An Improved SPEA2 Algorithm with Adaptive Selection of Evolutionary Operators Scheme for Multiobjective Optimization Problems

1. Introduction

The multiobjective optimization problems (MOPs) [1] usually have more than two objectives. However, the evolutionary multiobjective optimization (EMO) researchers are only interested in the problems whose objectives are in conflict. For example, to produce a product, it not only requires short production time but also needs high quality. Those two objectives are in conflict.

Because no single solution can simultaneously optimize all the objectives on the condition that these objectives are in conflict, therefore the purpose of the MOP is to achieve a group of Pareto optimal set and make it that the solutions distribution on the Pareto front has best possible approximation and uniformity. The traditional optimization algorithms can transform the multiobjective optimization problem into single objective problem with positive coefficient. The common weakness of the traditional algorithm is to produce single Pareto optima in a single run. However, the evolutionary algorithm is a population-based random search approach, which can generate a group of Pareto optimal solutions set in a single run and is very suitable for solving the MOPs. Since Schaffer [2] used multiobjective evolutionary algorithms (MOEAs) to solve MOPs, a variety of evolutionary algorithms have been developed. The characteristic of the first generation of MOEAs was using Pareto ranking to fitness assignment and making use of the niche or fitness sharing to maintain diversity. The representative algorithms contain nondominated sorting genetic algorithm (NSGA) [3], the multiobjective genetic algorithm (MOGA) [4], and the Niched Pareto genetic algorithm (NPGA) [5]. The feature of the second generation of MOEAs was whether the elitism is used or not. The classical algorithms are the Pareto archived evolution strategy (PAES) [6], the Pareto envelope based selection algorithm (PESA) [7] and its revised version PESA-II [8], the strength Pareto evolutionary algorithm (SPEA) [9] and its improved version SPEA2 [10], and the improved version of NSGA (NSGA-II) [11]. In recent years, some new frameworks of MOEAs have been proposed. The MOEA/D [12, 13] which combines traditional mathematical programming method with multiobjective evolutionary algorithm is one of the new frameworks. It shows high performance when
solving MOPs with complicated PS shapes [14]. Meanwhile, many other nature-inspired metaheuristics including Ant Colony Optimization [15, 16], Particle Swarm Optimization [17, 18], Immune Algorithm [19, 20], and Estimation of Distribution Algorithm [21, 22] have been successfully applied to handle MOPs. Moreover, MOEAs for complicated MOPs have also been extensively investigated, such as MOEA for constraint MOPs [23], dynamic MOPs [24], and many objective optimization problems [25].

SPEA2 is one of the second generation MOEAs. Bleuler et al. [26] considered the program size as a second, independent objective besides the program functionality and combined with SPEA2. Over the past decade, SPEA2 has been successfully combined with other optimization strategies to form improved SPEA2 algorithms. Kim et al. [27] added a more efficient crossover mechanism and an archive mechanism to maintain diversity of the solutions in the objective and variable spaces. Zheng et al. [28] combined SPEA2 with the parallel genetic algorithm (PGA) to obtain the final solution. Wu et al. [29] proposed a modified method of calculating fitness value based on SPEA2. A more reasonable strategy of elitism population selection is used to improve the distribution performance of the multiobjective optimization. Li et al. [30] combined several specific local search strategies with SPEA2 to enhance the algorithm’s exploiting capability. Belgasmi et al. [31] improved the performance of SPEA2 by adequately applying a multiobjective quasigradient local search to some candidate solutions that have lower density estimation. Al-Hajri and Abido [32] adopted truncation algorithms to manage the Pareto optimal set size. Meanwhile, the best compromise solution is extracted by using fuzzy set theory in SPEA2. Sheng et al. [33] present an Improved Strength Pareto Evolutionary Algorithm 2 (ISPEA2), which introduces a penalty factor in objective function constraints and adopts an adaptive crossover as well as a mutation operator in the evolutionary process; besides, it combines simulated annealing iterative process over SPEA2. Maheta and Dabhi [34] proposed the enhancements to improve convergence performance and diversity simultaneously for SPEA2. K-nearest neighbor density estimation technique is used to maintain diversity among solutions.

Some researchers have shown that the operators are more suitable for certain types of problems but can not be available in the whole evolutionary process. For instance, simulated binary crossover (SBX) is widely used in MOEAs, but Deb [35] observed that SBX operator was unable to address problems with variable linkages. Therefore, an efficient evolutionary operator plays an important role in the evolutionary process of the optimization methods. And the operators have a great influence on the algorithms’ performance. Therefore, it is necessary to designate efficient operators for the MOEAs. At present, many efficient evolutionary operators are designed to enhance the performance of algorithms [36–39]. Pulido and Coello [40] introduced the best elected evolutionary operator to solve a given problem. A microgenetic algorithm called mGA2 is proposed, which runs several simultaneous instances of mGA2 with different evolutionary operators. Periodically, the instance with the poorest performance was replaced by the best performance ones after several generations. Thus, all the parallel instances only worked with the best performing operators after several generations. A disadvantage of this approach is that once an operator had been discarded, it could not be used again in the remaining evolutionary process. Huang et al. [41] utilized four different DE operators. Four operators were chosen in an adaptive way: the operator which contributed the most to the search was given a higher probability to create new solutions. Nebro et al. proposed two improved NSGA-II algorithms which are NSGA-IIr and NSGA-IIa [42]. NSGA-IIr is an extension of NSGA-II which employed three different evolutionary operators: the simulated binary crossover (SBX), polynomial mutation (PM), and DE. These operators are randomly selected when a new solution is to be produced. NSGA-IIa applies the same evolutionary operators as NSGA-II-r does; each operator selection probability is adjusted by considering operator success in the last iteration. And the algorithms’ performance has been greatly improved by making use of the adaptive way with evolutionary operators.

In this paper, an improved SPEA2 algorithm with adaptive selection of evolutionary operators (AOSPEA) is proposed. Multiobjective evolutionary operators including the simulated binary crossover, polynomial mutation, and differential evolution operator are employed to enhance the convergence performance and diversity of the SPEA2. Simulation results on the standard benchmarks show that the proposed algorithm outperforms SPEA2, NSGA-II, and PESA-II.

The rest of the paper is organized as follows: Section 2 provides a brief description of SPEA2 framework. In Section 3, the main loop of AOSPEA, with a particularly detailed description and analysis of the proposed adaptive the selection of evolutionary operator’s scheme. The convergence and complexity analysis of AOSPEA are also presented in detail in this section. Section 4 describes the experimental results. Section 5 makes a conclusion.

2. Multiobjective Problems (MOPs)

2.1. The Description of Multiobjective Problems (MOPs). As no single solution can optimize all the objectives at the same time on the condition that these objectives are in conflict, the solution of a MOPs is a set of decision variable vectors rather than a unique solution. Let \( x_i, x_j \in \Omega \) be two decision vectors, \( x_i \) is said to dominate \( x_j \) (\( x_i \succ x_j \)), if \( f_i(x_i) \leq f_i(x_j) \) for all \( i = 1, 2, \ldots, m \), and \( F(x_i) \neq F(x_j) \). Besides, at least one objective function should satisfy \( f_i(x_i) < f_i(x_j) \). A point \( x^{*} \in \Omega \) is called Pareto optimal solution or nondominated solution if there is no \( x \in \Omega \) such that \( F(x) \) dominates \( F(x^{*}) \). The set of all the Pareto optimal solutions is called the Pareto set, denoted by \( PS \). The set of all the Pareto optimal objective vectors, \( PF = \{ F(x^{*}) | x^{*} \in PS \} \), is called the Pareto front. It is impossible to find entire \( PS \) of continuous MOPs; the purpose is aiming at finding a finite set of Pareto optimal vectors which are uniformly scattered along the true PF and highly representative of the entire PF.

In general, the Multiobjective Problems can be illustrated mathematically as follows:

\[
\min/\max \quad y = F(x) = (f_1(x), f_2(x), \ldots, f_n(x))
\]
sub to: \( g_i(X) \leq 0, \quad i = 1, 2, \ldots, k_1 \)
\[ g_i(X) = 0, \quad i = k_1 + 1, \ldots, k_2, \]
where: \( x = (x_1, x_2, \ldots, x_j) \in X, \)
\( y = (y_1, y_2, \ldots, y_t) \in Y. \)

(1)

In the equation, \( x \) is the decision vector and \( X \) is the decision space. \( y \) is the objective vector and \( Y \) is the objective space. The most difficult reason to treat the Multiobjective Problems is that each Objective \( f_j(x) \) is related, restrained, and even conflicted with each other. And in every MOPs, there are different objectives to confine the results.

2.2. Pareto Dominance. Vector \( y = (y_1, y_2, \ldots, y_k) \) is said to Pareto dominate another vector \( z = (z_1, z_2, \ldots, z_k) \) if and only if \( \forall j \in 1, \ldots, k, \quad y_j \geq z_j \exists j \in 1, \ldots, k, \quad y > z. \)

Definition 1 (Pareto optimal point). A vector \( z \in Z \) is said to be Pareto optimal (in \( Z \)) if and only if \( \neg \exists y \in Z, \quad y \succ z \).

Definition 2 (Pareto front). The Pareto front, denoted by \( Z^* \), of a set \( Z \) is given by \( \{ z \in Z | \neg \exists y \in Z, \quad y \succ z \} \).

Definition 3 (general MOPs). A general MOPs is defined as minimizing (or maximizing) \( F(x) = (f_1(x), f_2(x), \ldots, f_k(x)) \) subject to \( g_i(x) \leq 0, \quad i = \{1, 2, \ldots, m\} \) and \( h_j(x) = 0, \quad j = \{1, 2, \ldots, p\}, \quad x \in \Omega \). A MOPs solution minimizes (or maximizes) the components of a vector \( F(x) \) where \( x \) is a \( n \)-dimensional decision variable vector \( x = (x_1, x_2, \ldots, x_n) \) from some universe \( \Omega \). It is noted that \( g_i(x) \leq 0 \) and \( h_j(x) = 0 \) represent constraints that must be fulfilled while minimizing (or maximizing) \( F(x) \) and \( \Omega \) contains all possible \( x \) that can be used to satisfy an evaluation of \( F(x) \). Thus, MOPs consist of \( k \) objectives reflected in the \( k \) objective functions, \( m + p \) constraints on the objective functions, and \( n \) decision variables. The \( k \) objective functions may be linear or nonlinear and continuous or discrete in nature. The evaluation function, \( F: \Omega \rightarrow \Delta \), is a mapping from the vector of decision variables \( (x = x_1, x_2, \ldots, x_n) \) to output vectors \( (y = a_1, a_2, \ldots, a_k) \). The vector of decision variables \( x_i \) can also be continuous or discrete.

3. The Improved SPEA2 Algorithm with Adaptive Selection of Evolutionary Operators Scheme (AOSPEA)

3.1. Brief Introduction to SPEA2. SPEA2 is an improved version of the Strength Pareto Evolutionary Algorithm (SPEA). Compared with SPEA, a fine-grained fitness assignment strategy which incorporates density information is employed in SPEA2. The fixed archive size is adopted, that is, whenever the number of nondominated individuals is less than the predefined archive size, the archive is filled up by dominated individuals. Moreover, an alternative truncation method is used to replace the clustering technique in original SPEA but does not lose boundary points, which can guarantee the preservation of boundary solution. Finally, SPEA2 only makes members of the archive participate in the mating selection process. The procedure of the SPEA2 is as follows.

**SPEA2 Algorithm**

**Input:**

Ne: population size  
N: archive size  
T: maximum number of generations.

**Output:** NDS: nondominated set.

Step 1 (initialization). Generate an initial population \( P(0) \) and create an empty archive (external archive) \( A(0) \). Set \( t = 0 \).

Step 2 (fitness assignment). Calculate fitness values of individuals in \( P(t) \) and \( A(t) \).

Step 3 (environment selection). Copy all nondominated individuals in \( P(t) \) and \( A(t) \) to \( A(t + 1) \). If size of \( A(t + 1) \) exceeds \( N \) then reduce \( A(t + 1) \) by means of the truncation operator; otherwise if size of \( A(t + 1) \) is less than \( N \) then fill \( A(t + 1) \) with dominated individuals in \( P(t) \) and \( A(t) \).

Step 4 (termination). If \( t > T \) is satisfied, then stop and output NDS. Otherwise, continue.

Step 5 (mating selection). Perform binary tournament selection with replacement on \( A(t + 1) \) in order to fill mating pool. The size of mating pool is \( N \).

Step 6 (reproduction). Apply recombination and mutation operators to the mating pool \( P(t + 1) \) to the resulting population. Set \( t = t + 1 \); go to Step 2.

3.2. The Evolutionary Operators Used in the AOSPEA. Due to the fixed evolutionary operator in the SPEA2 algorithm, it is easy to trap into local optima. The single operator can hardly meet the whole evolutionary process and different operators in the stage should be designed according to their contribution. Therefore, three different evolutionary operators including DE operator [43], the simulated binary crossover (SBX) operator [44], and PM operator [45] are employed to improve the performance of SPEA2. The description of different operators is as follows.

(1) **DE Operator.** Differential evolution (DE) has three processes including mutation, crossover, and selection. DE owns good global search ability [46] and makes use of the differences between randomly selected vectors (individuals) as the source of evolutionary dynamics. Besides, DE can control the evolutionary variation similar to the concept jump in neighborhood search by adding weighted vectors to the target vector properly. Therefore, DE operators are adopted in SPEA2. It can efficiently improve the convergence and the exploration ability of the SPEA2. The procedure of DE operators is displayed as follows.
DE Operator

**Input**: Ne; population size, population \( P(t) \).

**Output**: A new individual \( x_i \).

**Step 1.** Randomly select three different individuals \( x_{r_1,j}, x_{r_2,j}, \) and \( x_{r_3,j} \); they cannot dominate each other from \( P(t) \).

**Step 2** (mutation operator). Produce the mutation individual \( v_{i,j+1} = \{v_{i,j,1}, v_{i,j,2}, \ldots, v_{i,j,D}\}; \ D \) is the number of dimensions with (4):

\[
v_{i,j+1} = x_{r,j} + F \cdot (x_{r_3,j} - x_{r_2,j}),
\]

where \( F \) is the scale factor; \( F \in [0, 2] \).

**Step 3** (crossover operator). Produce the new individual \( x_{i,j+1} \) with (6):

\[
x_{i,j+1} = \begin{cases} 
v_{i,j+1}, & \text{if (rand} (j) \leq \text{CR}) \text{ or } j = j_{\text{rand}}, \\
2x_{i,j} - x_{r,j}, & \text{if (rand} (j) > \text{CR}) \text{ or } j \neq j_{\text{rand}}.
\end{cases}
\]

where CR is a crossover rate, rand(\( j \)) is a uniformly distributed random number between 0 and 1, and \( j_{\text{rand}} \) is randomly selected from \( \{1, 2, \ldots, D\} \).

DE operator employs the relative position of nondominated solutions to produce the evolutionary direction of the ideal Pareto front and the new search space. Figure 1 describes the theory of the DE operator, where \( x_i \) is the offspring individual. It can be seen from Figure 1(a) that DE operator employs the relative position of nondominated individual in neighborhood to produce the offspring individual close to ideal Pareto front. Figure 1(b) expresses that DE operator can obtain more broad offspring.

(2) SBX Operator. One of the three operators is provided by the SBX operator which performs local search combined with random search near the recombinations parents. Unlike other real-parameter crossover operators, SBX uses a probability distribution which is similar in principle to the probability of creating children solution in crossover operators used in binary-coded GA. SBX operator possesses strong local search ability and maintains the diversity of the population. So it can maintain the distribution of the solution. The procedure of SBX operator is displayed as follows.

**SBX Operator**

**Input**: Ne; population size, population \( P(t) \).

**Output**: Two new individual \( y_{i,1} \) and \( y_{i,2} \).

**Step 1.** Randomly select two different individuals \( x^1 = (x_{1,1}^1, x_{2,1}^1, \ldots, x_D^1) \) and \( x^2 = (x_{1,2}^2, x_{2,2}^2, \ldots, x_D^2) \) from \( P(t) \).

**Step 2** (SBX operator). Produce the new individual \( y_{i,j+1} = \{y_{i,j,1}, y_{i,j,2}, \ldots, y_{i,j,D}\}; \ D \) is the number of dimensions with (7):

\[
y_{i,j}^1 = 0.5 \times \left[(1 + \beta_j) x_{i,j}^1 + (1 - \beta_j) x_{i,j}^2\right],
\]

\[
y_{i,j}^2 = 0.5 \times \left[(1 - \beta_j) x_{i,j}^1 + (1 + \beta_j) x_{i,j}^2\right],
\]

where

\[
\beta_j = \begin{cases} 
(2u_i)^{1/(\eta_\epsilon+1)}, & \text{if } u_i \leq 0.5 \\
\left(\frac{1}{2(1-u_i)}\right)^{1/(\eta_\epsilon+1)}, & \text{otherwise}
\end{cases}
\]

which is called spread factor; \( u_i \) is a uniformly distributed random number between 0 and 1.

The probability distribution of the spread factor is as follows:

\[
\rho(\beta_j) = \begin{cases} 
0.5 \left(\frac{\eta_\epsilon + 1}{\beta_j^\eta_\epsilon}\right)^{\frac{1}{\eta_\epsilon+2}}, & \text{if } \beta_j \leq 1 \\
0.5 \left(\frac{\eta_\epsilon + 1}{\beta_j^{\eta_\epsilon+2}}\right), & \text{otherwise},
\end{cases}
\]

where \( \eta_\epsilon \) is the distribution index which determines the shape of the distribution.

(3) PM Operator. PM operator attempts to simulate the offspring distribution of binary-encoded bit-flip mutation on real-valued decision variables. PM operator is of benefit to maintain the diversity of the population and efficiently explore the solution space. Figure 2 displays the theory of PM operator. It can be seen from Figure 2 that PM can obtain more broad offspring individuals. Meanwhile, compared with linear mutation, the offspring individuals produced by PM are close to ideal Pareto set. The procedure of PM operator is displayed as follows.

**PM Operator**

**Input**: An individual \( x_i \), the upper bound \( x_{UB,i} \), and the lower bound \( x_{LB,i} \) of the \( i \)th decision variable.

**Output**: A new individual \( y_i \).

**Step 1** (PM operator). Produce the new individual \( y_{i,j+1} = \{y_{i,j,1}, y_{i,j,2}, \ldots, y_{i,j,D}\}; \ D \) is the number of dimensions with (9):

\[
y_{i,j} = x_i + (x_{UB,i} - x_{LB,i}) \times \delta,
\]

where the probability distribution of \( \delta \) is \( p(\delta) = 0.5(\eta_\delta + 1)(1-|\delta|)^{\eta_\delta} \) and \( \eta_\delta \) is the distribution index which determines the shape of the distribution. \( \delta \) can be calculated as follows:

\[
\delta(u) = \begin{cases} 
(2u)^{1/(\eta_\epsilon+1)} - 1, & \text{if } u < 0.5 \\
1 - \left[2(1-u)^{1/(\eta_\epsilon+1)}\right], & \text{if } u \geq 0.5.
\end{cases}
\]

3.3. Adaptive Selection of Evolutionary Operators Scheme. AOSPEA makes use of three evolutionary operators including SBX, PM, and DE. The selection probability of each operator is a third in the first generation. In the following generations, the selection probability is assigned in an adaptive way. Assuming that the number of solutions in the external archive is total and the number of solutions in the external archive produced by SBX, PM, and DE is noSBX, noPM, and
Polynomial mutation
Linear mutation
PS

Figure 2: The theory of PM operator.

In order to avoid any operator to be discarded when producing no solutions in one generation, a minimum selection probability is set. The rest probability is assigned according to their contribution to the external archive. Assuming that the minimum selection probability is Thres, the selection probability of SBX is PSB, the selection probability of PM is PM, and the selection probability of DE is PD, and their selection probability can be calculated as follows:

\[
\begin{align*}
\text{contriSBX} &= \frac{\text{noSBX}}{\text{total}}, \\
\text{contriPM} &= \frac{\text{noPM}}{\text{total}}, \\
\text{contriDE} &= \frac{\text{noDE}}{\text{total}}. 
\end{align*}
\]  

\(\text{(9)}\)

In order to avoid any operator to be discarded when producing no solutions in one generation, a minimum selection probability is set. The rest probability is assigned according to their contribution to the external archive. Assuming that the minimum selection probability is Thres, the selection probability of SBX is PSB, the selection probability of PM is PM, and the selection probability of DE is PD, and their selection probability can be calculated as follows:

\[
\begin{align*}
\text{PSB} &= \text{Thres} + (1 - 3 \times \text{Thres}) \times \text{contriSBX}, \\
\text{PM} &= \text{Thres} + (1 - 3 \times \text{Thres}) \times \text{contriPM}, \\
\text{PD} &= \text{Thres} + (1 - 3 \times \text{Thres}) \times \text{contriDE}. 
\end{align*}
\]  

\(\text{(10)}\)

The algorithm chooses corresponding evolutionary operator to generate offspring according to their selection probability. Rand is a uniformly distributed random number between 0 and 1. If rand \(\leq\) PSB, the algorithm chooses SBX to generate new solutions. If PS \(\leq\) rand \(\leq\) PSB + PM, the algorithm selects PM to produce new solutions. Otherwise, the algorithm chooses DE to generate new solutions.

3.4. The Proposed Algorithm (AOSPEA). According to the above descriptions of the simulated binary crossover, polynomial mutation, and differential evolution operator, SPEA2, and adaptive selection of evolutionary operators’ scheme, an improved SPEA2 algorithm with adaptive selection of evolutionary operators scheme (AOSPEA) is proposed. The procedure is shown as follows. And Figure 3 gives the flowchart of the AOSPEA.

AOSPEA Algorithm

Input:

- \(N_e\): population size
- \(N\): archive size
- \(T\): maximum number of generations.

Output: NDS: nondominated set.

\text{Step 1 (initialization).} Generate an initial population \(P(0)\) and create an empty archive (external archive) \(A(0)\). Set \(t = 0\).

\text{Step 2 (fitness assignment).} Calculate fitness values of individuals in \(P(t)\) and \(A(t)\).

\text{Step 3 (environment selection).} Copy all nondominated individuals in \(P(t)\) and \(A(t)\) to \(A(t + 1)\). If size of \(A(t + 1)\) exceeds \(N\) then reduce \(A(t + 1)\) by means of the truncation operator; otherwise if size of \(A(t + 1)\) is less than \(N\) then fill \(A(t + 1)\) with dominated individuals in \(P(t)\) and \(A(t)\).

\text{Step 4 (termination).} If \(t > T\) is satisfied, then stop and output NDS. Otherwise, continue.
Step 5 (mating selection). Perform binary tournament selection with replacement on $A(t + 1)$ in order to fill mating pool. The size of mating pool is $Ne$.

Step 6 (reproduction). If $t = 0$, randomly select SBX, PM, and DE to generate individuals in $P(t + 1)$; if $t > 1$, assign minimum selection probability to all the operators and then assign the rest probability according to their contribution to the external archive. Set $t = t + 1$; go to Step 2.

3.5. Convergence Analysis. For multiobjective optimization with infinite optimal Pareto solutions, the evolutionary algorithms based on finite population cannot obtain all Pareto solutions. Therefore, the target of multiobjective optimization algorithms is to obtain a subset of ideal Pareto set and make the subset distribute as broadly and uniformly as possible. We employ finite Markov chain to prove that AOSPEA algorithm asymptotically converges to the ideal Pareto set with probability 1.

Definition 4. If $S$ is a finite set, a random sequence $\{X_t, t = 0, 1, 2, 3, \ldots\}$ is from $S$. And if all $t \geq 0$, $\{i, j\}$ belongs to $S \times S$, there will exist

$$P \left[ X_{t+1}X_t = i, X_{t-1} = i_{t-1}, \ldots, X_0 = i_0 \right] = P \left[ X_{t+1}X_t = i \right] = p_{ij};$$

the sequence $\{X_t, t = 0, 1, 2, \ldots\}$ can be called a homogeneous finite Markov chain with state space $S$. The homogeneous finite Markov chain has the property that the current status now corresponds with the status for previous time.
Definition 5. Let \( T_m : S^N \rightarrow S \) denote the mutation in DE operator and its probability distribution can be described with (13):

\[
P(T_m(x) = x_j) = \sum_{x_i, x_k, x_l \in S^N} P(T^N_m(x) = \{x_i, x_j, x_k, x_l\}) .
\]  

(12)

Definition 6. Let \( T_c : S^2 \rightarrow S \) denote crossover in DE operator and its probability distribution can be described with (14), in which \( k \) is the number of crossovers:

\[
P(T_c(x, y) = \eta_l) = C^k_D CR(1 - CR)^{D-k},
\]  

\( k = 1, 2, 3, \ldots, D. \)

Definition 7. Let \( T_{SBX} : S^N \rightarrow S \) denote SBX crossover and its probability distribution can be described with (15):

\[
P(T_{SBX}(x, \beta) = y_j) = \sum_{x_i, x_k \in S^N} P(T_{SBX}(x) = \{x_i, x_k\}) \times \beta.
\]  

(14)

Definition 8. Let \( T_{PM} : S^N \rightarrow S \) denote PM crossover and its probability distribution can be described with (16):

\[
P(T_{PM}(x, x_{UB}, x_{LB}, \delta) = y_j) = \delta.
\]  

(15)

Definition 9. The transfer matrix of the homogeneous finite Markov chain is irreducible. If

\[
p_{ij}(k) > 0,
\]  

\( \forall i, j \in S, S \in \{T_m, T_c, T_{SBX}, T_{PM}\}, k \in \{1, 2, \ldots\}, \)

(16)

where \( p_{ij}(k) \) denotes the probability from the status \( i \) to the status \( j \) though \( k \) steps.

Definition 10 (see [47]). If a matrix \( A = [a_{ij}]_{nm} \), there exists \( a_{ij} > 0, i = 0, 1, \ldots, n; j = 0, 1, \ldots, m; A \) is the positive matrix.

Lemma 11 (see [48]). A Markov chain which has finite space and irreducible transition matrix will infinitely visit any state in \( S \) and is not relevant to the initial distribution probability 1.

Theorem 12. The population sequence \( \{X_t, t = 0, 1, 2, \ldots\} \) of AOSPEA algorithm is homogeneous and nonperiodic Markov chain with positive transition matrix.

Proof. In AOSPEA algorithm, the state set \( S \in \{T_m, T_c, T_{SBX}, T_{PM}\} \) is a finite set. The population sequence of the proposed algorithm can be shown as follows:

\[
X_t \xrightarrow{M_{SBX}, M_{PM}, M_{DE}} X_{t+1} = T(X_t)
\]  

(17)

where \( a, b, \) and \( c \) which satisfy \( a + b + c = 1 \) denote the selection probability by DE, SBX, PM, and DE. \( M_{SBX}, M_{PM}, \) and \( M_{DE} \) represent the transition matrix of SBX, PM, and DE, respectively. \( T_m, T_c, T_{SBX}, \) and \( T_{PM} \) are not relevant to the time \( t \). Therefore, it can be concluded from Definition 4 that \( \{X_t, t = 0, 1, 2, \ldots\} \) can be described as Markov chain in finite state set \( S \).

The evolution process of AOSPEA can be described as \( X_t \xrightarrow{P} X_{t+1} \), that is, \( X_{t+1} = X_t \times P = X_t \times M_{SBX} \times M_{PM} \times M_{DE} \); the transition matrix of the population sequence \( P = M_{SBX} \times M_{PM} \times M_{DE} \), where there exist \( P(T_m(X) = y_j) > 0, P(T_c(x, y) = \eta_l) > 0, P(T_{SBX}(X, \beta) = y_j) > 0, \) and \( P(T_{PM}(x, x_{UB}, x_{LB}, \delta) = y_j) > 0, M_{SBX}, M_{PM}, M_{DE} \) are positive transition matrix. Therefore, \( P = M_{SBX} \times M_{PM} \times M_{DE} \) is positive matrix. Therefore, \( \{X_t, t = 0, 1, 2, \ldots\} \) is homogeneous and nonperiodic Markov chain with positive transition matrix.

\[ \square \]

Theorem 13. The population sequence \( \{X_t, t = 0, 1, 2, \ldots\} \) of AOSPEA algorithm is asymptotically converged to a subset of the Pareto set with probability 1.

Proof. PS is defined to the ideal Pareto set for multiobjective problem. The population sequence \( \{X_t, t = 0, 1, 2, \ldots\} \) of AOSPEA is converged to any subset of PS with probability 1; that is to say, when \( t \rightarrow +\infty \), for any element \( a \in X_t \), there exists \( a \in \text{PS} \).

We suppose \( a \notin \text{PS} \), so there exists a Pareto optimal solution \( b \) in PS, which satisfies that \( b \) dominates \( a \). Definition 4 has demonstrated that \( \{X_t, t = 0, 1, 2, \ldots\} \) is homogeneous and nonperiodic Markov chain, and the transition matrix \( P \) is the positive matrix. Due to the positive matrix \( P, P \) is irreducible from Definition 9. Therefore, it can be concluded from Lemma 11 that \( \{X_t, t = 0, 1, 2, \ldots\} \) visits any state in \( S \in \{T_m, T_c, T_{SBX}, T_{PM}\} \) and is not relevant to the initial distribution with probability 1. Thereupon, when \( t \rightarrow +\infty \), the Pareto optimal solution \( b \) appears in \( X_t \) with probability 1. Since AOSPEA only retain the Pareto optima, \( a \) will be obsoleted by \( b \), which contradicted to the prior assumption.

Therefore, the population sequence \( \{X_t, t = 0, 1, 2, \ldots\} \) of AOSPEA algorithm is gradually converged to a subset of the Pareto set with probability 1.

\[ \square \]

3.6. Computational Complexity. The algorithm mainly includes fitness assignment, environment selection, and evolutionary operation according to the algorithm process. Assuming that the size of the population is \( N \), the size of external archive is \( \bar{N} \), the number of objectives is \( m \), the number of generations is \( T \), and the dimension of decision variable is \( D \), the time complexity of one generation for the algorithm can be calculated as follows: the time complexity for fitness assignment is \( O(M^2 \log M) + O(M^2) + O(M^2) \), where \( M = N + \bar{N} \); the time complexity for calculating objective values is \( O(Nm) \), and the worst time complexity for environment selection is \( O(M^2) \). The evolutionary operation includes SBX, PM, and DE operation. Assuming that the proportion of them are \( a, b, \) and \( c \) \((a + b + c = 1)\), respectively, the time complexity of the total evolutionary operation is
$O((a + b + c)ND) = O(ND)$. The worst total time complexity of one generation is

$$O(M^2 \log M) + O(M^2) + O(M^2) + O(Nm) + O(M^3) + O(ND) = O(M^3).$$

So the worst total time complexity is $O(TM^3)$ in $T$ generations.

### 4. Experiment and Results

#### 4.1. Benchmark Problems

To verify the performance of the proposed algorithm, 17 well-known multiobjective function optimization test instances are employed in this paper. There are five ZDT (ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6) problems [49], four DTLZ (DTLZ1, DTLZ2, DTLZ3, and DTLZ4) problems [50], and eight LZ09 ($LZ09_{F1}$, $LZ09_{F2}$, $LZ09_{F3}$, $LZ09_{F4}$, $LZ09_{F5}$, $LZ09_{F6}$, $LZ09_{F7}$, and $LZ09_{F9}$) problems [14]. All the test instances are the minimization of the objectives.

#### 4.2. Performance Metrics

The performance of $PF_{\text{final}}$ obtained by an algorithm is evaluated by convergence and diversity. The following five performance metrics [51–53] are adopted to measure its performance in this paper. There are Generational Distance (GD), Inverted Generational Distance (IGD), Spacing (SP), Maximum Spread (MS), and Hypervolume (HV).

#### 4.3. Experimental Setting

All the algorithms were implemented in MATLAB. AOSPEA is compared with SPEA2, NSGA-II, PESA-II, and MODEA [46]. The simulations were running on a PC with 2.1-GHz CPU and 2-GB RAM.

The SBX and PM operators are used in all the algorithms. The parameter values are listed in Table 1, where $m$ is the number of variables. For SPEA2, the population size is 100 and the size of external archive is 100. For NSGA-II, the population size is 100. For PESA-II, the internal population size is 100, the archive size is 100, and the number of hypergrid cells of each dimension is 32. For AOSPEA, the population size is 100, the size of external archive is 100, the minimum selection probability is 0.05, and $CR = 1.0$ and $F = 0.5$. For DE operation, since the problem size and complexity are different, the number of function evaluations is differently designed. For ZDT problems, the number of function evaluations is kept at 15000. For DTLZ, $LZ09_{F6}$, $LZ09_{F7}$, and $LZ09_{F9}$ problems, the number of function evaluations is kept at 50000. For $LZ09_{F1}$–$F5$ problems, the number of function evaluations is kept at 25000. Each algorithm is run 20 times independently for each test instance.

#### 4.4. Experimental Results

In order to validate the effectiveness and efficiency of the adaptive scheme, a group of
Table 2: Compared results on the performance of adaptive scheme through the AOSPEA.

| Problems | AOSPEA with adaptive scheme | AOSPEA without adaptive scheme |
|----------|-----------------------------|-------------------------------|
|          | Best           | Mean              | SD             | Best           | Mean              | SD             |
| ZDT1     | 2.0542e – 001 | 3.6619e + 000 | 8.9979e – 005 | 8.5312e + 000 | 6.2351e + 001 | 8.9979e – 004 |
| ZDT2     | 2.0542e – 001 | 3.3286e + 000 | 1.4725e – 004 | 3.8524e + 000 | 4.6035e + 001 | 3.5604e – 003 |
| DTLZ1    | 1.3192e – 001 | 7.8933e – 001 | 1.1547e – 003 | 1.4201e + 000 | 8.5367e + 000 | 6.8359e – 002 |
| DTLZ2    | 1.2063e – 001 | 4.1406e – 001 | 1.1547e – 003 | 1.0076e + 000 | 3.1204e + 000 | 4.3518e – 002 |
| LZ09_F1  | 6.8972e – 001 | 3.6591e + 000 | 3.5637e – 004 | 2.2571e + 001 | 7.0986e + 001 | 2.6439e – 003 |
| LZ09_F2  | 1.1047e – 002 | 3.5931e + 000 | 8.5929e – 003 | 3.5931e + 000 | 6.2857e + 000 | 2.9061e – 002 |
| Average value | 2.2736e – 001 | 2.5743e + 000 | 1.9160e – 003 | 6.8292e + 000 | 4.2314e + 001 | 2.4674e – 002 |

experiments are executed and the statistical results are listed in Table 2. In Table 2, the HV is chosen as the performance measurement. For the nonadaptive selection scheme, the selection probability is set with fixed value. It can be seen that the average value obtained by AOSPEA with adaptive scheme is better than that of AOSPEA without adaptive scheme, which demonstrate the efficiency of adaptive scheme.

Table 3 shows the mean and standard deviation of the metrics in four algorithms for ZDT problems. In each table cell, the first line is the mean value, the second line
| Problems | Algorithm | GD          | SP          | HV          | MS          | IGD         |
|----------|-----------|-------------|-------------|-------------|-------------|-------------|
|          | SPEA2     | 2.5857e−005 | 2.0969e−003 | 3.6614e+000 | 9.9991e−001 | 1.5820e−004 |
|          |           | 4.3495e−006 | 1.7634e−004 | 2.7732e−004 | 2.3685e−004 | 1.6398e−005 |
|          | NSGA-II   | 2.0347e−004 | 5.8214e−003 | 3.6605e+000 | 9.9713e−001 | 8.9535e−004 |
| ZDT1     |           | 5.3345e−005 | 5.3461e−004 | 5.4561e−004 | 2.6547e−004 | 3.2995e−005 |
|          | PESA-II   | 7.7737e−003 | 5.5438e−003 | 3.6419e+000 | 9.9487e−001 | 5.9252e−004 |
|          | MODEA     | 6.9582e−004 | 3.9421e−004 | 3.5694e−003 | 1.4235e−003 | 3.4621e−005 |
|          | AOSPEA    | 1.1964e−005 | 1.8849e−003 | 3.6619e+000 | 9.9985e−001 | 1.5234e−004 |
|          |           | 3.1517e−006 | 2.2202e−004 | 8.9979e−005 | 1.1150e−004 | 1.5427e−005 |
|          | SPEA2     | 2.0697e−005 | 2.1141e−003 | 3.3287e+000 | 9.9935e−001 | 7.1849e−005 |
| ZDT2     |           | 3.6952e−006 | 1.6354e−003 | 2.1098e−004 | 3.6254e−004 | 3.2556e−006 |
|          | NSGA-II   | 2.4227e−005 | 6.0673e−003 | 3.3234e+000 | 9.9885e−001 | 4.7150e−004 |
|          | PESA-II   | 7.5982e−006 | 4.9912e−004 | 2.8301e−004 | 1.3691e−003 | 6.9781e−005 |
|          | MODEA     | 7.1028e−003 | 5.3460e−003 | 3.3067e+000 | 9.9266e−001 | 4.5366e−004 |
|          | AOSPEA    | 2.3626e−005 | 2.3824e−004 | 3.3218e+000 | 9.9896e−001 | 7.3002e−005 |
| ZDT3     |           | 3.3681e−005 | 3.2003e−004 | 3.6469e−003 | 9.8293e−001 | 3.5983e−006 |
|          | SPEA2     | 3.5021e−006 | 2.0351e−003 | 2.1239e−003 | 4.6293e−004 | 3.5983e−006 |
|          | NSGA-II   | 7.6543e−004 | 3.8254e−004 | 4.2365e−003 | 2.6353e−003 | 8.6342e−005 |
|          | MODEA     | 3.5967e−005 | 3.9821e−004 | 7.6196e−003 | 2.1239e−003 | 5.1324e−005 |
| ZDT4     |           | 8.2750e−005 | 2.8098e−003 | 4.8150e−000 | 9.9797e−001 | 3.6550e−004 |
|          | SPEA2     | 1.3478e−005 | 2.8932e−003 | 1.1334e−004 | 6.3041e−004 | 1.7126e−005 |
|          | NSGA-II   | 6.7015e−004 | 2.2502e−003 | 3.6523e+000 | 9.9890e−001 | 4.6003e−004 |
|          | PESA-II   | 4.8822e−004 | 4.2264e−003 | 3.6536e+000 | 9.9738e−001 | 3.6178e−003 |
|          | MODEA     | 9.2848e−003 | 8.1803e−003 | 3.6496e+000 | 9.8374e−001 | 3.1883e−003 |
| ZDT6     |           | 3.5659e−004 | 3.2013e−004 | 2.3058e−003 | 3.5627e−003 | 8.5671e−004 |
|          | SPEA2     | 3.5724e−005 | 3.5445e−003 | 3.6527e+000 | 9.9810e−001 | 3.7861e−003 |
|          | NSGA-II   | 5.3690e−004 | 3.2013e−004 | 2.3058e−003 | 3.5627e−003 | 8.5671e−004 |
|          | MODEA     | 3.7248e−005 | 3.2003e−004 | 4.3504e−004 | 2.8079e−004 | 3.2367e−004 |
|          | AOSPEA    | 1.4636e−004 | 1.9829e−003 | 3.6618e+000 | 9.9931e+000 | 1.5982e−004 |
|          |           | 1.6977e−005 | 2.1056e−004 | 1.2257e−004 | 1.5588e−004 | 1.5130e−005 |
|          | SPEA2     | 6.1089e−004 | 1.9464e−003 | 3.0328e+000 | 9.9730e−001 | 5.5711e−004 |
|          | NSGA-II   | 2.6390e−005 | 2.5681e−004 | 3.2156e−005 | 3.9657e−004 | 1.5396e−005 |
|          | PESA-II   | 7.8852e−004 | 2.8317e−003 | 3.0314e+000 | 9.9718e−001 | 6.4428e−003 |
|          | MODEA     | 3.6076e−005 | 2.6674e−004 | 3.2567e−005 | 8.7635e−004 | 4.5907e−005 |
|          | AOSPEA    | 2.0740e−004 | 1.5639e−003 | 3.0376e+000 | 9.9861e−001 | 4.3419e−004 |
|          |           | 1.7334e−005 | 1.3660e−004 | 1.2996e−005 | 2.3794e−004 | 1.2755e−005 |
is the standard deviation, and bold indicates the optimal values among the compared algorithms. Figure 4 shows the nondominated solutions achieved by five algorithms for ZDT problems. The ZDT2 and ZDT6 test instances are chosen. These five problems have two objectives. For GD, AOSPEA is optimal for all the test instances except for ZDT2. For SP, AOSPEA is optimal for all the test instances except for ZDT3. For MS, SPEA2, NSGA-II, MODEA, and AOSPEA are close to one, but AOSPEA is slightly better than SPEA2, NSGA-II, and MODEA. The MS of PESA-II is worst because its boundary solutions can be replaced. For IGD and HV, AOSPEA is optimal for all the test instances. As a whole, it is can be seen from Table 3 and Figure 2 that AOSPEA is better than that of the others. Because PESA-II just reserve nondominated solutions in the archive, SPEA2, NSGA-II, MODEA, and AOSPEA can reserve dominated solutions when the number of nondominated solutions is less than the size. So the table also presents the fact that it is useful to reserve dominated solutions for ZDT problems at the early stage.

Figure 5 shows boxplots of the metrics for ZDT1. A boxplot is used to show the statistical results of a group data through five numerical data which are the lower bound, lower quartile, median, upper quartile, and upper bound. For GD, SP, and IGD, the five numerical data of SPEA and AOSPEA are less than others. But AOSPEA is slightly better than SPEA2, and AOSPEA is steadier than SPEA2 because the difference of lower quartile and upper quartile of AOSPEA is less than SPEA2. AOSPEA has the same performance with SPEA2 for MS, but AOSPEA is steadier. AOPSEA is better than others for HV and shows good robustness.

Table 4 and Figure 6 show statistic values and nondominated solutions on DTLZ test instances. These four problems have three objectives. For GD, AOSPEA is optimal for all the test instances but DTLZ1. For SP, AOSPEA is optimal on DTLZ4 and a little worse than the optimal value on the rest test instances. For MS, AOSPEA and SPEA2 are better than NSGA-II, MODEA, and PESA-II. So AOSPEA has a better diversity considering SP and MS. For HV, AOSPEA is the best of all. For IGD, PESA-II is optimal for all the test instances but DTLZ2. The solutions obtained by PESA-II are easily gathered together due to the fact that the MS of PESA-II is small. So the IGD of PESA-II is optimal for DTLZ test instances. It is clear from Figure 3 that AOSPEA performs much better than the others in terms of diversity. Overall, AOSPEA is better than others for DTLZ test instances.

Figure 7 shows boxplots of the metrics for DTLZ4. The lower quartile, median, quartile, and upper quartile of AOSPEA are less than others for GD, but SPEA is steadier than PESA2. AOSPEA performs the best for SP and HV and shows better steady. PESA-II has the best value for IGD, but the MS performs the worst. For the value of MS and IGD, AOSPEA shows slighter worse than the best.

Table 5 and Figure 8 show statistic values, nondominated solutions, and boxplots on LZ09 test instances. These eight problems have two objectives except F6 has three objectives. For GD, AOSPEA is optimal on LZ09_F1 and LZ09_F2, PESA-II is optimal on LZ09_F4, LZ09_F5, and LZ09_F9, SPEA2 is optimal on LZ09_F6, and MODEA is optimal on LZ09_F7. AOSPEA is a little worse than the optimal value on the rest test instances but DTLZ1. For SP, AOSPEA is optimal on LZ09_F1 and LZ09_F7. AOSPEA has the
best diversity considering SP and MS. PESA-II has a good SP, but its MS is the worst. For HV, AOSPEA is optimal for all the test instances but LZ09_F7. For IGD, AOSPEA is optimal on LZ09_F1, SPEA2 is optimal on LZ09_F6 and LZ09_F7, and PESA-II is optimal for the rest of test instances. AOPSEA is slightly worse than the optimal for LZ09_F2, LZ09_F5, LZ09_F6, LZ09_F7, and LZ09_F9. Figure 4 shows that AOSPEA have better spread and it is hard to converge for all the algorithms on LZ09_F7 and LZ09_F9. Overall, AOPSEA is better than other algorithms for most of LZ09 test instances.

Figure 9 shows boxplots of the metrics for LA09_F1. The lower quartile, median quartile, and upper quartile of AOSPEA perform the best for GD, SP, HV, and IGD and show the best robustness at the same time. AOSPEA performs slightly worse than NSGA-II and NSGA-II obtains the best maximum spread. AOSPEA shows the best convergence and diversity on LA09_F1.

For LZ problems, AOSPEA was compared with other four typical MOEAS which are SPEA2, NSGA-II, PESA-II, and MODEA. For NSGA-II, it adopts the operation-based representation to encode a chromosome. The POX crossover method and bit-flip mutation are used as reproduction operators. The probability of crossover and mutation are set to 0.5 and 0.1, respectively. The population size is set to 30. The other settings of the above algorithms keep consistent with the proposed algorithm. Each instance is executed by SPEA2, NSGA-II, PESA-II, and MODEA for 20 times independently, respectively. Table 6 reports the computational results obtained by SPEA2, NSGA-II, PESA-II, and MODEA. Table 6 includes problem name (Instances) and problem size (Dimension). The results of the mean relative error (MRE, MRE = 100 × (MRE – BKS (or UB))/BKS (or UB)) and the running time (CPU times) of AOSPEA, SPEA2, NSGA-II, PESA-II, and MODEA. The graphical representation in Figure 10 shows the comparison of benchmark standard problems results obtained from AOSPEA with SPEA2, NSGA-II, PESA-II, and MODEA. From Table 5, we know that the results obtained by the proposed algorithms are...
better than these four typical algorithms. Although AOSPEA does not obtain the best known solutions in the number of generations for large problems, the evolutionary trend of the population does not stagnate. That is to say, AOSPEA can further optimize obtaining the better solution. Meanwhile, the CPU time of all the compared algorithms is provided in Figure 11; it can be seen that the running time of AOSPEA is also superior to other algorithms. The population size of SPEA2, NSGA-II, PESA-II, and MODEA must keep a certain scale; otherwise they are easily trapped in local optimum, but the large population will increase the running time. AOSPEA has strong disturbance capacity, so even if the population is
Figure 9: Boxplots of the metrics for LA09.F1.

Figure 10: MRE of AOSPEA algorithm compared with SPEA2, NSGA-II, PESA-II, and MODEA.
Mathematical Problems in Engineering

5. Conclusion and Future Work

In this paper, an improved SPEA2 algorithm with adaptive selection of evolutionary operators is proposed. Various evolutionary operators and hybrid evolutionary methods are employed, which can greatly improve the searching ability. The adaptive scheme can select the corresponding operators according to their contribution to the external archive in the whole evolutionary process. This kind of selective way can make sure the proposed algorithm achieves the optimal values as soon as possible. Meanwhile, a minimum selection relatively small, it can boost the searching and readily escape the local optimum.
| Problem | Algorithm | GD       | SP       | HV       | MS       | IGD          |
|---------|-----------|----------|----------|----------|------------|--------------|
|      |           | 3.7556e-003 | 3.5552e-002 | 3.6424e+000 | 9.9340e-001 | 7.7606e-003 |
|      | SPEA      | 1.8319e-003 | 3.0150e-002 | 2.0262e-002 | 9.4758e-003 | 4.5673e-003 |
| LZ09_F1 | NSGA-II   | 2.7616e-003 | 8.5761e-003 | 3.6499e+000 | 9.9960e-001 | 3.3906e-003 |
|      | PESA-II   | 2.1032e-003 | 2.3946e-003 | 2.4461e-003 | 7.7068e-004 | 3.0940e-003 |
|      | MODEA     | 2.4732e-003 | 1.4495e-002 | 3.6381e+000 | 9.9267e-001 | 3.2740e-003 |
|      | AOSPEA    | 3.3856e-004 | 2.1617e-003 | 3.6591e+000 | 9.9899e-001 | 3.3817e-004 |
|      |           | 3.8198e-005 | 2.5593e-004 | 3.5637e-004 | 1.1890e-003 | 2.6264e-005 |
|      | SPEA      | 1.9169e-003 | 1.6310e-002 | 3.3038e+000 | 6.7685e-001 | 1.5532e-003 |
|      | NSGA-II   | 3.3207e-003 | 3.4150e-002 | 1.1211e-001 | 1.5286e-001 | 2.7729e-003 |
| LZ09_F2 | PESA-II   | 3.2721e-003 | 5.8267e-003 | 3.4290e+000 | 8.8184e-001 | 1.6837e-003 |
|      | MODEA     | 2.7294e-003 | 9.4258e-003 | 4.7836e-002 | 1.3044e-001 | 1.4259e-003 |
|      | AOSPEA    | 1.0923e-003 | 8.6235e-004 | 3.1987e+000 | 5.4580e-001 | 5.3251e-004 |
|      |           | 9.0903e-004 | 1.1030e-003 | 1.7616e-001 | 1.5717e-001 | 5.9315e-004 |
|      | SPEA      | 2.0361e-003 | 5.8738e-003 | 3.4388e+000 | 8.9295e-001 | 1.0004e-003 |
|      | NSGA-II   | 3.4339e-003 | 9.8427e-003 | 3.5319e+000 | 9.6780e-001 | 3.5023e-003 |
|      | PESA-II   | 2.4645e-004 | 6.7163e-003 | 3.6437e-002 | 1.7932e-002 | 3.3908e-003 |
|      | MODEA     | 5.0573e-003 | 6.8359e-003 | 3.5621e+000 | 9.7804e-001 | 3.0103e-003 |
|      | AOSPEA    | 2.6604e-004 | 7.0248e-003 | 3.6591e+000 | 9.9899e-001 | 3.3817e-004 |
|      |           | 3.1731e-003 | 7.0248e-003 | 3.6591e+000 | 9.9899e-001 | 3.3817e-004 |
|      | SPEA      | 4.0761e-003 | 6.4456e-003 | 3.4087e+000 | 8.4116e-001 | 3.3267e-003 |
|      | NSGA-II   | 6.9912e-004 | 1.5510e-003 | 1.9678e-002 | 4.7887e-002 | 6.2428e-004 |
| LZ09_F3 | PESA-II   | 3.7630e-003 | 7.3996e-003 | 3.4417e+000 | 8.4098e-001 | 2.9932e-003 |
|      | MODEA     | 5.2942e-004 | 2.5646e-002 | 2.7276e-002 | 5.5535e-002 | 4.8189e-004 |
|      | AOSPEA    | 3.6948e-003 | 7.8493e-003 | 3.3937e+000 | 7.7238e-001 | 2.8426e-003 |
|      |           | 9.0100e-004 | 1.0287e-003 | 2.8955e-002 | 3.8750e-002 | 7.6701e-004 |
|      | SPEA      | 3.8743e-003 | 7.0855e-003 | 3.5228e+000 | 9.4187e-001 | 3.1143e-003 |
|      | NSGA-II   | 5.3053e-004 | 2.6757e-003 | 2.8387e-002 | 5.6646e-002 | 4.9298e-004 |
|      | PESA-II   | 4.5735e-003 | 6.6016e-006 | 3.6236e+000 | 9.9572e-001 | 3.2898e-003 |
|      | MODEA     | 1.7735e-003 | 2.1646e-006 | 1.1086e-002 | 3.6365e-003 | 4.8174e-003 |
|      | AOSPEA    | 4.2443e-003 | 7.6946e-003 | 3.5087e+000 | 9.3333e-001 | 3.9606e-003 |
|      |           | 9.0707e-004 | 5.4273e-003 | 5.0148e-002 | 7.5440e-002 | 1.1625e-003 |
|      | SPEA      | 3.9777e-003 | 5.8253e-003 | 3.5712e+000 | 9.8091e-001 | 3.4926e-003 |
|      | NSGA-II   | 4.9673e-004 | 2.2365e-003 | 2.1052e-002 | 1.6274e-002 | 2.5889e-004 |
| LZ09_F5 | PESA-II   | 2.5690e-003 | 1.4395e-002 | 3.4477e+000 | 8.3870e-001 | 1.8529e-003 |
|      | MODEA     | 5.4005e-004 | 1.8361e-002 | 5.7840e-002 | 9.7133e-002 | 4.8959e-004 |
|      | AOSPEA    | 2.9658e-004 | 9.9364e-003 | 3.5832e+000 | 9.9562e-001 | 3.5047e-003 |
|      |           | 4.8584e-004 | 2.6534e-002 | 2.2602e-002 | 1.5383e-002 | 2.8994e-004 |
|      | SPEA      | 2.6175e-003 | 6.8127e-003 | 3.6261e+000 | 9.9676e-001 | 3.0101e-003 |
|      | NSGA-II   | 7.1940e-004 | 2.1710e-003 | 6.0387e-003 | 4.6184e-003 | 2.0291e-003 |
Table 5: Continued.

| Problem | Algorithm | GD       | SP       | HV       | MS       | IGD       |
|---------|-----------|----------|----------|----------|----------|-----------|
| LZ09_F6| SPEA2     | 1.0019e-003 | 1.2787e-001 | 2.4692e-001 | 9.9883e-001 | 7.2812e-002 |
|         | NSGA-II   | 5.5156e-004 | 4.2707e-002 | 8.5666e-003 | 1.1774e-002 | 2.0840e-002 |
|         | PESA-II   | 7.0744e-003 | 2.2985e-001 | 1.5436e-001 | 9.7899e+000 | 1.6556e-001 |
|         | MODEA     | 8.7776e-003 | 1.9554e-001 | 6.5348e-002 | 7.7052e-006 | 8.6470e-002 |
|         | AOSPEA    | 3.0472e-002 | 1.0007e-001 | 9.1255e-002 | 6.6200e-001 | 2.1902e-001 |
|         |           | 2.0434e-002 | 1.1519e-001 | 1.2405e-001 | 1.7424e-001 | 1.6415e-001 |
|         |           | 5.3345e-003 | 2.1874e-001 | 1.5456e-001 | 9.8768e+000 | 1.5651e-001 |
|         |           | 8.5666e-003 | 1.3448e-001 | 6.7324e-002 | 7.7052e-006 | 8.6470e-002 |
|         |           | 4.2271e-003 | 1.2543e-001 | 2.9381e-001 | 9.9923e-001 | 1.5651e-001 |
|         |           | 8.3194e-003 | 1.4478e-001 | 5.9100e-002 | 1.2849e-003 | 1.6190e-001 |
| LZ09_F7| SPEA2     | 2.7152e-003 | 1.8610e-002 | 2.7996e+000 | 2.6474e-001 | 1.6641e-003 |
|         | NSGA-II   | 3.2924e-003 | 2.7461e-002 | 1.7878e-001 | 2.8438e-001 | 1.6817e-003 |
|         | PESA-II   | 1.4830e-002 | 1.1319e-003 | 3.0175e+000 | 5.5930e-001 | 1.3810e-002 |
|         | MODEA     | 1.9428e-003 | 1.8242e-002 | 2.8634e+000 | 3.4062e-001 | 1.0021e-002 |
|         | AOSPEA    | 8.6571e-003 | 6.3791e-003 | 2.7293e+000 | 1.3810e-002 | 1.5221e-002 |
|         |           | 9.2704e-003 | 6.3791e-003 | 2.7293e+000 | 1.3810e-002 | 1.5221e-002 |
|         |           | 1.9428e-003 | 1.8242e-002 | 2.8634e+000 | 3.4062e-001 | 1.0021e-002 |
|         |           | 8.6571e-003 | 6.3791e-003 | 2.7293e+000 | 1.3810e-002 | 1.5221e-002 |
|         |           | 4.2271e-003 | 1.2543e-001 | 2.9381e-001 | 9.9923e-001 | 1.5651e-001 |
|         |           | 8.3194e-003 | 1.4478e-001 | 5.9100e-002 | 1.2849e-003 | 1.6190e-001 |

Figure 11: CPU time of AOSPEA compared with SPEA2, NSGA-II, PESA-II, and MODEA.
Table 6: Results using AGS_PAES for MK problems.

| Instance | Dimension | AOSPEA MRE | CPU times | SPEA2 MRE | CPU times | NSGA-II MRE | CPU times | PESA-II MRE | CPU times | MODEA MRE | CPU times |
|----------|-----------|------------|-----------|-----------|-----------|-------------|-----------|-------------|-----------|-----------|-----------|
| ZDT1     | 30        | 4.003      | 735.1     | 4.591     | 839.6     | 4.612       | 856.9     | 5.317       | 857.5     | 5.401     | 858.9     |
| ZDT2     | 30        | 4.892      | 725.9     | 4.445     | 841.5     | 4.961       | 837.7     | 5.116       | 843.8     | 5.921     | 849.7     |
| ZDT3     | 30        | 5.129      | 100.3     | 3.512     | 150.3     | 5.378       | 125.4     | 5.865       | 126.6     | 5.977     | 127.3     |
| ZDT4     | 10        | 5.549      | 273.3     | 6.068     | 378.1     | 6.543       | 405.1     | 6.651       | 415.7     | 7.076     | 417.1     |
| ZDT6     | 10        | 6.002      | 479.5     | 6.252     | 592.1     | 6.298       | 601.3     | 6.769       | 621.9     | 8.012     | 633.3     |
| DTLZ1    | 7         | 4.198      | 279.1     | 4.226     | 399.8     | 4.331       | 400.9     | 4.821       | 412.9     |           |           |
| DTLZ2    | 12        | 5.012      | 434.8     | 5.203     | 452.9     | 5.302       | 440.9     | 5.825       | 449.7     |           |           |
| DTLZ3    | 12        | 3.471      | 499.2     | 3.821     | 645.1     | 4.828       | 621.0     | 5.336       | 621.6     | 5.921     | 623.1     |
| DTLZ4    | 12        | 4.021      | 619.6     | 4.325     | 778.9     | 5.952       | 781.4     | 5.409       | 812.9     | 5.612     | 824.3     |
| LZ09_F1  | 10        | 5.887      | 647.3     | 6.208     | 751.2     | 6.802       | 789.1     | 7.337       | 829.7     | 8.389     | 831.1     |
| LZ09_F2  | 30        | 9.021      | 1415.1    | 9.205     | 1549.9    | 9.999       | 1576.9    | 9.657       | 1597.5    | 10.897    | 1618.9    |
| LZ09_F3  | 30        | 7.312      | 1268.9    | 7.625     | 1459.9    | 7.883       | 1469.8    | 8.331       | 1480.4    | 9.441     | 1501.8    |
| LZ09_F4  | 30        | 8.872      | 1199.8    | 9.105     | 1410.1    | 9.919       | 1461.6    | 9.664       | 1472.2    | 9.982     | 1483.6    |
| LZ09_F5  | 30        | 5.885      | 1170.3    | 6.115     | 1312.9    | 6.231       | 1332.1    | 6.723       | 1367.2    | 6.905     | 1399.1    |
| LZ09_F6  | 10        | 9.002      | 1149.7    | 9.205     | 1341.1    | 9.902       | 1361.5    | 10.008      | 1392.1    | 12.911    | 1419.5    |
| LZ09_F7  | 10        | 8.884      | 1074.4    | 9.012     | 1231.1    | 9.218       | 1236.2    | 9.733       | 1266.8    | 10.023    | 1289.2    |
| LZ09_F9  | 30        | 5.768      | 1141.9    | 6.003     | 1366.5    | 6.177       | 1399.7    | 6.691       | 1430.3    | 7.116     | 1459.7    |
| MRE      |           | 7.395      | 7.552     | 7.629     | 7.895     | 8.915       |           |            |           |           |

Probability is also set to avoid some operators which would have strong search ability in the remaining process of the algorithm. The experimental results verify these points.

In spite of good results which are achieved, there are some shortcomings related the proposed algorithm. The strength of the AOSPEA is not quite obvious while optimizing the instances with high dimensions. Besides, there is no reliable method to set the value of minimum selection probability.

Further research will be conducted in following directions. Firstly, we will consist in improving the performance of AOSPEA by making use of the adaptive scheme to mutation operator and verifying its efficiency through a comparison with other types of MOEAs. Secondly, more than two objectives in the MOPs will be studied. Finally, the improved AOSPEA will be utilized to solve the multiobjective job shop and flow shop scheduling problems.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this manuscript.

Acknowledgments

This work was financially supported by the National Natural Science Foundation of China under Grant no. 61663023. It was also supported by the General and Special Program of the Postdoctoral Science Foundation of China, the Science Foundation for Distinguished Youth Scholars of Lanzhou University of Technology, and Lanzhou Science Bureau project under Grant nos. 2012M521802, 2013T60889, J201405, and 2013-4-64, respectively.

References

[1] X. Li, L. Gao, and W. Li, “Application of game theory based hybrid algorithm for multi-objective integrated process planning and scheduling,” Expert Systems with Applications, vol. 39, no. 1, pp. 288–297, 2012.

[2] J. D. Schaffer, Some experiments in machine learning using vector evaluated genetic algorithms [Ph.D. thesis], Vanderbilt University, 1985.

[3] N. Srinivas and K. Deb, “Multiobjective optimization using nondominated sorting in genetic algorithms,” Evolutionary Computation, vol. 2, no. 3, pp. 221–248, 1994.

[4] C. M. Fonseca, “Genetic algorithms for multiobjective optimization: formulation, discussion and generalization,” in Proceedings of the 5th International Conference on Genetic Algorithms, pp. 416–423, Urbana-Champaign, III, USA, June 1993.

[5] J. Horn, N. Nafpliotis, and D. Goldberg, “A Niched Pareto genetic algorithm for multiobjective optimization,” in Proceedings of the 1st IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Orlando, Fla, USA, June 1994.

[6] J. D. Knowles and D. W. Corne, “Approximating the nondominated front using the Pareto Archived Evolution Strategy,” Evolutionary Computation, vol. 8, no. 2, pp. 149–172, 2000.

[7] D. W. Corne, J. D. Knowles, and M. J. Oates, “The Pareto envelope-based selection algorithm for multiobjective optimization,” in Proceedings of the Parallel Problem Solving from Nature 6th International Conference, pp. 839–848, Paris, France, September 2000.

[8] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, “PESA-II: region-based selection in evolutionary multiobjective optimization,” in Proceedings of the Genetic and Evolutionary Computation Conference (GECCO ’01), 2001.
[9] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach,” IEEE Transactions on Evolutionary Computation, vol. 3, no. 4, pp. 257–271, 1999.

[10] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: improving the strength pareto evolutionary algorithm,” in Proceedings of the International Conference on Evolutionary Methods for Design Optimisation and Control with Application to Industrial Problems, pp. 95–100, 2002.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, 2002.

[12] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, “Objective reduction in many-objective optimization: linear and nonlinear algorithms,” IEEE Transactions on Evolutionary Computation, vol. 17, no. 1, pp. 77–99, 2013.

[13] Q. Zhang and H. Li, “MOEA/D: a multiobjective evolutionary algorithm based on decomposition,” IEEE Transactions on Evolutionary Computation, vol. 11, no. 6, pp. 712–731, 2007.

[14] H. Li and Q. Zhang, “Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II,” IEEE Transactions on Evolutionary Computation, vol. 13, no. 2, pp. 284–302, 2009.

[15] L. Ke, Q. Zhang, and R. Battiti, “MOEA/D-ACO: a multiobjective evolutionary algorithm using decomposition and ant colony,” IEEE Transactions on Cybernetics, vol. 43, no. 6, pp. 1845–1859, 2013.

[16] Z. Zhang, N. Zhang, and Z. Feng, “Multi-satellite control resource scheduling based on ant colony optimization,” Expert Systems with Applications, vol. 41, no. 6, pp. 2816–2823, 2014.

[17] H. Hirano and T. Yoshikawa, “A study on two-step search based on PSO to improve convergence and diversity for Many-Objective Optimization Problems,” in Proceedings of the IEEE Congress on Evolutionary Computation (CEC’13), pp. 1854–1859, June 2013.

[18] P. Melin, F. Olivas, O. Castillo, F. Valdez, J. Soria, and M. Valdez, “Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic,” Expert Systems with Applications, vol. 40, no. 8, pp. 3196–3206, 2013.

[19] M. Gong, T. Hou, B. Fu, and L. Jiao, “A non-dominated neighbor immune algorithm for community detection in networks,” in Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO ’11), ACM, Dublin, Ireland, July 2011.

[20] G. Costa Silva, R. M. Palhares, and W. M. Caminhas, “Immune inspired Fault Detection and Diagnosis: a fuzzy-based approach of the negative selection algorithm and participatory clustering,” Expert Systems with Applications, vol. 39, no. 16, pp. 12474–12486, 2012.

[21] Y. Gao, L. Peng, F. Li, M. Liu, and X. Hu, “EDA-Based multi-objective optimization using preference order ranking and multivariate gaussian copula,” in Advances in Neural Networks-ISNN 2013: 10th International Symposium on Neural Networks, Dalian, China, July 4–6, 2013, Proceedings, Part II, vol. 7952 of Lecture Notes in Computer Science, pp. 341–350, Springer, Berlin, Germany, 2013.

[22] L. Wang, C. Fang, P. N. Suganthan, and M. Liu, “Solving system-level synthesis problem by a multi-objective estimation of distribution algorithm,” Expert Systems with Applications, vol. 41, no. 5, pp. 2496–2513, 2014.

[23] A. A. El-Sawy, M. A. Hussein, E. M. Zaki, and A. A. Mousa, “Local search-inspired rough sets for improving multiobjective evolutionary algorithm,” Applied Mathematics, vol. 5, no. 13, pp. 1993–2007, 2014.

[24] K. Li, A. Fialho, S. Kwong, and Q. Zhang, “Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition,” IEEE Transactions on Evolutionary Computation, vol. 18, no. 1, pp. 114–130, 2014.

[25] K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints,” IEEE Transactions on Evolutionary Computation, vol. 18, no. 4, pp. 577–601, 2014.

[26] S. Bleuler, M. Brack, L. Thiele, and E. Zitzler, “Multiobjective genetic programming: reducing bloat using SPEA2,” in Proceedings of the Congress on Evolutionary Computation, May 2001.

[27] M. Kim, T. Hiroyasu, M. Miki, and S. Watanabe, “SPEA2+: improving the performance of the strength pareto evolutionary algorithm 2,” in Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN ’04), Birmingham, UK, September 2004, pp. 3242 of Lecture Notes in Computer Science, pp. 742–751, Springer, 2004.

[28] Z.-H. Zheng, Q. Ai, W.-H. Xu et al., “Multi-objective load dispatch in wind power integrated system based on pseudo-parallel SPEA2 algorithm,” Journal of Shanghai Jiaotong University, vol. 43, no. 8, pp. 1222–1227, 2009.

[29] T.-T. Wu, H.-T. Geng, J.-Y. Yang, and Q.-X. Song, “An improved individual evaluation and elitism selection for distribution performance of SPEA2,” in Proceedings of the 1st International Conference on Information Science and Engineering (ICISE ’09), pp. 125–128, IEEE, Nanjing, China, December 2009.

[30] Z. Li, Y. Li, and X. Duan, “Improved strength pareto evolutionary algorithm with local search strategies for optimal reactive power flow,” Information Technology Journal, vol. 9, no. 4, pp. 749–757, 2010.

[31] N. Belgasi, L. B. Said, and K. Ghedira, “Greedy local improvement of SPEA2 algorithm to solve the multiobjective capacitated transshipment problem,” in Learning and Intelligent Optimization, Lecture Notes in Computer Science, pp. 364–378, Springer, Berlin, Germany, 2011.

[32] M. T. Al-Hajri and M. A. Abido, “Multiobjective optimal power flow using Improved Strength Pareto Evolutionary Algorithm (SPEA2),” in Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA’11), pp. 1097–1103, Córdoba, Spain, November 2011.

[33] W. Sheng, Y. Liu, X. Meng, and T. Zhang, “An improved strength pareto evolutionary algorithm 2 with application to the optimization of distributed generations,” Computers & Mathematics with Applications, vol. 64, no. 5, pp. 944–955, 2012.

[34] H. H. Maheta and V. K. Dabhi, “An improved SPEA2 Multi objective algorithm with non dominated elitism and Generational Crossover,” in Proceedings of the International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT ’14), pp. 75–82, IEEE, Ghaziabad, India, February 2014.

[35] K. Deb, “Multi-objective optimisation using evolutionary algorithms: an introduction,” in Multi-Objective Evolutionary Optimisation for Product Design and Manufacturing, pp. 3–34, Springer, Berlin, Germany, 2011.

[36] M. D. Torres, A. Torres, F. Cuellar, M. D. L. L. Torres, E. P. D. León, and F. Pinales, “Evolutionary computation in the identification of risk factors. Case of TRALI,” Expert Systems with Applications, vol. 41, no. 3, pp. 831–840, 2014.
20 Mathematical Problems in Engineering

[37] F.-K. Wang and T. Du, "Implementing support vector regression with differential evolution to forecast motherboard shipments," Expert Systems with Applications, vol. 41, no. 8, pp. 3850–3855, 2014.

[38] T. T. Nguyen, Z. Li, S. Zhang, and T. K. Truong, "A hybrid algorithm based on particle swarm and chemical reaction optimization," Expert Systems with Applications, vol. 41, no. 5, pp. 2134–2143, 2014.

[39] F. Valdez, P. Melin, and O. Castillo, "A survey on nature-inspired optimization algorithms with fuzzy logic for dynamic parameter adaptation," Expert Systems with Applications, vol. 41, no. 14, pp. 6459–6466, 2014.

[40] G. T. Pulido and C. A. C. Coello, "The micro genetic algorithm 2: towards online adaptation in evolutionary multiobjective optimization," in Evolutionary Multi-Criterion Optimization, Springer, Berlin, Germany, 2003.

[41] V. L. Huang, S. Z. Zhao, R. Mallipeddi, and P. N. Suganthan, "Multi-objective optimization using self-adaptive differential evolution algorithm," in Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’09), pp. 190–194, Trondheim, Norway, May 2009.

[42] A. J. Nebro, J. J. Durillo, M. Machín, C. A. C. Coello, and B. Dorronsoro, "A study of the combination of variation operators in the NSGA-II algorithm," in Advances in Artificial Intelligence: 15th Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2013, Madrid, Spain, September 17–20, 2013. Proceedings, pp. 269–278, Springer, Berlin, Germany, 2013.

[43] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.

[44] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," Complex Systems, vol. 9, no. 3, pp. 1–15, 1994.

[45] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," Computer Science and Informatics, vol. 26, no. 4, pp. 30–45, 1996.

[46] M. Ali, P. Siarry, and M. Pant, "An efficient Differential Evolution based algorithm for solving multi-objective optimization problems," European Journal of Operational Research, vol. 217, no. 2, pp. 404–416, 2012.

[47] Y.-T. Qi, F. Liu, W.-Y. Chang, X.-L. Ma, and L.-C. Jiao, "Memetic immune algorithm for multiobjective optimization," Journal of Software, vol. 24, no. 7, pp. 1529–1544, 2013.

[48] M. Iosifescu, Finite Markov Processes and Their Applications, John Wiley & Sons, Chichester, UK, 1980.

[49] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," Evolutionary Computation, vol. 8, no. 2, pp. 173–195, 2000.

[50] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in Evolutionary Multiobjective Optimization, pp. 105–145, Springer, London, UK, 2005.

[51] C. A. Coello Coello, "Evolutionary multi-objective optimization: a historical view of the field," IEEE Computational Intelligence Magazine, vol. 1, no. 1, pp. 28–36, 2006.

[52] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in Evolutionary Multi-Criterion Optimization: 8th International Conference, EMO 2015, Guimarães, Portugal, March 29 – April 1, 2015. Proceedings, Part II, vol. 9019, pp. 110–125, Springer, Berlin, Germany, 2015.

[53] M. Emmerich, N. Beume, and B. Naujoks, An EMO Algorithm Using the Hypervolume Measure as Selection Criterion, Springer, Berlin, Germany, 2005.
