, 2007] Yuping Wang and Chuangyin Dang. An evolutionary algorithm for global optimization based on level-set evolution and latin squares. *IEEE Trans. Evol. Comput.*, 11(5):579–595, 2007.

[Zhang and Li, 2007] Qingfu Zhang and Hui Li. A multi-objective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comp.*, 11(6):712–731, 2007.