

## Research Article

# Latest Stored Information Based Adaptive Selection Strategy for Multiobjective Evolutionary Algorithm

Jiale Gao, Qinghua Xing, Chengli Fan, and Zhibing Liang

Air Force Engineering University, Xi'an, China

Correspondence should be addressed to Jiale Gao; [gaojiale\\_kgd@163.com](mailto:gaojiale_kgd@163.com)

Received 4 July 2017; Revised 18 October 2017; Accepted 15 November 2017; Published 17 December 2017

Academic Editor: Salvatore Alfonzetti

Copyright © 2017 Jiale Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The adaptive operator selection (AOS) and the adaptive parameter control are widely used to enhance the search power in many multiobjective evolutionary algorithms. This paper proposes a novel adaptive selection strategy with bandits for the multiobjective evolutionary algorithm based on decomposition (MOEA/D), named latest stored information based adaptive selection (LSIAS). An improved upper confidence bound (UCB) method is adopted in the strategy, in which the operator usage rate and abandonment of extreme fitness improvement are introduced to improve the performance of UCB. The strategy uses a sliding window to store recent valuable information about operators, such as factors, probabilities, and efficiency. Four common used DE operators are chosen with the AOS, and two kinds of assist information on operator are selected to improve the operators search power. The operator information is updated with the help of LSIAS and the resulting algorithmic combination is called MOEA/D-LSIAS. Compared to some well-known MOEA/D variants, the LSIAS demonstrates the superior robustness and fast convergence for various multiobjective optimization problems. The comparative experiments also demonstrate improved search power of operators with different assist information on different problems.

## 1. Introduction

Multiobjective optimization is a common problem that scientists and engineers face, which concerns optimizing problems with multiple and often conflicting objectives. In principle, there is no single solution for a multiobjective optimization problem (MOP), but a set of Pareto-optimal solutions. This paper considers the following continuous MOP:

$$\begin{aligned} & \text{minimize} && F(x) = (f_1(x), \dots, f_m(x)) \\ & \text{subject to} && x \in \Omega, \end{aligned} \quad (1)$$

where  $x = (x_1, x_2, \dots, x_n)$  is a  $n$ -dimensional decision variable vector,  $\Omega \subset R^n$  is the decision space, and  $F : \Omega \rightarrow R^m$  consists of  $m$  real-valued continuous objective functions  $f_1(x), \dots, f_m(x)$  [1, 2].

Over the past decades, a number of multiobjective evolutionary algorithms (MOEAs) have been proposed. The first MOEA, named vector evaluated genetic algorithm, has been used for MOPs since 1980s [3] and after that more and more attention has been attached to MOEA. The first generation of

intelligence algorithms for MOPs represented by NSGA [4], NPGA [5], and MOGA [6] were presented in 1990s. In 2000s, the second generation appeared, and the renowned ones include NSGA-II [7], SPEA-II [8], MOEA/D [9], and some biology imitated algorithms, such as MOPSO [2], MOIA [10], MOACO [11]. These algorithms are usually classified into 3 categories: Pareto based methods [4–8], indicator-based methods [12], and decomposition-based methods [9]. The first one evaluates individuals based on the nondominated ranking. The second one integrates the convergence and diversity into a single indicator to guide evolution. The last one decomposes an MOP into a set of single subproblems, and evaluates solutions with regard to reference vector.

An extreme difficulty for most MOEAs is how to promote the searching efficiency; that is to say how to improve the operator to induce a more probability for searching high dimensional space. There are two improving methods: enhancing operator with adaptive parameter control and using multiple operators with an adaptive operator selection (AOS). The simulated binary crossover (SBX) [13] and the differential operator (DE) [14] are widely adopted, and they

are usually combined with the binomial crossover or the polynomial mutation, as these combinations could bring powerful search ability [15, 16]. There are still some limitations for composite operators [17] and variants [18] when searching complex problems. However, more and more adaptive control strategies are considered indispensable. In [19],  $p$  best individuals are selected as donor vectors for the DE, and the parameter of every mutation vary within the range of the statistical data of each generation. Zhu et al. [17] design a novel recombination operator possessing the advantages of both the DE and the SBX. Its adaptive parameter control strategy is to allocate probabilities for SBX and DE according to the search period, whereas Zhao et al. [20] suggest that the different neighborhood sizes have an unavoidable influence on the search power of operators based on the framework of MOEA/D, and the experiments imply that adaptive selection of neighborhood sizes works very well.

The major intention of the adaptive parameter control is to solve an essential problem regarded as the exploration versus exploitation (EvE) dilemma. Exploitation means searching the local space deeply, that is to say making full use of the current operator with current parameters. However, exploration means the operator search power for unfamiliar areas, which is displayed by other operators or the configuration of the current operator with different parameters. In conclusion, the EvE dilemma can be considered as looking for a tradeoff of the search power both in unfamiliar and familiar areas.

The EvE dilemma is of great significance for the existence of AOS methods. The EvE dilemma has been intensively studied in the game theory community for dealing with the multiarmed bandit (MAB) problem, which was first proposed in 1952 [21]. An interesting strategy, the upper confidence bound (UCB) selection strategy, has been used to solve the EvE dilemma since 1994 [22]. It possesses distinctive advantages among many AOSs, and a host of improved UCB versions appears after that. The application of UCB strategy in solving the MAB problem has been widely recognized, and in the following part UCB is referred to as the MAB algorithm for convenience. DMAB [23] presents the combination of the MAB problem and two statistical tests and suggests that the operator selection is sensitive to any change of the reward distribution. SIMAB [24] uses a sliding time window to store recent rewards and applied operators with a mechanism of first in first out (FIFO). Compared with DMAB, SIMAB possesses less assessment and factors. Furthermore, two rank based methods, the Area Under Curve and the Sum of Ranks, are presented for assigning credits, respectively [25]. Inspired by SIMAB and rank based credit assignment, Li et al. [16] suggest that a decay factor is useful for the credit assignment and it can improve the selection probability of the best operator. Two modified MAB methods are provided, called UCB-Tuned and UCB-V, in which reward variance is used as a parameter for a better EvE tradeoff [26].

Whether enhancing operator with adaptive parameter control or using multiple operators with AOSs, the statistical data about the operators is vitally meaningful with no doubt. Enlightened by the sliding time window, we present a novel adaptive method which is used to store the information about the operators, called latest stored information based adaptive

selection (LSIAS) strategy. The information of operators includes operator names, operator efficacies, and parameters about operators such as neighborhood sizes, scaling factors, and other parameters of operators (the parameters about operators will be regarded as assist information on operator for convenience). In this paper, two kinds of assist information are taken into account and they are used within MOEA/D with dynamical resource allocation (MOEA/D-DRA) [27], which won the championship of the CEC 2009 MOEA contest. The reason of choosing a decomposition-based algorithm is mainly because each decomposed subproblem is a single objective problem, which easily gives an exact value to measure the performance of operators every time. To validate the effectiveness and robustness of LSIAS, 22 well-known benchmark problems such as ZDT problems [28], DTLZ problems [29], and UF problems [30] are adopted. The comparative experiments are demonstrated when the MOEA/D-LSIAS compares with some versions of MOEA/D, for example, MOEA/D-DE [31], MOEA/D-DRA [27], MOEA/D-FRRMAB [16], MOEA/D-STM [32], MOEA/D-UCB-T [21], and MOEA/D-AGR [33].

The rest of this paper is organized as follows. The background and some works regarding the AOS and the adaptive parameter control are described in Section 2. The detail description of the LSIAS strategy is given in Section 3, and its utilization with the MOEA/D-DRA is presented in Section 4. Section 5 analyzes the result of the comparison experiments. Finally, conclusions are summarized and further work along the direction of the adaptive selection is discussed.

## 2. Related Background

*2.1. Tchebycheff Approach in MOEA/D.* MOEA/D provides a method which decomposes an MOP into a series of single problems. It is suitable to evaluate the performance of operators. There are three common decomposition methods: the weighted sum approach, the Tchebycheff approach, and the boundary intersection method, which are all described in [9]. This paper employs the Tchebycheff approach and it is in the form

$$\begin{aligned} & \text{Minimize } g^{\text{tc}}(x | \lambda, z^*) = \max \{ \lambda_i | f_i(x) - z_i^* | \}, \\ & \text{subjective to } x \in \Omega, \end{aligned} \quad (2)$$

where  $z^* = (z_1^*, z_2^*, \dots, z_m^*)$  is the reference point and  $z_i^* = \min(f_i(x) | x \in \Omega)$  for each  $i = 1, 2, \dots, m$ . For each Pareto optimal point  $x^*$  there exists a weight vector  $\lambda$  such that  $x^*$  is the optimal solution of (2) and each optimal solution of (2) is a Pareto optimal solution of (1). Therefore, one is able to obtain different Pareto optimal solutions by altering the weight vector.

*2.2. Basic Differential Evolution Operator.* Differential evolution (DE) is a parallel direct search method. There are many various mutation strategies about DE. Here, several frequently used DE operators [15, 34–36] are given as follows:

- (i) DE/rand/1:  $v^i = x^i + F \times (x^{r_1} - x^{r_2})$ ,
- (ii) DE/rand/2:  $v^i = x^i + F \times (x^{r_1} - x^{r_2}) + F \times (x^{r_3} - x^{r_4})$ ,

- (iii) DE/target-to-rand/1:  $v^i = x^i + K \times (x^i - x^{r_1}) + F \times (x^{r_2} - x^{r_3})$ ,
- (iv) DE/target-to-rand/2:  $v^i = x^i + K \times (x^i - x^{r_1}) + F \times (x^{r_2} - x^{r_3}) + F \times (x^{r_4} - x^{r_5})$ ,
- (v) DE/best/1:  $v^i = x^{\text{best}} + F \times (x^{r_1} - x^{r_2})$ ,
- (vi) DE/best/2:  $v^i = x^{\text{best}} + F \times (x^{r_1} - x^{r_2}) + F \times (x^{r_3} - x^{r_4})$ ,
- (vii) DE/target-to-best/1:  $v^i = x^i + K \times (x^{\text{best}} - x^i) + F \times (x^{r_1} - x^{r_2})$ ,

where  $x^i$  is the target vector,  $v^i$  is the mutant vector, and both of them belong to the  $i$ th subproblem.  $x^{\text{best}}$  is one of the best individual vectors in the population. The scale factors  $F$  and  $K$  are used to control the influence of the mutant vector difference within the range  $[0, 1]$ . The donor vectors  $x^{r_1}$ ,  $x^{r_2}$ ,  $x^{r_3}$ ,  $x^{r_4}$ , and  $x^{r_5}$  are different individuals.

After the differential evolution, the crossover operation and the polynomial mutation usually go on to mutate the vector  $v^i$ . The crossover operation decides which dimension of the new solution will be replaced by  $x^i$  or the trial vector  $v^i$ . Basically, DE works with the binomial crossover [14], and the trial vector  $u^i = (u_1^i, \dots, u_n^i)$  is formed as follows:

$$u_j^i = \begin{cases} v_j^i, & \text{if } r < \text{CR or } j = j_{\text{rand}}, \\ x_j^i, & \text{otherwise,} \end{cases} \quad (3)$$

where  $r$  is a random number in the range  $[0, 1]$ . CR is the crossover rate and has the same range as  $r$ .  $j \in \{1, \dots, n\}$  and  $j_{\text{rand}}$  is a random integer within the range  $[1, n]$ .

The polynomial mutation provides a deep mutation for the  $u^i$  to generate a new solution  $y_j^i$ . The polynomial mutation is defined as follows:

$$y_j^i = \begin{cases} u_j^i + \sigma_j \times (X_{\text{max}_j} - X_{\text{min}_j}), & \text{if } r < p_m, \\ u_j^i, & \text{otherwise,} \end{cases} \quad (4)$$

where  $X_{\text{max}_j}$  and  $X_{\text{min}_j}$  are the lower and upper bounds of the  $j$ th variable, respectively,  $p_m$  is the mutation probability, and  $r$  is a random number within the range  $[0, 1]$ .  $\sigma_j$  is a mutation factor and is obtained by

$$\sigma_j = \begin{cases} (2 \times r)^{1/(1+\eta)} - 1, & \text{if } r < 0.5, \\ 1 - 2 \times (1 - r)^{1/(1+\eta)} & \text{otherwise,} \end{cases} \quad (5)$$

where  $r$  is a random number within the range  $[0, 1]$  and  $\eta$  is the distribution index and is usually set to 20.

The description above is a common DE mutation process. It has been validated that these different DE operators enjoy different search powers. "DE/rand/1" and "DE/rand/2" show strong exploration performance. "DE/target-to-rand/1," "DE/best/1," and "DE/best/2" manifest their perfect exploitation performance and are useful for unimodal problems. However, "DE/target-to-rand/1" is more suitable for rotated problems than other DE operators [19]. Accordingly, a combination of multiple DE operators could offer a strong search power to solve most MOPs.

**2.3. Adaptive Operator Selection.** Every operator is in possession of different search power. Although parameter self-adaptive adjustment improves their search power, it is limited by its best performance. The AOS offers intense search power by choosing different operators when faced with different dilemmas.

There are mainly two AOS methods: the probability based methods, such as the probability matching (PM) [37] and the adaptive pursuit (AP) [38], and the bandits based methods [39]. The similarity of all AOS methods is that they enjoy similar processes, which are applied operator based credit assignment and selection based credit accumulation. Nevertheless, the detail of selection and credit accumulation is different.

**2.3.1. Credit Assignment.** Two credit assignment methods are often used. One is the dynamic statistic information evaluation about operators, and the other is the search power evaluation which uses various complex statistics to detect outlier production. The former takes recent assist information on operator into account. Some recent assist information is employed as rewards which decide the credit assignment of operators [24]. Rank factor is used to increase the use frequency of better operators. The latter possesses a complex calculation in consideration of two measures, fitness and diversity. In [40], the evaluation criterion depends on the appearance probability of outlier solutions. This evaluation criterion does not regard the fitness as the unique criterion, as the authors argue that infrequent but powerful operators are as significant as frequent but powerless operators. Density estimator is adopted as evaluation criterion in [41, 42], and a statistic method is added in [42]. This method calculates the normalized relative fitness improvements from successful operators, and then it regards the mean value of the improvements brought by operators as the credit. In [26], four different credit assignment methods are adopted, which are Average Absolute Reward, Extreme Absolute Reward, Average Normalized Reward, and Extreme Normalized Reward. All the rewards of each method are evaluated, and the method with the max probability is chosen as the credit assignment method at current generation.

**2.3.2. Operator Selection.** Assume that there are  $K$  different operators, and  $s_{i,t}$  and  $\hat{q}_{i,t}$  are the probability vector and the estimate of the  $i$ th operator reward, respectively.

*Probability Matching*

$$\hat{q}_{i,t+1} = (1 - \alpha) \times \hat{q}_{i,t} + \alpha \times r, \quad (6)$$

$$s_{i,t+1} = p_{\text{min}} + (1 - K \times p_{\text{min}}) \times \frac{\hat{q}_{i,t+1}}{\sum_j^K \hat{q}_{j,t+1}}, \quad (7)$$

where  $r$  is the reward of selected and successfully applied operator and  $t$  is the time point.  $\alpha$  is a decay factor which alleviates the influence brought by the accumulation reward of previously used operators. In this method, the worst probability is  $p_{\text{min}}$  and the best probability is  $p_{\text{max}} = 1 - (K - 1) \times p_{\text{min}}$ . So each operator has a nonzero probability

to be chosen during the whole search process. As every operator performs differently at different phases, nonzero probability operator selections are very suitable for AOSs, and it manifests superior robustness.

#### Adaptive Pursuit

$$i_t^* = \arg \max_{i \in \{1, \dots, K\}} \{\hat{q}_{i,t}\}, \quad (8)$$

$$s_{i,t+1} = s_{i,t} + \beta \times (p_{\max} - s_{i,t}), \quad \text{if } i = i_t^*,$$

$$s_{i,t+1} = s_{i,t} + \beta \times (p_{\min} - s_{i,t}) \quad \text{otherwise,}$$

where  $\hat{q}_{i,t}$  is the same as  $\hat{q}_{i,t}$  in (6). When the best operator is successfully applied, it will get a relatively better reward. The accumulation is enlarged and it is the reason why this selection strategy always chooses the best operator, noted  $i_t^*$ .

#### Multiarmed Bandit

$$\hat{q}_{i,t} + C \times \sqrt{\frac{2 \times \ln \sum_{j=1}^K n_{j,t}}{n_{j,t}}}, \quad (9)$$

where  $\hat{q}_{i,t}$  is the same as that in (6),  $n_{j,t}$  is the successful times of  $j$ th operator at point  $t$ , and  $C$  is a scaling factor of the tradeoff of different search powers. In the MAB algorithm, the operator denotes the arm.

Each operator gets different credit after credit assignment, which is the key of operator selection. As (9) revealed, the selection depends on two factors: one is the credit value of operators ( $\hat{q}_{i,t}$ ) and the other is the usage number of operators (the part behind parameter  $C$ ). The parameter  $C$  plays a crucial role in deciding which factor plays a more important role. SIMAB [24] uses a sliding window with a mechanism of FIFO to store some latest information about operators. The latest information about operators truly reflects the operator performance. Due to the timeliness inadequacy of this accumulation, a decay factor is suggested in [16]. In [21], two modified MAB methods are proposed, which are MOEA/D-UCB-Tuned and MOEA/D-UCB-V. The two methods use a parameter called the rewards' variance to modify (9), and the experimental results show that UCB-T performs better than V on most test problems.

Except for the two methods mentioned above, there are some other kinds of adaptive operator selection methods, such as gradient based methods and multiple trial vector comparison based methods. Schütze et al. [43] propose a local search mechanism, Hypervolume Directed Search (HVDS). In HVDS, the gradient information is used to select search behaviors which are greedy search, descent direction, and search along the Pareto front. As these search behaviors are based on gradients, these methods cannot be used if objectives are not differentiable. Lara et al. [2] suggest a novel iterative search procedure, Hill Climber with Sidestep (HCS), in which it is capable of moving both toward and along the Pareto set depending on the distance of the current iterate toward this set. The search direction selection is on the basis of the dominance relation between several trial vectors and the old individual.

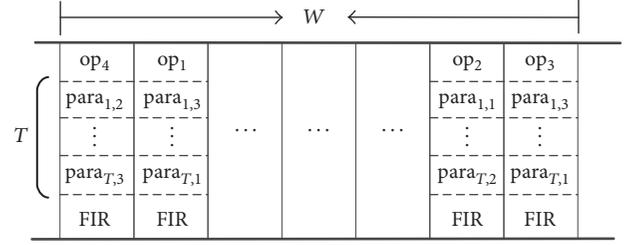


FIGURE 1: An illustration of sliding window in the way of FIFO.

### 3. The Proposed Algorithm

In this section, we present an improved bandit based method for MOPs, named latest stored information based adaptive selection strategy. This method attaches more attention to the AOS dynamic nature. It is mainly composed of two parts, credit assignment and operator selection.

**3.1. Credit Assignment.** Credit assignment contains two main tasks: one is to calculate credit value of applied operators; the other is to assign the credit fairly.

For the first task, fitness change is adopted as rewards of successful applied operators, which is regarded as Fitness Improvement Rate (FIR). During different search processes, the convergence levels of individuals are highly different. Normalization is used for FIR as follows:

$$\text{FIR} = \frac{\text{old}f - \text{new}f}{\text{old}f}, \quad (10)$$

where  $\text{old}f$  is the fitness value of the solution of last generation on the subproblem and  $\text{new}f$  is the fitness value of the current solution on the subproblem.

A sliding window with length  $W$  is used to store operators, their assist information, and FIRs with the mechanism of FIFO. It always stores the latest  $W$  configuration of operators and their related information.

Supposing that the operator number is  $K_0$ , the type number of assist information on operator is  $T$ , and the number of each type of assist information on operator is  $K_t$ ,  $t \in \{1, \dots, T\}$ . The structure of sliding window is shown as in Figure 1. As the sliding window revealed, the first layer is stored by operator names, the last is FIRs, and the middle layers are for parameters. It is worth noting that the locations of different types of parameters in the sliding window are ranked according to their significance, and this order is also the computation order of assist information on operator. Since the best suitable operator and the best suitable configuration of operator and assist information on operator are considered simultaneously, the credit value to operator is first to be assigned and the configuration of operator and assist information is considered later. The configurations among assist information on operator are not taken into consideration. The index of FIR is not described in Figure 1. This is mainly because the index of FIR is mainly associated with the credit assignment of different operators and different types of assist information on operator. The details of the index of FIR are illustrated in Figure 2.

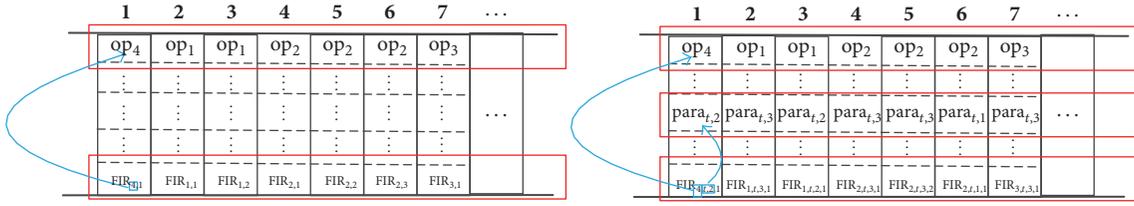


FIGURE 2: An illustration of the index of FIR in credit value computation.

As rewards guide operator selection methods, some unexpected extreme fitness improvement values brought by operators can appear. In case of that, we discard 5% best rewards and 5% worst rewards, and the rest of the rewards are denoted as  $R$ .

$$\begin{aligned} \text{FRR\_op}_i &= \frac{\sum_j^{|R_i|} \text{FIR}_{i,j}}{\sum_i^{K_0} \sum_j^{|R_i|} \text{FIR}_{i,j}}, \\ \text{FRR\_para}_{i,k_t} &= \frac{\sum_j^{|R_{i,k_t}|} \text{FIR}_{i,t,k_t,j}}{\sum_{k_t}^{K_t} \sum_j^{|R_{i,k_t}|} \text{FIR}_{i,t,k_t,j}}, \end{aligned} \quad (11)$$

where  $\text{FRR\_op}_i$  denotes credit value of operator  $i$  and  $\text{FRR\_para}_{i,k_t}$  is credit value of a configuration of operator  $i$  and  $k_t$ th parameter of parameter type  $t$ ;  $\text{FIR}_{i,t,k_t,j}$  denotes  $j$ th reward of operator  $i$  and  $k_t$ th parameter of parameter type  $t$  which belongs to  $R_{i,k_t}$ , and  $\text{FIR}_{i,j}$  denotes  $j$ th reward of operator  $i$  which belongs to  $R_j$ .  $|\cdot|$  denotes the cardinality of a set. But if the total usage number of the operator or the assist information is small, this method is not very useful. So this method is used under the condition that the total usage number is no less than 20.

**3.2. Operator Selection.** Based on the credit assignment described above, each configuration of operator and parameters gets its FRR. Then the best configuration will be selected. Refer to MAB algorithms; the selection of operators and parameters are defined as

$$\begin{aligned} \text{op} &= \arg \max_{i=\{1,\dots,K_0\}} \left( \text{FRR\_op}_i + C \times \sqrt{\frac{2 \times \ln \sum_{j=1}^{K_0} n_j}{n_i}} \right), \\ \text{para}_t &= \arg \max_{l=\{1,\dots,K_t\}} \left( \text{FRR\_para}_{l,k_t} + C \right. \\ &\quad \left. \times \sqrt{\frac{2 \times \ln \sum_{l=1}^{K_t} n_{t,l}}{n_{t,k_t}}} \right), \end{aligned} \quad (12)$$

where  $n_i$  is the successful applied number of  $i$ th operator,  $n_{t,l}$  is the successful applied number of  $j$ th parameter of parameter type  $t$ , and  $K$  is the number of operators or this type of parameters.

As (12) revealed, the selection depends on two factors. One is the credit value of operators ( $\text{FRR\_op}$ ) and the other is the usage number of operators (the part behind parameter  $C$ ). As the parameter  $C$  plays a crucial role in deciding which factor plays a more important role, we decide to redefine the parameter  $C$  with the difference between the best usage rate and the worst usage rate. The advantage of such a parameter  $C$  is that the selection of operators and parameters becomes sensitive. If the difference is a large value, usage number is the major factor. The configuration of the operator and parameter possessing maximum usage numbers has more possibility of giving full play to its search power. If the difference is a small value, the credit value is more important and the configuration with power search ability at this time will be selected more possibly by chance. The parameter  $C$  is defined as

$$C = \frac{3}{1 + \exp(-15 * ((n_{\max} - n_{\min}) / W) - 0.25)} + 2, \quad (13)$$

where  $n_{\max}$  and  $n_{\min}$  are the maximum and minimum usage numbers, respectively and  $W$  is the length of sliding window, and it also could be regarded as the total usage number.

The pseudocode of operator and parameter selection is given in Algorithm 1.

## 4. Integration of LSIAS with MOEA/D-DRA

The MOEA/D provides a decomposition method which could decompose a MOP into a series of single problems. It is the single problems that could give an exact value to measure the performance of every operator. Therefore, the main reason we choose MOEA/D is that the metrics for evaluating operators are easy to get.

Since the MOEA/D was presented in 2007, a good deal of research has been put forward to improve its performance and its applied range. Any improvement on MOEA/D could be of practical interests. Here we choose a famous improved version of MOEA/D as the framework, MOEA/D-DRA, as it is the champion of the CEC 2009 MOEA contest. Consequently, we investigate how to enhance the MOEA/D with LSIAS.

**4.1. MOEA/D-DRA.** In this paper, we use Tchebycheff with objective normalization instead of classic Tchebycheff. It is known that the objective normalization performs better than classic methods especially when the objective space becomes more complex.

```

Initialize latest stored information sliding window
If there are operators that have not been selected
    Select an operator randomly from the operator pool.
Else
    Calculating FRR_op according to Eq. (11).
    Calculating C according to Eq. (13).
    Get the operator according to Eq. (12).
End
For  $t = 1$  : parameter_typenumber
    If there is a parameter of this type that has not been combined with the
    selected operator and parameters, which are more important than the current type of
    parameter.
        Randomly select a parameter of this type;
    Else
        Calculating FRR_para $t, k_t$  according to Eq. (11).
        Get the suitable parameter  $k_t$  of type  $t$  according to Eq. (12).
    End
End

```

ALGORITHM 1: The procedure for the adaptive selection.

MOEA/D minimizes all these  $N$  objective functions simultaneously in a single run. Neighborhood relations among these single objective subproblems are defined based on the distances among their weight vectors. Each subproblem is optimized by using information mainly from its neighboring subproblems. For most versions of MOEA/D proposed, they receive about the same amount of computational effort. These subproblems, however, may have different computational difficulties; therefore, it is very reasonable to assign different amounts of computational effort to different problems. In MOEA/D-DRA,  $N/5$  subproblems are selected to be optimized based on the utility in a single run. We define and compute a utility  $\pi^i$  for each subproblem  $i$ . Computational efforts are distributed to these subproblems based on their utilities.  $\Delta^i$  is the relative improvement of subproblem  $i$  and its value is a positive number. If the solution of subproblem  $i$  is better than the one in last generation, the value of  $\pi^i$  is 1. But if the solution of subproblem  $i$  is not better than the one in last generation, the value of  $\pi^i$  is less than 1. Thus this subproblem is selected to be optimized with a large probability in next generation.

Suppose that the MOP is decomposed into  $N$  scalar subproblems, and each subproblem has a weight vector which is uniformly spread. In  $\Lambda = (\lambda^1, \dots, \lambda^N)^T$ , each  $\lambda^j = (\lambda_1^j, \dots, \lambda_m^j)^T$  matches the condition of  $\sum_{i=1}^m \lambda_i^j = 1$  and  $\lambda_i^j > 0$ . The objective function of  $j$ th subproblem can be stated as (2). When searching, MOEA/D-DRA maintains

- (i) a population of  $N$  points  $x^1, \dots, x^N \in \Omega$ , where  $x^i$  is the current solution to the  $i$ th subproblem;
- (ii)  $FV^1, \dots, FV^N$ , where  $FV^i$  is the  $F$ -value of  $x^i$ ; that is,  $FV^i = F(x^i)$  for each  $i = 1, \dots, N$ ;
- (iii)  $\pi^i$ : the utility of  $i$ th subproblem, which measures the improvement of the individual  $x^i$  between the

previous and the current generation. It is defined as

$$\pi^i = \begin{cases} 1, & \text{if } \Delta^i > 0, \\ \left(0.95 + 0.05 \times \frac{\Delta^i}{0.001}\right) \times \pi^i, & \text{otherwise,} \end{cases} \quad (14)$$

where  $\Delta^i$  is the relative improvement of subproblem  $i$ .

An important advantage of MOEA/D and its improved versions is that a better solution  $x^i$  is helpful for both the  $i$ th subproblem and the subproblems close to the  $i$ th. There is a congruent relationship between subproblems and vectors. Thus, each solution converges along its vector from beginning to end, and the excellent ones could bring valuable information to help its neighbor individuals evolve. Each subproblem has  $T$  neighborhoods which are selected based on the Euclidean distance. For generation  $t$ , the process is as follows:

- (1) Select  $T$  neighborhoods or all population as  $P$ ;
- (2) Randomly select several solution from  $P$ ;
- (3) Generate a new solution after using genetic operators;
- (4) Compare the new one with  $n_r$  old ones selected from  $P$  and replace them if the new one is better.

**4.2. MOEA/D-DRA with LSIAS.** To combine the LSIAS with MOEA/D-DRA, we set the content of latest stored information and define the reward calculation. In this paper, we choose four kinds of operator related information as the content of latest stored information, which are an operator pool, the scaling factor  $F$ , the neighborhood size, and the FIRs. They are stored in the sliding window as shown in Figure 3. The operator pool consists of four DE operators, expressed as  $op_i$  ( $i = 1, 2, 3, 4$ ).

1		2		...		W-1		W	
op <sub>4</sub>	op <sub>1</sub>					op <sub>2</sub>	op <sub>3</sub>		
ne <sub>1</sub>	ne <sub>2</sub>					ne <sub>3</sub>	ne <sub>3</sub>		
F <sub>4</sub>	F <sub>1</sub>					F <sub>2</sub>	F <sub>3</sub>		
FIR	FIR					FIR	FIR		

FIGURE 3: An illustration of latest stored information in the sliding window.

- (i) op<sub>1</sub>: DE/rand/1
- (ii) op<sub>2</sub>: DE/rand/2
- (iii) op<sub>3</sub>: DE/target-to-rand/1
- (iv) op<sub>4</sub>: DE/target-to-rand/2.

In MOEA/D-DRA, each vector  $\lambda$  employs  $T$  nearest vectors as its neighborhood. In this paper, suppose that all vectors are divided into several neighborhoods with the size of  $T$ . Choosing three nearest neighborhoods for  $\lambda$ , it means that each subproblem has three neighborhoods as its neighborhood pool to provide donor vectors for mutation. But just only one neighborhood can be used for every mutation.

The scaling factor  $F$  is used for the adaptive parameter control of DE operators. Because of the different search power of the four, four independent scaling factors are set. Every  $F$  follows a Cauchy distribution with location parameter  $F'$  and scale parameter  $\alpha_i$  ( $i = 1, 2, 3, 4$ ). If a new  $F$  is out of the range  $[0, 1]$ , it will be regenerated.

$$F_i = \text{Cauchy}(F', \alpha_i), \quad (15)$$

$$F'_i = w_F \times F'_i + (1 - w_F) \times \text{mean}_{\text{POW}}(F_i^{\text{success}}), \quad (16)$$

where  $w_F$  is a weight factor. It is validated that a little random perturbation of  $w_F$  can bring better effectiveness for the optimization process [17]. As a result, it is randomly generated within the range  $[0.8, 1]$ .  $F'_i$  is updated as (16) and it is initialized to be 0.5.  $F_i^{\text{success}}$  is the set of the scaling factor of  $i$ th operator which is stored in the sliding window.  $\text{mean}_{\text{POW}}(F_i^{\text{success}})$  is given as follows:

$$\text{mean}_{\text{POW}}(F_i^{\text{success}}) := \sum_{x \in F_i^{\text{success}}} \left( \frac{x^k}{|F_i^{\text{success}}|} \right)^{1/k}, \quad (17)$$

where  $|F_i^{\text{success}}|$  is the number of  $F_i^{\text{success}}$  in the sliding window.  $k$  is set to 1.5 which is proved to be the best in [36].

Assume that  $y$  is a new offspring for  $i$ th subproblem and  $x^i$  is the old one. If the new one is better than the old one, the former replaces the latter. The reward  $\Delta^i$  is the same as the relative improvement in [26], and it is given by

$$\Delta^i = \frac{g^{\text{te}}(x^i | \lambda^i, z^*) - g^{\text{te}}(y | \lambda^i, z^*)}{g^{\text{te}}(x^i | \lambda^i, z^*)}. \quad (18)$$

In MOEA/D-DRA, each new offspring is compared with  $n_r$  solutions selected from the neighborhood; thus the

$\text{FIR}_{\text{op,para}}$  is the sum of  $\Delta^i$  brought by  $y$  with the current operator and the current parameters.

The pseudocode of MOEA/D-LSIAS is demonstrated in Algorithm 2.

## 5. Experimental Studies

In this section, some experiments are adopted to analyze the performance of our algorithm. In Section 5.1, 22 well-known benchmark problems are briefly introduced in the experiments. Two performance measures and all parameter settings about the experiments are described, respectively, in Sections 5.2 and 5.3. Furthermore, some comparative experiments between MOEA/D-LSIAS and other state-of-the-art MOEA/D variants are given and analyzed in Section 5.4. Additionally, some experiments about parameters are conducted to analyze the effectiveness of parameters with different numbers in Sections 5.5 and 5.6.

**5.1. Benchmark Functions.** Three types of benchmark functions are adopted here to manifest the effectiveness and robustness of MOEA/D-LSIAS. There are 22 well-known benchmark problems in total including ZDT problems [28], DTLZ problems [29], and UF problems [30]. The solution set of these benchmark problems (the Pareto set) is not given by a single point but forms a  $(m - 1)$ -dimensional object, where  $m$  is the number of objectives involved in the MOP.

The most widely used ZDT problems are biobjective test problems, including ZDT1–4 and ZDT6. This kind of problems is easy to solve as it lacks some features, such as variable linkage and multimodality. Therefore, UF problems and DTLZ problems are covered in the experiments, of which various features can make up for the deficiency of ZDT problems. It is worth noting that UF1–7 are all biobjective test problems and DTLZ1–7 and UF8–10 are three-objective test problems. All the three types of benchmark functions are widely used for the evaluation of MOEAs.

**5.2. Performance Measure.** There are many performance measures for comparisons among algorithms, like Inverted Generational Distance (IGD) [44], Hypervolume (HV) [45], Spread [46],  $\Delta_p$  [47], and so on. Because all comparative algorithms in their literature employ either IGD or HV or both of them, they are also chosen to assess the performance of the proposed algorithm and its comparisons. The two metrics possess the ability of convergence assessment and diversity assessment.

**IGD.** Let  $P^*$  be a set of solutions distributed in the true Pareto-optimal front, which is a sphere composed of all best solutions. Let  $P$  be an approximation set got by MOEA. The IGD metrics of  $P$  is obtained by

$$\text{IGD}(P, P^*) = \frac{\sum_{x \in P^*} \text{dist}(x, P)}{|P^*|}, \quad (19)$$

where  $\text{dist}(x, P)$  is the minimum Euclidean distance between  $x$  and its nearest solution in  $P$ , and  $|P^*|$  is the number of elements in  $P^*$ . The true Pareto-optimal front set

```

Initialize the population  $\{x^1, \dots, x^N\}$ ,  $\Lambda = (\lambda^1, \dots, \lambda^N)^T$  and  $z^* = (z_1^*, \dots, z_m^*)^T$ .
Set Slwindow( $i$ )( $j$ ) = 0 for each  $i = 1, 2, 3, 4$  and  $j = 1, 2, \dots, W$ .
For  $i = 1 : N$ 
  For  $ne = 1 : 3$ 
     $B(i, ne) = (i_{T \times (ne-1)+1}, i_{T \times (ne-1)+2}, \dots, i_{T \times (ne-1)+T})$ ;
  End
   $\pi^i = 1$ ; Where  $i = 1, 2, \dots, N, i_1, i_2, \dots, i_{3 * T}$  are  $3 * T$  vectors closest to  $i$ th vector.
End
While stopping criterion is not satisfied
  Select  $\lfloor N/5 \rfloor$  individuals by 10-tournament selection based on  $\pi^i$  to form  $I$ .
  For  $i = 1 : |I|$ 
     $op = \text{OperatorSelection}(\text{Slwindow})$ ; (Algorithm 1)
    if  $\text{rand} < \delta$ 
       $ne = \text{NeighborhoodSelection}(\text{Slwindow})$ ; (Algorithm 1)
       $P = ne$ ;
    else
       $P = \{1, \dots, N\}$ 
    end
     $F_{op} = \text{ScaleSelection}(\text{Slwindow})$ ; Eq. ((15)–(17))
     $y = \text{Mutation}(op, P, F_{op})$ ; Eq. ((3)–(6))
    update the current reference point  $z^*$ 
     $c = 0$ ;
    while  $c < n_r$ 
      randomly select a solution  $x^j$  from  $P$ ;
       $\Delta^j = \frac{g^{te}(x^j | \lambda^j, z^*) - g^{te}(y | \lambda^j, z^*)}{g^{te}(x^j | \lambda^j, z^*)}$ 
      If  $\Delta^j > 0$ 
        Replace  $x^j$  with  $y$ ;
        Remove  $x^j$  from  $P$ ;
         $\text{FIR}_{op,ne} = \text{FIR}_{op,ne} + \Delta^j$ ;
      End
       $c++$ ;
    end
    UpdateSlwindow( $op, ne, F, \text{FRR}$ );
    If  $\text{mod}(\text{gen}, 50) == 0$ 
      UpdatePi( $\pi$ );
    End
  End
End

```

ALGORITHM 2: MOEA/D-LSIAS.

is known in advance and it can be got in <http://jmetal.sourceforge.net/problems.html>. In this paper, we, respectively, select 1,000 and 10,000 points uniformly distributed in the true Pareto-optimal front for biobjective and three-objective test problems.

*HV*. Assume that  $z^r = (z_1^r, \dots, z_m^r)^T$  is the reference point dominated by all the points in  $P^*$ . *HV* metrics is the size of the objective space dominated by the approximate set from  $P$  and bounded by  $z^r$ .

$$\begin{aligned}
 &HV(S) \\
 &= \text{Vol} \left( \bigcup_{x \in S} [f_1(x), z_1^r] \times \dots \times [f_m(x), z_m^r] \right), \quad (20)
 \end{aligned}$$

where  $\text{Vol}(\cdot)$  indicates the Lebesgue measure. It is noted that if a solution in  $P$  does not dominate  $z^r$ , it will be abandoned. In this paper,  $z^r$  is set to (2.0, 2.0) for ZDT1–ZDT4, ZDT6, and UF1–UF7, (1.0, 1.0, 1.0) for DTLZ1, (2.0, 2.0, 2.0) for DTLZ2–DTLZ6 and UF8–UF10, and (2.0, 2.0, 2.0) for DTLZ7.

*5.3. Experimental Settings*. In this paper, the proposed algorithm, MOEA/D-LSIAS, is compared with various MOEA/D improved versions, including MOEA/D-DE, MOEA/D-DRA, MOEA/D-FRRMAB, MOEA/D-STM, MOEA/D-UCB-T, and MOEA/D-ARG. All the basic experimental settings for the above competitive algorithms are shown in Table 1. Three kinds of benchmark functions are adopted, and the population size  $N$  and the maximum number of function

TABLE 1: The basic experimental parameter settings of the compared algorithms.

MOEA/D-DE	$F = 0.5, CR = 1.0, n_r = 2, \delta = 0.9, T = 30$
MOEA/D-DRA	$F = 0.5, CR = 1.0, n_r = 2, \delta = 0.9, T = 30$
MOEA/D-STM	$F = 0.5, CR = 1.0, n_r = 2, \delta = 0.9, T = 30$
MOEA/D-FRRMAB	$F = 0.5, CR = 1.0, K = 0.5, C = 5.0, W = 0.5 \times N, D = 1.0, n_r = 2, C = 5, \delta = 0.9, T = 30$
MOEA/D-UCB-T	$F = 0.5, CR = 1.0, K = 0.5, C = 5.0, W = 0.5 \times N, D = 1.0, n_r = 2, C = 5, \delta = 0.9, T = 30$
MOEA/D-ARG	$F = 0.5, CR = 1.0, K = 0.5, C = 5.0, n_r = 2, C = 5, \delta = 0.9, T = 30$
MOEA/D-LSIAS	$CR = 1.0, K = 0.5, W = 0.5 \times N, n_r = 2, \delta = 0.9, T = 30$

TABLE 2: Comparative results of all the compared algorithms on the ZDT test problems regarding IGD.

Problems	Algorithms						
	MOEA/D-DE	MOEA/D-DRA	MOEA/D-FRRMAB	MOEA/D-STM	MOEA/D-UCB-T	MOEA/D-ARG	MOEA/D-LSIAS
<b>ZDT1</b>							
Mean	6.375E-02	9.010E-03	1.021E-02	8.680E-03	1.059E-02	3.081E-02	<b>7.681E-03</b>
IQR	5.070E-04	2.537E-04	1.168E-04	5.558E-04	7.436E-04	5.785E-03	<b>4.594E-04</b>
Rank	7	3	4	2	5	6	<b>1</b>
<b>ZDT2</b>							
Mean	1.057E-02	8.514E-03	2.845E-02	9.654E-03	2.641E-02	6.074E-03	<b>6.070E-03</b>
IQR	4.235E-03	1.001E-03	8.366E-03	1.548E-03	2.365E-03	1.457E-03	<b>6.285E-05</b>
Rank	5	3	7	4	6	2≈	<b>1</b>
<b>ZDT3</b>							
Mean	7.352E-02	2.543E-02	8.548E-03	4.691E-02	<b>8.512E-03</b>	1.258E-02	9.109E-03
IQR	1.800E-03	4.530E-03	1.173E-03	2.268E-03	<b>9.564E-04</b>	5.576E-04	3.832E-04
Rank	7	5	2	6	<b>1</b>	4	3
<b>ZDT4</b>							
Mean	4.774E-02	6.098E-02	3.158E-02	<b>7.636E-03</b>	3.048E-02	7.950E-03	8.733E-03
IQR	2.968E-02	1.139E-03	4.698E-03	<b>2.364E-03</b>	4.454E-03	1.319E-03	7.935E-04
Rank	6	7	5	<b>1</b>	4	2	3
<b>ZDT6</b>							
Mean	<b>1.437E-03</b>	8.326E-03	1.912E-03	5.244E-03	1.848E-03	1.497E-03	1.698E-03
IQR	<b>1.320E-05</b>	2.415E-05	5.322E-05	8.236E-04	6.875E-04	1.764E-04	9.720E-05
Rank	<b>1</b>	7	5	6	4	2	3
Rank Sum	26	25	23	19	20	16	11
+/-/≈	1/4/0	0/5/0	1/4/0	1/4/0	1/4/0	2/2/1	

“+,” “-,” and “≈,” respectively, indicate the result of the algorithm is better than, worse than, and similar to that of MOEA/D-LSIAS based on the Wilcoxon’s rank sum test with a significance  $\alpha = 0.05$ .

evaluations for each are set differently.  $N$  is set to 100 for all ZDT benchmark problems.  $N$  is set to 300 for all DTLZ problems and all UF problems which are three-objective problems. The maximum number of function evaluations is set to 25,000 for ZDT problems and 300,000 for UF problems and DTLZ problems. The parameter settings of MOEA/D-LSIAS are as follows. The control parameters CR and C and the sliding window size  $W$  are shown in Table 1. The operator parameters are explained in Section 4 and the neighborhood size is set to 20. The mean value and the interquartile range (IQR) are used to measure all results of every test instance run 30 independent times. Furthermore, Wilcoxon’s rank sum test is employed to assess the statistical significance of all experimental results from the compared algorithms with a significance level  $\alpha = 0.05$ .

#### 5.4. Comparisons of MOEA/D-LSIAS with State-of-the-Art MOEA/D Variants

5.4.1. Comparisons on the ZDT Test Problems. Tables 2 and 3 provide the comparative experiment results of all the competitive algorithms on the ZDT test problems, which show the mean value and the interquartile range of IGD and HV over 30 independent runs, respectively. The best result for each test problem is highlighted in boldface with regard to the metric in Tables 2–7.

In Table 2, MOEA/D-LSIAS performs best on ZDT1 and ZDT2, while MOEA/D-UCB-T, MOEA/D-STM, and MOEA/D-DE are, respectively, the best on ZDT3, ZDT4, and ZDT6. Moreover, the Wilcoxon’s rank sum test indicates that MOEA/D-LSIAS is similar to MOEA/D-ARG on ZDT2

TABLE 3: Comparative results of all the compared algorithms on the ZDT test problems regarding HV.

Problems	Algorithms						
	MOEA/D-DE	MOEA/D-DRA	MOEA/D-FRRMAB	MOEA/D-STM	MOEA/D-UCB-T	MOEA/D-ARG	MOEA/D-LSIAS
<b>ZDT1</b>							
Mean	3.6251	3.6519	3.6466	3.6523	3.6471	3.6219	<b>3.6587</b>
IQR	1.853E-03	3.926E-03	7.616E-03	3.014E-03	1.037E-03	5.324E-03	<b>4.959E-03</b>
Rank	6	3	5	2	4	7	<b>1</b>
<b>ZDT2</b>							
Mean	3.2992	3.3181	3.2892	3.3107	3.2826	<b>3.3268</b>	3.3231
IQR	3.699E-03	1.176E-03	1.201E-03	9.140E-03	4.450E-03	<b>6.335E-03</b>	2.067E-04
Rank	5	3	6	4	7	<b>1</b>	2
<b>ZDT3</b>							
Mean	4.7789	4.7713	<b>4.8087</b>	4.7865	4.8063	4.7897	4.7995
IQR	6.202E-02	5.878E-03	<b>3.742E-03</b>	5.143E-03	4.715E-03	2.229E-03	8.258E-03
Rank	6	7	<b>1</b>	5	2	4	3
<b>ZDT4</b>							
Mean	3.6201	3.6054	3.6258	3.6432	3.6204	<b>3.6456</b>	3.6389
IQR	8.918E-02	4.392E-03	3.156E-03	3.332E-03	1.142E-02	<b>5.560E-03</b>	5.375E-03
Rank	6	7	4	2	5	<b>1</b>	3
<b>ZDT6</b>							
Mean	3.0420	3.0294	3.0367	3.0313	3.0375	3.0379	<b>3.0426</b>
IQR	8.849E-04	1.088E-04	4.135E-04	6.398E-04	2.256E-04	5.187E-04	<b>3.251E-04</b>
Rank	2 $\approx$	7	5	6	4	3	<b>1</b>
Rank Sum	25	27	21	19	22	16	10
+/-/ $\approx$	0/4/1	0/5/0	1/4/0	1/4/0	1/4/0	2/3/0	

“+,” “-,” and “ $\approx$ ,” respectively, indicate the result of the algorithm is better than, worse than, and similar to that of MOEA/D-LSIAS based on the Wilcoxon’s rank sum test with a significance  $\alpha = 0.05$ .

and to MOEA/D on ZDT6. These comparison results for the compared algorithms on ZDT test problems are summarized in the last two rows. The “+/-/ $\approx$ ” summarizes the competitive results on ZDT test problems regarding IGD. It can be observed that the compared algorithms except MOEA/D-DRA perform better than MOEA/D-LSIAS at least once. But the “Rank Sum” indicates that MOEA/D-LSIAS gets the first rank, and MOEA/D-ARG is the best among all the compared algorithms.

Table 3 provides the comparative results using HV. It is easy to find that MOEA/D-LSIAS gets the best performance on ZDT1 and ZDT6, while MOEA/D-ARG is the best on ZDT2 and ZDT4. MOEA/D-FRRMAB performs best only on ZDT3. The Wilcoxon’s rank sum test indicates that MOEA/D-LSIAS is similar to MOEA/D-DE on ZDT6. It is worth noting that MOEA/D-UCB-T and MOEA/D-FRRMAB perform similarly on most ZDT test problems. “+/-/ $\approx$ ” shows that MOEA/D-LSIAS performs better or in a similar way on over half of the ZDT test problems. The “Rank Sum” also shows that MOEA/D-LSIAS is the best. The advantages of MOEA/D-LSIAS are further confirmed by using HV.

**5.4.2. Comparisons on the DTLZ Test Problems.** DTLZ test problems are three-objective optimization problems, and they are obviously harder than ZDT test problems. Tables 4

and 5 are suggestive of the fact that MOEA/D-LSIAS gets the best performance.

Table 4 gives comparative results of all the compared algorithms on the DTLZ test problems regarding IGD. It is straightforward to find that MOEA/D-LSIAS performs better on more than half of the DTLZ test problems. MOEA/D-LSIAS performs best on 4 (i.e., DTLZ1, DTLZ3, DTLZ5, and DTLZ6) out of 7 DTLZ test problems. “+/-/ $\approx$ ” indicates that MOEA/D-LSIAS performs better than MOEA/D-DE, MOEA/D-DRA, MOEA/D-FRRMAB, and MOEA/D-STM on 6, 6, 7, 4, 7, and 5 out of 7 DTLZ test problems. The Wilcoxon’s rank sum test shows that MOEA/D-LSIAS performs similarly to MOEA/D-STM on DTLZ5. As observed from “Rank Sum,” MOEA/D-LSIAS is evidently the best one for the IGD metrics on DTLZ test problems.

In Table 5, the results of DTLZ test problems with regard to HV are provided. MOEA/D-LSIAS gets the best result on DTLZ1, DTLZ4, and DTLZ7, and it is worse than MOEA/D-STM on DTLZ2, worse than MOEA/D-ARG on DTLZ3 and DTLZ5, and worse than MOEA/D-UCB-T and MOEA/D-FRRMAB on DTLZ6. It achieves statistically similar results to MOEA/D-DRA on DTLZ1 and to MOEA/D-STM on DTLZ5 and DTLZ7. It is reasonable to conclude that MOEA/D-LSIAS achieves a superior performance on most of DTLZ test problems on account of the IGD and HV metric.

TABLE 4: Comparative results of all the compared algorithms on the DTLZ test problems regarding IGD.

Problems	Algorithms						
	MOEA/D-DE	MOEA/D-DRA	MOEA/D-FRRMAB	MOEA/D-STM	MOEA/D-UCB-T	MOEA/D-ARG	MOEA/D-LSIAS
DTLZ1							
Mean	1.429E-02	9.102E-03	9.854E-03	9.366E-03	9.853E-03	9.174E-03	<b>7.736E-03</b>
IQR	1.654E-03	2.801E-03	7.091E-04	2.720E-03	8.649E-04	1.235E-03	<b>7.318E-04</b>
Rank	7	2	6	4	5	3	<b>1</b>
DTLZ2							
Mean	2.624E-02	2.697E-02	3.887E-02	2.801E-02	3.827E-02	<b>1.023E-02</b>	2.684E-02
IQR	2.771E-04	6.600E-04	3.953E-03	1.180E-03	8.697E-03	<b>1.658E-04</b>	8.294E-04
Rank	2	3	7	5	6	<b>1</b>	4
DTLZ3							
Mean	8.947E-02	2.772E-02	2.154E-02	1.258E-02	2.137E-02	9.823E-03	<b>8.481E-03</b>
IQR	4.235E-04	8.906E-04	5.322E-03	2.340E-03	4.842E-03	8.248E-03	<b>1.845E-04</b>
Rank	7	6	5	3	4	2	<b>1</b>
DTLZ4							
Mean	5.979E-02	4.944E-02	9.127E-03	<b>8.522E-03</b>	9.108E-03	1.684E-02	8.724E-03
IQR	7.640E-03	2.602E-04	1.831E-04	<b>2.170E-03</b>	3.185E-04	7.549E-03	1.855E-04
Rank	7	6	4	<b>1</b>	3	5	2
DTLZ5							
Mean	1.147E-03	1.616E-03	8.349E-04	6.911E-04	8.515E-04	7.095E-04	<b>6.907E-04</b>
IQR	9.262E-05	1.273E-04	2.227E-05	9.740E-05	3.128E-05	2.227E-04	<b>1.830E-04</b>
Rank	6	7	4	2 $\approx$	5	3	<b>1</b>
DTLZ6							
Mean	3.030E-03	2.498E-03	8.938E-04	1.862E-03	8.917E-04	1.639E-03	<b>7.379E-04</b>
IQR	4.281E-04	1.089E-04	1.656E-05	2.540E-04	1.272E-04	1.297E-04	<b>8.716E-05</b>
Rank	7	6	3	5	2	4	<b>1</b>
DTLZ7							
Mean	1.853E-01	2.341E-01	3.892E-02	<b>1.490E-02</b>	3.992E-02	2.109E-02	3.169E-02
IQR	2.443E-02	3.167E-02	2.497E-03	<b>5.865E-03</b>	3.254E-03	2.497E-03	2.484E-03
Rank	6	7	4	<b>1</b>	5	2	3
Rank Sum	42	37	33	21	30	20	13
+/-/ $\approx$	1/6/0	1/6/0	0/7/0	2/4/1	0/7/0	2/5/0	

“+,” “-,” and “ $\approx$ ,” respectively, indicate the result of the algorithm is better than, worse than, and similar to that of MOEA/D-LSIAS based on the Wilcoxon’s rank sum test with a significance  $\alpha = 0.05$ .

5.4.3. *Comparisons on the UF Test Problems.* Table 6 lists compared results on the UF problems with regard to IGD, which possesses a more complicated Pareto optimal set than ZDT and DTLZ test problems. It is observed that MOEA/D-LSIAS performs best on 5 (i.e., UF1-UF2, UF4, UF6, and UF10) out of 10 UF test problems, while MOEA/D-ARG performs best on 3 (i.e., UF3, UF5, and UF7). MOEA/D-STM achieves best results on UF8 and UF9. The Wilcoxon’s rank sum test indicates that MOEA/D-LSIAS performs similarly to MOEA/D-STM on UF1 and to MOEA/D-DRA on UF10. The “+/-/ $\approx$ ” and the “Rank Sum” show that MOEA/D-ARG and MOEA/D-STM are serious rivals to MOEA/D-LSIAS. But it is evident that MOEA/D-LSIAS performs best when comparing UF test problems with regard to IGD.

Table 7 shows the results of the HV metrics for MOEA/D-LSIAS and various versions of MOEA/D. MOEA/D-LSIAS also performs best on 5 (i.e., UF2-4 and UF6-7) out of 10

UF test problems. In more detail, MOEA/D-LSIAS is worse than MOEA/D-STM on UF1, UF5, UF8, and UF9, worse than MOEA/D-ARG on UF9 and UF10, and worse than MOEA/D-FRRMAB on UF8. Moreover, the Wilcoxon’s rank sum test reveals that MOEA/D-LSIAS performs similarly to MOEA/D-ARG on UF4. As summarized in the “Rank Sum,” MOEA/D-LSIAS is better than MOEA/D-DE and MOEA/D-DRA on all the UF test problems. Regarding the comparison with MOEA/D-FRRMAB, MOEA/D-STM, MOEA/D-UCB-T, and MOEA/D-ARG, MOEA/D-LSIAS performs better or in a similar way on more than half of the UF test problems. Therefore, it is confirmed that MOEA/D-LSIAS possesses a better performance than all the compared algorithms by using HV.

5.5. *MOEA/D-LSIAS versus AOSs with Different Assist Information.* To investigate the effectiveness of different types

TABLE 5: Comparative results of all the compared algorithms on the DTLZ test problems regarding HV.

Problems	Algorithms						
	MOEA/D-DE	MOEA/D-DRA	MOEA/D-FRRMAB	MOEA/D-STM	MOEA/D-UCB-T	MOEA/D-ARG	MOEA/D-LSIAS
DTLZ1							
Mean	0.9556	0.9689	0.9597	0.9644	0.9598	0.9641	<b>0.9691</b>
IQR	3.852E-03	6.179E-03	2.155E-04	2.015E-03	7.257E-04	3.788E-03	<b>1.185E-03</b>
Rank	7	2≈	6	4	5	3	<b>1</b>
DTLZ2							
Mean	7.3994	7.3902	7.3763	<b>7.4067</b>	7.3755	7.3808	7.4058
IQR	1.643E-03	4.724E-04	2.769E-03	<b>8.723E-03</b>	7.733E-03	1.110E-03	1.950E-03
Rank	3	4	6	<b>1</b>	7	5	2
DTLZ3							
Mean	7.3846	7.3897	7.3924	7.4046	7.3965	<b>7.4151</b>	7.4124
IQR	4.863E-03	2.709E-03	1.370E-03	1.265E-03	2.354E-03	<b>7.248E-03</b>	4.440E-03
Rank	7	6	5	3	4	<b>1</b>	2
DTLZ4							
Mean	7.3975	7.3924	7.4027	7.4118	7.4008	7.4103	<b>7.4172</b>
IQR	6.593E-03	3.856E-03	2.402E-03	6.670E-03	9.325E-03	1.438E-03	<b>5.897E-04</b>
Rank	6	7	4	2	5	3	<b>1</b>
DTLZ5							
Mean	6.0502	6.0433	6.0833	6.0908	6.0852	<b>6.0975</b>	6.0910
IQR	2.831E-04	2.101E-03	3.231E-04	2.308E-04	7.224E-04	<b>9.367E-04</b>	3.040E-04
Rank	6	7	5	3≈	4	<b>1</b>	2
DTLZ6							
Mean	6.0613	6.0737	6.1042	6.0985	<b>6.1043</b>	6.1007	6.1022
IQR	6.649E-04	1.983E-04	4.537E-04	2.630E-04	<b>3.773E-04</b>	7.362E-04	3.300E-04
Rank	7	6	2	5	<b>1</b>	4	3
DTLZ7							
Mean	13.224	13.299	13.303	13.453	13.310	13.371	<b>13.456</b>
IQR	2.890E-01	9.242E-02	8.599E-03	1.823E-02	7.318E-02	4.492E-03	<b>6.090E-03</b>
Rank	7	6	5	2≈	4	3	<b>1</b>
Rank Sum	43	38	33	20	30	20	12
+/-/≈	0/7/0	0/6/1	1/6/0	1/4/2	1/6/0	2/5/0	

“+,” “-,” and “≈,” respectively, indicate the result of the algorithm is better than, worse than, and similar to that of MOEA/D-LSIAS based on the Wilcoxon's rank sum test with a significance  $\alpha = 0.05$ .

of assist information, a further experiment about diverse configurations of operator information is given. In Section 3, two types of assist information are considered, including adaptive selection of neighborhood and adaptive control of factor  $F$ . The compared algorithms are the same as MOEA/D-LSIAS except the operator information selection, and they all employ AOS with four DE operators.

- (1) Method 1 (M1): AOS with no assist information,
- (2) Method 2 (M2): AOS with adaptive selection of neighborhood,
- (3) Method 3 (M3): AOS with adaptive control of factor  $F$ .

Figure 4 provides the four algorithm comparison results on ZDT1, DTLZ1, and UF1. For the IGD metrics, the ones

with assist information are better than AOS with nothing on ZDT1, but it is absolutely not on DTLZ1 and UF1. M2 and M3 perform worse than M1 on UF1, while MOEA/D-LSIAS is better than M1 on all the three test problems. It is concluded that the configuration of adaptive control of factor  $F$  and adaptive selection of neighborhood are helpful to improve the performance of AOS regarding IGD.

Regarding the HV metric, M2, M3, and MOEA/D-LSIAS work better than M1 on ZDT1. M1 performs better than M2 and M3 on DTLZ1 and UF1, but is absolutely beaten by LSIA. The results indicate that the assist information is not absolutely helpful for the AOS, but a good configuration of assist information is more useful. The improvement value depends on the configuration of assist information, and AOS with the help of different assist information can display different search power on different test problem.

TABLE 6: Comparative results of all the compared algorithms on the UF test problems regarding IGD.

Problems	Algorithms						
	MOEA/D-DE	MOEA/D-DRA	MOEA/D-FRRMAB	MOEA/D-STM	MOEA/D-UCB-T	MOEA/D-ARG	MOEA/D-LSIAS
UF1							
Mean	1.022E-02	9.293E-03	8.340E-03	7.505E-03	8.409E-03	8.038E-03	<b>7.502E-03</b>
IQR	8.278E-03	6.368E-03	1.485E-03	9.039E-04	1.989E-03	7.415E-04	<b>8.263E-04</b>
Rank	7	6	4	2≈	5	3	<b>1</b>
UF2							
Mean	9.867E-03	6.103E-03	2.646E-03	4.915E-03	2.541E-03	3.154E-03	<b>2.156E-03</b>
IQR	9.865E-04	1.241E-03	7.355E-04	4.895E-04	3.173E-04	2.065E-04	<b>4.727E-04</b>
Rank	7	6	3	5	2	4	<b>1</b>
UF3							
Mean	1.038E-02	7.838E-03	8.511E-03	9.003E-03	8.527E-03	<b>6.424E-03</b>	7.232E-03
IQR	7.802E-03	7.802E-04	5.156E-04	3.695E-04	9.717E-04	<b>1.162E-03</b>	3.066E-04
Rank	7	3	4	6	5	<b>1</b>	2
UF4							
Mean	9.927E-02	9.089E-02	3.238E-02	7.803E-02	3.224E-02	2.477E-02	<b>2.143E-02</b>
IQR	7.616E-03	2.272E-03	5.353E-03	3.897E-03	4.437E-03	2.513E-03	<b>1.385E-03</b>
Rank	7	6	4	5	3	2	<b>1</b>
UF5							
Mean	9.915E-01	5.115E-01	4.575E-01	4.239E-01	4.567E-01	<b>2.436E-01</b>	3.835E-01
IQR	1.919E-01	2.443E-02	7.432E-02	9.265E-02	4.103E-02	<b>6.706E-02</b>	2.562E-02
Rank	7	6	5	3	4	<b>1</b>	2
UF6							
Mean	9.532E-02	9.404E-02	9.003E-02	8.991E-02	9.096E-02	1.306E-01	<b>7.366E-02</b>
IQR	1.353E-01	1.883E-02	4.049E-02	3.561E-02	3.336E-02	1.225E-02	<b>1.650E-02</b>
Rank	6	5	3	2	4	7	<b>1</b>
UF7							
Mean	1.034E-01	9.447E-02	8.305E-02	8.998E-02	8.301E-02	<b>6.618E-02</b>	7.024E-02
IQR	5.260E-02	5.761E-02	2.961E-02	2.348E-02	1.719E-02	<b>1.225E-02</b>	1.083E-02
Rank	7	6	4	5	3	<b>1</b>	2
UF8							
Mean	7.811E-02	7.058E-02	5.371E-02	<b>5.212E-02</b>	5.370E-02	6.160E-02	5.225E-02
IQR	1.436E-02	1.134E-02	9.804E-03	<b>9.431E-03</b>	7.871E-03	9.378E-03	1.354E-03
Rank	7	6	4	<b>1</b>	3	5	2
UF9							
Mean	6.076E-02	8.534E-02	2.377E-02	<b>1.690E-02</b>	2.404E-02	3.287E-02	2.977E-02
IQR	1.527E-02	1.239E-02	1.063E-02	<b>1.491E-02</b>	1.649E-02	2.750E-02	1.798E-02
Rank	6	7	2	<b>1</b>	3	5	4
UF10							
Mean	8.996E-01	4.268E-01	5.063E-01	4.774E-01	5.068E-01	4.678E-01	<b>4.264E-01</b>
IQR	8.008E-02	6.948E-02	6.401E-02	4.509E-02	7.312E-02	1.029E-02	<b>4.768E-02</b>
Rank	7	2≈	5	4	6	3	<b>1</b>
Rank Sum	68	53	38	34	38	32	17
+/-/≈	0/10/0	0/9/1	1/9/0	2/7/1	1/9/0	3/7/0	

“+,” “-,” and “≈,” respectively, indicate the result of the algorithm is better than, worse than, and similar to that of MOEA/D-LSIAS based on the Wilcoxon's rank sum test with a significance  $\alpha = 0.05$ .

5.6. *The Dynamics of Operator and Assist Information Selection.* For a further investigation for the dynamic performance, we analyze the usage of operators and operator with different assist information. Divide the search process into

50 search phases. Each phase is composed of 500 function evaluations for ZDT1, while it is composed of 6000 function evaluations for UF1 and DTLZ1. Then calculate the usage number of every phase during searching. The usage state

TABLE 7: Comparative results of all the compared algorithms on the UF test problems regarding HV.

Problems	Algorithms						
	MOEA/D-DE	MOEA/D-DRA	MOEA/D-FRRMAB	MOEA/D-STM	MOEA/D-UCB-T	MOEA/D-ARG	MOEA/D-LSIAS
UF1							
Mean	3.5883	3.5937	3.6157	<b>3.6265</b>	3.6159	3.6189	3.6225
IQR	1.081E-02	7.242E-02	4.570E-03	<b>6.592E-03</b>	1.043E-03	6.012E-03	5.251E-03
Rank	7	6	5	<b>1</b>	4	3	2
UF2							
Mean	3.6094	3.6158	3.6355	3.6202	3.6343	3.6269	<b>3.6383</b>
IQR	7.275E-03	3.945E-03	2.544E-03	2.519E-04	7.553E-04	5.185E-04	<b>4.229E-04</b>
Rank	7	6	2	5	3	4	<b>1</b>
UF3							
Mean	3.5457	3.5753	3.5671	3.5966	3.5688	3.6007	<b>3.6043</b>
IQR	9.167E-03	8.241E-03	8.259E-03	2.594E-03	4.248E-03	1.562E-03	<b>4.578E-03</b>
Rank	7	4	6	3	5	2	<b>1</b>
UF4							
Mean	3.1729	3.1667	3.2067	3.1818	3.2072	3.2229	<b>3.2244</b>
IQR	3.115E-02	5.524E-02	4.812E-02	3.325E-02	2.901E-02	1.099E-02	<b>2.980E-02</b>
Rank	6	7	4	5	3	2≈	<b>1</b>
UF5							
Mean	2.7922	2.8537	2.9587	<b>3.1919</b>	2.9228	3.0537	3.1661
IQR	2.840E-01	2.572E-01	2.056E-01	<b>1.468E-01</b>	1.345E-01	2.039E-01	2.280E-01
Rank	7	6	4	<b>1</b>	5	3	2
UF6							
Mean	2.7671	2.8021	2.9891	3.0071	2.9920	2.8444	<b>3.1059</b>
IQR	2.369E-01	1.365E-01	2.258E-02	1.052E-01	2.142E-01	2.151E-01	<b>1.387E-01</b>
Rank	7	6	4	2	3	5	<b>1</b>
UF7							
Mean	3.4147	3.4276	3.4396	3.4534	3.4392	3.4615	<b>3.4774</b>
IQR	1.933E-02	4.151E-02	3.163E-02	1.407E-02	4.332E-02	2.079E-02	<b>4.769E-02</b>
Rank	7	6	4	3	5	2	<b>1</b>
UF8							
Mean	6.3576	6.3327	<b>6.7828</b>	6.6325	6.7813	6.4792	6.5927
IQR	5.295E-02	8.850E-02	<b>1.058E-02</b>	3.126E-02	9.983E-03	1.030E-02	2.020E-02
Rank	6	7	<b>1</b>	3	2	5	4
UF9							
Mean	6.8127	7.0236	7.0345	<b>7.1575</b>	7.0327	7.1082	7.1056
IQR	3.398E-01	4.695E-01	4.586E-01	<b>2.257E-02</b>	5.999E-02	1.320E-02	8.853E-02
Rank	7	6	4	<b>1</b>	5	2	3
UF10							
Mean	3.6443	3.7456	3.7198	3.6211	3.7203	<b>3.8108</b>	3.8102
IQR	7.962E-02	7.893E-02	5.725E-02	5.124E-02	3.885E-02	<b>5.519E-02</b>	4.086E-02
Rank	6	3	5	7	4	<b>1</b>	2
Rank Sum	67	57	39	31	39	29	18
+/-/≈	0/10/0	0/10/0	1/9/0	4/6/0	1/9/0	2/7/1	

“+,” “-,” and “≈,” respectively, indicate the result of the algorithm is better than, worse than, and similar to that of MOEA/D-LSIAS based on the Wilcoxon's rank sum test with a significance  $\alpha = 0.05$ .

of operators and operators with different assist information during the whole search process on ZDT1, DTLZ1, and UF1 are shown in Figures 4–6.

In Figure 5, op4 shows effect search power during all the phases, but the impact of the other three operators is

unable to be ignored. op1 performs better with ne1 than the other neighborhoods, but op1 only provides effective search power at the beginning of the search phase. op3 has two good partners which are ne1 and ne3, and op3 plays an important role during the second half of the search. During the whole

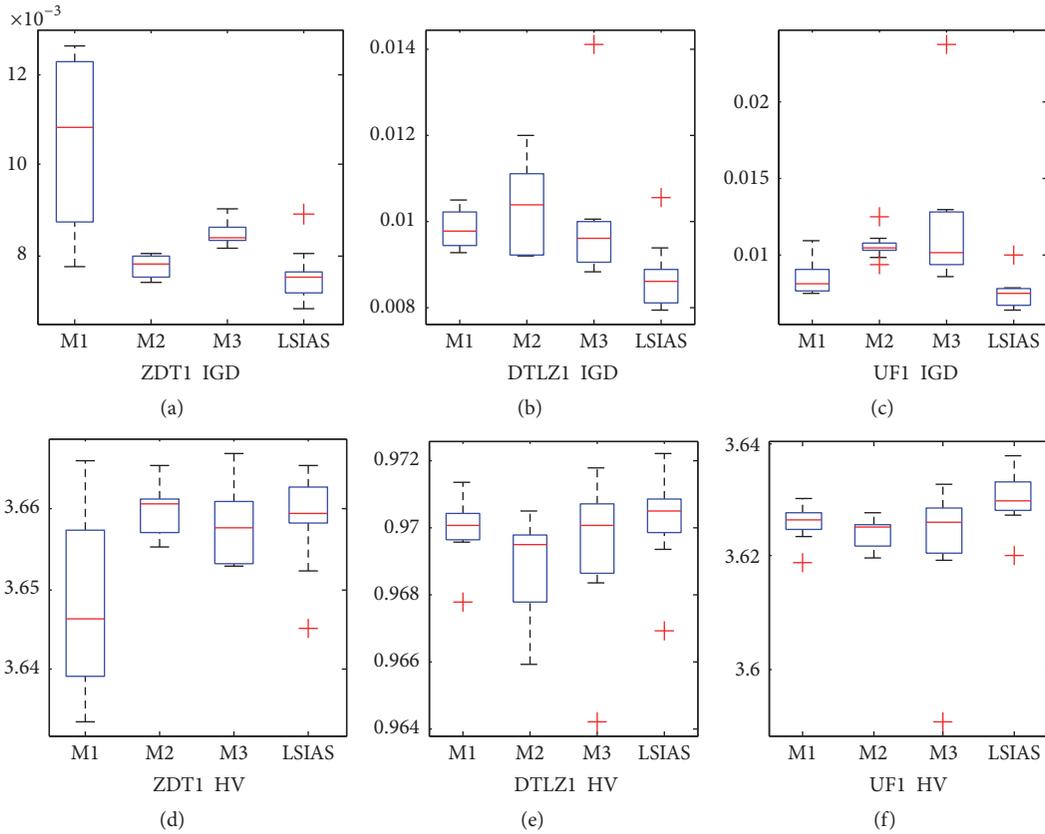


FIGURE 4: Box plots for the comparison of the MOEA/D-LSIAS and the compared algorithms on ZDT1, DTLZ1, and UF1 regarding IGD and HV.

search phase, all the assist information work well with op4.

In Figure 6, it is found that at the beginning of the search phase op3 with ne3 and op4 with ne2 possess powerful search capability. At the first half of search phase, op1 provides effective search, but the four operators provide effective search in turn. The important partners of op1 are ne1 and ne1. op2 works well with all the neighborhoods, but op4 only works well with ne2.

As observed from Figure 7, all of four operators work hard at the beginning, but op4 and op3 provide little help during the second half of search phase. ne3 offers significant effect for different operators during different search phases. It means that the farthest neighborhood set is very helpful for the search. ne1, ne2, and ne3 offer help for op1 and op2 in turn. op3 and op4 perform well with ne2 and ne3. It is obvious that op4 works well at the first half of the search phase, but it cannot show a good performance later.

## 6. Conclusions

In this paper, a novel AOS method called LSIAS is introduced. In the LSIAS, the best operator usage rate and the worst operator usage rate are used to improve the UCB method. In case of the influence of unexpected extremely large

or small fitness improvement, a credit assignment method which is abandonment of extreme fitness improvement is introduced. LSIAS is also an adaptive selection strategy, which is used to select operators and assist information on operator adaptively. Every latest used operator with all of its assist information and its efficiency are dynamically stored in a sliding window. Based on the efficiency, the best configuration of the operator and its assist information for each search phase are dynamically chosen.

Since the decomposition-based MOEAs could help in evaluating the efficiency of operators with their assist information easily, a variant of MOEA/D is adopted to investigate the performance of our proposed LSIAS. Four DE mutation operators and two assist information, which are neighborhoods and scaling factors, are used in this method within the MOEA/D framework. We conduct extensive experimental studies on three kinds of benchmark functions. The experiments show that LSIAS is robust and effective and its adaptive selection method of operators and assist information can significantly improve the performance of MOEA/D.

In the future work, the adaptive selection of operators with assist information will be used to investigate many-objective optimization problems and constraint optimization problems. Furthermore, as the indicator-based MOEAs can

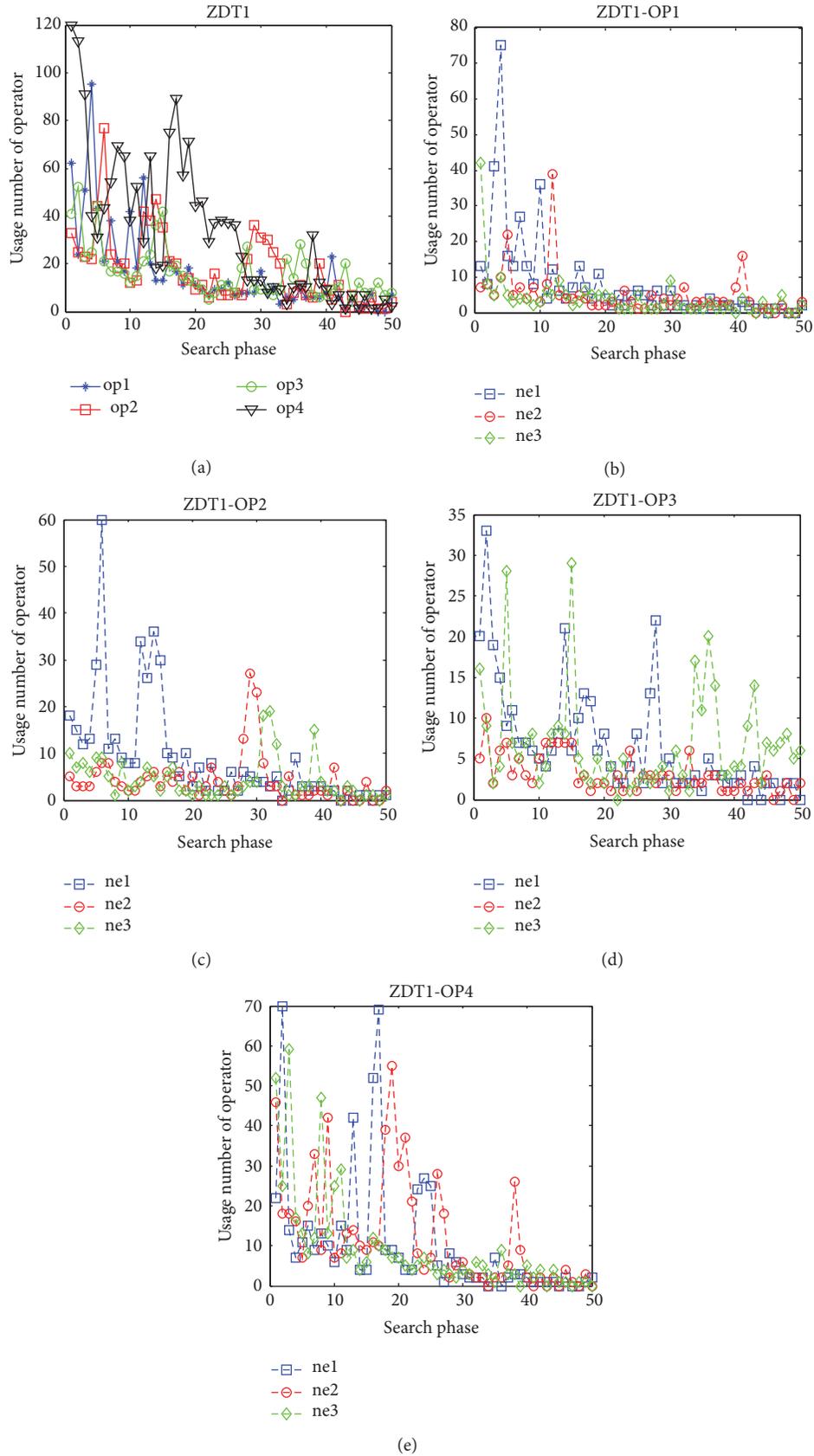


FIGURE 5: Operators and assist information usage number during the whole search process on ZDT1.

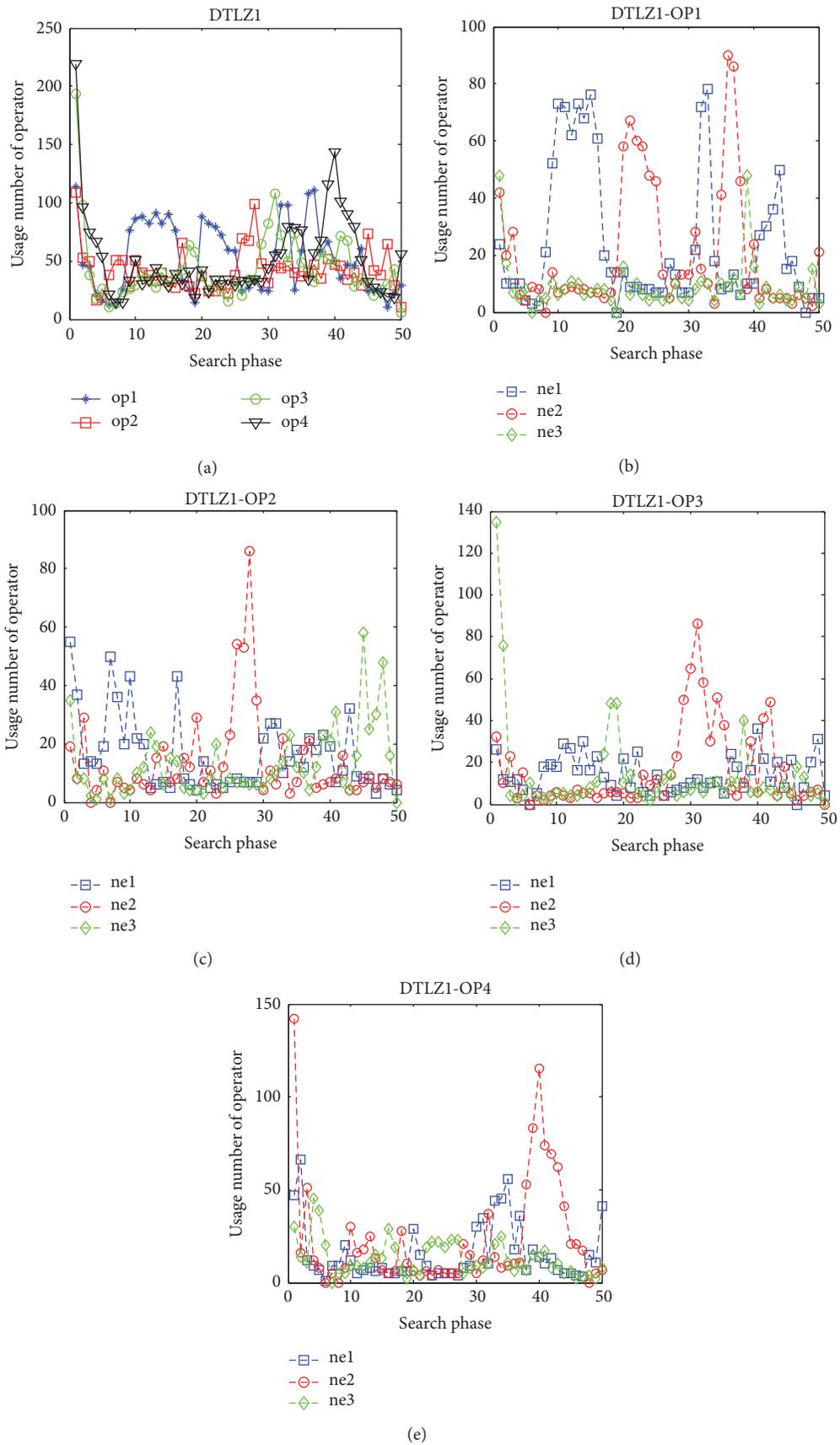


FIGURE 6: Operators and assist information usage number during the whole search process on DTLZ1.

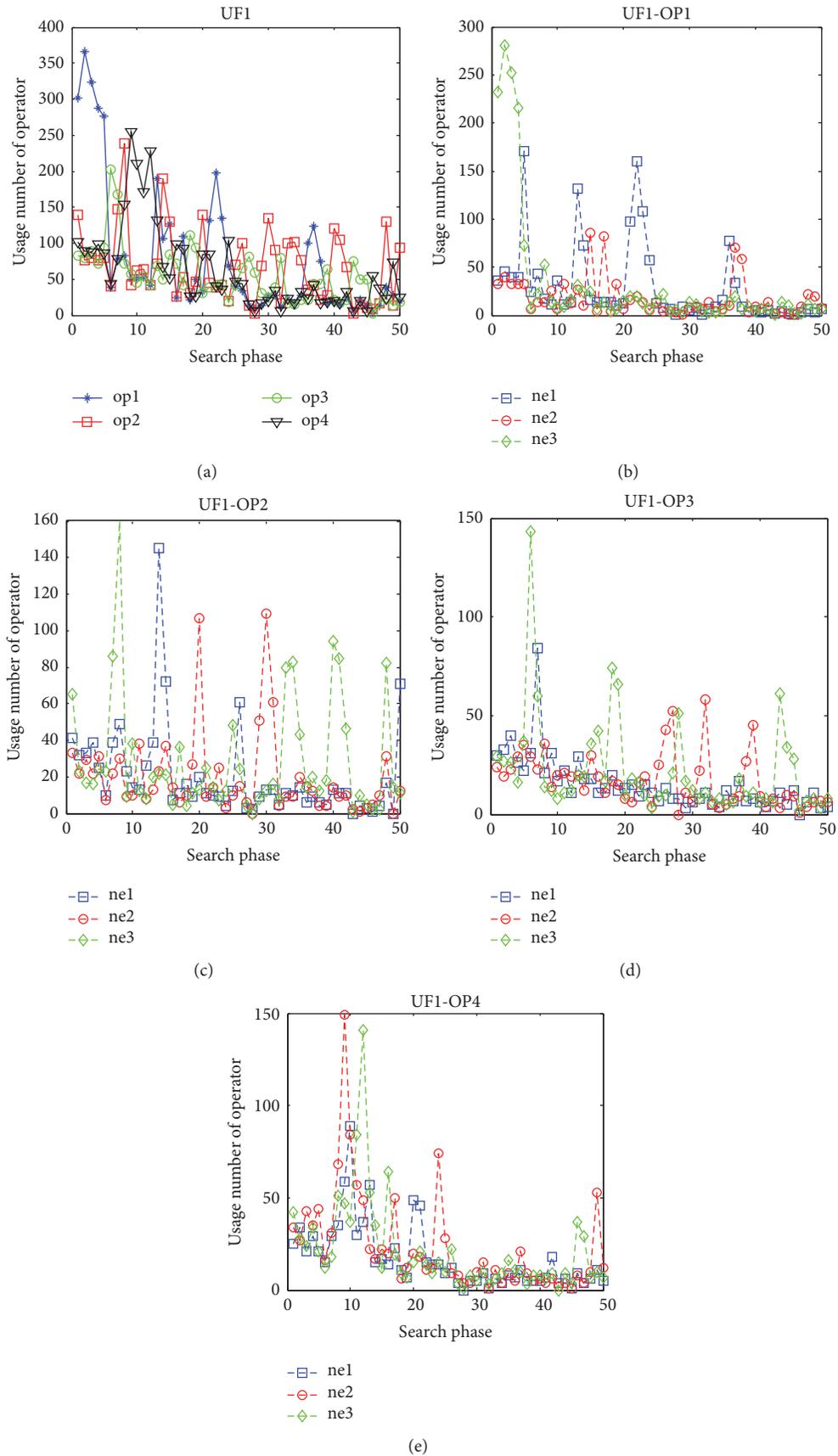


FIGURE 7: Operators and assist information usage number during the whole process on UF1.

also provide exact estimation for operators with assist information, the performance of LSIA within this algorithm can also be studied.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

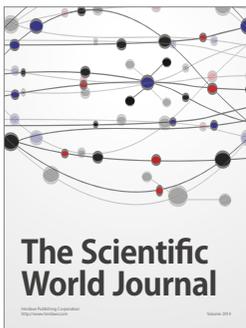
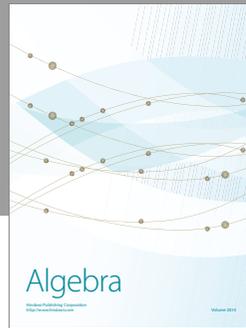
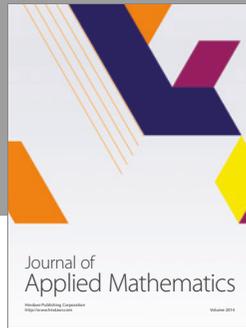
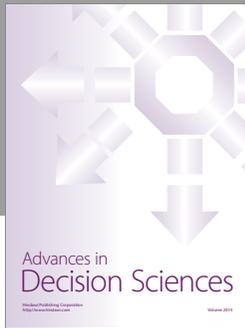
## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants nos. 7170129 and 71771216).

## References

- [1] Y. Xiang, Y. Zhou, and H. Liu, "An elitism based multi-objective artificial bee colony algorithm," *European Journal of Operational Research*, vol. 245, no. 1, pp. 168–193, 2015.
- [2] A. Lara, G. Sanchez, C. A. C. Coello, and O. Schütze, "HCS: A new local search strategy for memetic multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 112–132, 2010.
- [3] D. J. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithm," in *Proceedings of the First International Conference of Genetic Algorithms and Their Application*, 1985.
- [4] N. Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," *Evolutionary Computation*, vol. 2, pp. 221–248, 1994.
- [5] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, pp. 82–87, June 1994.
- [6] C. M. Fonseca and Fleming P. J., "Genetic algorithms for multiobjective optimization: formulation discussion and generalization," in *Proceedings of the 5th International Conference on Genetic Algorithms (ICGA '93)*, pp. 416–423, 1993.
- [7] 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.
- [8] Z. Eckart, M. Laumanns, and L. Thiele, *SPEA2: Improving the strength Pareto evolutionary algorithm*, 2001.
- [9] 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.
- [10] J.-K. Xiao, W.-M. Li, X.-R. Xiao, and C.-Z. Lv, "A novel immune dominance selection multi-objective optimization algorithm for solving multi-objective optimization problems," *Applied Intelligence*, vol. 46, no. 3, pp. 739–755, 2017.
- [11] M. Afzalirad and J. Rezaeian, "A realistic variant of bi-objective unrelated parallel machine scheduling problem: NSGA-II and MOACO approaches," *Applied Soft Computing*, vol. 50, pp. 109–123, 2017.
- [12] S. Santander-Jiménez and M. A. Vega-Rodríguez, "Using mixed mode programming to parallelize an indicator-based evolutionary algorithm for inferring multiobjective phylogenetic histories," *Soft Computing*, pp. 1–20, 2016.
- [13] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.
- [14] 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.
- [15] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.
- [16] 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.
- [17] Q. Zhu, Q. Lin, Z. Du et al., "A novel adaptive hybrid crossover operator for multiobjective evolutionary algorithm," *Information Sciences*, vol. 345, pp. 177–198, 2016.
- [18] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, no. 2, pp. 1679–1696, 2011.
- [19] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 482–500, 2012.
- [20] S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 3, pp. 442–446, 2012.
- [21] R. A. Gonçalves, C. P. Almeida, and A. Pozo, "Upper confidence bound (UCB) algorithms for adaptive operator selection in MOEA/D," *Lecture Notes in Computer Science (including sub-series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9018, pp. 411–425, 2015.
- [22] L. P. Kaelbling, "Associative Reinforcement Learning: A Generate and Test Algorithm," *Machine Learning*, vol. 15, no. 3, pp. 299–319, 1994.
- [23] L. DaCosta, Á. Fialho, M. Schoenauer, and M. Sebag, "Adaptive operator selection with dynamic multi-armed bandits," in *Proceedings of the 10th Annual Genetic and Evolutionary Computation Conference, GECCO 2008*, pp. 913–920, USA, July 2008.
- [24] Á. Fialho, L. Da Costa, M. Schoenauer, and M. Sebag, "Analyzing bandit-based adaptive operator selection mechanisms," *Annals of Mathematics and Artificial Intelligence*, vol. 60, no. 1–2, pp. 25–64, 2010.
- [25] Á. Fialho, M. Schoenauer, and M. Sebag, "Toward comparison-based adaptive operator selection," in *Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference, GECCO-2010*, pp. 767–774, USA, July 2010.
- [26] S. M. Venske, R. A. Gonçalves, and M. R. Delgado, "ADEMO/D: Multiobjective optimization by an adaptive differential evolution algorithm," *Neurocomputing*, vol. 127, pp. 65–77, 2014.
- [27] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *Proceedings of the Proceeding of the IEEE Congress on Evolutionary Computation (CEC '09)*, pp. 203–208, Trondheim, Norway, May 2009.
- [28] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
- [29] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the 2002*

- Congress on Evolutionary Computation, CEC 2002*, pp. 825–830, usa, May 2002.
- [30] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, “Multiobjective optimization test instances for the CEC 2009 special session and competition,” special session on performance assessment of multi-objective optimization algorithms, technical report, University of Essex, Colchester, UK and Nanyang technological University, Singapore, 2008.
- [31] 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.
- [32] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, “Stable matching-based selection in evolutionary multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 6, pp. 909–923, 2014.
- [33] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, “Adaptive Replacement Strategies for MOEA/D,” *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 474–486, 2016.
- [34] R. Storn, “On the usage of differential evolution for function optimization,” in *Proceedings of the Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS '96)*, pp. 519–523, June 1996.
- [35] K. V. Price, R. M. Storn, and J. A. Lampinen, “The differential evolution algorithm,” *Differential evolution: a practical approach to global optimization*, pp. 37–134, 2005.
- [36] A. W. Iorio and X. Li, “Solving rotated multi-objective optimization problems using differential evolution,” in *AI 2004: Advances in artificial intelligence*, vol. 3339 of *Lecture Notes in Comput. Sci.*, pp. 861–872, Springer, Berlin, 2004.
- [37] D. E. Goldberg, “Probability Matching, the Magnitude of Reinforcement, and Classifier System Bidding,” *Machine Learning*, vol. 5, no. 4, pp. 407–425, 1990.
- [38] D. Thierens, “An adaptive pursuit strategy for allocating operator probabilities,” in *Proceedings of the GECCO 2005 - Genetic and Evolutionary Computation Conference*, pp. 1539–1546, USA, June 2005.
- [39] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” *Machine Learning*, vol. 47, no. 2-3, pp. 235–256, 2002.
- [40] J. M. Whitacre, T. Q. Pham, and R. A. Sarker, “Use of statistical outlier detection method in adaptive evolutionary algorithms track: Genetic algorithms,” in *Proceedings of the 8th Annual Genetic and Evolutionary Computation Conference 2006*, pp. 1345–1352, usa, July 2006.
- [41] J. Maturana, F. Lardeux, and F. Saubion, “Autonomous operator management for evolutionary algorithms,” *Journal of Heuristics*, vol. 16, no. 6, pp. 881–909, 2010.
- [42] K. Li, Á. Fialho, and S. Kwong, “Multi-objective Differential Evolution with adaptive control of parameters and operators,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6683, pp. 473–487, 2011.
- [43] O. Schütze, V. A. S. Hernández, H. Trautmann, and G. Rudolph, “The hypervolume based directed search method for multi-objective optimization problems,” *Journal of Heuristics*, pp. 1–28, 2016.
- [44] M. Li, S. Yang, K. Li, and X. Liu, “Evolutionary algorithms with segment-based search for multiobjective optimization problems,” *IEEE Transactions on Cybernetics*, vol. 44, no. 8, pp. 1295–1313, 2014.
- [45] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, “Performance assessment of multiobjective optimizers: an analysis and review,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.
- [46] B. Akay, “Synchronous and asynchronous Pareto-based multi-objective Artificial Bee Colony algorithms,” *Journal of Global Optimization*, vol. 57, no. 2, pp. 415–445, 2013.
- [47] O. Schütze, X. Esquivel, A. Lara, and C. A. C. Coello, “Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 4, pp. 504–522, 2012.



# Hindawi

Submit your manuscripts at  
<https://www.hindawi.com>

