# A Multi-Operator Search Strategy based on Cheap Surrogate Models for Evolutionary Optimization

Wenyin Gong, Aimin Zhou, and Zhihua Cai

***Abstract*—It is well known that in evolutionary algorithms, different reproduction operators may be suitable for different problems or in different running stages. To improve the algorithm performance, the ensemble of multiple operators has become popular. Most ensemble techniques achieve this goal by choosing an operator according to a probability learned from the previous experience. In contrast to these ensemble techniques, in this paper we propose a cheap surrogate model based multi-operator search strategy for evolutionary optimization. In our approach, a set of candidate offspring solutions are generated by using the multiple offspring reproduction operators, and the best one according to the surrogate model is chosen as the offspring solution. Two major advantages of this approach are (a) each operator can generate a solution for competition compared to the probability based approaches, and (b) the surrogate model building is relatively cheap compared to that in the surrogate-assisted evolutionary algorithms. The model is used to implement multi-operator ensemble in two popular evolutionary algorithms, that is, differential evolution and particle swarm optimization. Thirty benchmark functions and the functions presented in the CEC 2013 are chosen as the test suite to evaluate our approach. Experimental results indicate that the new approach can improve the performance of single operator based methods in the majority of the functions.**

***Index Terms*—Evolutionary algorithm, global optimization, multi-operator ensemble, surrogate model.**

## I. Introduction

WITHOUT loss of generality, in this work, the following numerical optimization problem is considered:

$$\text{Minimize} \quad f(\mathbf{x}), \qquad \mathbf{x} \in S, \tag{1}$$

where $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$ is an $n$-dimensional decision variable vector, and $S \subseteq \mathbb{R}^n$ is a compact set which denotes the feasible region of the search space. Usually

$$S = [\underline{x}_j, \overline{x}_j]^n$$

where $j = 1, 2, \cdots, n$, $\underline{x}_j$ and $\overline{x}_j$ are respectively the lower bound and upper bound of $x_j$.

This work was partly supported by China National Instrumentation Program (2012YQ180132), the National Natural Science Foundation of China under Grant No. 61273313, 61203307 and 61075063, and the Fundamental Research Funds for the Central Universities at China University of Geosciences (Wuhan) under Grant No. CUG130413. (*Corresponding author*: A. Zhou.)

W. Gong and Z. Cai are with Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan, 430074, China. They are also with School of Computer Science, China University of Geosciences, Wuhan, 430074, China. (Email: wygong@cug.edu.cn; zhcai@cug.edu.cn).

A. Zhou is with the Shanghai Key Laboratory of Multidimensional Information Processing and the Department of Computer Science and Technology, East China Normal University, 500 Dongchuan Road, Shanghai 200241, China. (Email: amzhou@cs.ecnu.edu.cn).

*Evolutionary algorithms (EAs)* are a kind of search techniques based on the simulated evolutionary process of natural selection, variation, and genetics [1]. During the last few decades, a variety of EAs have been proposed, such as genetic algorithm (GA), evolution strategy (ES), evolutionary programming (EP) [1], differential evolution (DE) [2], particle swarm optimization (PSO) [3], etc. A major difference between these algorithms is in the choice of reproduction operator, that is, the way to generate new trial solutions. Meanwhile, new operators are being developed every day. On the one hand, these operators are more or less similar; on the other hand, different operators may be suitable for different problems or in different running stages.

In order to solve a wide range of problems, combining different search operators might be a way to improve current EAs. However, how to combine these operators more efficiently is still a challenging task. Most existing techniques firstly build a posterior probability model based on previous successful experience in each generation and then choose one for offspring reproduction according to the posterior probability model [4], [5], [6], [7]. However, it is not trivial to build and update a proper posterior probability model; furthermore, an incorrect model may mislead the search. To alleviate this problem, in this paper we propose a *cheap surrogate model (CSM)* based multi-operator search strategy for evolutionary optimization. In contrast to the probability-based techniques, a CSM is built in each generation first; a set of candidate solutions are generated for each individual by using multiple search operators, and the best one is chosen as the corresponding offspring solution according to the CSM. In comparison with the probability based approach, each search operator can generate candidate offspring solutions.

It should be noted that the major target of this paper is not to propose some new EAs but to propose a new strategy, CSM, to improve the performance of existing EAs. To this end, the CSM is first integrated into three advanced EAs: JADE [8], OLPSO [9], and CoBiDE [10], to implement a multi-operator ensemble for each of them. Secondly, we try to implement a hybrid DE and PSO method by combining DE and PSO search operators based on the CSM. These CSM-based approaches are systematically compared to the original EAs and some existing multi-operator search algorithms on 30 benchmarks chosen from the literature [7], [11], [12], [13], [14] and also on the CEC 2013 test suite [15]. Experimental results indicate that the CSM-based algorithms can obtain better performance than the corresponding single operator based algorithms in the majority of the test functions. Moreover, they can also provide better results than the compared multi-operator approaches

presented in [6] and [16].

The rest of the paper is organized as follows. Section II briefly reviews the related work, including the multi-operator search techniques and the surrogate-assisted optimization in EAs. In Section III, a novel multi-operator search strategy based on the computationally cheap surrogate models is proposed in detail, followed by the empirical study on the proposed CSM-based multi-operator EAs in Section IV. Finally, the paper is concluded in Section V with some suggestions for future work.

## II. Related Work

In this section, we first briefly review the multi-operator search techniques proposed in the literature, followed by a brief introduction of surrogate-assisted optimization in EAs.

### A. Multi-Operator Search

As mentioned above, a variety of search operators have been developed. Different operators might be suitable for different problems or in different running stages. Therefore, in order to solve a wide range of problems, the ensemble of different operators in one single algorithm represents one of the most promising areas of research in evolutionary computation.

The following are representative multi-operator search techniques.

- *Adaptive Operator Selection:* Adaptation or self-adaptation of the selection probabilities of different operators has been extensively studied recently [17], [18]. The key issue is how to assign the reproduction operator selection probabilities adaptively. To achieve this goal, a variety of strategies, such as decision making scheme [19], probability matching [20], adaptive pursuit [21], game theory [22], previous experience [5], [23], learning [7], and combinations of different strategies [6], have been proposed in the last few years.
- *Multi-Method based Search:* Since no single algorithm is always efficient for a diverse set of optimization problems, recently, it has been natural to use multiple methods simultaneously. In [24], a multi-method was proposed for multi-objective optimization, where the concepts of global information sharing and genetically adaptive offspring creation were utilized to tune the number of offspring generated by each method in a generation. This idea was also extended to scalar-objective optimization [25]. In [26], another cooperation strategy was proposed, where each constituent algorithm was run with a given budget, and different algorithms were interacted with a migration strategy.
- *Adaptive Memetic Algorithms:* Adaptive memetic algorithms represent another kind of multi-operator search strategy, where different local search operators are adaptively selected for individual improving. In memetic algorithms, the hill-climbing [27], gradient-based methods [28], [29], and some other search methods [4] are chosen as local search operators in the framework of EAs. A comprehensive survey and comparative study of memetic algorithms can be found in [30], [31].

In addition to the above mentioned strategies, there are also some other search techniques [16], [32], [33], [34], [35], which can be regarded as multi-operator search strategies. In all of these methods, how to choose a proper search operator while keeping the search efficiency plays a key role.

### B. Surrogate-Assisted Optimization

In many cases when the fitness evaluation needs a computational simulation or an experiment, it is expensive to evaluate an individual. A key issue with these kinds of optimization problems is how to balance the number of fitness evaluations and the solution quality. The *surrogate-assisted evolutionary algorithm (SAEA)* is a technique that deals with expensive optimization problems [36], [37]. The basic idea behind SAEA is that it builds an alternative function $\hat{f}(\mathbf{x})$, called surrogate model, based on some of the obtained individuals $\{(\mathbf{x}, f(\mathbf{x}))\}$, and estimates the fitness values of some new trial individuals through the surrogate model. Therefore, the number of function evaluations based on the original expensive function $f(\mathbf{x})$ is reduced.

In practice, there are two issues that should be considered in designing a SAEA.

- *Surrogate Model Definition:* This issue is related to how to define a surrogate model. It is arguable that all regression models can be applied here. In practice, some widely used models include the linear models, polynomial models, support vector regression, Kriging or Gaussian process [38], [39], radial basis function network [40], and some other techniques [41]. It should be noted that different models have their own properties and they should be used properly. For example, the linear models and polynomial models are cheap to build but not accurate, while the Gaussian process is accurate but expensive to build. In addition to these properties, there are some other issues to be considered when building a surrogate model, such as (a) whether to build a local or global model, (b) a single model or multiple models, and (c) the relationship between the model accuracy, the dimension of the search space, and the number of obtained individuals. Since a surrogate model does not come without cost, it is hard to choose a proper model in practice.
- *Surrogate Model Management:* This issue is related to how to use a surrogate model. A first concern is in which stage a surrogate model can be used. Actually, the surrogate model can be used in all stages of an EA, such as population initialization, offspring reproduction, pre-selection, and local search [36], [37]. However, a challenge might be how to use both the surrogate model and the original function to obtain a satisfactory solution while using a minimum number of original function evaluations. When and which individuals are evaluated in the surrogate model are other key issues to be considered. However, this task is not trivial and the individual-based, generation-based, and population-based strategies have been well studied in the literature. All of these strategies lead to some additional control parameters, which are not easy to set in practice.

No matter how the model is defined or how it is managed, it is arguable that it is highly related to the problem to be tackled. SAEA is extremely useful for practical problems [42], [43], [44] and it has also been extended to multi-objective optimization [45], [38]. As far as we know, most current SAEAs focus on expensive problems and not much work has been done on applying surrogate models to general optimization problems. A major concern might be that compared to fitness evaluation and EA itself, the surrogate modeling process is expensive. In this work, we propose to use a CSM to guide the search process of EAs.

## III. Proposed Method

As reviewed in Section II, most multi-operator search techniques actually choose one operator for offspring reproduction by maintaining a probability model. In addition, the surrogate models are mainly used for solving the expensive optimization problems in EAs. Recently, an estimation of distribution algorithm (EDA) based on nonparametric density estimation was proposed [46], where the nonparametric density estimation is used to estimate the quality of candidate offspring solutions sampled from the Gaussian distribution and thus to filter out low quality ones. Based on the above considerations and inspired by the work in [46], we propose a novel multi-operator search strategy, where firstly a set of candidate offspring solutions are generated by multiple reproduction operators for each solution and then a computationally CSM is built to filter these candidates and leave one as the offspring. Fig. 1 illustrates the basic idea of the proposed multi-operator search strategy, and more details are elucidated in the following sections.

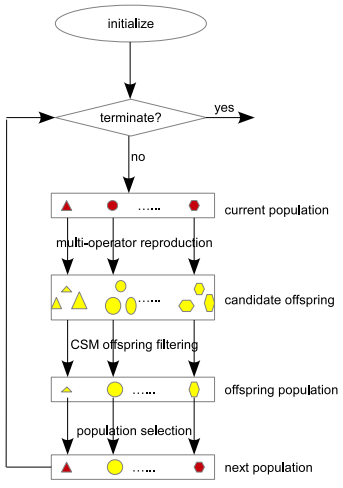



Fig. 1. An illustration of the proposed multi-operator search strategy.

### A. Candidate Generation

Suppose that we have $\lambda$ different reproduction operators in the operator pool, that is, $\mathbf{OP} = \{op_1, \cdots, op_\lambda\}$. The population contains $\mu$ solutions. In the evolutionary cycle, for each parent $\mathbf{x}_i$ $(i = 1, \cdots, \mu)$, $\lambda$ different candidate points $\mathbf{y}_{i,1}, \cdots, \mathbf{y}_{i,\lambda}$ are generated by each of the operators in the pool[1].

### B. Cheap Surrogate Model Building

After offspring generation, we need to estimate the quality of the candidate solutions. This is usually achieved by surrogate models. However, in surrogate model building, it is time consuming to build an accurate model and it is required that the dimension of the variable vector should not be high. For a general optimization problem, the algorithm time complexity and the scalability of the problem are two major concerns. For this reason, we turn to the CSMs based on density estimation, by which the two concerns can be solved.

It is reasonable to assume that the objective function $f(\mathbf{x})$ is continuous and not constant in the feasible region $S$. Let

$$f_{\max} = \max_{\mathbf{x} \in S} f(\mathbf{x})$$

and

$$F(\mathbf{x}) = \frac{f_{\max} - f(\mathbf{x})}{\int_S (f_{\max} - f(\mathbf{x}))\, \mathrm{d}\mathbf{x}}.$$

It can be proven that

$$0 \leq F(\mathbf{x}) \leq 1$$

and

$$\int_S F(\mathbf{x})\, \mathrm{d}\mathbf{x} = 1.$$

Therefore, $F(\mathbf{x})$ can be regarded as a probability density function, and

$$\arg\min_{\mathbf{x} \in S} f(\mathbf{x}) = \arg\max_{\mathbf{x} \in S} F(\mathbf{x}),$$

that is, the minimization of $f(\mathbf{x})$ is equal to finding the point with the highest probability density. In this section, we estimate the quality of a candidate offspring solution $\mathbf{x}$ by calculating the density $F(\mathbf{x})$ instead of calculating its function value $f(\mathbf{x})$.

In this work, the Parzen window method [47, Section 4.3] is employed to estimate the density probability of each candidate point. The Parzen window method (also known as kernel density estimation in pattern classification literature [47, Section 4.3]) is the most popular density estimation method, which calculates the density probability as follows:

$$\hat{F}(\mathbf{y}) = \frac{1}{\mu} \sum_{i=1}^{\mu} \left( \frac{1}{w} \varphi \left( \frac{||\mathbf{y} - \mathbf{x}_i||_2}{w} \right) \right). \tag{2}$$

Similar to [46], to make the better solutions contribute more to the density, the ranking of each solution is used. Then, the density estimation function in Equation (2) is modified as

$$\hat{F}(\mathbf{y}) = \frac{1}{\mu} \sum_{i=1}^{\mu} \left( \frac{R_i}{\mu} \frac{1}{w} \varphi \left( \frac{||\mathbf{y} - \mathbf{x}_i||_2}{w} \right) \right) \tag{3}$$

where

[1]In this work, we suppose that each reproduction operator only generates one candidate point. However, the operators that generate more than one candidate points can also be used in our proposed CSM-based multi-operator search strategy.

- $||x||_2 = \sqrt{\sum_{j=1}^{n} x_j^2}$ denotes the $L_2$ norm.
- $R_i$ is the ranking of solution $\mathbf{x}_i$ in the sorted population (from the best to the worst), and is calculated as [48]

$$R_i = \mu - i + 1 \tag{4}$$

- $\varphi(u)$ is the kernel. Different kernels can be used in Equation (3), in this work, two kernels are used:
  - *The Epanechnikov kernel*:

$$\varphi(u) = \frac{3}{4}(1-u^2)\mathbf{1}_{\{|u|\leq 1\}} \tag{5}$$

  - *The normal kernel*:

$$\varphi(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) \tag{6}$$

  The two kernels are selected because (a) the Epanechnikov kernel is optimal in a minimum variance sense [49], and (b) the normal kernel is often used in pattern recognition [47];
- $w$ is the window width and is calculated as [46]

$$w = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(\overline{a}_j - \underline{b}_j\right)^2} \tag{7}$$

  where $\overline{a}_j = \arg\max_{i=1,\cdots,\mu} x_{i,j}$ and $\underline{b}_j = \arg\min_{i=1,\cdots,\mu} x_{i,j}$. Note that to make sure $|u| \leq 1$ in Equation (5), when a new candidate point $\mathbf{y}$ is generated by the reproduction operator, if $y_j < \underline{b}_j$ or $y_j > \overline{a}_j$, then $\underline{b}_j$ or $\overline{a}_j$ will be updated immediately.

It should be noted that (a) the parameters $R_i$ and $w$ in the density estimation model $\hat{F}(x)$ are directly calculated from the given data without much cost, and (b) the kernel functions are not highly related to the dimension of the variable vector. It is also to be expected that the density estimation might not be as accurate as a general surrogate model. However, our target is not to build accurate surrogate models but to guide the search of an EA with learned information from the population.

### C. Offspring Filtering

Following the density probability estimation, one offspring $\mathbf{y}_i^*$ will be chosen as the offspring from the candidate points according to their estimated probabilities $\hat{F}(\mathbf{y}_{i,1}), \cdots, \hat{F}(\mathbf{y}_{i,\lambda})$ by Equation (3). Different techniques can be used as the filtering methods. In this work, two techniques are selected for illustration.

- *Greedy selection*: The point with the maximal density value is selected as the offspring, that is,

$$\mathbf{y}_i^* = \arg\max_{k=1,\cdots,\lambda} \hat{F}(\mathbf{y}_{i,k}) \tag{8}$$

- *Tournament selection*: In this technique, two different points are randomly selected from $\lambda$ candidate points, then the one with the higher density value will be selected as the offspring $\mathbf{y}_i^*$.

Finally, the offspring $\mathbf{y}_i^*$ will be evaluated and compared with its parent $\mathbf{x}_i$ for the survival selection.

**Algorithm 1** Framework of CSM-based multi-operator search

1: Initialize the population with $\mu$ solutions $\mathbf{x}_1, \cdots, \mathbf{x}_\mu$.
2: Evaluate the fitness for each solution.
3: **while** termination criterion is not satisfied **do**
4:  **for** $i = 1$ to $\mu$ **do**
5:  Generate the candidate points $\mathbf{y}_{i,1}, \cdots, \mathbf{y}_{i,\lambda}$ by each operator in $\mathbf{OP}$ for the parent $\mathbf{x}_i$.
6:  Estimate the density $\hat{F}(\mathbf{y}_{i,k})$ of each candidate point $\mathbf{y}_{i,k}$.
7:  Choose the offspring $\mathbf{y}_i^*$ according to the density values of the candidate points.
8:  Calculate $f(\mathbf{y}_i^*)$ for the offspring $\mathbf{y}_i^*$.
9:  Select the better one between $\mathbf{x}_i$ and $\mathbf{y}_i^*$ to the next generation.
10:  **end for**
11: **end while**

### D. Algorithm Framework

Combining the above-mentioned issues, the algorithm framework of the proposed CSM-based multi-operator search strategy is described in Algorithm 1. The candidate points are generated by different operators in $\mathbf{OP}$ for each parent in Line 5. Then, the density value of each point is estimated in Line 6, followed by the offspring filtering in Line 7. In Line 8, the offspring is evaluated and then compared with its parent solution in Line 9. According to Algorithm 1, since each operator in $\mathbf{OP}$ generates a candidate point in Line 5, no operator will be lost in the evolution process. By contrast in the selection probability based multi-operator search methods, if some operators have very small probabilities due to their poor performance in the previous stages, they may be lost when it comes to generate new candidate points in subsequent running stages, even though they would have been more suitable.

### E. Remarks

Note that our proposed CSM-based multi-operator search strategy is inspired by the work in [46]. However, there are significant differences between them: (a) In [46], the nonparametric density estimation technique is used to filter offspring from different candidate points which are only sampled from the Gaussian distribution; whereas in this work different kinds of reproduction operators are used to generate candidate points. (b) To make the better solutions contribute more density, the ranking of each parent is used as shown in Equation (3). In [46], the objective function values is used, but this may cause the better solutions to provide significantly greater density values than the worse ones if their objective function values are several orders of magnitude larger. In this way, it may lead to the premature convergence for the multimodal functions. (c) In [46] only the normal kernel is used in Equation (3), whereas in this work both the normal kernel and Epanechnikov kernel are used and their performance is empirically compared.

In the proposed CSM-based multi-operator search strategy, an additional computational cost is caused in lines 5, 6, and 7 of Algorithm 1. Suppose that the complexity of the generation operator (such as the operators in DE and PSO) is $O(n)$, then the complexity in line 5 is $O(\lambda \cdot n)$, where $n$ is the number of decision variables. In line 6, the complexity of density

estimation is $O(\lambda \cdot \mu \cdot n)$, when the CSM is used. In line 7, the complexity of offspring filtering is $O(\lambda)$. Therefore, the overall additional complexity of our method is $O(\lambda \cdot \mu \cdot n)$. In summary, in Algorithm 1, the overall complexity is $O(\lambda \cdot \mu^2 \cdot n)$.

It is clear that the framework shown in Algorithm 1 is generic, it can be used to implement the multi-operator ensemble for different EAs. To indicate the performance of our proposed CSM-based multi-operator search strategy, it is implemented as the multi-operator search for three advanced EAs: JADE [8], OLPSO [9], and CoBiDE [10]. As an illustration, it is also used to implement the hybrid DE and PSO, in which different DE operators and PSO operators are used.

## IV. Experimental Results and Analysis

In this section, the performance of the proposed CSM-based multi-operator search strategy is evaluated through the benchmark functions. Firstly, the influences of different kernels (see Section III-B) with different selection techniques (see Section III-C) are evaluated. Then, our approach is used to implement multi-operator search for JADE, OLPSO, CoBiDE, and hybrid DE and PSO. Finally,the performance of the CSM-based multi-operator search strategy is compared with other multi-operator search techniques proposed in the literature.

### A. Benchmark Functions

In this work, thirty benchmark functions, which are widely used in the literature [7], [11], [12], [13], [14][2], are selected as the test suite. The test suite contains $9$ unimodal functions ($f_{01} - f_{09}$) and $21$ multi-modal functions ($f_{10} - f_{30}$). These functions are briefly described in Table S-I in the supplementary file[3]. All functions are minimized with $n = 30$.

The maximal number of function evaluations (Max_NFEs) for all benchmark problems are set to be $100,000$. To compare the results of different algorithms, each function is optimized over $50$ independent runs. We use the same set of initial random populations to evaluate different algorithms, that is, all of the compared algorithms are started from the same initial population in each of the $50$ runs.

### B. Contributions of Surrogate Model Components

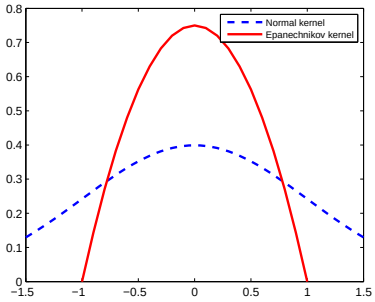
This section studies the contributions of the surrogate model components, including the kernel, the selection technique, and the ranking strategy. As an example, we choose the reproduction operators from JADE [8], [50] as the multiple search operators: "DE/current-to-$p$best/1" without archive, "DE/current-to-$p$best/1" with archive, "DE/rand-to-$p$best/1" without archive, and "DE/rand-to-$p$best/1" with archive. In these strategies, the archive-based strategies can improve the diversity of the population; while the current-based strategies can perform the local search around the current vector, and hence, they can improve the convergence of the algorithm.

[2]Note that in the 30 functions, most of them are also used in other literature, such as 12 functions used in [7], 21 functions used in [14].

[3]Due to the tight space limitation, some tables are provided in the supplementary file, which will be available online.

TABLE I
Average Rankings of CSM-JADE Variants (Friedman)

| Algorithm | Ranking |
|---|---|
| CSM-JADE1 | 3.2000 |
| CSM-JADE2 | 3.0667 |
| CSM-JADE3 | **1.9500** |
| CSM-JADE4 | **1.7833** |




(a)

Fig. 2. Comparison of the Epanechnikov kernel and the normal kernel.

*1) Influence of Kernel and Selection:* For comparison study, the CSM-based JADE (CSM-JADE, for short) is compared with the following four CSM-JADE variants:

- CSM-JADE1: CSM-JADE with normal kernel and tournament selection;
- CSM-JADE2: CSM-JADE with Epanechnikov kernel and tournament selection;
- CSM-JADE3: CSM-JADE with normal kernel and greedy selection;
- CSM-JADE4: CSM-JADE with Epanechnikov kernel and greedy selection.

For these four methods, the parameter settings are set to be the same as used in JADE [8], that is, $\mu = 100$, $p = 0.1$, and $c = 0.1$. Based on the average results obtained by these methods, their average rankings by the Friedman's test[4] are summarized in Table I for all functions. From the results, we can observe that:

- CSM-JADE4 provides the best ranking in the four variants.
- For the same kernel, the greedy selection gets better results than the tournament selection. The reason might be that the greedy selection can always pursue the maximal density of the candidate, while due to the randomness in the tournament selection it cannot always select the candidate with the maximal density.
- The Epanechnikov kernel is able to obtain better results than the normal kernel under the same selection in CSM-JADE. The reason is that the Epanechnikov kernel can pay more concentration than the normal kernel when $|u| \leq 1$ as shown in Fig. 2.

Based on these observations, in the following experiments, the Epanechnikov kernel with the greedy selection will be used in the CSM-based multi-operator search.

[4]The statistical tests used in this work are calculated by the KEEL software [51].

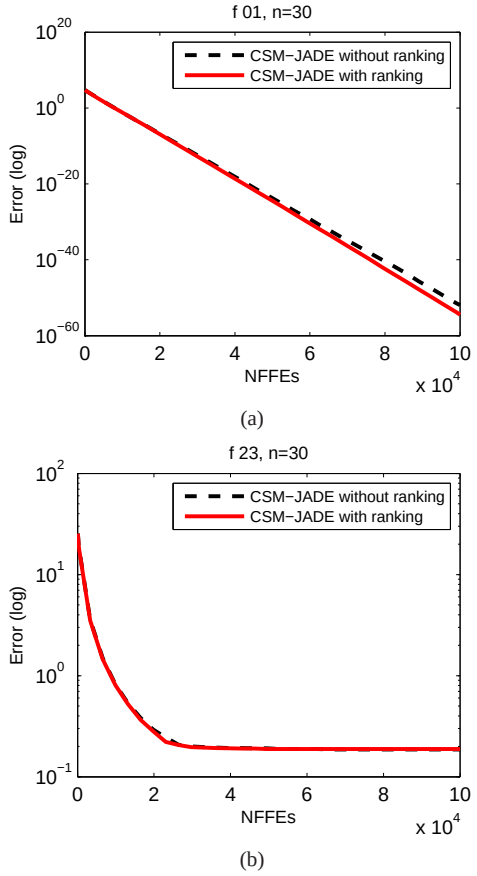



Fig. 3. Convergence curves of CSM-JADE and CSM-JADE-nonranking for the selected functions. (a): $f_{01}$; (b): $f_{23}$.

*2) Influence of Ranking:* Compared with Equation (2), the solution ranks are used to make the better solutions contribute more to the density in Equation (3). In this section, the influence of ranking in the CSM is empirically evaluated. CSM-JADE is compared with CSM-JADE-nonranking, in which only Equation (2) is used to estimate the density. In terms of mean values, the results reveal that there are no significant differences in 27 out of 30 functions based on the Wilcoxon test at $\alpha = 0.05$. Only in three functions $f_{01}, f_{03}, f_{24}$, CSM-JADE significantly outperforms CSM-JADE-nonranking. However, with the ranking strategy, CSM-JADE converges slightly faster than CSM-JADE-nonranking as shown in Fig. 3.

Therefore, we can conclude that in the CSM the kernel $\varphi(u)$ plays the main role in density estimation. Additionally, the ranking strategy can slightly improve the performance of the CSM because of the selection pressure on better solutions.

### C. Study on CSM-based JADE

To verify the improved performance of CSM-JADE[5], it is compared with four JADE variants (that is, JADE1: "DE/current-to-$p$best/1" without archive; JADE2: "DE/current-to-$p$best/1" with archive; JADE3: "DE/rand-to-$p$best/1" without archive; and JADE4: "DE/rand-to-$p$best/1" with archive) and JADE-r. In JADE-r, the four JADE mutation strategies are randomly selected for each target parent to generate the offspring. For all of the six algorithms, the parameter settings are kept unchanged as used in JADE [8].

The average and standard deviation values of the objective value of each function are described in Table S-II in the supplementary file, where the overall best and the second best values are highlighted in **gray boldface** and **boldface**, respectively. Note that similar to the reported results in [8], when several algorithms obtain almost the same global optimum for a function, the *intermediate* results are also reported in Table S-II with NFEs $= 10,000$, such as for functions $f_{05}$, $f_{09}$, etc. For these functions, the averaged intermediate results are used for the statistical test. In Table S-II, "$+$", "$=$", and "$-$" indicate that CSM-JADE respectively performs significantly better than, almost the same as, and significantly worse than the compared algorithm by the Wilcoxon test at $\alpha = 0.05$. In addition, the results of the multiple problem analysis [52] by the Wilcoxon test are also shown in Table II, where $R^+$ and $R^-$ mean the sum of ranks that CSM-JADE performs better than and worse than its competitor, respectively. According to the results in Table S-II, it is clearly observed that CSM-JADE significantly outperforms the compared algorithms in the majority of the functions. In 22 out of 30 functions, it is able to provide the best average results among the six algorithms. With respect to the multiple problem analysis, the results in Table II indicate that CSM-JADE performs on the whole significantly better than the other five JADE variants.

TABLE II
MULTIPLE PROBLEM ANALYSIS BY THE WILCOXON TEST BETWEEN CSM-JADE AND ITS COMPETITORS.

| | $R^+$ | $R^-$ | $p$-value |
|---|---|---|---|
| JADE1 | 358.0 | 77.0 | **0.0017** |
| JADE2 | 354.0 | 81.0 | **0.0023** |
| JADE3 | 359.0 | 76.0 | **0.0015** |
| JADE4 | 390.5 | 74.5 | **0.0026** |
| JADE-r | 366.0 | 69.0 | **8.09E-4** |

TABLE III
AVERAGE RANKINGS OF CSM-JADE, JADE1 - JADE4, AND JADE-R (FRIEDMAN)

| Algorithm | Ranking |
|---|---|
| JADE1 | 4.4500 |
| JADE2 | **2.9667** |
| JADE3 | 4.7500 |
| JADE4 | 3.1167 |
| JADE-r | 3.8333 |
| CSM-JADE | **1.8833** |

TABLE IV
$p$-VALUES OBTAINED FOR BONFERRONI-DUNN'S, HOLM'S, AND HOCHBERG'S PROCEDURES BETWEEN CSM-JADE AND ITS COMPETITORS.

| | $z$ | Unadjusted | Bonferroni-Dunn | Holm | Hochberg |
|---|---|---|---|---|---|
| JADE1 | 5.3135 | **0.00E+00** | **1.00E-06** | **0.00E+00** | **0.00E+00** |
| JADE2 | 2.2427 | **2.49E-02** | 1.25E-01 | **2.49E-02** | **2.49E-02** |
| JADE3 | 5.9346 | **0.00E+00** | **0.00E+00** | **0.00E+00** | **0.00E+00** |
| JADE4 | 2.5532 | **1.07E-02** | 5.34E-02 | **2.13E-02** | **2.13E-02** |
| JADE-r | 4.0369 | **5.40E-05** | **2.71E-04** | **1.62E-04** | **1.62E-04** |

As suggested in [53], besides the Wilcoxon test, other nonparametric tests are also important to comprehensively

[5]In the following sections, CSM-JADE4 is used. For brevity, it is denoted as CMS-JADE.

understand the performance of the proposed method. In Table III, the average rankings by the Friedman's test for the six algorithms are shown. Obviously, the proposed CSM-JADE gets the best average ranking.

Meanwhile, the $p$-values of the Bonferroni-Dunn's, Holm's, and Hochberg's procedures between CSM-JADE and its competitors on the objective values of all functions are also reported in Table IV. CSM-JADE obtains consistently better results compared with the other five JADE variants by the Holm's and Hochberg's procedures. In addition, it also outperforms JADE1, JADE3, and JADE-r by using the Bonferroni-Dunn's procedure.

TABLE V
THE AVERAGE FUNCTION EVALUATIONS AND NUMBER OF SUCCESSFUL RUNS TO ACHIEVE $f < 1.0 \times 10^{-8}$ ON THE FUNCTIONS FOR JADE2, CMS-JADE: THE FIRST VALUE IS THE AVERAGE FUNCTION VALUES IN TERMS OF $10^5$, AND THE BRACKETED VALUE IS THE NUMBER OF SUCCESSFUL RUNS.

| Prob | JADE2 | CSM-JADE |
|---|---|---|
| f01 | 0.27 (50) | **0.22** (50) |
| f02 | 0.45 (50) | **0.35** (50) |
| f03 | 0.98 (5) | **0.69** (50) |
| f05 | 0.10 (50) | **0.08** (50) |
| f07 | 0.09 (50) | **0.08** (50) |
| f08 | 0.89 (18) | **0.64** (48) |
| f09 | 0.18 (50) | **0.14** (50) |
| f11 | NA (0) | **0.99** (2) |
| f13 | 0.41 (50) | **0.33** (50) |
| f14 | 0.28 (50) | **0.23** (50) |
| f15 | 0.24 (50) | **0.19** (50) |
| f16 | 0.29 (50) | **0.23** (50) |
| f18 | NA (0) | **0.79** (4) |
| f25 | 0.22 (50) | **0.18** (50) |
| f26 | 0.66 (49) | **0.46** (50) |
| f30 | 0.60 (50) | **0.40** (50) |

Additionally, the convergence curves of the six JADE algorithms are plotted in Fig. S-1 in the supplementary file. Note that the convergence curves show the *median* error performance of the best solution over the total runs. From Fig. S-1, we clearly observe that in the majority of the functions CSM-JADE gets the fastest convergence speed among the six algorithms. To further investigate the run-time performance, the average function evaluations and number of successful runs to achieve $f < 1.0 \times 10^{-8}$ on the functions in Table S-I are reported in Table V. When no algorithm can get a successful run in the function, the results are not presented in Table V. The CSM-based JADE is compared with JADE2 (the second best JADE variant shown in Table III). From Table V, it can be clearly observed that the CSM-JADE consistently requires less average function evaluations to achieve $f < 1.0 \times 10^{-8}$ in the successful functions than JADE2. In addition, it can also provide higher successful runs in overall. Therefore, the results confirm that the CSM-based multi-operator search strategy is able to decrease the average function evaluations for a given acceptance criterion.

In this section, the capability of CSM-based JADE has been verified on benchmark functions at $n = 30$, it is also fruitful to demonstrate its performance on lower dimensional problems. To this aim, the functions shown in Table S-I at $n = 1$ and $n = 2$ are evaluated on JADE and CSM-JADE. At $n = 1$, functions $f_{10}, f_{19}$, and $f_{20}$ are not defined. The Max_NFEs $= 3,000$ and $6,000$ for functions at $n = 1$ and $n = 2$, respectively. According to the Wilcoxon test at $\alpha = 0.05$[6], CSM-JADE yields the results with $w/t/l = 21/6/0$ and $w/t/l = 28/2/0$ compared with JADE at $n = 1$ and $n = 2$, respectively. The results indicate that CSM-JADE obtains significant better results than JADE in the majority of functions. Therefore, CSM-JADE still works very well on lower dimensional problems.

To sum up, we can conclude that our proposed CSM-based multi-operator search strategy improves the performance of single mutation based JADE significantly. The cheap surrogate model is capable of estimating the density of different JADE mutation operators correctly and promoting the *cooperation* of these operators in the majority of the functions, and hence, it can select the most suitable mutation operator for a specific problem or in different running stages. It is the reason that CSM-JADE provides the best results in 22 test functions compared with other five JADE variants.

### *D. CSM with More Operators*

In the above section, we have combined the CSM with JADE. It is to be expected that the CSM can also work with other EAs. This section studies the generality of the CSM by applying it to more operators.

*1) CSM-based OLPSO:* In this section, CSM is integrated into an advanced PSO variant, that is, OLPSO [9]. OLPSO is an orthogonal learning based PSO, where two variants are proposed, that is, OLPSO-G and OLPSO-L. In this section, CMS-OLPSO is presented, in which both OLPSO-G and OLPSO-L are used to generate the candidate points for each parent particle. The results of CSM-OLPSO are compared with those of OLPSO-G, OLPSO-L, and OLPSO-r. OLPSO-r is similar to JADE-r, where a randomly selected strategy between OLPSO-G and OLPSO-L is used to generate the offspring. All of the four OLPSO variants use the same parameter settings as presented in [9], such as $\mu = 40$, $c = 2.0$, $G = 5$, etc.

The detailed results for all functions are tabulated in Table S-III in the supplementary file. Note that when several algorithms obtain the near global optimum for a function, the intermediate results at NFEs $= 20,000$ are also reported. In Table S-III, "$p$-value1", "$p$-value2", and "$p$-value3" indicate the $p$-values calculated by the Wilcoxon test in CSM-OLPSO vs OLPSO-G, CSM-OLPSO vs OLPSO-L, and CSM-OLPSO vs OLPSO-r, respectively. The summarized results of the multiple problem analysis by the Wilcoxon test, the average rankings by the Friedman's test, and the $p$-values of different statistical procedures are reported in Tables VI, VII, and VIII, respectively. In Table S-III, "$w$", "$t$", and "$l$" mean CSM-OLPSO wins in $w$ functions, ties in $t$ functions, and loses in $l$ functions, compared with its competitors.

Table S-III clearly shows that in the majority of the functions CSM-OLPSO significantly outperforms OLPSO-G, OLPSO-L, and OLPSO-r by the Wilcoxon test at $\alpha = 0.05$. In 22 out of 30 test functions, CSM-OLPSO gets the best mean fitness values among these four algorithms.

[6]For the sake of brevity, we do not report the detail results in the paper. Interested readers can contact the authors for more details.

TABLE VI
MULTIPLE PROBLEM ANALYSIS BY THE WILCOXON TEST BETWEEN CSM-OLPSO AND ITS COMPETITORS.

| | $R^+$ | $R^-$ | $p$-value |
|---|---|---|---|
| OLPSO-G | 371.0 | 67.0 | **0.0025** |
| OLPSO-L | 343.0 | 95.0 | **0.0229** |
| OLPSO-r | 435.0 | 3.0 | **7.45E-08** |

TABLE VII
AVERAGE RANKINGS OF CSM-OLPSO, OLPSO-G, OLPSO-L, OLPSO-R (FRIEDMAN)

| Algorithm | Ranking |
|---|---|
| OLPSO-G | 2.7167 |
| OLPSO-L | **2.4000** |
| OLPSO-r | 3.2500 |
| CMS-OLPSO | **1.6333** |

TABLE VIII
$p$-VALUES OBTAINED FOR BONFERRONI-DUNN'S, HOLM'S, AND HOCHBERG'S PROCEDURES BETWEEN CSM-OLPSO AND ITS COMPETITORS.

| | $z$ | Unadjusted | Bonferroni-Dunn | Holm | Hochberg |
|---|---|---|---|---|---|
| OLPSO-G | 3.2500 | **1.15E-03** | **3.46E-03** | **2.31E-03** | **2.31E-03** |
| OLPSO-L | 2.3000 | **2.14E-02** | **6.43E-02** | **2.14E-02** | **2.14E-02** |
| OLPSO-r | 4.8500 | **1.00E-06** | **4.00E-06** | **4.00E-06** | **4.00E-06** |

From Table VI, we can see that CSM-OLPSO provides significantly better results than OLPSO-G, OLPSO-L, and OLPSO-r by the Wilcoxon test at $\alpha = 0.05$ based on the multiple problem analysis.

The results in Table VII indicate that CSM-OLPSO obtains the first ranking, followed by OLPSO-L, OLPSO-G, and OLPSO-r. According to the results in Table VIII, it is clearly observed that CSM-OLPSO gets significantly better results compared with OLPSO-G, OLPSO-L, and OLPSO-r in all of the three statistical procedures.

Therefore, the above results indicate that the CMS-based multi-operator search strategy is also able to significantly improve the performance of OLPSO. In addition, the performance of OLPSO-r is the worst one among the four OLPSO variants, which means that the inappropriate integration of different operators may cause the algorithm to perform worse than the single operator based methods.

*2) CSM-based CoBiDE:* In this section, we evaluate the capability of our proposal for the multi-crossover search in EAs. Based on this consideration, the CoBiDE method proposed in [10] is used. In CoBiDE, to establish an appropriate coordinate system, covariance matrix learning is presented for the crossover operator. To achieve a good tradeoff between diversity and convergence, the original DE binomial crossover and the crossover based on covariance matrix learning are combined and controlled by a parameter $pb$ in CoBiDE [10]. Based on our proposed method, the CSM-CoBiDE method is presented, where the two kinds of crossover operators are all used to generate the candidate points, then one of them is selected based on its density. In this way, in CSM-CoBiDE the parameter $pb$ is not needed. In addition, two variants of CoBiDE only with one crossover are also used for comparison purpose, that is, CoBiDE1 with covariance matrix learning, and CoBiDE2 with DE binomial crossover. To make a fair comparison, the parameter settings of the four algorithms are set as used in the original CoBiDE method, that is, $\mu = 60, ps = 0.5, pb = 0.4$ [10]. The summarized results are presented in Table IX, X, and XI.

TABLE IX
SINGLE AND MULTIPLE PROBLEM ANALYSIS BY THE WILCOXON TEST BETWEEN CSM-COBIDE AND ITS COMPETITORS.

| | Single problem analysis | | | Multiple problem analysis | | |
|---|---|---|---|---|---|---|
| | $w$ | $t$ | $l$ | $R^+$ | $R^-$ | $p$-value |
| CoBiDE | 26 | 1 | 3 | 386.5 | 78.5 | **3.59E-03** |
| CoBiDE1 | 27 | 2 | 1 | 434.5 | 30.5 | **1.64E-05** |
| CoBiDE2 | 26 | 2 | 2 | 412.5 | 52.5 | **2.90E-04** |

TABLE X
AVERAGE RANKINGS OF CSM-COBIDE, COBIDE, COBIDE1, AND COBIDE2 (FRIEDMAN)

| Algorithm | Ranking |
|---|---|
| CoBiDE | 2.8667 |
| CoBiDE1 | 3.0667 |
| CoBiDE2 | **2.7167** |
| CSM-CoBiDE | **1.3500** |

TABLE XI
$p$-VALUES OBTAINED FOR BONFERRONI-DUNN'S, HOLM'S, AND HOCHBERG'S PROCEDURES BETWEEN CSM-COBIDE AND ITS COMPETITORS.

| | $z$ | Unadjusted | Bonferroni-Dunn | Holm | Hochberg |
|---|---|---|---|---|---|
| CoBiDE | 4.5500 | **5.00E-06** | **1.60E-06** | **1.10E-06** | **1.10E-06** |
| CoBiDE1 | 5.1500 | **0.00E+00** | **1.00E-06** | **1.00E-06** | **1.00E-06** |
| CoBiDE2 | 4.1000 | **4.10E-05** | **1.24E-04** | **4.10E-05** | **4.10E-05** |

The results in Table IX reveal that CSM-CoBiDE can provide significantly better results than CoBiDE, CoBiDE1, and CoBiDE2 in the majority of the functions. Moreover, according to the multiple problem analysis by the Wilcoxon test, it is clear that CSM-CoBiDE is significantly better than the three competitors. For the average rankings reported in Table X, the results exhibit that CSM-CoBiDE gets the first ranking, followed by CoBiDE2, CoBiDE, and CoBiDE1. According to the $p$-values for the Bonferroni-Dunn's, Holm's, and Hochberg's procedures in Table XI, CSM-CoBiDE obtains significantly better results than CoBiDE, CoBiDE1, and CoBiDE2.

Based on the above observations, we can conclude that the CSM-based multi-operator search strategy is also capable of selecting a more suitable crossover operator for the multi-crossover search techniques when solving different problems.

*3) CSM-based Hybrid DE and PSO:* This section addresses the ensemble of different types of reproduction operators. For this purpose, we develop a hybrid DE and PSO method (CSM-DE-PSO), where two DE operators ("DE/rand/1/bin" and "DE/rand/2/bin") and one PSO operator (OLPSO-L) are used to generate different candidates. According to the classification on hybrid DE and PSO [54], CSM-DE-PSO is the collaboration-based DE-PSO, where DE operators act on the personal best positions, but PSO operators act on the current positions. CSM-DE-PSO is compared with two DEs (that is, DE/rand/1/bin and DE/rand/2/bin), OLPSO-L, and DE-PSO-r. In DE-PSO-r, one of the three operators is randomly chosen to generate the offspring for each parent. For the five algorithms,

the population size $\mu = 40$. A self-adaptive parameter setting proposed in [55] for $CR$ and $F$ is used in the DE variants. For the PSO variants, the parameters are set to be the same as presented in OLPSO [9]. The results are summarized in Tables XII, XIII, and XIV.

TABLE XII
SINGLE AND MULTIPLE PROBLEM ANALYSIS BY THE WILCOXON TEST BETWEEN CSM-DE-PSO AND ITS COMPETITORS.

| | Single problem analysis | | | Multiple problem analysis | | |
|---|---|---|---|---|---|---|
| | $w$ | $t$ | $l$ | $R^+$ | $R^-$ | $p$-value |
| DE/rand/1/bin | 19 | 3 | 8 | 288.5 | 207.5 | $\geq$0.2 |
| DE/rand/2/bin | 21 | 3 | 6 | 329.5 | 166.5 | $\geq$0.2 |
| OLPSO-L | 27 | 1 | 2 | 439.0 | 26.0 | **1.99E-06** |
| DE-PSO-r | 25 | 2 | 3 | 409.5 | 86.5 | **0.0037** |

TABLE XIII
AVERAGE RANKINGS OF CSM-DE-PSO, DE/RAND/1/BIN, DE/RAND/2/BIN, OLPSO-L, AND DE-PSO-R (FRIEDMAN)

| Algorithm | Ranking |
|---|---|
| DE/rand/1/bin | **2.2419** |
| DE/rand/2/bin | 3.0645 |
| OLPSO-L | 4.0806 |
| DE-PSO-r | 3.4677 |
| CSM-DE-PSO | **2.1452** |

TABLE XIV
$p$-VALUES OBTAINED FOR BONFERRONI-DUNN'S, HOLM'S, AND HOCHBERG'S PROCEDURES BETWEEN CSM-DE-PSO AND ITS COMPETITORS.

| | $z$ | Unadjusted | Bonferroni-Dunn | Holm | Hochberg |
|---|---|---|---|---|---|
| DE/rand/1/bin | 0.2410 | 8.10E-01 | 3.24E+00 | 8.10E-01 | 8.10E-01 |
| DE/rand/2/bin | 2.2892 | **2.21E-02** | 8.83E-02 | **4.41E-02** | **4.41E-02** |
| OLPSO-L | 4.8193 | **1.00E-06** | **6.00E-06** | **6.00E-06** | **6.00E-06** |
| DE-PSO-r | 3.2932 | **9.91E-04** | **3.96E-03** | **2.97E-03** | **2.97E-03** |

In Table XII, CSM-DE-PSO can significantly outperform the four algorithms in the majority of the functions in terms of the single problem analysis by the Wilcoxon test at $\alpha = 0.05$. With respect to the multiple analysis by the Wilcoxon test, Table XII also indicates that CSM-DE-PSO gets better results. Especially, it performs significantly better than OLPSO-L and DE-PSO-r by the Wilcoxon test at $\alpha = 0.05$.

Considering the average rankings by the Friedman's test, Table XIII shows that CSM-DE-PSO obtains the first ranking, followed by DE/rand/1/bin, DE/rand/2/bin, DE-PSO-r, and OLPSO-L.

From the results in Table XIV, we can observe that CSM-DE-PSO consistently performs significantly better than OLPSO-L and DE-PSO-r in the Bonferroni-Dunn's, Holm's, and Hochberg's procedures. It is also better than DE/rand/2/bin in the Holm's and Hochberg's procedures. There are no significant differences between CSM-DE-PSO and DE/rand/1/bin based on the $p$-values of the Bonferroni-Dunn's, Holm's, and Hochberg's procedures.

In general, when different types of reproduction operators are combined, the CSM-based multi-operator search strategy is still able to obtain very promising results. It can promote the collaboration of different operators, and hence, CSM-DE-PSO can get the best overall performance.

### *E. Comparison with Other Multi-operator Search Techniques*

In the previous sections, it has been shown that the proposed search strategy is beneficial to improve the performance of the single operator based algorithms, and can obtain better results than the algorithms with a randomly selected operator. In this section, the performance of CSM-JADE is compared with other multi-operator selection techniques: PM-JADE [6] and AP-JADE [6]. PM-JADE is a multi-operator based JADE with probability matching technique, while AP-JADE uses the adaptive pursuit technique to assign the selection probabilities for different operators. In addition, since CoDE [16] is a prominent and simple DE variant with multiple mutation operators, our approach is also integrated into CoDE (the resulting algorithm is referred to as CSM-CoDE) to evaluate the enhanced performance when comparing with CoDE.

*1) Results on JADE variants:* For CSM-JADE and JADE-r, the parameter settings are the same as presented in [8]. For PM-JADE and AP-JADE, the parameters presented in [6] are used. Note that for these four JADE variants, the population size $\mu = 100$. The results are reported in Tables XV, XVI, and XVII.

TABLE XV
SINGLE AND MULTIPLE PROBLEM ANALYSIS BY THE WILCOXON TEST BETWEEN CSM-JADE AND ITS COMPETITORS.

| | Single problem analysis | | | Multiple problem analysis | | |
|---|---|---|---|---|---|---|
| | $w$ | $t$ | $l$ | $R^+$ | $R^-$ | $p$-value |
| JADE-r | 26 | 2 | 2 | 391.5 | 73.5 | **0.0023** |
| PM-JADE | 25 | 1 | 4 | 366.5 | 98.5 | **0.0162** |
| AP-JADE | 20 | 4 | 6 | 308.0 | 130.0 | 0.1624 |

TABLE XVI
AVERAGE RANKINGS OF CSM-JADE, JADE-R, PM-JADE, AND AP-JADE (FRIEDMAN)

| Algorithm | Ranking |
|---|---|
| JADE-r | 3.3667 |
| PM-JADE | 2.5500 |
| AP-JADE | **2.3500** |
| CSM-JADE | **1.7333** |

TABLE XVII
$p$-VALUES OBTAINED FOR BONFERRONI-DUNN'S, HOLM'S, AND HOCHBERG'S PROCEDURES BETWEEN CSM-JADE AND ITS COMPETITORS.

| | $z$ | Unadjusted | Bonferroni-Dunn | Holm | Hochberg |
|---|---|---|---|---|---|
| JADE-r | 4.9000 | **1.00E-06** | **3.00E-06** | **3.00E-06** | **3.00E-06** |
| PM-JADE | 2.4500 | **1.43E-02** | **4.29E-02** | **2.86E-02** | **2.86E-02** |
| AP-JADE | 1.8500 | 6.43E-02 | 1.93E-01 | 6.43E-02 | 6.43E-02 |

Table XV shows that CSM-JADE significantly outperforms JADE-r, PM-JADE, and AP-JADE in 26, 25, and 20 functions, respectively. In addition, in terms of the multiple problem analysis by the Wilcoxon test, the results indicate that CSM-JADE is significantly better than JADE-r and PM-JADE. Although CSM-JADE and AP-JADE are not significantly different, CSM-JADE is better than AP-JADE with respect to the sum of ranks.

The average rankings in terms of the fitness values shown in Table XVI show that CSM-JADE is the most effective algorithm among these four JADE variants.

According to the $p$-values for the Bonferroni-Dunn's, Holm's, and Hochberg's procedures in Table XVII, CSM-JADE obtains significantly better results than JADE-r and PM-JADE. It is also better than AP-JADE.

*2) Results on CoDE variants:* For CoDE and CSM-CoDE, the parameter settings are kept unchanged as used in the original CoDE literature [16]. There are three mutation operators in CoDE, that is, "DE/rand/1/bin", "DE/rand/2/bin", and "DE/current-to-rand/1". In CSM-CoDE, each of them will generate a candidate point for each parent, and then one of the three points is sampled by the CSM technique as the offspring. The results of CoDE and CSM-CoDE are provided in Table S-IV in the supplementary file, where the better results are highlighted in **boldface**. The $p$-values by the Wilcoxon test are also reported in the table. When both CoDE and CSM-CoDE obtain the near global optimum for a function, the intermediate results at NFEs $= 20,000$ are also reported.

From Table S-IV, it is clearly observed that in 26 out of 30 functions CSM-CoDE significantly outperforms CoDE based on the Wilcoxon test at $\alpha = 0.05$. In addition, according to the multiple problem analysis by the Wilcoxon test, $R^+ = 373.0$, $R^- = 92.0$, and $p = 0.0030$ when comparing CSM-CoDE with CoDE, which mean that CSM-CoDE performs on the whole significantly better than CoDE.

In summary, our proposed CSM-based multi-operator search strategy is able to obtain better performance than the compared multi-operator search techniques. Thus, we can expect that it may be an effective alternative to implement the ensemble of different operators for other EAs.

### *F. Performance on CEC 2013 Test Suite*

Above, the experiments are conducted on the functions presented in Table S-I. To better understand the performance of our proposal, the problems presented in CEC 2013 [15] are chosen as the test suite. In this test suite, there are 28 functions, where $c_{01} - c_{05}$ are unimodal functions, $c_{06} - c_{20}$ are basic multimodal functions, and $c_{21} - c_{28}$ are composition functions. In this section, all functions are tested at $n = 30$, and the Max_NFEs $= 300,000$. Since many highly-rotated functions are contained in the test suite, we evaluate the performance of CSM-CoDE and CSM-CoBiDE. The reason for choosing the two methods is that there are rotation-invariant operations in CoDE and CoBiDE, such as "DE/current-to-rand/1" mutation in CoDE [16] and covariance matrix learning in CoBiDE [10]. In addition, the proposed search strategy is still implemented into SHADE [56], which obtained very promising results in the CEC 2013 test suite [57]. In SAHDE [56], the "DE/current-to-$p$best/1" with archive is used to generate the mutant. In CSM-SHADE, another mutation strategy, that is, "DE/rand-to-$p$best/1" with archive, is also used. The results are reported in Table S-V in the supplementary file.

Comparing the results between CoDE and CSM-CoDE, CSM-CoDE significantly outperforms CoDE in 11 functions. In 15 functions there are no significant differences between them. In 2 functions $c_{06}$ and $c_{07}$, CoDE gets significantly better results than CSM-CoDE. According to the multiple problem analysis by the Wilcoxon test, $R^+ = 218.5, R^- = 187.5$, and $p > 0.2$ between CSM-CoDE and CoDE, which suggests that CSM-CoDE is on the whole slightly better than CoDE.

With respect to the results between CoBiDE and CSM-CoBiDE, Table S-V shows that CSM-CoBiDE provides significantly better results than CoBiDE in 15 functions, ties in 10 functions, and loses in 3 functions ($c_{06}, c_{23}$, and $c_{24}$). The results of the multiple problem analysis by the Wilcoxon test are $R^+ = 251.0, R^- = 155.0$, and $p > 0.2$ between CSM-CoBiDE and CoBiDE, which mean that CSM-CoBiDE can still obtain slightly better results than CoBiDE on the whole.

Table S-V also shows that CSM-SHADE wins in 11 functions, ties in 15 functions, and loses in 2 functions compared with SHADE in terms of the Wilcoxon test at $\alpha = 0.05$. The results of the multiple problem analysis by the Wilcoxon test are $R^+ = 293.5, R^- = 112.5$, and $p = 0.09574$ between CSM-SHADE and SHADE, which reveal that CSM-SHADE can obtain on the whole significantly better results than SHADE at $\alpha = 0.1$. Fig. S-2 and Fig. S-3 in the supplementary file also indicate that in the majority of functions CSM-SHADE converges faster than SHADE.

TABLE XVIII
AVERAGE RANKINGS OF IDE, CODE, CSM-CODE, COBIDE, CSM-COBIDE, SHADE, AND CSM-SHADE (FRIEDMAN)

| Algorithm | Ranking |
|---|---|
| IDE [58] | **3.3571** |
| CoDE | 4.6964 |
| CSM-CoDE | 4.5536 |
| CoBiDE | 4.3929 |
| CSM-CoBiDE | 4.2321 |
| SHADE [57] | 3.6786 |
| CSM-SHADE | **3.0893** |

In terms of the average rankings by the Friedman test, besides the above six algorithms, another recently presented DE variant with individual-dependent mechanism (IDE [58]) is also included for comparison in the CEC 2013 test suite. The results are reported in Table XVIII. Table XVIII reveals that each CSM-based method obtains better average ranking than its corresponding non-CSM-based method. Additionally, it is clear that CSM-SHADE gets the first ranking, followed by IDE and SHADE. Therefore, the results confirm that the proposed CSM-based multi-operator search strategy is beneficial to the performance enhancement of EAs.

According to the results, we also see that the shifted problems do not increase in difficulty in our proposal; however, the highly-rotated conditions may lead to difficulty when solving these problems. In addition, although the CSM-based multi-operator search strategy is still able to improve the performance of CoDE and CoBiDE in the CEC 2013 test suite, the improvements of CSM-CoDE and CSM-CoBiDE are not so significant when comparing the improvements in the functions presented in Table S-I. The reasons might be two-fold: (a) In this work, we only employ the simple and cheap surrogate model, which may not estimate the density exactly, especially for complex problems with a highly-rotated landscape. (b) In CoDE and CoBiDE, the rotation-invariant operators may not track the rotated landscape properly (such as the selected top solutions for covariance matrix learning in CoBiDE), in this way, the CSM-based multi-operator search strategy cannot pursue the right search direction.

TABLE XIX
ALGORITHM COMPLEXITY OF COBIDE AND CSM-COBIDE

| CoBiDE | | | | |
|---|---|---|---|---|
| | $T0$ | $T1$ | $T2$ | $(\hat{T}2 - T1)/T0$ |
| $n = 10$ | 0.0930 | 0.8110 | 1.0234 | 2.2839 |
| $n = 30$ | | 2.4020 | 3.3726 | 10.4366 |
| $n = 50$ | | 3.9620 | 6.8266 | 30.8022 |

| CSM-CoBiDE | | | | |
|---|---|---|---|---|
| | $T0$ | $T1$ | $T2$ | $(\hat{T}2 - T1)/T0$ |
| $n = 10$ | 0.0930 | 0.8110 | 3.1324 | 24.9613 |
| $n = 30$ | | 2.4020 | 9.1766 | 72.8452 |
| $n = 50$ | | 3.9620 | 17.3784 | 144.2624 |

### G. On the Algorithm Complexity

In this section, as an example, the algorithm complexity of CoBiDE and CSM-CoBiDE is calculated for function $c_{14}$ in the CEC 2014 test suite [15] according to the detailed instructions in [15]. Table XIX gives the computed algorithm complexity on $n = 10, 30$, and $50$. In Table XIX, $T0$ is calculated by running the test program below:

```
for i = 1:1000000
  x = 0.55+(double)i; x = x+x; x = x./2;
  x = x*x; x = sqrt(x); x = log(x);
  x = exp(x); y = x/x;
end
```

$T1$ is the computation time to evaluate $200,000$ evaluations just for $c_{14}$ of a certain dimension $n$, and $T2$ is the computation time for an algorithm with $200,000$ evaluations of the same $n$ dimension for function $c_{14}$. $\hat{T}2$ is the mean $T2$ values of $5$ independent runs.

According to the results shown in Table XIX, for both CoBiDE and CSM-CoBiDE, $T1$ and $\hat{T}2$ scale linearly with the number of dimensions, since $(\hat{T}2-T1)/T0$ grows linearly. In addition, the additional computation time of CSM-CoBiDE also increases linearly when compared with CoBiDE.

## V. CONCLUSIONS AND FUTURE WORK

In the family of evolutionary computation, different reproduction operators have been proposed during the last few decades. Generally, different operators are suitable for different problems or in different running stages. In order to solve a wide range of problems, in this paper, we proposed a novel multi-operator search technique based on the CSMs. In contrast to previous multi-operator search techniques, in our approach, each operator generates its own candidate point. In this way, no operator will be lost when it comes to generating new candidate points in subsequent running stages. Another advantage of our approach is that it is generic, it can implement an ensemble of different operators for different EAs.

To evaluate the performance of our approach, comprehensive experiments (including validation of different kernels and selection techniques, CSM-based JADE, CSM-based OLPSO, CSM-based CoBiDE, CSM-based DE-PSO, and comparison with other multi-operator search techniques) were conducted through benchmark functions. The results indicate that in the majority of the functions, CSM-based algorithms obtain significantly better results than their corresponding single operator based algorithms. The reason might be that in many cases our approach will estimate the density accurately, and hence it can promote the collaboration among different operators.

By carefully looking at the results shown in Table S-II, CSM-JADE provides the worst results in functions $f_{10}$ and $f_{21}$ among the six algorithms. It might be that the CSM used in this work is not able to estimate the density of these functions accurately. In our future work, we will try to use other CSMs, such as the approximated multivariate kernel density estimation [59], to develop more sophisticated CSM-based multi-operator search strategy. Applying CSM to multiobjective optimization [60] is another work worth future investigation.

## ACKNOWLEDGMENTS

The authors would like to thank Dr. Y. Wang for providing the source code of CoBiDE. The source code of SHADE is obtained from Prof. P. N. Suganthan's webpage. The first author would also like to thank Mr. J. Xu for the implementation of CSM-CoBiDE.

## REFERENCES

[1] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford, UK: Oxford University Press, 1996.
[2] R. Storn and K. Price, "Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces," *J. of Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec 1997.
[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of IEEE International Conference on Neural Networks IV*, 1995, pp. 1942 – 1948.
[4] Y.-S. Ong and A. J. Keane, "Meta-Lamarckian learning in memetic algorithms," *IEEE Trans. on Evol. Comput.*, vol. 8, no. 2, pp. 99–110, Apr 2004.
[5] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. on Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr 2009.
[6] W. Gong, A. Fialho, Z. Cai, and H. Li, "Adaptive strategy selection in differential evolution for numerical optimization: An empirical study," *Information Sciences*, vol. 181, no. 24, pp. 5364 – 5386, 2011.
[7] C. Li, S. Yang, and T. T. Nguyen, "A self-learning particle swarm optimizer for global optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 43, no. 3, pp. 627 – 646, 2012.
[8] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. on Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct 2009.
[9] Z.-H. Zhan, J. Zhang, Y. Li, and Y.-H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Trans. on Evol. Comput.*, vol. 15, no. 6, pp. 832 – 847, 2011.
[10] Y. Wang, H.-X. Li, T. Huang, and L. Li, "Differential evolution based on covariance matrix learning and bimodal distribution parameter setting," *Applied Soft Computing*, vol. 18, pp. 232–247, 2014.
[11] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. on Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul 1999.
[12] Y.-W. Leung and Y. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization," *IEEE Trans. on Evol. Comput.*, vol. 5, no. 1, pp. 41–53, Feb 2001.
[13] M. M. Ali, C. Khompatraporn, and Z. B. Zabinsky, "A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems," *J. of Global Optim.*, vol. 31, no. 4, pp. 635–672, 2005.
[14] S. Rahnamayan, H. Tizhoosh, and M. Salama, "Opposition-based differential evolution," *IEEE Trans. on Evol. Comput.*, vol. 12, no. 1, pp. 64–79, Feb 2008.
[15] J. J. Liang, B.-Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session and competition on real-parameter optimization," Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China & Nanyang Technological University, Singapore, Tech. Rep., 2013.

[16] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. on Evol. Comput.*, vol. 15, no. 1, pp. 55–66, 2011.

[17] Á. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Trans. on Evol. Comput.*, vol. 3, no. 2, pp. 124–141, Jul 1999.

[18] J. M. Whitacre, "Adaptation and self-organization in evolutionary algorithms," Ph.D. dissertation, School of Chemical Sciences and Engineering, The University of New South Wales, 2007.

[19] H. J. C. Barbosa and A. M. e Sá, "On adaptive operator probabilities in real coded genetic algorithms," in *Workshop on Advances and Trends in Artificial Intelligence for Problem Solving (SCCC '00)*, Santiago, Chile, 2000.

[20] T.-P. Hong, H.-S. Wang, and W.-C. Chen, "Simultaneously applying multiple mutation operators in genetic algorithms," *Journal of Heuristics*, vol. 6, no. 4, pp. 439–455, 2000.

[21] D. Thierens, "An adaptive pursuit strategy for allocating operator probabilities," in *Proc. Genetic Evol. Comput. Conf.*, 2005, pp. 1539–1546.

[22] H. Dong, J. He, H. Huang, and W. Hou, "Evolutionary programming using a mixed mutation strategy," *Information Sciences*, vol. 177, no. 1, pp. 312 – 327, 2007.

[23] Y. Wang, B. Li, T. Weise, J. Wang, B. Yuan, and Q. Tian, "Self-adaptive learning based particle swarm optimization," *Information Sciences*, vol. 181, no. 20, pp. 4515 – 4538, 2011.

[24] J. A. Vrugt and B. A. Robinson, "Improved evolutionary optimization from genetically adaptive multimethod search," *Proc. of the National Academy of Sciences*, vol. 104, no. 3, pp. 708–711, 2007.

[25] J. Vrugt, B. Robinson, and J. Hyman, "Self-adaptive multimethod search for global optimization in real-parameter spaces," *IEEE Trans. on Evol. Comput.*, vol. 13, no. 2, pp. 243–259, Apr. 2009.

[26] F. Peng, K. Tang, G. Chen, and X. Yao, "Population-based algorithm portfolios for numerical optimization," *IEEE Trans. on Evol. Comput.*, vol. 14, no. 5, pp. 782 – 800, Oct. 2010.

[27] G. Magyar, M. Johnsson, and O. Nevalainen, "An adaptive hybrid genetic algorithm for the three-matching problem," *IEEE Trans. on Evol. Comput.*, vol. 4, no. 2, pp. 135–146, Jul. 2000.

[28] M. Hu, T. Wu, and J. D. Weir, "An intelligent augmentation of particle swarm optimization with multiple adaptive methods," *Information Sciences*, vol. 213, no. 0, pp. 68 – 83, 2012.

[29] M. Hu, T. Wu, and J. Weir, "An adaptive particle swarm optimization with multiple adaptive methods," *IEEE Trans. on Evol. Comput.*, vol. 17, no. 5, pp. 705–720, 2013.

[30] Y.-S. Ong, M.-H. Lim, N. Zhu, and K.-W. Wong, "Classification of adaptive memetic algorithms: A comparative study," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 1, pp. 141–152, 2006.

[31] X. Chen, Y.-S. Ong, M.-H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Trans. on Evol. Comput.*, vol. 15, no. 5, pp. 591–607, 2011.

[32] W. Gong, Z. Cai, C. X. Ling, and H. Li, "Enhanced differential evolution with adaptive strategies for numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part: Cybernetics*, vol. 41, no. 2, pp. 397–413, 2011.

[33] R. Mallipeddi, P. Suganthan, Q. Pan, and M. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, no. 2, pp. 1679–1696, 2011.

[34] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Multi-operator based evolutionary algorithms for solving constrained optimization problems," *Computers & Operations Research*, vol. 38, no. 12, pp. 1877 – 1896, 2011.

[35] J.-H. Zhong, M. Shen, J. Zhang, H.-H. Chung, Y.-H. Shi, and Y. Li, "A differential evolution algorithm with dual populations for solving periodic railway timetable scheduling problem," *IEEE Trans. on Evol. Comput.*, vol. 17, no. 4, pp. 512–527, 2013.

[36] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing*, vol. 9, no. 1, pp. 212–221, 2005.

[37] ——, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.

[38] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with gaussian process model," *IEEE Trans. on Evol. Comput.*, vol. 14, no. 3, pp. 456–474, 2010.

[39] B. Liu, Q. Zhang, and G. Gielen, "A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Trans. on Evol. Comput.*, vol. PP, no. 99, pp. 1–1, 2013.

[40] R. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions," *IEEE Trans. on Evol. Comput.*, vol. PP, no. 99, pp. 1–1, 2013.

[41] A. Brownlee, J. McCall, and Q. Zhang, "Fitness modeling with markov networks," *IEEE Trans. on Evol. Comput.*, vol. 17, no. 6, pp. 862–879, 2013.

[42] Y.-S. Ong, P. Nair, and K. Lum, "Max-min surrogate-assisted evolutionary algorithm for robust design," *IEEE Trans. on Evol. Comput.*, vol. 10, no. 4, pp. 392–404, 2006.

[43] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Trans. on Evol. Comput.*, vol. 14, no. 3, pp. 329–355, 2010.

[44] S. Koos, J.-B. Mouret, and S. Doncieux, "The transferability approach: Crossing the reality gap in evolutionary robotics," *IEEE Trans. on Evol. Comput.*, vol. 17, no. 1, pp. 122–145, 2013.

[45] J. Knowles and H. Nakayama, "Meta-modeling in multiobjective optimization," in *Multiobjective Optimization*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, vol. 5252, pp. 245–284.

[46] L. Zhou, A. Zhou, G. Zhang, and C. Shi, "An estimation of distribution algorithm based on nonparametric density estimation," in *2011 IEEE Congress on Evolutionary Computation (CEC)*, 2011, pp. 1597–1604.

[47] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification, 2nd Edition*. John Wiley & Sons, Inc., 2001.

[48] W. Gong and Z. Cai, "Differential evolution with ranking-based mutation operators," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2066 – 2081, 2013.

[49] V. Epanechnikov, "Non-parametric estimation of a multivariate probability density," *Theory of Probability & Its Applications*, vol. 14, no. 1, pp. 153–158, 1969.

[50] J. Zhang and A. C. Sanderson, *Adaptive Differential Evolution: A Robust Approach to Multimodal Problem Optimization*. Berlin: Springer-Verlag, 2009.

[51] J. Alcalá-Fdez, L. Sánchez, and S. García, "KEEL: A software tool to assess evolutionary algorithms to data mining problems," 2012. [Online]. Available: http://www.keel.es/

[52] S. García and F. Herrera, "An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons," *Journal Of Machine Learning Research*, vol. 9, pp. 2677–2694, 2008.

[53] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization," *Journal of Heuristics*, vol. 15, no. 6, pp. 617 – 644, 2009.

[54] B. Xin, J. Chen, J. Zhang, H. Fang, and Z.-H. Peng, "Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: A review and taxonomy," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 5, pp. 744–767, 2012.

[55] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. on Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec 2006.

[56] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *2013 IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 71 – 78.

[57] ——, "Evaluating the performance of SHADE on CEC 2013 benchmark problems," in *2013 IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 1952 – 1959.

[58] L. Tang, Y. Dong, and J. Liu, "Differential evolution with an individual-dependent mechanism," *IEEE Trans. on Evol. Comput.*, 2014, in press.

[59] M. Kristan, A. Leonardis, and D. Skočaj, "Multivariate online kernel density estimation with gaussian kernels," *Pattern Recognition*, vol. 44, no. 10 - 11, pp. 2630 – 2642, 2011.

[60] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

**Wenyin Gong** received the B.Eng., M.Eng., and PhD degrees in computer science from China University of Geosciences, Wuhan, China, in 2004, 2007, and 2010, respectively.

He is an Associate Professor with School of Computer Science, China University of Geosciences, Wuhan, China. His research interests include evolutionary algorithms, evolutionary optimization, and their applications. He has published over 30 research papers in journals and international conferences.

He served as a referee for over 10 international journals, such as IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON CYBERNETICS, *IEEE Computational Intelligence Magazine*, *ACM Transactions on Intelligent Systems and Technology*, *Information Sciences*, *European Journal of Operational Research*, *Applied Soft Computing*, *International Journal of Hydrogen Energy*, and so on.

**Aimin Zhou** (S'08-M'10) received the B.Sc. and M.Sc. degrees in computer science from Wuhan University, Wuhan, China, in 2001 and 2003, respectively, and the Ph.D. degree in computer science from University of Essex, Colchester, U.K., in 2009.

He is an Associate Professor with the Shanghai Key Laboratory of Multidimensional Information Processing, and the Department of Computer Science and Technology, East China Normal University, Shanghai, China. His research interests include evolutionary computation and optimization, machine learning, image processing, and their applications. He has published over 40 peer-reviewed papers.

**Zhihua Cai** received the Bsc degree from Wuhan University, China, in 1986, the Msc degree from Beijing University of Technology, China, in 1992, and the PhD degree from China University of Geosciences, in 2003.

He is currently a faculty member at School of Computer Science, China University of Geosciences, Wuhan, China. His main research areas include data mining, machine learning, evolutionary computation, and their applications. He has published over 50 research papers.

# SUPPLEMENTARY FILE FOR "A MULTI-OPERATOR SEARCH STRATEGY BASED ON CHEAP SURROGATE MODELS FOR EVOLUTIONARY OPTIMIZATION"

TABLE S-I

THE 30 BENCHMARK FUNCTIONS USED IN OUR EXPERIMENTAL STUDY, WHERE $n$ IS THE NUMBER OF VARIABLES AND $S \subseteq \mathbb{R}^n$. ALL OF THE FUNCTIONS ARE MINIMIZED, AND THE MINIMAL VALUE OF EACH FUNCTION IS 0 AT $n = 30$.

| Prob. | $S$ |
|---|---|
| $f_{01} = \sum_{i=1}^{n} x_i^2$ | $[-100, 100]^n$ |
| $f_{02} = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$ | $[-10, 10]^n$ |
| $f_{03} = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$ | $[-100, 100]^n$ |
| $f_{04} = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$ | $[-100, 100]^n$ |
| $f_{05} = \sum_{i=1}^{n-1} (\lfloor x_i + 0.5 \rfloor)^2$ | $[-100, 100]^n$ |
| $f_{06} = \sum_{i=1}^{n} x_i^4 + \text{rndreal}[0, 1)$ | $[-1.28, 1.28]^n$ |
| $f_{07} = \sum_{i=1}^{n} \lvert x_i \rvert^{i+1}$ | $[-1, 1]^n$ |
| $f_{08} = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5 i x_i)^2 + (\sum_{i=1}^{n} 0.5 i x_i)^4$ | $[-5, 10]^n$ |
| $f_{09} = 1 - \exp\left(0.5 \sum_{i=1}^{n} x_i^2\right)$ | $[-1, 1]^n$ |
| $f_{10} = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | $[-30, 30]^n$ |
| $f_{11} = \sum_{i=1}^{n} (-x_i \sin(\sqrt{\lvert x_i \rvert})) + 418.98288727243369 \times n$ | $[-500, 500]^n$ |
| $f_{12} = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$ | $[-5.12, 5.12]^n$ |
| $f_{13} = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + \exp(1)$ | $[-32, 32]^n$ |
| $f_{14} = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$ | $[-600, 600]^n$ |
| $f_{15} = \frac{\pi}{n}\left\{10\sin^2(\pi y_i) + \sum_{i=1}^{n-1}(y_i - 1)^2 \cdot [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ | $[-50, 50]^n$ |
| $f_{16} = \frac{1}{10}\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$ | $[-50, 50]^n$ |
| $f_{17} = \sum_{i=1}^{n} \sin(x_i)\left(\sin\left(\frac{i x_i^2}{\pi}\right)\right)^{20} + 29.6308838503244$ | $[0, \pi]^n$ |
| $f_{18} = \sum_{i=1}^{n} \lvert x_i \sin x_i + 0.1 x_i \rvert$ | $[-10, 10]^n$ |
| $f_{19} = \sum_{i=1}^{n-1}\left(0.5 + \frac{\sin^2\left(\sqrt{100x_i^2 + x_{i+1}^2}\right) - 0.5}{1 + 0.001(x_i^2 - 2x_i x_{i+1} + x_{i+1}^2)^2}\right)$ | $[-100, 100]^n$ |
| $f_{20} = -\sum_{i=1}^{n-1}\left(\exp\left(\frac{-(x_i^2 + x_{i+1}^2 + 0.5 x_i x_{i+1})}{8}\right) \times \cos\left(4\sqrt{x_i^2 + x_{i+1}^2 + 0.5 x_i x_{i+1}}\right)\right)$ | $[-5, 5]^n$ |
| $f_{21} = \sum_{i=1}^{n}(x_i - 1)^2 + \sum_{i=2}^{n} x_i x_{i-1} + \frac{n(n+4)(n-1)}{6}$ | $[-n^2, n^2]^n$ |
| $f_{22} = -\left(1 + \frac{0.2k}{K + 0.1}\right)\cos(K\pi)\exp(-\frac{K}{2\pi}) + 1.143833$<br>$k = \sqrt{\sum_{i=1}^{n}(x_i - b_i)^2}$<br>$K = \sqrt{n}(\max\lvert x_i - b_i \rvert)$ | $[-15, 15]^n$ |
| $f_{23} = 1 - \cos(2\pi \parallel x \parallel) + 0.1 \parallel x \parallel$, where $\parallel x \parallel = \sqrt{\sum_{i=1}^{n} x_i^2}$ | $[-100, 100]^n$ |
| $f_{24} = -\left(A\prod_{i=1}^{n}\sin(x_i - z) + \prod_{i=1}^{n}\sin(B(x_i - z))\right) + A + 1$, where $A = 2.5$, $B = 5$, $z = 30$ | $[0, 180]^n$ |
| $f_{25} = 0.1n - 0.1 * \sum_{i=1}^{n}\cos(5\pi x_i) + \sum_{i=1}^{n} x_i^2$ | $[-1, 1]^n$ |
| $f_{26} = \sum_{i=1}^{n}\left(\sum_{k=0}^{k_{\max}}\left[a^k\cos(2\pi b^k(x_i + 0.5))\right]\right) - n\sum_{k=0}^{k_{\max}}\left[a^k\cos(\pi b^k)\right]$, where $a = 0.5$, $b = 3$, $k_{\max} = 20$ | $[-0.5, 0.5]^n$ |
| $f_{27} = 10n + \sum_{i=1}^{n}[y_i^2 + 10\cos(2\pi y_i)]$<br>$y_i = \begin{cases} x_i, & \lvert x_i \rvert < 0.5 \\ \text{round}(2x_i)/2, & \text{otherwise} \end{cases}$ | $[-100, 100]^n$ |
| $f_{28} = -0.0001\left(\left\lvert\left(\prod_{i=1}^{n}\sin x_i\right)\exp\left(\left\lvert 100 - \frac{\sqrt{\sum_{i=1}^{n} x_i^2}}{\pi}\right\rvert\right)\right\rvert + 1\right)^{0.1} + 1.68347744296512$ | $[-10, 10]^n$ |
| $f_{29} = -\sum_{i=1}^{n}\sin x_i \sin^{20}\left(\frac{i x_i^2}{\pi}\right) + 29.6308838503244$ | $[0, \pi]^n$ |
| $f_{30} = \frac{1}{n}\sum_{i=1}^{n}\left(x_i^4 - 16x_i^2 + 5x_i\right) + 78.3323314075429$ | $[-5, 5]^n$ |

TABLE S-II
COMPARISON ON THE RESULTS OF CSM-JADE, JADE1, JADE2, JADE3, JADE4, AND JADE-R IN ALL FUNCTIONS.

| Prob | JADE1 | JADE2 | JADE3 | JADE4 | JADE-r | CSM-JADE |
|---|---|---|---|---|---|---|
| f01 | 2.71E-37 ± 1.55E-36 $^{+}$ | **4.93E-40 ± 3.31E-39** $^{+}$ | 3.72E-37 ± 1.60E-36 $^{+}$ | 3.59E-37 ± 2.53E-36 $^{+}$ | 2.41E-35 ± 1.10E-34 $^{+}$ | **3.19E-55 ± 1.26E-54** |
| f02 | 1.15E-14 ± 8.00E-14 $^{+}$ | **1.13E-16 ± 7.34E-16** $^{+}$ | 5.31E-15 ± 3.38E-14 $^{+}$ | 1.38E-15 ± 8.69E-15 $^{+}$ | 1.20E-16 ± 3.68E-16 $^{+}$ | **1.80E-26 ± 1.14E-25** |
| f03 | 7.53E-09 ± 3.16E-08 $^{+}$ | 7.41E+02 ± 1.78E+03 $^{+}$ | **4.82E-13 ± 9.43E-13** $^{+}$ | 1.59E+03 ± 2.83E+03 $^{+}$ | 1.68E+02 ± 8.57E+02 $^{+}$ | **5.33E-14 ± 1.99E-13** |
| f04 | 7.96E-02 ± 3.10E-02 $^{+}$ | **3.53E-02 ± 1.64E-02** $^{+}$ | 8.74E-02 ± 3.38E-02 $^{+}$ | 4.63E-02 ± 2.51E-02 $^{+}$ | 5.82E-02 ± 3.00E-02 $^{+}$ | **1.37E-03 ± 8.16E-04** |
| f05 | 3.50E+00 ± 1.22E+00 $^{+}$ | **2.60E-01 ± 5.22E-01** $^{+}$ | 5.80E+00 ± 1.77E+00 $^{+}$ | 1.60E+00 ± 1.31E+00 $^{+}$ | 2.68E+00 ± 2.62E+00 $^{+}$ | **2.00E-02 ± 1.40E-01** |
| | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 |
| f06 | 2.08E-03 ± 1.07E-03 $^{+}$ | **1.28E-03 ± 4.49E-04** $^{=}$ | 2.10E-03 ± 7.69E-04 $^{+}$ | 1.62E-03 ± 5.97E-04 $^{+}$ | 1.91E-03 ± 9.37E-04 $^{+}$ | **1.36E-03 ± 6.16E-04** |
| f07 | 1.18E-43 ± 4.84E-43 $^{+}$ | **5.61E-48 ± 3.79E-47** $^{-}$ | 1.22E-39 ± 4.55E-39 $^{+}$ | 1.26E-43 ± 4.03E-43 $^{=}$ | 1.54E-40 ± 7.58E-40 $^{=}$ | **2.67E-44 ± 1.12E-43** |
| f08 | **2.31E-02 ± 1.63E-01** $^{+}$ | 1.68E+01 ± 1.93E+01 $^{+}$ | **9.08E-05 ± 6.42E-04** $^{+}$ | 2.26E+01 ± 2.38E+01 $^{+}$ | 1.03E+01 ± 1.95E+01 $^{+}$ | 3.73E-01 ± 2.59E+00 |
| f09 | 1.31E-04 ± 4.14E-05 $^{+}$ | **4.72E-05 ± 1.52E-05** $^{+}$ | 2.21E-04 ± 6.13E-05 $^{+}$ | 9.11E-05 ± 2.65E-05 $^{+}$ | 1.19E-04 ± 6.58E-05 $^{+}$ | **3.64E-06 ± 1.30E-06** |
| | 2.22E-18 ± 1.55E-17 | 0.00E+00 ± 0.00E+00 | 1.78E-17 ± 4.07E-17 | 0.00E+00 ± 0.00E+00 | 1.33E-17 ± 3.61E-17 | 0.00E+00 ± 0.00E+00 |
| f10 | 3.10E+00 ± 1.21E+00 $^{-}$ | 3.48E+00 ± 1.45E+00 $^{-}$ | **3.21E-01 ± 1.09E+00** $^{-}$ | **8.26E-02 ± 5.64E-01** $^{-}$ | 1.67E+00 ± 1.86E+00 $^{-}$ | 2.68E+01 ± 1.89E+01 |
| f11 | 9.97E-05 ± 4.43E-05 $^{+}$ | **7.88E-07 ± 4.08E-07** $^{+}$ | 1.02E-04 ± 4.23E-05 $^{+}$ | 1.06E-06 ± 4.70E-07 $^{+}$ | 5.43E-05 ± 7.19E-05 $^{+}$ | **1.09E-07 ± 7.88E-08** |
| f12 | 1.17E-04 ± 4.84E-05 $^{+}$ | **1.32E-06 ± 6.22E-07** $^{+}$ | 1.73E-04 ± 6.40E-05 $^{+}$ | 1.87E-06 ± 8.47E-07 $^{+}$ | 7.26E-05 ± 8.46E-05 $^{+}$ | **8.12E-08 ± 6.28E-08** |
| f13 | 1.14E+00 ± 2.48E-01 $^{+}$ | **4.53E-01 ± 1.12E-01** $^{+}$ | 1.54E+00 ± 2.17E-01 $^{+}$ | 7.93E-01 ± 2.16E-01 $^{+}$ | 1.03E+00 ± 4.83E-01 $^{+}$ | **9.69E-02 ± 2.72E-02** |
| | 5.77E-15 ± 1.78E-15 | 4.07E-15 ± 4.97E-16 | 6.63E-15 ± 1.56E-15 | 4.14E-15 ± 6.96E-16 | 4.99E-15 ± 1.60E-15 | 4.07E-15 ± 4.97E-16 |
| f14 | 1.00E+00 ± 3.12E-02 $^{+}$ | **8.48E-01 ± 9.06E-02** $^{+}$ | 1.04E+00 ± 1.29E-02 $^{+}$ | 9.69E-01 ± 5.03E-02 $^{+}$ | 9.72E-01 ± 6.92E-02 $^{+}$ | **2.13E-01 ± 6.91E-02** |
| | 2.32E-18 ± 1.62E-17 | 0.00E+00 ± 0.00E+00 | 1.80E-11 ± 1.26E-10 | 0.00E+00 ± 0.00E+00 | 3.25E-21 ± 1.29E-20 | 0.00E+00 ± 0.00E+00 |
| f15 | 2.04E-01 ± 1.15E-01 $^{+}$ | **3.42E-02 ± 2.30E-02** $^{+}$ | 3.78E-01 ± 1.67E-01 $^{+}$ | 7.21E-02 ± 3.97E-02 $^{+}$ | 1.68E-01 ± 1.44E-01 $^{+}$ | **2.26E-03 ± 2.97E-03** |
| | 1.57E-32 ± 0.00E+00 | 1.57E-32 ± 0.00E+00 | 1.69E-32 ± 6.07E-33 | 1.69E-31 ± 1.07E-30 | 1.57E-32 ± 0.00E+00 | 1.57E-32 ± 0.00E+00 |
| f16 | 4.69E+00 ± 2.16E+00 $^{+}$ | **1.42E+00 ± 5.02E-01** $^{+}$ | 5.69E+00 ± 1.65E+00 $^{+}$ | 2.32E+00 ± 6.29E-01 $^{+}$ | 3.23E+00 ± 1.46E+00 $^{+}$ | **3.32E-01 ± 1.62E-01** |
| | 3.12E-22 ± 2.00E-21 | 1.51E-32 ± 8.43E-33 | 2.02E-21 ± 9.93E-21 | 7.77E-26 ± 5.42E-25 | 8.46E-25 ± 5.92E-24 | 1.35E-32 ± 0.00E+00 |
| f17 | 1.09E+00 ± 1.40E-01 $^{+}$ | 7.33E-01 ± 1.22E-01 $^{+}$ | 1.00E+00 ± 1.36E-01 $^{+}$ | **6.82E-01 ± 1.07E-01** $^{+}$ | 8.47E-01 ± 2.19E-01 $^{+}$ | **5.38E-01 ± 9.45E-02** |
| f18 | 7.32E-04 ± 3.00E-04 $^{+}$ | **7.31E-05 ± 6.12E-05** $^{+}$ | 1.04E-03 ± 2.86E-04 $^{+}$ | 7.79E-05 ± 6.18E-05 $^{+}$ | 4.85E-04 ± 4.92E-04 $^{+}$ | **1.27E-05 ± 2.92E-05** |
| f19 | 6.48E+00 ± 3.30E-01 $^{+}$ | 6.50E+00 ± 3.23E-01 $^{+}$ | **6.45E+00 ± 3.31E-01** $^{+}$ | 6.51E+00 ± 3.68E-01 $^{+}$ | 6.54E+00 ± 2.79E-01 $^{+}$ | **5.97E+00 ± 2.65E-01** |
| f20 | **2.36E+00 ± 5.01E-02** $^{+}$ | 2.38E+00 ± 4.43E-02 $^{+}$ | 2.36E+00 ± 5.30E-02 $^{+}$ | 2.38E+00 ± 4.54E-02 $^{+}$ | 2.36E+00 ± 5.07E-02 $^{+}$ | **2.31E+00 ± 3.39E-02** |
| f21 | 4.71E+00 ± 1.18E+01 $^{-}$ | 2.80E+01 ± 5.93E+01 $^{-}$ | **7.29E-06 ± 2.22E-05** $^{-}$ | **1.84E-07 ± 5.83E-07** $^{-}$ | 3.89E+00 ± 7.48E+00 $^{-}$ | 6.71E+02 ± 2.83E+02 |
| f22 | 1.14E+00 ± 6.38E-04 $^{+}$ | 1.14E+00 ± 4.53E-04 $^{+}$ | 1.14E+00 ± 8.79E-04 $^{+}$ | 1.14E+00 ± 6.45E-04 $^{+}$ | 1.14E+00 ± 4.95E-04 $^{+}$ | 1.14E+00 ± 2.12E-03 |
| f23 | 1.98E-01 ± 3.77E-02 $^{+}$ | **1.73E-01 ± 4.41E-02** $^{=}$ | 2.02E-01 ± 3.77E-02 $^{+}$ | **1.74E-01 ± 4.35E-02** $^{=}$ | 1.92E-01 ± 2.73E-02 $^{+}$ | 1.88E-01 ± 3.28E-02 |
| f24 | 1.36E+00 ± 9.22E-02 $^{=}$ | 1.37E+00 ± 5.98E-02 $^{=}$ | 1.36E+00 ± 7.43E-02 $^{=}$ | 1.35E+00 ± 9.07E-02 $^{=}$ | **1.33E+00 ± 8.95E-02** $^{=}$ | **1.35E+00 ± 8.64E-02** |
| f25 | 2.80E-02 ± 1.17E-02 $^{+}$ | **4.25E-03 ± 2.06E-03** $^{+}$ | 5.84E-02 ± 2.91E-02 $^{+}$ | 9.77E-03 ± 4.07E-03 $^{+}$ | 2.17E-02 ± 1.95E-02 $^{+}$ | **2.64E-04 ± 1.17E-04** |
| | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 |
| f26 | 1.29E-06 ± 6.17E-06 $^{+}$ | **6.20E-10 ± 4.20E-09** $^{+}$ | 9.05E-06 ± 2.20E-05 $^{+}$ | 1.99E-09 ± 8.75E-09 $^{+}$ | 1.90E-06 ± 1.14E-05 $^{+}$ | **0.00E+00 ± 0.00E+00** |
| f27 | 1.25E-01 ± 5.98E-02 $^{+}$ | **8.48E-04 ± 4.58E-04** $^{+}$ | 1.26E-01 ± 5.02E-02 $^{+}$ | 1.06E-03 ± 5.60E-04 $^{+}$ | 6.61E-02 ± 7.62E-02 $^{+}$ | **3.33E-04 ± 2.45E-04** |
| f28 | 1.74E-04 ± 3.54E-05 $^{+}$ | 1.17E-04 ± 2.45E-05 $^{+}$ | 1.78E-04 ± 4.70E-05 $^{+}$ | **1.14E-04 ± 2.75E-05** $^{+}$ | 1.41E-04 ± 4.88E-05 $^{+}$ | **6.88E-05 ± 1.57E-05** |
| f29 | 1.09E+00 ± 1.40E-01 $^{+}$ | 7.33E-01 ± 1.22E-01 $^{+}$ | 1.00E+00 ± 1.36E-01 $^{+}$ | **6.82E-01 ± 1.07E-01** $^{+}$ | 8.47E-01 ± 2.19E-01 $^{+}$ | **5.38E-01 ± 9.45E-02** |
| f30 | 1.19E-10 ± 6.10E-11 $^{+}$ | 1.25E-12 ± 1.91E-12 $^{+}$ | 1.80E-10 ± 1.21E-10 $^{+}$ | **1.03E-12 ± 1.57E-12** $^{+}$ | 7.53E-11 ± 9.29E-11 $^{+}$ | **4.83E-14 ± 5.72E-15** |
| $w/t/l$ | 27/1/2 | 24/3/3 | 27/1/2 | 25/3/2 | 26/2/2 | – |

TABLE S-III
COMPARISON ON THE RESULTS OF CSM-OLPSO, OLPSO-G, OLPSO-L, AND OLPSO-R IN ALL FUNCTIONS.

| Prob | OLPSO-G | OLPSO-L | OLPSO-r | CMS-OLPSO | $p$-value1 | sig1 | $p$-value2 | sig2 | $p$-value3 | sig3 |
|---|---|---|---|---|---|---|---|---|---|---|
| f01 | **1.23E-21 ± 5.04E-21** | 1.71E-19 ± 3.67E-19 | 6.25E-09 ± 2.42E-08 | **8.22E-31 ± 3.64E-30** | 1.78E-15 | + | 1.78E-15 | + | 1.78E-15 | + |
| f02 | 3.63E-10 ± 2.40E-09 | **2.82E-11 ± 9.09E-11** | 4.32E-06 ± 5.10E-06 | **5.56E-15 ± 3.66E-14** | 2.43E-13 | + | 1.78E-15 | + | 1.78E-15 | + |
| f03 | 3.67E+03 ± 2.53E+03 | 5.12E+03 ± 1.58E+03 | **3.08E+03 ± 1.15E+03** | **1.25E+03 ± 4.65E+02** | 1.78E-15 | + | 1.78E-15 | + | 4.44E-14 | + |
| f04 | 4.16E+00 ± 1.97E+00 | 2.53E+00 ± 9.61E-01 | **4.48E-01 ± 2.17E-01** | **2.63E-01 ± 2.13E-01** | 1.78E-15 | + | 1.78E-15 | + | 7.95E-06 | + |
| f05 | 1.60E+03 ± 6.24E+02 | **1.13E+03 ± 2.42E+02** | 2.03E+03 ± 4.70E+02 | **7.45E+02 ± 2.11E+02** | 1.78E-15 | + | 1.24E-14 | + | 1.78E-15 | + |
| | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | | | | | | |
| f06 | 3.45E-02 ± 1.74E-02 | **2.48E-02 ± 5.61E-03** | 3.63E-02 ± 1.28E-02 | **1.38E-02 ± 5.06E-03** | 3.68E-13 | + | 9.77E-14 | + | 8.88E-15 | + |
| f07 | **1.21E-55 ± 6.77E-55** | 9.18E-41 ± 4.84E-40 | 3.60E-47 ± 2.39E-46 | **9.37E-72 ± 6.56E-71** | 1.78E-15 | + | 1.78E-15 | + | 1.78E-15 | + |
| f08 | **4.98E+01 ± 2.40E+01** | 8.83E+01 ± 2.05E+01 | 5.82E+01 ± 1.74E+01 | **2.88E+01 ± 1.05E+01** | 5.22E-07 | + | 1.78E-15 | + | 9.52E-13 | + |
| f09 | 1.69E-19 ± 6.98E-20 | **9.11E-20 ± 5.07E-20** | 1.14E-13 ± 2.81E-13 | **1.04E-19 ± 2.15E-20** | 1.19E-07 | + | 1.83E-01 | = | 1.78E-15 | + |
| f10 | 4.22E+01 ± 3.50E+01 | **1.73E+01 ± 2.48E+01** | 8.59E+01 ± 4.00E+01 | **3.96E+01 ± 3.48E+01** | 5.27E-01 | = | 8.48E-04 | - | 5.76E-08 | + |
| f11 | 5.10E+02 ± 2.06E+02 | **9.00E+01 ± 9.44E+01** | 3.81E+02 ± 2.19E+02 | **2.86E+02 ± 1.82E+02** | 8.11E-07 | + | 3.08E-09 | - | 1.86E-02 | + |
| f12 | 1.15E+00 ± 9.51E-01 | 1.99E-02 ± 1.41E-01 | **6.29E-07 ± 2.62E-06** | **1.73E-20 ± 1.23E-19** | 7.11E-15 | + | 3.55E-15 | + | 1.78E-15 | + |
| f13 | **9.07E-11 ± 9.07E-11** | 1.59E-09 ± 8.17E-10 | 5.02E-05 ± 3.07E-05 | **3.82E-13 ± 4.75E-13** | 1.78E-15 | + | 1.78E-15 | + | 1.78E-15 | + |
| f14 | 7.42E-03 ± 1.35E-02 | **8.23E-14 ± 3.26E-13** | 3.89E-08 ± 2.41E-07 | **2.93E-20 ± 2.94E-20** | 2.27E-13 | + | 1.36E-12 | + | 1.78E-15 | + |
| f15 | **1.21E-22 ± 3.64E-22** | 4.73E-21 ± 1.13E-20 | 1.41E-07 ± 1.40E-07 | **6.08E-25 ± 4.29E-24** | 1.35E-11 | + | 3.55E-15 | + | 1.78E-15 | + |
| f16 | **2.62E-19 ± 1.11E-18** | 7.96E-18 ± 2.92E-17 | 3.72E-06 ± 3.01E-06 | **5.22E-25 ± 1.65E-24** | 3.55E-15 | + | 1.78E-15 | + | 1.78E-15 | + |
| f17 | 3.76E-01 ± 1.83E-01 | **1.60E-01 ± 7.45E-02** | 9.34E-01 ± 3.10E-01 | **1.42E-01 ± 6.32E-02** | 7.94E-13 | + | 4.84E-01 | = | 1.78E-15 | + |
| f18 | **3.11E-11 ± 9.58E-11** | 2.78E-05 ± 2.84E-05 | 5.45E-04 ± 2.03E-04 | **1.39E-08 ± 6.79E-08** | 3.05E-02 | - | 3.55E-15 | + | 1.78E-15 | + |
| f19 | **4.31E+00 ± 7.59E-01** | 7.84E+00 ± 7.24E-01 | 9.00E+00 ± 6.49E-01 | **6.94E+00 ± 1.46E+00** | 4.49E-13 | - | 8.81E-04 | + | 5.71E-12 | + |
| f20 | **2.14E+00 ± 5.98E-02** | 2.19E+00 ± 3.53E-02 | 2.43E+00 ± 1.40E-01 | **2.13E+00 ± 1.28E-02** | 4.78E-01 | = | 9.77E-14 | + | 1.78E-15 | + |
| f21 | 1.04E+03 ± 8.63E+02 | **9.01E+02 ± 6.07E+02** | 2.06E+03 ± 1.02E+03 | **5.29E+02 ± 4.46E+02** | 8.48E-04 | + | 1.19E-03 | + | 5.86E-14 | + |
| f22 | 1.14E+00 ± 4.34E-03 | 1.13E+00 ± 6.79E-03 | **1.08E+00 ± 2.32E-02** | **1.07E+00 ± 2.24E-02** | 1.78E-15 | + | 1.78E-15 | + | 1.43E-03 | + |
| f23 | 9.66E-01 ± 2.54E-01 | 5.89E-01 ± 9.23E-02 | **4.81E-01 ± 8.70E-02** | **2.36E-01 ± 6.15E-02** | 1.78E-15 | + | 1.78E-15 | + | 1.78E-15 | + |
| f24 | **1.80E+00 ± 6.23E-01** | **3.34E+00 ± 1.12E-01** | 3.46E+00 ± 5.51E-02 | 3.36E+00 ± 1.77E-01 | 1.78E-15 | - | 6.49E-02 | = | 8.24E-07 | + |
| f25 | 1.36E+00 ± 2.88E-01 | **1.03E+00 ± 1.58E-01** | 1.51E+00 ± 1.98E-01 | **8.42E-01 ± 1.99E-01** | 1.78E-15 | + | 1.78E-15 | + | 1.78E-15 | + |
| | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | | | | | | |
| f26 | 5.14E-05 ± 1.49E-04 | **1.86E-05 ± 3.74E-05** | 1.25E-03 ± 9.57E-04 | **2.82E-07 ± 1.40E-06** | 2.12E-12 | + | 3.31E-10 | + | 1.78E-15 | + |
| f27 | 4.74E+00 ± 2.31E+00 | **3.63E-01 ± 6.06E-01** | 4.98E-01 ± 1.15E+00 | **1.60E-01 ± 3.70E-01** | 7.11E-15 | + | 3.80E-06 | + | 6.36E-06 | + |
| f28 | 1.82E+01 ± 5.59E+00 | **1.43E+01 ± 2.41E+00** | 2.28E+01 ± 3.99E+00 | **9.57E+00 ± 2.25E+00** | 1.78E-15 | + | 1.78E-15 | + | 1.78E-15 | + |
| | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | | | | | | |
| f29 | **6.60E-05 ± 2.02E-04** | 1.02E-01 ± 1.47E-01 | 2.00E+00 ± 6.27E-01 | **2.89E-04 ± 9.08E-04** | 2.49E-01 | = | 2.43E-13 | + | 1.78E-15 | + |
| f30 | 4.98E+00 ± 2.22E+00 | **2.60E-01 ± 4.87E-01** | 5.94E-01 ± 1.18E+00 | **3.40E-01 ± 6.88E-01** | 5.33E-15 | + | 1.92E-02 | + | 2.07E-02 | + |
| $w/t/l$ | | | | | 24/3/3 | | 25/3/2 | | 30/0/0 | |

TABLE S-IV
COMPARISON ON THE RESULTS OF CSM-CODE AND CODE IN ALL FUNCTIONS.

| Prob | CoDE | | CSM-CoDE | | $p$-value | sig. |
|---|---|---|---|---|---|---|
| | Mean | Std | Mean | Std | | |
| f01 | 1.40E-17 | 1.10E-17 | **3.88E-34** | 6.28E-34 | 1.78E-15 | + |
| f02 | 1.32E-09 | 7.48E-10 | **4.87E-18** | 6.27E-18 | 1.78E-15 | + |
| f03 | 3.84E-03 | 3.53E-03 | **3.00E-07** | 4.20E-07 | 1.78E-15 | + |
| f04 | 3.19E-04 | 1.50E-04 | **8.84E-07** | 7.74E-07 | 1.78E-15 | + |
| f05 | 2.28E+00 | 1.79E+00 | **0.00E+00** | 0.00E+00 | 1.78E-15 | + |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | | |
| f06 | 7.77E-03 | 2.15E-03 | **5.37E-03** | 1.79E-03 | 1.28E-06 | + |
| f07 | 1.71E-42 | 6.84E-42 | **4.12E-72** | 1.83E-71 | 1.78E-15 | + |
| f08 | 1.90E-07 | 2.08E-07 | **2.83E-13** | 5.99E-13 | 1.78E-15 | + |
| f09 | 8.31E-05 | 4.59E-05 | **4.20E-08** | 2.19E-08 | 1.78E-15 | + |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | | |
| f10 | **1.68E+01** | 8.00E+00 | 2.90E+01 | 1.99E+01 | 2.02E-11 | - |
| f11 | 6.03E-08 | 1.91E-07 | **1.09E-13** | 4.36E-13 | 1.78E-15 | + |
| f12 | 1.02E+01 | 3.16E+00 | **4.03E-07** | 2.23E-06 | 1.78E-15 | + |
| f13 | 1.06E-09 | 6.05E-10 | **4.14E-15** | 0.00E+00 | 1.78E-15 | + |
| f14 | 7.90E-14 | 4.19E-13 | **0.00E+00** | 0.00E+00 | 1.78E-15 | + |
| f15 | 5.24E-19 | 6.31E-19 | **1.57E-32** | 0.00E+00 | 1.78E-15 | + |
| f16 | 1.05E-16 | 1.43E-16 | **3.22E-32** | 4.05E-32 | 1.78E-15 | + |
| f17 | 4.01E+00 | 5.61E-01 | **1.65E+00** | 6.06E-01 | 1.78E-15 | + |
| f18 | 5.92E-02 | 1.35E-01 | **1.26E-03** | 5.59E-04 | 1.78E-15 | + |
| f19 | 8.25E+00 | 2.82E-01 | **7.82E+00** | 7.75E-01 | 5.64E-05 | + |
| f20 | 2.77E+00 | 3.33E-01 | **2.14E+00** | 5.29E-02 | 1.78E-15 | + |
| f21 | **2.56E+01** | 5.26E+01 | 2.26E+02 | 1.44E+02 | 1.35E-11 | - |
| f22 | **1.05E+00** | 4.50E-02 | 1.13E+00 | 6.79E-03 | 3.55E-15 | - |
| f23 | **2.06E-01** | 2.03E-02 | 2.10E-01 | 3.03E-02 | 1.98E-03 | - |
| f24 | 2.16E+00 | 1.73E-01 | **2.04E+00** | 2.72E-01 | 1.96E-02 | + |
| f25 | 8.94E-03 | 5.57E-03 | **4.78E-06** | 3.35E-06 | 1.78E-15 | + |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | | |
| f26 | 1.57E-05 | 1.34E-05 | **0.00E+00** | 0.00E+00 | 1.78E-15 | + |
| f27 | 2.33E+01 | 2.20E+00 | **1.04E+01** | 3.98E+00 | 1.78E-15 | + |
| f28 | 1.55E-02 | 3.72E-03 | **1.56E-03** | 6.58E-04 | 1.78E-15 | + |
| f29 | 4.01E+00 | 5.61E-01 | **1.65E+00** | 6.06E-01 | 1.78E-15 | + |
| f30 | 2.27E+00 | 1.06E+00 | **3.30E-03** | 3.55E-03 | 1.78E-15 | + |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | | |
| $w/t/l$ | | | | | 26/0/4 | |

TABLE S-V
EXPERIMENTAL RESULTS OF CODE, CSM-CODE, COBIDE, CSM-COBIDE, SHADE, AND CSM-SHADE IN THE CEC 2013 TEST SUITE.

| Prob | CoDE | CSM-CoDE | CoBiDE | CSM-CoBiDE | SHADE | CSM-SHADE |
|---|---|---|---|---|---|---|
| c01 | 5.91E-22 ± 4.16E-22 | **0.00E+00 ± 0.00E+00** + | 2.43E-10 ± 1.27E-10 | **1.94E-20 ± 1.04E-20** + | 7.44E-18 ± 3.01E-18 | **1.29E-26 ± 9.16E-27** + |
| | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 |
| c02 | 8.73E+04 ± 6.30E+04 | **4.59E+04 ± 1.92E+04** + | 1.47E+05 ± 8.18E+04 | **1.13E+05 ± 6.16E+04** = | 9.00E+03 ± 7.47E+03 | **3.72E+03 ± 2.92E+03** + |
| c03 | 5.68E-02 ± 1.14E-01 | **1.28E-04 ± 1.97E-04** + | 2.45E+00 ± 2.66E+00 | **8.92E-01 ± 1.52E+00** + | 4.02E+01 ± 2.13E+02 | **9.52E-08 ± 2.28E-07** + |
| c04 | 2.57E+05 ± 7.05E+05 | **1.37E+05 ± 2.58E+05** = | **9.64E+03 ± 3.92E+04** | 1.63E+04 ± 6.35E+04 = | **1.92E-04 ± 3.01E-04** | 1.37E+04 ± 6.81E+04 - |
| c05 | 4.83E-15 ± 2.10E-15 | **7.85E-20 ± 3.10E-20** + | 8.10E-07 ± 8.02E-08 | **7.95E-11 ± 3.14E-11** + | 1.29E-10 ± 7.69E-11 | **3.07E-11 ± 1.65E-11** + |
| | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 1.04E-31 ± 1.84E-31 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 |
| c06 | **1.33E+00 ± 5.23E+00** | 5.36E+00 ± 3.33E+00 - | **2.46E+00 ± 8.79E-01** | 7.13E+00 ± 1.78E+00 - | 5.96E-01 ± 3.73E+00 | **5.28E-01 ± 3.28E+00** - |
| c07 | **6.17E+00 ± 6.30E+00** | 1.17E+01 ± 9.96E+00 - | **5.09E+00 ± 3.69E+00** | 6.64E+00 ± 7.43E+00 = | 4.60E+00 ± 5.39E+00 | **1.93E+00 ± 2.03E+00** + |
| c08 | 2.09E+01 ± 6.39E-02 | 2.09E+01 ± 6.83E-02 = | 2.09E+01 ± 6.20E-02 | 2.09E+01 ± 6.03E-02 = | 2.07E+01 ± 1.76E-01 | **2.06E+01 ± 9.31E-02** + |
| c09 | 1.41E+01 ± 3.61E+00 | **1.40E+01 ± 3.44E+00** = | **1.22E+01 ± 3.01E+00** | 1.26E+01 ± 3.25E+00 = | 2.75E+01 ± 1.77E+00 | **2.61E+01 ± 3.27E+00** + |
| c10 | **2.87E-02 ± 2.34E-02** | 3.76E-02 ± 1.67E-02 = | 2.70E-02 ± 1.47E-02 | **2.27E-02 ± 1.45E-02** + | 7.69E-02 ± 3.58E-02 | **6.84E-02 ± 3.43E-02** = |
| c11 | 5.71E+00 ± 2.77E+00 | **1.02E-11 ± 1.58E-11** + | 2.05E+00 ± 9.09E-01 | 2.34E-09 ± 1.09E-09 + | 3.21E+01 ± 2.16E+00 | **1.15E+01 ± 7.97E-01** + |
| | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 | 0.00E+00 ± 0.00E+00 |
| c12 | **3.86E+01 ± 8.80E+00** | 4.35E+01 ± 1.31E+01 = | 4.86E+01 ± 1.21E+01 | **4.84E+01 ± 9.13E+00** = | 2.30E+01 ± 3.73E+00 | **2.22E+01 ± 3.61E+00** = |
| c13 | **7.78E+01 ± 3.18E+01** | 9.64E+01 ± 2.43E+01 = | 8.82E+01 ± 2.81E+01 | **7.96E+01 ± 2.00E+01** = | 5.03E+01 ± 1.34E+01 | **4.28E+01 ± 1.42E+01** = |
| c14 | 3.77E+01 ± 1.40E+01 | **7.21E-01 ± 1.66E+00** + | 3.40E+00 ± 1.55E+00 | **7.87E-01 ± 1.00E+00** + | **3.18E-02 ± 2.33E-02** | 3.58E-02 ± 2.91E-02 = |
| c15 | 3.47E+03 ± 6.51E+02 | **3.34E+03 ± 4.16E+02** = | **2.92E+03 ± 3.65E+02** | 3.10E+03 ± 5.19E+02 = | **3.22E+03 ± 2.64E+02** | 3.28E+03 ± 3.53E+02 = |
| c16 | 1.83E+00 ± 8.99E-01 | **7.41E-01 ± 8.75E-01** + | **8.37E-01 ± 9.69E-01** | 1.05E+00 ± 9.76E-01 = | **9.13E-01 ± 1.85E-01** | 9.79E-01 ± 4.36E-01 = |
| c17 | 3.04E+01 ± 6.57E-03 | 3.04E+01 ± 2.61E-02 + | 3.05E+01 ± 1.18E-02 | **3.04E+01 ± 6.68E-03** + | 3.04E+01 ± 3.83E-14 | 3.04E+01 ± 1.35E-06 + |
| c18 | **6.56E+01 ± 8.58E+00** | 7.55E+01 ± 2.29E+01 = | 7.29E+01 ± 1.07E+01 | **7.03E+01 ± 1.26E+01** + | 7.25E+01 ± 5.58E+00 | **7.06E+01 ± 6.84E+00** + |
| c19 | 3.29E+00 ± 1.22E+00 | **2.47E+00 ± 5.41E-01** + | 2.56E+00 ± 4.74E-01 | **1.75E+00 ± 3.99E-01** + | 1.36E+00 ± 1.20E-01 | **1.35E+00 ± 1.01E-01** = |
| c20 | **1.02E+01 ± 7.98E-01** | 1.06E+01 ± 6.67E-01 = | 1.08E+01 ± 5.89E-01 | **1.03E+01 ± 4.63E-01** + | 1.05E+01 ± 6.04E-01 | **1.03E+01 ± 5.18E-01** = |
| c21 | 3.00E+02 ± 0.00E+00 | 3.00E+02 ± 0.00E+00 = | 3.00E+02 ± 6.54E-13 | 3.00E+02 ± 0.00E+00 + | 3.09E+02 ± 5.65E+01 | **3.00E+02 ± 0.00E+00** = |
| c22 | 2.01E+02 ± 7.65E+01 | **1.13E+02 ± 4.61E+00** + | 1.66E+02 ± 5.58E+01 | **1.11E+02 ± 2.95E+00** + | **9.81E+01 ± 2.52E+01** | 1.06E+02 ± 1.02E+00 = |
| c23 | **3.83E+03 ± 7.77E+02** | 4.06E+03 ± 7.60E+02 = | **3.09E+03 ± 6.40E+02** | 3.61E+03 ± 6.62E+02 - | 3.51E+03 ± 4.11E+02 | **3.50E+03 ± 5.03E+02** = |
| c24 | 2.13E+02 ± 5.21E+00 | 2.13E+02 ± 6.03E+00 = | **2.05E+02 ± 3.29E+00** | 2.08E+02 ± 4.95E+00 - | **2.05E+02 ± 5.29E+00** | 2.06E+02 ± 4.49E+00 = |
| c25 | **2.56E+02 ± 1.85E+01** | 2.58E+02 ± 1.62E+01 = | 2.54E+02 ± 1.28E+01 | **2.46E+02 ± 2.09E+01** + | 2.59E+02 ± 1.96E+01 | **2.54E+02 ± 1.74E+01** + |
| c26 | 2.00E+02 ± 3.32E-03 | 2.00E+02 ± 9.86E-04 + | 2.00E+02 ± 3.62E-03 | 2.00E+02 ± 4.37E-03 + | 2.02E+02 ± 1.48E+01 | 2.02E+02 ± 1.50E+01 = |
| c27 | **5.28E+02 ± 1.28E+02** | 5.65E+02 ± 1.15E+02 = | 5.32E+02 ± 1.08E+02 | **4.13E+02 ± 7.59E+01** + | 3.88E+02 ± 1.09E+02 | **3.46E+02 ± 3.77E+01** = |
| c28 | 3.00E+02 ± 0.00E+00 | 3.00E+02 ± 0.00E+00 = | 3.00E+02 ± 0.00E+00 | 3.00E+02 ± 0.00E+00 = | 3.00E+02 ± 0.00E+00 | 3.00E+02 ± 0.00E+00 = |
| $w/t/l$ | | 11/15/2 | | 15/10/3 | | 11/15/2 |

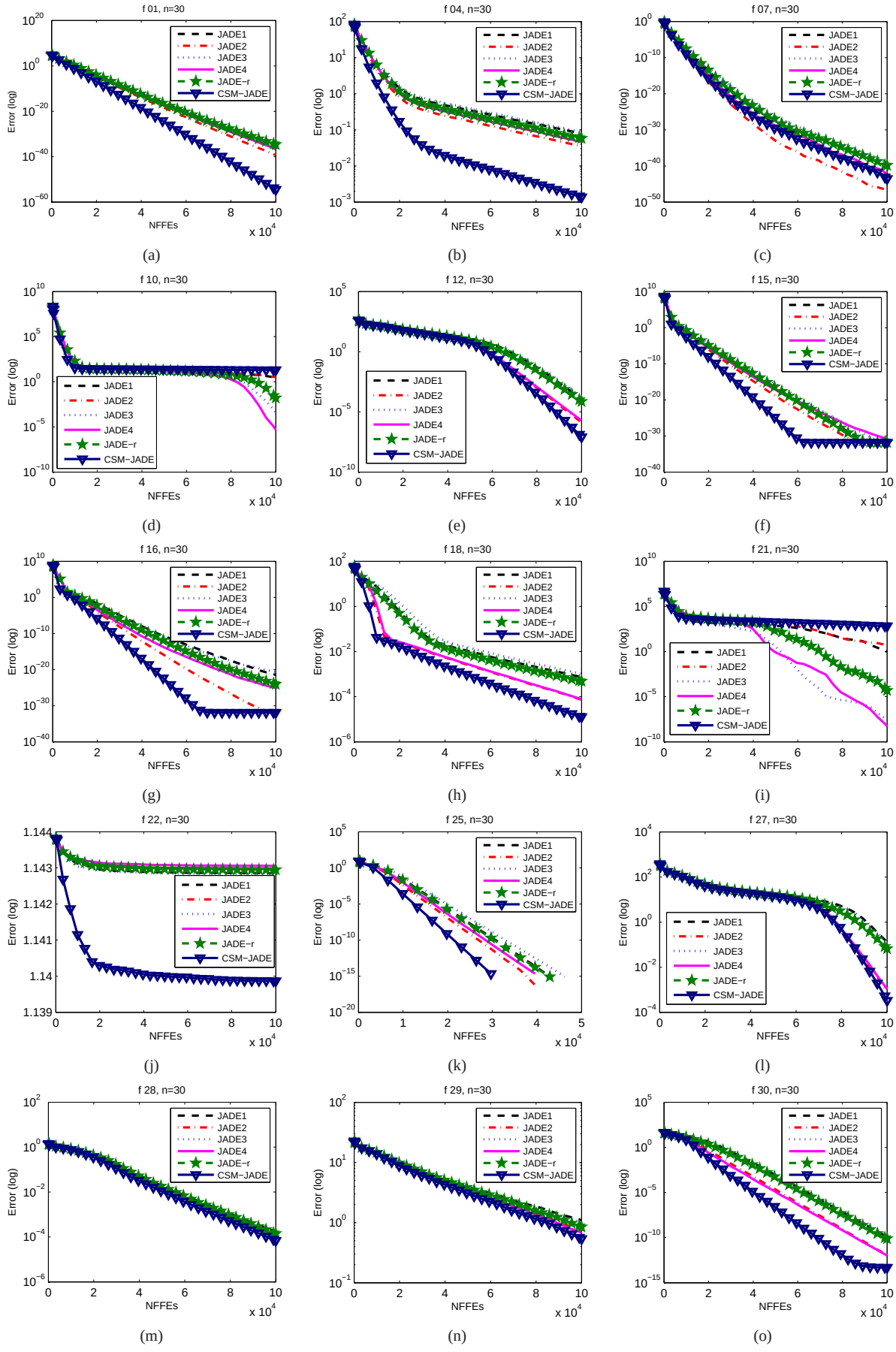



Fig. S-1. Convergence curves of different JADE variants for the selected functions. (a): $f_{01}$; (b): $f_{04}$; (c): $f_{07}$; (d): $f_{10}$; (e): $f_{12}$; (f): $f_{15}$; (g): $f_{16}$; (h): $f_{18}$; (i): $f_{21}$; (k): $f_{22}$; (k): $f_{25}$; (i) - (o): $f_{27}$ - $f_{30}$.

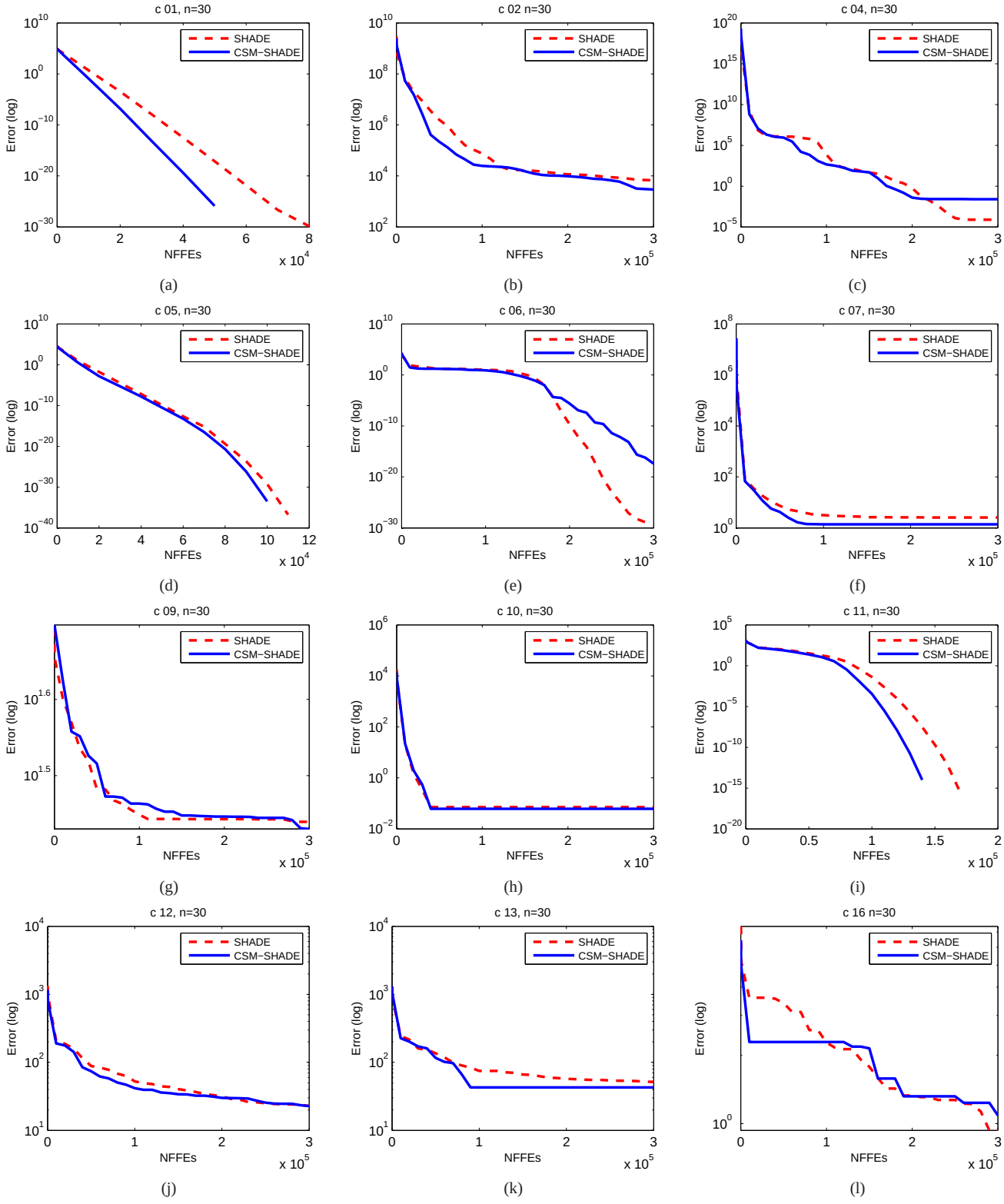



Fig. S-2. Convergence curves of SHADE and CSM-SHADE for the selected functions in the CEC 2013 test suite.

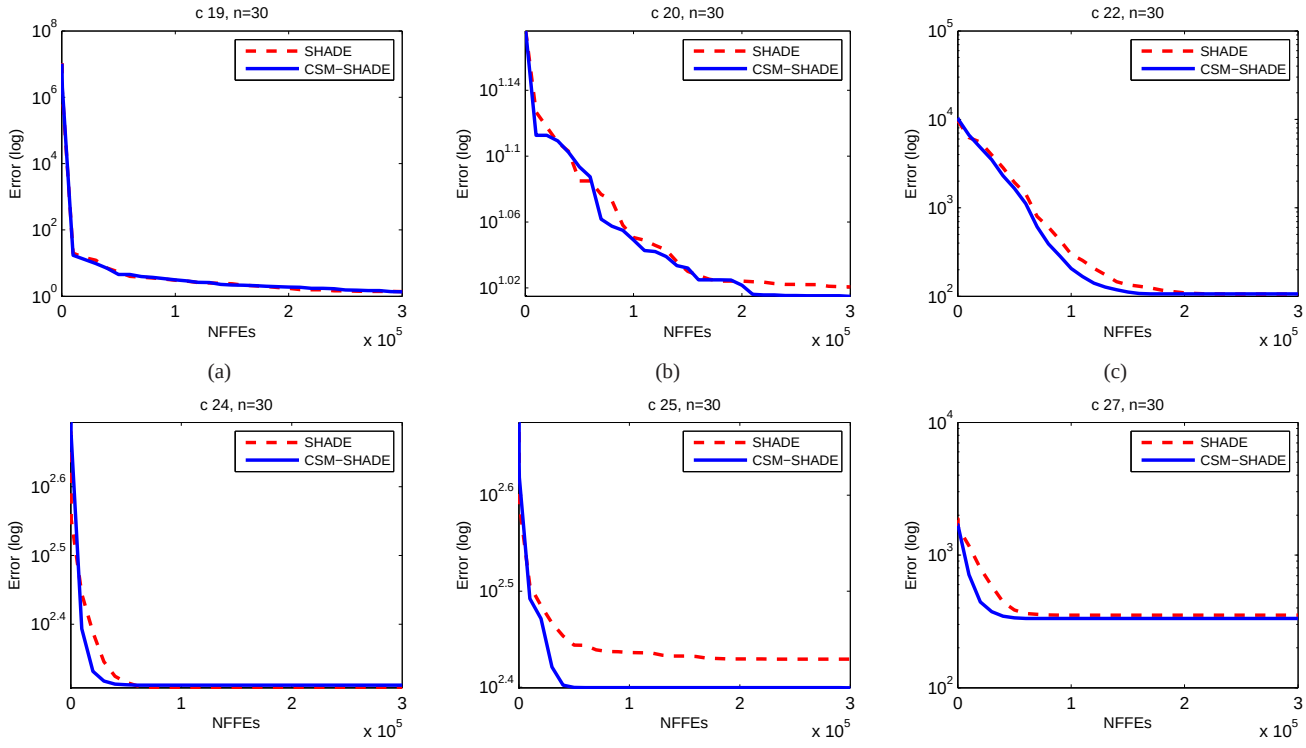



Fig. S-3. Convergence curves of SHADE and CSM-SHADE for the selected functions in the CEC 2013 test suite.