IADE: An Improved Differential Evolution Algorithm to Preserve Sustainability in a 6G Network

Zhou Zhou, Mohammad Shojafar, *Senior Member, IEEE*, Jemal Abawajy, *Senior Member, IEEE* and Ali Kashif Bashir *Senior Member, IEEE*

Abstract—Differential evolution (DE) algorithm is utilized to find an optimized solution in real multidimensional applications like 5G/6G networked devices and support unlimited connectivity for terrestrial networks due to high efficiency, robustness, and easy achievements. With the development of new emerging networks and the rise of big data, the DE algorithm encounters a series of challenges, such as the slow convergence rate in late iteration, strong parameter dependence, and easiness of falling into local optimum. These issues exponentially increase the energy and power consumption of communications and computing technologies in 5G/6G networks like a networked data center. To address this and leverage a practical solution, this paper introduces *IADE*, an improved adaptive DE algorithm, to solve the problems mentioned earlier. IADE improves the scaling factor, crossover probability, variation, and selection strategy of the DE algorithm. In IADE, the parameters adaptively adjusted with the population's iterative evolution to meet the parameter's different requirements values of network steering traffic in each period. Numerous experiments are carried out through the benchmark function to evaluate the performance of IADE, and the results obtained from the experiment illustrate that IADE surpasses the benchmark algorithms in terms of solution accuracy and convergence speed for large tasks around 10%, respectively.

Index Terms—6th Generation (6G), Intelligent cloud, intelligent load balancing, network resource optimization, networked data center.

1 Introduction

With the rapid development of heterogeneous devices, intelligent terminals and infrastructures to cover diversified applications and their demands, current 4G and 5G networks lack real-time monitoring of the quickly rising traffic demands. Both industry and academia have started envisioning the new generation called 6G to enable the real-time Artificial Intelligence (AI) approaches to speed up the design and optimization of 6G networks with a high level of intelligence [1]. To provide a green 6G, we require to apply an AI mechanism to atomize the network models and achieve higher efficiency, lower delay, and minimize the cost of network transmission. At this time, the emergence of edge computing brings a solution to the efficiency problem of the cloud computing system [2]. Hence, edge nodes are the devices that relocated very close to the devices/users. Hence, these decisions can reduce the consumption of data transmission to the remote cloud data center [3]. Edge cloud requires appropriate AI techniques to ensure the Quality-of-Service (QoS) requirements of the 6G system and optimize the resource allocation required per demands by reducing node energy consumption and achieve efficient processing per task. To support such network evolution, selecting the appropriate AI mechanism to stimulate the infrastructure designers to tackle the energy efficiency challenges in the future green 6G. Differential evolution (DE) [4], [5] algorithm is a fast convergence AI method that could find the global minimum of problem space using a few control parameters and would be a colorful option for a volatile environment like 5G and 6G. DE is swarm intelligence optimization algorithm that simulates the natural evolution rule. It optimizes the evolution direction and searches ranges through individual cooperation and competition within the population. The authors in [4]–[6] initiated it to solve the Chebyshev polynomial problem defined by recursive sequences of the orthogonal polynomial. Later, the DE algorithm is selected to address the network resource optimization problem [7], [8]. Similar to all evolutionary algorithms, the DE algorithm has wide applications in 5G/6G networked devices [9], [10] and supports unlimited connectivity for terrestrial networks due to high efficiency and robustness [11], [12]. Under the same precision requirements, in contrast to other evolutionary algorithms, the DE algorithm possesses fast convergence speed [13], [14]. Currently, the exploration of the DE algorithm has become a research hotspot [15], [16].

1.1 Motivations

To provide a green 6G, we require to apply an AI mechanism to atomize the network models and achieve higher efficiency, lower delay, and minimize the cost of network transmission. At this time, the emergence of edge computing brings a solution to the efficiency problem of the cloud

- *Z. Zhou is with the Department of Computer Engineering and Applied Mathematics at Changsha University, China and with the College of Information Science and Engineering, Hunan University, Changsha, China Email: zhouzhou03201@126.com*
- *M. Shojafar is with the 5GIC & 6GIC, Institute for Communication Systems (ICS), University of Surrey, Guildford, GU27XH, United Kingdom E-mail: m.shojafar@surrey.ac.uk*
- *J. Abawajy is with the Faculty of Science, Engineering and Built Environment, Deakin University, Geelong, Australia Email: jemal.abawajy@deakin.edu.au*
- *A.K. Bashir is with the Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, UK and with School of Electrical Engineering and Computer Science, National University of Science and Technology (NUST), Islamabad, Pakistan and with the School of Information and Communication Engineering, University of Electronics Science and Technology of China (UESTC), China Email: dr.alikashif.b@ieee.org*

Corresponding Author: Mohammad Shojafar

computing system [2]. Hence, edge nodes are the devices that relocated very close to the devices/users. Hence, these decisions can reduce the consumption of data transmission to the remote cloud data center [3]. Edge nodes can equipped with AI mechanisms to enhance the QoS of the demands and optimize the resource allocations. However, selecting appropriate AI method helps us to tackle a series of challenges, such as the slow convergence rate in late iteration, strong parameter dependence, and easiness to fall into local optimum. These issues exponentially increase the energy and power consumption of communications and computing technologies in 5G/6G network like a networked data center. To address such issues, we propose an improved adaptive DE algorithm named *IADE*. The proposed IADE is fast converge and can be applied in large scale network systems which is essential in 6G environment.

1.2 The goal of the paper and contributions

Our goal is to design a novel green solution optimizes the consumption of power of resource and consumption of network resources in a data center associated in 6G environment. The major contributions of this work are as follows:

- We design a 6G network architecture reflecting the data traffic communication between the IoT and cloud infrastructure.
- We design an improved adaptive DE algorithm named IADE by enhancing the mutation factor F, crossover factor CR, and the standard DE algorithm's selection strategies.
- We assess the proposed IADE algorithm against the state-of-the-art through testing on thirty classical benchmark functions.
- We analyze the application of the proposed algorithm in intelligent IoT data traversed through a cloud.

The rest of the paper is as follows. We summarize the related work in Section 2. In Section 3, we put forward the IADE algorithm to preserve sustainability in 6G Networks, and the experimental results and evaluation are analyzed in Section 4. Section 5 explores some real applications can be tuned based on IADE. Section 6 concludes the paper and provide some outlooks.

2 Related works

With the development of new emerging networks and the rise of big data, the DE algorithm encounters a series of challenges, such as the slow convergence rate of the DE algorithm in late iteration, strong parameter dependence, and easiness to fall into local optimum. Researchers adopt strategies to solve the above problems, which can be divided into *three* perspectives, namely, *parameter optimization*, *evolution strategy improvement*, and *hybrid algorithm researches*.

(1) Parameter optimization [17]–[21]: Improved parameters are involved in this strategy, including population size, crossover factor, and mutation factor. The main purpose is to accelerate the algorithm performance by improving the manner of assuming the value of parameters. Improved approaches include setting fixed parameters, random parameters, and adaptive parameters [17]. The fixed parameter value is set to the empirical value and remains unchanged throughout the evolution of the population. The parameters in the standard DE algorithm belong to this category. The parameters need to be continuously adjusted to obtain better algorithm performance. Setting the random parameters and self-adaptive parameters can avoid the uncertainty of artificial control parameters for the purpose of improving the algorithm search capability and convergence speed. Linear functions with random probabilities are required in setting a way to assume the random value of parameters [18]. The self-adaptive setting of parameter value [19] refers to the self-adjustment of parameter value through feedback according to evolutionary generations or fitness value as well as the individual crowd. For example, Wang et al. [20] initiated the DE by using the combination of composite trial vector and mutation policy [21].

(2) Evolution strategy improvement [22]–[26]: Regarding the evolution strategy of the DE algorithm, it includes a series of operations such as selection strategy, crossover strategy, and mutation policy. To some extent, in terms of search capability and convergence speed, the evolution strategy of the DE algorithm plays a decisive role. Main improvement measures of such methods include the initiation of a new evolution strategy. For example, to handle the problem of multi-objective optimization, Ali et al. [25] initiated an improved optimization method. Qiu et al. [26] also put forward a minimax DE algorithm that pursues for the optimized time and space cost.

(3) Hybrid algorithm research [27]–[29]: In research, combining the DE algorithm with other algorithms is feasible to improve the algorithm optimization capability. Hybrid algorithm research belongs to algorithm refactoring, which enables algorithm limitations in some perspectives to be improved through other methods. For example, the DE algorithm and other algorithms can be used in different stages of searching for the best optimization process to display their respective advantages so that new methods can be constructed. Li et al. [27] combined the genetic algorithm with the DE algorithm to deal with the problem of multi-objective optimization in a sensor network. Li et al. [28] also proposed a new algorithm combining the DE algorithm with a PSO algorithm to address the problem of resource assignment in a wireless sensor network. Recently, the authors in [30] introduce a joint traffic-aware approach addressing the a priority and power of Virtual machine placement in a cloud data centers.

In addition to the above three types of algorithms, some recent algorithms such as [31]–[36] also show excellent performance in the process of finding solutions. For example, in Ref. [31], a single-objective real-parameter optimization method is proposed and the experimental results illustrate its effectiveness. in Ref. [32], [35], two adaptive DE algorithms have been put forward to solve the global optimization problems and the proposed algorithms are highly competitive.

Comparison vs. IADE: The DE algorithm explained earlier facing some problems such as slow convergence rate in late iteration, strong parameter dependence, and easiness to fall into local optimum. To overcome these problems, we introduce IADE which is a fast converge AI approach and can deal with large scale network systems which is essential

TABLE 1: Comparisons between different DE approaches to cover the sustainability where ○:= the method does not support the property, and •:= the method supports the property. SF:=Scale Factor; SS:= Selection Strategy; MP:= Mutation Policy.

Ref.	Type	SF	Population	Crossover	SS	MP
[18]	(1)	○	•	•	•	○
[19]	(1)	○	•	•	•	○
[20]	(1)	○	•	•	•	○
[25]	(2)	○	○	•	•	•
[26]	(2)	○	○	•	•	•
[27]	(3)	○	•	•	•	•
[28]	(3)	○	•	•	•	•
[22]	(3)	○	•	•	•	•
IADE	(3)	•	•	•	•	•

in 6G environment. Table 1 summarises the various DE methods comparisons and describes IADE characteristics compared to other DE methods.

3 IADE: The Proposed Algorithm

This section illustrates the proposed 6G-enabled network instantiated on the networked cloud computing (section 3.1) and then delineate the proposed IADE steps (sections 3.2-3.5).

3.1 The Considered Architecture

The 6G network is an organic unit that requires extensive connectivity and provides diverse quality of service requirements. Fig. 1 shows the architecture of the 6G distributed networked cloud computing.

In this figure, the architecture includes ground wireless access composed of a mass ground access point/distributed antenna, space-based satellite access composed of different constellation satellites, space-based wireless access composed of low-altitude aircraft or airship, and sea-based wireless access composed of ships and ships. Al of the heterogeneous networks are managing through the 5G core (see the colorful box inside the Fig. 1). Other access modes can realize heterogeneous fusion network function management and resource cooperative management through an integrated core network. Also, the center of the cloud is an essential part of the 6G network like [37]. It is responsible for assigning tasks to computing resources to meet the diverse quality of service requirements. However, how to provide differentiated and intelligent network services for diversified 6G business scenarios? To address this issue, we raise the IADE algorithm to optimize the allocation of processing resources in the networked data center to improve network resource utilization and reduce the energy consumption cost. The main idea of the IADE algorithm includes improvements of mutation factor, crossover factor, mutation strategy, and selection strategy.

3.2 Improvements of Mutation Factor

The mutation factor F of the DE algorithm can mitigate the search range and population variety. The DE algorithm is highly sensitive to the setting of the mutation factor F, which is generally valued at the reals between [0, 2] to control the amplitude of differential items [38]. An inappropriate setting F can result in no production of new mutation individuals in the mutation stage, increased and gradually the same individual similarity, as well as the influence of convergence speed. When F set at a large value, the differences of individuals are increased, which will also improve the search scope of the algorithm. This process is close to the global stochastic search, which leads to the slow convergence of the algorithm, low exploration efficiency, and low quality of the obtained solution. When F set at a low value, the search scope of the algorithm becomes small, whereas the local research is enhanced. Although the convergence speed is accelerating, the algorithm will be of the local optimization solution. Therefore, a reasonable set of the mutation factor F is essential for the quality of the best optimization solution to the algorithm. In the standard DE algorithm, F set as a fixed constant and remains constant throughout the iterative solution process, which will result in an algorithm that is incapable of meeting the requirements of the parameters at each period in the evolution process. Therefore, the algorithm easily falls into the optimal local value so that global optimization cannot be obtained. To solve the shortcomings of the DE algorithm in this perspective, the value of F will be adaptively set in this paper, that is, the value of F automatically set along with the evolution process. When the algorithm is initially iterated, F is set as a large value, which allows the algorithm to possess a large search space to guarantee population diversity. In the later stage, the value of F is continuously reduced as the evolutionary algebra continues to increase. At such time, the algorithm can perform local searches in a small scope where a feasible solution is available, thereby avoiding destroying the optimal solution and improving search precision. Mutation factor F value can be described in Eq. (1) and Eq. (2).

$$w(t) = \frac{\cos(\frac{|t-T|}{T}\pi) + 1}{2} \tag{1}$$

$$F(t) = F_{\max} - (F_{\max} - F_{\min}) * w(t) \tag{2}$$

where parameter t is an evolutionary algebra, and T is the largest evolutionary algebra of the population. The $w(t)$ is the weighting factor within the range of [0.4, 1]. In this paper, the maximum value of the mutation factor is $F_{\max} = 1$, whereas the minimum value of the mutation factor is $F_{\min} = 0.4$.

3.3 Improvements of Crossover Factor

Crossover operations of the DE algorithm could effectively improve the variety of individuals in a population. Crossover factor *CR* in the crossover strategy determines children and parents, and the extent of the dimensional component exchanged between the crossover individuals. A high *CR* indicates a high contribution of the mutation variable. In each dimension component of the crossover vector, the high proportion of each dimension component that belongs to the mutation vector can reduce the diversity of the population and be beneficial to the algorithm to perform a local search to accelerate the convergence rate. A low *CR* value is beneficial in maintaining the population variety and global research capability. In the standard DE

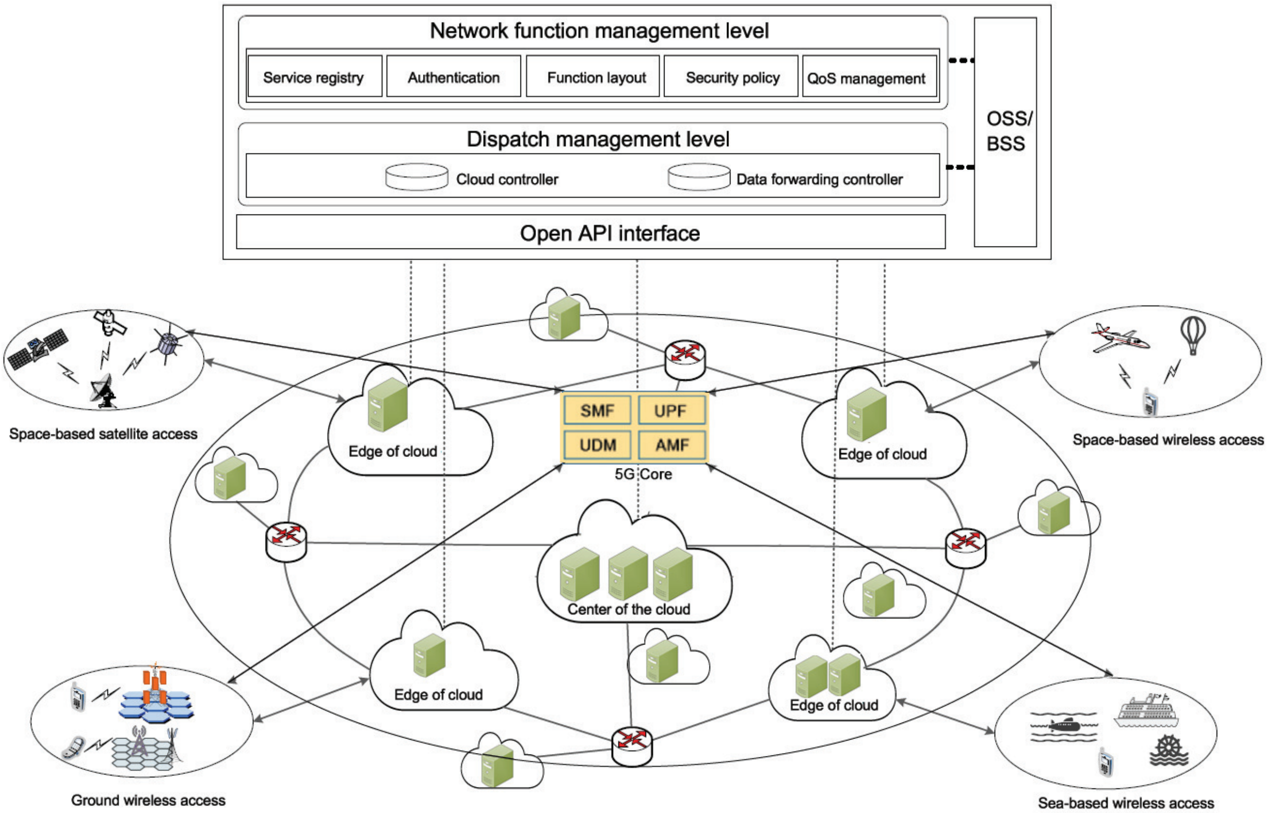


Fig. 1: Architecture of the 6G distributed networked cloud computing. OSS/BSS:= operations/business support system; SMF:= session management function; UPF:= user plane functions; UDM:= unified data management; AMF:= Core access and mobility management function.

algorithm, crossover factor *CR* is the fixed-parameter, which cannot satisfy the requirements of each stage in the evolution process to *CR* values. Therefore, the value of crossover factor *CR* needs to be constantly changed in the evolution process according to the operation. The value of CR can be depicted as,

$$CR(t) = CR_{\min} + (CR_{\max} - CR_{\min}) * w(t) \tag{3}$$

Similar to mutation factor F, the value of *CR* needs to be set according to the current evolutional generation t. With the increase of evolutional generation t and the value of weight factor , *CR* value also tends to be increased, which can accelerate the convergence speed of fitness value at the later stage of evolution and guarantee the convergence accuracy at the same time. The $w(t)$ is the weighting factor within the range of [0.4, 1]. In this paper, the minimum value of the crossover factor is $CR_{\min} = 0.6$, and the maximum value of the crossover factor is $CR_{\max} = 0.9$.

3.4 Improvements of Mutation Strategy

In the DE algorithm, a single mutation pattern is often difficult to meet the needs of population evolution. Therefore, a combination of various strategies can be employed to integrate respective advantages. For example, DE/rand/1 mode can guarantee population variety with great global search capability, which is not easy to achieve in the local optimum with slow convergence speed. However, the DE/best/1 mode offers faster convergence speed than the DE/rand/1 mode. The improved mutation strategy is achieved by combining these two mutation models. The resulting strategy can simultaneously take search ability and convergence into consideration. The improved mutation pattern is:

$$u = 1 - (\frac{t}{T})^2 \tag{4}$$

$$v_i(t) = u*x_{r1}(t)+(1-u)*x_{best}(t)+F*(x_{r2}(t)-x_{r3}(t)) \tag{5}$$

where parameter t means the evolutional generation, T refers to the largest evolutionary generation of the population and x_i represents the current individual. x_{r1}, x_{r2} and x_{r3} represent three randomly chosen individuals varying from x_i , and x_{best} represents the best individual in the contemporary population. Parameter u is varied along with the iteration times t, and its image is a parabola in the interval [0, 1], in which the value grows slowly in the early period so that the value of u also decreases slowly. In the early population evolution $u \to 1$, the mutation pattern can be mostly inclined to DE/rand/1, in which the algorithm has strong global search performance so that individual population diversity can be guaranteed. In the later $u \to 0$, the mutation strategy is inclined to DE/best/1, in which the algorithm could achieve improved balance among the stability, search capability, and optimization speed to avoid slow convergence speed in the early algorithm period. The improved algorithm can adaptively select different mutation strategies at different stages of the population evolution process so that search capability and convergence speed are balanced.

Eq. (5) is further improved according to the mutation strategy to better control the evolution direction of the population in the later stage. An "optimal difference value" is added on the basis of the original, and the weighting factors are added to control the respective contribution rates of "random difference" and "optimal difference". Improved mutation strategy is defined in Eq. (6) and Eq. (7).

$$\lambda = 1 - \sqrt{\frac{t}{T}} \tag{6}$$

$$\begin{aligned} v_i(T) =& u * x_{r1}(T) + (1-u) * x_{best}(T) \\ & + F * [\lambda(x_{r2}(T) - x_{r3}(T)) \\ & + (1-\lambda)(x_{best}(G) - x_{r4}(T))] \end{aligned} \tag{7}$$

Where parameter u is varied along with the iteration time t, and its image is parabola in the interval [0, 1]. Parameter x_{best} is the best individual in the contemporary population, λ is a weighting factor for controlling the disturbance of the mutated individual in 'random difference' and 'optimal difference'. The value of λ ranges between [0,1], which slowly increases with the evolutionary generation of the population, thereby giving the weight more importance in the early evolution period of 'random difference'. It (refer to λ) contributes the most to the variation vector. The algorithm can strengthen the global search scope. In the later stages of evolution, individuals in the population will converge, thereby reducing the potential range of the 'optimal solution' and speeding up the search for optimal solutions.

3.5 Improvements of Selection Strategy

In the selection operation steps of the standard DE algorithm, the two individuals for comparing are the crossover individual produced by the target individual and crossover operations. The crossover operations further adjust each dimensional component of individual vector quantity obtained by the mutation operations so that individuals become increasingly varied, and the algorithm can search for the optimal solution in a wide range. However, the mutation operation steps may also produce a global optimization solution. As the mutation operation can directly lead to the crossover operations of mutation individuals, the local component of the mutation vector is substituted by crossover operations, which will destroy the optimized solution.

In addition, in the crossover operations of the standard DE algorithm, the individual crossover vector is composed by selecting part of the parent individual or mutation individual according to the crossover probability *CR*. Another different crossover individual vector can be constituted through reverse selection according to the crossover probability *CR*, as shown in the emphasized part of Fig. 2. However, only one crossover individual vector used in the standard DE algorithm among all the produced vectors could result in the loss of the individual vector of the optimized solution.

Therefore, the algorithm can obtain the current optimized solution as much as possible to avoid destroying the potential optimal individual in the population. The

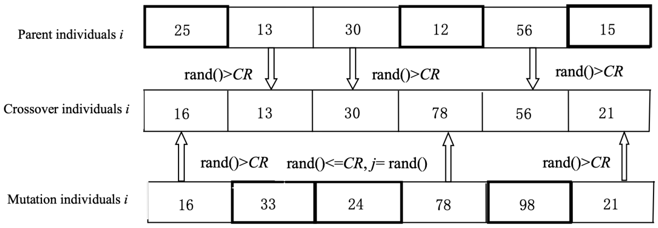


Fig. 2: Reverse crossover individual vector

Algorithm 1 IADE Algorithm

Input: Initial fitness function f^0, T, NP, D, $Fmax$, $Fmin$, $CRmax$, $CRmin$
Output: Best fitness values f^*
  NP: population size
  T: largest evolutionary generation of NP
  D: population individual
$g \leftarrow 1$ // Initialize the value of g
for $g \leq |G|$ **do**
  Calculate F using Eq. (2) //Dynamically obtain the mutation factor
  Calculate CR using Eq. (3) //Dynamically gain the crossover factor
  for $x_i \in NP$ **do**
    Calculate $v_i(T)$ using Eq. (7)
  end for
  for $x_i \in NP$ **do**
  $CI(i)$: ith crossover individual
  $RCI(i)$: reverse ith crossover individual
    $CI(i) \leftarrow CR(i)$
    $RCI(i) \leftarrow CR(i)^T$
  end for
  for $x_i \in NP$ **do**
  select the individual with the minimum function value to the next generation
    Calculate $f(x_i)$ using Eq. (9) // Obtain the value of $f(x_i)$
    $f^* \leftarrow f(x_i)$
  end for
end for
$g \leftarrow g + 1$
return f^*

---

standard DE algorithm is improved. Selection operation steps of improved DE algorithm aim to select the individual with optimal 'fitness value' into the next generation from mutation vector v_i, crossover vector u_i, reverse crossover vector h_i, and target vector x_i. Eq. (8) shows the improved selection strategy:

$$x_i^{t+1} = \begin{cases} u_i^{t+1}, \min\left(f\left(u_i^{t+1}\right), f\left(v_i^{t+1}\right), f\left(h_i^{t+1}\right), f\left(x_i^t\right)\right) = f\left(u_i^{t+1}\right) \\ v_i^{t+1}, \min\left(f\left(u_i^{t+1}\right), f\left(v_i^{t+1}\right), f\left(h_i^{t+1}\right), f\left(x_i^t\right)\right) = f\left(v_i^{t+1}\right) \\ h_i^{t+1}, \min\left(f\left(u_i^{t+1}\right), f\left(v_i^{t+1}\right), f\left(h_i^{t+1}\right), f\left(x_i^t\right)\right) = f\left(h_i^{t+1}\right) \\ x_i^t, \min\left(f\left(u_i^{t+1}\right), f\left(v_i^{t+1}\right), f\left(h_i^{t+1}\right), f\left(x_i^t\right)\right) = f\left(x_i^t\right) \end{cases} \tag{8}$$

In the selection operation steps of the standard DE algorithm, when the fitness value of the crossover individual is higher than that of the individual parent, the individual parent will be maintained. When the fitness value of mutation vector v_i, crossover vector u_i and reverse crossover vector h_i is higher than that of the parent target vector x_i, the average value of the optimal individual and median individual is regarded as the value of the individual in the next generation so as to accelerate the convergence speed and population variety of algorithm. The selection strategy in Eq. (8) is further improved, and the final selection strategy formula is as follows:

$$x_i^{t+1}=\begin{cases} u_i^{t+1}, \min\left(f\left(u_i^{t+1}\right), f\left(v_i^{t+1}\right), f\left(h_i^{t+1}\right), f\left(x_i^{t}\right)\right)=f\left(u_i^{t+1}\right) \\ v_i^{t+1}, \min\left(f\left(u_i^{t+1}\right), f\left(v_i^{t+1}\right), f\left(h_i^{t+1}\right), f\left(x_i^{t}\right)\right)=f\left(v_i^{t+1}\right) \\ h_i^{t+1}, \min\left(f\left(u_i^{t+1}\right), f\left(v_i^{t+1}\right), f\left(h_i^{t+1}\right), f\left(x_i^{t}\right)\right)=f\left(h_i^{t+1}\right) \\ \left(x_{\text{best}}+x_{avg}\right)/2, \min\left(f\left(u_i^{t+1}\right), f\left(v_i^{t+1}\right), f\left(h_i^{t+1}\right), f\left(x_i^{t}\right)\right)=f\left(x_i^{t}\right) \end{cases} \tag{9}$$

In Eq.(9), $x_{best,j}$ represents the individual with the optimal fitness value in the current population, and x_{avg} refers to the median value of all the individuals. The calculation formula of x_{avg} individual vector $x_{avg,j}$ is as follows:

$$x_{avg,j}=\frac{1}{NP}*\sum_{i=1}^{NP} x_{i,j} \tag{10}$$

Fig. 3 shows the selection mechanism of IADE algorithm.

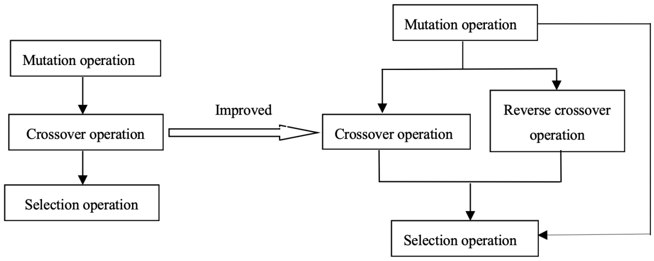


Fig. 3: Selection mechanism of IADE algorithm

The pseudo-code of IADE algorithm is shown below, it mainly involves the mutation factor, crossover factor, mutation, and selection strategy. The specific introduction of the mutation factor, crossover factor, mutation, and selection strategy are listed in Sections 3.2, 3.3, 3.4, 3.5, respectively.

Fig. 4 indicates the working process of the t-th iteration of the population in the IADE algorithm.

The complexity of IADE: Based on the main idea of IADE (refer to Algorithm1), the first big *for* loop (lines 2-16) includes three smaller for loops with the size NP, where refers to the size of population. Thus, the computational complexity of each loop is of in the order of NP, $O(NP)$. Thus, the total time complexity of IADE is characterized by $O(G_m \times NP)$, where parameter G_m corresponds to the number of iterations.

4 Performance Evaluations

In this part, we explain the simulation settings/setup and the results. In this way, first we set up the simulation settings (see Section 4.1). Then, we present the results which are separated into two parts: First, we describe the systematic analysis (see Section 4.2). Then, we compare our algorithm against the state-of-the-art (see Section 4.3).

4.1 Simulation Setup

To make a comparison and evaluation for the proposed IADE algorithm, thirty benchmark functions [39] are selected for simulation experiments. Moreover, to embody the performance of the IADE algorithm on the objective function, the IADE algorithm and the other three DE algorithms with different mutation strategies are compared through experiments. In addition, the winners [40]–[42] of CEC2017

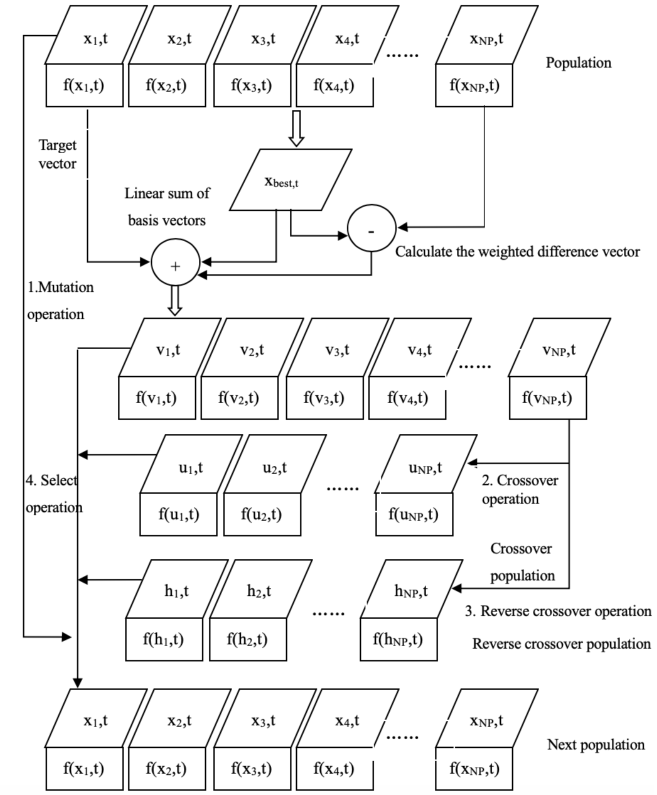


Fig. 4: Working process of the t-th iteration of the population in IADE algorithm

unconstrained optimization competition and recent algorithm [43], [44] also have added to make a comparison. The DE algorithms with different mutation policies are: $\mathrm{DE/rand/1}$, $\mathrm{DE/best/1}$ and $\mathrm{DE/rand-to-best/2}$. The MATLAB program implementation of IADE is available in [45].

TABLE 2: Environmental parameter

Name	Values
Operation system	Windows 7 64bit
RAM	4G
ICPU	2.5GHz
Disk	500G
Test tool	MATLAB2016a

Tables 2 and 3 show the IADE, $\mathrm{DE/rand/1}$, $\mathrm{DE/best/1}$ and $\mathrm{DE/rand-}$ to $\mathrm{-best/2}$ algorithm parameter settings. To

TABLE 3: Parameter value of the algorithms

Algorithm	NP	F	CR
DE/rand/1	100	0.5	0.9
DE/best/1	100	0.5	0.9
DE/rand - to - best/2	100	0.5	0.9
IADE	100	[0.4,1]	[0.3,0.9]

ensure fairness in the performance test of the algorithm and to remove random error in the experiment, each algorithm of each test function is run 51 times separately. The mean function values of the 51 running statistics are taken as the final result and are included in the comparison of other algorithms.

TABLE 4: Result of the IADE algorithms for D=10 and D=30

	D=10					D=30				
Func	Mean	Best	Worst	Median	Std	Mean	Best	Worst	Median	Std
F1	8.54E+02	1.00E+02	3.12E+03	1.00E+02	1.51E+03	8.49E+03	2.52E+02	1.62E+04	6.86E+03	6.38E+03
F2	/	/	/	/	/	/	/	/	/	/
F3	3.00E+02	3.00E+02	3.00E+02	3.00E+02	8.90E-10	3.00E+02	3.00E+02	3.02E+02	3.00E+02	7.95E-01
F4	3.51E+03	3.51E+03	3.51E+03	3.51E+03	4.55E-13	1.94E+04	1.94E+04	1.94E+04	1.94E+04	0.00E+00
F5	6.18E+02	6.14E+02	6.22E+02	6.17E+02	3.88E+00	7.93E+02	7.80E+02	8.18E+02	7.92E+02	1.47E+01
F6	6.43E+02	6.38E+02	6.52E+02	6.40E+02	6.73E+00	6.55E+02	6.49E+02	6.62E+02	6.54E+02	4.63E+00
F7	7.38E+02	7.23E+02	7.57E+02	7.36E+02	1.53E+01	9.04E+02	8.35E+02	1.00E+03	8.67E+02	7.38E+01
F8	8.13E+02	8.10E+02	8.17E+02	8.13E+02	4.02E+00	9.24E+02	8.95E+02	9.63E+02	9.10E+02	3.24E+01
F9	1.54E+03	1.46E+03	1.58E+03	1.56E+03	5.51E+01	5.44E+03	4.94E+03	6.02E+03	5.26E+03	4.41E+02
F10	1.77E+03	1.40E+03	2.03E+03	1.82E+03	2.69E+02	4.75E+03	3.65E+03	6.04E+03	4.51E+03	1.12E+03
F11	1.12E+03	1.11E+03	1.12E+03	1.12E+03	6.18E+00	1.22E+03	1.16E+03	1.28E+03	1.23E+03	4.95E+01
F12	8.05E+03	1.33E+03	2.74E+04	1.74E+03	1.29E+04	5.57E+04	1.33E+04	1.40E+05	4.42E+04	4.87E+04
F13	4.96E+03	1.31E+03	1.59E+04	1.31E+03	7.30E+03	1.57E+04	2.65E+03	4.69E+04	8.94E+03	1.84E+04
F14	1.42E+03	1.40E+03	1.44E+03	1.41E+03	2.02E+01	1.64E+03	1.54E+03	1.77E+03	1.62E+03	8.18E+01
F15	1.51E+03	1.50E+03	1.52E+03	1.51E+03	6.90E+00	6.84E+03	1.63E+03	2.48E+04	1.88E+03	1.01E+04
F16	1.70E+03	1.60E+03	1.84E+03	1.67E+03	1.17E+02	2.37E+03	2.10E+03	2.56E+03	2.51E+03	2.31E+02
F17	1.76E+03	1.73E+03	1.85E+03	1.73E+03	6.04E+01	2.08E+03	2.00E+03	2.27E+03	2.05E+03	1.11E+02
F18	1.83E+03	1.82E+03	1.84E+03	1.82E+03	9.11E+00	1.47E+04	2.45E+03	4.85E+04	6.48E+03	1.93E+04
F19	3.45E+03	3.44E+03	3.46E+03	3.45E+03	8.35E+00	7.20E+03	2.42E+03	1.22E+04	7.71E+03	4.67E+03
F20	2.06E+03	2.03E+03	2.14E+03	2.05E+03	5.02E+01	2.53E+03	2.31E+03	2.78E+03	2.44E+03	2.27E+02
F21	2.23E+03	2.20E+03	2.34E+03	2.20E+03	6.78E+01	2.38E+03	2.35E+03	2.42E+03	2.37E+03	2.88E+01
F22	2.30E+03	2.30E+03	2.30E+03	2.30E+03	1.00E+00	4.08E+03	2.30E+03	7.45E+03	2.30E+03	2.49E+03
F23	2.62E+03	2.61E+03	2.64E+03	2.62E+03	1.43E+01	2.73E+03	2.71E+03	2.76E+03	2.73E+03	2.00E+01
F24	2.75E+03	2.74E+03	2.76E+03	2.74E+03	1.04E+01	2.94E+03	2.88E+03	3.00E+03	2.95E+03	5.02E+01
F25	2.91E+03	2.90E+03	2.95E+03	2.90E+03	2.35E+01	2.90E+03	2.89E+03	2.91E+03	2.90E+03	9.48E+00
F26	2.94E+03	2.90E+03	3.00E+03	2.92E+03	4.97E+01	5.56E+03	4.92E+03	6.37E+03	5.29E+03	6.13E+02
F27	3.10E+03	3.09E+03	3.10E+03	3.09E+03	5.62E+00	3.24E+03	3.22E+03	3.25E+03	3.25E+03	1.34E+01
F28	3.17E+03	3.17E+03	3.17E+03	3.17E+03	6.95E-13	3.54E+03	3.54E+03	3.54E+03	3.54E+03	1.51E-12
F29	3.68E+03	3.65E+03	3.73E+03	3.68E+03	3.40E+01	7.34E+03	7.08E+03	7.81E+03	7.22E+03	2.95E+02
F30	5.08E+07	5.08E+07	5.08E+07	5.08E+07	6.10E-08	2.65E+09	2.65E+09	2.65E+09	2.65E+09	4.42E-06

4.1.1 Benchmark Test Functions

We evaluate the performance of the proposed algorithm on the benchmark CEC 2017 [39], [46], [47]. Its mainly includes 30 functions (denoted by F1-F30). The F1-F3 belong to "unimodal functions"; F4-F10 is "simple multimodal functions"; F11-F20 is "hybrid functions"; F21-F30 is "composition functions". Each function needs to be run 51 times. For the 30 functions, the search range is $[-100, 100]^D$, D is the dimensions. In this paper, D={10, 30, 50, 100}.

4.2 IADE Systematic Analysis

The first experiment is to test the performance of the IADE in terms of numerical results (systematic analysis). In this experiment, we choose the benchmark CEC 2017 including 30 function with the dimensions 10, 30, 50, and 100. The numerical results are depicted in Table 4 and Table 5, respectively.

Table 4 illustrates the performance of the IADE algorithm when D=10 and D=30, while Table 5 shows the performance of the IADE algorithm when D=50 and D=100 with running 51 times for the 30 functions. In above two Tables, the "Mean" represents the average value for the 51 independent running results, the "Best" means the optimal value for the 51 independent running results, the "Worst" is the worst value, the "Medium" is the middle value, and "Std" corresponds to standard variance. As the running result of F2 function from the CEC 2017 is unstable, we use the symbol "/" instead. Table 4 and Table 5 indicate that the the "Mean" and "Medium" value increases with the growth of the dimension (for example, when the dimension from D=10 to D=100), the reason is that the number of dimensions growths, the difficulty of finding the solution to the problem increases and the time taken growths.

4.2.1 Convergence analysis of the IADE algorithm on different functions

This section will test the convergence of the IADE algorithm on different functions. We still use CEC 2017 benchmark set. It includes 30 functions and all of the 30 functions can be divided into four types that is "unimodal functions" (F1-F3 function), "simple multimodal functions" (F4-F10), "hybrid functions" (F11-F20), and "composition functions" (F21-F30). To test the performance of the IADE algorithm, we randomly choose one function from each of the four types of functions, that is F3, F8, F11, and F21 functions. Fig. 5 and Fig. 6 the convergence of the four functions on dimension D=50 and D=100, respectively.

On the whole, Fig. 5 and Fig. 6 show that the IADE algorithm has a better performance than DE/best/1 algorithm in most cases. The solution performance of the IADE algorithm is considerably better than that of the $\mathrm{DE/rand/1}$ and $\mathrm{DE/rand-}$ to-best/2 algorithms, and has great advantages in the convergence speed and solution accuracy of the function values.

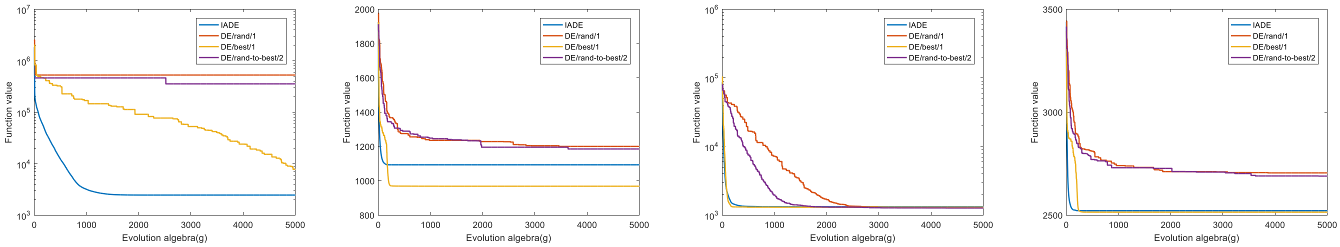
Experimental results display that the improved differential evolution algorithm IADE offers better solution accuracy and convergence speed than DE/best/1, DE/rand/1 and DE/rand-to-best/2 algorithms.

4.3 Comparison with the state-of-the-art algorithms and statistical analysis

In this section, we compare IADE against the conventional DE and also with state-of-the-art algorithms in-

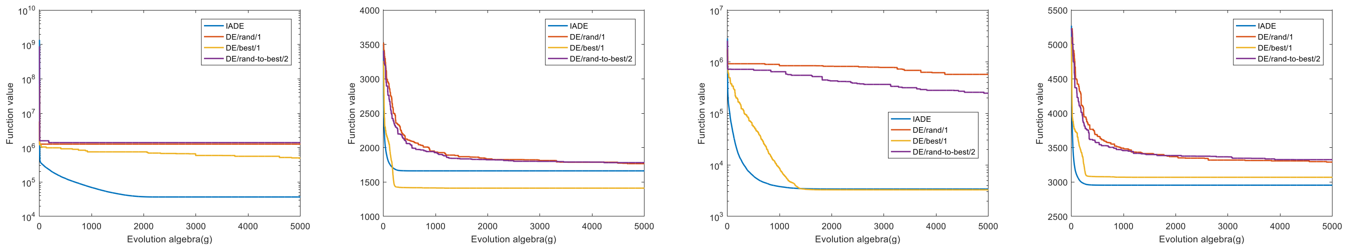
TABLE 5: Result of the IADE algorithms under the D=50 and D=100

	D=50					D=100				
Func	Mean	Best	Worst	Median	Std	Mean	Best	Worst	Median	Std
F1	3.25E+03	1.06E+02	1.24E+04	8.80E+02	4.46E+03	9.46E+04	3.69E+02	5.26E+05	8.87E+03	1.85E+05
F2	/	/	/	/	/	/	/	/	/	/
F3	2.46E+03	3.09E+02	1.11E+04	1.57E+03	3.21E+03	3.68E+04	2.43E+04	4.96E+04	3.61E+04	8.20E+03
F4	3.12E+04	3.12E+04	3.12E+04	3.12E+04	1.24E-11	7.98E+04	7.98E+04	7.98E+04	7.98E+04	1.23E-10
F5	8.88E+02	8.54E+02	9.76E+02	8.81E+02	3.44E+01	1.41E+03	1.36E+03	1.47E+03	1.40E+03	3.44E+01
F6	6.62E+02	6.56E+02	6.69E+02	6.61E+02	3.92E+00	6.62E+02	6.59E+02	6.63E+02	6.63E+02	1.18E+00
F7	1.23E+03	1.13E+03	1.39E+03	1.23E+03	7.64E+01	2.59E+03	2.10E+03	2.97E+03	2.59E+03	2.92E+02
F8	1.09E+03	1.03E+03	1.18E+03	1.08E+03	4.66E+01	1.66E+03	1.41E+03	1.87E+03	1.67E+03	1.28E+02
F9	1.31E+04	1.24E+04	1.40E+04	1.32E+04	5.76E+02	2.39E+04	2.31E+04	2.52E+04	2.38E+04	7.15E+02
F10	7.98E+03	6.04E+03	9.70E+03	7.95E+03	1.36E+03	1.57E+04	1.36E+04	1.75E+04	1.60E+04	1.19E+03
F11	1.32E+03	1.24E+03	1.42E+03	1.34E+03	6.22E+01	3.40E+03	2.47E+03	6.49E+03	3.14E+03	1.16E+03
F12	9.99E+05	4.13E+05	2.02E+06	8.33E+05	5.62E+05	1.07E+07	5.01E+06	1.39E+07	1.17E+07	2.86E+06
F13	8.85E+03	2.05E+03	3.03E+04	5.93E+03	8.49E+03	1.02E+04	4.88E+03	1.56E+04	1.09E+04	3.42E+03
F14	5.27E+03	1.93E+03	9.15E+03	4.59E+03	2.38E+03	2.31E+05	7.64E+04	4.85E+05	1.84E+05	1.40E+05
F15	1.00E+04	1.82E+03	3.30E+04	6.38E+03	9.72E+03	4.79E+03	1.92E+03	8.26E+03	4.11E+03	2.66E+03
F16	3.07E+03	2.48E+03	3.75E+03	2.97E+03	4.27E+02	5.59E+03	4.53E+03	6.67E+03	5.56E+03	6.96E+02
F17	3.27E+03	2.58E+03	3.64E+03	3.35E+03	3.10E+02	4.86E+03	3.69E+03	5.94E+03	4.85E+03	7.32E+02
F18	1.12E+05	3.78E+04	3.33E+05	7.83E+04	8.98E+04	5.51E+05	2.03E+05	8.72E+05	5.78E+05	2.28E+05
F19	1.67E+08	1.67E+08	1.67E+08	1.67E+08	4.01E+01	2.82E+09	2.82E+09	2.82E+09	2.82E+09	1.06E+02
F20	3.39E+03	3.09E+03	3.82E+03	3.35E+03	2.65E+02	5.27E+03	4.23E+03	5.93E+03	5.56E+03	6.49E+02
F21	2.52E+03	2.46E+03	2.62E+03	2.51E+03	4.60E+01	2.95E+03	2.82E+03	3.19E+03	2.90E+03	1.46E+02
F22	1.01E+04	9.06E+03	1.12E+04	1.01E+04	7.53E+02	1.83E+04	1.69E+04	2.09E+04	1.83E+04	1.13E+03
F23	2.97E+03	2.89E+03	3.02E+03	2.97E+03	3.90E+01	3.47E+03	3.34E+03	3.68E+03	3.45E+03	9.11E+01
F24	3.13E+03	3.07E+03	3.17E+03	3.13E+03	3.48E+01	4.07E+03	3.93E+03	4.31E+03	4.04E+03	1.09E+02
F25	3.09E+03	3.06E+03	3.12E+03	3.09E+03	1.91E+01	3.38E+03	3.26E+03	3.46E+03	3.38E+03	6.40E+01
F26	8.18E+03	6.31E+03	1.07E+04	7.69E+03	1.53E+03	1.87E+04	1.32E+04	2.36E+04	1.92E+04	3.28E+03
F27	3.62E+03	3.43E+03	3.86E+03	3.58E+03	1.34E+02	3.92E+03	3.72E+03	4.69E+03	3.84E+03	2.80E+02
F28	3.88E+03	3.88E+03	3.94E+03	3.88E+03	2.03E+01	5.46E+03	5.22E+03	5.77E+03	5.41E+03	1.54E+02
F29	1.55E+05	1.55E+05	1.57E+05	1.55E+05	6.22E+02	5.20E+04	5.05E+04	5.30E+04	5.22E+04	9.53E+02
F30	6.21E+09	6.21E+09	6.21E+09	6.21E+09	7.78E-05	2.06E+10	2.06E+10	2.06E+10	2.06E+10	1.98E-01



(a) F3 (b) F8 (c) F11 (d) F21

Fig. 5: Convergence function value results for D=50



(a) F3 (b) F8 (c) F11 (d) F21

Fig. 6: Convergence function value results for D=100

cluding the JSO [40], LSHADE-SPACMA [41], EBWO [42], EBLSHADE [43] and EAGDE [44].

4.3.1 Comparison of IADE algorithm Vs. DE

In the first experiment, we make a comparison between the DE algorithm and IADE algorithm. In this paper, we involve four main improvements: (i) Improvements of Mutation Factor, (ii) Improvements of Crossover Factor, (iii) Improvements of Mutation Strategy, (iiii) Improvements of Selection Strategy. To embody the advantage for each improvement strategy, we do the following evaluation.

(1) To test the individual effect of "Improvements of Mutation Factor" on the performance of IADE algorithm, the same IADE, version without the other three improvements (Improvements of Crossover Factor, Improvements of Mutation Strategy, and Improvements of Selection Strategy). This version can be called IADE-1.
(2) To test the individual effect of "Improvements of

TABLE 6: Performance comparison among DE, IADE-1 and IADE-2

	DE	IADE-1					IADE-2				
Func	Mean	Mean	Best	Worst	Median	Std	Mean	Best	Worst	Median	Std
F1	5.59E+10	6.05E+07	1.00E+02	2.42E+08	1.00E+02	1.21E+08	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.83E-14
F2	/	/	/	/	/	/	/	/	/	/	/
F3	7.02E+04	5.12E+02	3.00E+02	1.15E+03	3.00E+02	4.24E+02	3.00E+02	3.00E+02	3.00E+02	3.00E+02	2.61E-07
F4	1.94E+04	1.94E+04	1.94E+04	1.94E+04	1.94E+04	6.16E+00	1.94E+04	1.94E+04	1.94E+04	1.94E+04	2.10E-12
F5	8.48E+02	7.75E+02	7.70E+02	7.80E+02	7.75E+02	5.02E+00	7.54E+02	7.47E+02	7.59E+02	7.54E+02	6.56E+00
F6	6.63E+02	6.46E+02	6.43E+02	6.51E+02	6.45E+02	3.57E+00	6.41E+02	6.39E+02	6.42E+02	6.42E+02	1.30E+00
F7	1.21E+03	8.35E+02	7.68E+02	8.96E+02	8.39E+02	6.98E+01	7.80E+02	7.56E+02	8.02E+02	7.80E+02	2.11E+01
F8	1.04E+03	8.35E+02	8.29E+02	8.45E+02	8.33E+02	7.53E+00	8.36E+02	8.32E+02	8.43E+02	8.33E+02	5.02E+00
F9	7.44E+03	4.98E+03	4.55E+03	5.48E+03	4.94E+03	3.94E+02	5.66E+03	4.87E+03	7.55E+03	5.10E+03	1.27E+03
F10	5.96E+03	5.20E+03	4.05E+03	7.93E+03	4.41E+03	1.83E+03	6.02E+03	3.79E+03	8.07E+03	6.12E+03	2.01E+03
F11	1.55E+08	1.20E+03	1.18E+03	1.22E+03	1.20E+03	1.54E+01	1.15E+03	1.12E+03	1.17E+03	1.15E+03	3.13E+01
F12	2.36E+10	4.94E+06	4.60E+04	9.86E+06	4.92E+06	5.63E+06	5.66E+04	4.21E+03	8.50E+04	6.85E+04	3.81E+04
F13	7.20E+09	1.79E+07	9.44E+03	7.17E+07	2.13E+04	3.58E+07	1.34E+03	1.33E+03	1.36E+03	1.35E+03	1.32E+01
F14	5.42E+08	1.47E+03	1.43E+03	1.52E+03	1.46E+03	3.61E+01	1.43E+03	1.43E+03	1.44E+03	1.43E+03	3.85E+00
F15	1.56E+03	1.91E+03	1.74E+03	2.16E+03	1.88E+03	1.86E+02	1.51E+03	1.51E+03	1.51E+03	1.51E+03	1.52E+00
F16	2.57E+03	2.46E+03	2.11E+03	3.20E+03	2.27E+03	5.04E+02	2.02E+03	1.67E+03	2.38E+03	2.02E+03	2.93E+02
F17	1.99E+03	2.13E+03	1.87E+03	2.48E+03	2.09E+03	2.56E+02	1.83E+03	1.82E+03	1.86E+03	1.83E+03	1.63E+01
F18	4.79E+03	2.74E+05	1.92E+03	5.38E+05	2.78E+05	3.05E+05	1.91E+03	1.87E+03	1.96E+03	1.90E+03	4.15E+01
F19	1.93E+03	4.48E+05	2.31E+03	1.77E+06	1.17E+04	8.80E+05	2.30E+03	2.30E+03	2.30E+03	2.30E+03	2.24E+00
F20	2.10E+03	2.52E+03	2.28E+03	2.72E+03	2.54E+03	1.84E+02	2.12E+03	2.06E+03	2.24E+03	2.09E+03	8.35E+01
F21	2.89E+03	2.40E+03	2.34E+03	2.48E+03	2.39E+03	7.20E+01	2.34E+03	2.33E+03	2.35E+03	2.34E+03	8.67E+00
F22	8.56E+03	5.96E+03	5.12E+03	7.55E+03	5.57E+03	1.09E+03	8.04E+03	5.64E+03	9.42E+03	8.54E+03	1.76E+03
F23	5.44E+03	2.72E+03	2.71E+03	2.74E+03	2.72E+03	1.31E+01	2.70E+03	2.68E+03	2.72E+03	2.70E+03	1.54E+01
F24	4.90E+03	2.91E+03	2.87E+03	2.96E+03	2.91E+03	3.65E+01	2.87E+03	2.86E+03	2.88E+03	2.87E+03	1.27E+01
F25	6.74E+03	2.89E+03	2.89E+03	2.91E+03	2.89E+03	1.03E+01	2.89E+03	2.89E+03	2.89E+03	2.89E+03	2.84E-02
F26	1.30E+04	4.27E+03	4.05E+03	4.58E+03	4.24E+03	2.47E+02	4.00E+03	3.88E+03	4.07E+03	4.03E+03	8.27E+01
F27	7.30E+03	3.21E+03	3.20E+03	3.22E+03	3.21E+03	6.60E+00	3.19E+03	3.19E+03	3.20E+03	3.19E+03	4.53E+00
F28	8.50E+03	3.58E+03	3.54E+03	3.62E+03	3.58E+03	3.35E+01	3.54E+03	3.54E+03	3.54E+03	3.54E+03	2.63E-13
F29	8.32E+04	7.33E+03	7.14E+03	7.63E+03	7.27E+03	2.30E+02	7.03E+03	7.03E+03	7.05E+03	7.03E+03	1.18E+01
F30	7.76E+09	2.65E+09	2.65E+09	2.65E+09	2.65E+09	5.93E+05	2.65E+09	2.65E+09	2.65E+09	2.65E+09	0.00E+00

Crossover Factor” on the performance of IADE algorithm, the same IADE, version without the other three improvements (Improvements of Mutation Factor, Improvements of Mutation Strategy, and Improvements of Selection Strategy). This version can be called IADE-2.

(3) To test the individual effect of ”Improvements of Mutation Strategy” on the performance of IADE algorithm, the same IADE, version without the other three improvements (Improvements of Mutation Factor, Improvements of Crossover Factor, and Improvements of Selection Strategy). This version can be called IADE-3.

(4) To test the individual effect of ”Improvements of Selection Strategy” on the performance of IADE algorithm, the same IADE, version without the other three improvements (Improvements of Mutation Factor, Improvements of Crossover Factor, and Improvements of Mutation Strategy). This version can be called IADE-4.

Table 6 and Table 7 show the experimental results with the dimension D=30 among the DE and IADE-1, IADE-2, IADE-3, and IADE-4 algorithm.

To obtain the statistical results for the DE and IADE-1, IADE-2, IADE-3 and IADE-4 algorithm, we select the ”Wilcoxon” method [48] do a statistical test. Table 8 illustrates the statistical results and final decision that is which algorithm is better As shown in Table 8, we have test four groups, that is ”IADE-1 vs.- DE”, ”IADE-2 vs.- DE”, ”IADE-3 vs.- DE”, and ”IADE-4 vs.- DE”. Sign ”+” represents the first algorithm is better than the second algorithm, that is to say the first algorithm owns the better performance. Contrary to sign ”+”, sign ”-” means the the second algorithm has a better than the first one; Sign ”≈” means that there is no significant different between the two algorithms. Sign ”Dec.” represent the decision making. Table 8 shows that, IADE-1, IADE-2, IADE-3, and IADE-4 owns the better performance than DE algorithm.

We also compare IADE against the state-of-the-art algorithms including the JSO [40], LSHADE-SPACMA [41], EBWO [42], EBLSHADE [43] and EAGDE [44]. Table 9 presents the comparison results.

To determine which algorithm is better than other one. We have made a ”Wilcoxon” statistical test. The test results are shown in Table 10.

In Table 10, we have tested five groups, that is ”IADE vs.- JSO”, ”IADE vs.- LSADE-SPACMA”, ”IADE vs.- EBWO”, ”IADE vs.- EBLSHADE”, and IADE vs.- EAGDE”. Sign ”+” represents the first algorithm is better than the second algorithm. Contrary to sign ”+”, sign ”-” means the the second algorithm has a better than the first one; Sign ”≈” means that there is no significant different between the two algorithms. Sign ”Dec.” represent the decision making. As Table 10 states, IADE has the better performance compared with the EAGDE algorithm. However, for the other four algorithms under the CEC 2017 function set, the IADE algorithm presents some weaknesses in terms of convergence.

For comparisons and evaluations, we select the five algorithms that are the JSO [40], SHADE-SPACMA [41], EBWO [42], EBLSHADE [43], EAGDE [44], and DE as comparison algorithms to embody the advantage of IADE in terms of total execution time, load balance, and delivered QoS. The parameter value is listed in Table 11. The task size is randomly and uniformly generated and the values are [5000, 10000] MI; the number of VMs was 10; the processing capacities of VMs were generated as random numbers, and its values belong to [1000, 5000].

TABLE 7: Performance comparison among DE, IADE-3 and IADE-4

	DE	IADE-3					IADE-4				
Func	Mean	Mean	Best	Worst	Median	Std	Mean	Best	Worst	Median	Std
F1	5.59E+10	8.07E+03	1.00E+02	2.44E+04	3.90E+03	1.15E+04	1.06E+10	6.92E+09	1.56E+10	8.98E+09	3.66E+09
F2	/	/	/	/	/	/	/	/	/	/	/
F3	7.02E+04	3.00E+02	3.00E+02	3.00E+02	3.00E+02	5.13E-13	1.93E+05	6.66E+04	2.72E+05	1.88E+05	8.06E+04
F4	1.94E+04	1.94E+04	1.94E+04	1.94E+04	1.94E+04	3.39E-02	2.14E+04	2.10E+04	2.21E+04	2.11E+04	4.81E+02
F5	8.48E+02	8.06E+02	7.87E+02	8.28E+02	8.04E+02	2.12E+01	8.41E+02	8.24E+02	8.82E+02	8.34E+02	2.36E+01
F6	6.63E+02	6.50E+02	6.46E+02	6.55E+02	6.50E+02	3.58E+00	6.70E+02	6.64E+02	6.83E+02	6.65E+02	8.73E+00
F7	1.21E+03	7.81E+02	7.63E+02	8.25E+02	7.69E+02	2.92E+01	1.05E+03	1.00E+03	1.16E+03	1.03E+03	6.65E+01
F8	1.04E+03	8.58E+02	8.43E+02	8.69E+02	8.60E+02	1.15E+01	1.00E+03	9.54E+02	1.05E+03	1.02E+03	4.55E+01
F9	7.44E+03	4.94E+03	4.50E+03	5.20E+03	5.03E+03	3.23E+02	7.87E+03	6.59E+03	8.86E+03	7.95E+03	8.30E+02
F10	5.96E+03	3.35E+03	2.70E+03	3.75E+03	3.47E+03	4.97E+02	6.36E+03	5.16E+03	7.52E+03	6.52E+03	9.40E+02
F11	1.55E+08	1.24E+03	1.19E+03	1.29E+03	1.24E+03	4.71E+01	5.58E+03	2.82E+03	7.01E+03	6.37E+03	1.80E+03
F12	2.36E+10	1.98E+04	4.60E+03	4.24E+04	1.62E+04	1.78E+04	4.76E+08	9.92E+07	8.65E+08	5.72E+08	3.45E+08
F13	7.20E+09	3.48E+04	2.94E+03	6.61E+04	3.50E+04	3.52E+04	1.48E+05	1.09E+05	2.01E+05	1.50E+05	3.75E+04
F14	5.42E+08	1.49E+03	1.46E+03	1.52E+03	1.49E+03	2.73E+01	7.95E+05	7.45E+03	2.92E+06	3.09E+05	1.22E+06
F15	1.56E+03	1.22E+04	1.53E+03	4.36E+04	1.85E+03	2.09E+04	8.24E+04	4.35E+04	1.42E+05	6.48E+04	4.38E+04
F16	2.57E+03	2.29E+03	2.18E+03	2.51E+03	2.24E+03	1.55E+02	3.36E+03	2.67E+03	3.94E+03	3.46E+03	5.31E+02
F17	1.99E+03	1.91E+03	1.78E+03	2.13E+03	1.87E+03	1.57E+02	2.26E+03	1.94E+03	2.53E+03	2.33E+03	2.43E+02
F18	4.79E+03	1.98E+03	1.84E+03	2.31E+03	1.89E+03	2.16E+02	1.03E+07	1.93E+05	2.10E+07	1.02E+07	1.01E+07
F19	1.93E+03	2.41E+03	2.38E+03	2.46E+03	2.40E+03	3.22E+01	8.22E+06	3.96E+06	1.34E+07	9.15E+06	3.70E+06
F20	2.10E+03	2.28E+03	2.19E+03	2.39E+03	2.26E+03	8.35E+01	2.76E+03	2.68E+03	2.89E+03	2.72E+03	8.97E+01
F21	2.89E+03	2.40E+03	2.35E+03	2.52E+03	2.37E+03	8.24E+01	2.53E+03	2.49E+03	2.58E+03	2.52E+03	4.15E+01
F22	8.56E+03	2.79E+03	2.30E+03	4.24E+03	2.31E+03	9.67E+02	7.54E+03	3.89E+03	9.04E+03	8.18E+03	2.08E+03
F23	5.44E+03	2.77E+03	2.71E+03	2.88E+03	2.74E+03	7.90E+01	2.95E+03	2.85E+03	3.07E+03	2.93E+03	8.60E+01
F24	4.90E+03	2.92E+03	2.88E+03	2.98E+03	2.90E+03	4.40E+01	3.14E+03	3.06E+03	3.20E+03	3.14E+03	5.84E+01
F25	6.74E+03	2.89E+03	2.89E+03	2.90E+03	2.89E+03	3.22E+00	3.83E+03	3.29E+03	4.99E+03	3.62E+03	6.71E+02
F26	1.30E+04	4.47E+03	4.08E+03	4.72E+03	4.55E+03	2.77E+02	6.92E+03	6.18E+03	8.12E+03	6.66E+03	8.56E+02
F27	7.30E+03	3.23E+03	3.22E+03	3.24E+03	3.23E+03	1.03E+01	3.43E+03	3.31E+03	3.51E+03	3.46E+03	8.41E+01
F28	8.50E+03	3.60E+03	3.56E+03	3.64E+03	3.59E+03	3.33E+01	4.29E+03	4.07E+03	4.50E+03	4.24E+03	1.95E+02
F29	8.32E+04	7.93E+03	7.64E+03	8.35E+03	7.86E+03	3.02E+02	1.06E+04	9.05E+03	1.24E+04	1.04E+04	1.23E+03
F30	7.76E+09	2.65E+09	2.65E+09	2.65E+09	2.65E+09	4.97E+04	3.39E+09	3.24E+09	3.52E+09	3.41E+09	1.06E+08

TABLE 8: Wilcoxon's test results on D=30

Algorithms	+	≈	-	Dec.
IADE-1 vs.- DE	23	2	5	+
IADE-2 vs.- DE	25	2	3	+
IADE-3 vs.- DE	25	2	3	+
IADE-4 vs.- DE	18	2	10	+

4.3.2 *Comparison based on Total Execution Time*

The second experiment is to evaluate the indicator of task execution time. In general, the lower the task execution time, the higher the performance. Fig. 7 illustrates the performance of the algorithms, as shown below:

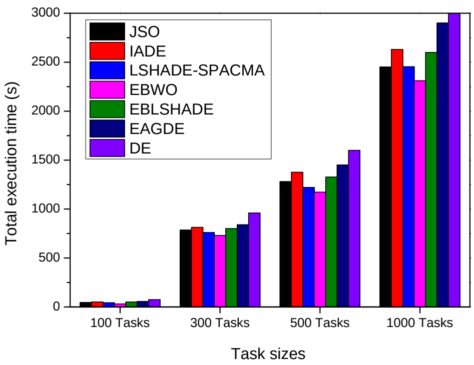


Fig. 7: The total execution time comparison

Fig. 7 indicates that the total execution time grows with the increase of task number. This figure also shows that, in terms of total execution time, EBWO leads to the best performance, LSHADE-SPACMA the second, JSO is the third, EBLSHADE fouth, IADE the fifth, EAGDE the sixth, DE the worst. The EBWO algorithm is the best because it leverages the covariance matrix to generate a new solution and improve the local search capability of the algorithm. IADE is better than EAGDE and DE algorithm. The reason is that IADE improves the scaling factor, crossover probability, variation, and selection strategy of the DE algorithm. Thus, the parameters can be adaptively adjusted with the iterative evolution of the population. Therefore, the IADE algorithm leads to better performance in terms of total execution time. LSHADE-SPACMA is better than the JSO algorithm, and the reason is that LSHADE-SPACMA leverages the advantage of both LSHADE-SPA and CMA-ES.

4.3.3 *Comparisons based on Workload Balance*

The third experiment evaluates the indicator of workload balance. Fig. 8 displays the comparison of workload balance.

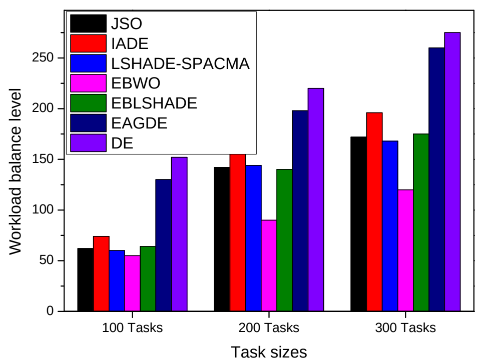


Fig. 8: The workload balance comparison

Regarding the workload balance level, the less value of the workload balance, the better. Fig. 8 present the workload

TABLE 9: Comparison with the state-of-the-art algorithms for CEC 2017 and CEC 2013

	CEC 2017 and D=30					CEC 2013 and D=30	
Func	JSO	IADE	LSHADE-SPACMA	EBWO	EBLSHADE	EAGDE	IADE
F1	0.00E+00	8.49E+03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.0000E+00
F2	/	/	/	/	/	/	/
F3	0.00E+00	3.00E+02	0.00E+00	0.00E+00	0.00E+00	2.15E+03	1.4000E+03
F4	5.87E+01	1.94E+04	5.86E+01	5.65E+01	5.90E+01	3.85E+00	5.3905E-01
F5	8.56E+00	7.93E+02	3.45E+00	2.78E+00	6.70E+00	0.00E+00	0.0000E+00
F6	6.04E-09	6.55E+02	0.00E+00	0.00E+00	1.50E-08	4.84E+00	7.6467E+00
F7	3.89E+01	9.04E+02	3.38E+01	3.35E+01	3.70E+01	2.34E+00	2.0225E+00
F8	9.09E+00	9.24E+02	3.20E+00	2.02E+00	8.00E+00	2.09E+01	2.1135E+00
F9	0.00E+00	5.44E+03	0.00E+00	0.00E+00	0.00E+00	2.56E+01	5.0132E+01
F10	1.53E+03	4.75E+03	1.44E+03	1.41E+03	1.40E+03	1.34E-02	2.0656E-02
F11	3.04E+00	1.22E+03	1.78E+01	4.49E+00	3.40E+01	0.00E+00	0.0000E+00
F12	1.70E+02	5.57E+04	6.15E+02	4.63E+02	1.00E+03	8.46E+01	5.8227E+01
F13	1.48E+01	1.57E+04	1.46E+01	1.49E+01	1.60E+01	1.04E+02	1.3153E+02
F14	2.18E+01	1.64E+03	2.34E+01	2.19E+01	2.20E+01	2.00E-01	2.1266E-01
F15	1.09E+00	6.84E+03	4.46E+00	3.69E+00	3.80E+00	5.46E+03	3.4056E+03
F16	7.89E+01	2.37E+03	2.52E+01	4.26E+01	4.20E+01	2.27E+00	2.0076E+00
F17	3.29E+01	2.08E+03	3.04E+01	2.98E+01	3.30E+01	3.04E+01	1.0033E+01
F18	2.04E+01	1.47E+04	2.34E+01	2.21E+01	2.30E+01	1.80E+02	6.2975E+02
F19	4.50E+00	7.20E+03	1.03E+01	8.04E+00	6.10E+00	2.72E+00	1.9645E+00
F20	2.94E+01	2.53E+03	8.38E+01	3.57E+01	3.10E+01	1.16E+01	6.1500E+01
F21	2.09E+02	2.38E+03	2.07E+02	1.99E+02	2.10E+02	2.94E+02	6.1500E+02
F22	1.00E+02	4.08E+03	1.00E+02	1.00E+02	1.00E+02	1.11E+02	8.9199E+02
F23	3.51E+02	2.73E+03	3.55E+02	3.51E+02	3.60E+02	5.75E+03	4.7833E+03
F24	4.26E+02	2.94E+03	4.29E+02	4.18E+02	4.30E+02	2.05E+02	1.2658E+02
F25	3.87E+02	2.90E+03	3.87E+02	3.87E+02	3.90E+02	2.80E+02	1.3945E+02
F26	9.20E+02	5.56E+03	9.53E+02	5.37E+02	9.80E+02	2.00E+02	1.4000E+03
F27	4.97E+02	3.24E+03	5.05E+02	5.02E+02	5.10E+02	6.22E+02	2.2453E+02
F28	3.09E+02	3.54E+03	3.11E+02	3.08E+02	3.40E+02	3.00E+02	3.0902E+03
F29	4.34E+02	7.34E+03	4.45E+02	4.33E+02	4.40E+02		
F30	1.97E+03	2.65E+09	2.01E+03	1.99E+03	2.00E+03		

TABLE 10: Wilcoxon's test results on D=30

Algorithms	+	≈	-	Dec.
IADE vs.- JSO	0	1	29	-
IADE vs.- LSADE-SPACMA	0	1	29	-
IADE vs.- EBWO	0	1	29	-
IADE vs.- EBLSHADE	0	1	29	-
IADE vs.- EAGDE	13	3	12	+

TABLE 11: Parameter settings

Parameter	**Values**
Task size	5000-10000MI
Task number	100-1000
Number of VMs	10
Processing capacities of VMs	1000-5000MI
Mutation factor F	0.5
Crossover factor CR	0.9
Population size NP	100

balance level of our proposed algorithm, IADE, in terms of workload balance and highlights that IADE has a better performance than EAGDE and DE. The reason behind this can be explained as follows. Due to the limitation of fixed parameters and single mutation strategy, the solution accuracy and speed of the two algorithms are lower than IADE. The workload balance is slower than IADE. In the field of workload balance level, the algorithms (EBWO, LSHADE-SPACMA, JSO, and EBLSHADE) have a better workload balance level than the IADE algorithm. This is because these algorithms have some advantages in finding a solution.

4.3.4 *Comparison based on Quality of Services*

For any multi-objective scheduling or energy-aware algorithms, the Quality of Services (QoS) indicator plays a key role during evaluation. The third experiment is to test the indicator of QoS delivered. As the QoS factor is related to the execution time and workload balance, the definition of QoS indicator (T_{QoS}) can be defined as follows:

$$T_{QoS} = \frac{1}{T_E \times T_B} \tag{11}$$

Where parameter T_{QoS} corresponds to the QoS delivered by the comparisons, T_E is the task execution time, and T_B represents the workload balance. As there is no uniform unit for parameter T_E and T_B, it needs to do a normalization.

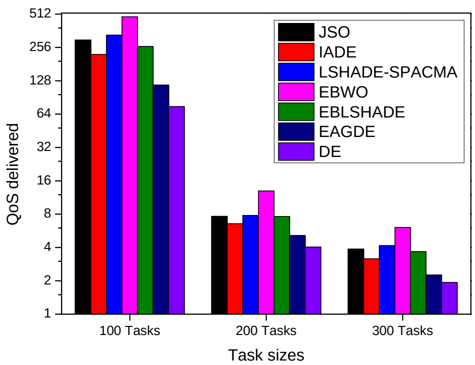


Fig. 9: The delivered QoS comparison

According to the definition of Eq. (11), Fig. 9 displays the comparison of the QoS delivered. From this figure, we understand that when the task numbers increase, the QoS delivered decreases for all algorithms, and interestingly, IADE performs better than EAGDE and DE. The reason can be explained as follows: according to the definition of Eq. (11), the QoS indicator is related to task execution time and workload balance. Compared with the EAGDE and DE algorithms (as the two highest QoS delivered after IADE), IADE has a better performance in task execution time and workload balance. Therefore, it confirms that IADE performs the best performance. Also, LSHADE-SPACMA is better than JSO. The same reason could explain the reason. As the four algorithms (EBWO, LSHADE-SPACMA, JSO, and EBLSHADE) have some advantages in terms of task execution time and workload balance, according to the Eq. (11), they lead to better QoS performance.

5 Application of the IADE in 6G Environment

We indicate that our IADE could be extended and applied in many fields, such as in 5G/6G networked devices, intelligence cloud, Blockchain [49], large-scale networked data center [50] intending to meet the requirement of sustainable development. This part will apply the IADE algorithm to task scheduling within the intelligent cloud to evaluate its performance. In this case, IADE could be tuned and applied as an intelligence solution over MEC to tackle task scheduling of the lower-level demands coming from IoT or industry 4.0 applications and optimize the energy consumption of the engaged entities like servers and data centers. Additionally, this method can be placed as encapsulated service over the hosts connected to the switches and locally monitor the traffic flows in each switch equipped with an edge node.

6 Conclusions and Future Directions

This paper proposed an improved adaptive differential evolution algorithm, IADE, defining a green network data steering to preserve resource allocation in the 6G network. IADE improves the mutation factor F, crossover factor CR, mutation, and selection strategies of the standard DE algorithm applying network data traffic. The IADE algorithm and other variant mode DE algorithms are tested using thirty classical benchmark functions. Also, we assess the application of the proposed algorithm in intelligent IoT data traversed through a Cloud. The experimental results confirm that the IADE algorithm is more capable of convergence and maximizing local optimization than other DE algorithms. The proposed algorithm can be applied in large-scale network intelligence combined with a networked data center, providing network sustainability and improving service quality for different levels of resource-constrained devices. In future, we plan to extend IADE as multi-objective solutions jointly the metrics related to the large-scale environment-aware network capabilities and concurrent connections between the network devices, increasing companies' investment return rate.

Acknowledgments

Mohammad Shojafar is supported by Marie Curie Global Fellowship funded by European Commission with grant agreement MSCA-IF-GF-839255.

References

[1] S. Zhang and D. Zhu, "Towards artificial intelligence enabled 6g: State of the art, challenges, and opportunities," *Computer Networks*, p. 107556, 2020.

[2] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.

[3] A. Mukherjee, P. Goswami, M. A. Khan, L. Manman, L. Yang, and P. Pillai, "Energy efficient resource allocation strategy in massive iot for industrial 6g applications," *IEEE Internet of Things Journal*, 2020.

[4] H. Hamedmoghadam, M. Jalili, P. Moradi, and X. Yu, "A global optimization approach based on opinion formation in complex networks," *IEEE Transactions on Network Science and Engineering*, vol. 6, no. 3, pp. 173–187, 2018.

[5] X. Wu, Y. Wang, J. Liu, and N. Fan, "A new hybrid algorithm for solving large scale global optimization problems," *IEEE Access*, vol. 7, pp. 103 354–103 364, 2019.

[6] J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization: Algorithm jso," in *2017 IEEE congress on evolutionary computation (CEC)*. IEEE, 2017, pp. 1311–1318.

[7] A. Nordio, A. Tarable, C.-F. Chiasserini, and E. Leonardi, "Belief dynamics in social networks: A fluid-based analysis," *IEEE Transactions on Network Science and Engineering*, vol. 5, no. 4, pp. 276–287, 2017.

[8] F. Martín, L. Moreno, S. Garrido, and D. Blanco, "Kullback-leibler divergence-based differential evolution markov chain filter for global localization of mobile robots," *Sensors*, vol. 15, no. 9, pp. 23 431–23 458, 2015.

[9] D. Chen, C. Yang, P. Gong, L. Chang, J. Shao, Q. Ni, A. Anpalagan, and M. Guizani, "Resource cube: Multi-virtual resource management for integrated satellite-terrestrial industrial iot networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 11 963–11 974, 2020.

[10] O. S. Badarneh and D. B. da Costa, "Cascaded fluctuating two-ray fading channels," *IEEE Communications Letters*, vol. 23, no. 9, pp. 1497–1500, 2019.

[11] M. M. M. Fuad, "Differential evolution-based weighted combination of distance metrics for k-means clustering," in *International Conference on Theory and Practice of Natural Computing*. Springer, 2014, pp. 193–204.

[12] E. Bochinski, T. Senst, and T. Sikora, "Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3924–3928.

[13] C. Lim, "A survey on congestion control for rpl-based wireless sensor networks," *Sensors*, vol. 19, no. 11, p. 2567, 2019.

[14] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 482–500, 2011.

[15] W. Ejaz, S. K. Sharma, S. Saadat, M. Naeem, A. Anpalagan, and N. A. Chughtai, "A comprehensive survey on resource allocation for cran in 5g and beyond networks," *Journal of Network and Computer Applications*, p. 102638, 2020.

[16] S. K. Sharma, I. Woungang, A. Anpalagan, and S. chatzinotas, "Toward tactile internet in beyond 5g era: Recent advances, current issues, and future directions," *IEEE Access*, vol. 8, pp. 56 948–56 991, 2020.

[17] D. Qing-feng and Y. Xiao-yu, "Research survey of differential evolution algorithms," *CAAI Transactions on Intelligent Systems*, vol. 12, no. 04, pp. 431–442, 2017.

[18] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2008.

[19] J. Tvrdík and R. Poláková, "Competitive differential evolution applied to cec 2013 problems," in *2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 1651–1657.

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

[21] R. Mallipeddi, P. N. Suganthan, Q.-K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied soft computing*, vol. 11, no. 2, pp. 1679–1696, 2011.

[22] L. Tang, Y. Zhao, and J. Liu, "An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 209–225, 2013.
[23] A. Zamuda, J. Brest, and E. Mezura-Montes, "Structured population size reduction differential evolution with multiple mutation strategies on cec 2013 real parameter optimization," in *2013 IEEE congress on evolutionary computation*. IEEE, 2013, pp. 1925–1931.
[24] R. Mallipeddi, P. N. Suganthan, Q.-K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied soft computing*, vol. 11, no. 2, pp. 1679–1696, 2011.
[25] M. Ali, P. Siarry, and M. Pant, "An efficient differential evolution based algorithm for solving multi-objective optimization problems," *European journal of operational research*, vol. 217, no. 2, pp. 404–416, 2012.
[26] X. Qiu, J.-X. Xu, Y. Xu, and K. C. Tan, "A new differential evolution algorithm for minimax optimization in robust design," *IEEE transactions on cybernetics*, vol. 48, no. 5, pp. 1355–1368, 2017.
[27] Y. Li, S. Wang, X. Hong, and Y. Li, "Multi-objective task scheduling optimization in cloud computing based on genetic algorithm and differential evolution algorithm," in *2018 37th Chinese Control Conference (CCC)*. IEEE, 2018, pp. 4489–4494.
[28] F. Li, J. Liu, H. Shi, and Z. Fu, "Multi-objective particle swarm optimization algorithm based on decomposition and differential evolution," *Control Decis*, vol. 32, no. 3, pp. 403–410, 2017.
[29] F. Qiao, J. Wu, J. Li, A. K. Bashir, S. Mumtaz, and U. Tariq, "Trustworthy edge storage orchestration in intelligent transportation systems using reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, 2020.
[30] S. Omer, S. Azizi, M. Shojafar, and R. Tafazolli, "A priority, power and traffic-aware virtual machine placement of iot applications in cloud data centers," *Journal of Systems Architecture*, pp. 1–25, 2021.
[31] A. W. Mohamed, "Solving stochastic programming problems using new approach to differential evolution algorithm," *Egyptian informatics journal*, vol. 18, no. 2, pp. 75–86, 2017.
[32] A. K. Mohamed, A. A. Hadi, and A. W. Mohamed, "Generalized adaptive differential evolution algorithm for solving cec 2020 benchmark problems," in *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*. IEEE, 2020, pp. 391–396.
[33] V. Stanovov, S. Akhmedova, and E. Semenkin, "Selective pressure strategy in differential evolution: exploitation improvement in solving global optimization problems," *Swarm and Evolutionary Computation*, vol. 50, p. 100463, 2019.
[34] R. Polakova, J. Tvrdik, and P. Bujok, "Differential evolution with adaptive mechanism of population size according to current population diversity," *Swarm and Evolutionary Computation*, vol. 50, p. 100519, 2019.
[35] A. W. Mohamed and A. K. Mohamed, "Adaptive guided differential evolution algorithm with novel mutation for numerical optimization," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 2, pp. 253–277, 2019.
[36] L. Cui, G. Li, Z. Zhu, Q. Lin, K.-C. Wong, J. Chen, N. Lu, and J. Lu, "Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism," *Information Sciences*, vol. 422, pp. 122–143, 2018.
[37] A. K. Bashir, R. Arul, S. Basheer, G. Raja, R. Jayaraman, and N. M. F. Qureshi, "An optimal multitier resource allocation of cloud ran in 5g using machine learning," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 8, p. e3627, 2019.
[38] Y. Fan, R. Jin, J. Geng, and B. Liu, "A hybrid optimized algorithm based on differential evolution and genetic algorithm and its applications in pattern synthesis of antenna arrays," *Acta Electronica Sinica*, vol. 32, no. 12, pp. 1997–2000, 2004.
[39] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective bound constrained real-parameter numerical optimization," *Technical Report*, 2016.
[40] J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization: Algorithm jso," in *2017 IEEE congress on evolutionary computation (CEC)*. IEEE, 2017, pp. 1311–1318.
[41] A. W. Mohamed, A. A. Hadi, A. M. Fattouh, and K. M. Jambi, "Lshade with semi-parameter adaptation hybrid with cma-es for solving cec 2017 benchmark problems," in *2017 IEEE Congress on evolutionary computation (CEC)*. IEEE, 2017, pp. 145–152.
[42] A. Kumar, R. K. Misra, and D. Singh, "Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase," in *2017 IEEE congress on evolutionary computation (CEC)*. IEEE, 2017, pp. 1835–1842.
[43] A. W. Mohamed, A. A. Hadi, and K. M. Jambi, "Novel mutation strategy for enhancing shade and lshade algorithms for global numerical optimization," *Swarm and Evolutionary Computation*, vol. 50, p. 100455, 2019.
[44] A. K. Mohamed and A. W. Mohamed, "Real-parameter unconstrained optimization based on enhanced agde algorithm," in *Machine Learning Paradigms: Theory and Application*. Springer, 2019, pp. 431–450.
[45] 2021, "IADE source code," https://github.com/mshojafar/sourcecodes/tree/master/Zhou2021-IADE_Sourcecode.
[46] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, "Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm," *International Journal of Machine Learning and Cybernetics*, pp. 1–29, 2020.
[47] A. A. Hadi, A. W. Mohamed, and K. M. Jambi, "Single-objective real-parameter optimization: Enhanced lshade-spacma algorithm," in *Heuristics for Optimization and Learning*. Springer, 2021, pp. 103–121.
[48] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
[49] M. I. S. Assaqty, Y. Gao, X. Hu, Z. Ning, V. C. M. Leung, Q. Wen, and Y. Chen, "Private-blockchain-based industrial iot for material and product tracking in smart manufacturing," *IEEE Network*, vol. 34, no. 5, pp. 91–97, 2020.
[50] J. An, F. Le Gall, J. Kim, J. Yun, J. Hwang, M. Bauer, M. Zhao, and J. Song, "Toward global iot-enabled smart cities interworking using adaptive semantic adapter," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5753–5765, 2019.

Zhou Zhou received the Ph.D. degree from Central South University, Changsha, China, in 2017, majoring in computer science. He is currently an Associate Professor at Changsha University. He has published over 20 academic papers in cloud/edge computing and green computing, and also holds several Chinese patents. His research interests include edge computing, cloud computing, energy consumption model, energy-efficient resource management, and virtual machine deployment.

Mohammad Shojafar (M'17-SM'19) is a Senior Lecturer (Associate Professor) in the network security and an Intel Innovator, Profassional ACM member, and a Marie Curie Alumni, working in the 5G & 6G Innovation Centre (5GIC/6GIC) at the University of Surrey, UK. Before joining 5GIC/6GIC, he was a Senior Researcher and a Marie Curie Fellow in the SPRITZ Security and Privacy Research group at the University of Padua, Italy. Also, he was CNIT Senior Researcher at the University of Rome Tor Vergata contributed to 5G PPP European H2020 "SUPERFLUIDITY" project. Dr. Mohammad was a PI of PRISENODE project, a 275k euro Horizon 2020 Marie Curie global fellowship project in the areas of Fog/Cloud security collaborating at the University of Padua. He also is a PI on 750k euro project named AUTOTRUST supported by European Space Agency in 2021. He is a PI in an Italian SDN security and privacy (60k euro) supported by the University of Padua in 2018. He was contributed to some Italian projects in telecommunications like GAUChO, SAMMClouds, and SC2. He received his Ph.D. degree from Sapienza University of Rome, Rome, Italy, in 2016 with an "Excellent" degree. He is an Associate Editor in *IEEE Transactions on Network and Service Management*, *IEEE Transactions on Consumer Electronics*, *IEEE Systems Journal*, *Computer Networks* and *IET Communications* journals. For additional information: http://mshojafar.com

Jemal Abawajy (SM'11) is a full professor at Faculty of Science, Engineering and Built Environment, Deakin University, Australia. Professor Abawajy has served on the editorial-board of numerous international journals and currently serving as associate editor of the International Journal of Big Data Intelligence and International Journal of Parallel, Emergent and Distributed Systems. Prof. Abawajy has delivered more than 50 keynote and seminars worldwide and has been involved in the organization of more than 300 international conferences in various capacity including chair and general co-chair. He has also served on the editorial-board of numerous international journals including *IEEE Transactions on Cloud Computing*. Prof. Abawajy is the author/co–author of more than 250 refereed articles and supervised numerous PhD students to completion. For additional information:https://www.deakin.edu.au/about-deakin/people/jemal-abawajy

Ali Kashif Bashir (SM'16) is a Reader/Associate Professor and Program Leader of BSc (H) Computer Forensics and Security at the Department of Computing and Mathematics, Manchester Metropolitan University, United Kingdom. He is also with School of Electrical Engineering and Computer Science, National University of Science and Technology, Islamabad (NUST) as an Adjunct Professor and School of Information and Communication Engineering, University of Electronics Science and Technology of China (UESTC) as an Affiliated Professor and Chief Advisor of Visual Intelligence Research Center, UESTC. He is a senior member of IEEE, member of IEEE Industrial Electronic Society, member of ACM, and Distinguished Speaker of ACM. His past assignments include Associate Professor of ICT, University of the Faroe Islands, Denmark; Osaka University, Japan; Nara National College of Technology, Japan; the National Fusion Research Institute, South Korea; Southern Power Company Ltd., South Korea, and the Seoul Metropolitan Government, South Korea. He has worked on several research and industrial projects of South Korean, Japanese and European agencies and Government Ministries. In his career, he has obtained over 2.5 Million USD funding. He received his Ph.D. in computer science and engineering from Korea University South Korea. He has authored over 180 research articles; received funding as PI and Co-PI from research bodies of South Korea, Japan, EU, UK and Middle East; supervising/co-supervising several graduate (MS and PhD) students. His research interests include internet of things, wireless networks, distributed systems, network/cyber security, network function virtualization, machine learning, etc. He is serving as the Editor-in-chief of the IEEE FUTURE DIRECTIONS NEWSLETTER. He is also serving as area editor of KSII Transactions on Internet and Information Systems; associate editor of IEEE Internet of Things Magazine, IEEE Access, Peer J Computer Science, IET Quantum Computing, Journal of Plant Disease and Protection. He is leading many conferences as a chair (program, publicity, and track) and had organized workshops in flagship conferences like IEEE Infocom, IEEE Globecom, IEEE Mobicom, etc.