Selective-Candidate Framework with Similarity Selection Rule for Evolutionary Optimization

Sheng Xin Zhang[a*1], Wing Shing Chan[a], Zi Kang Peng[b], Shao Yong Zheng[b*], Kit Sang Tang[a]

[a]Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong
[b]School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, 510006, China

Abstract

Achieving better exploitation and exploration capabilities (EEC) have always been an important yet challenging issue in evolutionary optimization algorithm (EOA) design. The difficulties lie in obtaining a good balance in EEC, which is cooperatively determined by operations and parameters in an EOA. When deficiencies in exploitation or exploration are observed, most existing works only consider supplementing it, either by designing new operations or by altering the parameters. Unfortunately, when different situations are encountered, these proposals may fail to be the winner. To address these problems, this paper proposes an explicit EEC control method named selective-candidate framework with similarity selection rule (SCSS). On the one hand, M (M > 1) candidates are generated from each current solution with independent operations and parameters to enrich the search. While on the other hand, a similarity selection rule is designed to determine the final candidate. By considering the fitness ranking of the current solution and its Euclidian distance to each of these M candidates, superior current solutions select the closest to be the final candidate for efficient local exploitation while inferior ones would favor the farthest candidate for exploration purpose. In this way, the rule is able to synthesize exploitation and exploration, making the evolution more effective. The proposed SCSS framework is general and easy to implement. It has been applied to three classic, four state-of-the-art and four up-to-date EOAs from the branches of differential evolution, evolution strategy and particle swarm optimization. As confirmed with simulation results, significant performance enhancement is achieved.

Keywords: Evolution status, similarity selection, exploitation and exploration, differential evolution (DE), covariance matrix adaptation evolution strategy (CMA-ES), particle swarm optimization (PSO), global optimization.

*Corresponding authors.
 E-mail addresses: shengxinzhang@gmail.com (S. X. Zhang), zhengshaoy@mail.sysu.edu.cn (S. Y. Zheng).

1. Introduction

Constructed on a population basis, evolutionary optimization algorithm (EOA) explores a searching space by iteratively performing genetic operations (for evolutionary algorithms, EAs [1, 2]) or social learning processes (for swarm intelligences, SIs [3]) to generate new solutions. The way how these solutions are sampled, gives the feature of a particular method and also determines its exploitation and exploration capabilities (EEC). For differential evolution (DE) [4-8] and evolution strategy (ES) [9], the genetic operations are mutation and crossover/recombination. While for particle swarm optimization (PSO) [10], the social learning procedures consist of the velocity and position update equations. Commonly, EEC of EOAs is indispensably controlled by the genetic operations/social learning, together with the associated parameters (e.g. mutation and crossover factors in DE, normal distribution in ES and acceleration coefficients in PSO), which cooperatively locate the sampled solutions. Since EEC is the cornerstone of evolutionary optimization [11] and has a direct impact on the performance, researchers had put a lot of effort on designing appropriate exploitation and exploration schemes [12]. Existing works can be summarized under the following three categories.

(1) **EEC controlled by genetic operations/social learning.** Generally, genetic operations/social learning determines the evolution direction. In this category, research works solely focus on genetic operations/social learning. Along this line, various types of operators, such as proximity-based [13], ranking-based [14], multiobjective sorting-based [15], collective information-based [16] mutation, jumping genes-based crossover [17], local best-based [18], orthogonal learning-based [19] and heterogeneous-based [20] velocity update equations were designed, favoring an exploitation or exploration trend. Besides these newly designed operations, EEC has also been controlled by the ensemble of multiple DE mutation strategies [21-25], the combination of different types of optimizers, such as covariance matrix adaptation ES (CMA-ES) [26] and PSO in [27], CMA-ES and DE in [28], CMA-ES, DE and PSO in [29] and the memetic algorithms [30, 31].

(2) **EEC controlled by parameter tuning.** Parameters control the evolution scale. In this category, researchers pursued efficient parameter tuning schemes, that included deterministic and adaptive ones. Population size is a common parameter in evolutionary optimization. Related works include linear population size reduction scheme [32], restart CMA-ES with increasing population size scheme [33] and bi-population restart CMA-ES with dual population size tuning schemes [34]. Apart from population size, extra parameters introduced in a specific algorithm may also need fine-tuning, such as the mutation and crossover factors [35] of DE, the new greediness parameter p of the "current-to-pbest/1" mutation [36], etc.

(3) **EEC controlled by the combination of genetic operations/social learning and parameter tuning.**
There are also some works [37, 38] aimed at simultaneously controlling genetic operations/social learning and parameters. In [37], Mallipeddi et al. proposed to improve DE with an ensemble of parameters and mutation strategies. In [38], Wang et al. proposed to use three different mutation strategies combined with three different pairs of control parameters to generate solutions for selecting the fittest. These methods strike a balance between exploitation and exploration using two steps. The first step maintains a mutation strategy pool with diverse searching characteristics while the second step emphasizes exploitation by fitness-based reward [37] or greedy pre-selection [38]. However, there are some issues that may hinder the performance. On the one hand, both methods are greedy and there is no explicit mechanism to remedy premature convergence. While on the other hand, multiple candidates are evaluated for each current solution [38], resulting in a higher total computation cost.

In this paper, we propose a selective-candidate framework with similarity selection rule (SCSS), which simultaneously considers the operations (i.e. evolution direction) and parameters (i.e. evolution scale) that affect the generation of candidates while addressing the issues in category (3). The features, motivations and contributions of SCSS are summarized as follows.

1) SCSS first generates M ($M > 1$) candidates for each current solution by M independent reproduction procedures. Afterwards, one of them will become the final candidate for each current solution based on a selective rule. The big challenge here is that it should be effective and efficient. On the one hand, it is required to provide a potentially excellent candidate with balanced EEC for next generation, while on the other hand, it should not involve objective function evaluation which requires additional cost. To resolve these issues, a similarity selection (SS) rule based on fitness ranking and Euclidian distance information is designed to strike a balanced EEC while avoiding evaluation of all the candidates.

2) SCSS also considers the fitness ranking of the population, which provides relative location information of individuals. For superior current solutions, the closest candidate measured by Euclidian distance in solution space will be selected as the final candidate for local search (exploitation) purpose. While for inferior ones, the farthest candidate is favored for basin-jumping (exploration) purpose.

3) Based on the above design, the proposed SCSS framework is expected to meet the challenge in 1) and enhance the performance. The main contributions of this work are summarized as follows.

   a) Different algorithms may be suitable for solving different optimization problems [39-41]. This study provides a generic method that is readily applied to different types of EOAs.

   b) The proposed method provides an explicit EEC control paradigm based on fitness and Euclidian distance measures, which is straight-forward, simple and easy-understanding.

   c) Extensive study shows that the proposed method achieves a balanced EEC and consequently

demonstrates remarkable performance enhancement of several start-of-the-art and top algorithms available in the literature [18, 26, 32, 36, 42-46]. In addition, its working mechanism, benefits and real-world applications are also presented and analyzed.

The rest of this paper is organized as follows. Section 2 describes the proposed framework. Section 3 presents the experimental study and relevant discussions, while section 4 concludes this paper.

2 Proposed Method

2.1 Motivations

Generally, the procedures[1] for EAs/SIs can be summarized as **Algorithm 1.** It is common that one candidate is generated from a current solution based on the reproduction procedure. However, due to the stochastic process in operations and randomness in parameters, obtained candidate is not guaranteed to be located within promising searching areas. Obviously, if the reproduction procedure repeats, candidates from the same current solution are likely to be different, bringing up various building blocks, resulting in different searching performance. This is not only observed in classic EAs [2, 5, 9] and SIs [10, 50, 51], but also in many of their variants (eg. improved EAs [6, 7, 26, 42-49] and SIs [18, 19, 52]). To alleviate the possible adverse effect from randomness and to improve the performance of these algorithms, we propose a generic selective-candidate framework with similarity selection rule (SCSS). Here M candidates ($M > 1$) are generated from each of the current solutions by M independent reproduction procedures. One of which is selected as the final competitor against the current solution based on a specific selective rule.

Algorithm 1. General Procedures of EAs and SIs

1: Initialize population $X = \{x_1, x_2, ..., x_{NP}\}$;

2: **While** the stopping criteria are not met **Do**

3:    Determine the control parameters CP for genetic operations/social learning;

4:    Produce a new population Y via genetic operations/social learning on X;

5:    Evaluate the fitness of Y;

6:    Select solutions as new X from $X \cup Y$ to enter next iteration.

7: **End While**

2.2 SCSS Framework

The pseudo-code of the proposed SCSS framework is presented in **Algorithm 2**, which consists of two components, i.e. multiple candidates generation and similarity selection (SS) rule.

---

[1] For brevity, a review of three typical algorithms, DE, ES and PSO is presented in the supplementary file.

2.2.1 Multiple Candidates Generation

As seen from Algorithm 2, the SCSS framework performs M independent reproductions with M sets of independent parameters (i.e. evolution scale) and operations (i.e. evolution direction) (lines 5-7). Thus, for each current solution x_i, it owns a pool of candidate y_i^m $\{m = 1, 2, \ldots, M\}$. One solution y_i is then selected from the corresponding M candidates for each x_i by SS rule (lines 14 and 18), as a result, the actual parameters in use are recorded (lines 15 and 19).

Algorithm 2. SCSS Framework

1: Initialize population $X = \{x_1, x_2, \ldots, x_{NP}\}$;

2: **While** the stopping criteria are not met **Do**

3: Determine the fitness ranking $rank(i)$ of each individual $i\{i = 1, 2, \ldots, NP\}$; // *fitness ranking for SS rule*

------------------------------ **Multiple Candidates Generation** -------------------------

4: **For** $i = 1: NP$

5:    **For** $m = 1: M$

6:    Determine the control parameters $CP^m = \{cp_1^m, cp_2^m, \ldots, cp_{NP}^m\}$ for genetic operations/social learning, following the original design of the baseline;

7:    Produce new solution y_i^m via genetic operations/social learning on x_i;

8:    Calculate $dist_i^m =$ Euclidian distance (y_i^m, x_i); // *similarity calculation for SS rule*

9:    **End For**

10: **End For**

----------------------- **Similarity Selection Rule (Scheme 1 for example)** ----------------------------

11:    **For** $i = 1: NP$

12:     **If** $rank(i) \leq \text{ceil}(NP \times GD)$ //*GD is a greedy degree parameter, which controls the trade-off of EEC*

13:      $index = \underset{m \in \{1,2,\ldots,M\}}{\arg\min}(dist_i^m)$;

14:      $y_i = y_i^{index}$;

15:      $cp_i = cp_i^{index}$;

16:     **Else**

17:      $index = \underset{m \in \{1,2,\ldots,M\}}{\arg\max}(dist_i^m)$;

18:      $y_i = y_i^{index}$;

19:      $cp_i = cp_i^{index}$;

20:     **End If**

21:    **End For**

22: Evaluate the fitness of **Y**;

23: Select solutions as new **X** from **X**∪**Y** to enter next iteration.

24: **End While**

2.2.2 Similarity Selection Rule

Apparently, the major challenge in the SCSS framework is how to determine the final competitor from *M* candidates. On one hand, the selective rule should be effective to bring in performance enhancement. On the other hand, it should be efficient to reduce the computational load.

Hence, we propose a similarity selection (SS) rule, as given in **Algorithm 2** (lines 11-21). The rule simultaneously considers the fitness ranking information *rank*(*i*) of current solution x_i and its Euclidian distance $dist_i^m$ to each of the *M* candidates y_i^m, which is defined as

$$dist_i^m = \sqrt{\sum_{j=1}^{D}(y_{i,j}^m - x_{i,j})^2} \ ,$$

where *D* is the number of decision variables.

By adjusting SS, the amount of exploitation and exploration can be directly controlled. For instance, favoring candidates closest to the current solutions are exploitative while preferring the ones farthest to the current solutions could encourage exploration.

However, it should be remarked that the appropriate choice of SS for a specific algorithm is dependent on the EEC of the given algorithm. For illustration purposes, assume that the EEC is represented by a searching radius (SRAD). A larger SRAD implies a more explorative characteristic, and vice versa. The effects of SRAD on the performance of an algorithm are illustrated in Fig. 1, in which a minimization problem is assumed. In Fig. 1 (a), Optimizer 1 is very explorative. The large SRAD facilitates a more random-like search and there is little risk suffered from local optima. However, this large SRAD would also make the individuals such as Individual 1 (blue dot) and 2 (red dot) hard to refine. In contrast, Optimizer 2 in Fig. 1(b) is very exploitative and has a small SRAD. In this case, Individuals 1 and 2 focus more on local searches. It is difficult for them to move from basin I to basin II, which is important for diversity enhancement. Different from Optimizers 1 and 2, Optimizer 3 has a balanced EEC with an appropriate SRAD, as shown in Fig. 1(c). However, a drawback is that the SRAD is the same for the superior Individual 1 and the inferior Individual 2. For the superior Individual 1, this SRAD may not be small enough for an efficient local search while for the inferior Individual 2, the SRAD may not be large enough for it to jump from basin I to basin II.

Regarding different cases: 1) for an explorative optimizer (as in Fig.1(a)), the SRAD should be reduced to concentrate the search; 2) for an exploitative optimizer (as in Fig.1(b)), the SRAD should be enlarged to

encourage exploration to new searching areas; and 3) for a well-performing optimizer with balanced EEC, different searching tasks should be assigned to solutions with different potentials.
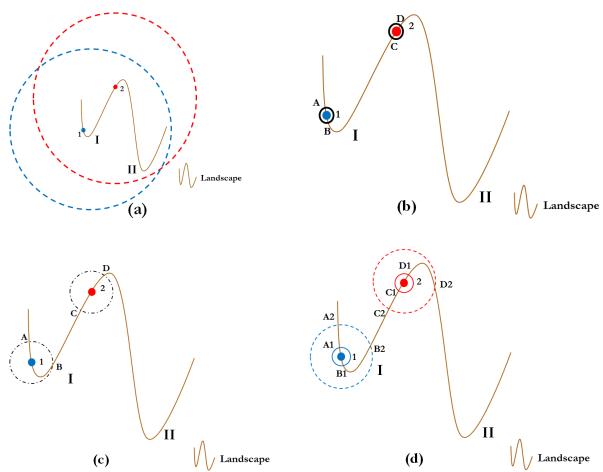

Fig.1 Illustration of the effects of SRAD on the performance of an algorithm (for minimization problem).

Optimizer 4 in Fig. 1(d) illustrates an improved version of Optimizer 3 based on multiple candidates generation (SCSS with $M = 2$). The possible candidates generated could be close to the current solutions with a small SRAD (solid line circles in Fig.1(d)), such as A1, B1, C1 and D1, or away from with a large SRAD (dotted line circles in Fig. 1(d)), such as A2, B2, C2 and D2. Indeed, the SRAD size of Optimizer 3 ranges between those of the dotted line circles and the solid line circles of Optimizer 4. Therefore, compared with the SRAD of Optimizer 3, the solid line circles of Optimizer 4 provide a smaller radius for local search while the dotted line circles could be large enough for basin-jumping.

On the one hand, new best solutions are likely to be located in the area near the top-ranked solutions in the context of a continuous landscape. To achieve a better efficiency in exploitation, closest candidates of superior solutions are considered, targeting steady improvements for promising areas. On the other hand, farthest candidates of inferior solutions are preferred, aiming for better exploration.

In view of the above, two SS schemes are proposed as follows:

Scheme 1: **If** $rank(i) \leq$ ceil($NP \times GD$)

Select the closest candidate from y_i^m {$m = 1, 2, …, M$} for individual x_i;

Else

Select the farthest candidate from y_i^m {$m = 1, 2, …, M$} for individual x_i;

End If

Scheme 2: **If** $rand_i(0,1) > rank(i)/NP$

Select the closest candidate from y_i^m {$m = 1, 2, …, M$} for individual x_i;

Else

Select the farthest candidate from y_i^m {$m = 1, 2, …, M$} for individual x_i;

End If

where $rank(i) \in$ {1, 2, …,NP} is the fitness ranking of individual x_i and $rank(i)$=1 is the best. *ceil (.)* is a ceiling function. $rand_i(0,1)$ is a uniformly distributed random number within (0,1) for individual x_i {$i = 1, 2, …,NP$}.

In Scheme 1, the proportion of top individuals preferring the closest candidates is controlled by a greedy degree parameter *GD* in the range [0,1]. Specifically, the superior $GD \times 100\%$ selects the nearest candidates while the inferior $(1 - GD) \times 100\%$ portion selects the farthest candidates. The larger the *GD* value is, the exploitative Scheme 1 becomes.

In Scheme 2, higher ranked individuals are associated with higher probabilities in using the closest candidates, while lower ranked ones are likely to utilize the farthest candidates. One of the advantages is that Scheme 2 is parameterless. As shown later in Section 4, Scheme 2 works well for most of the advanced EA and SI variants.

Based on Algorithm 2, the SCSS variants for existing EAs and SIs can be easily implemented. As examples, the work flow of three SCSS variants, namely SCSS-DE, SCSS-ES and SCSS-PSO for the classic DE, ES, and PSO are given in Algorithms S1, S2 and S3 in the supplementary file, respectively.

2.2.3 Time Complexity

This subsection discusses the time complexity of the proposed method. Considering DE as an example, its time complexity is $O(NP \cdot D \cdot Gen_{max})$, where *NP* is population size, *D* is the number of decision variables of the problem and Gen_{max} is the maximum number of generations. In SCSS-DE, the complexity of fitness ranking and Euclidian distance calculation for each generation are $O(NP \cdot \log_2 NP)$ and $O(M \cdot NP \cdot D)$, respectively. Besides, the complexity of *M* reproductions is $O(M \cdot NP \cdot D)$. Since $\log_2 NP << D$, the overall complexity is $O(M \cdot NP \cdot D \cdot Gen_{max})$. As investigated in Section 4, $M = 2 << NP$ is sufficient for advanced

DEs, such as the JADE [36] and L-SHADE [32] algorithms. Thus, the complexity of advanced SCSS-DEs remains as $O(NP \cdot D \cdot Gen_{max})$.

3 Simulation

In this section, the effectiveness of the proposed SCSS framework and its working mechanism are investigated through comprehensive experiments conducted using the CEC2014 [53] and CEC2017 [54] benchmark function sets. Each function set consists of 30 functions with diverse mathematic characteristics, such as unimodal, multimodal, hybrid and composition. Since the CEC function suits are with bounded constraints, to make the comparison fair, the constraint handling technique adopted in the SCSS variants is kept the same as the corresponding baselines. The solution error value, defined as $f(x)$ - $f(x^*)$, is used to measure the performance of the compared algorithms, where $f(x)$ is the smallest fitness obtained after $10^4 \times D$ function evaluations and $f(x^*)$ is the fitness of the global optimal x^*. Following [53, 54], solution error values smaller than 10^{-8} are considered as zero. For each test function, 51 independent runs are performed, while the mean and standard deviations of the solution error values are reported. Besides, in order to draw statistically sounded conclusions, Wilcoxon signed-rank test [55] with 5% significance level is applied to compare the performance. The symbols "−", "=" and "+" represent that the baseline algorithms perform significantly worse than, similar to or better than the corresponding SCSS variants, respectively. The significant ones are marked in **bold**.

3.1 Performance Enhancement of Classic EAs and SIs

The proposed SCSS framework is first integrated with three classic EAs and SIs, i.e. DE and ES from EA family and PSO from SIs. Performance of the resulting variants, SCSS-DE, SCSS-ES and SCSS-PSO are compared with the baseline algorithms, respectively.

The parameter settings for the compared algorithms are summarized as follows:

DE and SCSS-DE: $NP = 100$, $F = 0.7$, $CR = 0.5$;

ES and SCSS-ES: $\mu = 25$, $\lambda = 100$, intermediate recombination is used;

PSO and SCSS-PSO: $NP = 20$, $w = 0.9$, $c_1 = 2.0$, and $c_2 = 2.0$;

In addition, regarding the SS rule, Scheme 1 with $GD = 1$ and $M = 2$ is adopted in the three SCSS variants. These settings are based on the experimental findings given later in Section. 3.3. The comparison results on 30-D and 50-D CEC2014 functions are summarized in Fig.2.

As observed in Fig. 2, the effectiveness of the proposed SCSS framework on all the considered algorithms is confirmed. In the total 180 cases, SCSS variants win in 125 (=21+26+15+22+27+14) cases and only lose in one case. Specifically, in the 30-D cases, SCSS-DE and SCSS-ES perform significantly better than their

corresponding baselines on 21 and 26 functions and lose on one and no function, respectively. SCSS-PSO wins PSO on 15 functions and ties on 15 functions. In the 50-*D* case, SCSS-DE, SCSS-ES, and SCSS-PSO win the baselines on 22, 27 and 14 functions, respectively, and the rests are tie. It is remarked that, since the classic algorithms use fixed parameter settings, these performance improvements are attributed to the control of the randomness of the reproduction operations by SCSS, such as the random selection of parents for mutation and dimension-wise crossover in DE. In summary, the proposed SCSS framework significantly enhances the performance of these basic algorithms.
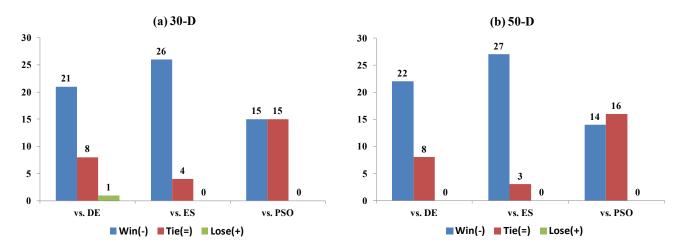

Fig.2 Comparison results of three SCSS-based classic algorithms with the baselines on CEC2014 test functions: (a) 30-*D*, (b) 50-*D*. Scheme 1 with *GD* = 1 and *M* = 2 for all the three SCSS variants.

3.2 Performance Enhancement of Advanced EAs and SIs

Thanks to the efforts by EA and SI researchers, the performance of the classic algorithms had been greatly improved by many advanced variants. Thus, it is essential to investigate whether our proposed method could also further enhance these algorithms. For demonstration, SCSS is incorporated into four advanced baselines, namely JADE [36], SHADE [42], CMA-ES [26] and LIPS [18]. Parameter settings for the compared algorithms are set the same as those recommended in their original literature. Additionally, for the SCSSs, Scheme 2 is utilized as the SS rule in SCSS-JADE, SCSS-SHADE and SCSS-LIPS, while Scheme 1 with *GD* = 0 is applied for SCSS-CMA-ES. The reproduction times *M* is set to 2 for SCSS-JADE and SCSS-SHADE, 4 for SCSS-LIPS and 5 for SCSS-CMA-ES. These settings are the best, as indicated later by the parameter sensitivity analyses in Section. 3.3.

The experimental results on 30-*D* and 50-*D* CEC2014 functions are shown in Table S1 and Table S2, respectively, in the supplementary file and further summarized in Fig. 3.

As observed from Fig. 3, SCSS also exhibits remarkable improvements on the advanced algorithms. Out of the total 240 cases, SCSS wins in 134 (=14+14+17+23+16+11+13+26) cases and just loses in 17 (=1+0+5+2+1+0+5+3) cases. More specifically, for the advanced DEs, i.e. JADE and SHADE, SCSS

improves their performance on 55 functions and is inferior on 2 functions. For CMA-ES, SCSS wins in 17 and 13 cases and loses in 5 cases on the 30-*D* and 50-*D* functions, respectively. For the advanced PSO algorithm, i.e. LIPS, SCSS-LIPS is superior on more than 20 functions and inferior on far fewer functions in both 30-*D* and 50-*D* cases.

Considering the diverse mathematical properties of the test functions, it can be concluded that SCSS consistently works well on various types of functions, including unimodal, multimodal, hybrid and composition.
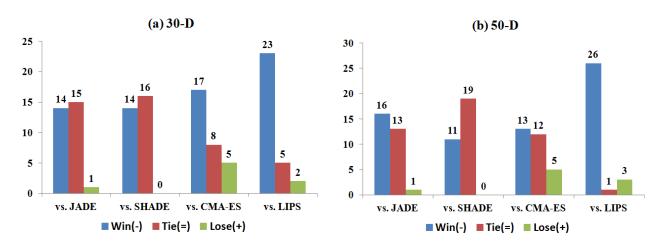

Fig.3 Comparison results of four SCSS-based advanced algorithms with the baselines on CEC2014 test functions: (a) 30-*D*, (b) 50-*D*. Scheme 2 is utilized in SCSS-JADE, SCSS-SHADE and SCSS-LIPS, while Scheme 1 with *GD* = 0 is applied for SCSS-CMA-ES. The reproduction times *M* is set to 2 for SCSS-JADE and SCSS-SHADE, 4 for SCSS-LIPS and 5 for SCSS-CMA-ES.

3.3 Working Mechanism of SS Rule

3.3.1 Influence of SS rule on the performance of SCSS

The performance sensitivity of SCSS to the SS rule is firstly investigated. Performance of seven SCSSs, i.e. SCSS-DE, SCSS-ES, SCSS-PSO, SCSS-JADE, SCSS-SHADE, SCSS-CMA-ES and SCSS-LIPS with different SS rules (i.e. Scheme 1 with six *GD* values, i.e. 0, 0.2, 0.4, 0.6, 0.8, 1 and Scheme 2) are compared with those of the baseline algorithms, respectively. The *M* value for all the SCSS variants in this experiment is set as 2. The completed comparison results "-/=/+" are given in Table S3 in the supplementary file, while Fig. 4 presents the P-N values (defined as the number of "−" minus the number of "+") as a summary.

From Fig. 4, the followings can be observed:

(1) For the classic algorithms, including DE, ES, and PSO, SCSS variants adopting larger *GD* values perform better than those with smaller ones. The reason lies in that classic algorithms are usually explorative and deficit in exploitation (the case in Fig.1(a)). Large *GD* values could encourage

exploitation to remedy the blindness of the search. While small GD values, such as $GD=0$, make the algorithms even more explorative and deteriorate the performance, as can be observed from Fig. 4.

(2) For the advanced algorithms, Scheme 2 is the best choice for SCSS-SHADE and SCSS-LIPS and the third best choice for SCSS-JADE. Also, for SCSS-JADE and SCSS-SHADE, the performance of SCSSs with Scheme 1 significantly degenerates when GD is too large ($GD = 1$) or too small ($GD = 0$). It is because JADE and SHADE themselves maintain relatively balanced EEC (the case in Fig.1 (c)). $GD=1$ would over-emphasize exploitation and make the algorithms too greedy while an over-explorative setting $GD = 0$ may deteriorate the performance on test functions which need more exploitation.

(3) For SCSS-CMA-ES, Scheme 1 with $GD = 0$ achieves the best performance, indicating that the original CMA-ES (the case in Fig. 1(b)) needs more exploration for performance enhancement. This observation is in consistent with the statements in some CMA-ES literature, such as PS-CMA-ES [27] and IPOP-CMA-ES [33] that CMA-ES could benefit from enhanced exploration capability when solving difficult CEC benchmarks.

In conclusion, the choice of a best SS rule depends on the EEC of the baselines while Scheme 2 consistently performs significantly better than or similar to the baselines. As a design rule of thumb, for an optimizer with relatively balanced EEC, Scheme 2 is recommended.
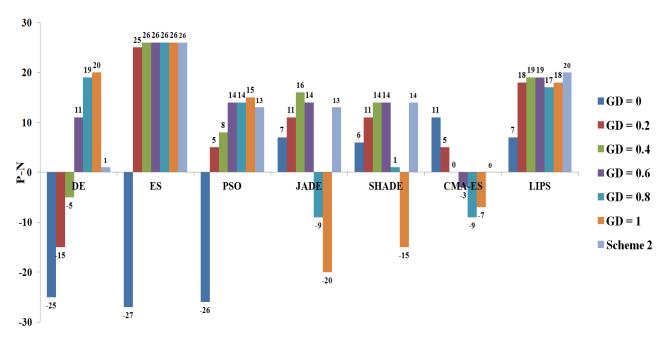

Fig.4 P-N values of SCSS variants with different SS rules against the baselines on 30-D CEC2014 test functions. (P-N value ＝ the number of functions that SCSS variant outperforms the baseline － the number of functions that SCSS variant loses to the baseline).

3.3.2 Behavior of SS rule

In the proposed SCSS framework, the selection of the closest or farthest candidates is conducted based on the fitness ranking of the current solutions. In this way, SCSS adjusts the level of exploration/exploitation according to their potentials. In the experiment conducted on SCSS-DE ($GD = 1$) and SCSS-SHADE (Scheme 2), SS rule is compared with a randomly selecting (RS) manner (i.e. selecting manner in the baseline algorithm). The total distance TD between the selected candidates and the current solutions against the rank on 30-D functions F5 and F13 in the median run is shown in Fig. 5.

From Fig. 5, we have the following observations: 1) on the explorative DE, SS enhances the exploitation on all the ranks, resulting in smaller TD values than that of RS; 2) on SHADE with relative balanced EEC, for ranks smaller than $NP/2 = 50$, SS yields smaller TD compared to RS, resulting in more exploitation. While for ranks larger than 50, it is the opposite case; 3) on SHADE, for RS, TD varies little with the rank but TD significantly increases with the rank for SS. Since the searching radius SRAD can be roughly calculated as TD/Gen_{max}, where Gen_{max} is the maximum number of generations and it is the same for both SHADE and SCSS-SHADE, $SRAD \propto TD$. This means that SRAD increases with the rank in SS while it is the same in RS. In other words, SS is a finer strategy; 4) the smaller TD values of SHADE compared to that of DE reveal that SHADE is more exploitative than DE. Therefore, unlike the case in SCSS-DE, enlarging GD in SCSS-SHADE may make the algorithm over-exploitative and deteriorate the performance, which is also observed from Fig. 4.
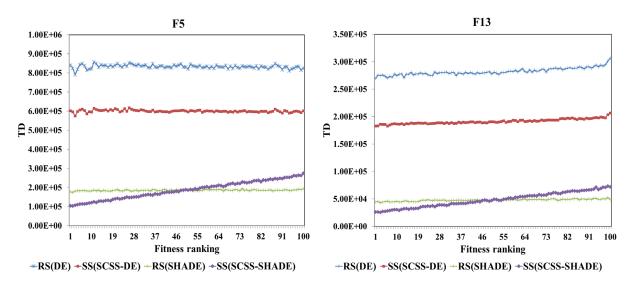

Fig.5 TD against the rank on 30-D CEC2014 functions F5 and F13. (The similar phenomena can be observed on all the CEC functions)

3.3.3 Benefit of SS rule

To further demonstrate the benefit of SS (Scheme 2), it is compared with the following three variants:

Variant-oppo: An opposite version of Scheme 2 is defined as follows:

> **If** $rand_i(0,1) > rank(i)/NP$
>
> > Select the farthest candidate from y_i^m {$m = 1, 2, …, M$} for individual x_i;
>
> **Else**
>
> > Select the closest candidate from y_i^m {$m = 1, 2, …, M$} for individual x_i;
>
> **End If**

Variant-Meval: Scheme 2 is replaced with true function evaluations. Specifically, for each current solution, the M candidates are evaluated and the fittest one is selected as offspring, like CoDE [38].

Variant-CSM: Instead of using Scheme 2, the cheap surrogate model (CSM) proposed in [25] is used to determine the offspring from M candidates.

For direct comparison, other settings are unaltered and experiments were conducted with JADE [36]. From Table 1 and Table S4, the results are summarized as follows.

(1) SCSS-JADE exhibits better performance than Variant-oppo. In addition, comparing Table S1 with Table S4, it is also observed that Variant-oppo performs significantly worse than the baseline, concluding that the opposite version is an inappropriate selective rule. This confirms the illustrations given in Section 2.2.2 and Fig. 1 (d).

(2) SCSS-JADE performs better than Variant-Meval. This can be explained by the fact that, in Variant-Meval, M ($M = 2$) function evaluations are consumed to determine each offspring per geneation and, as a result, the maximum number of iterations is reduced. (Note: The total number of evaluations are fixed.)

(3) SCSS-JADE also outperforms Variant-CSM. To have an in-depth insight into the working processes of SS and CSM, Fig. 6 plots their average prediction accuracy (PA) on thirty 30-D CEC2014 functions. The PA is calculated as the number of trials that correctly selects the fittest candidate divided by the number of total trials. From Fig. 6, we have the following observations and discussions.

  1) Overall, PA varies with problems that pose different degree of difficulities.

  2) For SS, exploitation part (EiP) has higher PA than the exploration part (ErP) on all the functions. This is understable as ErP is responsible for broadening the search region.

  3) Comparing EiP with CSM, it is seen that EiP has higher PA on 24 out of the total 30 functions. As pointed out in the original paper [25], since CSM is a cheap model, it may not estimate the density exactly, especially for the highly-rotated CEC test functions.

4) Althourgh high PA is generally more desirable, higher PA does not necessarily contribute to better performance on some functions. This can be confirmed by the observation on F17, F18 and F24. On these three functions, although CSM has higher PA than EiP, its performance is significantly inferior to SS (see Table S4). It is because CSM includes no mechanism for exploration while SS simultaneously maintains two strategies (i.e. superior/inferior solutions select the closest/farthest candidates) for synthesizing exploitaton and exploration purposes, respectively. The latter strategy always attemps to explore far-away areas, where new exploitation may then be emerged once the offspring of inferior solutions becomes elites. For this reason, it is expected that exploration could also benefit exploitation and should work cooperatively. In fact, this has been verified by the overwhelmingly better performance of SCSS-JADE with Scheme 2 over $GD = 1$ (see Fig. 4).

5) Besides the accuracy, it is noted that SS rule has lower complexity ($O(M \cdot NP \cdot D)$) than CSM ($O(M \cdot NP^2 \cdot D)$ [25]), which is more significant with larger NP value.

Table 1 Comparison results of SCSS-JADE with three variants on 30-D CEC2014 test functions

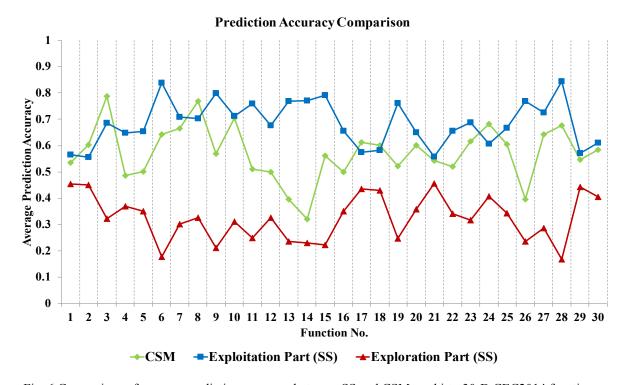|	-/=/+
Variant-oppo vs. SCSS-JADE	24/5/1
Variant-Meval vs. SCSS-JADE	16/14/0
Variant-CSM vs. SCSS-JADE	18/11/1

Prediction Accuracy Comparison

Fig. 6 Comparison of average prediction accuracy between SS and CSM on thirty 30-D CEC2014 functions.

3.3.4 Combined effects of operations and parameters by SS rule

The SS rule considers candidates that reveal the combined effects of operations and parameters, which makes SCSS a general framework that can be easily applied to various types of EAs and SIs. The effects of SCSS on the randomness of operations and parameters of the previously considered algorithms are summarized as follows.

(1) For the three classic algorithms DE, ES and PSO, since the parameters are fixed during the entire evolution process, SCSS reveals the effect of operations;

(2) For the advanced DEs, i.e. JADE and SHADE, except the operations, since different reproduction procedure m may use different F and CR, SCSS reveals their combined effects;

(3) In the advanced ES, i.e. CMA-ES, new individuals are generated from the center of best solutions by following a normal distribution. Thus, in SCSS-CMA-ES, different normal distributions are sampled in different reproduction procedures;

(4) In the advanced PSO, i.e. LIPS, SCSS uses different independently generated φ_j in the position update equation, which is a uniformly distributed random number ranged in [0, 4.1/ *neighborhood size*] for each dimension j [18].

3.4 Performance Sensitivity to M

In SCSS, M ($M > 1$) reproduction procedures should be performed. Indeed, if M is set to 1, SCSS variants degenerate to baselines. Apparently, the performance of the SCSS is influenced by M. Therefore, in this subsection, SCSS variants with five different M values, i.e. $M = 2, 3, 4, 5$ and 10 are compared. Except M, other parameter settings for the compared algorithms are set the same as those used previously in Sections 3.1 and 3.2. Performance comparisons of the SCSS variants with the baselines on 30-D CEC2014 functions are summarized in Table S5 and Fig. 7. In addition, to show the dynamic performance variation with increasing M, the performance of the SCSS variants using adjacent M settings are also compared with each other, as shown in Table S6 and Fig. 8.

It can be observed from Fig. 7 that all of the M settings significantly improve the performance of the baselines except SCSS-JADE and SCSS-SHADE with $M = 10$.

In Fig.8, for clarity, the algorithms are divided into two categories. Category 1 includes the SCSS variants which may perform significantly better with $M > 2$ than with $M = 2$, while Category 2 lists the SCSS variants which perform similarly or even worse with increased M values.

In Category 1, it is observed that the performance of DE and ES consistently improves when M increases. In this paper, we only investigate cases up to $M=10$ since these classic algorithms are significantly inferior to the advanced algorithms. Moreover, increasing M will increase the computational complexity of the algorithm. For CMA-ES and LIPS, SCSS variants with $M = 5$ and $M = 4$ show the best performance,

respectively. It is noticed that in the SCSS-CMA-ES, *GD* is set to 0, thus, larger *M* values would make the algorithm more explorative.

In Category 2, enlarging *M* does not bring significant performance improvements. On the contrary, it may even significantly degrade the performance, eg. $M > 4$ for JADE and SHADE, or $M > 2$ for PSO. The reason is that different from those in Category 1 (eg. DE, ES and LIPS), JADE, SHADE and PSO include elite individuals in their reproduction processes. Specifically, the top-ranked individuals used in the "current-to-*pbest*/1" mutation strategy of JADE and SHADE and the global best *gbest* used in the velocity update equation of PSO. Therefore, algorithm with too large an *M* value is potentially too greedy, making the algorithms stuck in local optima.

Overall, it can be concluded that the appropriate *M* value is relatively small for the advanced variants.
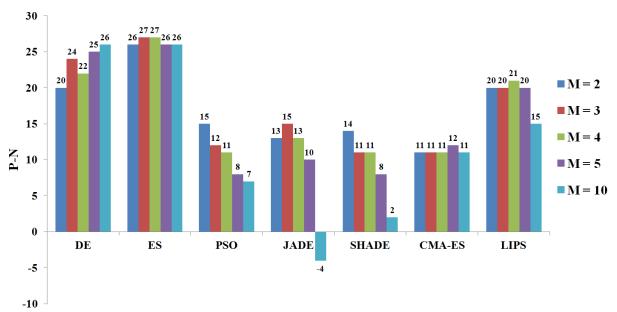
Fig.7 P-N values of SCSS variants with different M settings against the baselines. (P-N value = the number of functions that SCSS variant outperforms the baseline − the number of functions that SCSS variant loses to the baseline).

(a) CATEGORY 1

(b) CATEGORY 2
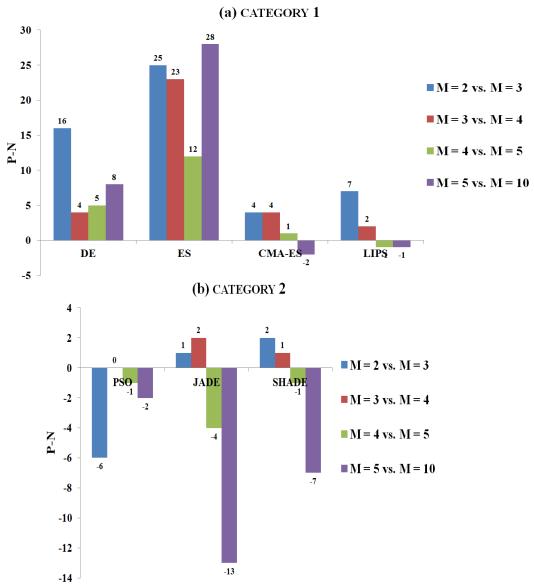
Fig.8 P-N values between SCSS variants (A vs. B) with adjacent M settings (P-N value $=$ the number of functions that B outperforms A $-$ the number of functions that B loses to A).

3.5 Application in Top Methods from CEC Competitions

From Sections 3.3 and 3.4, it can be concluded that advanced SCSS-DEs with Scheme 2 and $M = 2$, SCSS-CMA-ES with Scheme 1($GD = 0$) and $M = 5$ exhibit promising performance. In this subsection, to demonstrate the flexibility, SCSS is further applied with these settings to four highly competitive algorithms from the CEC competitions. Among them, L-SHADE [32] is the winner of the CEC2014 competition, UMOEA-II [43] and L-SHADE_EpSin [44] are the joint-winner of the CEC2016 competition and jSO [45] is one of the best-performing algorithms in the CEC2017 competition. Their source codes are available at http://www.ntu.edu.sg/home/epnsugan/. Parameter settings for these top algorithms are set the same as the original literature.

As shown in Table S7, Table S8 and Fig. 9, SCSS also enhances the performance of these top methods. Out of the total 240 cases, SCSSs win in 88 (=10+9+8+7+18+10+13+13) cases and lose in 12 (=2+1+0+2+2+3+0+2) cases. Specifically, in the 30-*D* case, SCSS-L-SHADE, SCSS-UMOEA-II, SCSS-L-SHADE_EpSin, and SCSS-jSO perform significantly better than the corresponding baselines in 10, 9, 8 and 7 cases and underperform in 2, 1, 0 and 2 cases, respectively. In the 50-*D* case, the performance improvements are more significant. SCSS-L-SHADE, SCSS-UMOEA-II, SCSS-L-SHADE_EpSin and SCSS-jSO exhibit superior performance on 18, 10, 13 and 13 functions respectively and are inferior on far fewer functions.

Fig. 10 shows the convergence plot of SCSS-L-SHADE versus L-SHADE on six selected 50-*D* CEC2014 functions. As observed, SCSS-L-SHADE exhibits better convergence performance than L-SHADE. In conclusion, these performance enhancements indicate that the proposed SCSS framework is a better alternative for these top algorithms.
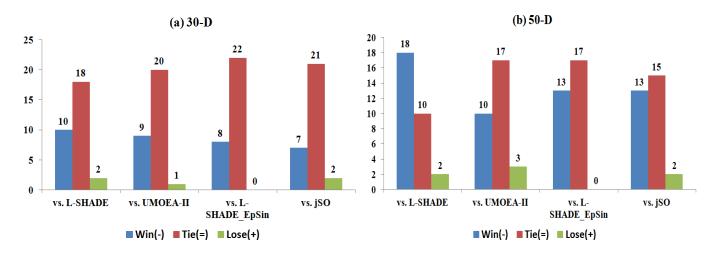
Fig.9 Comparison results of four SCSS-based top algorithms with the baselines on CEC2014 test functions: (a) 30-D, (b) 50-D.

3.6 Performance on CEC2017 Test Suit and Scalability Study

To assess the performance of SCSS on a wider variety of functions, in this subsection, we further test the advanced SCSS variants on the recently developed CEC2017 test suite [54]. This test suite also has 30 functions, but with several new features, such as new basic functions, graded level of linkages and rotated trap functions [54].

Parameter settings for the algorithms are the same as those used in Sections 3.2 and 3.5. Tables S9-S12 present the experimental results on 30-*D* and 50-*D* functions and Table 2 summarizes the comparison results. From Table 2, it is clear that SCSS also significantly improves the performance of the baselines on the CEC2017 functions. In the total 480 cases, SCSS wins in 225 cases, ties in 240 cases and loses in 15 cases.
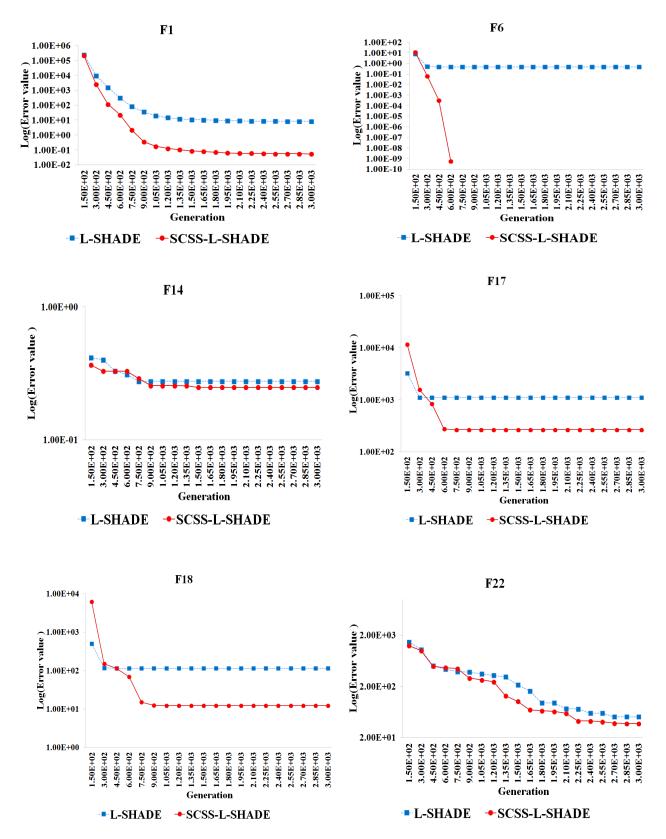
Fig.10 Convergence plot of SCSS-L-SHADE versus L-SHADE on six selected 50-D CEC2014 functions in the median run. (Note: On F6, SCSS-L-SHADE reaches the global optimal at generation 750)

Table 2 Comparison results of SCSS variants with the baselines on CEC2017 test suit

-/=/+	30-D	50-D
JADE vs. SCSS-JADE	19/11/0	18/10/2
SHADE vs. SCSS-SHADE	7/23/0	11/19/0
CMA-ES vs. SCSS-CMA-ES	18/11/1	16/14/0
LIPS vs. SCSS-LIPS	28/1/1	28/1/1
L-SHADE vs. SCSS- L-SHADE	9/18/3	15/15/0
UMOEA-II vs. SCSS-UMOEA-II	3/24/3	14/14/2
L-SHADE_EpSin vs. SCSS-L-SHADE_EpSin	7/21/2	13/17/0
jSO vs. SCSS-jSO	7/23/0	12/18/0
Total	225/240/15	

To study scalability, the SCSS framework is also tested on 100-D CEC2017 functions. The four top methods are selected for the experiment and the parameters are set the same as those used previously.

As shown in Table S13 and Fig. 11, SCSS still yields remarkable performance improvements on the higher dimensional functions, which are much more difficult than the lower dimensional ones. In the total of 120 cases, SCSS outperforms in 70 (=20+14+16+20) cases and underperforms in 6 (=2+2+0+2) cases. These improvements should be attributed to the balanced exploitation and exploration maintained by the SS rule.

Furthermore, the overall performances of the considered algorithms are compared according to multiple problem Wilcoxon's test [56] and Friedman's test [56]. Based on multiple problems Wilcoxon's test, Table 3 shows that the SCSS variants perform significantly better than the corresponding baselines at $\alpha = 0.05$. With respect to the Friedman's test, Table 4 indicates that SCSS-jSO is the best-performing algorithm, which achieves the smallest ranking value of 2.76, followed by SCSS-L-SHADE_EpSin.
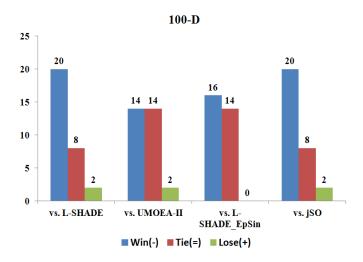
Fig.11 Comparison results of four SCSS-based top algorithms with the baselines on 100-D CEC2017 test functions.

Table 3 Comparison results of the top SCSS variants with the baselines on 30-D, 50-D and 100-D CEC2017 benchmark set according to multi-problem Wilcoxon's test

	$R+$	$R-$	p-value	$\alpha = 0.05$
SCSS-L-SHADE vs. L-SHADE	3235.0	770.0	**0.0E+00**	**Yes**
SCSS-UMOEA-II vs. UMOEA-II	3052.5	952.5	**1.7E-05**	**Yes**
SCSS-L-SHADE_EpSin vs. L-SHADE_EpSin	3077.0	1018.0	**3.4E-05**	**Yes**
SCSS-jSO vs. jSO	3710.5	384.5	**0.0E+00**	**Yes**

Table 4 Overall performance ranking of the considered algorithms on 30-D, 50-D and 100-D CEC2017 benchmark set by Friedman's test

Algorithm	Ranking
SCSS-jSO	**2.76**
SCSS-L-SHADE_EpSin	3.60
jSO	3.88
L-SHADE_EpSin	4.33
SCSS-L-SHADE	4.57
SCSS-UMOEA-II	5.19
L-SHADE	5.67
UMOEA-II	5.96

3.7 Application in Real-world Problems

We have also applied the proposed method to 22 real-world applications, from [57], where detailed descriptions and source codes of the problems are available. These problems come from various scientific and engineering fields, such as frequency-modulated (FM) sound waves parameter estimation problem, Lennard-Jones potential problem, spread spectrum radar polly phase code design problem, large scale transmission pricing problem and so on [57]. They have a wide range of dimensionality from one up to 216 and are very challenging [57]. As an example, we focused on SCSS-L-SHADE and L-SHADE. Each algorithm has 30 trials with each trial assigned $10^4 \times D$ function evaluations. Table 5 tabulates the mean and standard deviations of the solution error values. As shown, SCSS-L-SHADE performs significantly better on 11 problems (including P1, P5-P7, P9, P11, P12, P16-P18 and P20) and loses on none. This demonstrates the reliable performance of SCSS when incorporated with L-SHADE for real-world applications.

Table 5 Performance comparisons (mean (std)) of SCSS-L-SHADE with L-SHADE
on 22 CEC2011 real-world problems

	L-SHADE	SCSS-L-SHADE		L-SHADE	SCSS-L-SHADE
cec11 P1	0.73 (2.73) —	**0.34 (1.86)**	cec11 P12	1050159.77 (1254.39) —	**1047950.07 (1191.33)**
cec11 P2	-27.68 (0.38) =	-27.79 (0.54)	cec11 P13	15444.51 (1.56) =	15444.19 (0.00)
cec11 P3	0.00 (0.00) =	0.00 (0.00)	cec11 P14	18093.89 (33.47) =	18093.73 (33.53)
cec11 P4	18.98 (3.09) =	17.69 (3.34)	cec11 P15	32740.43 (0.21) =	32740.41 (0.18)
cec11 P5	-36.84 (0.02) —	**-36.82 (0.16)**	cec11 P16	123355.03 (580.33) —	**123000.40 (381.84)**
cec11 P6	-29.16555 (0.00) —	**-29.16598 (0.00)**	cec11 P17	1735648.35 (7377.90) —	**1729536.58 (5961.35)**
cec11 P7	1.16 (0.07) —	**1.11 (0.09)**	cec11 P18	925951.66 (758.39) —	**925373.83 (489.51)**
cec11 P8	220.00 (0.00) =	220.00 (0.00)	cec11 P19	934334.22 (700.86) =	934138.43 (617.75)
cec11 P9	369.60 (125.46) —	**292.23 (104.70)**	cec11 P20	926086.29 (462.05) —	**925719.66 (674.85)**
cec11 P10	-21.60 (0.11) =	-21.62 (0.08)	cec11 P21	15.50 (0.57) =	15.50 (0.62)
cec11 P11	48154.11(369.11) —	**47274.03 (410.89)**	cec11 P22	14.54 (2.40) =	14.09 (3.05)
-/=/+	**11/11/0**				

4 Conclusion

To address the potential adverse effect of randomness in evolutionary algorithms, a selective-candidate framework with similarity selection rule (SCSS) is proposed in this paper. In SCSS, each current solution owns a pool of M candidates generated by M reproduction procedures. The final candidate is then determined from the pool by a similarity selection method, which is designed based on fitness ranking and Euclidian distance measures. We have described the motivation of the design (Section 2.2.2), incorporated the design into several classic, advanced and top algorithms from EA and SI families (Sections 3.1, 3.2, 3.5 and 3.6), analyzed its working mechanism (Sections 3.3 and 3.4) and have also applied it to solve 22 real-world problems (Section 3.7). Comprehensive experiments show that 1) SCSS significantly enhances the performance of the considered algorithms; 2) Scheme 2 performs consistently well, especially on the advanced and top algorithms and is thus recommended; 3) the appropriate M value is relatively small (2 to 4) for the advanced and top algorithms with balanced EEC. According to Section 3.4, $M = 2$ should be the first choice when testing SCSS in a new metaheuristic since it always brings improvements. One may then further increase M to see whether better performance can be achieved.

The supplementary document and MATLAB demo codes of SCSS can be downloaded from https://zsxhomepage.github.io/.

References

[1] T. Bäck, Evolutionary Algorithms in Theory and Practice. London, U.K.: Oxford Univ. Press, 1996.

[2] K. S. Tang, K. F. Man, S. Kwong, Q. He, Genetic algorithms and their applications, IEEE Signal Process. Mag. 13 (1996) 22–37.

[3] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems. Oxford, U.K.: Oxford Univ. Press, 1999.

[4] F. Neri, V. Tirronen, Recent advances in differential evolution: a survey and experimental analysis, Artif. Intell. Rev. 33 (1–2) (2010) 61–106.

[5] S. Das, S. M. Sankha, P.N. Suganthan, Recent advances in differential evolution – An updated survey, Swarm Evol. Comput. 27 (2016) 1-30.

[6] R. D. Al-Dabbagh, F. Neri, N. Idris, M. S. Baba, Algorithm design issues in adaptive differential evolution: review and taxonomy, Swarm Evol. Comput. 43 (2018) 284-311.

[7] A. P. Piotrowski, J. J. Napiorkowski, Step-by-step improvement of JADE and SHADE-based algorithms: Success or failure? Swarm Evol. Comput. 43 (2018) 88-108.

[8] S. X. Zhang, L. M. Zheng, K. S. Tang, S. Y. Zheng, W. S. Chan, Multi-layer competitive-cooperative framework for performance enhancement of differential evolution, Inf. Sci., 482, (2019) 86-104.

[9] T. Bäck, H.-P. Schwefel, An overview of evolutionary algorithms for parameter optimization, Evol. Comput. 1 (1993) 1–23.

[10] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, November 27–December 1, 1995, pp. 1942–1948.

[11] A. E. Eiben, A. Schippers, On evolutionary exploration and exploitation, Fundamenta Inform. 35 (1998) 35–50.

[12] M. Črepinšek, S.-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: a survey, ACM Comput. Surv. 45 (3) (2013) 1–33 Art. No. 35.

[13] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, M. N. Vrahatis, Enhancing differential evolution utilizing proximity-based mutation operators, IEEE Trans. Evol. Comput. 15 (2011) 99–119.

[14] W. Gong, Z. Cai, Differential evolution with ranking-based mutation operators, IEEE Trans. Cybernet. 43 (2013) 2066–2081.

[15] J. Wang, J. Liao, Y. Zhou, Y. Cai, Differential evolution enhanced with multiobjective sorting-based mutation operators, IEEE Trans. Cybernet. 44 (2014) 2792–2805.

[16] L. M. Zheng, S. X. Zhang, K. S. Tang, S. Y. Zheng, Differential evolution powered by collective information, Inf. Sci. 399 (2017) 13–29.

[17] S. Y. Zheng, S. X. Zhang, A jumping genes inspired multi-objective differential evolution algorithm for microwave components optimization problems, Appl. Soft Comput. 59 (2017) 276-287.

[18] B. Y. Qu, P. N. Suganthan, S. Das, A distance-based locally informed particle swarm model for multimodal optimization, IEEE Trans. Evol. Comput. 17 (2013) 387-402.

[19] Z. H. Zhan, J. Zhang, Y. Li, Y. H. Shi, Orthogonal learning particle swarm optimization, IEEE Trans. Evol. Comput. 15 (2011) 832– 847.

[20] N. Lynn, P. N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. 24 (2015) 11–24.

[21] A. K. Qin, V. L. Huang, P. N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2009) 398–417.

[22] G. Wu, R. Mallipeddi, P. N. Suganthan, R. Wang, H. Chen, Differential evolution with multi-population based ensemble of mutation strategies, Inf. Sci. 329 (2016) 329–345.

[23] L. Cui, G. Li, Q. Lin, J. Chen, and N. Lu, Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations, Comput. Oper. Res. 67 (2016) 155-173.

[24] L. Tang, Y. Dong, J. Liu, Differential evolution with an individual-dependent mechanism, IEEE Trans. Evol. Comput. 19 (2015) 560-574.

[25] W. Gong, A. Zhou, Z. Cai, A multi-operator search strategy based on cheap surrogate models for evolutionary optimization, IEEE Trans. Evol. Comput. 19 (2015) 746–758.

[26] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (2001) 159–195.

[27] C. L. Müller, B. Baumgartner, I. F. Sbalzarini, Particle swarm CMA evolution strategy for the optimization of multi-funnel landscapes, in: Proc of the IEEE Congress on Evolutionary Computation, 2009, pp. 2685–2692.

[28] X. He, Y. Zhou, Enhancing the performance of differential evolution with covariance matrix self-adaptation, Appl. Soft Comput. 64 (2018) 227-243.

[29] J. Vrugt, B. Robinson, J. Hyman, Self-adaptive multimethod search for global optimization in real-parameter spaces, IEEE Trans. Evol. Comput. 13 (2009) 243–259.

[30] F. Neri, C. Cotta, Memetic algorithms and memetic computing optimization: A literature review, Swarm Evol. Comput. 2 (2012) 1-14.

[31] A. LaTorre, S. Muelas, J-M. Peña, A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test, Soft Comput 15 (2011) 2187–2199.

[32] R. Tanabe, A.S. Fukunaga, Improving the search performance of shade using linear population size reduction, in: Evolutionary Computation (CEC), 2014 IEEE Congress on, IEEE, 2014, pp. 1658–1665.

[33]A. Auger, N. Hansen, A restart CMA evolution strategy with increasing population size, in: Proc of the IEEE Congress on Evolutionary Computation, Sep. 2005, pp. 1769–1776.

[34]N. Hansen, Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed, in: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, ACM, 2009, pp. 2389–2396.

[35]A. Draa, S. Bouzoubia, I. Boukhalfa, A sinusoidal differential evolution algorithm for numerical optimization, Appl. Soft Comput. 27 (2015) 99–126.

[36]J. Zhang, A. C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (2009) 945–958.

[37]R. Mallipeddi, P. N. Suganthan, Q.-K. Pan, M. F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft Comput. 11 (2011) 1679–1696.

[38]Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evol. Comput. 15 (2011) 55–66.

[39]P. P. Biswas, P. N. Suganthan, G. Wu, G. A. J. Amaratunga, Parameter estimation of solar cells using datasheet information with the application of an adaptive differential evolution algorithm, Renewable Energy, 132 (2019) 425-438.

[40]A. P. Piotrowski, J. J. Napiorkowski, Performance of the air2stream model that relates air and stream water temperatures depends on the calibration method, Journal of Hydrology, 561 (2018) 395–412.

[41]W. Du, Q. Miao, L. Tong, Y. Tang, Identification of fractional-order systems with unknown initial values and structure, Physics Letters A, 381 (2017) 1943–1949.

[42]R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution, In: Proceedings of the IEEE Congress on Evolutionary Computation 2013, June20–23, Cancún, México, 2013, pp.71–78.

[43]S. Elsayed, N. Hamza, R. Sarker, Testing united multi-operator evolutionary algorithms-II on single objective optimization problems, in: Proc of the IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, 2016, pp. 2966–2973.

[44]N. H. Awad, M. Z. Ali, P. N. Suganthan, R. G. Reynolds, An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems, in: Proc of the IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, 2016, pp. 2958–2965.

[45]J. Brest, M. S Maučec, B. Bošković, Single objective real-parameter optimization: algorithm jSO, in: Proc of the IEEE Congress on Evolutionary Computation, San Sebastian, 2017, pp. 1311–1318.

[46]A. P. Piotrowski, J. J. Napiorkowski, Some metaheuristics should be simplified, Inf. Sci. 427 (2018) 32-62.

[47] L. M. Zheng, S. X. Zhang, S. Y. Zheng, Y. M. Pan, Differential evolution algorithm with two-step subpopulation strategy and its application in microwave circuit designs, IEEE Trans. Ind. Inf. 12 (3) (2016) 911–923.

[48] S. X. Zhang, S. Y. Zheng, L. M. Zheng, An efficient multiple variants coordination framework for differential evolution, IEEE Trans. Cybernet. 47 (2017) 2780-2793.

[49] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (2006) 646–657.

[50] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471.

[51] R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems, Comput. Aided Design. 43 (3) (2011)303–315.

[52] Z. K. Peng, S. X. Zhang, S. Y. Zheng, Y. L. Long, Collective information based teaching-learning-based optimization for global optimization, Soft Comput, https://doi.org/10.1007/s00500-018-03741-2.

[53] J. J. Liang, B. Y. Qu , P. N. Suganthan, Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore (2013) .

[54] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, P. N. Suganthan, Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization, Nanyang Technol. Univ., Singapore, Nov. 2016.

[55] D. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures. London, U.K.: Chapman & Hall, 2003.

[56] S. García, A. Fernández, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability, Soft Comput. 13 (2009) 959–977.

[57] S. Das, P.N. Suganthan, Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems, Jadavpur University, Nanyang Technological University, Technical Report, 2010.

Supplemental file of "Selective-Candidate Framework with Similarity Selection Rule for Evolutionary Optimization"

Sheng Xin Zhang [a*], Wing Shing Chan [a], Zi Kang Peng [b], Shao Yong Zheng [b*], Kit Sang Tang [a]

[a] Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong
[b] School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, 510006, China

Review of Evolutionary Algorithms and Swarm Intelligences

We briefly review and present the flow of three popular EAs and SIs including DE, ES and PSO and then give the general procedures.

1. DE

Differential evolution (DE) as proposed by Storn and Price [1] is a simple yet powerful EA. At each generation g, three genetic operations, namely mutation, crossover, and selection are included.

Initialization: Given a D-dimensional minimization problem, DE starts with a population $P_0 = \{x_{1,0}, x_{2,0}, ..., x_{NP,0}\}$ of NP individuals which is uniformly sampled from the entire searching space.

Mutation: Mutation in DE is performed by combining a basic vector with one or more difference vectors to generate a mutant vector $v_{i,g}$ $\{i = 1, 2, ..., NP\}$. The classic "rand/1" mutation strategy is formulated as follows.

$$v_{i,g} = x_{r1,g} + F \times (x_{r2,g} - x_{r3,g}) \tag{1}$$

where r_1, r_2 and r_3 are three distinct integers within $[1, NP]$ and are different from the index i, while F is a mutation factor between 0 and 1.

Crossover: After mutation, crossover is performed between the mutant vector $v_{i,g}$ and the current vector $x_{i,g}$ to generate a trial vector $u_{i,g}$ as follows.

$$u_{i,j,g} = \begin{cases} v_{i,j,g} & \text{if } rand_j(0,1) \leq CR \text{ or } j = j_{rand} \\ x_{i,j,g} & \text{otherwise} \end{cases} \tag{2}$$

where $rand_j(0,1)$ is a uniform random number in $(0, 1)$, j_{rand} is a randomly generated integer from $[1, D]$, and CR is a crossover factor within $[0,1]$.

Selection: Selection compares the fitness of $u_{i,g}$ with that of the corresponding $x_{i,g}$ and selects the better one to enter into the next generation.

$$x_{i,g+1} = \begin{cases} u_{i,g} & \text{if } f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} & \text{otherwise} \end{cases} \tag{3}$$

2. ES

Evolution strategy (ES) first appeared in 1964 at the Technical University of Berlin (TUB), and was used to solve hydrodynamic problems [2]. Different versions of ES have been proposed since this first version. Generally, ES can be categorized according to the number of parents and offspring involved in each generation. (1+1)-ES includes only one parent, which generates one offspring for each generation by means of Gaussian mutation. $(\mu + 1)$-ES uses μ $(\mu > 1)$ parents to generate one offspring per generation. $(\mu + \lambda)$-ES utilizes μ parents to generate λ $(\lambda > \mu)$ offspring and then chooses μ individuals from the $(\mu + \lambda)$ individuals to enter next generation, while (μ, λ)-ES chooses μ individuals only from the λ offspring.

Initialization: Given a D-dimensional minimization problem, ES starts with an initial population $P_0 = \{x_{1,0}, x_{2,0}, ..., x_{\mu,0}\}$ of μ individuals. Each individual $x_{i,0} = [x_{i,1,0}, x_{i,2,0}, ..., x_{i,D,0}, \sigma_{i,1,0}, \sigma_{i,2,0}, ..., \sigma_{i,D,0}]$, $(i = 1, 2, ..., \mu)$ has D variables and D independent standard deviations. The initial standard deviation $\sigma_{i,0}$ is calculated as

$$\sigma_{i,0} = \frac{\Delta x_i}{\sqrt{D}} \tag{4}$$

where Δx_i is the Euclidian distance between $x_{i,0}$ and the fittest individual in the initial population.

Recombination: At each generation g, recombination is performed on two randomly selected individuals to produce a new individual $xr_{i,g}$ $\{i = 1, 2, ..., \lambda\}$. Different recombination strategies are specified as follows:

$$xr_{i,j,g} = \begin{cases} x_{p,j,g}, \text{ without recombination} \\ x_{p,j,g} \text{ or } x_{q,j,g}, \text{ discrete recombination} \\ x_{p,j,g} + \chi \cdot (x_{q,j,g} - x_{p,j,g}), \text{ intermediate recombination} \end{cases} \tag{5}$$

where p and q are the two distinct integers uniformly selected from the set $\{1, 2, …, \mu\}$, $j = 1, 2, …, D$ is the dimension to be recombined and χ is a constant value usually set to 0.5 [3].

Mutation: Following recombination, mutation is performed to generate λ mutant individuals $\boldsymbol{xm}_{i,g}\{i = 1, 2, …, \lambda\}$ as described by the following:

$$\sigma_{i,j,g} = \sigma_{i,j,g} \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \tag{6}$$

$$xm_{i,j,g} = xr_{i,j,g} + N(0, \sigma_{i,j,g}) \tag{7}$$

where $j = 1, 2, …, D$, $N(0,1)$ and $N_i(0,1)$ are two normal distributions, τ' and τ are constants usually set as unity.

Selection: Select μ fittest individuals from the set of $\mu + \lambda$ individuals $((\mu + \lambda)$-ES), or from the set of λ offspring produced by mutation $((\mu, \lambda)$-ES).

3. *PSO*

Particle swarm optimization (PSO) as proposed by Kennedy and Eberhart [4] imitates the swarm behavior of animals, such as birds flocking and fish schooling. Given a D-dimensional minimization problem, PSO explores the searching space by utilizing a swarm of NP particles with each particle associated with a velocity vector $\boldsymbol{v}_i = [v_{i1}, v_{i2},..., v_{iD}]$ and a position vector $\boldsymbol{x}_i = [x_{i1}, x_{i2},..., x_{iD}]$, $i = 1, 2, …, NP$. During the searching process, each individual historical best position vector is recorded in $\boldsymbol{pbest}_i = [p_{i1}, p_{i2},..., p_{iD}]$ and the global best position vector is stored in $\boldsymbol{gbest} = [gb_1, gb_2,..., gb_D]$. Based on $\boldsymbol{pbest}_i$ and $\boldsymbol{gbest}$, particles update their velocity and position at each iteration by using Eq. (8) and (9) respectively:

$$v_{ij} = w \times v_{ij} + c_1 \times r_{1j} \times (pbest_{ij} - x_{ij}) + c_2 \times r_{2j} \times (gbest_j - x_{ij}) \tag{8}$$

$$x_{ij} = x_{ij} + v_{ij} \tag{9}$$

where w is the inertia weight, c_1 and c_2 are the acceleration constants, which are commonly set to 2.0. r_{1j} and r_{2j} are two uniformly distributed random numbers within $(0, 1)$ for each dimension j. The updated velocity $\left|v_{ij}\right|$ on each dimension is bounded by a maximum value $V_{MAXj,.}$ If $\left|v_{ij}\right|$ exceeds V_{MAXj}, then it is set as $sign(v_{ij})\ V_{MAXj.}$

4. *General Procedures*

From above, the general procedures for EAs and SIs is summarized as **Algorithm 1.**

Algorithm 1. General Procedures of EAs and SIs

1: Initialize population $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{NP}\}$;
2: **While** the stopping criteria are not met **Do**
3: Determine the control parameters CP for genetic operations
   or social learning;
4: Produce a new population $\boldsymbol{Y}$ via genetic operations or social
   learning on $\boldsymbol{X}$;
5: Evaluate the fitness of $\boldsymbol{Y}$;
6: Select solutions as new $\boldsymbol{X}$ from $\boldsymbol{X} \cup \boldsymbol{Y}$ to enter next iteration.
7: **End While**
----

[1] R. Storn and K. Price, Differential evolution–A simple and efficient adaptive scheme for global optimization over continuous spaces, Berkeley, CA, Tech. Rep., 1995, tech. Rep. TR-95-012.
[2] T. Bäck and H.-P. Schwefel, An overview of evolutionary algorithms for parameter optimization, Evol. Comput., 1 (1993) 1–23.
[3] T. Bäck, Evolutionary Algorithms in Theory and Practice. London, U.K.: Oxford Univ. Press, 1996.
[4] J. Kennedy and R. C. Eberhart, Particle swarm optimization, in Proc. IEEE Int. Conf. Neural Netw., 4 (1995) 1942–1948.

SCSS variants:

The arrows "$\Leftarrow$" highlight the differences between the SCSS variants and the baseline algorithms.

Algorithm S1. SCSS-DE

1: Set the population size NP, initialize the population $\boldsymbol{P}_0 = \{\boldsymbol{x}_{1,0}, \boldsymbol{x}_{2,0}, ..., \boldsymbol{x}_{NP,0}\}$, set F and CR, set the generation counter $g = 0$;

2: Set GD; $\Leftarrow$

3: **While** the stopping criteria are not met **Do**

4:   Determine the fitness ranking $rank(i)$ of each individual i $\{i = 1, 2,..., NP\}$; $\Leftarrow$

5: **For** $m = 1: M$ $\Leftarrow$

6:   **For** $i = 1: NP$ **Do**

------------------------Mutation------------------------

7:     Generate a mutant vector $\boldsymbol{v}_i^m{}_{,g}$ using Eq. (1);

------------------------Crossover------------------------

8:     Generate a trial vector $\boldsymbol{u}_i^m{}_{,g}$ using Eq. (2);

9:     $dist_i^m$ = Euclidian distance ($\boldsymbol{u}_i^m{}_{,g}$, $\boldsymbol{x}_{i,g}$); $\Leftarrow$

10:   **End For**

11: **End For** $\Leftarrow$

12: **For** $i = 1: NP$ **Do**

13:   **If** $rank(i) \leq \text{ceil}(NP \times GD)$ $\Leftarrow$

14:     $index = \underset{m \in \{1,2,...,M\}}{\arg\min}(dist_i^m)$; $\Leftarrow$

15:     $\boldsymbol{u}_{i,g} = \boldsymbol{u}_i^{index}{}_{,g}$; $\Leftarrow$

16:   **Else** $\Leftarrow$

17:     $index = \underset{m \in \{1,2,...,M\}}{\arg\max}(dist_i^m)$; $\Leftarrow$

18:     $\boldsymbol{u}_{i,g} = \boldsymbol{u}_i^{index}{}_{,g}$; $\Leftarrow$

19:   **End If** $\Leftarrow$

20: **End For**

21:   Evaluate the fitness of $\boldsymbol{u}_{i,g}$ $\{i = 1, 2, ..., NP\}$;

------------------------Selection------------------------

22: **For** $i = 1: NP$ **Do**

23:   **If** $f(\boldsymbol{u}_{i,g}) \leq f(\boldsymbol{x}_{i,g})$

24:     $\boldsymbol{x}_{i,g+1} = \boldsymbol{u}_{i,g}$;

25:   **Else**

26:     $\boldsymbol{x}_{i,g+1} = \boldsymbol{x}_{i,g}$;

27:   **End If**

28: **End For**

29:   $g = g + 1$;

30: **End While**

Algorithm S2. SCSS-ES

1: Set the population size μ, initialize the population $P_0 = \{\boldsymbol{x}_{1,0}, \boldsymbol{x}_{2,0}, ..., \boldsymbol{x}_{\mu,0}\}$, set the generation counter $g = 0$;

2: Set GD; $\Leftarrow$

3: **While** the stopping criteria are not met **Do**

4: Determine the fitness ranking $RANK(k)$ of each individual k $\{k = 1, 2,..., \mu\}$; $\Leftarrow$

5: **For** $i = 1: \lambda$ **Do**

----------------------Recombination------------------

6:   Randomly choose p and q, use the pth and qth individuals from $\boldsymbol{P}_g$ to generate a new individual $\boldsymbol{xr}_{i,g}$ with the recombination strategy, i.e. Eq. (5);

7:   Calculate the fitness $rank(i)$ of individual $i\{i = 1, 2, ..., \lambda\}$ as $(RANK(p) + RANK(q))/2$; $\Leftarrow$

8: **End For**

9: **For** $m = 1: M$ $\Leftarrow$

10:   **For** $i = 1: \lambda$ **Do**

------------------------Mutation------------------------

11:   Use Eq. (6) and (7) to mutate the individual $\boldsymbol{xr}_{i,g}$ produced by recombination and generate a mutant individual $\boldsymbol{xm}_i^m{}_{,g}$;

12:   $dist_i^m$ = Euclidian distance ($\boldsymbol{xm}_i^m{}_{,g}$, $\boldsymbol{xr}_{i,g}$); $\Leftarrow$

13:   **End For**

14: **End For** $\Leftarrow$

15: **For** $i = 1: \lambda$ **Do**

16:   **If** $rank(i) \leq \text{ceil}(\lambda \times GD)$ $\Leftarrow$

17:     $index = \underset{m \in \{1,2,...,M\}}{\arg\min}(dist_i^m)$; $\Leftarrow$

18:     $\boldsymbol{xm}_{i,g} = \boldsymbol{xm}_i^{index}{}_{,g}$; $\Leftarrow$

19:   **Else**

20:     $index = \underset{m \in \{1,2,...,M\}}{\arg\max}(dist_i^m)$; $\Leftarrow$

21:     $\boldsymbol{xm}_{i,g} = \boldsymbol{xm}_i^{index}{}_{,g}$; $\Leftarrow$

22:   **End If** $\Leftarrow$

23: **End For**

24:   Evaluate the fitness of all the new individuals $\boldsymbol{xm}_{i,g}$ $\{i = 1, 2, ..., \lambda\}$;

------------------------Selection------------------------

25:   Select μ fittest individuals $\boldsymbol{x}_{i,g}$ $\{i = 1, 2, ..., \mu\}$ from the $\mu + \lambda$ individuals to form a new population $\boldsymbol{P}_{g+1}$.

26:   $g = g + 1$;

27: **End While**

Algorithm S3. SCSS-PSO

1: Set the swarm size NP, initialize positions $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{NP}\}$, initialize velocities $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_{NP}\}$, record each particle's historical best position in $\boldsymbol{pbest}_i$ and the global best position in $\boldsymbol{gbest}$, set w, c_1 and c_2, set iteration counter $IT = 0$;

2: Set GD; $\Leftarrow$

3: **While** the stopping criteria are not met **Do**

4: Determine the fitness ranking $rank(i)$ of each particle i $\{i = 1, 2, ..., NP\}$;

5: **For** $m = 1: M$ $\Leftarrow$

6:   **For** $i = 1: NP$ **Do**

7:   **For** $j = 1: D$ **Do**

8:     Update v_{ij}^m using Eq. (8);

9:     Adjust v_{ij}^m if it exceeds V_{MAXj};

10:     Update x_{ij}^m using Eq. (9);

11:   **End For**

12:   $dist_i^m$ = Euclidian distance ($\boldsymbol{x}_i^m$, $\boldsymbol{pbest}_i$); $\Leftarrow$

13:   **End For**

14: **End For** $\Leftarrow$

15: **For** $i = 1: NP$ **Do**

16:   **If** $rank(i) \leq \text{ceil}(NP \times GD)$ $\Leftarrow$

17:     $index = \underset{m \in \{1,2,...,M\}}{\arg\min}(dist_i^m)$; $\Leftarrow$

18:     $\boldsymbol{x}_i = \boldsymbol{x}_i^{index}$; $\Leftarrow$

```
19:      Else                                          ⇐
20:          index = arg max (dist_i^m) ;              ⇐
                     m∈{1,2,...,M}
21:          x_i = x_i^index;                          ⇐
22:      End If                                        ⇐
23:  End For
24: For i = 1: NP Do
25:     Evaluate the fitness of the new position x_i;
26:     If f (x_i) ≤ f (pbest_i)
27:         pbest_i = x_i;
28:     End If
29:     If f (x_i) ≤ f (gbest)
30:         gbest = x_i;
31:     End If
32: End For
33: IT = IT + 1;
34: End While
```

Remark 1: In SCSS framework, the control parameters that are actually used, cp_i of y_i should be determined (lines 15 and 19 in Algorithm 2) for the reason that different reproduction procedure m may use different CP and the CP may have further usages. For example, in the JADE and SHADE algorithms, control parameters F and CR are generated according to Cauchy and normal distributions, respectively and after the selection of DE, successful CP are archived to determine new location parameters of Cauchy and normal distributions. Thus, in SCSS, the generations of F and CR are independent in each reproduction procedure m and the successful CP that are actually used is archived. In Algorithms S1 and S3, this is not shown because the classic DE and PSO use pre-defined fixed CP, i.e. F and CR in DE and w, c_1 and c_2 in PSO.

Remark 2: In PSO, the personal best position of each particle is regarded as a current solution for the similarity calculation (line 12 in Algorithm S3).

Remark 3: Different from the one-to-one reproduction procedures in DE and PSO, λ offspring is generated by using μ parents in ES. Therefore, we treat the λ new individuals **XR** produced by recombination as the current solutions, and their fitness rankings are calculated to be the average ranking of the pth and qth individuals used to perform recombination (lines 6 and 7 in Algorithm S2).

TABLE CAPTIONS

TABLE S1 PERFORMANCE COMPARISONS OF FOUR SCSS-BASED **ADVANCED ALGORITHMS** WITH THE BASELINES ON 30-D CEC2014 BENCHMARK SET

TABLE S2 PERFORMANCE COMPARISONS OF FOUR SCSS-BASED **ADVANCED ALGORITHMS** WITH THE BASELINES ON 50-D CEC2014 BENCHMARK SET

TABLE S3 COMPARISONS RESULTS OF SCSS VARIANTS WITH DIFFERENT SS RULES AGAINST THE BASELINES ON 30-D CEC2014 TEST FUNCTIONS ($M = 2$ FOR ALL THE SCSS VARIANTS, BEST ENTRIES ARE HIGHLIGHTED)

TABLE S4 PERFORMANCE COMPARISONS OF SCSS-JADE WITH THREE VARIANTS ON 30-D CEC2014 BENCHMARK SET

TABLE S5 PERFORMANCE COMPARISON OF SCSS VARIANTS WITH DIFFERENT M SETTINGS WITH THE BASELINES (BEST ENTRIES ARE HIGHLIGHTED)

TABLE S6 PERFORMANCE COMPARISON BETWEEN SCSS VARIANTS WITH ADJACENT M SETTINGS

TABLE S7 PERFORMANCE COMPARISONS OF FOUR SCSS-BASED **TOP ALGORITHMS** WITH THE BASELINES ON 30-D CEC2014 BENCHMARK SET

TABLE S8 PERFORMANCE COMPARISONS OF FOUR SCSS-BASED **TOP ALGORITHMS** WITH THE BASELINES ON 50-D CEC2014 BENCHMARK SET

TABLE S9 PERFORMANCE COMPARISONS OF FOUR SCSS-BASED **ADVANCED ALGORITHMS** WITH THE BASELINES ON 30-D CEC2017 BENCHMARK SET

TABLE S10 PERFORMANCE COMPARISONS OF FOUR SCSS-BASED **ADVANCED ALGORITHMS** WITH THE BASELINES ON 50-D CEC2017 BENCHMARK SET

TABLE S11 PERFORMANCE COMPARISONS OF FOUR SCSS-BASED **TOP ALGORITHMS** WITH THE BASELINES ON 30-D CEC2017 BENCHMARK SET

TABLE S12 PERFORMANCE COMPARISONS OF FOUR SCSS-BASED **TOP ALGORITHMS** WITH THE BASELINES ON 50-D CEC2017 BENCHMARK SET

TABLE S13 PERFORMANCE COMPARISONS OF FOUR SCSS-BASED **TOP ALGORITHMS** WITH THE BASELINES ON 100-D CEC2017 BENCHMARK SET

TABLE S1 PERFORMANCE (MEAN(STD)) COMPARISONS OF FOUR SCSS-BASED ADVANCED ALGORITHMS WITH THE BASELINES ON 30-D CEC2014 BENCHMARK SET

		JADE	SCSS-JADE	SHADE	SCSS-SHADE	CMA-ES	SCSS-CMA-ES	LIPS	SCSS-LIPS
Unimodal Functions	cec14 F1	2.04E+03 = (2.59E+03)	1.47E+03 (2.14E+03)	1.61E+03 = (2.04E+03)	1.50E+03 (2.68E+03)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	2.84E+07 - (2.65E+07)	**5.42E+06 (6.50E+06)**
	cec14 F2	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)	2.58E+03 = (4.30E+03)	5.84E+03 (8.14E+03)
	cec14 F3	2.08E-05 - (1.13E-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)	3.93E+03 - (3.64E+03)	**2.13E+03 (1.95E+03)**
Simple Multimodal Functions	cec14 F4	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)	2.74E+02 - (1.13E+02)	**1.40E+02 (6.49E+01)**
	cec14 F5	2.03E+01 - (3.12E-02)	**2.03E+01 (7.09E-02)**	2.02E+01 - (2.78E-02)	**2.01E+01 (2.29E-02)**	**2.00E+01 + (3.27E-05)**	2.13E+01 (5.20E-01)	**2.00E+01 + (8.23E-05)**	2.09E+01 (4.90E-02)
	cec14 F6	8.76E+00 = (2.72E+00)	7.33E+00 (3.86E+00)	6.42E+00 - (3.15E+00)	**4.12E+00 (3.37E+00)**	4.12E+01 - (9.58E+00)	**4.19E+00 (5.18E+00)**	1.48E+01 - (2.70E+00)	**7.72E+00 (2.24E+00)**
	cec14 F7	3.38E-04 - (1.71E-03)	1.93E-04 (1.38E-03)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	1.64E-03 = (3.51E-03)	1.59E-03 (4.45E-03)	1.59E-03 - (4.86E-03)	2.37E-03 (4.57E-03)
	cec14 F8	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	4.08E+02 - (8.57E+01)	**2.31E+02 (2.00E+02)**	5.35E+01 - (1.26E+01)	**2.64E+01 (6.79E+00)**
	cec14 F9	2.58E+01 - (3.62E+00)	**2.13E+01 (4.82E+00)**	2.10E+01 - (3.81E+00)	**1.92E+01 (3.44E+00)**	6.35E+02 - (1.23E+02)	**2.17E+02 (2.74E+02)**	6.29E+01 - (1.82E+01)	**3.62E+01 (8.74E+00)**
	cec14 F10	**4.49E-03 + (1.05E-02)**	9.39E-03 (1.52E-02)	5.31E-03 = (1.01E-02)	7.76E-03 (1.17E-02)	4.92E+03 - (7.43E+02)	**3.49E+03 (1.10E+03)**	1.97E+03 - (4.14E+02)	**9.61E+02 (2.63E+02)**
	cec14 F11	1.66E+03 - (2.67E+02)	**1.54E+03 (2.28E+02)**	1.48E+03 = (2.35E+02)	1.50E+03 (2.02E+02)	5.10E+03 - (8.25E+02)	**3.58E+03 (1.15E+03)**	2.54E+03 - (4.39E+02)	**2.02E+03 (4.10E+02)**
	cec14 F12	2.60E-01 - (4.06E-02)	**2.27E-01 (4.87E-02)**	2.10E-01 = (2.67E-02)	**1.68E-01 (2.45E-02)**	3.76E-01 - (4.02E-01)	**2.40E-01 (1.01E+00)**	1.78E-01 = (4.81E-02)	7.59E-01 (1.02E+00)
	cec14 F13	2.10E-01 - (3.53E-02)	**1.85E-01 (3.68E-02)**	2.23E-01 - (3.61E-02)	**2.04E-01 (3.18E-02)**	**2.62E-01 + (7.72E-02)**	4.24E-01 (1.46E-01)	3.06E-01 - (6.43E-02)	**2.75E-01 (5.22E-02)**
	cec14 F14	2.24E-01 = (3.09E-02)	2.32E-01 (3.71E-02)	2.27E-01 - (3.04E-02)	**2.09E-01 (3.26E-02)**	**3.71E-01 + (9.68E-02)**	5.66E-01 (2.97E-01)	**2.45E-01 + (3.56E-02)**	3.10E-01 (7.15E-02)
	cec14 F15	3.11E+00 - (4.17E-01)	**2.86E+00 (3.22E-01)**	2.97E+00 - (3.67E-01)	**2.59E+00 (3.03E-01)**	3.49E+00 - (7.56E-01)	3.21E+00 (6.63E-01)	1.08E+01 - (3.87E+00)	**3.92E+00 (8.93E-01)**
	cec14 F16	9.49E+00 = (3.17E-01)	9.34E+00 (4.29E-01)	9.51E+00 = (3.99E-01)	9.50E+00 (4.24E-01)	1.43E+01 - (4.33E-01)	**1.38E+01 (7.44E-01)**	1.15E+01 - (4.96E-01)	**1.06E+01 (4.65E-01)**
Hybrid Functions	cec14 F17	1.24E+03 - (3.35E+02)	**8.28E+02 (3.47E+02)**	9.44E+02 - (3.12E+02)	**5.78E+02 (2.32E+02)**	1.56E+03 = (4.64E+02)	1.71E+03 (3.84E+02)	2.89E+05 - (3.04E+05)	**1.86E+05 (2.99E+05)**
	cec14 F18	2.11E+02 - (8.15E+02)	**4.72E+01 (2.34E+01)**	3.44E+01 - (1.74E+01)	**2.05E+01 (1.20E+01)**	**1.35E+02 + (4.50E+01)**	1.78E+02 (7.13E+01)	4.88E+02 = (7.08E+02)	4.92E+02 (9.08E+02)
	cec14 F19	4.52E+00 - (6.74E-01)	**4.01E+00 (8.54E-01)**	3.95E+00 = (4.72E-01)	3.84E+00 (6.58E-01)	1.01E+01 - (2.11E+00)	**6.74E+00 (1.58E+00)**	2.54E+01 - (2.49E+01)	**8.85E+00 (2.76E+00)**
	cec14 F20	2.02E+03 = (2.81E+03)	1.88E+03 (2.44E+03)	1.09E+01 - (4.61E+00)	**8.41E+00 (3.45E+00)**	2.89E+02 - (1.01E+02)	**1.49E+02 (5.45E+01)**	1.47E+04 = (7.71E+03)	1.23E+04 (7.41E+03)
	cec14 F21	4.07E+03 - (1.89E+04)	**2.41E+02 (1.15E+02)**	2.13E+02 = (1.01E+02)	1.90E+02 (1.12E+02)	1.04E+03 - (3.50E+02)	**8.64E+02 (3.05E+02)**	1.11E+05 - (8.42E+04)	**4.26E+04 (5.58E+04)**
	cec14 F22	1.30E+02 = (6.92E+01)	1.10E+02 (6.90E+01)	6.36E+01 = (4.93E+01)	7.12E+01 (6.10E+01)	3.07E+02 - (2.29E+02)	**1.16E+02 (1.10E+02)**	3.27E+02 - (1.20E+02)	**2.28E+02 (1.10E+02)**
Composition Functions	cec14 F23	3.15E+02 = (4.02E-13)	3.15E+02 (4.02E-13)	3.15E+02 = (4.02E-13)	3.15E+02 (4.02E-13)	**3.15E+02 + (3.15E-12)**	3.15E+02 (2.57E-11)	3.24E+02 - (5.26E+00)	**3.16E+02 (5.73E-01)**
	cec14 F24	2.26E+02 = (3.11E+00)	2.25E+02 (3.27E+00)	2.24E+02 = (1.01E+00)	2.24E+02 (1.21E+00)	2.33E+02 - (6.83E+00)	**2.26E+02 (6.96E+00)**	2.39E+02 - (4.83E+00)	**2.33E+02 (5.09E+00)**
	cec14 F25	2.05E+02 - (2.18E+00)	**2.03E+02 (6.04E-01)**	2.04E+02 - (1.04E+00)	**2.03E+02 (4.63E-01)**	2.04E+02 - (2.42E+00)	**2.03E+02 (5.20E-01)**	2.16E+02 - (3.59E+00)	2.11E+02 (1.97E+00)
	cec14 F26	1.00E+02 - (3.77E-02)	**1.00E+02 (3.56E-02)**	1.00E+02 - (3.26E-02)	**1.00E+02 (3.42E-02)**	1.31E+02 - (1.37E+02)	**1.26E+02 (1.58E+02)**	1.32E+02 - (4.40E+01)	**1.09E+02 (2.68E+01)**
	cec14 F27	3.60E+02 - (5.07E+01)	3.44E+02 (5.09E+01)	3.16E+02 = (3.71E+01)	3.21E+02 (4.03E+01)	4.40E+02 - (2.10E+02)	**3.40E+02 (3.93E+01)**	6.03E+02 - (1.66E+02)	**4.79E+02 (9.74E+01)**
	cec14 F28	7.99E+02 = (2.34E+01)	8.01E+02 (1.64E+01)	7.95E+02 = (1.99E+01)	7.93E+02 (2.17E+01)	4.43E+03 - (3.23E+03)	**1.25E+03 (1.41E+03)**	1.78E+03 - (3.95E+02)	**1.12E+03 (1.70E+02)**
	cec14 F29	7.33E+02 - (1.60E+01)	**7.20E+02 (7.10E+01)**	7.25E+02 - (1.02E+01)	**7.12E+02 (5.40E+01)**	7.88E+02 - (9.18E+01)	8.00E+02 (1.45E+02)	1.34E+04 - (5.19E+04)	**1.29E+03 (2.46E+02)**
	cec14 F30	1.55E+03 = (6.33E+02)	1.53E+03 (6.34E+02)	1.45E+03 - (6.13E+02)	**1.19E+03 (3.57E+02)**	2.30E+03 - (5.50E+02)	**1.58E+03 (5.95E+02)**	3.84E+04 - (2.59E+04)	**1.08E+04 (6.59E+03)**
-/=/+		**14/15/1**		**14/16/0**		**17/8/5**		**23/5/2**	

TABLE S2 PERFORMANCE COMPARISONS OF FOUR SCSS-BASED ADVANCED ALGORITHMS WITH THE BASELINES
ON 50-D CEC2014 BENCHMARK SET

		JADE	SCSS-JADE	SHADE	SCSS-SHADE	CMA-ES	SCSS-CMA-ES	LIPS	SCSS-LIPS
Unimodal Functions	F1 cec14	1.88E+04 = (1.26E+04)	1.97E+04 (1.52E+04)	2.24E+04 = (1.14E+04)	2.66E+04 (1.09E+04)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	1.29E+08 - (7.81E+07)	**8.45E+06 (1.32E+07)**
	F2 cec14	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)	**7.57E+02 + (1.40E+03)**	1.72E+03 (2.71E+03)
	F3 cec14	3.06E+03 = (2.03E+03)	**2.01E+03 (2.98E+03)**	3.13E-06 - (1.39E-05)	**1.02E-07 (3.42E-07)**	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	1.67E+04 - (6.05E+03)	**1.14E+04 (5.51E+03)**
Simple Multimodal Functions	F4 cec14	1.37E+01 = (3.36E+01)	2.32E+01 (4.20E+01)	2.81E+01 = (4.30E+01)	**3.08E+01 (4.60E+01)**	3.28E+01 = (4.68E+01)	1.35E+01 (3.42E+01)	7.09E+02 - (3.77E+02)	**2.08E+02 (5.28E+01)**
	F5 cec14	2.04E+01 - (3.27E-02)	**2.02E+01 (2.06E-01)**	2.02E+01 = (2.34E-02)	**2.02E+01 (2.30E-02)**	**2.00E+01 + (1.77E-06)**	2.14E+01 (3.67E-01)	**2.00E+01 + (1.49E-05)**	2.11E+01 (3.62E-02)
	F6 cec14	1.59E+01 = (6.47E+00)	1.67E+01 (6.84E+00)	6.87E+00 = (5.99E+00)	5.35E+00 (4.96E+00)	7.68E+01 - (1.08E+01)	**1.74E+01 (1.85E+01)**	3.71E+01 - (4.26E+00)	**2.33E+01 (3.96E+00)**
	F7 cec14	4.15E-03 = (5.75E-03)	2.42E-03 (4.81E-03)	1.59E-03 = (3.91E-03)	1.69E-03 (4.22E-03)	5.32E-04 - (2.22E-03)	6.77E-04 (2.42E-03)	5.88E-03 - (1.93E-02)	**7.25E-04 (2.57E-03)**
	F8 cec14	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	7.39E+02 - (1.09E+02)	**6.12E+02 (2.31E+02)**	1.44E+02 - (1.89E+01)	**6.73E+01 (1.23E+01)**
	F9 cec14	5.43E+01 - (7.72E+00)	**3.86E+01 (8.83E+00)**	4.03E+01 = (5.05E+00)	3.95E+01 (5.80E+00)	1.13E+03 - (2.41E+02)	**5.88E+02 (4.78E+02)**	1.81E+02 - (2.84E+01)	**1.08E+02 (2.14E+01)**
	F10 cec14	1.05E-02 = (9.47E-03)	1.25E-02 (1.56E-02)	5.14E-03 = (8.35E-03)	9.06E-03 (1.30E-02)	8.43E+03 - (9.42E+02)	**7.21E+03 (1.17E+03)**	4.33E+03 - (5.04E+02)	**2.52E+03 (4.62E+02)**
	F11 cec14	3.82E+03 - (2.72E+02)	**3.53E+03 (2.87E+02)**	3.65E+03 = (3.25E+02)	3.55E+03 (3.46E+02)	8.23E+03 - (9.32E+02)	**7.25E+03 (1.10E+03)**	5.15E+03 - (4.95E+02)	**4.20E+03 (6.68E+02)**
	F12 cec14	2.61E-01 = (3.01E-02)	**2.14E-01 (7.30E-02)**	2.07E-01 = (2.79E-02)	**1.71E-01 (2.59E-02)**	2.71E-01 - (2.55E-01)	**7.63E-02 (4.56E-01)**	2.63E-01 = (7.48E-02)	6.84E-01 (1.12E+00)
	F13 cec14	3.13E-01 = (4.70E-02)	**2.75E-01 (3.91E-02)**	3.20E-01 = (3.32E-02)	3.12E-01 (4.02E-02)	**3.48E-01 + (7.71E-02)**	8.08E-01 (1.59E-01)	4.31E-01 = (5.93E-02)	4.12E-01 (5.70E-02)
	F14 cec14	3.00E-01 = (2.93E-02)	3.18E-01 (9.22E-02)	2.86E-01 = (6.25E-02)	2.69E-01 (4.02E-02)	**4.43E-01 + (2.50E-01)**	1.26E+00 (4.03E-01)	**2.71E-01 + (3.14E-02)**	3.48E-01 (1.19E-01)
	F15 cec14	7.27E+00 = (8.65E-01)	**5.94E+00 (6.97E-01)**	6.35E+00 = (7.66E-01)	**5.66E+00 (5.90E-01)**	6.41E+00 = (1.25E+00)	6.02E+00 (1.20E+00)	7.62E+01 - (4.32E+01)	**1.20E+01 (2.95E+00)**
	F16 cec14	1.77E+01 = (5.34E-01)	1.80E+01 (1.05E+00)	1.79E+01 = (4.14E-01)	1.79E+01 (3.62E-01)	2.38E+01 = (5.19E-01)	2.40E+01 (6.18E-01)	2.05E+01 - (6.41E-01)	**1.94E+01 (6.37E-01)**
Hybrid Functions	F17 cec14	2.29E+03 = (6.74E+02)	2.53E+03 (7.80E+02)	2.74E+03 = (8.65E+02)	2.74E+03 (8.27E+02)	2.69E+03 = (6.15E+02)	2.60E+03 (5.98E+02)	4.00E+06 - (5.97E+06)	**7.38E+05 (1.42E+06)**
	F18 cec14	1.64E+02 = (4.16E+01)	1.66E+02 (4.06E+01)	1.47E+02 = (4.44E+01)	1.39E+02 (4.31E+01)	**2.30E+02 + (4.57E+01)**	2.67E+02 (7.08E+01)	3.26E+02 - (1.64E+02)	**2.53E+02 (7.76E+01)**
	F19 cec14	1.48E+01 = (5.97E+00)	**1.06E+01 (5.22E+00)**	1.63E+01 - (7.08E+00)	**1.28E+01 (4.48E+00)**	1.84E+01 - (2.57E+00)	**1.46E+01 (2.30E+00)**	5.78E+01 - (2.86E+01)	**4.25E+01 (2.26E+01)**
	F20 cec14	8.19E+03 = (6.72E+03)	**1.99E+03 (4.70E+03)**	1.92E+02 = (6.69E+01)	**1.10E+02 (4.37E+01)**	4.44E+02 - (1.22E+02)	**2.71E+02 (8.53E+01)**	3.02E+04 - (1.09E+04)	**1.91E+04 (7.19E+03)**
	F21 cec14	1.29E+03 - (4.85E+02)	**2.36E+04 (1.61E+05)**	1.40E+03 - (4.92E+02)	**1.01E+03 (3.33E+02)**	1.70E+03 = (4.32E+02)	1.62E+03 (3.71E+02)	5.78E+05 - (4.16E+05)	**1.71E+05 (1.07E+05)**
	F22 cec14	4.78E+02 = (1.66E+02)	**3.76E+02 (1.61E+02)**	3.76E+02 = (1.18E+02)	3.38E+02 (1.09E+02)	4.19E+02 = (2.61E+02)	**3.20E+02 (2.11E+02)**	8.43E+02 - (2.08E+02)	**5.69E+02 (1.88E+02)**
Composition Functions	F23 cec14	**3.44E+02 + (4.55E-13)**	3.44E+02 (5.16E-13)	3.44E+02 = (4.31E-13)	3.44E+02 (4.50E-13)	3.44E+02 = (2.32E-05)	3.44E+02 (2.38E-05)	3.77E+02 - (1.34E+01)	**3.50E+02 (1.83E+00)**
	F24 cec14	2.74E+02 = (2.05E+00)	2.75E+02 (1.89E+00)	2.73E+02 = (1.93E+00)	2.72E+02 (1.89E+00)	3.67E+02 - (5.44E+02)	**2.76E+02 (2.43E+00)**	2.95E+02 - (6.01E+00)	**2.80E+02 (3.16E+00)**
	F25 cec14	2.23E+02 - (3.19E+00)	**2.11E+02 (6.51E+00)**	2.18E+02 = (5.01E+00)	**2.11E+02 (6.05E+00)**	2.05E+02 - (9.61E-01)	**2.05E+02 (2.18E-01)**	2.40E+02 - (8.81E+00)	**2.25E+02 (4.59E+00)**
	F26 cec14	1.04E+02 - (1.95E+01)	**1.00E+02 (8.92E-02)**	1.02E+02 = (1.40E+01)	**1.00E+02 (5.89E-02)**	**1.17E+02 + (5.81E+01)**	1.09E+02 (4.04E+01)	1.66E+02 - (4.65E+01)	**1.36E+02 (4.82E+01)**
	F27 cec14	4.65E+02 = (5.76E+01)	**4.35E+02 (5.42E+01)**	3.91E+02 = (4.89E+01)	3.79E+02 (4.65E+01)	5.33E+02 - (1.06E+02)	**4.57E+02 (7.00E+01)**	1.39E+03 - (1.29E+02)	**9.91E+02 (8.80E+01)**
	F28 cec14	1.15E+03 = (3.72E+01)	**1.12E+03 (3.47E+01)**	1.13E+03 = (4.00E+01)	1.11E+03 (3.05E+01)	7.61E+03 - (5.87E+03)	**4.39E+03 (2.98E+03)**	4.52E+03 - (7.42E+02)	**2.55E+03 (3.27E+02)**
	F29 cec14	8.81E+02 = (5.80E+01)	8.94E+02 (9.69E+01)	9.01E+02 = (6.55E+01)	9.02E+02 (6.54E+01)	8.86E+02 = (6.70E+01)	8.94E+02 (8.74E+01)	8.33E+06 - (4.37E+07)	**2.09E+03 (5.43E+02)**
	F30 cec14	9.78E+03 - (7.82E+02)	**9.26E+03 (8.07E+02)**	9.35E+03 - (6.62E+02)	**8.87E+03 (6.64E+02)**	9.31E+03 = (7.96E+02)	9.45E+03 (1.09E+03)	2.84E+05 - (1.17E+05)	**6.41E+04 (2.21E+04)**
-/=/+		**16/13/1**		**11/19/0**		**13/12/5**		**26/1/3**	

TABLE S3 COMPARISONS RESULTS OF SCSS VARIANTS WITH DIFFERENT SS RULES AGAINST THE BASELINES ON 30-D CEC2014 TEST FUNCTIONS ($M = 2$ FOR ALL THE SCSS VARIANTS, BEST ENTRIES ARE HIGHLIGHTED)

-/=/+ (P-N)	Scheme 1						Scheme 2
	$GD = 0$	$GD = 0.2$	$GD = 0.4$	$GD = 0.6$	$GD = 0.8$	$GD = 1.0$	
DE	0/5/25(-25)	1/13/16(-15)	2/21/7(-5)	11/19/0(11)	19/11/0 (19)	**21/8/1(20)**	5/21/4(1)
ES	0/3/27 (-27)	25/5/0 (25)	**26/4/0 (26)**	**26/4/0 (26)**	**26/4/0 (26)**	**26/4/0 (26)**	**26/4/0 (26)**
PSO	0/4/26 (-26)	10/15/5(5)	10/18/2(8)	14/16/0(14)	14/16/0(14)	**15/15/0(15)**	13/17/0(13)
JADE	14/9/7 (7)	15/11/4 (11)	**19/8/3 (16)**	15/14/1 (14)	5/11/14 (-9)	2/6/22 (-20)	14/15/1 (13)
SHADE	12/12/6 (6)	14/13/3 (11)	**15/14/1 (14)**	**14/16/0 (14)**	5/21/4 (1)	3/9/18 (-15)	**14/16/0 (14)**
CMA-ES	**13/15/2 (11)**	6/23/1 (5)	0/30/0 (0)	1/25/4 (-3)	1/19/10 (-9)	1/21/8 (-7)	2/26/2 (0)
LIPS	16/5/9 (7)	22/4/4 (18)	22/5/3 (19)	22/5/3 (19)	21/5/4 (17)	20/8/2 (18)	**23/4/3 (20)**

TABLE S4 PERFORMANCE COMPARISONS OF SCSS-JADE WITH THREE VARIANTS ON 30-D CEC2014 BENCHMARK SET

	Variant-oppo	Variant-Meval	Variant-CSM	SCSS-JADE		Variant-oppo	Variant-Meval	Variant-CSM	SCSS-JADE
F1 cec14	1.81E+05 - (1.28E+06)	2.41E+03 = (3.07E+03)	**5.50E+00 + (1.56E+01)**	1.47E+03 (2.14E+03)	F16 cec14	9.91E+00 - (2.48E-01)	9.52E+00 - (3.18E-01)	9.28E+00 = (3.48E-01)	9.34E+00 (4.29E-01)
F2 cec14	0.00E+00 = (0.00E+00)	0.00E+00 = (0.00E+00)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	F17 cec14	2.85E+05 - (4.29E+05)	1.17E+03 - (4.00E+02)	2.50E+04 - (1.46E+05)	**8.28E+02 (3.47E+02)**
F3 cec14	2.90E+00 - (3.03E+00)	1.49E-02 - (4.36E-02)	**0.00E+00 + (0.00E+00)**	**0.00E+00 (0.00E+00)**	F18 cec14	2.85E+03 - (3.60E+03)	9.30E+01 - (2.05E+02)	1.61E+02 - (2.26E+02)	**4.72E+01 (2.34E+01)**
F4 cec14	**0.00E+00 = (0.00E+00)**	5.49E-09 = (3.92E-08)	1.24E+00 - (8.88E+00)	**0.00E+00 (0.00E+00)**	F19 cec14	4.86E+00 - (7.86E-01)	4.29E+00 = (6.58E-01)	4.84E+00 - (7.48E-01)	**4.01E+00 (8.54E-01)**
F5 cec14	2.03E+01 - (2.70E-02)	2.03E+01 - (2.83E-02)	2.03E+01 - (3.27E-02)	**2.03E+01 (7.09E-02)**	F20 cec14	3.53E+03 - (2.22E+03)	3.21E+03 - (2.01E+03)	3.18E+03 - (2.43E+03)	**1.88E+03 (2.44E+03)**
F6 cec14	1.24E+01 - (1.20E+00)	1.02E+01 - (1.96E+00)	7.04E+00 = (3.90E+00)	7.33E+00 (3.86E+00)	F21 cec14	7.95E+04 - (8.65E+04)	3.49E+04 - (5.81E+04)	2.30E+04 - (6.33E+04)	**2.41E+02 (1.15E+02)**
F7 cec14	0.00E+00 = (0.00E+00)	1.45E-04 = (1.04E-03)	1.45E-04 = (1.04E-03)	1.93E-04 (1.38E-03)	F22 cec14	1.64E+02 - (7.95E+01)	1.20E+02 = (7.57E+01)	1.59E+02 - (7.21E+01)	**1.10E+02 (6.90E+01)**
F8 cec14	0.00E+00 = (0.00E+00)	0.00E+00 = (0.00E+00)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	F23 cec14	3.15E+02 = (2.48E-11)	3.15E+02 = (4.02E-13)	3.15E+02 = (4.02E-13)	3.15E+02 (4.02E-13)
F9 cec14	3.91E+01 - (5.50E+00)	2.88E+01 - (4.10E+00)	2.48E+01 - (4.32E+00)	**2.13E+01 (4.82E+00)**	F24 cec14	2.26E+02 - (3.01E+00)	**2.25E+02 - (2.76E+00)**	2.27E+02 - (3.30E+00)	**2.25E+02 (3.27E+00)**
F10 cec14	**2.45E-03 + (6.77E-03)**	7.35E-03 = (1.24E-02)	1.55E-02 - (1.71E-02)	9.39E-03 (1.52E-02)	F25 cec14	2.05E+02 - (2.05E+00)	2.04E+02 - (1.30E+00)	2.07E+02 - (2.00E+00)	**2.03E+02 (6.04E-01)**
F11 cec14	2.24E+03 - (1.84E+02)	1.87E+03 - (2.53E+02)	1.65E+03 - (2.38E+02)	**1.54E+03 (2.28E+02)**	F26 cec14	1.00E+02 - (3.71E-02)	1.00E+02 - (2.52E-02)	1.02E+02 - (1.40E+01)	**1.00E+02 (3.56E-02)**
F12 cec14	3.76E-01 - (3.71E-02)	3.12E-01 - (5.18E-02)	2.55E-01 - (3.58E-02)	**2.27E-01 (4.87E-02)**	F27 cec14	3.61E+02 - (5.23E+01)	3.64E+02 - (5.32E+01)	3.21E+02 = (2.90E+01)	3.44E+02 (5.09E+01)
F13 cec14	2.59E-01 - (3.58E-02)	2.06E-01 - (2.93E-02)	1.98E-01 - (3.61E-02)	**1.85E-01 (3.68E-02)**	F28 cec14	8.15E+02 - (1.91E+01)	8.02E+02 = (1.75E+01)	8.02E+02 = (4.59E+01)	**8.01E+02 (1.64E+01)**
F14 cec14	2.46E-01 - (3.02E-02)	2.29E-01 - (3.45E-02)	2.85E-01 - (8.52E-02)	**2.32E-01 (3.71E-02)**	F29 cec14	1.28E+03 - (4.43E+02)	7.29E+02 = (1.19E+01)	7.89E+02 = (2.20E+02)	**7.20E+02 (7.10E+01)**
F15 cec14	4.30E+00 - (4.90E-01)	3.55E+00 - (3.24E-01)	3.25E+00 - (3.55E-01)	**2.86E+00 (3.22E-01)**	F30 cec14	1.97E+03 = (6.55E+02)	1.64E+03 = (6.52E+02)	2.11E+03 - (6.37E+02)	**1.53E+03 (6.34E+02)**
-/=/+	**24/5/1**	16/14/0	18/11/1						

TABLE S5 PERFORMANCE COMPARISON OF SCSS VARIANTS WITH DIFFERENT *M* SETTINGS
WITH THE BASELINES (BEST ENTRIES ARE HIGHLIGHTED)

-/=/+ (P-N)	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 10$
DE	21/8/1 (20)	25/4/1 (24)	23/6/1 (22)	26/3/1 (25)	**27/2/1 (26)**
ES	26/4/0 (26)	**27/3/0 (27)**	**27/3/0 (27)**	27/2/1 (26)	27/2/1 (26)
PSO	**15/15/0 (15)**	13/16/1(12)	12/17/1(11)	10/18/2(8)	10/17/3 (7)
JADE	14/15/1 (13)	**16/13/1 (15)**	14/15/1 (13)	13/14/3 (10)	8/10/12 (-4)
SHADE	**14/16/0 (14)**	13/15/2 (11)	14/13/3 (11)	12/14/4 (8)	12/8/10 (2)
CMA-ES	13/15/2 (11)	15/11/4 (11)	15/11/4 (11)	**17/8/5 (12)**	17/7/6 (11)
LIPS	23/4/3 (20)	23/4/3 (20)	**23/5/2(21)**	23/4/3 (20)	20/5/5 (15)

TABLE S6 PERFORMANCE COMPARISON BETWEEN SCSS VARIANTS
WITH ADJACENT *M* SETTINGS

CATEGORY 1				
-/=/+ (P-N)	$M = 2$ v.s. $M = 3$	$M = 3$ v.s. $M = 4$	$M = 4$ v.s. $M = 5$	$M = 5$ v.s. $M = 10$
DE	17/12/1 (16)	5/24/1 (4)	6/23/1 (5)	12/14/4 (8)
ES	25/5/0 (25)	23/7/0(23)	12/18/0 (12)	28/2/0 (28)
CMA-ES	8/18/4 (4)	7/20/3(4)	1/29/0 (1)	5/18/7 (-2)
LIPS	8/21/1 (7)	2/28/0 (2)	0/29/1 (-1)	3/23/4 (-1)
CATEGORY 2				
-/=/+ (P-N)	$M = 2$ v.s. $M = 3$	$M = 3$ v.s. $M = 4$	$M = 4$ v.s. $M = 5$	$M = 5$ v.s. $M = 10$
PSO	1/22/7 (-6)	1/28/1 (0)	0/29/1 (-1)	0/28/2 (-2)
JADE	3/25/2 (1)	5/22/3 (2)	4/18/8 (-4)	5/7/18 (-13)
SHADE	4/24/2 (2)	4/23/3 (1)	7/15/8 (-1)	5/13/12 (-7)

TABLE S7 PERFORMANCE COMPARISONS OF FOUR SCSS-BASED TOP ALGORITHMS WITH THE BASELINES
ON 30-D CEC2014 BENCHMARK SET

		L-SHADE	SCSS-L-SHADE	UMOEA-II	SCSS-UMOEA-II	L-SHADE_EpSin	SCSS-L-SHADE_EpSin	jSO	SCSS-jSO
Unimodal Functions	F1 cec14	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)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
	F2 cec14	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)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
	F3 cec14	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)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
Simple Multimodal Functions	F4 cec14	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)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
	F5 cec14	2.01E+01 - (3.46E-02)	**2.01E+01 (5.37E-02)**	2.00E+01 (1.03E-03)	2.00E+01 (4.78E-05)	2.01E+01 - (2.98E-02)	**2.01E+01 (4.75E-02)**	2.09E+01 (8.04E-02)	2.09E+01 (4.80E-02)
	F6 cec14	9.01E-03 = (6.43E-02)	9.01E-03 (6.43E-02)	1.99E-01 = (1.35E+00)	4.24E-06 (1.86E-05)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	8.61E-06 = (3.52E-05)	1.02E-02 (7.27E-02)
	F7 cec14	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)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
	F8 cec14	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)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
	F9 cec14	7.22E+00 = (1.33E+00)	7.38E+00 (1.63E+00)	8.97E+00 = (1.79E+00)	9.03E+00 (2.07E+00)	1.31E+01 - (1.94E+00)	**1.24E+01 (2.15E+00)**	8.76E+00 = (1.97E+00)	**7.57E+00 (1.62E+00)**
	F10 cec14	5.72E-03 = (1.11E-02)	7.35E-03 (1.37E-02)	1.63E-03 = (5.65E-03)	4.08E-03 (8.35E-03)	4.49E-03 = (9.60E-03)	4.90E-03 (1.07E-02)	1.43E+00 = (1.02E+00)	1.64E+00 (9.94E-01)
	F11 cec14	1.23E+03 = (1.92E+02)	1.24E+03 (1.85E+02)	1.41E+03 = (3.01E+02)	1.43E+03 (3.18E+02)	1.14E+03 = (2.09E+02)	1.16E+03 (2.03E+02)	1.20E+03 = (2.73E+02)	1.26E+03 (2.45E+02)
	F12 cec14	1.73E-01 = (2.13E-02)	1.65E-01 (3.01E-02)	1.01E-01 = (5.51E-02)	1.08E-01 (6.90E-02)	1.54E-01 = (2.30E-02)	1.46E-01 (2.77E-02)	**4.17E-01 + (4.93E-01)**	9.00E-01 (7.61E-01)
	F13 cec14	1.05E-01 = (1.35E-02)	1.08E-01 (1.56E-02)	1.14E-01 = (1.81E-02)	1.09E-01 (2.15E-02)	1.34E-01 - (1.64E-02)	**1.24E-01 (1.61E-02)**	**1.37E-01 + (2.24E-02)**	1.52E-01 (3.04E-02)
	F14 cec14	2.38E-01 - (2.69E-02)	**1.90E-01 (2.41E-02)**	2.29E-01 - (2.52E-02)	**2.10E-01 (3.27E-02)**	1.93E-01 = (2.91E-02)	1.93E-01 (2.44E-02)	2.26E-01 = (4.08E-02)	2.30E-01 (3.63E-02)
	F15 cec14	2.28E+00 - (2.93E-01)	**2.16E+00 (2.47E-01)**	2.44E+00 = (4.60E-01)	2.29E+00 (5.34E-01)	2.37E+00 - (2.41E-01)	**2.24E+00 (2.91E-01)**	2.37E+00 - (2.73E-01)	**2.13E+00 (3.37E-01)**
	F16 cec14	**8.51E+00 + (3.61E-01)**	8.65E+00 (4.00E-01)	**9.15E+00 + (5.25E-01)**	9.57E+00 (6.20E-01)	8.30E+00 = (4.58E-01)	8.26E+00 (3.76E-01)	8.58E+00 = (7.71E-01)	8.60E+00 (7.27E-01)
Hybrid Functions	F17 cec14	2.09E+02 - (1.13E+02)	**8.89E+01 (4.59E+01)**	1.29E+02 - (7.85E+01)	**7.77E+01 (4.25E+01)**	1.94E+02 - (8.71E+01)	**1.42E+02 (8.41E+01)**	6.38E+01 = (2.31E+01)	6.22E+01 (2.13E+01)
	F18 cec14	6.89E+00 - (3.23E+00)	**3.01E+00 (1.50E+00)**	4.85E+00 = (1.76E+00)	**3.89E+00 (1.47E+00)**	6.02E+00 = (2.44E+00)	5.68E+00 (2.09E+00)	2.14E+00 = (1.23E+00)	2.19E+00 (1.17E+00)
	F19 cec14	3.75E+00 - (5.74E-01)	**3.08E+00 (6.64E-01)**	2.69E+00 - (6.23E-01)	**2.23E+00 (6.65E-01)**	2.63E+00 = (8.21E-01)	2.78E+00 (6.45E-01)	2.04E+00 = (7.16E-01)	1.86E+00 (6.30E-01)
	F20 cec14	2.84E+00 = (1.04E+00)	2.59E+00 (1.07E+00)	3.57E+00 = (1.41E+00)	3.72E+00 (1.34E+00)	2.34E+00 = (1.06E+00)	2.67E+00 (1.18E+00)	2.04E+00 = (8.67E-01)	1.97E+00 (8.07E-01)
	F21 cec14	9.08E+01 - (7.29E+01)	**3.33E+01 (5.40E+01)**	7.84E+01 - (7.25E+01)	**2.43E+01 (4.11E+01)**	9.09E+01 = (7.94E+01)	9.96E+01 (8.91E+01)	2.86E+01 = (4.42E+01)	1.18E+01 (8.29E+00)
	F22 cec14	2.45E+01 = (3.35E+00)	**2.31E+01 (2.00E+00)**	3.43E+01 = (2.47E+01)	**2.54E+01 (4.05E+00)**	5.17E+01 - (5.09E+01)	**3.76E+01 (3.85E+01)**	2.91E+01 - (2.45E+01)	**2.31E+01 (3.73E+00)**
Composition Functions	F23 cec14	3.15E+02 = (4.02E-13)	3.15E+02 (3.18E-13)	3.15E+02 = (4.02E-13)	3.15E+02 (4.02E-13)	3.15E+02 = (4.02E-13)	3.15E+02 (4.16E-13)	3.15E+02 = (4.16E-13)	3.15E+02 (4.02E-13)
	F24 cec14	2.24E+02 - (1.46E+00)	**2.22E+02 (3.44E+00)**	2.24E+02 - (1.95E+00)	**2.22E+02 (4.63E+00)**	2.11E+02 - (1.10E+01)	2.11E+02 (1.10E+01)	2.09E+02 - (1.08E+01)	**2.02E+02 (5.83E+00)**
	F25 cec14	2.03E+02 - (5.33E-02)	**2.03E+02 (4.10E-02)**	2.03E+02 - (3.95E-02)	**2.03E+02 (4.46E-02)**	2.03E+02 = (3.95E-02)	2.03E+02 (3.24E-02)	2.03E+02 = (2.75E-02)	2.03E+02 (2.60E-02)
	F26 cec14	1.00E+02 = (1.47E-02)	1.00E+02 (1.38E-02)	1.00E+02 = (1.92E-02)	1.00E+02 (1.98E-02)	1.00E+02 - (1.25E-02)	**1.00E+02 (1.64E-02)**	1.00E+02 = (2.13E-02)	1.00E+02 (2.44E-02)
	F27 cec14	**3.00E+02 + (1.25E-13)**	3.00E+02 (2.16E-13)	3.02E+02 = (1.40E+01)	3.02E+02 (1.40E+01)	3.00E+02 - (1.85E-13)	**3.00E+02 (9.09E-14)**	3.00E+02 = (2.30E-13)	3.00E+02 (1.23E-05)
	F28 cec14	8.35E+02 = (1.83E+01)	8.33E+02 (1.96E+01)	8.39E+02 = (1.42E+01)	8.35E+02 (1.53E+01)	8.37E+02 = (1.56E+01)	8.37E+02 (1.81E+01)	8.25E+02 - (2.15E+01)	**8.16E+02 (1.94E+01)**
	F29 cec14	7.16E+02 = (2.52E+00)	7.15E+02 (1.55E+00)	7.17E+02 - (3.10E+00)	**7.16E+02 (2.28E+00)**	7.22E+02 = (1.17E+01)	7.20E+02 (6.36E+00)	7.16E+02 - (2.07E+00)	**7.15E+02 (1.17E+00)**
	F30 cec14	1.40E+03 = (6.66E+02)	1.37E+03 (6.31E+02)	9.28E+02 = (3.55E+02)	9.35E+02 (4.83E+02)	1.46E+03 = (6.33E+02)	1.51E+03 (6.72E+02)	6.20E+02 - (1.67E+02)	**5.70E+02 (1.73E+02)**
-/=/+		**10/18/2**		9/20/1		**8/22/0**		7/21/2	

Note: The structural bias that affects the performance of UMOEA-II and L-SHADE_EpSin were removed according to the suggestions in [5]. In detail, in UMOEA-II and SCSS-UMOEA-II, the mutation strategy $V_{i,g} = F_i \times X_{r1,g} + (X_{r2,g} - X_{r3,g})$ was modified as $V_{i,g} = X_{r1,g} + (X_{r2,g} - X_{r3,g})$ by setting $F_i = 1$. In L-SHADE_EpSin and SCSS-L-SHADE_EpSin, the local search procedures were skipped.

[5] A. P. Piotrowski and J. J. Napiorkowski, Some metaheuristics should be simplified, Inf. Sci, 427 (2018) 32-62.

TABLE S8 PERFORMANCE COMPARISONS OF FOUR SCSS-BASED TOP ALGORITHMS WITH THE BASELINES
ON 50-D CEC2014 BENCHMARK SET

		L-SHADE	SCSS-L-SHADE	UMOEA-II	SCSS-UMOEA-II	L-SHADE_EpSin	SCSS-L-SHADE_EpSin	jSO	SCSS-jSO
Unimodal Functions	F1 cec14	9.71E+02 - (1.66E+03)	**1.04E+02 (5.89E+02)**	1.17E-03 - (9.11E-04)	**5.83E-04 (3.83E-04)**	1.33E-02 - (7.34E-02)	**5.13E-05 (3.62E-04)**	1.49E+01 - (3.06E+01)	**1.59E+00 (2.80E+00)**
	F2 cec14	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)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
	F3 cec14	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)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
Simple Multimodal Functions	F4 cec14	8.23E+01 = (3.38E+01)	7.62E+01 (4.00E+01)	**2.69E+01 + (4.42E+01)**	5.00E+01 (4.95E+01)	5.65E+01 - (4.83E+01)	**4.12E+01 (4.81E+01)**	5.02E+01 = (4.93E+01)	5.79E+01 (4.86E+01)
	F5 cec14	2.03E+01 - (3.08E-02)	**2.02E+01 (8.40E-02)**	2.00E+01 (6.24E-04)	2.00E+01 (4.88E-06)	2.03E+01 - (3.24E-02)	**2.02E+01 (7.18E-02)**	2.11E+01 = (5.59E-02)	2.11E+01 (5.17E-02)
	F6 cec14	9.14E-02 - (2.74E-01)	**5.69E-02 (2.45E-01)**	3.49E-01 - (4.91E-01)	**8.13E-02 (3.21E-01)**	2.04E-04 - (2.15E-04)	**2.14E-05 (4.97E-05)**	3.80E-03 - (5.50E-03)	**3.66E-02 (1.44E-01)**
	F7 cec14	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)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
	F8 cec14	3.64E-08 - (3.14E-08)	**2.37E-08 (4.23E-08)**	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	3.53E-09 = (2.52E-08)	0.00E+00 (0.00E+00)	**0.00E+00 + (0.00E+00)**	1.82E-09 (6.41E-09)
	F9 cec14	1.26E+01 - (2.44E+00)	**1.18E+01 (2.02E+00)**	1.60E+01 - (4.61E+00)	**1.39E+01 (3.94E+00)**	3.03E+01 - (5.20E+00)	**1.90E+01 (5.72E+00)**	1.59E+01 - (3.69E+00)	**1.13E+01 (2.93E+00)**
	F10 cec14	1.72E-01 - (5.24E-02)	**1.38E-01 (5.18E-02)**	**1.30E+00 + (1.19E+00)**	3.53E+00 (2.29E+00)	4.17E-02 = (2.19E-02)	3.73E-02 (1.78E-02)	9.92E+00 = (3.90E+00)	8.40E+00 (3.24E+00)
	F11 cec14	3.42E+03 - (3.46E+02)	**3.28E+03 (3.38E+02)**	3.94E+03 - (7.60E+02)	3.93E+03 (6.03E+02)	3.09E+03 - (3.06E+02)	3.00E+03 (3.23E+02)	3.22E+03 = (3.37E+02)	3.26E+03 (3.75E+02)
	F12 cec14	2.44E-01 - (3.53E-02)	**2.11E-01 (3.26E-02)**	1.63E-01 - (1.06E-01)	1.68E-01 (1.06E-01)	2.16E-01 - (2.70E-02)	**1.99E-01 (2.81E-02)**	3.69E-01 = (4.10E-01)	7.48E-01 (7.45E-01)
	F13 cec14	1.60E-01 - (1.74E-02)	**1.50E-01 (2.08E-02)**	1.63E-01 = (2.40E-02)	1.60E-01 (2.33E-02)	2.06E-01 - (2.08E-02)	**1.90E-01 (2.35E-02)**	1.92E-01 = (2.83E-02)	2.01E-01 (4.22E-02)
	F14 cec14	3.23E-01 - (4.96E-02)	**2.49E-01 (9.34E-02)**	3.01E-01 - (2.29E-02)	**2.63E-01 (2.99E-02)**	1.89E-01 - (2.33E-02)	**1.84E-01 (3.13E-02)**	2.91E-01 - (4.34E-02)	**2.73E-01 (4.15E-02)**
	F15 cec14	5.30E+00 - (5.66E-01)	**4.99E+00 (4.75E-01)**	5.39E+00 - (1.04E+00)	5.13E+00 (1.06E+00)	5.68E+00 - (4.74E-01)	**5.04E+00 (5.05E-01)**	5.18E+00 - (4.85E-01)	**4.68E+00 (6.92E-01)**
	F16 cec14	**1.69E+01 + (4.35E-01)**	1.71E+01 (4.88E-01)	**1.84E+01 + (7.63E-01)**	1.86E+01 (6.65E-01)	1.67E+01 - (3.44E-01)	**1.65E+01 (4.28E-01)**	1.70E+01 + (9.41E-01)	1.73E+01 (7.30E-01)
Hybrid Functions	F17 cec14	1.63E+03 - (3.52E+02)	**5.59E+02 (2.32E+02)**	1.11E+03 - (3.60E+02)	**3.94E+02 (1.81E+02)**	3.60E+02 = (1.60E+02)	3.51E+02 (1.39E+02)	3.51E+02 = (1.70E+02)	**1.76E+02 (1.11E+02)**
	F18 cec14	1.05E+02 - (1.38E+01)	**2.30E+01 (6.42E+00)**	5.70E+01 - (2.14E+01)	**1.56E+01 (4.28E+00)**	1.89E+01 = (6.40E+00)	1.83E+01 (6.76E+00)	1.08E+01 - (3.24E+00)	**7.21E+00 (2.16E+00)**
	F19 cec14	**8.11E+00 + (1.87E+00)**	9.64E+00 (1.45E+00)	8.17E+00 = (2.20E+00)	7.66E+00 (2.39E+00)	9.99E+00 - (8.84E-01)	**9.76E+00 (8.22E-01)**	9.25E+00 - (8.19E-01)	8.56E+00 (7.29E-01)
	F20 cec14	1.45E+01 - (3.75E+00)	**7.96E+00 (1.96E+00)**	1.34E+01 - (3.52E+00)	**9.33E+00 (3.05E+00)**	6.04E+00 = (2.23E+00)	5.93E+00 (1.86E+00)	5.67E+00 = (1.95E+00)	5.17E+00 (1.71E+00)
	F21 cec14	5.59E+02 - (1.62E+02)	**3.42E+02 (1.11E+02)**	4.38E+02 - (1.27E+02)	**3.49E+02 (1.32E+02)**	3.25E+02 = (9.65E+01)	3.08E+02 (1.05E+02)	3.03E+02 - (9.88E+01)	**2.36E+02 (8.45E+01)**
	F22 cec14	1.03E+02 = (7.30E+01)	9.95E+01 (7.03E+01)	1.81E+02 = (8.35E+01)	1.93E+02 (1.19E+02)	9.35E+01 = (6.13E+01)	**6.34E+01 (5.00E+01)**	1.51E+02 - (1.00E+02)	**1.03E+02 (8.34E+01)**
Composition Functions	F23 cec14	3.44E+02 - (3.20E-13)	3.44E+02 (3.46E-13)	3.44E+02 = (4.67E-13)	3.44E+02 (4.73E-13)	3.44E+02 = (2.93E-13)	3.44E+02 (3.18E-13)	3.44E+02 = (3.03E-13)	3.44E+02 (3.46E-13)
	F24 cec14	2.75E+02 - (4.98E-01)	**2.74E+02 (1.13E+00)**	2.75E+02 - (8.57E-01)	**2.75E+02 (7.27E-01)**	2.68E+02 = (1.23E+00)	2.68E+02 (1.50E+00)	2.72E+02 - (1.80E+00)	**2.70E+02 (2.18E+00)**
	F25 cec14	2.05E+02 - (3.48E-01)	**2.05E+02 (2.33E-01)**	2.05E+02 = (2.98E-01)	2.05E+02 (3.00E-01)	2.05E+02 = (1.39E-01)	2.05E+02 (9.28E-02)	2.05E+02 - (1.82E-01)	**2.05E+02 (1.35E-01)**
	F26 cec14	1.00E+02 = (1.98E-02)	1.00E+02 (1.66E-02)	1.00E+02 = (2.50E-02)	1.00E+02 (2.05E-02)	1.00E+02 - (4.98E-02)	**1.00E+02 (3.46E-02)**	1.00E+02 = (2.37E-02)	1.00E+02 (3.87E-02)
	F27 cec14	3.42E+02 - (2.68E+01)	**3.35E+02 (2.17E+01)**	3.34E+02 - (3.31E+01)	**3.23E+02 (2.59E+01)**	3.17E+02 = (2.28E+01)	3.25E+02 (2.34E+01)	3.10E+02 = (1.85E+01)	**3.10E+02 (1.84E+01)**
	F28 cec14	1.13E+03 = (3.69E+01)	1.12E+03 (3.09E+01)	1.12E+03 = (2.83E+01)	1.11E+03 (2.69E+01)	1.14E+03 = (3.72E+01)	1.14E+03 (3.83E+01)	1.09E+03 = (2.81E+01)	1.08E+03 (3.04E+01)
	F29 cec14	8.04E+02 = (3.34E+01)	8.02E+02 (3.22E+01)	8.05E+02 = (4.27E+01)	7.95E+02 (3.95E+01)	8.05E+02 = (2.77E+01)	8.13E+02 (4.03E+01)	8.04E+02 = (4.11E+01)	8.03E+02 (4.48E+01)
	F30 cec14	8.59E+03 = (4.15E+02)	8.53E+03 (3.14E+02)	8.62E+03 = (4.71E+02)	8.64E+03 (5.04E+02)	8.50E+03 = (3.71E+02)	8.60E+03 (4.33E+02)	8.38E+03 = (3.90E+02)	8.30E+03 (3.38E+02)
-/=/+		**18/10/2**		**10/17/3**		**13/17/0**		**13/15/2**	

TABLE S9 PERFORMANCE COMPARISONS OF FOUR SCSS-BASED ADVANCED ALGORITHMS WITH THE BASELINES
ON 30-D CEC2017 BENCHMARK SET

		JADE	SCSS-JADE	SHADE	SCSS-SHADE	CMA-ES	SCSS-CMA-ES	LIPS	SCSS-LIPS
Unimodal Functions	F1 cec17	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)	**8.03E+02** + **(1.45E+03)**	2.73E+03 (4.18E+03)
	F2 cec17	1.58E-05 = (8.56E-06)	1.70E-05 (9.99E-06)	1.77E-05 = (1.03E-05)	1.39E-05 (8.49E-06)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	2.33E+01 - (9.02E+01)	**1.87E-03** **(1.95E-04)**
	F3 cec17	1.18E+04 - (1.92E+04)	**7.74E+02** **(5.53E+03)**	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	1.60E+04 - (7.66E+03)	**7.74E+03** **(3.55E+03)**
Simple Multimodal Functions	F4 cec17	5.18E+01 = (2.08E+01)	5.14E+01 (2.06E+01)	5.47E+01 = (1.62E+01)	5.29E+01 (1.76E+01)	**3.99E+01** + **(2.74E+01)**	4.30E+01 (2.55E+01)	1.64E+02 - (9.39E+01)	**1.11E+02** **(4.93E+01)**
	F5 cec17	2.83E+01 - (4.01E+00)	**2.17E+01** **(4.50E+00)**	1.99E+01 = (3.24E+00)	1.97E+01 (3.18E+00)	6.58E+02 - (2.22E+02)	**1.34E+02** **(2.26E+02)**	6.43E+01 - (1.35E+01)	**3.43E+01** **(9.30E+00)**
	F6 cec17	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	9.91E+01 - (1.56E+01)	**3.99E+01** **(4.70E+01)**	8.27E+00 - (5.05E+00)	**4.58E-01** **(5.87E-01)**
	F7 cec17	5.61E+01 = (3.87E+00)	**5.19E+01** **(4.41E+00)**	5.09E+01 = (3.87E+00)	**4.92E+01** **(2.84E+00)**	3.66E+03 - (1.11E+03)	**2.71E+02** **(8.12E+02)**	9.77E+01 - (2.10E+01)	**7.32E+01** **(1.09E+01)**
	F8 cec17	2.84E+01 - (5.00E+00)	**2.39E+01** **(4.09E+00)**	2.16E+01 = (3.42E+00)	2.07E+01 (3.64E+00)	5.79E+02 - (1.43E+02)	**1.60E+02** **(2.05E+02)**	6.23E+01 - (1.31E+01)	**3.58E+01** **(8.35E+00)**
	F9 cec17	2.13E-02 = (9.01E-02)	7.02E-03 (2.43E-02)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	1.37E+04 - (3.23E+03)	**5.85E+03** **(7.13E+03)**	6.01E+02 - (4.21E+02)	**2.07E+01** **(2.53E+01)**
	F10 cec17	1.88E+03 = (2.70E+02)	**1.79E+03** **(2.39E+02)**	1.73E+03 = (2.71E+02)	1.72E+03 (2.46E+02)	4.93E+03 - (5.98E+02)	**4.05E+03** **(1.01E+03)**	2.80E+03 - (4.44E+02)	**2.15E+03** **(3.40E+02)**
Hybrid Functions	F11 cec17	3.37E+01 - (2.26E+01)	**2.28E+01** **(2.00E+01)**	2.10E+01 = (2.53E+01)	2.13E+01 (2.47E+01)	1.67E+02 - (5.67E+01)	**1.20E+02** **(3.97E+01)**	1.99E+02 - (1.41E+02)	**8.58E+01** **(4.31E+01)**
	F12 cec17	1.48E+03 = (8.87E+02)	1.30E+03 (7.31E+02)	2.03E+03 - (2.68E+03)	**1.20E+03** **(5.83E+02)**	1.51E+03 = (3.69E+02)	1.55E+03 (3.41E+02)	1.85E+06 - (6.00E+06)	**1.78E+05** **(2.11E+05)**
	F13 cec17	4.36E+01 = (2.16E+01)	3.92E+01 (1.61E+01)	3.84E+01 = (1.76E+01)	**2.68E+01** **(1.20E+01)**	1.57E+03 - (7.42E+02)	1.35E+03 (7.07E+02)	5.74E+03 - (5.63E+03)	**2.78E+03** **(4.82E+03)**
	F14 cec17	9.70E+03 - (1.12E+04)	**2.05E+03** **(7.03E+03)**	2.73E+01 = (5.83E+00)	2.61E+01 (4.08E+00)	1.85E+02 = (5.74E+01)	1.66E+02 (5.33E+01)	1.40E+04 - (1.13E+04)	**8.81E+03** **(2.02E+04)**
	F15 cec17	1.94E+03 - (3.78E+03)	**1.14E+02** **(6.60E+02)**	1.32E+01 = (9.70E+00)	1.05E+01 (5.76E+00)	3.09E+02 - (1.32E+02)	2.83E+02 (1.36E+02)	2.35E+03 - (3.05E+03)	**1.40E+03** **(2.16E+03)**
	F16 cec17	3.92E+02 - (1.27E+02)	**3.27E+02** **(1.28E+02)**	2.91E+02 = (1.16E+02)	**2.43E+02** **(1.35E+02)**	5.92E+02 - (2.96E+02)	**3.36E+02** **(2.36E+02)**	7.30E+02 - (2.21E+02)	**4.78E+02** **(1.61E+02)**
	F17 cec17	8.33E+01 - (2.86E+01)	**7.21E+01** **(2.09E+01)**	4.83E+01 = (1.29E+01)	5.10E+01 (9.63E+00)	2.80E+02 - (2.03E+02)	**1.45E+02** **(9.83E+01)**	2.89E+02 - (1.19E+02)	**1.52E+02** **(6.88E+01)**
	F18 cec17	5.06E+04 - (7.16E+04)	**7.69E+03** **(3.87E+04)**	7.32E+01 = (4.20E+01)	**3.43E+01** **(1.53E+01)**	2.07E+02 - (8.94E+01)	1.98E+02 (7.43E+01)	1.71E+05 - (1.53E+05)	**1.16E+05** **(6.72E+04)**
	F19 cec17	1.88E+03 - (4.75E+03)	**1.20E+01** **(6.37E+00)**	7.83E+00 = (3.06E+00)	7.40E+00 (2.40E+00)	2.04E+02 - (8.72E+01)	**1.73E+02** **(6.95E+01)**	1.55E+03 = (1.99E+03)	1.61E+03 (3.30E+03)
	F20 cec17	9.72E+01 - (5.22E+01)	**7.83E+01** **(4.58E+01)**	6.23E+01 = (3.64E+01)	5.43E+01 (3.33E+01)	1.38E+03 - (3.73E+02)	**2.05E+02** **(1.65E+02)**	3.21E+02 - (1.02E+02)	**1.83E+02** **(7.84E+01)**
Composition Functions	F21 cec17	2.28E+02 - (4.78E+00)	**2.22E+02** **(4.93E+00)**	2.21E+02 = (3.13E+00)	2.20E+02 (3.86E+00)	4.92E+02 - (2.67E+02)	**3.03E+02** **(1.56E+02)**	2.65E+02 - (1.55E+01)	**2.39E+02** **(9.85E+00)**
	F22 cec17	1.00E+02 = (2.56E-05)	1.39E+02 (2.76E+02)	1.00E+02 = (1.00E-13)	1.00E+02 (1.00E-13)	5.70E+03 - (1.03E+03)	**3.05E+03** **(2.50E+03)**	1.58E+02 - (4.06E+02)	**1.00E+02** **(2.11E-13)**
	F23 cec17	3.75E+02 - (6.33E+00)	**3.71E+02** **(6.99E+00)**	3.68E+02 = (4.87E+00)	3.66E+02 (5.71E+00)	1.99E+03 - (8.26E+02)	**6.46E+02** **(6.87E+02)**	4.45E+02 - (3.32E+01)	**3.91E+02** **(1.08E+01)**
	F24 cec17	4.40E+02 - (4.90E+00)	**4.36E+02** **(5.27E+00)**	4.38E+02 - (3.82E+00)	**4.36E+02** **(3.77E+00)**	4.74E+02 = (9.73E+01)	4.57E+02 (1.09E+01)	5.00E+02 - (2.83E+01)	**4.49E+02** **(1.02E+01)**
	F25 cec17	3.87E+02 = (1.86E-01)	**3.87E+02** **(1.72E-01)**	3.87E+02 - (1.38E-01)	**3.87E+02** **(1.33E-01)**	3.87E+02 - (2.74E+00)	**3.87E+02** **(2.71E-02)**	4.29E+02 - (2.71E+01)	3.99E+02 (1.32E+01)
	F26 cec17	1.19E+03 - (1.51E+02)	**1.16E+03** **(8.12E+01)**	1.12E+03 = (6.24E+01)	1.09E+03 (6.26E+01)	1.20E+03 - (4.75E+02)	**1.20E+03** **(3.22E+02)**	1.47E+03 - (8.10E+02)	**1.14E+03** **(5.73E+02)**
	F27 cec17	5.01E+02 = (7.16E+00)	5.03E+02 (7.65E+00)	5.02E+02 = (5.62E+00)	5.02E+02 (4.92E+00)	8.04E+02 - (1.74E+03)	**4.86E+02** **(1.08E+01)**	6.12E+02 - (2.52E+01)	**5.56E+02** **(1.69E+01)**
	F28 cec17	3.41E+02 = (5.64E+01)	3.34E+02 (5.44E+01)	3.34E+02 = (5.47E+01)	3.30E+02 (4.90E+01)	3.51E+02 = (6.13E+01)	3.42E+02 (5.34E+01)	5.00E+02 - (9.70E+01)	**3.90E+02** **(7.31E+01)**
	F29 cec17	4.85E+02 = (2.28E+01)	**4.74E+02** **(1.52E+01)**	4.63E+02 = (2.62E+01)	4.65E+02 (1.66E+01)	7.88E+02 - (1.84E+02)	**6.36E+02** **(1.25E+02)**	9.73E+02 - (1.78E+02)	**7.05E+02** **(7.69E+01)**
	F30 cec17	2.79E+03 = (2.00E+03)	2.13E+03 (1.42E+02)	2.10E+03 = (1.27E+02)	2.08E+03 (1.39E+02)	2.22E+03 = (2.09E+02)	2.19E+03 (2.20E+02)	1.19E+05 - (1.81E+05)	**1.20E+04** **(5.61E+03)**
-/=/+		**19/11/0**		**7/23/0**		**18/11/1**		**28/1/1**	

TABLE S10 Performance comparisons of four SCSS-based advanced algorithms with the baselines on 50-D CEC2017 benchmark set

		JADE	SCSS-JADE	SHADE	SCSS-SHADE	CMA-ES	SCSS-CMA-ES	LIPS	SCSS-LIPS
Unimodal Functions	F1 cec17	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)	**1.17E+03** + **(2.02E+03)**	2.89E+03 (4.25E+03)
	F2 cec17	**4.21E-05** + **(1.21E-05)**	4.93E-05 (1.63E-05)	5.08E-05 = (1.48E-05)	5.41E-05 (1.87E-05)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	7.62E+02 - (7.84E+02)	**3.25E-03** **(4.46E-04)**
	F3 cec17	1.42E+04 - (3.38E+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)	9.27E+04 (2.23E+04)	**6.53E+04** **(1.57E+04)**
Simple Multimodal Functions	F4 cec17	5.46E+01 = (5.18E+01)	5.37E+01 (5.01E+01)	6.40E+01 = (5.03E+01)	5.50E+01 (4.53E+01)	4.34E+01 = (4.79E+01)	3.61E+01 (4.31E+01)	6.66E+02 - (3.39E+02)	**2.52E+02** **(7.79E+01)**
	F5 cec17	5.18E+01 - (9.01E+00)	**3.98E+01** **(9.33E+00)**	4.35E+01 - (5.40E+00)	**3.89E+01** **(6.36E+00)**	1.03E+03 - (1.78E+02)	**6.32E+02** **(4.78E+02)**	1.68E+02 - (2.62E+01)	**1.00E+02** **(2.00E+01)**
	F6 cec17	**0.00E+00** + **(0.00E+00)**	5.77E-07 (2.18E-06)	1.59E-06 = (2.26E-06)	1.67E-06 (1.87E-06)	9.54E+01 - (1.04E+01)	**7.49E+01** **(3.66E+01)**	2.41E+01 - (5.43E+00)	**4.92E+00** **(2.13E+00)**
	F7 cec17	9.89E+01 - (8.16E+00)	**8.94E+01** **(8.04E+00)**	8.91E+01 - (5.48E+00)	**8.60E+01** **(5.82E+00)**	6.42E+03 - (1.55E+03)	**1.65E+03** **(2.74E+03)**	3.74E+02 - (6.09E+01)	**1.74E+02** **(2.69E+01)**
	F8 cec17	5.43E+01 - (8.64E+00)	**4.17E+01** **(8.53E+00)**	4.21E+01 = (6.54E+00)	4.10E+01 (7.27E+00)	1.09E+03 - (2.12E+02)	**5.94E+02** **(4.60E+02)**	1.74E+02 - (3.49E+01)	**1.02E+02** **(1.71E+01)**
	F9 cec17	1.44E+00 = (1.52E+00)	1.46E+00 (1.26E+00)	3.87E-01 = (3.94E-01)	3.55E-01 (4.33E-01)	3.08E+04 = (5.49E+03)	2.64E+04 (1.16E+04)	4.44E+03 - (1.45E+03)	**8.85E+02** **(5.90E+02)**
	F10 cec17	3.70E+03 - (3.77E+02)	**3.49E+03** **(3.97E+02)**	3.48E+03 = (3.77E+02)	3.43E+03 (3.50E+02)	8.04E+03 - (9.92E+02)	**7.19E+03** **(1.22E+03)**	5.14E+03 - (6.66E+02)	**4.24E+03** **(6.02E+02)**
Hybrid Functions	F11 cec17	1.57E+02 - (5.18E+01)	**1.32E+02** **(3.61E+01)**	8.67E+01 - (2.71E+01)	**6.88E+01** **(1.66E+01)**	2.88E+02 - (6.63E+01)	**2.08E+02** **(5.01E+01)**	2.35E+03 - (2.45E+03)	**2.58E+02** **(8.87E+01)**
	F12 cec17	7.02E+03 = (6.81E+03)	6.57E+03 (3.92E+03)	5.66E+03 = (3.09E+03)	6.95E+03 (4.86E+03)	2.66E+03 = (6.49E+02)	2.64E+03 (6.45E+02)	1.35E+07 - (4.17E+07)	**1.84E+06** **(1.55E+06)**
	F13 cec17	2.52E+02 = (1.52E+02)	2.10E+02 (1.23E+02)	2.94E+02 - (1.94E+02)	**1.33E+02** **(5.36E+01)**	2.55E+03 = (7.76E+02)	2.28E+03 (7.63E+02)	6.58E+03 - (3.64E+03)	**1.16E+03** **(7.74E+02)**
	F14 cec17	6.91E+04 - (1.19E+05)	**5.09E+03** **(2.12E+04)**	1.82E+02 - (4.59E+01)	**8.43E+01** **(2.75E+01)**	3.16E+02 = (7.64E+01)	2.97E+02 (9.08E+01)	1.32E+05 - (3.30E+05)	**2.61E+04** **(2.66E+04)**
	F15 cec17	1.13E+03 - (2.51E+03)	**1.92E+02** **(9.30E+01)**	2.52E+02 - (1.05E+02)	**1.28E+02** **(5.77E+01)**	4.88E+02 = (1.68E+02)	4.84E+02 (1.20E+02)	1.97E+03 - (1.89E+03)	**8.09E+02** **(6.53E+02)**
	F16 cec17	9.06E+02 - (1.65E+02)	**7.24E+02** **(1.67E+02)**	7.26E+02 = (1.83E+02)	7.44E+02 (1.31E+02)	9.06E+02 - (3.97E+02)	**5.49E+02** **(3.04E+02)**	1.44E+03 - (3.37E+02)	**9.12E+02** **(2.46E+02)**
	F17 cec17	6.40E+02 - (1.59E+02)	**5.52E+02** **(1.55E+02)**	4.78E+02 = (1.37E+02)	4.90E+02 (1.25E+02)	9.86E+02 - (2.57E+02)	**5.71E+02** **(2.25E+02)**	1.16E+03 - (2.11E+02)	**7.70E+02** **(1.70E+02)**
	F18 cec17	1.82E+05 - (4.33E+05)	**1.59E+02** **(1.54E+02)**	1.38E+02 - (8.50E+01)	**1.10E+02** **(7.29E+01)**	3.60E+02 = (1.23E+02)	3.31E+02 (1.07E+02)	1.21E+06 - (2.22E+06)	**3.56E+05** **(2.38E+05)**
	F19 cec17	9.41E+02 - (2.46E+03)	**1.19E+02** **(4.55E+01)**	1.14E+02 - (4.32E+01)	**7.53E+01** **(3.39E+01)**	2.71E+02 = (1.30E+02)	2.43E+02 (7.61E+01)	3.34E+03 = (4.99E+03)	3.26E+03 (5.11E+03)
	F20 cec17	4.74E+02 - (1.35E+02)	**3.97E+02** **(1.28E+02)**	3.46E+02 = (1.19E+02)	3.27E+02 (9.96E+01)	2.37E+03 - (5.04E+02)	**8.23E+02** **(8.32E+02)**	6.79E+02 - (1.67E+02)	**4.60E+02** **(1.57E+02)**
Composition Functions	F21 cec17	2.54E+02 = (1.03E+01)	**2.41E+02** **(8.60E+00)**	2.44E+02 = (6.19E+00)	2.42E+02 (7.15E+00)	7.97E+02 - (4.85E+02)	**4.13E+02** **(3.21E+02)**	3.60E+02 - (3.55E+01)	**3.01E+02** **(1.72E+01)**
	F22 cec17	3.68E+03 - (1.67E+03)	**3.41E+03** **(1.45E+03)**	3.50E+03 = (1.50E+03)	3.27E+03 (1.68E+03)	9.11E+03 - (1.09E+03)	**7.94E+03** **(1.30E+03)**	4.55E+03 - (2.41E+03)	**3.92E+03** **(1.87E+03)**
	F23 cec17	4.79E+02 = (1.09E+01)	**4.65E+02** **(1.01E+01)**	4.66E+02 - (8.46E+00)	**4.60E+02** **(8.48E+00)**	3.18E+03 - (6.79E+02)	**1.20E+03** **(1.18E+03)**	7.13E+02 - (6.14E+01)	**5.59E+02** **(2.46E+01)**
	F24 cec17	5.40E+02 - (8.46E+00)	**5.29E+02** **(6.59E+00)**	5.35E+02 - (8.93E+00)	**5.30E+02** **(6.90E+00)**	7.00E+02 - (2.49E+02)	**5.72E+02** **(2.19E+01)**	7.71E+02 - (7.71E+01)	**6.05E+02** **(1.99E+01)**
	F25 cec17	5.23E+02 = (3.28E+01)	5.20E+02 (3.62E+01)	5.15E+02 = (3.61E+01)	5.08E+02 (3.75E+01)	5.02E+02 = (3.32E+01)	4.94E+02 (2.97E+01)	9.66E+02 - (2.15E+02)	**6.35E+02** **(4.87E+01)**
	F26 cec17	1.63E+03 = (1.22E+02)	**1.50E+03** **(1.34E+02)**	1.45E+03 - (9.07E+01)	**1.41E+03** **(9.53E+01)**	1.90E+03 - (5.02E+02)	**1.76E+03** **(5.10E+02)**	3.87E+03 - (6.48E+02)	**2.19E+03** **(6.09E+02)**
	F27 cec17	5.58E+02 = (2.58E+01)	5.55E+02 (2.94E+01)	5.37E+02 = (1.88E+01)	5.31E+02 (1.33E+01)	7.55E+02 - (1.17E+03)	**4.76E+02** **(1.37E+01)**	1.19E+03 - (9.61E+01)	**8.66E+02** **(6.62E+01)**
	F28 cec17	4.91E+02 = (2.25E+01)	4.94E+02 (2.11E+01)	4.82E+02 = (2.44E+01)	4.85E+02 (2.38E+01)	4.70E+02 = (2.01E+01)	4.64E+02 (1.60E+01)	1.49E+03 - (4.96E+02)	**6.25E+02** **(5.57E+01)**
	F29 cec17	4.60E+02 = (6.92E+01)	4.72E+02 (7.48E+01)	4.38E+02 = (5.83E+01)	4.46E+02 (5.42E+01)	1.04E+03 - (2.96E+02)	**6.93E+02** **(1.73E+02)**	2.02E+03 - (3.35E+02)	**1.12E+03** **(1.80E+02)**
	F30 cec17	6.64E+05 = (9.01E+04)	6.56E+05 (8.03E+04)	6.57E+05 = (7.82E+04)	6.54E+05 (6.50E+04)	7.86E+05 = (1.45E+05)	7.87E+05 (1.72E+05)	3.31E+07 - (1.45E+07)	**4.90E+06** **(1.58E+06)**
-/=/+		**18/10/2**		**11/19/0**		**16/14/0**		**28/1/1**	

TABLE S11 Performance comparisons of four SCSS-based top algorithms with the baselines
on 30-D cec2017 benchmark set

		L-SHADE	SCSS-L-SHADE	UMOEA-II	SCSS-UMOEA-II	L-SHADE_EpSin	SCSS-L-SHADE_EpSin	jSO	SCSS-jSO
Unimodal Functions	cec17 F1	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)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
	cec17 F2	4.06E-09 - (8.59E-09)	**0.00E+00** **(0.00E+00)**	4.14E-08 = (5.51E-08)	3.23E-08 (5.00E-08)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	6.65E-08 = (9.56E-08)	9.39E-08 (9.54E-08)
	cec17 F3	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)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
Simple Multimodal Functions	cec17 F4	5.86E+01 = (3.75E-14)	5.86E+01 (3.27E-14)	5.86E+01 = (4.90E-14)	5.87E+01 (7.78E-01)	5.86E+01 = (2.88E-14)	5.86E+01 (2.93E-14)	5.86E+01 = (2.13E-14)	5.86E+01 (2.41E-14)
	cec17 F5	7.02E+00 = (1.52E+00)	7.61E+00 (1.58E+00)	8.29E+00 = (2.19E+00)	8.54E+00 (2.06E+00)	1.22E+01 - (1.60E+00)	**1.06E+01** **(2.43E+00)**	8.32E+00 - (1.74E+00)	**7.49E+00** **(1.80E+00)**
	cec17 F6	3.38E-09 = (1.98E-08)	1.14E-08 (3.73E-08)	1.81E-08 = (8.05E-08)	6.71E-09 (2.74E-08)	8.05E-09 = (3.25E-08)	0.00E+00 (0.00E+00)	9.39E-09 = (3.29E-08)	1.74E-08 (4.45E-08)
	cec17 F7	**3.79E+01** + **(1.18E+00)**	3.91E+01 (2.03E+00)	4.04E+01 = (2.73E+00)	4.06E+01 (2.68E+00)	4.35E+01 - (2.48E+00)	**4.19E+01** **(2.75E+00)**	3.84E+01 = (1.83E+00)	**3.75E+01** **(1.33E+00)**
	cec17 F8	7.11E+00 = (1.58E+00)	8.09E+00 (2.13E+00)	8.45E+00 = (1.86E+00)	8.54E+00 (2.36E+00)	1.35E+01 - (1.50E+00)	**1.26E+01** **(2.46E+00)**	8.81E+00 - (2.17E+00)	**7.57E+00** **(2.04E+00)**
	cec17 F9	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)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
	cec17 F10	1.41E+03 = (2.31E+02)	1.44E+03 (2.33E+02)	1.69E+03 = (3.17E+02)	1.63E+03 (3.04E+02)	1.35E+03 = (1.90E+02)	1.28E+03 (2.38E+02)	1.49E+03 = (2.66E+02)	1.54E+03 (2.18E+02)
Hybrid Functions	cec17 F11	3.73E+01 - (2.91E+01)	**3.36E+01** **(2.90E+01)**	1.34E+01 = (2.02E+01)	1.53E+01 (2.34E+01)	1.58E+01 = (2.30E+01)	1.97E+01 (2.55E+01)	9.87E+00 = (1.89E+01)	6.46E+00 (1.39E+01)
	cec17 F12	1.04E+03 - (3.37E+02)	**6.95E+02** **(3.16E+02)**	8.28E+02 - (3.18E+02)	**2.84E+02** **(1.85E+02)**	4.03E+02 = (2.22E+02)	3.77E+02 (2.15E+02)	1.66E+02 - (8.86E+01)	**8.34E+01** **(7.27E+01)**
	cec17 F13	1.92E+01 - (4.61E+00)	**1.73E+01** **(4.88E+00)**	1.53E+01 = (6.24E+00)	1.61E+01 (5.99E+00)	1.42E+01 = (6.02E+00)	1.54E+01 (5.86E+00)	1.60E+01 = (5.76E+00)	1.63E+01 (4.50E+00)
	cec17 F14	**2.19E+01** + **(1.22E+00)**	2.22E+01 (3.11E+00)	2.22E+01 = (3.42E+00)	2.22E+01 (4.58E+00)	2.13E+01 = (4.65E+00)	2.26E+01 (1.20E+00)	2.20E+01 = (1.08E+00)	2.14E+01 (3.19E+00)
	cec17 F15	3.54E+00 - (1.56E+00)	**2.80E+00** **(1.34E+00)**	3.30E+00 - (1.70E+00)	**2.83E+00** **(2.22E+00)**	2.41E+00 = (1.44E+00)	2.58E+00 (1.61E+00)	1.26E+00 = (8.34E-01)	1.03E+00 (8.73E-01)
	cec17 F16	4.00E+01 = (2.74E+01)	3.43E+01 (1.48E+01)	9.31E+01 = (9.08E+01)	7.11E+01 (8.16E+01)	5.09E+01 - (4.44E+01)	**3.12E+01** **(3.38E+01)**	6.50E+01 = (6.92E+01)	5.02E+01 (6.73E+01)
	cec17 F17	3.29E+01 = (6.27E+00)	3.44E+01 (5.90E+00)	**4.07E+01** + **(8.68E+00)**	4.46E+01 (1.00E+01)	2.83E+01 = (6.47E+00)	2.91E+01 (5.86E+00)	3.45E+01 - (7.04E+00)	**3.17E+01** **(7.19E+00)**
	cec17 F18	2.23E+01 - (1.28E+00)	**2.04E+01** **(2.79E+00)**	2.15E+01 = (6.94E-01)	2.13E+01 (7.26E-01)	2.13E+01 = (9.45E-01)	2.13E+01 (9.30E-01)	2.08E+01 = (3.79E-01)	1.95E+01 (4.82E+00)
	cec17 F19	5.96E+00 = (1.87E+00)	5.90E+00 (2.05E+00)	6.38E+00 = (1.91E+00)	7.13E+00 (2.35E+00)	5.24E+00 = (1.63E+00)	5.10E+00 (1.87E+00)	4.53E+00 = (1.90E+00)	4.06E+00 (1.43E+00)
	cec17 F20	3.01E+01 = (5.93E+00)	2.99E+01 (4.37E+00)	4.27E+01 = (9.05E+00)	3.97E+01 (7.88E+00)	2.83E+01 = (7.68E+00)	2.60E+01 (5.45E+00)	3.01E+01 = (8.53E+00)	2.75E+01 (7.25E+00)
Composition Functions	cec17 F21	2.08E+02 = (1.65E+00)	2.08E+02 (1.53E+00)	2.09E+02 = (2.11E+00)	2.10E+02 (2.43E+00)	2.12E+02 - (2.62E+00)	**2.10E+02** **(2.50E+00)**	2.09E+02 - (1.93E+00)	**2.08E+02** **(2.04E+00)**
	cec17 F22	1.00E+02 = (9.20E-14)	1.00E+02 (1.00E-13)	1.00E+02 = (1.39E-13)	1.00E+02 (1.87E-13)	1.00E+02 = (1.00E-13)	1.00E+02 (1.00E-13)	1.00E+02 = (9.20E-14)	1.00E+02 (1.00E-13)
	cec17 F23	3.54E+02 = (3.16E+00)	3.54E+02 (2.98E+00)	3.54E+02 = (4.25E+00)	3.54E+02 (3.85E+00)	3.55E+02 = (2.86E+00)	3.55E+02 (3.71E+00)	3.51E+02 - (3.46E+00)	**3.50E+02** **(3.15E+00)**
	cec17 F24	4.28E+02 = (1.58E+00)	4.28E+02 (1.87E+00)	**4.28E+02** + **(2.39E+00)**	4.29E+02 (2.35E+00)	4.29E+02 - (2.73E+00)	**4.27E+02** **(2.07E+00)**	4.26E+02 = (2.38E+00)	4.26E+02 (3.06E+00)
	cec17 F25	3.87E+02 - (1.97E-02)	**3.87E+02** **(1.26E-02)**	3.87E+02 - (2.43E-02)	**3.87E+02** **(1.71E-02)**	3.87E+02 = (5.91E-03)	3.87E+02 (5.70E-03)	3.87E+02 = (5.99E-03)	3.87E+02 (6.30E-03)
	cec17 F26	9.85E+02 - (3.55E+01)	**9.65E+02** **(3.66E+01)**	9.51E+02 = (3.60E+01)	9.52E+02 (4.31E+01)	9.55E+02 = (3.92E+01)	**9.35E+02** **(4.45E+01)**	9.30E+02 = (3.65E+01)	9.25E+02 (4.04E+01)
	cec17 F27	5.07E+02 = (4.03E+00)	5.06E+02 (5.63E+00)	5.03E+02 = (4.75E+00)	5.01E+02 (6.09E+00)	5.05E+02 = (4.52E+00)	5.05E+02 (4.34E+00)	4.97E+02 = (6.63E+00)	4.95E+02 (7.76E+00)
	cec17 F28	3.39E+02 = (5.61E+01)	3.27E+02 (4.88E+01)	3.20E+02 = (4.37E+01)	3.26E+02 (4.74E+01)	**3.06E+02** + **(2.63E+01)**	3.24E+02 (4.66E+01)	3.13E+02 = (3.54E+01)	3.02E+02 (1.60E+01)
	cec17 F29	**4.36E+02** + **(7.53E+00)**	4.42E+02 (1.15E+01)	**4.38E+02** + **(1.62E+01)**	4.45E+02 (1.19E+01)	**4.29E+02** + **(6.34E+00)**	4.35E+02 (8.65E+00)	4.32E+02 = (1.58E+01)	4.27E+02 (2.42E+01)
	cec17 F30	1.99E+03 - (5.56E+01)	**1.97E+03** **(4.32E+01)**	1.97E+03 = (3.05E+01)	1.98E+03 (3.66E+01)	1.99E+03 = (7.24E+01)	1.99E+03 (5.68E+01)	1.97E+03 = (1.68E+01)	1.97E+03 (1.11E+01)
-/=/+		**9/18/3**		**3/24/3**		**7/21/2**		**7/23/0**	

TABLE S12 PERFORMANCE COMPARISONS OF FOUR SCSS-BASED TOP ALGORITHMS WITH THE BASELINES
ON 50-D CEC2017 BENCHMARK SET

		L-SHADE	SCSS-L-SHADE	UMOEA-II	SCSS-UMOEA-II	L-SHADE_EpSin	SCSS-L-SHADE_EpSin	jSO	SCSS-jSO
Unimodal Functions	F1 cec17	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)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
	F2 cec17	5.01E-06 - (3.12E-06)	**1.66E-06** **(9.79E-07)**	1.37E-05 - (6.95E-06)	**6.55E-06** **(4.16E-06)**	2.23E-07 - (1.36E-07)	**9.62E-08** **(6.14E-08)**	1.38E-05 = (8.23E-06)	1.48E-05 (8.26E-06)
	F3 cec17	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	**3.00E-10** + **(2.14E-09)**	1.54E-08 (2.31E-08)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
Simple Multimodal Functions	F4 cec17	7.23E+01 = (4.94E+01)	7.34E+01 (5.05E+01)	7.22E+01 = (4.97E+01)	8.27E+01 (5.36E+01)	5.04E+01 = (4.38E+01)	4.51E+01 (3.97E+01)	5.85E+01 = (4.56E+01)	4.87E+01 (4.11E+01)
	F5 cec17	1.19E+01 = (2.46E+00)	1.20E+01 (1.99E+00)	1.61E+01 = (4.55E+00)	**1.43E+01** **(3.11E+00)**	2.90E+01 - (6.65E+00)	**1.94E+01** **(6.64E+00)**	1.56E+01 - (2.65E+00)	**1.26E+01** **(2.70E+00)**
	F6 cec17	7.12E-08 - (2.58E-07)	**2.22E-08** **(6.76E-08)**	1.66E-04 - (5.76E-04)	**1.16E-07** **(2.28E-07)**	2.57E-07 - (3.41E-07)	**4.20E-08** **(6.98E-08)**	4.10E-07 = (5.52E-07)	2.85E-07 (5.12E-07)
	F7 cec17	6.50E+01 = (2.23E+00)	6.46E+01 (2.12E+00)	7.04E+01 = (5.17E+00)	6.85E+01 (5.14E+00)	7.98E+01 - (7.02E+00)	**7.15E+01** **(5.69E+00)**	6.66E+01 = (3.10E+00)	**6.33E+01** **(2.66E+00)**
	F8 cec17	1.21E+01 = (2.39E+00)	1.17E+01 (2.56E+00)	1.58E+01 = (4.09E+00)	1.43E+01 (4.17E+00)	3.07E+01 - (3.99E+00)	**1.96E+01** **(6.59E+00)**	1.69E+01 - (3.43E+00)	**1.20E+01** **(2.67E+00)**
	F9 cec17	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)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
	F10 cec17	3.32E+03 - (2.81E+02)	**3.12E+03** **(3.27E+02)**	3.75E+03 = (5.99E+02)	3.64E+03 (5.22E+02)	3.07E+03 = (2.91E+02)	**2.89E+03** **(2.90E+02)**	3.21E+03 - (3.78E+02)	**3.05E+03** **(3.63E+02)**
Hybrid Functions	F11 cec17	4.80E+01 - (6.64E+00)	**3.37E+01** **(4.65E+00)**	4.42E+01 = (9.48E+00)	**3.16E+01** **(4.51E+00)**	2.75E+01 = (2.01E+00)	2.71E+01 (2.06E+00)	2.66E+01 - (3.13E+00)	**2.50E+01** **(4.12E+00)**
	F12 cec17	2.07E+03 = (5.21E+02)	2.10E+03 (4.81E+02)	2.17E+03 = (5.36E+02)	2.01E+03 (4.99E+02)	1.38E+03 = (3.79E+02)	1.36E+03 (3.67E+02)	1.61E+03 - (4.42E+02)	**1.29E+03** **(3.66E+02)**
	F13 cec17	6.52E+01 = (2.98E+01)	**5.09E+01** **(2.89E+01)**	4.69E+01 - (1.73E+01)	**3.56E+01** **(1.57E+01)**	3.76E+01 = (2.60E+01)	4.29E+01 (2.23E+01)	3.17E+01 = (2.01E+01)	2.60E+01 (2.09E+01)
	F14 cec17	3.06E+01 - (3.73E+00)	**2.48E+01** **(2.30E+00)**	2.85E+01 = (3.30E+00)	**2.70E+01** **(2.35E+00)**	2.71E+01 = (2.68E+00)	2.67E+01 (2.57E+00)	2.50E+01 = (2.34E+00)	2.51E+01 (2.46E+00)
	F15 cec17	4.53E+01 - (1.40E+01)	**2.77E+01** **(3.82E+00)**	3.45E+01 = (6.42E+00)	**2.69E+01** **(3.14E+00)**	2.51E+01 = (3.17E+00)	2.39E+01 (2.44E+00)	2.37E+01 - (2.77E+00)	**2.12E+01** **(1.81E+00)**
	F16 cec17	3.76E+02 = (1.36E+02)	3.49E+02 (1.17E+02)	4.58E+02 = (1.68E+02)	4.07E+02 (1.69E+02)	3.31E+02 - (1.25E+02)	**2.68E+02** **(1.16E+02)**	4.77E+02 = (1.36E+02)	4.45E+02 (1.55E+02)
	F17 cec17	2.32E+02 = (6.72E+01)	2.04E+02 (9.33E+01)	3.14E+02 = (1.18E+02)	3.01E+02 (1.07E+02)	2.40E+02 - (6.48E+01)	**2.04E+02** **(8.12E+01)**	2.93E+02 = (1.10E+02)	2.61E+02 (1.04E+02)
	F18 cec17	5.06E+01 - (1.72E+01)	**2.80E+01** **(3.87E+00)**	3.26E+01 = (7.70E+00)	**2.60E+01** **(2.90E+00)**	2.53E+01 = (2.70E+00)	2.46E+01 (2.15E+00)	2.46E+01 - (2.42E+00)	**2.24E+01** **(1.14E+00)**
	F19 cec17	3.50E+01 - (1.39E+01)	**1.71E+01** **(3.01E+00)**	2.08E+01 = (3.32E+00)	**1.70E+01** **(3.00E+00)**	1.62E+01 = (3.11E+00)	1.56E+01 (2.97E+00)	1.42E+01 - (2.73E+00)	**1.17E+01** **(2.65E+00)**
	F20 cec17	1.56E+02 = (4.95E+01)	1.72E+02 (6.37E+01)	2.60E+02 = (1.20E+02)	2.80E+02 (1.16E+02)	1.35E+02 - (5.03E+01)	**1.07E+02** **(2.47E+01)**	1.17E+02 = (6.45E+01)	1.14E+02 (6.57E+01)
Composition Functions	F21 cec17	2.16E+02 - (2.26E+00)	**2.14E+02** **(2.74E+00)**	2.20E+02 - (5.20E+00)	**2.18E+02** **(4.64E+00)**	2.30E+02 - (6.27E+00)	**2.20E+02** **(6.07E+00)**	2.17E+02 - (2.73E+00)	**2.14E+02** **(3.27E+00)**
	F22 cec17	2.84E+03 = (1.53E+03)	3.33E+03 (8.42E+02)	2.82E+03 = (2.11E+03)	2.78E+03 (2.16E+03)	1.54E+03 = (1.62E+03)	2.10E+03 (1.46E+03)	1.07E+03 = (1.61E+03)	1.63E+03 (1.79E+03)
	F23 cec17	4.33E+02 = (4.04E+00)	**4.30E+02** **(4.60E+00)**	4.42E+02 = (8.43E+00)	**4.37E+02** **(7.54E+00)**	4.43E+02 - (6.60E+00)	**4.35E+02** **(7.00E+00)**	4.30E+02 - (6.16E+00)	**4.26E+02** **(6.54E+00)**
	F24 cec17	5.12E+02 - (3.01E+00)	**5.11E+02** **(2.81E+00)**	5.12E+02 = (4.82E+00)	5.11E+02 (3.86E+00)	5.13E+02 - (5.58E+00)	**5.08E+02** **(4.57E+00)**	5.08E+02 = (4.54E+00)	5.07E+02 (3.77E+00)
	F25 cec17	4.82E+02 - (4.55E+00)	**4.81E+02** **(3.57E+00)**	4.82E+02 = (6.18E+00)	**4.81E+02** **(2.33E+00)**	4.80E+02 = (1.44E-02)	4.81E+02 (3.52E+00)	4.81E+02 - (2.32E+00)	**4.81E+02** **(3.15E+00)**
	F26 cec17	1.21E+03 - (4.31E+01)	**1.17E+03** **(3.93E+01)**	1.21E+03 = (6.22E+01)	1.19E+03 (5.77E+01)	1.27E+03 - (7.63E+01)	**1.18E+03** **(1.08E+02)**	1.13E+03 = (4.90E+01)	1.12E+03 (5.07E+01)
	F27 cec17	5.43E+02 = (2.15E+01)	5.38E+02 (1.56E+01)	5.36E+02 = (1.67E+01)	**5.31E+02** **(1.78E+01)**	5.33E+02 = (1.56E+01)	5.28E+02 (1.16E+01)	5.14E+02 = (1.01E+01)	5.10E+02 (1.37E+01)
	F28 cec17	4.64E+02 - (1.51E+01)	**4.60E+02** **(5.68E+00)**	4.73E+02 - (2.25E+01)	**4.64E+02** **(1.55E+01)**	4.60E+02 = (6.84E+00)	4.60E+02 (6.84E+00)	4.59E+02 = (3.03E-13)	4.59E+02 (3.32E-13)
	F29 cec17	3.53E+02 = (1.08E+01)	3.57E+02 (1.44E+01)	**3.62E+02** + **(1.91E+01)**	3.84E+02 (1.93E+01)	3.49E+02 = (9.11E+00)	3.49E+02 (1.14E+01)	3.65E+02 = (1.52E+01)	3.65E+02 (1.40E+01)
	F30 cec17	6.68E+05 = (8.12E+04)	6.51E+05 (8.03E+04)	6.68E+05 = (1.02E+05)	6.38E+05 (5.48E+04)	6.50E+05 = (6.32E+04)	6.72E+05 (8.23E+04)	6.08E+05 = (3.03E+04)	6.04E+05 (2.57E+04)
-/=/+		**15/15/0**		**14/14/2**		**13/17/0**		**12/18/0**	

TABLE S13 PERFORMANCE COMPARISONS OF FOUR SCSS-BASED TOP ALGORITHMS WITH THE BASELINES ON 100-D CEC2017 BENCHMARK SET

		L-SHADE	SCSS-L-SHADE	UMOEA-II	SCSS-UMOEA-II	L-SHADE_EpSin	SCSS-L-SHADE_EpSin	jSO	SCSS-jSO
Unimodal Functions	cec17 F1	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)	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)
	cec17 F2	**3.16E-04** + **(5.07E-05)**	3.41E-04 (5.81E-05)	9.66E-05 = (1.75E-05)	9.31E-05 (1.35E-05)	1.58E-04 - (4.22E-05)	**1.38E-04** **(4.25E-05)**	**3.10E-04** + **(5.45E-05)**	3.66E-04 (6.77E-05)
	cec17 F3	**5.47E-06** + **(6.19E-06)**	1.07E-03 (1.73E-03)	**2.84E-06** + **(3.01E-06)**	6.60E-06 (4.57E-06)	5.35E-09 - (1.11E-08)	**2.20E-10** **(1.57E-09)**	**2.71E-06** + **(2.72E-06)**	1.52E-04 (1.69E-04)
Simple Multimodal Functions	cec17 F4	2.01E+02 - (7.69E+00)	**2.00E+02** **(8.00E+00)**	1.87E+02 = (4.03E+01)	1.93E+02 (3.12E+01)	2.04E+02 = (9.79E+00)	2.05E+02 (1.11E+01)	1.94E+02 = (2.35E+01)	1.96E+02 (1.09E+01)
	cec17 F5	3.78E+01 - (7.64E+00)	**2.69E+01** **(6.48E+00)**	3.53E+01 - (7.62E+00)	**2.79E+01** **(7.14E+00)**	6.06E+01 - (7.15E+00)	**4.15E+01** **(6.26E+00)**	4.29E+01 - (7.17E+00)	**2.84E+01** **(5.43E+00)**
	cec17 F6	1.37E-03 - (8.75E-04)	**5.37E-04** **(4.36E-04)**	8.12E-03 - (5.54E-03)	**2.61E-03** **(2.27E-03)**	3.51E-05 - (1.38E-05)	**9.41E-06** **(5.14E-06)**	1.61E-04 - (4.30E-04)	**1.68E-05** **(1.18E-05)**
	cec17 F7	1.51E+02 - (4.80E+00)	**1.38E+02** **(4.48E+00)**	1.41E+02 - (9.72E+00)	**1.36E+02** **(9.40E+00)**	1.67E+02 - (9.13E+00)	**1.45E+02** **(5.70E+00)**	1.41E+02 - (6.94E+00)	**1.27E+02** **(4.53E+00)**
	cec17 F8	3.92E+01 - (5.48E+00)	**2.75E+01** **(5.11E+00)**	3.60E+01 - (7.09E+00)	**2.78E+01** **(7.23E+00)**	5.73E+01 - (9.38E+00)	**3.87E+01** **(6.26E+00)**	4.31E+01 - (5.58E+00)	**2.99E+01** **(5.62E+00)**
	cec17 F9	1.56E-01 - (2.22E-01)	**1.42E-02** **(6.64E-02)**	5.35E-01 - (5.13E-01)	**9.17E-02** **(1.35E-01)**	0.00E+00 = (0.00E+00)	0.00E+00 (0.00E+00)	4.60E-02 - (1.11E-01)	**0.00E+00** **(0.00E+00)**
	cec17 F10	1.14E+04 - (6.11E+02)	**1.05E+04** **(4.67E+02)**	1.19E+04 = (1.25E+03)	1.13E+04 (1.59E+03)	1.05E+04 - (5.15E+02)	**9.57E+03** **(4.63E+02)**	9.71E+03 - (6.59E+02)	**9.23E+03** **(6.08E+02)**
Hybrid Functions	cec17 F11	3.86E+02 - (9.53E+01)	**1.54E+02** **(5.30E+01)**	4.27E+02 - (1.03E+02)	**1.58E+02** **(4.12E+01)**	4.16E+01 = (2.39E+01)	4.26E+01 (2.91E+01)	1.06E+02 - (3.82E+01)	**7.21E+01** **(3.10E+01)**
	cec17 F12	2.37E+04 = (1.05E+04)	2.25E+04 (8.53E+03)	4.52E+03 = (8.56E+02)	4.86E+03 (1.42E+03)	5.28E+03 = (1.39E+03)	**4.62E+03** **(7.33E+02)**	2.05E+04 = (1.06E+04)	**1.41E+04** **(8.02E+03)**
	cec17 F13	1.36E+03 - (8.06E+02)	**2.45E+02** **(7.34E+01)**	3.60E+02 - (1.47E+02)	**1.64E+02** **(4.77E+01)**	7.92E+01 = (2.87E+01)	8.36E+01 (3.44E+01)	1.60E+02 - (4.19E+01)	**1.12E+02** **(2.79E+01)**
	cec17 F14	2.55E+02 - (3.25E+01)	**1.01E+02** **(2.01E+01)**	2.35E+02 - (3.25E+01)	**7.25E+01** **(1.56E+01)**	5.13E+01 = (8.93E+00)	4.86E+01 (6.46E+00)	6.28E+01 - (1.18E+01)	**3.95E+01** **(4.08E+00)**
	cec17 F15	2.50E+02 = (4.87E+01)	2.59E+02 (4.34E+01)	2.67E+02 - (5.38E+01)	**2.21E+02** **(4.82E+01)**	7.28E+01 = (3.14E+01)	7.73E+01 (2.83E+01)	1.64E+02 - (4.20E+01)	**9.73E+01** **(3.56E+01)**
	cec17 F16	1.79E+03 - (2.58E+02)	**1.55E+03** **(2.39E+02)**	1.67E+03 = (4.55E+02)	1.64E+03 (4.27E+02)	1.55E+03 - (2.51E+02)	**1.31E+03** **(2.61E+02)**	1.84E+03 = (3.15E+02)	1.74E+03 (2.99E+02)
	cec17 F17	1.20E+03 - (2.21E+02)	**1.04E+03** **(2.00E+02)**	1.36E+03 = (3.13E+02)	1.28E+03 (2.62E+02)	1.16E+03 - (1.72E+02)	**9.23E+02** **(1.76E+02)**	1.26E+03 - (2.63E+02)	**1.13E+03** **(2.20E+02)**
	cec17 F18	2.15E+02 - (4.60E+01)	2.11E+02 (5.33E+01)	2.35E+02 = (6.29E+01)	2.16E+02 (4.72E+01)	7.92E+01 = (2.19E+01)	7.59E+01 (1.83E+01)	1.76E+02 - (4.05E+01)	**1.11E+02** **(3.07E+01)**
	cec17 F19	1.77E+02 - (2.31E+01)	**1.63E+02** **(2.46E+01)**	1.76E+02 - (2.65E+01)	**1.52E+02** **(2.50E+01)**	5.22E+01 = (6.65E+00)	5.09E+01 (5.78E+00)	1.07E+02 - (2.14E+01)	**5.22E+01** **(5.72E+00)**
	cec17 F20	1.57E+03 - (2.42E+02)	**1.50E+03** **(1.79E+02)**	1.93E+03 = (3.61E+02)	1.89E+03 (3.11E+02)	1.44E+03 - (1.96E+02)	**1.23E+03** **(1.89E+02)**	1.38E+03 = (2.44E+02)	1.29E+03 (2.12E+02)
Composition Functions	cec17 F21	2.69E+02 - (5.81E+00)	**2.59E+02** **(4.38E+00)**	2.56E+02 = (6.84E+00)	2.55E+02 (6.49E+00)	2.83E+02 - (1.41E+01)	**2.64E+02** **(5.61E+00)**	2.64E+02 - (6.56E+00)	**2.49E+02** **(5.18E+00)**
	cec17 F22	1.19E+04 - (5.24E+02)	**1.12E+04** **(6.26E+02)**	1.27E+04 = (1.81E+03)	1.25E+04 (1.61E+03)	1.08E+04 - (5.90E+02)	**9.54E+03** **(5.05E+02)**	1.07E+04 - (6.27E+02)	**1.01E+04** **(6.70E+02)**
	cec17 F23	5.68E+02 = (7.98E+00)	5.67E+02 (7.15E+00)	5.70E+02 = (9.40E+00)	5.70E+02 (1.34E+01)	5.98E+02 - (7.21E+00)	**5.92E+02** **(6.32E+00)**	5.69E+02 = (1.37E+01)	5.67E+02 (1.14E+01)
	cec17 F24	9.19E+02 - (8.98E+00)	**9.12E+02** **(8.61E+00)**	9.22E+02 - (8.89E+00)	**9.16E+02** **(1.16E+01)**	9.37E+02 - (2.15E+01)	**9.08E+02** **(8.10E+00)**	9.01E+02 - (1.04E+01)	**8.96E+02** **(7.84E+00)**
	cec17 F25	7.46E+02 = (3.47E+01)	7.44E+02 (3.50E+01)	7.49E+02 - (2.76E+01)	**7.29E+02** **(3.77E+01)**	6.93E+02 = (4.53E+01)	6.89E+02 (4.55E+01)	7.18E+02 = (3.87E+01)	7.13E+02 (4.26E+01)
	cec17 F26	3.41E+03 - (1.02E+02)	**3.31E+03** **(9.92E+01)**	3.42E+03 - (9.37E+01)	**3.32E+03** **(9.49E+01)**	3.24E+03 - (2.51E+02)	**3.06E+03** **(9.06E+01)**	3.20E+03 - (8.46E+01)	**3.12E+03** **(9.03E+01)**
	cec17 F27	6.58E+02 - (1.38E+01)	**6.47E+02** **(1.57E+01)**	6.41E+02 - (1.79E+01)	**6.32E+02** **(1.61E+01)**	5.92E+02 = (1.37E+01)	5.90E+02 (1.81E+01)	5.86E+02 - (2.05E+01)	**5.77E+02** **(2.28E+01)**
	cec17 F28	5.28E+02 = (2.19E+01)	5.34E+02 (2.30E+01)	**5.18E+02** + **(3.80E+01)**	5.28E+02 (3.07E+01)	5.15E+02 = (1.95E+01)	5.22E+02 (2.30E+01)	5.29E+02 = (2.78E+01)	5.25E+02 (2.86E+01)
	cec17 F29	1.53E+03 - (1.92E+02)	1.48E+03 (1.83E+02)	1.40E+03 = (2.46E+02)	1.48E+03 (2.33E+02)	1.23E+03 = (1.62E+02)	1.21E+03 (1.42E+02)	1.33E+03 - (2.02E+02)	**1.25E+03** **(1.82E+02)**
	cec17 F30	2.43E+03 - (1.45E+02)	**2.34E+03** **(1.32E+02)**	2.36E+03 = (1.26E+02)	2.36E+03 (1.53E+02)	2.34E+03 = (1.35E+02)	2.37E+03 (1.92E+02)	2.31E+03 = (1.23E+02)	2.27E+03 (1.06E+02)
-/=/+		**20/8/2**		**14/14/2**		**16/14/0**		**20/8/2**	