Physica A 542 (2020) 123526

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

A Multi-objective Improved Squirrel Search Algorithm based on Decomposition with External Population and Adaptive Weight Vectors Adjustment☆

Yanjiao Wang, Tianlin Du *

School of Electrical Engineering, Northeast Electric Power University, Jilin 132012, China

ARTICLE INFO

Article history:
Received 18 March 2019
Received in revised form 27 September 2019
Available online 30 December 2019

MSC:
00-01
99-00

Keywords:
Multi-objective optimization
Decomposition
Squirrel Search Algorithm
Adaptive weight vectors adjustment

ABSTRACT

In order to further improve the performance of the Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) in solving multi-objective optimization problems, this paper constructs a multi-objective optimization algorithm by taking MOEA/D as the multi-objective framework and the Squirrel Search Algorithm (SSA) as the core evolutionary strategy. Besides, both of them are modified and the Multi-objective Improved Squirrel Search Algorithm based on Decomposition with External Population and Adaptive Weight Vectors Adjustment (MOEA/D-EWA-ISSA) is proposed. MOEA/D-EWA-ISSA establishes an external population for every individual to retain the evolutionary information and maintain the population diversity, external individuals participate in evolution and produce better offspring, the convergence and distribution of Pareto Front (PF) are improved. As for SSA, the jumping search method and the progressive search method are introduced to it, different evolutionary strategies are provided to solve subproblems, which further improves the ability of core evolutionary strategy to solve subproblems and the convergence of the obtained PF. Furthermore, MOEA/D-EWA-ISSA adjusts every weight vector adaptively according to the population's actual evolutionary direction and the representative neighbor weight vectors, the distribution of the obtained PF is improved as well. The experimental results on multi-objective test functions show that the convergence and distribution of PF have obvious improvement when the improved SSA, the improved MOEA/D and the whole MOEA/D-EWA-ISSA are used to solve multi-objective optimization problems.

1. Introduction

A Multi-objective Optimization Problem (MOP) has more than one objective to be optimized, which is also an important and common problem in engineering fields [1,2], a minimizing MOP can be described as follows:

$$\begin{array}{ll} \textit{minimize} & F(x) = (f_1(x), \ldots, f_m(x))^T \\ \textit{subjectto} & x \in \Omega \end{array} \quad (1)$$

☆ This research was funded by the National Natural Science Foundation of China under grants NO. 61501107 and NO. 61603073, and the Project of Scientific and Technological Innovation Development of Jilin NO. 201750227 and NO. 201750219.

* Correspondence to: Changchun Road, School of Electrical Engineering, Northeast Electric Power University, Jilin 132012, China.
E-mail address: 2201700466@neepu.edu.cn (T. Du).

https://doi.org/10.1016/j.physa.2019.123526

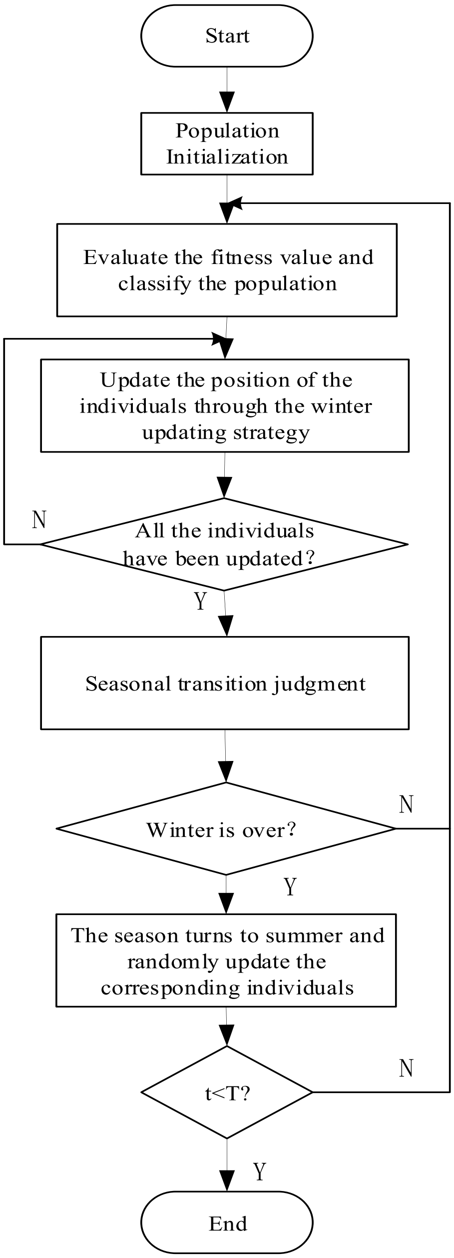


Fig. 1. The procedure of the standard SSA.

where $F = (f_1, \ldots, f_m)^T$ is m-dimensional object vector, $x = (x_1, \ldots, x_n)$ is n-dimensional decision vector and Ω is n-dimensional decision space. The set of Pareto optimal solutions in decision space is called Pareto optimal solution set (PS) and set of target vectors corresponding to all Pareto optimal solutions is called Pareto Front (PF) [3].

Instead of optimizing the MOP as a whole, the Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) decomposes the MOP into N subproblems by weight vectors, which makes the optimization process more efficient [4]. The studies on MOEA/D mainly include the following aspects.

1.1. Weight vectors adjustment

The weight vectors of MOEA/D are uniformly distributed, however, the PF obtained by MOEA/D will be uneven if the ideal PF is not ideal hyperplane. Guo et al. [5] adjusted the weight vectors according to PF's shape but the modification is not suitable for discontinuous PF; MOEA/D-AWA proposed by Qi et al. [6] deletes the overcrowded subproblems and

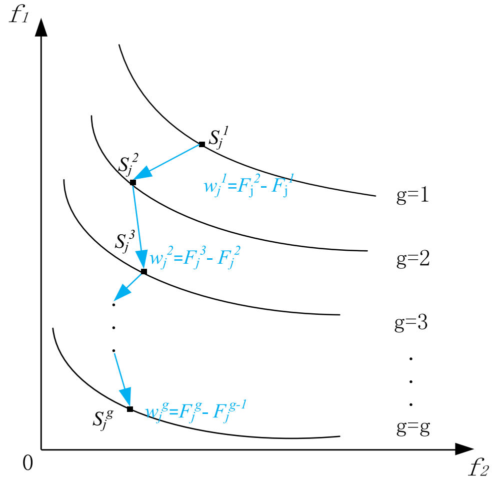


Fig. 2. Changing process of Pareto Fronts.

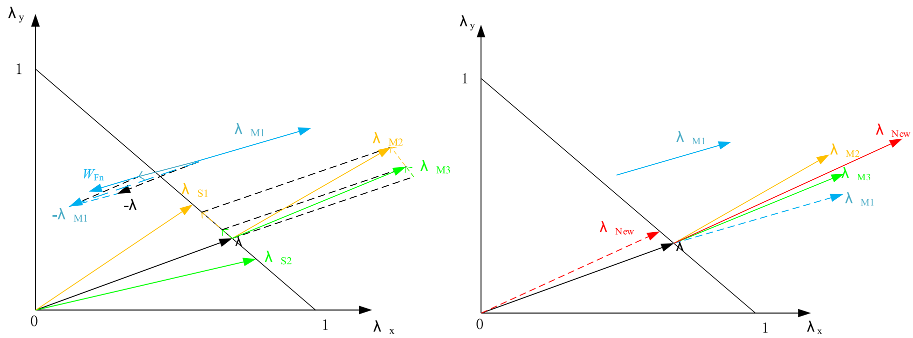
0. 0512	0. 0488	0. 0476	. . .	0. 0216	0. 0193	0. 0186	0. 0156	. . .	0. 0369	0. 0382	0. 0407

(a) st_j of all subproblems

1	2	100	3	4	99	98	. . .	42	40	39	41

(b) Corresponding number of subproblems after sorting ST in descending order

Fig. 3. Selection of representative neighbor weight vectors.



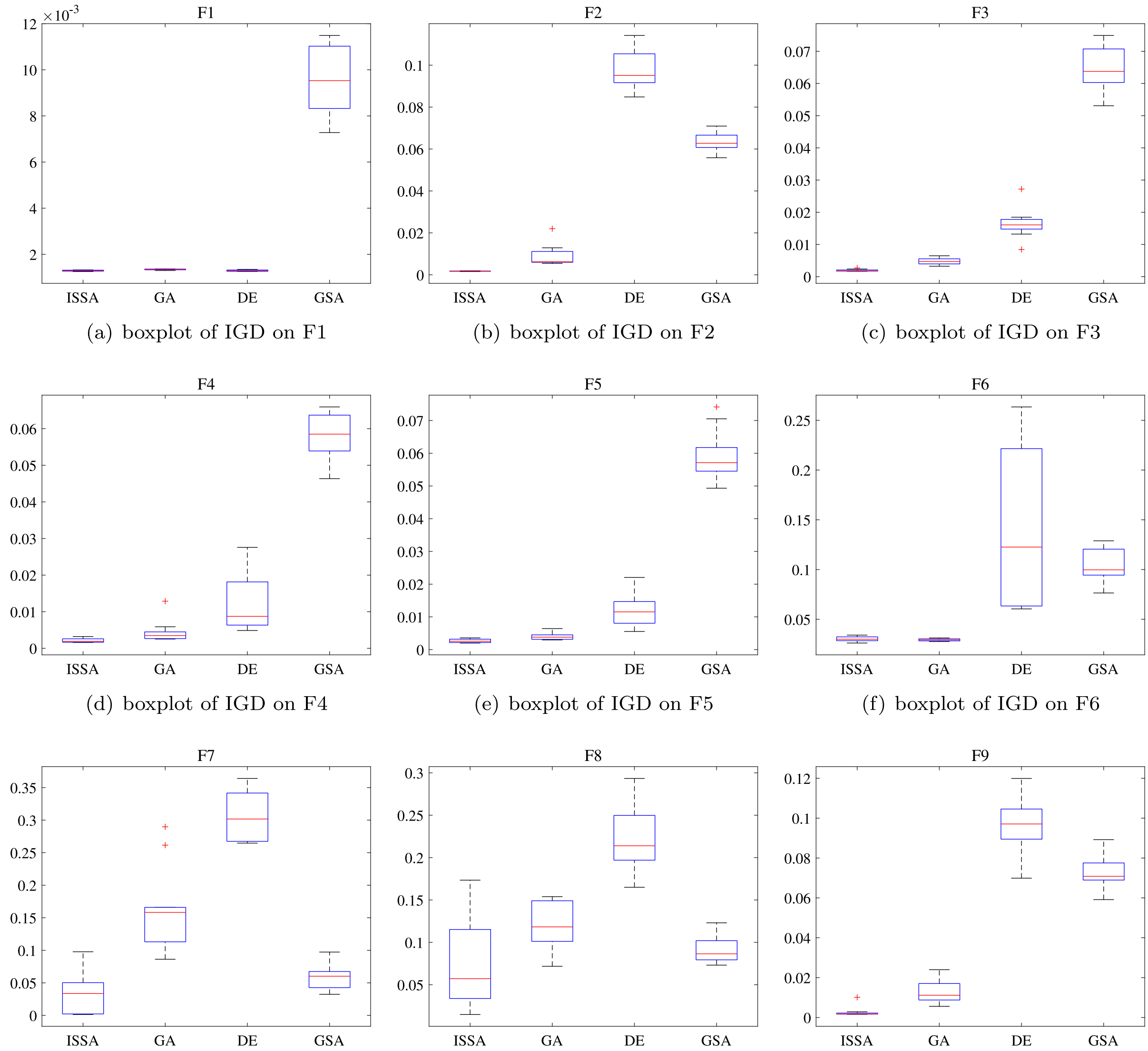
(a) Weight vector adaptive process (b) Weight vector adaptive result

Fig. 4. Weight vector adaptive adjustment.

adds new subproblems into the sparse regions, but it also generates other overcrowded or sparse regions. In original MOEA/D, weight vectors with uniform distribution are generated at the start of optimization, except for adjusting the weight vectors during the optimization, Tan et al. generates the weight vectors by uniform design [7], which causes more uniform distribution of weight vectors especially for three objective functions, however, the solution effect is worse for the search space which is more difficult to optimize.

1.2. Improvement of decomposition approaches

There are three decomposition approaches: Weighted Sum Approach (WS), Tchebycheff Approach (TCH) and Penalty-based Boundary Intersection Approach (PBI) in MOEA/D. Sato [8] proposed inverted PBI (IPBI), solutions are evolved from

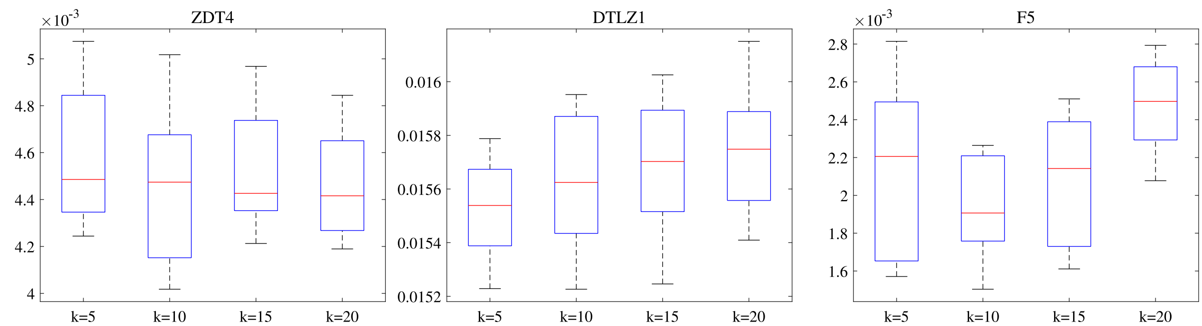


(a) boxplot of IGD on F1 (b) boxplot of IGD on F2 (c) boxplot of IGD on F3

(d) boxplot of IGD on F4 (e) boxplot of IGD on F5 (f) boxplot of IGD on F6

(g) boxplot of IGD on F7 (h) boxplot of IGD on F8 (i) boxplot of IGD on F9

Fig. 5. Boxplots of LZ09.



(a) boxplot of ZDT4 on different k (b) boxplot of DTLZ1 on different k (c) boxplot of F5 on different k

Fig. 6. Boxplots of MOEA/D-EWA-ISSA on different weight vectors adjustment frequencies.

the nadir point by maximizing the scalarizing function value but the parameter θ still has great impact on PF. Jiang et al. [9] proposed a two-phase (TP) strategy, TCH is adopted in the first phase, the reverse TCH which is similar to IPBI will be used in the second phase if the PF is convex. As a result, the MOP with convex PF has much better performance but the one with nonconvex PF has not obvious improvement.

1.3. The alternative mechanism

MOEA/D requires the offspring of each individual replace the individuals only locate in its neighbor, Wang et al. proposed Adaptive Replacement Strategies (AGR) [10], which makes the offspring can replace individuals outside its corresponding neighbor, besides, the adaptive size of neighbor explore the search space in later evolutionary stage. However, the comparison of the offspring and each individual increases the computation and decreases the efficiency.

1.4. Selection and improvement of core evolutionary strategy

The convergence abilities of simulated binary crossover (SBX) and polynomial mutation in original MOEA/D are limited. Revising the core evolutionary strategy is also an effective method to improve the convergence of PF. V Venske S M et al. [11] took $DE/rand/1$, $DE/rand/2$ and $DE/nonlinear$ as the core evolutionary strategy, the convergence of PF improves but the distribution has almost no change. Bi et al. [12] proposed multi-objective gravitational search algorithm based on decomposition (MOGSA/D) using Gravitational Search Algorithm (GSA) to improve the convergence, however, the process of generating weight vectors by prejudgment is complex and difficult to realize.

Some new evolutionary algorithms are proposed in recent years [13–15], evolutionary algorithms with better convergence performance are also applied into complex optimization problems and engineering applications [16–18]. Concerning the Squirrel Search Algorithm (SSA) has better optimization performance [19], this paper proposes a Multi-objective Improved Squirrel Search Algorithm based on Decomposition with External Population and Adaptive Weight Vectors Adjustment (MOEA/D-EWA-ISSA), which takes SSA as the core evolutionary strategy. Both SSA and the MOEA/D framework are modified in order to further improve the convergence and the distribution of the obtained PF, the specific innovations are as follows: (a) Design different searching strategies for SSA: MOEA/D-EWA-ISSA is the integration of MOEA/D and SSA, instead of simple superposition, the jumping search method and the progressive search method are introduced into SSA according to the characters of MOEA/D and SSA, different strategies are suitable for different evolutionary stages and improve the convergence speed and distribution of PF. (b) Establish an extra population for every individual: population diversity is an important factor in both single-objective optimization and multi-objective optimization, extra population is used to retain the evolutionary information and maintain the population diversity, which further improves the convergence and the distribution. (c) Weight vectors adaptive adjustment: weight vectors are unique to MOEA/D, which provides convergence efficient but still has some disadvantages on distribution, MOEA/D-EWA-ISSA adjusts every weight vector by population's actual evolutionary direction and representative neighbor weight vectors according to the distribution of PF. Allocate the computing resources adaptively and improve the distribution. The experimental results on the multi-objective test functions show that the convergence and distribution of MOEA/D-EWA-ISSA have obvious advantages especially on complex optimization problems.

The remaining sections are arranged as follows: Section 2 reviews the basic SSA and the original MOEA/D. Section 3 presents the proposed improved SSA (ISSA) and the improved MOEA/D with external population and adaptive weight vectors adjustment (MOEA/D-EWA). The experiments and results analysis are reported in Section 4. Section 5 concludes this paper and the last two parts are the acknowledgments and references.

2. Introduction of related theories

2.1. The standard SSA

SSA divides the population with P individuals into three types: F_h is the individual with the best fitness value, F_a contains N_{fs} $(1 < N_{fs} < P)$ individuals whose fitness values are worse than F_h in order and N_{fs} can be various according to different problems, the remaining individuals are recorded as F_n. The update rules for each iteration are as follows: F_h remains unchanged, individuals in F_a glide towards F_h, individuals in F_n glide towards any individual randomly selected from F_h and F_a. Formula (2) is the specific description:

$$\begin{cases} FS_i^{t+1} = FS_i^t + d_g \times G_c \times (F^t - FS_i^t) & \text{if } r > P_{dp} \\ FS_{ij}^{t+1} = FS_{Lj} + Lévy(n) \times (FS_{Uj} - FS_{Lj}) & \text{if } r \leqslant P_{dp} \end{cases} \tag{2}$$

F is the destination of FS_i; r is a random number between 0 and 1; P_{dp} valued 0.1 represents the predators appearance probability, there is no predator when $r > P_{dp}$ and FS_i glides towards F, the predators appear when $r \leqslant P_{dp}$ and FS_i updates the position randomly; FS_{Uj} and FS_{Lj} are the upper and lower bounds of the search range on every dimension; the detailed calculation process of d_g, G_c and *Lévy(n)*(n) are in the Ref. [19].

Judge the season according to formula (3) and formula (4) after all the individuals updating their positions. Season is winter when $S_c^t > S_{\min}$ and execute the next iteration directly; season is summer when $S_c^t \leqslant S_{\min}$, the individuals who glide towards F_a and do not encounter the predators randomly update their positions in the same way as $r \leqslant P_{dp}$ in formula (2).

$$S_c^t = \sqrt{\sum_{k=1}^{D}(F_{ai,k}^t - F_{h,k}^t)^2} \quad i = 1, 2, \ldots, N_{fs} \tag{3}$$

$$S_{\min} = 10e^{-6}/(365)^{t/(T/2.5)} \tag{4}$$

t is the current iteration, T is the total iteration and D is the dimension of individuals. The updating process of the standard SSA is shown in Fig. 1.

2.2. The original MOEA/d

MOEA/D decomposes the MOP into N subproblems by N uniformly distributed weight vectors and TCH, each weight vector corresponds to a subproblem and each subproblem corresponds to an individual. Each weight vector has a corresponding neighborhood containing T weight vectors closest to itself, therefore, each subproblem has a neighborhood with T subproblems and each individual has a neighborhood with T individuals. The weight vector is shown in formula (5) and the subproblem is shown in formula (6).

$$\lambda^j = (\lambda_1^j, \ldots, \lambda_m^j)^T \quad j = 1, \ldots, N \quad \lambda_i^j \geqslant 0 \quad i = 1, \ldots, m \sum_{i=1}^{m} \lambda_i^j = 1 \tag{5}$$

$$\begin{array}{ll} minimize & g^{te}(x|\lambda, z^*) = \max\{\lambda_i|f_i(x) - z_i^*|\} \\ subjectto & x \in \Omega \end{array} \tag{6}$$

$z^* = (z_1^*, \ldots, z_m^*)^T$ is the reference point and $z_i^* = \min\{f_i(x)|x \in \Omega\}$, there is a weight vector corresponding to subproblem (6) for each Pareto optimal solution x^*.

In each iteration, each subproblem is solved by genetic operation with the corresponding individual and its neighbor individuals, and the offspring y' can replace all the individuals in the neighborhood as long as it satisfies $g^{te}(y'|\lambda^j, z^*) \leqslant g^{te}(x^j|\lambda^j, z^*)$.

3. MOEA/D-EWA-ISSA

In order to further improve the performance of solving the MOP, this paper proposes MOEA/D-EWA-ISSA, which modifies the MOEA/D framework and takes the improved SSA as the core evolutionary strategy, the pseudo code of MOEA/D-EWA-ISSA is shown in Algorithm 1.

Algorithm 1 MOEA/D-EWA-ISSA

Step (1) Initialization:
Generate N weight vectors $\lambda_1, \ldots, \lambda_N$ with uniform distribution;
Determine the neighbor of each weight vector $\lambda_j, j = 1, \ldots, N$;
Generate the initial population $X = (x^1, \ldots, x^N)$ and $x^i = (x_1^i, \ldots, x_D^i)$;
Set external population $Ey = \varnothing$ for every individual;
Calculate the target vectors $F_1, \ldots, F_N$;
Initialize the reference point $z = (z_1, \ldots, z_m)$;
Step (2) Update:
for individual in the X **do**
  Reproduce the offspring y' ;
  Update the neighbor: y' can take replace any individual in the neighbor as long as it satisfies $g^{te}(y'|\lambda^j, z^*) \leqslant g^{te}(x^j|\lambda^j, z^*)$;
  Calculate the w_j^g;
  Record *num_alt* and update the extra population Ey;
  Update the reference point z;
end for
Step (3): Adjust the weight vector:
if the evolution completes $k\%$ **then**
  Calculate ST;
  for weight vector $\lambda_j, j = 1, \ldots, N$ **do**
    Calculate $W_{F_{jn}}$ and select λ_{rs1} and λ_{rs2};
    Calculate λ_{M1}, λ_{M2}, λ_{M3} and generate the new weight vector λ_{jnew};
  end for
end if

Update the size of neighbor and update the neighbor of each weight vector;
Step (4): Stopping condition:
If g is up to G, stop and output PF, otherwise, return to **Step 2**.

3.1. Establishment and renewal of external population

According to the original MOEA/D, the offspring y' will be abandoned if there is no neighbor individual satisfying $g^{te}(y'|\lambda^j, z^*) \leqslant g^{te}(x^j|\lambda^j, z^*)$, however, y' carries evolutionary information different from y, which may produce better individuals and promote the convergence of the algorithm. MOEA/D-EWA-ISSA establishes an external population Ey for every individual in order to make y' possible to participate in evolution. The specific methods are as follows.

Firstly, the upper limit of Ey is equal to T, both of them are adjusted by formula (7) every time the evolution completes k%. Next, num_alt is used to record the number of neighbor individuals replaced by y'. If $num\_alt = 0$, the corresponding y' enters Ey. If $Ey \neq \varnothing$ and $num\_alt > T_g$, empty Ey because the current subproblem and its neighbor subproblems are far from the optimal solution. If $0 < num\_alt \leqslant T_g$, Ey remains unchanged. If the size of Ey is up to T, delete y' which minimizes the current subproblem.

$$\begin{aligned} T &= T_0/2 + T_g \\ T_g &= \lfloor T_0/2 \times (1 - g/G) \rfloor \end{aligned} \tag{7}$$

T_0 is the initial value of T, g is the current iteration and G is the total iteration.

The upper limit of Ey decreases gradually and Ey updates faster. Ey contains similar information to the current individual, which is easier to produce better offspring. When the size of Ey is up to the upper limit, the deleted external individual has the closet fitness value to the current individual, which maintains the population diversity and improves the distribution of obtained PF. Besides, the size of neighborhood also decreases gradually, more concentrated evolutionary information can be used for every subproblem, which develops the search space more sufficiently and improves the convergence of obtained PF.

3.2. The improved SSA

3.2.1. The motivation of revising SSA

In early stage of solving MOPs, the fitness values of neighbor individuals on g_i^{te} are not good enough and the individuals are far away from optimal solutions, which should pay more attention to improve the convergence speed to improve the convergence of PF. Individuals get better fitness values on their corresponding subproblems with the evolution progressing, the neighbor subproblems have similar solutions, however, the individuals are still far from the optimal solutions of their corresponding subproblems, the individuals distribute dispersedly, in order to improve the distribution of PF, the search space should be fully developed to maintain the population diversity. In the later stage of optimization, individuals are closer to their corresponding optimal solutions gradually. Explore the search space deeply will further improve the convergence of PF.

If replace the evolutionary strategy of MOEA/D by SSA directly, according to the updating rules of SSA: in early evolutionary stage, the fitness values of individuals in sub_P on g_i^{te} are not good enough and y' is likely to replace more than one neighbor individual, F_h and F_{ai} $(i = 1, \ldots, N_{fs})$ may be the same, therefore, $S_c^g \leqslant S_{\min}$ and the squirrel population is in summer; individuals in sub_P no longer overlap with the evolution progressing, due to the neighbor individuals are far from the optimal solutions, the distances between excellent individuals may satisfy $S_c^g > S_{\min}$ and the population is in winter; in the later stage of the evolution, the individuals in sub_P are closer to their corresponding optimal solutions, the distances between the excellent individuals get smaller again, $S_c^g \leqslant S_{\min}$ and the squirrel population is in summer again.

Considering the analysis above, take SSA as the core evolutionary strategy, fully develop the search space when $S_c^g > S_{\min}$ and the squirrel population is in winter; improve the convergence speed and deeply explore the search space when $S_c^g \leqslant S_{\min}$ and the squirrel population is in summer. Besides, due to different searching methods have different impact on improving the convergence speed and maintaining the population diversity, design two different searching methods for winter and summer respectively.

3.2.2. The description of the improved SSA

MOEA/D-EWA-ISSA takes the improved SSA as the core evolutionary strategy to solve subproblems, the obtained PF is the set of current optimal solutions of all subproblems. The variables involved in ISSA are as follows:

g_i^{te}: the divided ith subproblem;
FS_i: the individual corresponding to g_i^{te};
sub_P: the subpopulation of FS_i, which contains the individuals corresponding to all neighbor subproblems;
F_h: the best individual in sub_P;
F_{ai} $(i = 1, \ldots, N_{fs})$: individuals whose fitness values worse than F_h in order;
$dist(FS_i, F_h)$: the distance between FS_i and F_h;
$dist\_min$: the minimum distance between F_h and every individual in sub_P;

dist_ave: the average distance between F_h and every individual in *sub_P*;

Fr, Fr_1, Fr_2: $Ey = \varnothing$: Fr, Fr_1 and Fr_2 are randomly selected in *sub_P* which are different form F_h and F_a; $Ey \neq \varnothing$: Fr and Fr_2 are randomly selected in Ey; Fr_1 is randomly selected in *sub_P* which is different form F_h and F_a. The pseudo code of ISSA is shown in Algorithm 2.

1. Winter search method when $S_c^t > S_{\min}$

(a) $r > P_{dp}$, no predators appear

(1) The jumping search method

If $dist(FS_i, F_h) < dist\_ave$, there is a shorter distance between FS_i and F_h. Besides, the optimal solution of neighbor subproblems are similar to each other, the offspring will provide better evolutionary information for the neighbor subproblems of g_i^{te} if enhance the guidance of F_h. Therefore, the jumping search method requires FS_i glide towards F_h by formula (8):

$$FS_i^{t+1} = FS_i^t + d_g \times G_c \times (F_h^t - FS_i^t) + d_g \times G_c \times (F_{ai}^t - F_r^t) \tag{8}$$

FS_i glides only towards F_h, enhance the guidance of F_h and improve the convergence speed. Considering the updating principle of SSA requires F_h remains unchanged every iteration, if $FS_i = F_h$, FS_i will not be updated and the optimization process will stagnate. Take differential vector between F_a and neighbor individual or external individual as disturbance, which utilizes the evolutionary information more sufficiently and maintains the population diversity.

(2) The progressive search method

If $dist(FS_i, F_h) \geqslant dist\_ave$, there is a longer distance between FS_i and F_h. Besides, the neighbor subproblems are similar to each other, solving processes of g_i^{te} and its neighbor will fall into local optimal with single search method, the search space needs deep exploration. The progressive search method selects r_d dimensions by formula (9), the lth selected dimension mutates according to formula (10).

$$r_d = ceil(D \times dist\_min / dist\,(FS_i, F_h)) \tag{9}$$

$$FS_i^{t+1}(l) = FS_i^t(l) \pm L\acute{e}vy(n) \times (FS_{Ul} - FS_{Ll}) \tag{10}$$

Due to the neighbor subproblems have similar solutions, individuals closer to F_h have more dimensions to be mutated to supplement the population diversity and avoid the subpopulations fall into local optimal, besides, individuals further from F_h retain more evolutionary information to ensure the evolutionary direction as well as explore the search space.

(b) $r \leqslant P_{dp}$, predators appear

FS_i is considered to be dead, select two individuals Fr_1 and Fr_2 randomly and produce a new one by SBX.

According to the basic SSA, the individuals encounter the predators update their locations by generating a new one randomly, excessive randomness decreases the convergence speed. However, the individuals in the population or the external individuals carries the current evolutionary information, the new generated individual is the result of the communication among the current evolutionary information, which maintains the population diversity as well as ensure the convergence speed.

2. Summer search method when $S_c^t \leqslant S_{\min}$

(a) $r > P_{dp}$, no predators appear

(1) The jumping search method

When $dist(FS_i, F_h) < dist\_ave$, in order to explore the search space deeply, the jumping search method updates the position by formula (11):

$$FS_i^{t+1} = F_h^t + d_g \times G_c \times (F_h^t - FS_i^t) + d_g \times G_c \times (F_{ai}^t - F_r^t) \tag{11}$$

Formula (11) takes F_h as the base vector and pays more attention to explore the search space, which improves the convergence speed. The two disturbances are: the differential vector between FS_i and F_h, the differential vector between F_{ai} and Fr, which can supplement the population diversity at the same time.

(2) The progressive search method

When $dist(FS_i, F_h) \geqslant dist\_ave$, the progressive search method selects r_d dimensions by formula (9), the lth selected dimension mutates according to formula (12).

$$FS_i^{t+1}(l) = F_h^t(l) \pm L\acute{e}vy(n) \times (FS_{Ul} - FS_{Ll}) \tag{12}$$

Considering the individuals are close to each other, the selected dimensions are mutated based on F_h to communicate with the best individual, improve the convergence speed and maintain the population diversity. According to formula (8) and formula (10), formula (8) mutates the individuals based on FS_i, which further explores the search space and pays more attention to retain the evolutionary information; formula (10) mutates the individuals based on F_h, which maintains the evolutionary information and pays more attention to deeply search around the elite individuals.

(b) $r \leqslant P_{dp}$, predators appear

FS_i is considered to be dead, select two individuals Fr_1 and Fr_2 randomly and produce a new one by SBX.

Comparing the jumping search method and the progressive search method, the jumping search method requires the individuals glide towards the neighborhood optimal individual or update the position based on the neighborhood optimal individual, which pays more attention on improving the convergence speed and maintain the population diversity by perturbation vectors. The progressive search method mutates part of dimensions of the individuals and the other dimensions remain unchanged, which pays more attention on maintaining the population diversity and fully exploring the search space, besides, the communication with the neighborhood optimal individual will improve the convergence speed as well.

Algorithm 2 The improved SSA

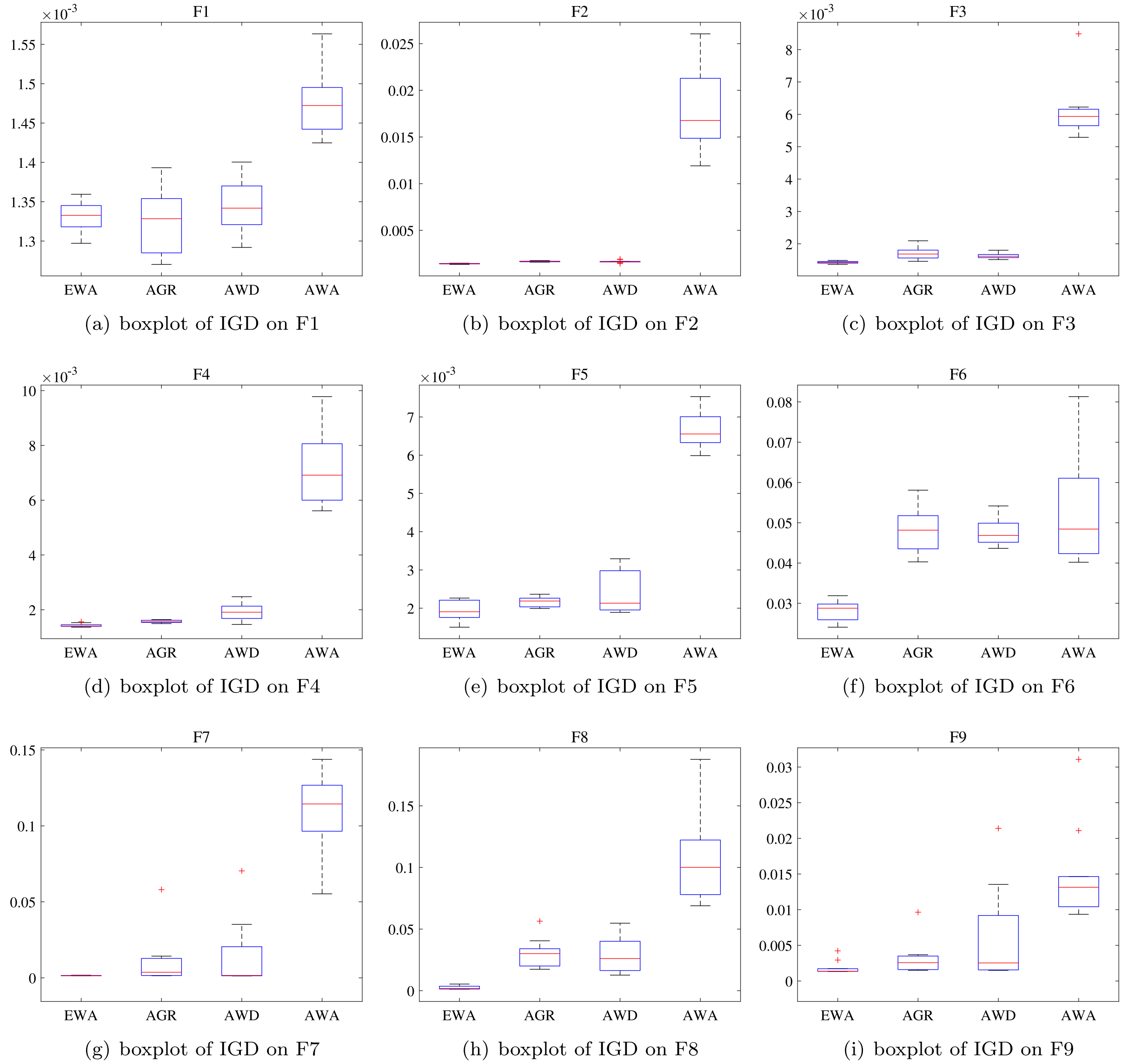
if $S_c^t \geqslant S_{\min}$ **then**
  season = winter;
else
  season = summer;
end if
if season == winter **then**
  if $r > P_{dp}$ **then**
    if $dist(FS_i, F_h) < dist\_ave$ **then**
      $FS_i^{t+1} = FS_i^t + d_g \times G_c \times (F_h^t - FS_i^t) + d_g \times G_c \times (F_{ai}^t - FS_r^t)$ % the jumping search method;
    else
      $r_d = ceil(D \times dist\_\min / dist(FS_i, F_h))$ % the progressive search method ;
      select r_d dimensions randomly;
      change them through $Lévy(n)$ flight based on FS_i;
      e.g. assuming the selected dimension is l;
      $FS_i^{t+1}(l) = FS_i^t(l) \pm Lévy(n) \times (FS_{Ul} - FS_{Ll})$;
    end if
  end if
  if $r \leqslant P_{dp}$ **then**
    select two individuals Fr_1 and Fr_2 from the subpopulation ;
    produce the new one through SBX;
  end if
end if
if season == summer **then**
  if $r > P_{dp}$ **then**
    if $dist(FS_i, F_h) < dist\_ave$ **then**
      $FS_i^{t+1} = F_h^t + d_g \times G_c \times (F_h^t - FS_i^t) + d_g \times G_c \times (F_{ai}^t - FS_r^t)$ % the jumping search method;
    else
      $r_d = ceil(D \times dist\_\min / dist(FS_i, F_h))$ % the progressive search method;
      select r_d dimensions randomly;
      change them through $Lévy(n)$ flight based on F_h;
      e.g. assuming the selected dimension is l;
      $FS_i^{t+1}(l) = F_h^t(l) \pm Lévy(n) \times (FS_{Ul} - FS_{Ll})$;
    end if
  end if
  if $r \leqslant P_{dp}$ **then**
    select two individuals Fr_1 and Fr_2 from the subpopulation;
    produce the new one through SBX;
  end if
end if

3.3. Adaptive weight vectors adjustment

Although the subproblems are decomposed by uniformly distributed weight vectors, the PF obtained by MOEA/D will be uneven if the ideal PF is not ideal hyperplane. In order to improve the distribution of the obtained PF. It is obvious that the nonuniform distribution of PF causes different subproblems have different densities of the neighbor individuals' distribution, MOEAD-EWA-ISSA judges the distribution of PF and adjusts the weight vectors adaptively every time k% of the whole evolution is completed, k can be different according to the specific problem. Every weight vector is adjusted adaptively by the actual evolutionary direction of PF and its representative neighbor weight vectors, as a result, there are more weight vectors in the sparse region and less weight vectors in the crowded region. The details are as follows.

3.3.1. Distribution judgment

A large number of studies show that the sparse weight vectors cause the neighbor individuals' fitness values have greater differences on the same subproblem, similarly, the crowded weight vectors cause the neighbor individuals' fitness values have smaller differences on the same subproblem. Therefore, in terms of every subproblem g_j^{te}, the distribution



(a) boxplot of IGD on F1 (b) boxplot of IGD on F2 (c) boxplot of IGD on F3

(d) boxplot of IGD on F4 (e) boxplot of IGD on F5 (f) boxplot of IGD on F6

(g) boxplot of IGD on F7 (h) boxplot of IGD on F8 (i) boxplot of IGD on F9

Fig. 7. Boxplots of LZ09.

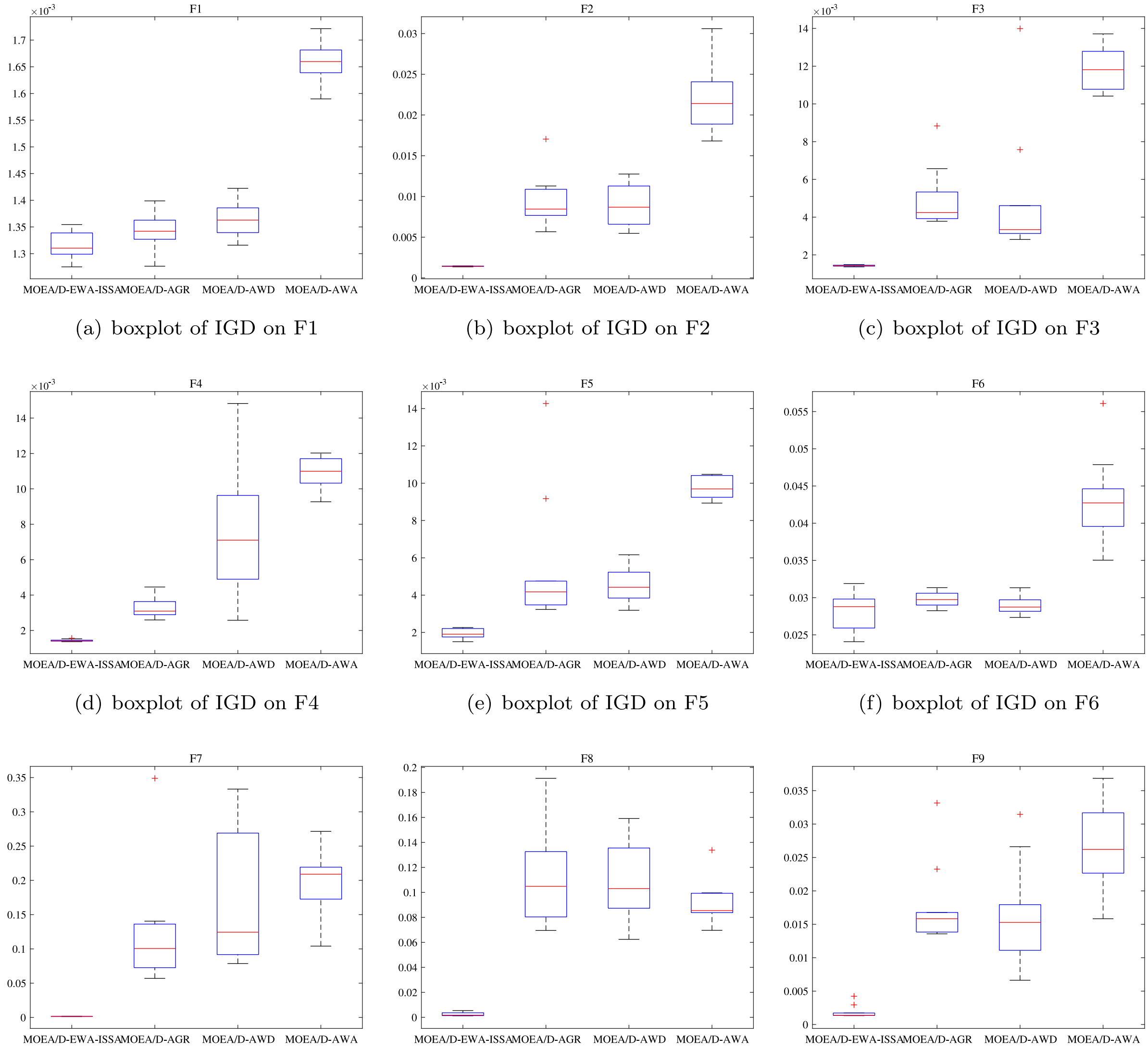
of PF can be analyzed according to fitness values' variance of all neighbor individuals on g_j^{te}. Take g_1^{te} as an example, the specific distribution judgment of PF is as follows.

Assuming that the corresponding weight vector of g_1^{te} is λ_1, T is equal to 20 and the neighbor weight vectors of λ_1 are $\lambda_1 \sim \lambda_{20}$, therefore, the corresponding individual of λ_1 is FS_1 and its neighbor individuals are $FS_1 \sim FS_{20}$. Firstly, calculate the fitness values of $FS_1 \sim FS_{20}$ on g_1^{te}. Next, calculate the variance of the 20 fitness values and name it as st_1. Calculate repeatedly in this way, N variances $st_1 \sim st_N$ can be obtained and note them as ST. The minimum of ST is named as st_min, the corresponding subproblem has the concentrated distribution and noted as $g_{densest}^{te}$. The greater value of st_j represents the greater difference among neighbor individuals on g_j^{te} and the sparser distribution around λ_j and vice versa. ST is used to adjust the weight vectors and the specific method is introduced in 3.3.3.

3.3.2. Selection of auxiliary vectors and calculation of weight vectors

(a) Generation of $W_{F_{jn}}$ which is related to the actual evolutionary direction of PF

Assuming that the obtained PF after each iteration is shown in Fig. 2, S_j^g represents the location of FS_j on the PF obtained by gth iteration and $F_j^g = (f_{1j}^g, f_{2j}^g)$ is the coordinate of S_j^g. Meanwhile, F_j^g represents the target vector of FS_j obtained by gth iteration, the difference between the target vectors obtained by two adjacent iterations of FS_j is recorded as w_j^g and



(a) boxplot of IGD on F1 (b) boxplot of IGD on F2 (c) boxplot of IGD on F3

(d) boxplot of IGD on F4 (e) boxplot of IGD on F5 (f) boxplot of IGD on F6

(g) boxplot of IGD on F7 (h) boxplot of IGD on F8 (i) boxplot of IGD on F9

Fig. 8. Boxplots of LZ09.

calculated by formula (13):

$$w_j^g = F_j^g - F_j^{g-1} g > 1 \tag{13}$$

The value of k divides the whole evolutionary process into a certain number of substages, when the evolution comes to the last iteration of nth substage, FS_j generates the auxiliary vector $W_{F_{jn}}$ according to the $W_{j1} \sim W_{j(n-1)}$ produced by the previous $n-1$ substages. $W_{F_{jn}}$ is used to adjust the jth weight vector. The details are introduced in 3.3.3. W_{jn} and $W_{F_{jn}}$ are calculated by formula (14) and formula (15), w_j^g in formula (14) and formula (15) is calculated by formula (13).

$$\begin{aligned} W_{j1} &= \sum_{g=2}^{G/k} \left(g / \sum_{h=2}^{G/k} h \right) \times w_j^g \\ &= \left(2 / \sum_{h=2}^{G/k} h \right) \times w_j^2 + \left(3 / \sum_{h=2}^{G/k} h \right) \times w_j^3 + \cdots + \left(0.1G / \sum_{h=2}^{G/k} h \right) \times w_j^{G/k} \quad n = 1 \end{aligned}$$

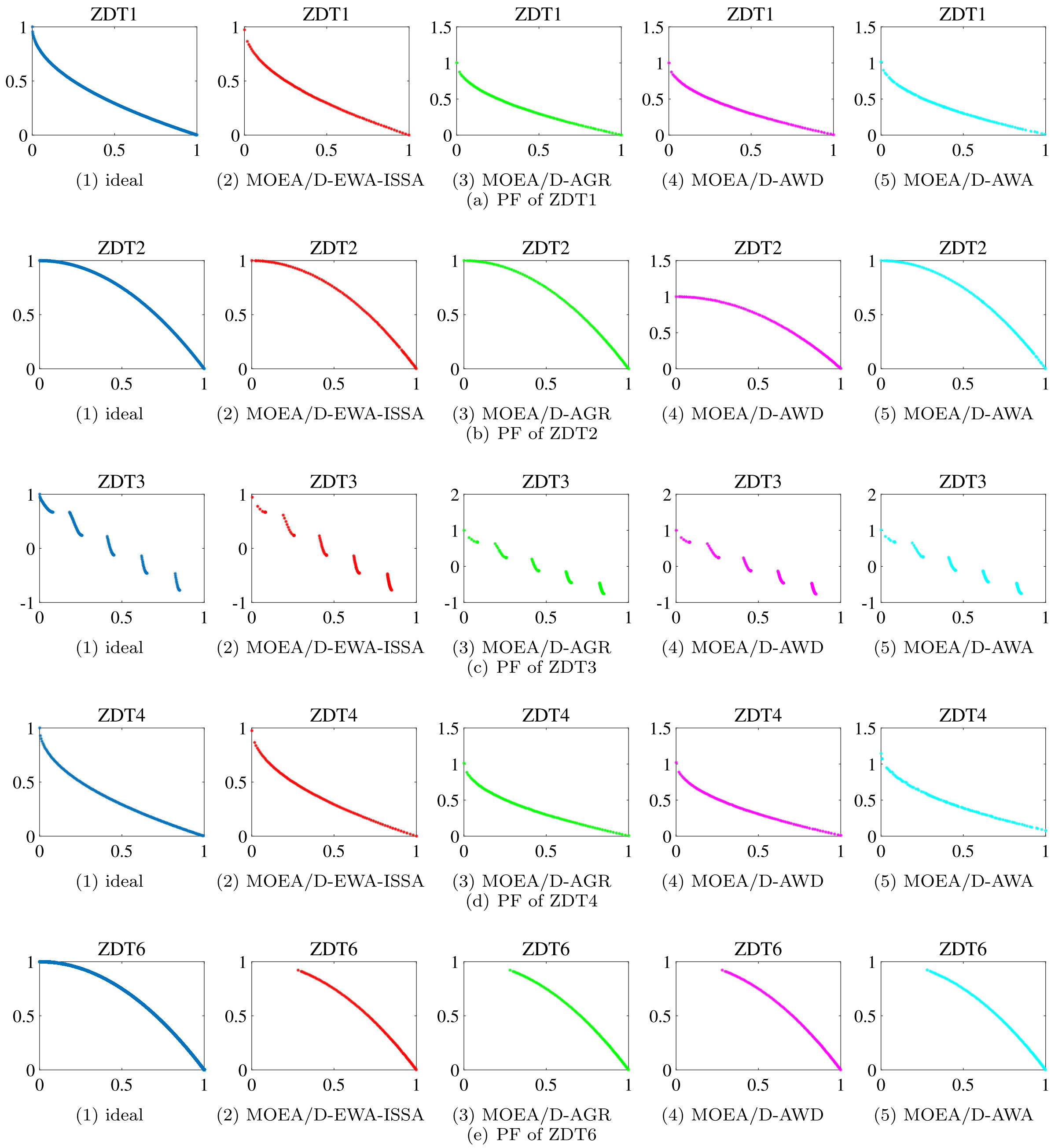


Fig. 9. PF of ZDT and DTLZ.

$$
\begin{aligned}
W_{jn} &= \sum_{g=(n-1)\times G/k+1}^{n\times G/k} \left((g-(n-1)\times G/k) \Big/ \sum_{h=1}^{G/k} h \right) \times w_j^g \\
&= \left(1 \Big/ \sum_{h=1}^{G/k} h \right) \times w_j^{(n-1)\times G/k+1} + \left(2 \Big/ \sum_{h=1}^{G/k} h \right) \times w_j^{(n-1)\times G/k+2} + \cdots \\
&\quad + \left(G/k \Big/ \sum_{h=1}^{G/k} h \right) \times w_j^{n\times G/k} \quad n > 1 \\
W_{F_{j1}} &= W_{j1} \quad n = 1
\end{aligned}
\tag{14}
$$

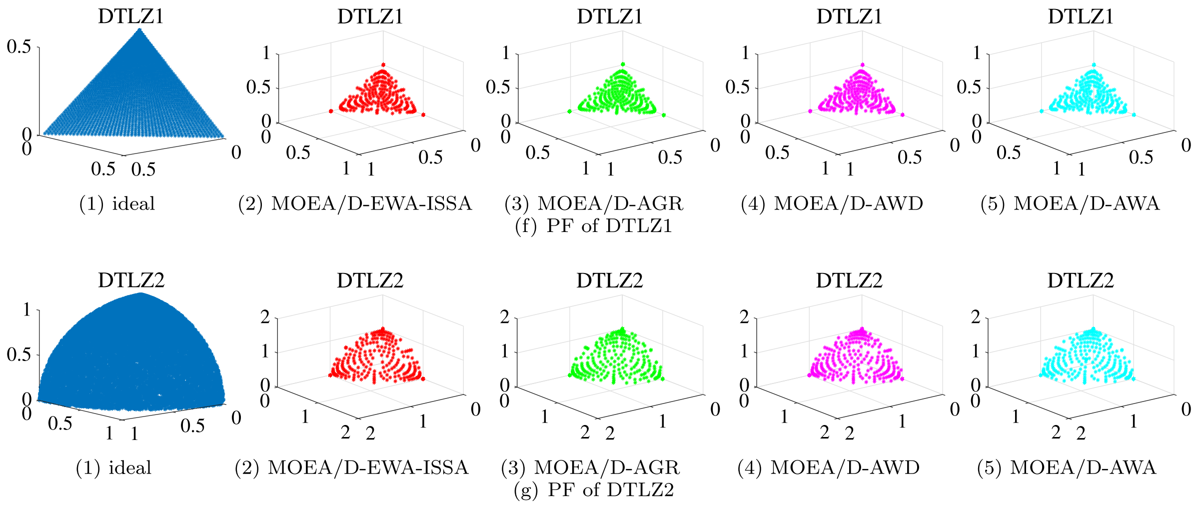


Fig. 9. (*continued*).

$$W_{F_{jn}} = 1/k \times (n-1) \times W_{j(n-1)} + W_{jn} \quad n > 1 \tag{15}$$

(b) Selection of representative neighbor weight vectors λ_{rs1} and λ_{rs2}

For each weight vector λ_j, select the nearest neighbor weight vector with sparser distribution and record it as λ_{rs1}. That is, λ_{rs1} should meet the following requirements:

(1) $st_{rs1} > st_j$;
(2) λ_{rs1} is the neighborhood of λ_j;
(3) λ_{rs1} is closest to λ_j;

If there is no weight vector satisfying the conditions above at the same time, $\lambda_{rs1} = \lambda_j$.

For each weight vector λ_j, select the nearest neighbor weight vector with more concentrated distribution and record it as λ_{rs2}. That is, λ_{rs2} should meet the following requirements:

(1) $st_{rs2} < st_j$;
(2) λ_{rs2} is the neighborhood of λ_j;
(3) λ_{rs2} is closest to λ_j;

If there is no weight vector satisfying the conditions above at the same time, $\lambda_{rs2} = \lambda_j$.

Fig. 3 illustrates the selection of λ_{rs1} and λ_{rs2}.

Fig. 3(a) is the *ST* obtained by MOEAD-EWA-ISSA on ZDT1 after a certain iteration, in order to ensure the qualified λ_{rs1} and λ_{rs2}, sort the numbers in Fig. 3(a) in descending order, Fig. 3(b) shows the corresponding number of every st_j after sorting *ST* in descending order. Assuming that $T = 20$, the neighborhood of λ_1 contains $\lambda_1 \sim \lambda_{20}$, the neighborhood of λ_{41} contains $\lambda_{32} \sim \lambda_{51}$, the neighborhood of λ_{100} contains $\lambda_{81} \sim \lambda_{100}$: if $j = 1$, $r_{s1} = 1$, $r_{s2} = 2$; if $j = 2$, $r_{s1} = 1$, $r_{s2} = 3$; if $j = 100$, $r_{s1} = 100$, $r_{s2} = 99$; if $j = 99$, $r_{s1} = 100$, $r_{s2} = 98$; if $j = 39$, $r_{s1} = 40$, $r_{s2} = 41$; if $j = 41$, $r_{s1} = 39$, $r_{s2} = 41$.

3.3.3. *Generation of* λ_{jnew}

It has mentioned above that MOEAD-EWA-ISSA adjusts the weight vectors every time *k*% of the evolution is completed. If one substage is completed, every individual FS_j will produce a new weight vector λ_{jnew} by the auxiliary vectors $W_{F_{jn}}$, λ_{rs1} and λ_{rs2} introduced in 3.3.2, λ_{jnew} will be used in the next substage. The detailed derivations are as follows.

$$\lambda_{jnew} = \lambda_{M1} + \lambda_{M2} + \lambda_{M3} \tag{16a}$$

$$\lambda_{M1} = \lambda_j - c \times rand \times W_{F_{jn}} \tag{16b}$$

$$\lambda_{M2} = \lambda_j + c \times rand \times (\lambda_{rs1} - \lambda_j) \tag{16c}$$

$$\lambda_{M3} = \lambda_j + c \times L\acute{e}vy(n) \times (\lambda_j - \lambda_{rs2}) \tag{16d}$$

$$c = abs(Norm(0, st\_\min + (st_j - st\_\min)/N)) \times (st_j - st\_\min)/N \tag{17}$$

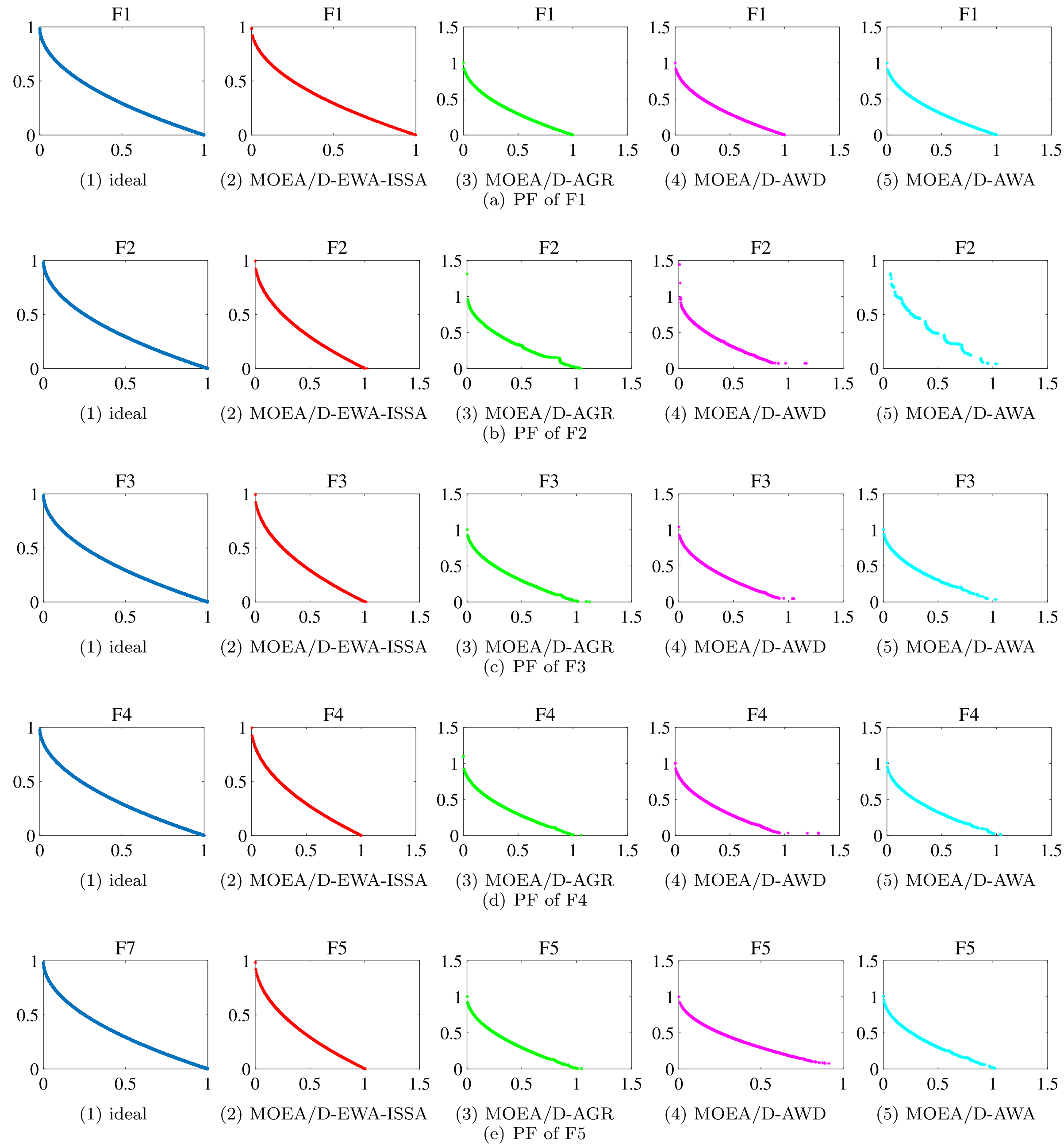


Fig. 10. PF of F9.

c is the coefficient related to st_j, which can reflect the distribution difference of g_j^{te} and $g_{densest}^{te}$; *rand* is a random number between 0 and 1, *Lévy(n)* is a random number which obeys *Lévy(n)* flight, the random number produced by *Norm* obeys Gaussian distribution.

According to formula (16a), λ_{jnew} is the sum weight vector of λ_{M1}, λ_{M2} and λ_{M3}. λ_{M1} is related to the actual evolutionary direction of PF, λ_{M2} and λ_{M3} are related to the representative neighbor vectors. The detailed calculation process is as follows.

(a) According to Fig. 2, the actual evolutionary direction of PF is contrary to the direction of weight vectors, that is, PF evolves towards $-\lambda$. λ_{M1} is the sum of λ_j and $-c \times rand \times W_{F_{jn}}$ in order to adjust the λ_j along the original λ_j. λ_{M1} is produced according to the actual evolutionary direction of PF and λ_{M1} is shown in formula (16b). Besides, if $n > 2$, $W_{F_{jn}}$ contains not only W_{jn} produced by the nth substage but also $1/k \times (n-1) \times W_{j(n-1)}$ produced by the previous substage, therefore, the impact of $W_{F_{jn}} (j = 1, \ldots, n-1)$ become smaller and smaller with the evolution proceeding.

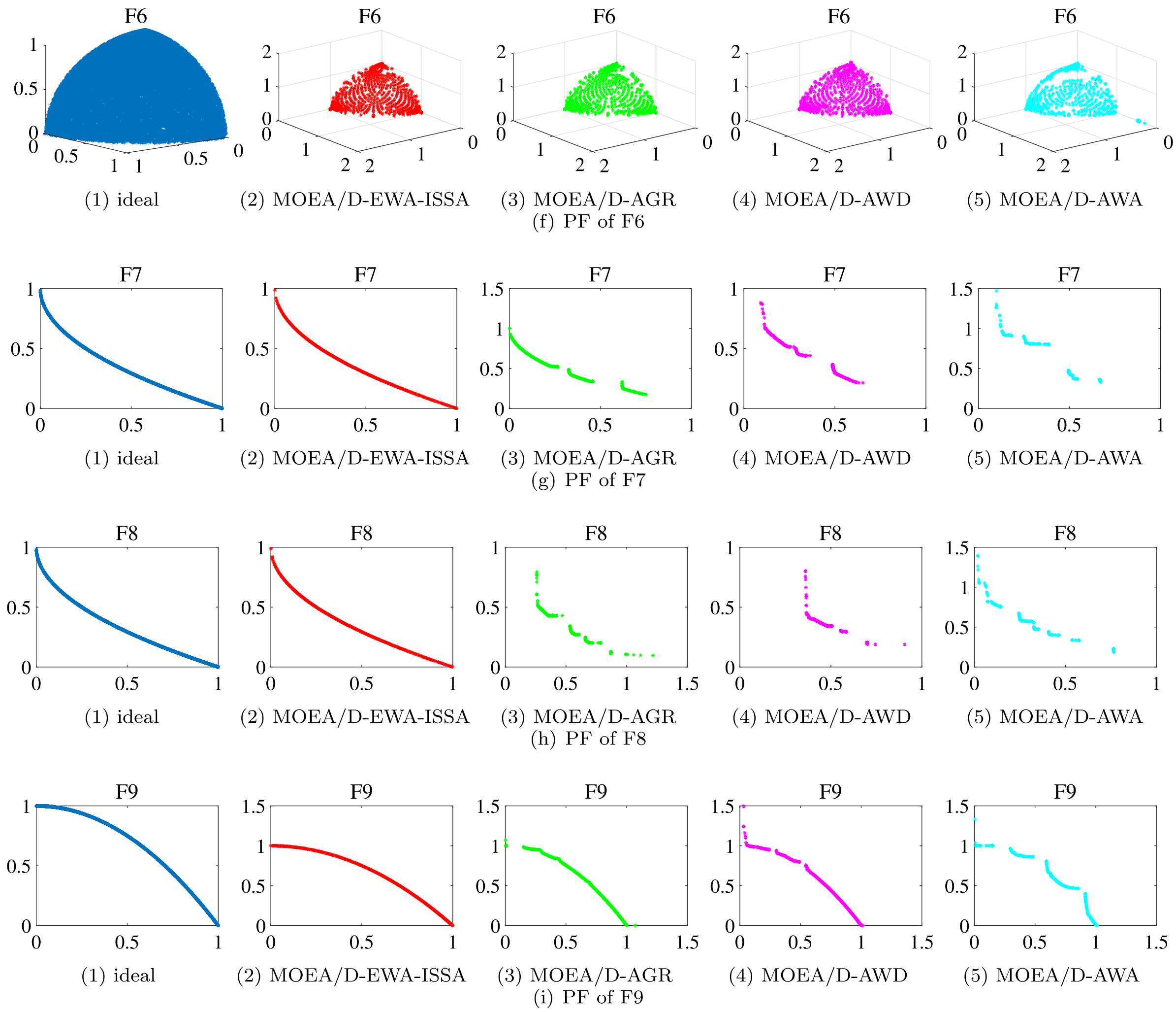


Fig. 10. (*continued*).

(b) According to formula (16c) and Fig. 4(a): λ_{M2} is the sum of λ_j and $c \times rand \times (\lambda_{rs1} - \lambda_j)$, $(\lambda_{rs1} - \lambda_j)$ points to λ_{rs1}, λ_{jnew} will be closer to sparser distributed λ_{rs1}.

(c) According to formula (16d) and Fig. 4(a): λ_{M3} is the sum of λ_j and $c \times rand \times (\lambda_j - \lambda_{rs2})$, $(\lambda_j - \lambda_{rs2})$ points to λ_j rather than λ_{rs2}, λ_{jnew} will be further away from more concentrated distributed λ_{rs2}. Besides, *Lévy(n)* flight leads to flight in short distance with greater possibilities and walk in long distance occasionally, therefore, *Lévy*(n) is smaller than *rand* in most cases, which keeps λ_{jnew} closer to λ_{rs1} and further away from λ_{rs2} as far as possible.

λ_{jnew} is the intersection of $\sum_{i=1}^{m} \lambda_i = 1$ and the sum weight vector of λ_{M1}, λ_{M2}, λ_{M3}, λ_{jnew} is shown in Fig. 4(b).

In summary, the adaptive adjustment of weight vectors proposed in this paper judges the distribution of the current PF, generates more concentrated weight vectors in sparse regions, produces sparser weight vectors in crowded regions, which improves the distribution of the obtained PF.

According to the improvements of MOEA/D in Section 1, due to core evolutionary strategy and MOEA/D are merged together rather than simply superimposed, adjusting either of them only cannot achieve good results in convergence and distribution of PF, MOEAD-EWA-ISSA combines MOEA/D and SSA, improves the multi-objective framework and the core evolutionary strategy according to the different requirements of MOEA/D in different evolutionary stages and the characters of SSA. In addition, as for the weight vectors adjustment, if adjust the generation of weight vector at the beginning of optimization, the difficulty of optimizing different problems in different search space will be ignored, if adjust part of the weight vectors in the evolutionary process, the PF will be too centralized or too dispersed easily, if all the weight vectors are adjusted, the real distribution of PF is difficult to analyze. MOEAD-EWA-ISSA comprehensively analyzes the evolution and the actual distribution of PF based on the PF itself without other supplementary methods, adjusts all the weight vectors adaptively, which improves the distribution of the obtained PF.

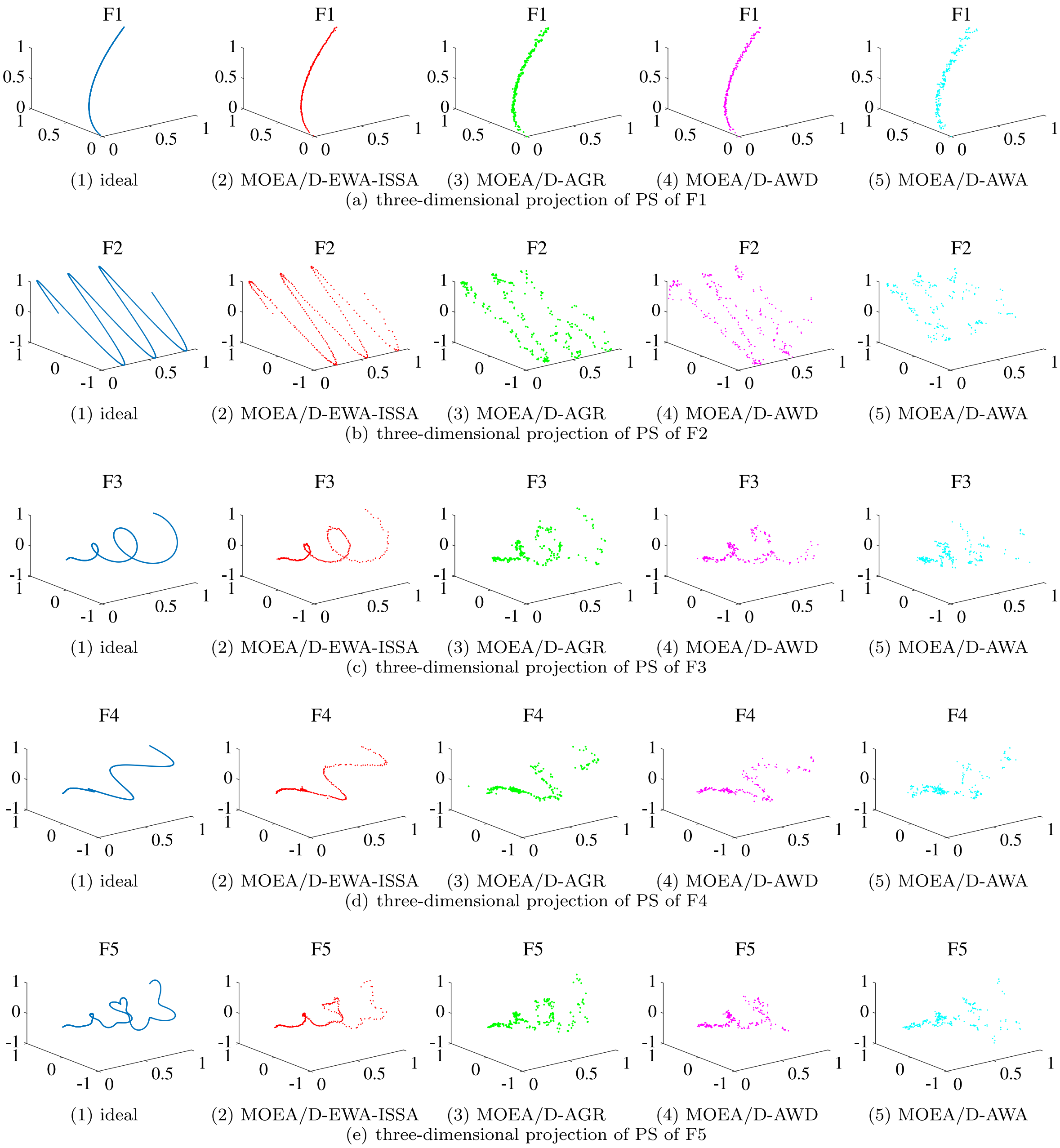


Fig. 11. Three-dimensional projection of LZ09.

4. Analysis of experimental results

In this section, a series of experiments are carried out to verify the performance of MOEA/D-EWA-ISSA. All the experiments work on CPU: Intel I Core I i5-7200M, 4G RAM, 2.70 GHZ, Windows 10 and Matlab R2016a. The test functions include: classical test functions ZDT (ZDT1–ZDT4 and ZDT6), DTLZ (DTLZ1 and DTLZ2), test functions LZ09 (F1–F9) with complex PS. The Inverted Generational Distance (IGD) is used to measure the performance. PF has better convergence and distribution when the value of IGD is smaller. IGD is defined as formula (18).

$$IGD(P^*, P) = \frac{\sum_{i=1}^{|P^*|} d(p_i^*, P)}{|P^*|} \tag{18}$$

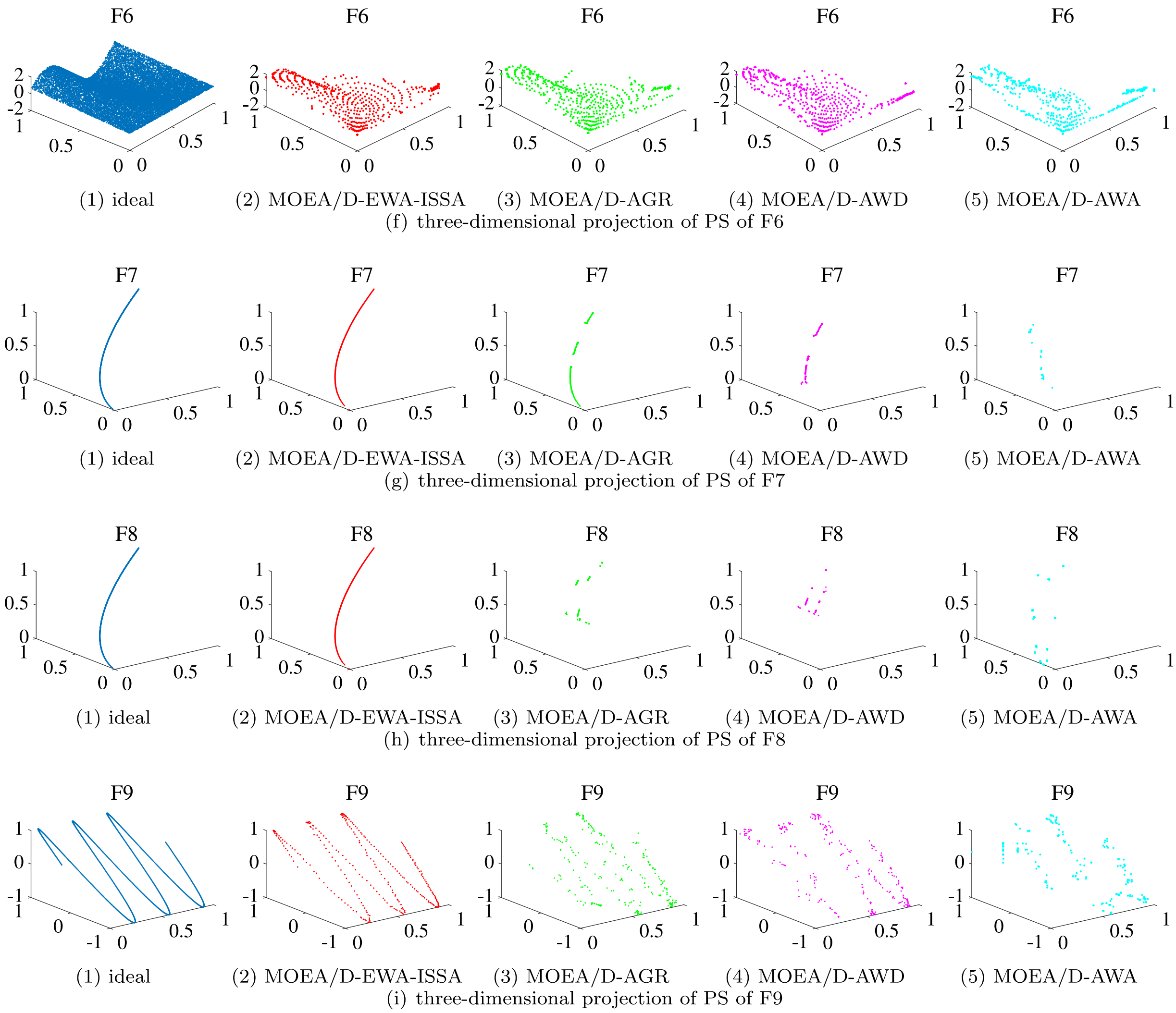


Fig. 11. (*continued*).

P^* is the set of uniform sampling points on ideal PF, P is the PF obtained by algorithms, $d(p_i^*, p)$ is the minimum Euclidean distance between p_i^* and P, $|P^*|$ is the size of P^*.

The experiments include the following two parts: one is to verify the effectiveness of the core evolutionary strategy and the improved MOEA/D framework; the other is to verify the performance of the complete MOEA/D-EWA-ISSA. The details are as follows.

4.1. Verification of the improved strategies

MOEA/D-EWA-ISSA not only improves the core evolutionary strategy, but also modifies the MOEA/D framework. Therefore, the following two experiments are carried out: (a) Verification of the improved core evolutionary strategy: combine the original MOEA/D with ISSA, GA, DE and GSA and compare them on the test functions above. (b) Verification of the improved MOEA/D framework: take ISSA as the core evolutionary strategy and combine it with Adaptive Weight Vector Design (MOEA/D-AWD) [5], Adaptive Weight Adjustment (MOEA/D-AWA) [6] and Adaptive Replacement Strategies for MOEA/D (MOEA/D-AGR) [10] frameworks which have better optimization effect on solving MOP, compare them on the test functions above.

In order to ensure the fairness of comparison, for ZDT, the population size N of each algorithm is 100, the maximum evaluation times are 25 000; for DTLZ, F1–F5 and F7–F9, the population size N of each algorithm is 300, the maximum evaluation times are 300 000; for F6, the population size N of each algorithm is 595, the maximum evaluation times are 595 000. To avoid the occasionality of a single run, all experiments are executed 30 times independently.

Table 1
Experimental results of ZDT and DTLZ.

Function	GA	DE	GSA	ISSA
ZDT1	5.14024 e−03/3.8974 e−04	6.6894 e−02/1.1478 e−02	1.1658 e−01/1.3672 e−01	**5.0846 e−03**/**2.1230 e−04**
ZDT2	**3.8811 e−03**/**4.9555 e−05**	2.1335 e+00/7.2055 e−02	2.0917 e−01/3.3022 e−01	4.7685 e−03/1.3518 e−04
ZDT3	1.2695 e−02/**1.7943 e−04**	1.5902 e−01/5.7023 e−02	2.0903 e−01/1.4680 e−01	**1.1649 e−02**/4.2834 e−04
ZDT4	**6.0204 e−02**/**7.3141 e−02**	4.2874 e+01/1.3075 e+01	1.4834 e−01/6.4926 e−02	6.4911 e−01/2.9882 e−01
ZDT6	1.8944 e−03/5.6449 e−05	5.3167 e+00/2.0215 e+00	1.8844 e−03/9.3365 e−05	**1.8777 e−03**/**4.7222 e−05**
DTLZ1	1.5801 e−02/**1.8973 e−04**	1.1392 e+01/5.2328 e+00	1.8322 e+01/5.8562 e+00	**1.5700 e−02**/1.9752 e−04
DTLZ2	4.9467 e−02/3.6653 e−03	**4.4348 e−02**/9.1176 e−03	4.9778 e−02/1.0541 e−02	4.7745 e−02/**1.4616 e−03**

Table 2
The result of Friedman test.

Algorithm	Rank							Total rank (R_j)	Average rank (R_i)	Sort
	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	DTLZ1	DTLZ2			
GA	2	1	2	1	3	2	3	14	2	3
DE	3	4	3	4	4	3	1	22	3.1429	2
GSA	4	3	4	2	2	4	4	23	3.2857	1
ISSA	1	2	1	3	1	1	2	11	1.5714	4

Table 3
The result of Holm test.

i	Algorithm	$z = (R_i - R_4)/SE = (R_i - R_4)\Big/\sqrt{\frac{k(k+1)}{6n}} = (R_i - R_4)\Big/\sqrt{\frac{4\times(4+1)}{6\times7}} = (R_i - R_4)/0.6901$	P_i	$\alpha(k-i)$
1	GSA	$(3.2857 - 1.5714)/0.6901 = 2.4841$	0.0130	0.0167
2	DE	$(3.1429 - 1.5714)/0.6901 = 2.2772$	0.0229	0.0250
3	GA	$(2 - 1.5714)/0.6901 = 0.6211$	0.5319	0.0500

4.1.1. Comparison of the core evolutionary strategies

The parameters of each algorithm are set as follows:
GA: distribution index $mu = 2$, crossover probability $p_c = 1$, mutation probability$p_m = 1$;
DE: crossover probability $CR = 0.6$, $F = 0.5$;
GSA: gravitational constant $G_0 = 100$, $\alpha = 20$;
ISSA: $N_{fs} = 2$;
MOEA/D: neighborhood size $T = 20$.

Table 1 shows the IGD values of the original MOEA/D framework combined with GA, DE, GSA and ISSA on ZDT and DTLZ. The number before '/' is the average value of IGD obtained by 30 experiments, the number after '/' is the deviation of IGD obtained by 30 experiments.

Table 1 shows that ZDT2 and ZDT4 have the smallest IGD when the core evolution strategy is GA; DTLZ2 has the smallest IGD when the core evolution strategy is DE; ZDT1, ZDT3, ZDT6 and DTLZ1 have the smallest IGD when the core evolution strategy is ISSA.

In order to compare the differences of each algorithm, Friedman test and Holm test [20] are taken on the data in Table 1. The specific process is shown below:

Experimental results of k classifiers on n datasets, Friedman statistics is defined as $\chi_r^3 = \frac{12}{nk(k+1)}\sum_{j=1}^{k} R_j^2 - 3n(k+1)$, in terms of Table 2, k refers to the total number of algorithms and n equals to the total number of test functions, that is, k is equal to 4 and n is equal to 7, $\chi_r^3 = \frac{12}{nk(k+1)}\sum_{j=1}^{k} R_j^2 - 3n(k+1) = \frac{12}{7\times4\times(4+1)}(14^2 + 22^2 + 23^2 + 11^2) - 3\times7\times(4+1) = 9$, besides, $\alpha = 0.05$, $d_f = 4 - 1 = 3$ at the 5% significant level and $\chi_{0.05}^3 = 7.81 < 9$ according to Chi-square distribution table. Therefore, the four algorithms are considered to have significant differences at the 5% significance level.

To further compare the performance of the four algorithms, assuming that the convergence and distribution are better than the other three algorithms when the core evolutionary strategy is ISSA. Holm test is carried out and the results are shown in Table 3.

It can be seen from Table 3 that $P_1 < \alpha(k-1)$, $P_2 < \alpha(k-2)$, $P_3 > \alpha(k-3)$, the original hypothesis is rejected at the 5% significance level. Therefore, compared with MOEA/D combined with DE and MOEA/D combined with GSA, MOEA/D combined with ISSA has significantly better performance. Besides, MOEA/D combined with ISSA has smaller average rank though it does not outperform the MOEA/D combined with GA significantly. In summary, for classical test functions ZDT and DTLZ, compared with common core evolutionary strategies including GA, DE and GSA, MOEA/D combined with ISSA has obvious advantages in convergence and distribution.

Table 4
Experimental results of LZ09.

Function	GA	DE	GSA	ISSA
F1	1.3397 e−03/2.6597 e−05	1.2968 e−03/3.0916 e−05	9.5940 e−03/1.5890 e−03	**1.2928 e−03**/**2.4768 e−05**
F2	9.0587 e−03/5.2311 e−03	9.8064 e−02/9.1583 e−03	6.3389 e−02/4.9342 e−03	**1.7321 e−03**/**1.0473 e−04**
F3	4.7984 e−03/1.0618 e−03	1.6436 e−02/4.7280 e−03	6.4051 e−02/7.2659 e−03	**1.9364 e−03**/**3.5233 e−04**
F4	4.4594 e−03/3.1418 e−03	1.1879 e−02/7.5777 e−03	5.7535 e−02/6.3742 e−03	**2.1591e−03**/**5.7890 e−04**
F5	4.0986 e−03/1.1203 e−03	1.2057 e−02/4.9919 e−03	5.8823 e−02/8.0349 e−03	**2.6696 e−03**/**6.0445 e−04**
F6	**2.9454 e−02**/**1.2787 e−03**	1.4550 e−01/8.5439 e−02	1.0216 e−01/1.7924 e−02	3.0184 e−02/2.4211 e−03
F7	1.6188 e−01/6.6462 e−02	3.0659 e−01/3.9350 e−02	6.0441 e−02/**1.9489 e−02**	**3.4091 e−02**/3.1700 e−02
F8	1.1870 e−01/3.0763 e−02	2.2475 e−01/4.1581 e−02	9.1510 e−02/**1.5043 e−02**	**7.4214 e−02**/5.5884 e−02
F9	1.2942 e−02/5.7202 e−03	9.6188 e−02/1.3702 e−02	7.1939 e−02/8.7943 e−03	**2.7262 e−03**/**2.6182 e−03**

Test each algorithm on LZ09 and the results of IGD are shown in Table 4, the corresponding boxplots are shown in Fig. 5.

Table 4 shows that MOEA/D with ISSA as the core evolutionary strategy has the minimum IGD on all test functions except F6. Besides, compared with the other three core evolutionary strategies, ISSA has the best stability on F2, F4 and F5; the stability on F1 is almost equal to GA and DE; although IGD of F3 and F9 have abnormal values, IGD obtained by ISSA is smaller than that obtained by the other three core evolutionary strategies, so the stability is still good; for F7 and F8, ISSA obtains the minimum IGD, but the stability is not good enough. In summary, the convergence and distribution of PF obtained by MOEA/D combined with ISSA have great advantages when solving functions with complex PS, the stability of optimization results obtained by MOEA/D combined with ISSA also outperforms the other three core evolutionary strategies.

According to the analysis of 4.1.1, when ISSA is the core evolutionary strategy of MOEA/D, the advantages on ZDT and DTLZ are not as significant as the advantages on LZ09, therefore, MOEA/D combined with ISSA will has more significant advantages if test functions are more complex.

4.1.2. Comparison of the improved MOEA/D framework

This paper proposes a multi-objective framework MOEA/D-EWA, in order to verify the effectiveness of MOEA/D-EWA, take ISSA as the core evolutionary strategy and combine it with MOEA/D-EWA, MOEA/D-AWD, MOEA/D-AWA and MOEA/D-AGR to compare on the test functions mentioned above. In terms of MOEA/D-EWA, the parameter k is determined by experiments as follows.

Set $k = 5$, 10, 15 and 20 respectively, Fig. 6 are the boxplots of IGD obtained by MOEA/D-EWA-ISSA on several test functions.

According to Fig. 6, in terms of classical two-objective test function with less difficulty to optimize, the optimizing effect is less affected by frequency. The classical three-objective test function with greater difficulty to be optimized showed in Fig. 6 has better optimization results with the adjusting frequency become higher, but the higher adjusting frequency will cause more computation. As for the test function with complicated PS, the higher frequency of adjusting the weight vector (the smaller value of k) will adjust the weight vectors when the obtained PF is far from the ideal PF, which may lead to worse distribution of the PF; the lower frequency of adjusting the weight vector (the larger value of k) will not adjust the weight vectors in time and the distribution will not be good enough as well. In order to take account of the computational complexity and the effect of solving different types of problems, the value of k is set to 10 in this paper.

Table 5 shows the IGD values of the four MOEA/D frameworks combined with ISSA on ZDT and DTLZ. The number before '/' is the average value of IGD obtained by 30 experiments, the number after '/' is the deviation of IGD obtained by 30 experiments.

The parameters of each algorithm are set as follows:

MOEA/D-AWA: $\delta = 0.9$, $nus = 0.05 \times N$ ($nus = 30$ for F6), $rate\_evol = 0.8$, wag: ZDT1-4, ZDT6, DTLZ2, F1–F9: $wag = 100$, DTLZ1: $wag = 125$;

MOEA/D-AWD: Adjust the weight vectors every 100 generations;

MOEA/D-AGR: $T_{\max} = 0.4 \times N$, $\gamma = 0.25$;

MOEA/D-EWA: $N_{fs} = 2$, $T_0 = 20$, $k = 10$.

Table 5 shows that AWA has the smallest IGD on ZDT2 and DTLZ2, AWD has the smallest IGD on ZDT6 and ISSA has the smallest IGD on ZDT1, ZDT3, ZDT4 and DTLZ1.

In order to compare the differences of each MOEA/D framework, Friedman test is taken on the data in Table 5 and the results are as follows: $R_{AWA} = 2.1429$, $R_{AWD} = 3.4286$, $R_{AGR} = 3$, $R_{EWA} = 1.4286$, $\chi_r^3 = 10.03$. The four MOEA/D frameworks are considered to have significant differences at the 5% significance level because of $\chi_{0.05}^3 = 7.81 < \chi_r^3$.

To further compare the performance of the four MOEA/D frameworks, assuming that the convergence and distribution of EWA are better than AWA, AWD and AGR. Holm test is carried out and the results are shown in Table 6.

It can be seen from Table 6 that $P_1 < \alpha(k-1)$, $P_2 < \alpha(k-2)$, $P_3 > \alpha(k-3)$, the original hypothesis is rejected at the 5% significance level. Therefore, EWA has much better performance compared with AWD and AGR. Besides, EWA has smaller average rank though it does not outperform AWA. In summary, for classical test functions ZDT and DTLZ, compared with AWD, AGR and AWA, EWA has obvious advantages in convergence and distribution.

Table 5
Experimental results of ZDT and DTLZ.

Function	AWA	AWD	AGR	EWA
ZDT1	6.4158 e−03/6.2658 e−04	5.1519 e−03/1.9254 e−04	5.1289 e−03/**1.7425 e−04**	**4.7367 e−03**/2.1289 e−04
ZDT2	**4.4668 e−03**/6.9027 e−04	4.8057 e−03/2.5890 e−04	4.7820e -03/**1.6177 e−04**	4.5250 e−03/2.2656 e−04
ZDT3	1.3352 e−02/3.4625 e−04	1.5076 e−02/5.3616 e−04	1.3498 e−02/**1.9045 e−04**	**1.2586 e−02**/3.6636 e−04
ZDT4	2.6502 e−01/1.1741 e−01	6.1545 e−01/1.8562 e−01	5.7485 e−01/2.2971 e−01	**4.4637 e−03/3.3040 e−04**
ZDT6	1.8894 e−03/4.4985 e−05	**1.8650 e−03/4.1670 e−06**	1.8966 e−03/3.2280 e−05	1.8767 e−03/1.2955 e−04
DTLZ1	1.5626 e−02/1.9867 e−04	1.5783 e−02/2.7547 e−04	1.5665 e−02/**1.7515 e−04**	**1.5618 e−02**/2.5018 e−04
DTLZ2	**4.4288 e−02**/8.7163 e−03	4.8380 e−02/**9.5144 e−04**	4.8129 e−02/2.1968 e−03	4.8091 e−02/1.7316 e−03

Table 6
The result of Holm test.

i	Algorithm	$z = (R_i - R_4)/SE = (R_i - R_4)\Big/\sqrt{\frac{k(k+1)}{6n}} = (R_i - R_4)\Big/\sqrt{\frac{4\times(4+1)}{6\times 7}} = (R_i - R_4)/0.6901$	P_i	$\alpha(k-i)$
1	AWD	$(3.4286 - 1.4286)/0.6901 = 2.8981$	0.0038	0.0167
2	AGR	$(3.0000 - 1.4286)/0.6901 = 2.2771$	0.0229	0.0250
3	AWA	$(2.1429 - 1.4286)/0.6901 = 1.0351$	0.3007	0.0500

Table 7
Experimental results of LZ09.

Function	AWA	AWD	AGR	EWA
F1	1.4744 e−03/4.0479 e−05	1.3447 e−03/3.5254 e−05	**1.3216 e−03**/4.0987 e−05	1.3314 e−03/**2.0387 e−05**
F2	1.7711 e−02/4.1740 e−03	1.6592 e−03/1.4219 e−04	1.6668 e−03/4.6588 e−05	**1.4251 e−03/4.2749 e−05**
F3	6.1097 e−03/8.7981 e−04	1.6325 e−03/8.3256 e−05	1.7118 e−03/1.8725 e−04	**1.4267 e−03/3.7352 e−05**
F4	7.1487 e−03/1.3818 e−03	1.9341 e−03/3.3058 e−04	1.5689 e−03/**5.0226 e−05**	**1.4325 e−03**/6.5070 e−05
F5	6.6495 e−03/4.5755 e−04	2.4114 e−03/5.4062 e−04	2.1592 e−03/**1.2607 e−04**	**1.9390 e−03**/2.6127 e−04
F6	5.2280 e−02/1.3050 e−02	4.7729 e−02/3.4225 e−03	4.8207 e−02/6.0206 e−03	**2.7961 e−02/2.6076 e−03**
F7	1.0696 e−01/2.9502 e−02	1.4354 e−02/2.2726 e−02	1.0339 e−02/1.7420 e−02	**1.4361 e−03/1.2995 e−04**
F8	1.0676 e−01/3.4902 e−02	2.8703 e−02/1.4333 e−02	3.0839 e−02/1.1648 e−02	**2.5052 e−03/1.5841 e−03**
F9	1.4892 e−02/6.6217 e−03	6.0895 e−03/6.7406 e−03	3.1533 e−03/2.4357 e−03	**1.8413e−03/9.7504 e−04**

Similar to the experiments above, each MOEA/D framework is tested on LZ09. The results of IGD are shown in Table 7 and the corresponding boxplots are shown in Fig. 7.

Table 7 shows that EWA achieves the minimum IGD on all functions except for F1. In addition, it can be seen from Fig. 7 that compared with AWA, AWD and AGR, the stability of EWA has obvious advantages on F3, F6, F7 and F8; as for F2, the stability of EWA is almost equal to AGR, as for F5, the stability of EWA is next to AGR on F5; IGD of F4 and F9 have abnormal values, however, IGD obtained by EWA is smaller than that obtained by the other three MOEA/D frameworks, the stability of EWA on F4 and F9 is still good enough. In summary, compared with other three improved MOEA/D frameworks, the convergence and distribution of PF obtained by EWA have significant advantages.

According to the analysis of 4.1.1, when take EWA as the framework of multi-objective optimization algorithm, the advantages on ZDT and DTLZ are not as significant as the advantages on LZ09, therefore, MOEA/D-EWA will has more significant advantages if test functions are more complex.

4.2. Overall performance analysis of MOEA/D-EWA-ISSA

Compare MOEA/D-EWA-ISSA with original MOEA/D-AWA, MOEA/D-AWD and MOEA/D-AGR on all the test functions mentioned above. Table 8 shows the IGD values of the four algorithms on ZDT and DTLZ. The number before '/' is the average value of IGD obtained by 30 experiments, the number after '/' is the deviation of IGD obtained by 30 experiments. The core evolutionary strategy of MOEA/D-EWA-ISSA is ISSA, the core evolutionary strategies of other three algorithms are all GA. The parameter settings are the same as 4.1.

Table 8 shows that MOEA/D-AWA has the smallest IGD on DTLZ2, MOEA/D-AGR has the smallest IGD on ZDT2 and ZDT6, MOEA/D-EWA-ISSA has the smallest IGD on ZDT1, ZDT3, ZDT4 and DTLZ1.

In order to compare the differences of the four algorithms, Friedman test is taken on the data in Table 8 and the results are as follows: $R_{MOEA/D-AWA} = 3.1429$, $R_{MOEA/D-AWD} = 3.1429$, $R_{MOEA/D-AGR} = 2.2857$, $R_{MOEA/D-EWA-ISSA} = 1.4286$, $\chi_r^3 = 8.49$. The four MOEA/D frameworks are considered to have significant differences at the 5% significance level because of $\chi_{0.05}^3 = 7.81 < \chi_r^3$.

Table 8
Experimental results of ZDT and DTLZ.

Function	MOEA/D-AWA	MOEA/D-AWD	MOEA/D-AGR	MOEA/D-EWA-ISSA
ZDT1	9.7059 e−03/1.4989 e−03	5.3593 e−03/2.9983e−04	5.2675 e−03/3.0472e−04	**4.7367 e−03/2.1289 e−04**
ZDT2	4.5250 e−03/2.2656 e−04	4.3903 e−03/4.5991e−04	**3.9156 e−03/1.3756e−04**	4.25239 e−03/2.2172e−04
ZDT3	1.9112 e−02/3.0031 e−03	2.4914 e−02/2.8717 e−02	1.3750 e−02/**2.2298e−04**	**1.2586 e−02**/3.6636 e−04
ZDT4	2.7641 e−01/1.6184 e−01	5.3157 e−02/5.9232 e−02	5.2772 e−02/5.7733 e−02	**4.4637 e−03/3.3040 e−04**
ZDT6	1.9686 e−03/1.1728e−04	1.8856 e−03/**2.5542e−05**	**1.8751 e−03/** 2.5810e−05	1.8767 e−03/1.2955 e−04
DTLZ1	1.5690 e−02/3.1823e−04	1.5706 e−02/**1.2079e−04**	1.5828 e−02/1.6811e−04	**1.5618 e−02**/2.5018 e−04
DTLZ2	**3.9461 e−02**/8.7573 e−03	4.9037 e−02/2.3326 e−03	4.9237 e−02/2.5291 e−03	4.8091 e−02/**1.7316 e−03**

Table 9
The result of Holm test.

i	Algorithm	$z=(R_i-R_4)/SE$ $=(R_i-R_4)\Big/\sqrt{\frac{k(k+1)}{6n}}$ $=(R_i-R_4)\Big/\sqrt{\frac{4\times(4+1)}{6\times 7}}$ $=(R_i-R_4)/0.6901$	P_i	$\alpha(k-i)$
1	AWD	(3.1429 − 1.4286)/0.6901 = 2.4841	0.0130	0.0167
2	AGR	(3.1429 − 1.4286)/0.6901 = 2.4841	0.0130	0.0250
3	AWA	(2.2857 − 1.4286)/0.6901 = 1.2420	0.2131	0.0500

Table 10
Experimental results of LZ09.

Function	MOEA/D-AWA	MOEA/D-AWD	MOEA/D-AGR	MOEA/D-EWA-ISSA
F1	1.6359 e−03/5.7430e−05	1.3638 e−03/3.8634e−05	1.3466 e−03/5.7081e−05	**1.3314 e−03/2.0387 e−05**
F2	2.1742 e−02/4.1401 e−03	8.9605 e−03/2.6912 e−03	9.2633 e−03/3.2620 e−03	**1.4251 e−03/4.2749 e−05**
F3	1.1830 e−02/1.1665 e−03	4.9120 e−03/3.4897 e−03	4.9471 e−03/1.6222 e−03	**1.4267 e−03/3.7352 e−05**
F4	1.0921 e−02/8.3654e−04	7.6681 e−03/3.7327 e−03	3.3136 e−03/6.1042e−04	**1.4325 e−03/6.5070 e−05**
F5	9.7929 e−03/5.9219e−04	4.5614 e−03/9.6165e−04	5.4853 e−03/3.5495 e−03	**1.9390 e−03/2.6127 e−04**
F6	4.3171 e−02/5.7212 e−03	2.9021 e−02/1.3069 e−03	2.9795 e−02/**1.0763 e−03**	**2.7961 e−02**/2.6076 e−03
F7	1.9809 e−01/4.6417 e−02	1.6375 e−01/9.3614 e−02	1.2176 e−01/8.5386 e−02	**1.4361 e−03/1.2995 e−04**
F8	9.1646 e−02/1.7145 e−02	1.0814 e−01/3.1472 e−02	1.1101 e−01/3.5994 e−02	**2.5052 e−03/1.5841 e−03**
F9	2.6435 e−02/6.4207 e−03	1.6365 e−02/7.5717 e−03	1.7690 e−02/6.1297 e−03	**1.8413e−03/9.7504 e−04**

To further compare the performance of the four algorithms, assuming that the convergence and distribution of MOEA/D-EWA-ISSA are better than MOEA/D-AWA, MOEA/D-AWD and MOEA/D- AGR. Holm test is carried out and the results are shown in Table 9.

It can be seen from Table 9 that $P_1 < \alpha(k-1)$, $P_2 < \alpha(k-2)$, $P_3 > \alpha(k-3)$, the original hypothesis is rejected at the 5% significance level. Therefore, MOEA/D-EWA-ISSA has much better performance compared with MOEA/D-AWD and MOEA/D-AGR. Besides, MOEA/D-EWA-ISSA has smaller average rank though it does not outperform MOEA/D-AWA. In summary, for classical test functions ZDT and DTLZ, compared with MOEA/D-AWD, MOEA/D-AGR and MOEA/D- AWA, MOEA/D-EWA-ISSA has obvious advantages in convergence and distribution.

Similar to the experiments above, each algorithm is tested on LZ09. The results of IGD are shown in Table 10 and the corresponding boxplots are shown in Fig. 8.

Table 10 shows that MOEA/D-EWA-ISSA achieves the minimum IGD on all functions. In addition, it can be seen from Fig. 8 that compared with MOEA/D-AWA, MOEA/D-AWD and MOEA/D-AGR, the stability of EWA has obvious advantages on F1, F2, F3, F5, F7 and F8; as for F6, MOEA/D-EWA-ISSA has the minimum IGD, however, the stability is only better than AWA; though IGD of F4 and F9 have abnormal values, IGD obtained by MOEA/D-EWA-ISSA is much smaller than that obtained by the other three algorithms, therefore, the stability of MOEA/D-EWA-ISSA is still good enough. In summary, compared with other three algorithms, the convergence and distribution of PF obtained by MOEA/D-EWA-ISSA have significant advantages.

The experiments of 4.2 show that the advantages of MOEA/D-EWA-ISSA on ZDT and DTLZ are not as significant as the advantages on LZ09, therefore, MOEA/D-EWA-ISSA will has more significant advantages if test functions are more complex. Fig. 9 are the PF obtained by four algorithms on ZDT and DTLZ, Fig. 10 are the PF obtained by four algorithms on LZ09. Fig. 11 are the obtained three-dimensional projection of PS on LZ09.

5. Conclusion

This paper proposes a Multi-objective Improved Squirrel Search Algorithm based on Decomposition with External Population and Adaptive Weight Vectors Adjustment (MOEA/D-EWA-ISSA), MOEA/D-EWA-ISSA takes MOEA/D as the framework and adjusts every weight vector by the actual evolutionary direction of PF and the representative neighbor weight vectors, which improves the distribution of the obtained PF. MOEA/D-EWA-ISSA takes SSA as the core evolutionary strategy and introduces the jumping search method and the progressive search method for it, which improves the ability of core evolutionary strategy on solving each subproblem, therefore, the convergence of the obtained PF is improved as well. Besides, in order to fully retain the evolutionary information and maintain the population diversity, MOEA/D-EWA-ISSA establishes an external population for every individual, the evolution involved in external individuals improves the convergence and distribution of the obtained PF. The experimental results show that the improved core evolution strategy ISSA, the improved multi-objective framework MOEA/D-EWA and the integrated MOEA/D-EWA-ISSA can significantly improve the convergence and distribution of the obtained PF in solving multi-objective problems. The proposed algorithm will have more significant advantages if the multi-objective problems are more complex. As for the future work, there are many multi-objective optimization problems occurs in engineering application fields, taking the proposed algorithm into practice is the future research direction, in addition, the difficulty of optimization increases with more objective functions, apply SSA and MOEA/D in solving many-objective optimization problems is also the future work.

Acknowledgments

The authors disclosed receipt of the following financial support for the research, authorship of this article: This work was supported in part by the National Natural Science Foundation of China under grants NO. 61501107 and NO. 61501106.

References

[1] M. Jain, A. Rani, N. Pachauri, V. Singh, A.P. Mittal, Design of fractional order 2-DOF PI controller for real-time control of heat flow experiment, J. Eng. Sci. Technol. 22 (1) (2019) 215–228.
[2] Q. Jiang, L. Wang, Y. Lin, X. Hei, G. Yu, X. Lu, An efficient multi-objective artificial raindrop algorithm and its application to dynamic optimization problems in chemical processes, Appl. Soft Comput. 58 (2017) 354–377.
[3] J. Zhang, L. Xing, A survey of multiobjective evolutionary algorithms, in: IEEE International Conference on Computational Science & Engineering, 2017, pp. 93–100.
[4] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2008) 712–731, 2008.
[5] X. Guo, X. Wang, Z. Wei, MOEA/D with adaptive weight vector design, in: 2015 11th International Conference on Computational Intelligence and Security, 2015, pp. 291–294.
[6] Y. Qi, X. Ma, F. Liu, MOEA/D with adaptive weight adjustment, Evol. Comput. 22 (2) (2014) 231–264.
[7] Y.Y. Tan, Y.C. Jiao, H. Li, X.K. Wang, A modification to MOEA/D-DE for multiobjective optimization problems with complicated Pareto sets, Inform. Sci. 213 (2012) 14–28.
[8] Hiroyuki Sato, Analysis of inverted PBI and comparison with other scalarizing functions in decomposition based MOEAs, J. Heuristics 21 (6) (2015) 819–849.
[9] S. Jiang, S. Yang, An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts, IEEE Trans. Cybern. 46 (2) (2016) 421–437.
[10] Z. Wang, Q. Zhang, A. Zhou, Adaptive replacement strategies for MOEA/D, IEEE Trans. Cybern. 46 (2) (2017) 474–486.
[11] S.M. Venske, R.A. Gonçalves, M.R. Delgado, ADEMO/D: Multiobjective optimization by an adaptive differential evolution algorithm, Neurocomputing 127 (2014) 65–77.
[12] X.J. Bi, P.F. Diao, Y.J. Wang, Multi-objective gravity search algorithm based on decomposition, J. Harbin Inst. Technol. 47 (11) (2015) 69–75.
[13] M. Jain, S. Maurya, A. Rani, V. Singh, Owl search algorithm: a novel nature-inspired heuristic paradigm for global optimization, J Intell. Fuzzy Syst. 34 (3) (2018) 1573–1582.
[14] A.F. Nematollahi, A. Rahiminejad, B. Vahidi, A novel physical based meta-heuristic optimization method known as Lightning Attachment Procedure Optimization, Appl. Soft Comput. 59 (2017) 596–621.
[15] F. Fausto, E. Cuevas, A. Valdivia, A global optimization algorithm inspired in the behavior of selfish herds, Biosystems 160 (2017) 39–55.
[16] M. Jain, A. Rani, V. Singh, An improved Crow Search Algorithm for high-dimensional problems, J. Intell. Fuzzy Syst. 33 (6) (2017) 3597–3614.
[17] K. Vivek, M. Deepak, Chetna, M. Jain, A. Rani, V.A. Singh, Development of multi-verse optimizer (mvo) for labview, in: 17th East European Conference on Advances in Databases and Information Systems and Associated Satellite Events, 2018, pp. 731–739.
[18] G. Dhiman, V. Kumar, Multi-objective spotted hyena optimizer: A multi-objective optimization algorithm for engineering problems, Knowl.-Based Syst. 150 (2018) 175–197.
[19] M. Jain, V.A. Singh, Rani, A novel nature-inspired algorithm for optimization: Squirrel search algorithm, Swarm Evol. Comput. 44 (2018) 148–175.
[20] Janez Demišar, D. Schuurmans, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (1) (2006) 1–30.