IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 11, 2020, accepted October 28, 2020, date of publication November 6, 2020, date of current version November 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3036438

Contribution Based Co-Evolutionary Algorithm for Large-Scale Optimization Problems

MOHAMED A. MESELHI, SABER M. ELSAYED, (Member, IEEE), RUHUL A. SARKER, (Member, IEEE), AND DARYL L. ESSAM, (Member, IEEE)
School of Engineering and Information Technology, University of New South Wales at Canberra, Canberra, ACT 2612, Australia
Corresponding author: Mohamed Atef Meselhi (m.radwan@unsw.edu.au)

This work was supported by the Australian Research Council Discovery Project under Grant DP190102637.

ABSTRACT The solution of large-scale optimization problems is the key to many decision-making processes in practice. However, it is a challenging research topic when considered both the quality of solutions and the required computational time. One of the popular approaches for these problems is to divide the problems into a number of smaller sub-problems, that are then solved separately with an exchange of some information using the cooperative co-evolution (CC) concept. However, the characteristics of sub-components could be different, and their contributions to the overall performance can also be different while solving the problem. In the CC approach, it usually applies one optimizer and allocates equal computational budget to all sub-components. In this article, a new algorithm is proposed with the use of multiple optimizers, along with a need-based allocation of computational budget for the sub-components. In the proposed algorithm, a group of optimizers cooperate in an effective way to evolve the sub-components, depending on heuristic fuzzy rules. The performance of our proposed algorithm was evaluated by solving a number of large-scale global optimization benchmark functions. The empirical results show that the proposed algorithm outperforms equal allocation CC, a single selection characteristic, a single candidate optimizer and state-of-the-art algorithms.

INDEX TERMS Cooperative co-evolution, large-scale optimization, fuzzy logic.

I. INTRODUCTION

Optimization algorithms, such as evolutionary algorithms (EAs) [1] and swarm intelligence (SI) [2], have emerged as effective methods for solving a wide variety of optimization problems, such as single objective or multi-objective problems, with discrete and/or continuous variables, in different fields including, but not limited to, engineering and science [3], [4]. However, the performance of these algorithms significantly deteriorates with growing numbers of decision variables [5].

As of the literature, the cooperative co-evolution (CC) approach [6] is a popular choice for solving large-scale optimization problems [7], [8]. The CC method applies a divide and conquer strategy, firstly to decompose the high-dimensional problems into a number of smaller sub-problems and to then solve them individually by exchanging some information among the sub-problems during the search process. Liu *et al.* [9] have demonstrated that CC approaches are highly effective in dealing with high dimensional continuous problems. For practical cases, problem decomposition is very important, as the sub-problems are usually interdependent because they have some common variables in two or more sub-problems. The performance of the CC algorithms is highly dependent on the number of common variables and the complexity of the sub-problems. So it is expected that a problem must be decomposed in a way that minimizes the number of common variables in the sub-problems and maximizes the number of highly related variables in each sub-problem. In other words, the inter-dependencies between sub-problems should be as minimum as possible. There exist several decomposition methods, such as static [9], random [10] and variable interaction grouping [11], that have dealt with grouping of decision variables in the global optimization domain.

It is worth mentioning that a suitable grouping results in a significant improvement in the quality of solutions when an

The associate editor coordinating the review of this manuscript and approving it for publication was Huaqing Li.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

appropriate search approach is applied. Interestingly, some sub-problems may make higher contributions than others during the search process [12]. Therefore, the round-robin mechanism of CC [6], which gives equal consideration to all sub-problems, despite their different contributions, might waste a considerable amount of computational resources due to inappropriate resource allocation. Here, the share of higher resources is given to sub-problems that contribute more, and so this could provide a significant advantage [13].

In the literature, CC approaches usually use a single optimizer repeatedly for all sub-problems. However, as the characteristics of sub-problems may vary significantly due to the problem's structure and decomposition, it is unlikely that a single optimizer will perform the best for all sub-problems. It is noted that different optimizers, such as differential evolution (DE) [14], covariance matrix adaptation ES (CMA-ES) [15] and particle swarm optimization (PSO) [2], are suitable for problems with a variety of properties in their sub-problems. DE is considered useful for problems in which feasible regions are parallel to the axes, although it has a weakness of how it handles local optima when solving multi-modal functions [16]. CMA-ES provides an effective way of dealing with uni-modal problems, but is also weak in handling local optima in multi-modal functions [17]. PSO has demonstrated high convergence ratios in the first phases of the optimization process, but in the refinement stage, its performance is slow and it may become trapped in local optima [16]. Hybridization [18]–[20] appears to be a complementary framework which tries to capture and merge the strengths of each algorithm to achieve superior results. For large-scale problems, divided into a number of smaller sub-problems, the use of multiple optimizers with complementary abilities could offer significant advantages, where an optimizer can be selected for each sub-problem based on its rate of progress in the search process.

Motivated by the above two aspects, in this article, a performance-based computational budget allocation among the sub-problems, with appropriate use of multiple optimizers, called fuzzy contribution-based cooperative co-evolution (F3C), is proposed as an effective alternative to a round-robin strategy with a single optimizer. In F3C, a group of optimizers cooperate in an effective way to evolve its sub-components based on heuristic fuzzy rules. This heuristic emphasizes on the most effective sub-component, with its optimizer using two complementary criteria, namely, fitness improvement and population diversity. Its performance is evaluated by solving a good number of large-scale global optimization (LSGO) benchmark functions. The empirical results show that it outperforms CC with equal allocation, single selection characteristic, single candidate optimizer and state-of-the-art algorithms.

The remainder of this article is structured as follows: Section II highlights the related literature; Section III describes the proposed algorithm; Section IV presents the experimental results; and finally, Section V concludes the paper with directions for future work.

II. RELATED LITERATURE

This section provides a brief review about the CC framework, existing contribution-based techniques and an overview of fuzzy theory.

A. COOPERATIVE CO-EVOLUTION

To mitigate the curse of dimensionality that affects the performance of EAs, two basic approaches have been adopted. The first tries to improve those of classic EAs by embedding in them some features, such as intelligent sampling [21], advanced initialization [22], parallelization [23], adaptation of operators and/or parameters [24], a local search mechanism [25] and hybridization [26]. The other aims to apply a divide-and-conquer strategy to divide a large-scale problem into smaller sub-problems (also called sub-components) and to then solve each of them latter as an independent problem, which is called the CC framework.

The standard CC framework has two phases, that is decomposition and optimization phases. In the first, a divide-and-conquer mechanism is used to decompose the problem into smaller ones. In the second, it subsequently solves them in a cooperative manner [6], [27].

As of the literature, several decomposition techniques have been suggested to divide the high dimensional problem into many smaller sub-components. This list of these techniques include CC with variable interaction learning [11], random grouping (RG) [28], differential grouping (DG) [29], global DG (GDG) [30], extended DG (XDG) [31], improved DG (DG2) [32], fast inter-dependency identification [33], enhanced DG (EDG) [34] and recursive DG (RDG) [35].

As previously mentioned, in the optimization stage, each sub-problem is regarded as an independent problem that is optimized iteratively for a predetermined number of iterations in a round-robin fashion. Then, the obtained solution is used to update a context vector (a complete solution consisting of the best solution of each sub-problem) at each iteration [36]. It is worth mentioning that the classic CC algorithm [6] optimizes sub-problems iteratively in a round-robin fashion until the stopping condition is satisfied. This means that all the sub-problems have equivalent computational resources, which negatively affects the performance of EAs when solving large-scale optimization problems. In other words, a considerable amount of computational budget is wasted on solving an optimization problem with different contributions' sub-components.

B. CONTRIBUTION-BASED COOPERATIVE CO-EVOLUTION

To alleviate the above-mentioned drawback, several studies proposed methodologies that give more consideration to sub-problems with higher contributions to the overall objective value [13], [37]–[39]. These methods are known as contribution-based CC (CBCC).

Omidvar *et al.* [12] proposed a CBCC method for assigning the available resources among the sub-components, by calculating the total improvement each makes towards the

main optimization problem. It depends on the contribution estimated in the testing phase and has two versions, namely, CBCC1 and CBCC2. The former optimizes a selected sub-component for only one iteration, whereas the latter optimizes it for as long as it continues to improve the overall fitness and then the algorithm applies the testing phase again to select another sub-component.

However, the CBCC1 and CBCC2 algorithms suffer from over-exploratory and over-exploitative problems, respectively [40]. They focus on only the sub-problems that have high initial contributions to the overall fitness value in the first cycle, but do not have fast responses to changes in their contributions later. Therefore, the information accumulated from the first cycle, limits their selection strategies to sub-problems with high initial effects on the overall fitness value. To lessen these issues, the CBCC3 algorithm, which is an enhanced version of the CBCC1 and CBCC2 ones, was proposed. Its policy of resource allocation, concentrates on optimizing only the most recent sub-problem, which contributes more to enhancing the overall objective value in the exploitation stage. According to [40], the CBCC3 version is superior to the traditional CC, CBCC1 and CBCC2 algorithms, for dealing with unbalanced contribution problems. However, its major drawback is that it depends on the magnitude of the contributions of the sub-components in the selection process.

In another approach, called CC with adaptive optimizer iterations (CCAOI), introduced in [41], the existing computational budget assigned to several sub-problems is dynamically adapted. It computes the number of iterations each sub-problem has to carry out in each cycle using an indicator, δi, (the contribution factor of the sub-problem), which should be normalized with the generation number performed by the optimizer.

With a similar aim, the CC framework (CCFR) approach was proposed in [42]. It has two aspects which distinguish it from the abovementioned methods: a sub-problem is excluded if it does not contribute to improving the overall objective value; and the contribution of each sub-problem is updated dynamically in each cycle, with the one with the highest contribution chosen to complete the evolutionary stage. This means that this method evaluates the recent contribution of each sub-problem, based on the mean value of its contributions in the last two generations.

A Bandit-based CC (BBCC) framework [43] has been proposed to formulate the resource allocation problem as a dynamic multi-armed bandit one. It employs efficient bandit algorithms that seek to learn the contribution done by each sub-component to the fitness improvement value and allocating the appropriate resources respectively.

Ren *et al.* [44] proposed a fine-grained resource allocation strategy, called FCRACC, which allocates resources based on both the evolution characteristics of CC and the optimal solution of a mathematical model for its resource allocation. FCRACC allocates more computational resources to the subproblem, which is most expected to make the highest contribution to the fitness improvement at the next iteration.

Recently, an improved version of CCFR, called CCFR2, was proposed in [45], which correlates the subpopulation size with the size of subproblems and considers unequal-sized subpopulations in when calculating contributions. Unlike CCFR, it saves the computational resources required to both obtain the best overall solution before the start of co-evolution and evaluating population during the co-evolution, if the current best solution is the same with the last best one.

C. FUZZY THEORY

Zadeh [46] presented the field of fuzzy theory in the mid-60s and now fuzziness is fast becoming a fundamental of our daily lives. In this research, he clarified that ''fuzzy sets'' are sets with boundaries that are not obvious, and they are used mainly to represent uncertainty [47]. In general, a fuzzy set (S) is in X, where X is the universe of discourse with its elements denoted by x; for instance, a fuzzy set ($S = \{x_1, x_2, x_3, x_4\}$) in X, is characterized by a membership function marked by $\mu(x)$ that maps each point (x) to a real value in a [0,1] interval. The value of the membership function ($\mu_S(x)$) represents the degree of the membership of x in S and the mapping is restricted to only $\mu_S(x) \in [0, 1]$. The membership value of an element, specifies the degree to which it belongs to its fuzzy set, and when $\mu_S(x)$ is close to 1, x clearly belongs to S. There are various types of membership functions, including triangular, Gaussian, trapezoidal and the generalized bell curve.

Since fuzzy language is regarded as a natural language for handling ambiguous and uncertain expressions, fuzzy logic has been exploited for word-based computing [48] and can be considered the process of transition from total to partial truth. In other words, the value of a Boolean variable (i.e., 1 or 0) is changed to a linguistic weighting variable which takes values such as ''medium'', ''very low'', ''high'', etc. Therefore, fuzzy logic represents a human's reasoning mechanism that seeks to translate all truths as approximate, with a false value represented as a partial truth. This mechanism seeks to convert the input data to output by using a formula of ''IF-THEN'' rules, which the fuzzy system then maps to their mathematical equivalents.

III. PROPOSED APPROACH

The F3C approach adopts heuristic fuzzy rules that place more emphasis on better-performing optimizers for solving sub-problems, based on their contributions to improving the overall objective value and maintaining diversity within the population.

Similar to the typical CC framework, it starts by decomposing a large-scale problem into several smaller sub-problems using an appropriate decomposition method. This is followed by an optimization stage in which the sub-components generated are optimized for a certain number of generations.

Throughout the optimization process, fuzzy rules are used to measure the effectiveness of a sub-component-optimizer ($C_i\, O_j$) pair based on two core characteristics, namely, fitness improvement and population diversity, which can be

easily expressed as linguistic variables. During the first co-evolutionary cycle, all the sub-components formed in the decomposition stage are optimized in a round-robin fashion using each member in the optimizer' group (O_j) to measure the initial effectiveness of each (C_i O_j) pair. Firstly, the contribution of each sub-component (C_i) is calculated as the relative fitness improvement (FI) before (i.e., f_b) and after (i.e., f_a) using O_j by:

$$FI_{C_i,O_j} = \frac{(f_b - f_a)}{f_b}, \quad \forall\, i = 1, 2, \ldots, k \;\; and \;\; j = 1, 2, \ldots, m, \tag{1}$$

where k is the number of sub-components generated and m the number of candidate optimizers used.

Then, the diversity of the population is calculated as the average distance of each individual in x_i to the best solution among them as:

$$D_{C_i,O_j} = \frac{\sum_{i=1}^{NP} dis(\vec{x}_i, \vec{x}_b)}{NP} \tag{2}$$

where $dis(\vec{x}_z, \vec{x}_b)$ is the Euclidean distance between the zth individual and the best individual in x, NP the number of individuals and $\vec{x}_b$ the best solution among the individuals.

Then, in each generation, to determine which (C_i O_j) pair will use the computational resources, the inputs (diversity and fitness improvement) are fuzzified so that the effectiveness can be determined and a conclusion drawn based on the application of the rules on the fuzzy sets generated. This process involves the following three steps.

1) Fuzzification establishes to what extent the numeric values of both the fitness improvement and population diversity belong to each of the fuzzy sets (S_i) using a membership function. Although the proposed F3C algorithm could employ different types of membership functions, the Gaussian function is considered in this study for the simplicity in its design that requires only two parameters [49], as shown in Fig. 1 and defined as:

$$\mu_{S_i}(x) = e^{-(x-m_i)^2/2\sigma_i^2} \tag{3}$$

where m_i and σ_i are the mean and standard deviation of the ith fuzzy set (S_i), respectively.
In this study, as suggested in [50], five linguistic levels, namely, "Poor (P)", "Fair (F)", "Good (G)", "Very Good (VG)" and "Excellent (E)" are used to determine which C_i O_j pair is the most effective, as shown in Fig. 2. The parameters, including the mean and the standard deviation, of these membership functions, are given in Table 1.
2) An inference mechanism that formulates human thinking by mapping the inputs to the outputs is applied. It supports the decision-making process by creating fuzzy inferences for antecedents and IF-THEN rules. It also applies fuzzy operators on the antecedents and obtains the overall consequent using the aggregation of fuzzy rules. The IF-THEN rules are based on the Mamdani fuzzy interference model [51], with a typical one expressed as:

$$Rule_i: \text{IF } FI \text{ is } A_1 \text{ AND } D \text{ is } B_2 \text{ THEN } EFF \text{ is } C_3 \tag{4}$$

where i indicates the rule number ($i = 1, 2, \ldots, n$), n is the number of rules ($n = 25$) and A, B *and* C represent the fuzzy sets for the inputs and outputs.

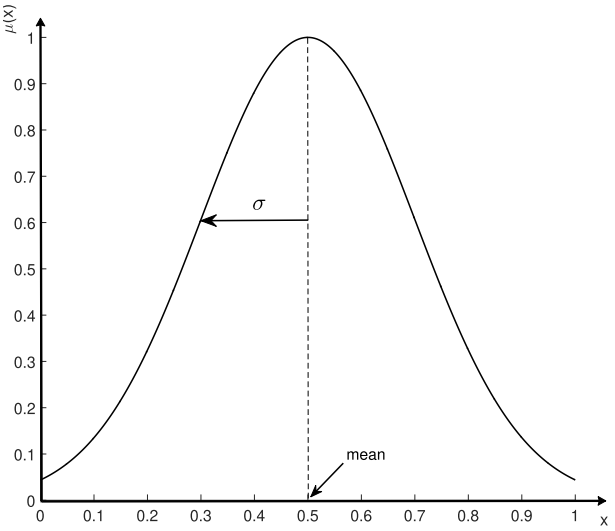


FIGURE 1. Gaussian membership function.

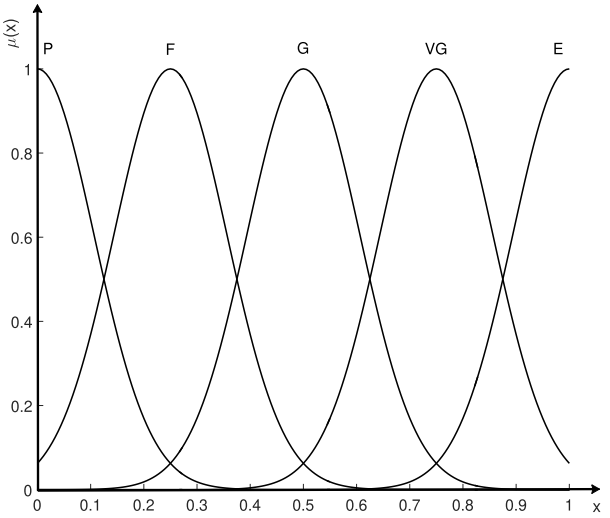


FIGURE 2. Fuzzy membership functions of inputs or output.

TABLE 1. Parameters of Membership Functions for Fitness Improvement, Diversity and Effectiveness.

Membership function	Fitness improvement		Diversity		Effectiveness	
	mean	σ	mean	σ	mean	σ
Poor (P)	0	0.1062	0	0.1062	0	0.1062
Fair (F)	0.25	0.1062	0.25	0.1062	0.25	0.1062
Good (G)	0.5	0.1062	0.5	0.1062	0.5	0.1062
Very Good (VG)	0.75	0.1062	0.75	0.1062	0.75	0.1062
Excellent (E)	1	0.1062	1	0.1062	1	0.2038

TABLE 2. IF-THEN Rules.

AND		Fitness improvement				
		P	F	G	VG	E
Diversity	P	P	P	F	G	VG
	F	P	F	G	G	VG
	G	F	G	VG	VG	E
	VG	G	G	VG	E	E
	E	VG	VG	E	E	E

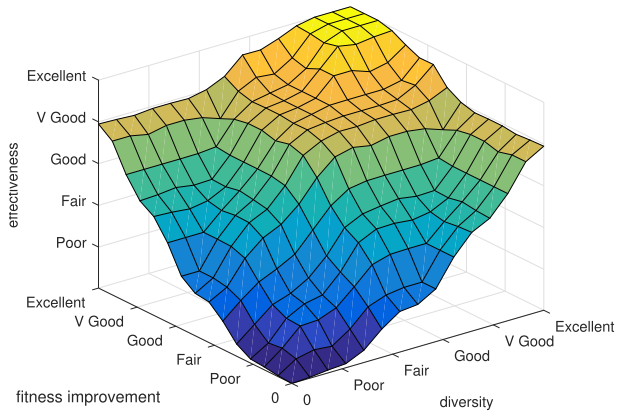


FIGURE 3. Fuzzy heuristic output surface.

The IF-THEN rules, which formulate the consequent based on the two antecedents, are shown in Table 2, in which the logical AND connective represents the intersection between two fuzzy sets denoted by the minimum membership value of the antecedents as:

$$A \quad \textbf{AND} \quad B = A \cap B = \min(\mu_A(x), \mu_B(x)) \tag{5}$$

3) A defuzzification step is applied to convert the fuzzy output from the inference mechanism to its original crisp value using the center-of-gravity method, which is considered the most prevalent. It locates the center of an area under the surface of the membership function which is expressed as:

$$y^{\star} = \frac{\int_a^b \mu(y) \cdot y \, dy}{\int_a^b \mu(y) \, dy} \tag{6}$$

where $y^{\star}$ and y are the crisp and fuzzy outputs, respectively. Fig. 3 shows the decision surface of the effectiveness obtained by the fuzzy system which is based on the different values of the two inputs, fitness improvement and diversity. It is clear that, when both these values are increased, so does the output value of the effectiveness.

Based on the mentioned points, Algorithm 1 summarizes the steps in the proposed F3C algorithm. Firstly, an initial population (P) of size NP is randomly generated ($P = \{\vec{X}_1, \vec{X}_2 \ldots \vec{X}_{NP}\}$) and the best individual ($\vec{X}_{best}$) and its fitness value ($f(\vec{X}_{best})$) recorded to build the context vector (lines 1 to 3). Then, in the decomposition stage (line 4), all the sub-components are constructed based on the interactions among the decision variables using a *decomposition* function. Before the optimization stage begins, the control parameters of the candidate optimizers are set as suggested in the literature (line 5) and then, during the optimization phase, any group of candidate optimizers can be applied. Also, in lines 6 to 8, the F3C algorithm initializes all the elements of the FI, D and EFF matrices to 0, 0 and 1, respectively.

First, each sub-component is evolved with each optimizer for a pre-defined cycle (i.e., number of fitness evaluations), with FI as well as D recorded for each pair. Subsequently, the fuzzy-based heuristic takes place with FI and D considered as input. The numeric values of these inputs are assigned to their appropriate fuzzy sets (i.e., "P", "F", "G", "VG" and "E") and then, both IF-THEN rules and fuzzy operators are applied on them to obtain the effectiveness of each pair. Finally, this fuzzy output is converted to its original crisp value ($EFF_t \in [0, 1], \ \forall t = 1, 2, \ldots, (k * m)$). Once this step is carried out, the pair with the largest effectiveness is selected as:

$$[C_i, \ O_j] = max(EFF[1, \ 2, \ \ldots, (k * m)]), \tag{7}$$

and the optimization process places emphasis on this pair for the subsequent cycle. This process (lines 10 to 22) is repeated until all the available computational resources are consumed.

Algorithm 1 FBCC

1: generate initial population P randomly;
2: $[V_1, \ldots, V_{NP}] \leftarrow evaluate(f(\vec{X}_i), \forall i = 1, \ldots, NP$;
3: $(\vec{X}_{best}, f_{best}) \leftarrow P(min([V_1, \ldots, V_{NP}]))$;
4: $G = \{g_1, \ldots, g_k\} \leftarrow decompose(f(x), dim)$;
5: Initialization of m candidate optimization algorithms;
6: $FI_t \leftarrow 0, \forall t = 1, \ldots, k * m$;
7: $D_t \leftarrow 0, \forall t = 1, \ldots, k * m$;
8: $EFF_t \leftarrow 1, \forall t = 1, \ldots, k * m$;
9: $cy \leftarrow 0$;
10: **while** $FEs < FEs_{max}$ **do**
11: $cy \leftarrow cy + 1$;
12: $Pair\_idx = max(EFF[1, \ 2, \ \ldots, k * m])$;
13: **for** $j = 1 : length$(Pair_idx) **do**
14: $[C\_selected, O\_selected] \leftarrow Pair\_idx$;
15: Evaluate sub-problem $C\_selected$ within the context vector using the candidate optimizer $O\_selected$;
16: Measure the fitness improvement FI based on Equation (1);
17: Measure the Diversity D based on Equation (2);
18: update the effectiveness EFF;
19: update the context vector;
20: $FEs = FEs + used\_FEs$;
21: **end for**
22: **end while**

IV. EXPERIMENTAL STUDY

In this section, the numerical experiments conducted to measure the effectiveness of the proposed F3C strategy, which uses 20 test functions from the CEC'2010 benchmark problems for LSGO [52], are presented and analyzed. Also, its

effectiveness for solving 15 other large-scale problems taken from the CEC'2013 benchmark problems [53] is evaluated.

In the decomposition phase, the enhanced differential grouping (EDG) method [34] is used to automatically divide the high-dimensional problems into low-dimensional sub-problems. In the first stage, this method is capable of effectively detecting both separable and nonseparable components. Then, a further examination is conducted on the nonseparable ones to identify their direct and indirect interdependencies, to combine them in the same subproblem. In the optimization phase, variants of three powerful optimization algorithms, 1) SaNSDE [54], 2) CMA-ES [17] and 3) SLPSO [55], are considered.

The experimental results obtained for each benchmark test problem are based on 25 independent runs, with both the means and standard deviations of the best solutions calculated. The population size is set as suggested in the corresponding papers; that is, SaNSDE used a population size of 100, SLPSO used $100 + D/10$ solutions, while CMA-ES used a population size equal to $4+(3*\ln(D))$. The maximum number of fitness evaluations (FEs), divided between the grouping and optimizing stages, is 3×10^6, and the cycle size is 10^4, as suggested in [52] and [38], respectively, with the group of candidate optimizers using their recommended original settings.

In this article, the two types of non-parametric statistical hypothesis tests used, are the Wilcoxon signed rank test [56] and Friedman ranking test [57]. The former is conducted to verify any significant differences between algorithms. Using a 5% significance level, if the p-value is less than or equal to 5%, the null hypothesis is rejected, otherwise it is accepted. To compare any two algorithms, one of three signs ($+$,$-$, and $\approx$) is used. $+$ designates that the first algorithm has more significant performance than the second, '$-$', the opposite (i.e., the second algorithm outperforms the first) and $\approx$, that there is no significant difference between the two algorithms. If the first algorithm performs better than the second, the Wilcoxon test adopts $R+$ which represents the sum of the ranks for those functions, otherwise this test shows those ranks with $R-$, while the Friedman test ranks all the algorithms according to their average fitness values.

A. BEHAVIOR OF F3C

In this section, an analysis of the F3C's behavior is presented, with the numbers of FEs the sub-components consume on some selected functions displayed in Fig. 4. In each sub-figure, the x-axis is the sub-component's index and the y-axis the FEs it consumes. It is obvious that there is an uneven allocation of the computational resources, as the F3C optimizes certain sub-components more than the remaining sub-problems.

As shown in Fig. 4(f), the decision variables of f_9 are grouped into 11 sub-components, 10 of which are non-separable with 50 decision variables and one is separable with 500. According to F3C, most of the computational budget is consumed by the first sub-component, the separable one. This indicates that the higher-dimensionality sub-components often provide higher contributions to enhancing the overall fitness value, than the lower ones.

TABLE 3. Results Obtained From Proposed Algorithm Using F3C and Equal-Allocation CC for CEC'2010 Benchmark Problems.

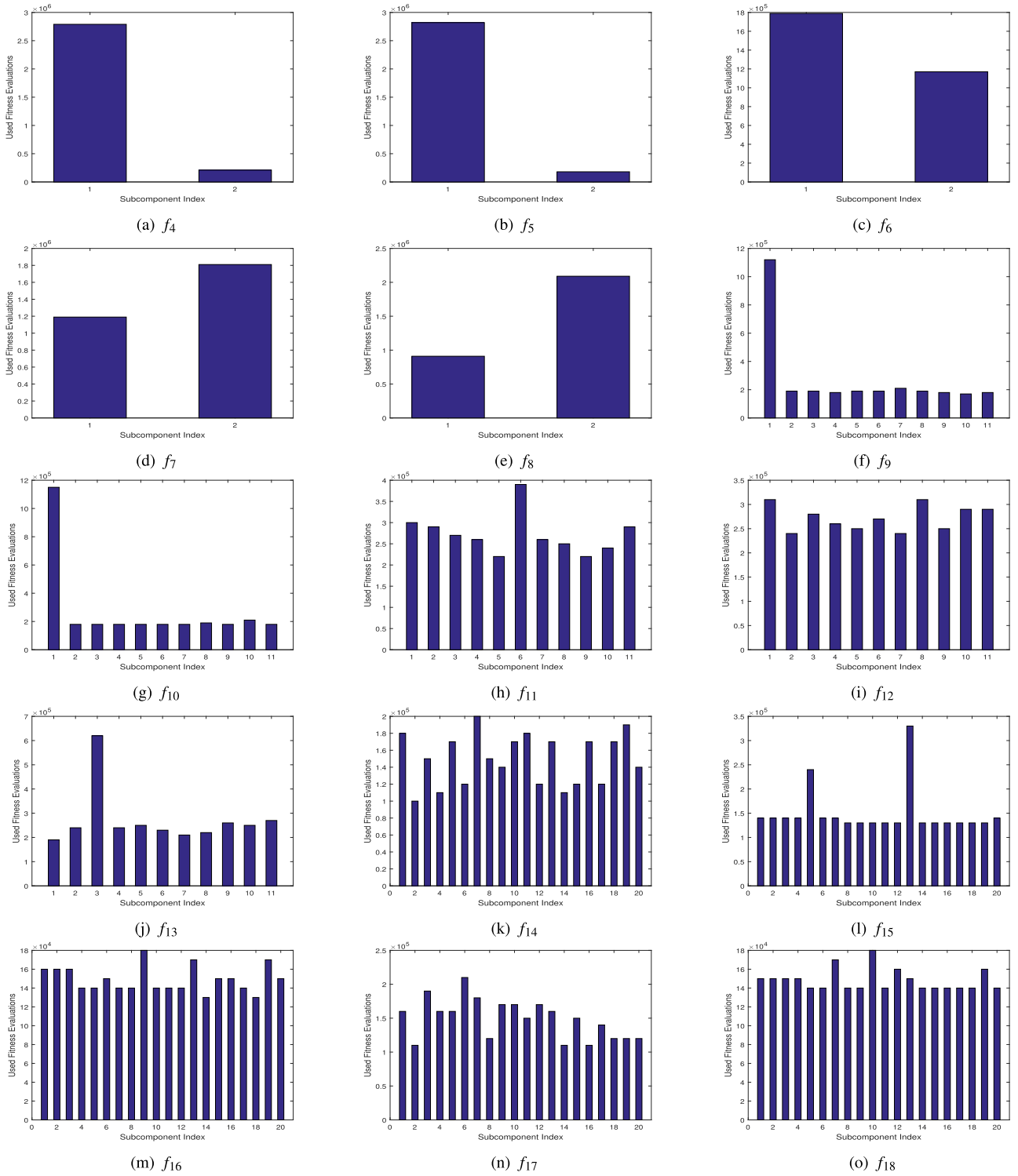
Function	F3C	Equal
	Mean ± (Std.)	Mean ± (Std.)
f_1	**1.96E-08** ± (9.75E-08)	2.80E+03 ± (7.92E+03)
f_2	2.72E+03 ± (6.36E+02)	**2.37E+03** ± (1.33E+02)
f_3	**9.71E-01** ± (4.51E-01)	1.04E+00 ± (3.23E-01)
f_4	**1.99E-04** ± (7.38E-04)	2.24E+05 ± (2.28E+05)
f_5	9.57E+07 ± (1.48E+07)	**7.40E+07** ± (1.99E+07)
f_6	**7.97E-01** ± (5.17E-01)	9.45E-01 ± (4.36E-01)
f_7	**3.14E-19** ± (1.66E-19)	1.22E-18 ± (1.57E-19)
f_8	8.25E-17 ± (2.85E-16)	**2.84E-17** ± (5.07E-18)
f_9	**3.93E-05** ± (1.39E-04)	6.06E+05 ± (2.04E+05)
f_{10}	**2.31E+03** ± (6.19E+02)	2.84E+03 ± (1.28E+02)
f_{11}	**4.32E-02** ± (3.48E-01)	1.54E-01 ± (3.61E-01)
f_{12}	**6.09E-22** ± (1.21E-22)	9.68E-16 ± (2.69E-16)
f_{13}	**1.37E+01** ± (2.42E+01)	9.31E+01 ± (4.73E+01)
f_{14}	**6.97E+03** ± (1.83E+04)	3.81E+05 ± (4.91E+04)
f_{15}	**1.90E+03** ± (1.03E+02)	1.96E+03 ± (1.03E+02)
f_{16}	**1.63E-12** ± (7.86E-14)	9.85E-11 ± (4.83E-10)
f_{17}	**2.03E-23** ± (2.19E-24)	2.19E-22 ± (6.07E-23)
f_{18}	**5.59E+02** ± (5.74E+02)	6.50E+02 ± (7.98E+01)
f_{19}	**4.03E+05** ± (5.01E+05)	5.66E+05 ± (4.05E+04)
f_{20}	**8.03E+02** ± (7.21E+01)	9.48E+02 ± (6.30E+00)

TABLE 4. Results Obtained From Different Characteristics for CEC'2010 Benchmark Problems.

Fun	Fitness Improvement	Diversity	Both
	Mean ± (Std.)	Mean ± (Std.)	Mean ± (Std.)
f_1	7.11E-02 ± (2.65E-01)	**2.35E-09** ± (1.10E-08)	1.96E-08 ± (9.75E-08)
f_2	2.75E+03 ± (7.02E+02)	**2.55E+03** ± (4.86E+02)	2.72E+03 ± (6.36E+02)
f_3	1.03E+00 ± (3.28E-01)	1.02E+00 ± (4.07E-01)	**9.71E-01** ± (4.51E-01)
f_4	2.62E-02 ± (1.26E-01)	1.45E+00 ± (6.57E+00)	**1.99E-04** ± (7.38E-04)
f_5	9.87E+07 ± (1.62E+07)	9.87E+07 ± (1.62E+07)	**9.57E+07** ± (1.48E+07)
f_6	9.64E-01 ± (3.75E-01)	8.44E-01 ± (4.95E-01)	**7.97E-01** ± (5.17E-01)
f_7	1.17E-18 ± (1.27E-19)	1.20E-18 ± (1.62E-19)	**3.14E-19** ± (1.66E-19)
f_8	**2.68E-17** ± (9.31E-18)	2.91E-17 ± (9.97E-18)	8.25E-17 ± (2.85E-16)
f_9	1.49E-04 ± (3.35E-04)	1.15E+03 ± (3.97E+03)	**3.93E-05** ± (1.39E-04)
f_{10}	2.82E+03 ± (1.49E+02)	**2.01E+03** ± (5.72E+02)	2.31E+03 ± (6.19E+02)
f_{11}	1.18E-01 ± (3.30E-01)	7.88E-02 ± (2.73E-01)	**4.32E-02** ± (3.48E-01)
f_{12}	6.26E-22 ± (9.20E-23)	6.41E-22 ± (1.22E-22)	**6.09E-22** ± (1.21E-22)
f_{13}	**1.01E+01** ± (1.16E+01)	3.29E+01 ± (4.76E+01)	1.37E+01 ± (2.42E+01)
f_{14}	9.54E+03 ± (3.92E+03)	9.26E+05 ± (1.81E+06)	**6.97E+03** ± (1.83E+04)
f_{15}	1.92E+03 ± (7.81E+01)	1.92E+03 ± (7.81E+01)	**1.90E+03** ± (1.03E+02)
f_{16}	**1.55E-12** ± (6.47E-14)	1.67E-12 ± (9.47E-14)	1.63E-12 ± (7.86E-14)
f_{17}	2.62E-23 ± (1.29E-23)	2.41E-23 ± (5.50E-24)	**2.03E-23** ± (2.19E-24)
f_{18}	**5.58E+02** ± (9.54E+01)	5.63E+02 ± (1.01E+02)	5.59E+02 ± (5.74E+02)
f_{19}	4.29E+05 ± (5.32E+05)	**3.06E+05** ± (4.95E+05)	4.03E+05 ± (5.01E+05)
f_{20}	8.17E+02 ± (5.85E+01)	8.19E+02 ± (5.81E+01)	**8.03E+02** ± (7.21E+01)

TABLE 5. Results Obtained From Wilcoxon Signed Rank Test of the Proposed Algorithm Using Different Selection Characteristics for CEC'2010 Benchmark Problems.

Algorithms	Better	Equal	Worse	$R+$	$R-$	p-value	Decision
Both criteria VS FI	16	0	5	176	34	**0.008**	$+$
Both criteria VS D	15	0	5	151	59	0.086	$\approx$



(a) f_4 (b) f_5 (c) f_6
(d) f_7 (e) f_8 (f) f_9
(g) f_{10} (h) f_{11} (i) f_{12}
(j) f_{13} (k) f_{14} (l) f_{15}
(m) f_{16} (n) f_{17} (o) f_{18}

FIGURE 4. Fitness evaluations used to optimize each sub-component.

The experimental results of the proposed F3C and equal allocation CC, including mean and standard deviation, are illustrated in Table 3. As the former framework outperforms the latter, which illustrates the limitation of allocating equal shares of the available computational resources, more emphasis should be placed on the more effective $C_i\,O_j$ pair.

B. COMPARISONS OF DIFFERENT SELECTION CHARACTERISTICS

In this section, the behavior of the proposed algorithm is demonstrated by measuring the influence of combining different criteria on the selection of the most effective $C_i\,O_j$ pair. Thus, F3C adopted selection criteria based on:

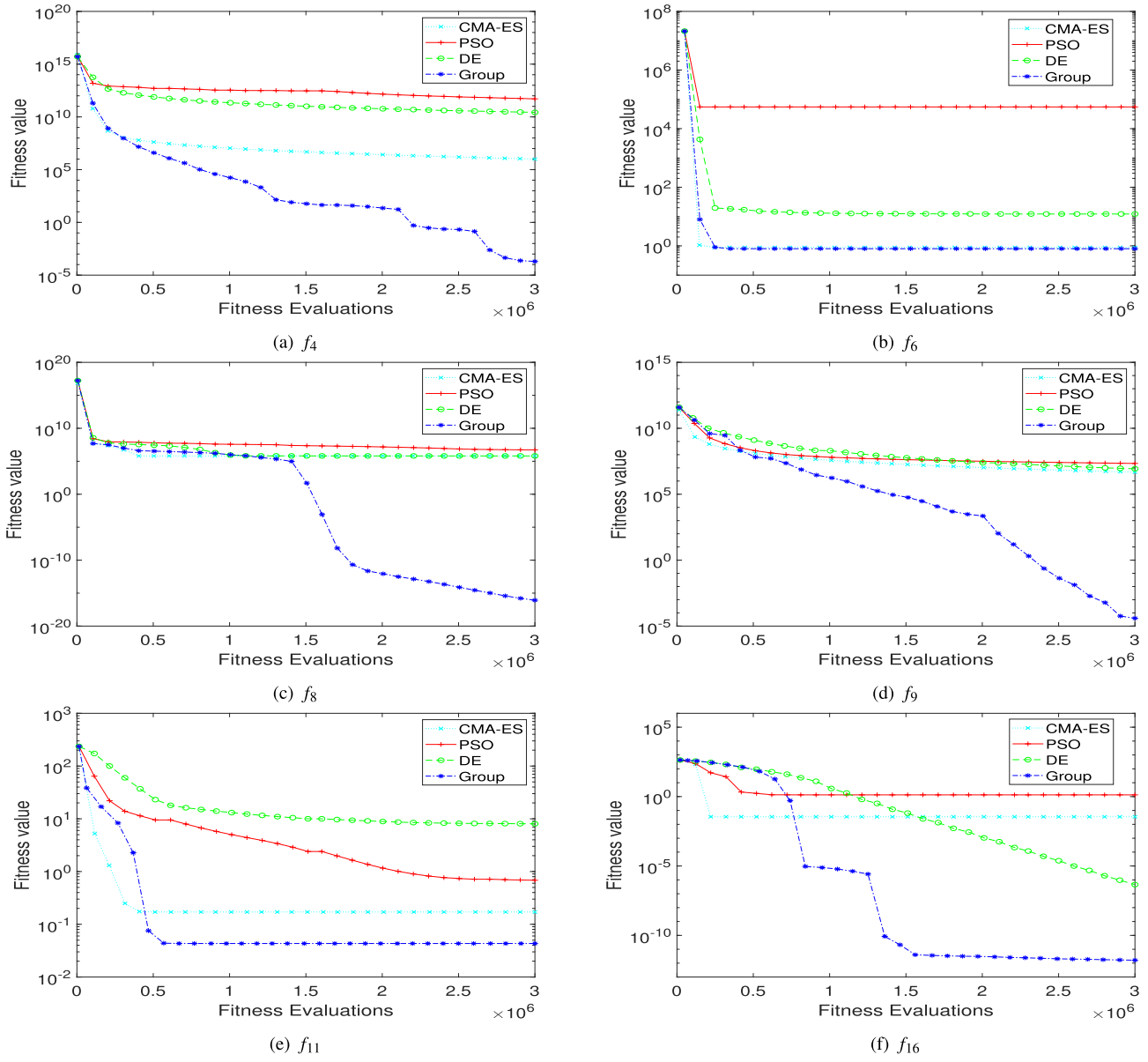


FIGURE 5. Convergence graphs of mean fitness values generated by group members using proposed approach based on 25 independent runs.

1) fitness improvement; 2) the diversity of the population and 3) both of them, with the results shown in Table 4, which clearly illustrate the importance of merging these characteristics on evaluating the effectiveness of $C_i\,O_j$ pair. Clearly, the F3C algorithm using complementary characteristics exhibits better results compared to using a single selection criterion and achieves the best results for 12 out of 20 test functions. According to the Wilcoxon test results shown in Table 5, regarding the average fitness values achieved, there is a significant difference only between using fitness improvement and both characteristics as the selection criteria.

C. COMPARISONS OF PROPOSED AND CANDIDATE OPTIMIZERS

The performance of the proposed F3C algorithm is compared with those of the other candidate optimizers, the CMA-ES, PSO and DE algorithms, and the results are shown in Table 6. It is clear that F3C's results are better than the others for the majority of test problems, which indicates that grouping complementary candidate optimizers in an appropriate manner enhances the overall optimization process. Also, for the test functions on which it is not the best, it obtains the second-best fitness values.

TABLE 6. Results Obtained From Proposed Algorithm and Other Candidate Optimizers for CEC'2010 Benchmark Problems.

Function	F3C	DECC	PSOCC	CMAESCC
	Mean ± (Std.)	Mean ± (Std.)	Mean ± (Std.)	Mean ± (Std.)
f_1	1.96E-08 ± (9.75E-08)	7.29E+04 ± (2.16E+05)	**1.28E-17** ± (3.45E-17)	2.79E+05 ± (2.89E+04)
f_2	2.72E+03 ± (6.36E+02)	3.10E+03 ± (3.58E+02)	**1.56E+03** ± (7.77E+01)	4.43E+03 ± (2.03E+02)
f_3	**9.71E-01** ± (4.51E-01)	1.13E+01 ± (2.05E+00)	1.60E+00 ± (1.92E-01)	1.13E+00 ± (2.61E-01)
f_4	**1.99E-04** ± (7.38E-04)	2.64E+10 ± (1.13E+10)	4.62E+11 ± (1.59E+11)	9.95E+05 ± (9.16E+04)
f_5	9.57E+07 ± (1.48E+07)	**6.55E+07** ± (1.04E+07)	2.01E+08 ± (1.32E+08)	9.91E+07 ± (1.73E+07)
f_6	**7.97E-01** ± (5.17E-01)	1.26E+01 ± (1.21E+00)	5.50E+04 ± (2.75E+05)	8.86E-01 ± (4.62E-01)
f_7	**3.14E-19** ± (1.66E-19)	8.11E+02 ± (6.48E+02)	1.85E-12 ± (4.34E-12)	7.80E-19 ± (9.71E-19)
f_8	**8.25E-17** ± (2.85E-16)	6.39E+05 ± (1.49E+06)	5.23E+06 ± (2.16E+07)	6.38E+05 ± (1.49E+06)
f_9	**3.93E-05** ± (1.39E-04)	8.42E+06 ± (1.29E+06)	2.16E+07 ± (2.13E+06)	4.72E+06 ± (4.55E+05)
f_{10}	**2.31E+03** ± (6.19E+02)	4.51E+03 ± (1.46E+02)	5.78E+03 ± (1.81E+02)	2.84E+03 ± (1.54E+02)
f_{11}	**4.32E-02** ± (3.48E-01)	8.04E+00 ± (6.84E-01)	6.84E-01 ± (1.04E+00)	1.64E-01 ± (3.86E-01)
f_{12}	**6.09E-22** ± (1.21E-22)	1.64E+03 ± (5.31E+02)	4.06E+01 ± (5.89E+01)	6.61E-22 ± (1.57E-22)
f_{13}	1.37E+01 ± (2.42E+01)	1.85E+03 ± (5.39E+02)	8.91E+02 ± (4.74E+02)	**5.58E+00** ± 3.82E+00)
f_{14}	6.97E+03 ± (1.83E+04)	3.42E+07 ± (2.43E+06)	8.35E+07 ± (6.24E+06)	**3.42E-16** ± (1.47E-15)
f_{15}	**1.90E+03** ± (1.03E+02)	4.69E+03 ± (1.61E+02)	6.79E+03 ± (8.04E+01)	1.91E+03 ± (9.20E+01)
f_{16}	**1.63E-12** ± (7.86E-14)	3.93E-07 ± (3.73E-08)	1.33E+00 ± (1.74E+00)	3.52E-02 ± (1.76E-01)
f_{17}	2.03E-23 ± (2.19E-24)	4.16E+02 ± (6.21E+01)	2.89E+03 ± (1.08E+03)	**1.15E-23** ± (4.79E-25)
f_{18}	5.59E+02 ± (5.74E+02)	9.16E+02 ± (6.49E+01)	2.14E+03 ± (9.54E+02)	**2.02E+01** ± (2.19E+01)
f_{19}	4.03E+05 ± (5.01E+05)	9.56E+05 ± (8.18E+04)	2.98E+06 ± (1.58E+05)	**5.42E+03** ± (5.93E+02)
f_{20}	**8.03E+02** ± (7.21E+01)	3.09E+05 ± (8.69E+05)	2.20E+03 ± (1.12E+02)	8.22E+02 ± (7.58E+01)

TABLE 7. Results Obtained From Proposed and State-of-the-Art Algorithms for CEC'2010 Benchmark Problems.

Fun.	Stats	F3C	CCFR2	TPLSO	FCRACC	APEC	CCFR	CBCC1	CBCC2	CCOS	MLSHA-DE-SPA	CSO	EADE	MOS	MA-SW-Chain
f_1	Mean	1.96E-08	4.63E-17	2.93E-19	1.22E-23	2.34E-17	1.19E-05	1.96E+06	6.38E+06	1.46E-07	4.01E-23	4.50E-12	1.00E-18	**1.50E-28**	3.80E-14
	Std.	9.75E-08	1.55E-17	1.81E-20	2.88E-23	1.16E-17	4.98E-06	2.05E+06	1.82E+07	6.54E-07	2.88E-23	5.94E-13	3.94E-18	5.55E-28	4.91E-14
f_2	Mean	2.72E+03	3.98E-02	6.17E+02	1.12E+02	5.45E+02	2.75E+01	4.33E+03	4.18E+03	2.59E+03	4.70E+01	7.42E+03	1.47E+03	**0.00E+00**	8.40E+02
	Std.	6.36E+02	1.99E-01	3.10E+01	2.00E+02	2.58E+01	5.35E+00	3.80E+02	5.38E+02	2.57E+02	7.83E+00	2.86E+02	7.43E+01	0.00E+00	4.88E+01
f_3	Mean	9.71E-01	1.33E+01	9.04E-14	1.30E+01	6.27E-12	1.22E+01	1.12E+01	1.10E+01	1.64E+00	9.84E-14	2.60E-09	6.76E+00	**0.00E+00**	5.76E-13
	Std.	4.51E-01	3.39E-01	2.15E-15	1.03E+00	1.98E-12	3.73E-01	8.96E-01	7.32E-01	2.27E-01	9.70E-15	2.62E-10	2.75E-01	0.00E+00	2.73E-13
f_4	Mean	**1.99E-04**	4.33E+10	3.63E+11	1.43E+10	1.89E+12	8.01E+10	1.81E+11	1.65E+10	2.06E+10	1.84E+11	7.25E+11	1.15E+11	5.16E+11	2.97E+11
	Std.	7.38E-04	1.61E+10	7.11E+10	1.85E+10	5.56E+11	5.12E+10	1.08E+11	3.62E+09	2.43E+09	6.32E+10	1.23E+11	4.44E+10	1.85E+11	6.19E+10
f_5	Mean	9.57E+07	7.88E+07	9.25E+06	7.10E+07	2.24E+08	6.72E+07	7.02E+07	6.43E+07	4.59E+07	6.07E+07	**2.86E+06**	9.66E+07	4.93E+08	2.18E+08
	Std.	1.48E+07	1.54E+07	2.36E+06	9.09E+06	4.45E+07	1.18E+07	1.05E+07	1.31E+07	8.55E+06	9.52E+06	1.79E+06	2.68E+07	6.93E+07	5.75E+07
f_6	Mean	7.97E-01	3.73E+05	1.71E-07	1.31E+01	5.20E+06	6.85E+05	8.14E+04	4.11E+04	3.52E+04	**6.60E-09**	8.21E-07	1.90E+01	1.97E+07	1.42E+05
	Std.	5.17E-01	5.86E+05	9.00E-09	2.51E+00	1.49E+06	7.20E+05	2.84E+05	2.05E+05	1.76E+05	1.29E-09	2.68E-08	1.50E-01	1.15E+05	3.96E+05
f_7	Mean	3.14E-19	2.14E-04	2.60E-04	**4.92E-20**	1.95E+05	1.44E-03	1.23E+05	1.26E+10	1.51E-11	8.08E-06	2.01E+04	4.38E-01	3.54E+07	1.17E+02
	Std.	1.66E-19	8.44E-05	1.71E-04	1.79E-20	1.20E+05	3.26E-04	1.09E+05	1.48E+10	2.25E-11	7.44E-06	3.86E+03	1.86E+00	3.22E+07	2.37E+02
f_8	Mean	**8.25E-17**	3.19E+05	3.01E+06	3.32E+05	1.67E+4	3.19E+05	7.50E+06	3.72E+07	6.38E+05	4.43E+03	3.87E+07	1.59E+05	3.75E+06	6.90E+06
	Std.	2.85E-16	1.10E+06	2.60E+05	1.10E+06	3.22E+04	1.10E+06	1.84E+07	3.47E+07	1.49E+06	2.30E+04	6.81E+04	7.97E+05	4.40E+06	1.90E+07
f_9	Mean	**3.93E-05**	8.00E+06	2.07E+07	3.66E+06	8.78E+07	9.40E+06	1.02E+07	3.40E+08	6.76E+06	1.65E+07	7.03E+07	3.59E+07	1.13E+07	1.49E+07
	Std.	1.39E-04	7.90E+05	1.12E+06	5.91E+05	1.28E+07	9.71E+05	3.84E+06	2.67E+08	8.82E+05	2.34E+06	5.73E+06	3.21E+06	1.61E+06	1.61E+06
f_{10}	Mean	2.31E+03	2.02E+03	**5.37E+02**	8.05E+03	3.94E+03	1.41E+03	2.59E+03	4.90E+03	2.34E+03	5.01E+03	9.60E+03	2.61E+03	6.28E+03	2.01E+03
	Std.	6.19E+02	2.24E+02	2.93E+01	6.22E+02	1.80E+02	7.72E+01	1.48E+02	6.37E+02	1.95E+02	2.72E+02	7.67E+01	1.40E+02	3.12E+02	1.59E+02
f_{11}	Mean	4.32E-02	1.06E+01	**1.25E-12**	1.30E+01	5.88E+01	1.98E+01	2.69E+01	2.75E+01	1.36E-01	8.96E+01	4.02E-08	1.21E+02	3.08E+01	3.86E+01
	Std.	3.48E-01	1.99E+00	4.53E-14	1.05E+00	7.65E+00	3.39E+00	2.64E+00	3.18E+00	3.84E-01	1.38E+01	5.12E-09	5.69E+00	6.07E+00	8.06E+00
f_{12}	Mean	**6.09E-22**	2.68E-01	4.45E+03	3.50E-04	4.51E+04	4.50E-01	3.53E+04	5.07E+04	6.65E+00	4.35E+01	4.37E+05	3.02E+04	4.39E+03	3.24E-06
	Std.	1.21E-22	3.57E-01	3.86E+02	1.53E-04	3.92E+03	8.90E-01	1.11E+04	1.10E+04	6.40E+00	9.61E+00	6.22E+04	5.52E+03	2.92E+03	5.78E-07
f_{13}	Mean	**1.37E+01**	3.68E+02	6.19E+02	1.99E+02	1.33E+03	5.34E+02	9.06E+04	1.29E+07	4.01E+02	9.95E+01	6.29E+02	1.02E+03	3.32E+02	9.83E+02
	Std.	2.42E+01	1.16E+02	1.04E+02	6.58E+01	4.61E+02	1.57E+02	6.11E+04	7.36E+06	8.89E+01	8.47E+01	2.32E+02	2.02E+02	1.19E+02	5.66E+02
f_{14}	Mean	**6.97E+03**	3.05E+07	6.55E+07	1.82E+07	2.34E+08	3.13E+07	2.24E+07	5.35E+09	4.71E+07	6.02E+07	2.49E+08	1.47E+08	2.05E+07	3.85E+07
	Std.	1.83E+04	2.20E+06	2.78E+06	1.58E+06	1.41E+07	3.85E+06	2.27E+06	6.00E+08	4.55E+06	6.03E+06	1.53E+07	8.42E+06	3.60E+06	3.86E+06
f_{15}	Mean	**1.90E+03**	4.64E+03	9.60E+03	8.57E+03	7.92E+03	3.21E+03	2.84E+03	3.22E+03	4.59E+03	3.77E+03	1.01E+04	3.26E+03	1.29E+04	2.68E+03
	Std.	1.03E+02	2.76E+02	6.15E+01	5.77E+02	2.58E+02	1.65E+02	2.65E+02	4.17E+02	1.44E+02	9.59E+01	5.23E+01	1.49E+02	3.48E+02	9.95E+01
f_{16}	Mean	**1.63E-12**	5.81E+00	1.75E-12	2.86E+00	7.51E+01	2.05E+01	1.87E+01	1.91E+01	1.72E-01	2.49E+02	5.89E-08	2.97E+02	3.96E+02	9.95E+01
	Std.	7.86E-14	2.10E+00	3.21E-13	7.07E+00	7.48E+00	3.53E+00	3.83E+00	2.76E+00	5.10E-01	1.65E+01	5.61E-09	9.36E+00	3.47E+00	1.53E+01
f_{17}	Mean	**2.03E-23**	6.20E+01	5.38E+04	5.10E+00	1.29E+02	6.71E+01	1.49E+01	1.24E+02	3.85E+03	6.16E+02	2.20E+06	1.53E+05	8.45E+03	1.27E+00
	Std.	2.19E-24	3.27E+01	2.08E+03	5.00E+00	4.86E+03	8.69E+01	7.01E+00	5.72E+01	1.11E+03	9.20E+01	1.55E+05	1.52E+04	5.04E+03	1.24E-01
f_{18}	Mean	**5.59E+02**	1.18E+03	1.39E+03	1.05E+03	2.67E+03	1.24E+03	4.10E+09	1.23E+11	1.12E+03	5.73E+02	1.73E+03	2.18E+03	8.96E+02	1.57E+03
	Std.	5.74E+02	1.70E+02	2.57E+02	1.49E+02	5.15E+02	1.81E+02	3.93E+09	1.21E+11	1.12E+03	1.76E+02	5.22E+02	3.36E+02	4.03E+02	6.73E+02
f_{19}	Mean	4.03E+05	1.77E+06	1.19E+06	9.57E+05	1.26E+06	1.23E+06	9.12E+05	9.11E+05	9.17E+05	4.51E+05	1.01E+07	1.34E+06	5.49E+05	**3.80E+05**
	Std.	5.01E+05	1.03E+05	3.91E+04	6.25E+04	5.20E+04	8.83E+04	7.11E+04	6.02E+04	4.88E+04	2.34E+04	5.64E+05	6.45E+04	8.38E+04	2.34E+04
f_{20}	Mean	8.03E+02	1.03E+08	1.31E+03	2.17E+07	2.13E+03	1.68E+09	1.41E+07	6.97E+09	2.47E+03	1.26E+02	9.85E+02	2.11E+03	**9.23E+01**	1.06E+03
	Std.	7.21E+01	4.35E+07	7.91E+01	3.28E+07	2.02E+02	1.76E+09	1.96E+07	1.12E+09	2.11E+02	9.55E+01	1.05E+03	1.45E+02	8.99E+01	1.06E+03

Note: we have obtained and used the source codes of CCFR2, CCOS, CCFR and EADE from their corresponding authors, and the results of other algorithms are taken from the published papers.

As a further illustration, Fig. 5 shows the convergence graphs of the F3C, DE, PSO and CMA-ES algorithms on some selected functions, with each point in the plots calculated by taking the average of 25 independent runs. It can be

TABLE 8. Results Obtained From Wilcoxon Signed Rank Test of Proposed and State-of-the-Art Algorithms for the CEC'2010 Benchmark Problems.

Algorithms	Better	Equal	Worse	$R+$	$R-$	p-value	Decision
CCFR2	16	0	4	173	37	**0.011**	+
TPLSO	13	0	7	153	57	0.073	≈
FCRACC	16	0	4	177	33	**0.007**	+
APEC	17	0	3	197	13	**0.001**	+
CCFR	17	0	3	171	39	**0.014**	+
CBCC1	19	0	1	192	18	**0.001**	+
CBCC2	19	0	1	197	13	**0.001**	+
CCOS	18	0	2	183	27	**0.004**	+
MLSHADE-SPA	14	0	6	160	50	**0.040**	+
CSO	15	0	5	179	31	**0.006**	+
EADE	18	0	2	200	10	**0.000**	+
MOS	16	0	4	192	18	**0.001**	+
MA-SW-Chain	15	0	5	170	40	**0.015**	+

TABLE 9. Results Obtained From Friedman Ranking Test of Proposed and Other State-of-the-Art Algorithms for CEC'2010 Benchmark Problems.

Algorithms	Mean ranking
F3C	**3.35**
CCFR2	6.83
TPLSO	6.55
FCRACC	5.70
APEC	10.05
CCFR	7.23
CBCC1	8.80
CBCC2	10.30
CCOS	6.50
MLSHADE-SPA	5.80
CSO	9.60
EADE	9.30
MOS	7.85
MA-SW-Chain	7.15

seen that the proposed F3C algorithm produces better convergence speeds and that for all the approaches, the available resources (3×10^6) are not sufficient to converge to the global optimum (i.e., 0).

D. COMPARISONS OF PROPOSED AND STATE-OF-THE-ART ALGORITHMS

In this section, the performance of the proposed F3C algorithm is compared with those of the following state-of-the-art ones in the literature, namely, CCFR2 [45], two-phase learning-based swarm optimizer (TPLSO) [58], FCRACC [44], affinity propagation assisted and evolution consistency based decomposition (APEC) [59], CCFR [42], CC with optimizer selection (CCOS) [38], CBCC1 [12], CBCC2 [12], LSHADE semi-parameter adaptation memetic framework (MLSHADE-SPA) [60], competitive swarm optimizer (CSO) [61], enhanced adaptive differential evolution (EADE) [62], multiple offspring sampling (MOS) [63] and memetic algorithm based on local search chains (MA-SW-Chain) [25].

The results presented in Table 7 demonstrate that the F3C algorithm is capable of achieving the best results for 10 of the 20 test problems. It is clear that the F3C algorithm outperforms the others on the majority of partially separable functions ($f_4 - f_{18}$) and as for both the fully separable ($f_1 - f_3$) and fully nonseparable (f_{19} and f_{20}) ones, all the decision variables are grouped and optimized together in one

TABLE 10. Results Obtained From Proposed and State-of-the-Art Algorithms for CEC'2013 Benchmark Problems.

Fun.	Stats	F3C	CCFR2	TPLSO	FCRACC	APEC	CCFR	BBCC	CBCC1	CBCC2	CBCC3	CCOS	CRO	CSO	MLCC	MA-SW-Chain
f_1	Mean	9.76E-08	**0.00E+00**	3.13E-19	1.45E-23	2.72E-17	5.72E-17	NA	1.40E+07	1.40E+07	8.65E+05	2.16E-02	1.84E+06	7.76E-12	1.32E-06	8.49E-13
	Std.	3.92E-07	0.00E+00	1.39E-20	4.62E-23	1.68E-17	1.08E-17	NA	3.60E+07	3.60E+07	NA	7.61E-02	1.43E+06	1.30E-12	6.47E-07	1.11E-12
f_2	Mean	4.55E+03	**3.16E-18**	7.17E+02	1.26E+02	4.98E+02	5.49E-01	NA	2.10E+04	2.10E+04	1.41E+04	4.61E+03	9.84E+02	8.54E+03	2.11E-02	1.22E+03
	Std.	4.08E+02	7.19E-18	2.52E+01	2.64E+01	2.17E+01	1.50E+00	NA	9.90E+02	9.90E+02	NA	4.48E+02	9.65E+01	2.67E+02	3.84E-02	1.16E+02
f_3	Mean	2.09E+01	**2.00E+01**	2.16E+01	2.01E+01	2.01E+01	**2.00E+01**	NA	2.10E+01	2.10E+01	2.06E+01	2.08E+01	2.01E+01	2.16E+01	2.00E+01	2.14E+01
	Std.	2.35E-02	3.56E-07	5.20E-03	6.17E-02	1.01E-02	3.12E-07	NA	1.10E-02	1.10E-02	NA	9.56E-03	2.36E-02	6.15E-03	9.65E-04	5.73E-02
f_4	Mean	**2.98E-01**	2.00E+07	5.57E+09	7.62E+06	2.19E+10	4.66E+07	3.27E+08	1.60E+08	6.60E+10	3.39E+07	3.83E+07	1.55E+10	1.31E+10	1.62E+12	4.58E+09
	Std.	1.49E+00	9.87E+06	8.76E+08	2.51E+06	9.16E+09	2.34E+07	1.41E+08	6.00E+07	5.60E+09	1.77E+07	1.07E+07	7.90E+09	2.54E+09	8.40E+11	2.51E+09
f_5	Mean	2.07E+06	5.57E+06	6.71E+05	4.59E+06	1.19E+07	2.44E+06	1.45E+06	2.50E+06	2.40E+06	2.14E+06	1.24E+06	2.38E+07	**5.92E+05**	1.92E+07	1.87E+06
	Std.	4.26E+05	1.80E+06	8.95E+04	5.00E+05	1.88E+06	3.20E+05	3.23E+05	4.20E+05	4.50E+05	4.24E+05	2.14E+05	6.26E+06	1.08E+05	5.57E+06	3.13E+05
f_6	Mean	1.02E+06	1.06E+06	1.06E+06	1.05E+06	1.05E+06	1.06E+06	1.04E+06	1.10E+06	1.10E+06	1.05E+06	1.06E+06	1.06E+06	1.06E+06	1.05E+06	**1.01E+06**
	Std.	3.12E+04	1.32E+03	9.27E+02	1.79E+03	7.48E+03	1.22E+03	1.47E+05	1.90E+03	1.70E+03	3.37E+03	3.08E+03	1.13E+03	1.10E+03	3.52E+03	1.56E+04
f_7	Mean	**2.64E-21**	2.88E+07	2.95E+05	1.54E+06	9.04E+07	7.60E+06	2.14E+05	1.90E+07	9.60E+07	2.95E+07	5.91E+06	2.78E+08	5.86E+06	8.44E+09	3.45E+06
	Std.	4.84E-22	3.99E+07	1.38E+05	7.67E+06	9.29E+07	1.85E+07	1.56E+05	2.40E+07	3.70E+08	2.78E+07	1.63E+07	1.65E+08	2.56E+06	4.44E+09	1.29E+06
f_8	Mean	**1.90E+05**	2.17E+10	1.33E+14	2.25E+09	4.76E+14	9.47E+09	6.96E+12	2.00E+13	1.00E+12	6.74E+10	9.13E+10	4.56E+14	2.64E+14	7.66E+16	4.85E+13
	Std.	6.41E+05	6.47E+10	2.10E+13	1.61E+09	2.13E+14	1.66E+10	4.21E+12	2.80E+13	1.30E+11	8.86E+10	1.23E+11	3.07E+14	5.86E+13	3.75E+16	1.04E+13
f_9	Mean	1.56E+08	3.50E+08	**3.98E+07**	3.99E+08	7.97E+08	1.85E+08	1.21E+08	2.50E+08	2.20E+08	1.70E+08	1.08E+08	5.27E+08	6.07E+07	1.19E+09	1.07E+08
	Std.	4.08E+07	1.32E+08	5.35E+06	5.05E+07	7.79E+07	2.84E+07	3.42E+07	3.80E+07	2.80E+07	3.16E+07	2.98E+07	5.02E+07	1.61E+07	3.51E+08	1.71E+07
f_{10}	Mean	**9.15E+07**	9.47E+07	9.39E+07	9.27E+07	9.24E+07	9.47E+07	9.21E+07	9.40E+07	9.40E+07	9.28E+07	9.41E+07	9.44E+07	9.40E+07	9.28E+07	9.18E+07
	Std.	1.95E+06	3.91E+05	1.78E+05	2.62E+05	4.77E+05	2.43E+05	1.30E+07	2.80E+05	2.30E+05	7.16E+05	4.56E+05	2.70E+05	1.52E+07	4.79E+05	1.08E+06
f_{11}	Mean	**4.61E+00**	5.31E+09	1.29E+08	4.83E+08	5.18E+09	2.74E+08	5.99E+07	3.00E+09	4.90E+10	7.70E+08	2.20E+08	2.91E+10	9.30E+11	1.05E+12	2.19E+08
	Std.	4.56E+00	2.39E+10	3.88E+07	4.59E+08	5.68E+09	3.00E+08	6.64E+07	1.00E+10	9.50E+10	2.80E+08	5.72E+07	2.18E+10	1.03E+10	5.06E+11	3.04E+07
f_{12}	Mean	**9.80E+02**	3.78E+07	1.12E+03	1.08E+07	2.16E+03	1.52E+09	NA	6.10E+08	6.10E+08	5.81E+07	2.60E+03	3.69E+03	1.07E+03	8.82E+04	1.25E+03
	Std.	6.66E+01	2.09E+07	8.13E+01	3.84E+07	1.46E+02	7.10E+08	NA	7.10E+08	7.10E+08	NA	5.03E+02	3.38E+03	7.78E+01	3.19E+4	1.07E+02
f_{13}	Mean	7.70E+08	1.09E+09	9.07E+07	3.41E+08	4.13E+09	9.04E+08	NA	9.50E+08	9.50E+08	6.03E+08	2.86E+08	5.33E+09	6.68E+08	4.49E+10	**1.98E+07**
	Std.	7.16E+08	3.68E+08	3.55E+07	1.28E+08	1.30E+09	4.62E+08	NA	5.40E+08	5.40E+08	NA	1.63E+08	6.23E+08	2.44E+08	1.24E+10	1.86E+06
f_{14}	Mean	2.85E+07	8.33E+08	**2.30E+07**	2.70E+08	4.21E+10	3.06E+09	NA	2.20E+09	2.20E+09	1.11E+09	9.88E+08	6.08E+10	3.62E+09	8.66E+11	1.36E+08
	Std.	2.85E+06	5.15E+08	4.38E+06	3.12E+08	2.89E+10	2.72E+09	NA	2.10E+09	2.10E+09	NA	2.34E+09	3.53E+10	1.44E+09	3.68E+11	2.15E+07
f_{15}	Mean	**2.40E+06**	6.13E+06	2.42E+06	4.68E+06	8.08E+06	7.83E+06	NA	8.30E+06	8.30E+06	7.11E+06	3.06E+06	1.88E+07	7.87E+07	3.72E+08	5.71E+06
	Std.	2.50E+05	5.85E+05	1.19E+05	1.22E+06	1.31E+06	1.73E+06	NA	3.30E+06	3.30E+06	NA	7.32E+05	3.48E+06	6.50E+06	2.62E+08	7.73E+05

TABLE 11. Results Obtained From Wilcoxon Signed Rank Test of Proposed and State-of-the-Art Algorithms for CEC'2013 Benchmark Problems.

Algorithms	Better	Equal	Worse	$R+$	$R-$	p-value	Decision
CCFR2	12	0	3	114	6	**0.002**	+
TPLSO	9	0	6	73	47	0.460	≈
FCRACC	11	0	4	101	19	**0.02**	+
APEC	12	0	3	113	7	**0.003**	+
CCFR	12	0	3	114	6	**0.002**	+
BBCC	6	0	2	27	9	0.208	≈
CBCC1	15	0	0	120	0	**0.001**	+
CBCC2	15	0	0	120	0	**0.001**	+
CBCC3	13	0	2	107	13	**0.008**	+
CCOS	11	0	4	87	33	0.125	≈
CRO	13	0	2	116	4	**0.001**	+
CSO	11	0	4	92	28	0.069	≈
MLCC	13	0	2	115	5	**0.002**	+
MA-SW-Chain	9	0	6	81	39	0.233	≈

sub-component (where size = 1000), there is no significant impact on selecting the most effective sub-components for resource allocation.

Based on the Wilcoxon signed rank test, Table 8 shows the comparison of the average values of the proposed F3C and other peer algorithms. It can be seen that the F3C algorithm obtains higher R+ than R− values and its performance is significantly better than those of all the others, except the TPLSO algorithm for which there is no significant difference between them. Furthermore, of the 260 cases, it is superior, inferior and similar to the others for 213, 47 and 0, respectively. Therefore, it can be deduced that it performs better than the others for 81.92% of all instances and is only outperformed for 18.07%. Also, the results obtained from the Friedman ranking test presented in Table 9 demonstrate that it ranks first (has the smallest value), as shown in bold, followed by FCRACC, MLSHADE-SPA, CCOS, TPLSO, CCFR2, MA-SW-Chain, CCFR, MOS, CBCC1, EADE, CSO, APEC and CBCC2, respectively.

E. TESTING F3C ON CEC'2013 BENCHMARK PROBLEMS

In this section, a set of 15 benchmark functions proposed in the CEC'2013 special session on LSGO [53] are solved by adopting an ideal decomposition, whereby all the FEs are involved in the optimization stage. The performance of the proposed F3C algorithm is compared with those of the following recently proposed state-of-the-art ones in the literature, namely, CCFR2 [45], TPLSO [58], FCRACC [44], APEC [59], CCFR [42], BBCC [43], CCOS [38], CBCC1 [12], CBCC2 [12], CBCC3 [40], coral reefs optimization (CRO) [64], CSO [61], multilevel CC (MLCC) [65] and MA-SW-Chain [25].

The results of the comparison presented in Table 10 demonstrate that the F3C algorithm is superior for 7 of the 15 test problems. Also, similar to its performance for the CEC'2010 benchmark problems, it is better than the other algorithms for the majority of the partially separable functions ($f_4 - f_{11}$).

The Wilcoxon signed rank test results shown in Table 11 are the average values obtained from all the algorithms. It can be seen that the F3C algorithm has higher R+ than R− values and outperforms the others. Also, as it is superior, inferior and similar to the others in 162, 41 and 0 cases, respectively. Although there is no significant difference between F3C and some peer algorithms (their p-value is greater than 0.05), it is more successful. It performs the best for 79.8% of cases. Furthermore, the Friedman rank test, according to the obtained mean results illustrated in Table 12, shows that it ranks first, followed by TPLSO, MA-SW-Chain, FCRACC, CCOS, CCFR2, CBCC3, CCFR, CSO, APEC, CBCC1, MLCC, CBCC2 and CRO, respectively.

TABLE 12. Results Obtained From Friedman Ranking Test of Proposed and State-of-the-Art Algorithms for CEC'2013 Benchmark Problems.

Algorithms	Mean ranking
F3C	**3.60**
CCFR2	7.07
TPLSO	4.87
FCRACC	4.97
APEC	8.63
CCFR	7.27
CBCC1	10.00
CBCC2	10.67
CBCC3	7.20
CCOS	6.30
CRO	10.90
CSO	8.27
MLCC	10.40
MA-SW-Chain	4.87

Note: BBCC is not included in this test because of the non-availability of some functions.

V. CONCLUSION

In this article, the F3C algorithm for handling LSGO problems is presented. In contrast to a typical CC framework, in which all the sub-components share the computational resources equally and only a single optimization algorithm is used to solve them, in this one, a group of optimizers cooperate in an effective way to evolve the sub-components according to heuristic fuzzy rules. This heuristic was used to allocate a computational budget to the most effective sub-component and optimizer pair based on two criteria, namely, the fitness improvement and diversity of the population. The performance of our proposed F3C algorithm was evaluated by using it to solve many LSGO benchmark functions. Interestingly, the experimental analysis revealed that it outperforms an equal-allocation CC, single selection criterion, single-candidate optimizer and state-of-the-art algorithms.

In the future, the potential of adding adaptive subpopulations to each candidate optimizers within F3C will be investigated. The selection of optimizer can be done based on a learning process which will definitely consume extra computational time. So it needs to find a compromise design that will improve the effectiveness and efficiency of the algorithm. We are further interested in changing the decomposition structure while solving a problem and using an alternative design of allocations, such as switching optimizers between

sub-problems and/or using more than one operator or algorithm in a single sub-problem.

REFERENCES

[1] T. Bäck, D. B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. Boca Raton, FL, USA: CRC Press, 1997.
[2] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. MHS 6th Int. Symp. Micro Mach. Human Sci.*, 1995, pp. 39–43.
[3] Y. Zhang, D.-W. Gong, X.-Z. Gao, T. Tian, and X.-Y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Inf. Sci.*, vol. 507, pp. 67–85, Jan. 2020.
[4] Y. Zhang, S. Cheng, Y. Shi, D.-W. Gong, and X. Zhao, "Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm," *Expert Syst. Appl.*, vol. 137, pp. 46–58, Dec. 2019.
[5] R. E. Bellman, "Dynamic programming," in *Rand Corporation Research Study*. Princeton, NJ, USA: Princeton Univ. Press, 1957.
[6] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Berlin, Germany: Springer, 1994, pp. 249–257.
[7] Z. Ren, A. Chen, M. Wang, Y. Yang, Y. Liang, and K. Shang, "Bi-hierarchical cooperative coevolution for large scale global optimization," *IEEE Access*, vol. 8, pp. 41913–41928, 2020.
[8] X.-F. Song, Y. Zhang, Y.-N. Guo, X.-Y. Sun, and Y.-L. Wang, "Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 882–895, Oct. 2020.
[9] Y. Liu, X. Yao, Q. Zhao, and T. Higuchi, "Scaling up fast evolutionary programming with cooperative coevolution," in *Proc. Congr. Evol. Comput.*, vol. 2, 2001, pp. 1101–1108.
[10] Z. Yang, K. Tang, and X. Yao, "Differential evolution for high-dimensional function optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Sep. 2007, pp. 3523–3530.
[11] W. Chen, T. Weise, Z. Yang, and K. Tang, "Large-scale global optimization using cooperative coevolution with variable interaction learning," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Berlin, Germany: Springer, 2010, pp. 300–309.
[12] M. N. Omidvar, X. Li, and X. Yao, "Smart use of computational resources based on contribution for cooperative co-evolutionary algorithms," in *Proc. 13th Annu. Conf. Genetic Evol. Comput.*, 2011, pp. 1115–1122.
[13] M. N. Omidvar, X. Li, and K. Tang, "Designing benchmark problems for large-scale continuous optimization," *Inf. Sci.*, vol. 316, pp. 419–436, Sep. 2015.
[14] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[15] N. Hansen, "The CMA evolution strategy: A tutorial," 2016, *arXiv:1604.00772*. [Online]. Available: http://arxiv.org/abs/1604.00772
[16] S. M. Elsayed, "Evolutionary approach for constrained optimization," Ph.D. dissertation, School Eng. Inf. Technol., Univ. New South Wales, Canberra, ACT, Australia, 2012.
[17] N. Hansen, "Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed," in *Proc. 11th Annu. Conf. Companion Genetic Evol. Comput. Conf.*, 2009, pp. 2389–2396.
[18] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, 2003.
[19] F. Hutter, L. Xu, H. H. Hoos, and K. Leyton-Brown, "Algorithm runtime prediction: Methods & evaluation," *Artif. Intell.*, vol. 206, pp. 79–111, Jan. 2014.
[20] M. A. Muñoz, Y. Sun, M. Kirley, and S. K. Halgamuge, "Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges," *Inf. Sci.*, vol. 317, pp. 224–245, Oct. 2015.
[21] A. LaTorre, S. Muelas, and J.-M. Pena, "Multiple offspring sampling in large scale global optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–8.
[22] B. Kazimipour, X. Li, and A. K. Qin, "A review of population initialization techniques for evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 2585–2592.
[23] M. A. Meselhi, S. M. Elsayed, D. L. Essam, and R. A. Sarker, "Fast differential evolution for big optimization," in *Proc. Softw., Knowl., Inf. Manage. Appl. (SKIMA)*, 2017, pp. 1–6.
[24] J. Brest and M. S. Maučec, "Self-adaptive differential evolution algorithm using population size reduction and three strategies," *Soft Comput.*, vol. 15, no. 11, pp. 2157–2174, Nov. 2011.
[25] D. Molina, M. Lozano, and F. Herrera, "MA-SW-chains: Memetic algorithm based on local search chains for large scale continuous global optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–8.
[26] B. Kazimipour, M. N. Omidvar, X. Li, and A. K. Qin, "A novel hybridization of opposition-based learning and cooperative co-evolutionary for large-scale optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 2833–2840.
[27] M. A. Potter and K. A. D. Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evol. Comput.*, vol. 8, no. 1, pp. 1–29, Mar. 2000.
[28] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Inf. Sci.*, vol. 178, no. 15, pp. 2985–2999, Aug. 2008.
[29] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 378–393, Jun. 2014.
[30] Y. Mei, M. N. Omidvar, X. Li, and X. Yao, "A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization," *ACM Trans. Math. Softw.*, vol. 42, no. 2, p. 13, 2016.
[31] Y. Sun, M. Kirley, and S. K. Halgamuge, "Extended differential grouping for large scale global optimization with direct and indirect variable interactions," in *Proc. Genetic Evol. Comput. Conf.*, 2015, pp. 313–320.
[32] M. N. Omidvar, M. Yang, Y. Mei, X. Li, and X. Yao, "DG2: A faster and more accurate differential grouping for large-scale black-box optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 6, pp. 929–942, Dec. 2017.
[33] X.-M. Hu, F.-L. He, W.-N. Chen, and J. Zhang, "Cooperation coevolution with fast interdependency identification for large scale optimization," *Inf. Sci.*, vol. 381, pp. 142–160, Mar. 2017.
[34] M. A. Meselhi, R. A. Sarker, D. L. Essam, and S. M. Elsayed, "Enhanced differential grouping for large scale optimization," in *Proc. 10th Int. Joint Conf. Comput. Intell. (IJCCI INSTICC)*, vol. 1, 2018, pp. 217–224.
[35] Y. Sun, M. Kirley, and S. K. Halgamuge, "A recursive decomposition method for large scale continuous optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 647–661, Oct. 2018.
[36] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 210–224, Apr. 2012.
[37] Y. Guo, X. Cao, H. Yin, and Z. Tang, "Coevolutionary optimization algorithm with dynamic sub-population size," *Int. J. Innov. Comput., Inf. Control*, vol. 3, no. 2, pp. 435–448, 2007.
[38] Y. Sun, M. Kirley, and X. Li, "Cooperative co-evolution with online optimizer selection for large-scale optimization," in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2018, pp. 1079–1086.
[39] G. A. Trunfio, P. Topa, and J. Wąs, "A new algorithm for adapting the configuration of subcomponents in large-scale optimization with cooperative coevolution," *Inf. Sci.*, vol. 372, pp. 773–795, Dec. 2016.
[40] M. N. Omidvar, B. Kazimipour, X. Li, and X. Yao, "CBCC3—A contribution-based cooperative co-evolutionary algorithm with improved exploration/exploitation balance," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 3541–3548.
[41] G. A. Trunfio, "Adaptation in cooperative coevolutionary optimization," in *Adaptation and Hybridization in Computational Intelligence*. Cham, Switzerland: Springer, 2015, pp. 91–109.
[42] M. Yang, M. N. Omidvar, C. Li, X. Li, Z. Cai, B. Kazimipour, and X. Yao, "Efficient resource allocation in cooperative co-evolution for large-scale global optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 493–505, Aug. 2017.
[43] B. Kazimipour, M. N. Omidvar, A. K. Qin, X. Li, and X. Yao, "Bandit-based cooperative coevolution for tackling contribution imbalance in large-scale optimization problems," *Appl. Soft Comput.*, vol. 76, pp. 265–281, Mar. 2019.
[44] Z. Ren, Y. Liang, A. Zhang, Y. Yang, Z. Feng, and L. Wang, "Boosting cooperative coevolution for large scale optimization with a fine-grained computation resource allocation strategy," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4180–4193, Dec. 2019.
[45] M. Yang, A. Zhou, C. Li, J. Guan, and X. Yan, "CCFR2: A more efficient cooperative co-evolutionary framework for large-scale global optimization," *Inf. Sci.*, vol. 512, pp. 64–79, Feb. 2020.
[46] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965.
[47] M. Mukaidono, *Fuzzy Logic for Beginners*. Singapore: World Scientific, 2001.

[48] L. A. Zadeh, "Fuzzy logic = computing with words," in *Computing With Words in Information/Intelligent Systems 1*. Berlin, Germany: Springer, 1999, pp. 3–23.
[49] D. Wu, "Twelve considerations in choosing between Gaussian and trapezoidal membership functions in interval type-2 fuzzy logic controllers," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Jun. 2012, pp. 1–8.
[50] Y. Pan, L. Zhang, Z. Li, and L. Ding, "Improved fuzzy Bayesian network-based risk analysis with interval-valued fuzzy sets and D–S evidence theory," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 9, pp. 2063–2077, Sep. 2020.
[51] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proc. Inst. Elect. Eng.*, vol. 121, no. 12, pp. 1585–1588, Dec. 1974.
[52] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise, "Benchmark functions for the CEC'2010 special session and competition on large-scale global optimization," Nat. Inspired Comput. Appl. Lab., Univ. Sci. Technol. China, Hefei, China, Tech. Rep., 2009.
[53] X. Li, K. Tang, M. N. Omidvar, Z. Yang, K. Qin, and H. China, "Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization," *Gene*, vol. 7, no. 33, p. 8, 2013.
[54] Z. Yang, K. Tang, and X. Yao, "Self-adaptive differential evolution with neighborhood search," in *Proc. IEEE Congr. Evol. Comput. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1110–1116.
[55] R. Cheng and Y. Jin, "A social learning particle swarm optimization algorithm for scalable optimization," *Inf. Sci.*, vol. 291, pp. 43–60, Jan. 2015.
[56] R. F. Woolson, "Wilcoxon signed-rank test," in *Wiley Encyclopedia of Clinical Trials*. Hoboken, NJ, USA: Wiley, 2007, pp. 1–3.
[57] E. Theodorsson-Norheim, "Friedman and quade tests: BASIC computer program to perform nonparametric two-way analysis of variance and multiple comparisons on ranks of several related samples," *Comput. Biol. Med.*, vol. 17, no. 2, pp. 85–99, Jan. 1987.
[58] R. Lan, Y. Zhu, H. Lu, Z. Liu, and X. Luo, "A two-phase learning-based swarm optimizer for large-scale optimization," *IEEE Trans. Cybern.*, early access, Mar. 9, 2020, doi: 10.1109/TCYB.2020.2968400.
[59] Q. Yang, W. Chen, and J. Zhang, "Evolution consistency based decomposition for cooperative coevolution," *IEEE Access*, vol. 6, pp. 51084–51097, 2018.
[60] A. A. Hadi, A. W. Mohamed, and K. M. Jambi, "LSHADE-SPA memetic framework for solving large-scale optimization problems," *Complex Intell. Syst.*, vol. 5, no. 1, pp. 25–40, Mar. 2019.
[61] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 191–204, Feb. 2015.
[62] A. W. Mohamed, "Solving large-scale global optimization problems using enhanced adaptive differential evolution algorithm," *Complex Intell. Syst.*, vol. 3, no. 4, pp. 205–231, Dec. 2017.
[63] A. LaTorre, S. Muelas, and J.-M. Peña, "A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: A scalability test," *Soft Comput.*, vol. 15, no. 11, pp. 2187–2199, Nov. 2011.
[64] S. Salcedo-Sanz, C. Camacho-Gómez, D. Molina, and F. Herrera, "A coral reefs optimization algorithm with substrate layers and local search for large scale global optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 3574–3581.
[65] Z. Yang, K. Tang, and X. Yao, "Multilevel cooperative coevolution for large scale optimization," in *Proc. IEEE Congr. Evol. Comput. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1663–1670.

MOHAMED A. MESELHI received the B.Sc. and M.Sc. degrees in computer engineering from Zagazig University, Egypt, and the Ph.D. degree in computer science from the University of New South Wales (UNSW) at Canberra, Australia, in 2020. He is currently a Research Associate with the School of Engineering and Information Technology (SEIT), UNSW at Canberra. His research interests include evolutionary algorithms and large-scale optimization.

SABER M. ELSAYED (Member, IEEE) received the Ph.D. degree in computer science from the Australian Defence Force Academy, University of New South Wales at Canberra, Canberra, Australia, in 2012. He is currently a Senior Lecturer with the School of Engineering and Information Technology, University of New South Wales at Canberra. His research interests include evolutionary algorithms, constraint-handling techniques for evolutionary algorithms, scheduling, big data, and cybersecurity using computational intelligence. He was a member of the program committee of several international conferences. He was a recipient of several IEEE-CEC competitions. He is an Editorial Board Member of the *International Journal of Business Intelligence and Data Mining*. He serves as a reviewer for several international journals.

RUHUL A. SARKER (Member, IEEE) received the Ph.D. degree from Dalhousie University, Halifax, Canada, in 1992. He is currently a Professor with the School of Engineering and Information Technology, University of New South Wales at Canberra, ACT, Australia. He is the Lead Author of the book *Optimization Modelling: A Practical Approach* (CRC Press, 2007). He published more than 300 refereed articles in international journals, edited books, and conference proceedings. His current research interests include evolutionary optimization and applied operations research. He is currently an Associate Editor of *Memetic-computing* journal, the *Journal of Industrial & Management Optimization*, and *Flexible Services and Manufacturing Journal*.

DARYL L. ESSAM (Member, IEEE) received the B.Sc. degree from the University of New England, Australia, in 1990, and the Ph.D. degree from the University of New South Wales at Canberra, Australia, in 2000. In 1991, he was an Associate Lecturer with the University of New England. Since 1994, he has been with the University of New South Wales at Canberra, where he is currently a Senior Lecturer. His research interest includes genetic algorithms, with a focus on both genetic programming and multiobjective optimization. He was a Finance Chair of the IEEE CEC 2003 and a Proceedings Co-Chair of WCCI 2012.