An improved cooperation search algorithm for the multi-degree reduction of Ball Bézier surfaces

Huanxin Cao
Northwestern Polytechnical University
Hongchan Zheng (zhenghc@nwpu.edu.cn)
Northwestern Polytechnical University
Gang Hu
Xi'an University of Technology

Research Article

Keywords: Metaheuristic method, CSA, Ball B´ezier surface, Multi-degree reduction

Posted Date: September 22nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1261686/v1

An improved cooperation search algorithm for the multi-degree reduction of Ball Bézier surfaces

Huanxin Cao[a], Hongchan Zheng[a,*], Gang Hu[b,*]

[a]*School of Mathematics and Statistics*
Northwestern Polytechnical University, Xi'an, China
[b]*Department of Applied Mathematics*
Xi'an University of Technology, Xi'an, China

Abstract

Cooperation search algorithm (CSA) is a new metaheuristic algorithm inspired from the team cooperation behaviors in modern enterprises and is characterized by fast convergence. However for some complex problems, it may get trapped into local optima and suffer from premature convergence for the shortcoming of population updating guided only by leading individuals. In this paper, an improved cooperation search algorithm (CCSA) is proposed by incorporating the mutation and crossover operators in DE algorithms to alleviate the shortcoming. The two operators can be used to increase population's diversity significantly, and thus improve population's exploration capability and accuracy significantly. CCSA has been tested on 23 benchmark functions and CEC 2017 benchmark suite. Experimental results demonstrate the better performance of CCSA on convergence speed and accuracy as compared to other existing optimizers. Furthermore, aiming at the problem that there is no universal approach for the multi-degree reduction of Ball Bézier surfaces under different interpolation constrains, we propose a new method to solve this problem by introducing metaheuristic methods, where the change of interpolation constrains are treated as the change of decision variables. The modeling examples show that the proposed method is effective and easy to implement under different interpolation constrains, which achieves the automatic and intelligent degree reduction of Ball Bézier surfaces.

Keywords: Metaheuristic method, CSA, Ball Bézier surface, Multi-degree reduction

---

1. Introduction

Mathematical optimization refers to the seeking of a solution within an available solution space such that the objective function reaches its maximum or minimum. A lot of real-world problems in practice can be attributed to optimization problems. Therefore, the solving methods for optimization problems and their solution accuracy have always been the focus

*Corresponding author

Email addresses: `huanxincao@sina.cn` (Huanxin Cao), `zhenghc@nwpu.edu.cn` (Hongchan Zheng), `hugang@xaut.edu.cn` (Gang Hu)

of many scholars. Although conventional methods based on derivative theory have the merit of high precision, they have requirements of differentiability and continuation of function. Besides, they are sensitive to the initial value.

Inspired form the abundant mechanisms and principles in nature, scholars have developed numerous metaheuristic methods to address various optimization problems in the past few decades. These metaheuristic methods have the advantages of free derivative information demand, easy implementation, high flexibility, high robustness and excellent ability to escape from local optima. Thus these metaheuristic methods have been successfully applied to solve the optimization problems in many fields, such as process control, task scheduling, image processing, electric power design and many other engineering design problems. Metaheuristic methods are well appreciated by scientific community and are becoming increasingly popular due to the characteristic of derivative-free mechanism and their superiority over conventional methods, especially in case of multimodal, discrete and non-differential complex optimization problems.

According to the thought origin and employed search mechanism, metaheuristic methods can be broadly classified into 6 main categories [1]:(1) breeding-based evolutionary algorithms, which mimic the evolution laws of natural species. These algorithms generally start with a randomly generated population, then a process of reproduction and survival iterated over successive generations enables the population to potentially evolve towards promising regions. The representatives include the genetic algorithm (GA) [2], differential evolution (DE) [3]; (2) swarm intelligence based algorithms, which simulate the collective behavior of decentralized self-organized systems, in either natural or artificial environments to effectively collect and utilize the obtained information about search space to guide the search of population during the process of iterations. The representatives of this category include particle swarm optimization (PSO) [4], grey wolf optimization (GWO) [5], cuckoo search (CS) [6], squirrel search algorithm (SSA) [7] and marine predators algorithm (ODMPA)[8]; (3) physics/chemistry based algorithms, which are inspired form the physical/chemical rules in universe. The representatives include gravitational search algorithm (GSA) [9], multi-verse optimizer (MVO) [10] and chemical reaction optimization algorithm (CRO) [11]; (4) social human behavior based algorithms, which mimic the intelligent social behaviors of human beings. The representatives include ideology algorithm (IA) [12] and most valuable player algorithm (MVPA) [13]; (5) plants based algorithms, such as forest optimization algorithms (FOA) [14]. (6) algorithms with miscellaneous sourcs of inspiration i.e. algorithms which do not fit in any of the previous categories, such as the Ying-Yang Pair Optimization (YYOP) [15] and so on.

For a metaheuristic algorithm, the search mechanism usually includes two aspects: exploration for discovering new regions in the search space, and exploitation for developing the potential of available promising areas [16]. However, it is hard to keep a proper balance between exploration and exploitation because of the randomness rooted in the search mechanism. As a result, no single optimizer is found to be adequate to optimally solve all optimization problems. This means that one method may have good performance in solving some problems but may suffer form degraded performance in solving other problems [17].

With the progress of technology, optimization problems in reality become more and more complex and high-dimensional. As the existing metaheuristic methods are used to solve these high-dimensional complex problems, they may get trapped into local optima and suffer from premature convergence, and thus fail to yield feasible solution. Therefore, it is of great importance to further develop some new and effective metaheuristic methods for real-world complex problems with unknown decision spaces. This practical necessity keeps the research field of metaheuristic methods open, allowing the improvement of existing methods and the proposition of novel methods for better optimization. Besides, extending the scope of application of metaheuristic algorithms to solve the problems in different fields is also the focus of many scholars.

As for the improvement of existing algorithms, because different operators in diverse algorithms have their own unique strengths, so it is natural to think that the combination of multiple operators in one algorithm framework may help to improve the overall performance of the algorithm. For example, in order to enhance the global search ability of the grey wolf optimizer (GWO) algorithm, the mutation and crossover operators are employed in the in the memory-based grey wolf optimizer (mGWO) [18]. Beside, chaotic mapping [19] and opposition-based learning [20] are introduced into the GWO algorithm to maintain population diversity and enhance the exploration capability. To improve the search efficiency of the sine cosine algorithm (SCA) [21], the optimal neighborhood update strategy and quadratic interpolation strategy are combined in the variant of the SCA algorithm[22]. Random replacement strategy and double adaptive weights are introduced into the (WOA) algorithm to enhance its convergence speed and exploration ability respectively [23]. The local search capability of the squirrel search algorithm (SSA) algorithm is improved significantly by embedding the reproduction mechanism of the invasive weed optimization (IWO) [24] in the ISSA algorithm [25]. Moreover, there are a large number of works on the improvement of the PSO algorithm, including the opposition-based particle swarm algorithm (OBPSO) [26], the synergy of the sine-cosine algorithm and particle swarm optimizer (SCA-PSO) [27], etc.

Cooperation search algorithm (CSA) [28] is a novel social human behavior based algorithm proposed by Feng in 2021, which simulates the efficient team cooperation behaviors and dynamic position updating mechanism in modern enterprises to achieve population evolution during the optimization procedure. Compared with most of the existing methods, the CSA has superiority performance in convergence rate, but as used to solve some high-dimensional complex problems, it may get trapped into local optimal just like most of the metaheuristic methods with quick convergence. This is because the movement direction of the search agents are almost entirely guided by leading individuals in the population. As a result, much of the other search information the search agents obtained is abandoned in the next iteration[29]. Thus these algorithms have a high probability to stuck in local optima and suffer from premature convergence.

For alleviating the premature convergence of the CSA, this paper proposes an improved cooperation search algorithm (CCSA) by incorporating a mutation and crossover operators in differential evolution methods. The CCSA employs the mutation operator to explore the neighborhood potential areas of personal best solutions to enhance population diversity. Then the crossover operator is employed to make full use of the excellent solution structure of individuals and strengthen the collaboration among individuals to improve population's global exploration capability and search accuracy.

Furthermore, aiming at the problem that there is no universal approach for the multi-degree reduction of Ball Bézier surfaces under different interpolation constrains, metaheuristic algorithms are introduced into the multi-degree reduction of Ball Bézier surfaces in this paper.

In the field of Computer Aided Design (CAD), there has been much research on curves and surfaces. However, objects without thickness do not exist in nature at all, and the curve/surface without thickness is just a kind of mathematical simplification. Besides, in recent years, 3D printing finds extensive applications in modern industry manufacture and also arouses great interests in many other fields. 3D printing turns 3D digital models into real objects by building them up layer by layer. With the development of multi-material 3D printer, 3D printing provides an appealing way of fabricating complex solid objects of the real world [30, 31], thus the modeling of various 3D solid objects has attracted considerable attention.

In order to effectively represent 3D freeform objects of uneven thickness, scholars presented Ball curves and surfaces, which are modeled based on skeleton and radius function. Theoretically, all solid objects can be modeled with Ball curves or surfaces, and thus they have extensive application in the modeling of objects with uneven thickness. For example, cerebral vascular [32], plant stems [33, 34], 3D human modeling [35] and other objects with uneven thickness [36] all can be modeled by Ball curves and surfaces. Since Ball curves and surfaces are defined by explicit parametric equations, the deformation of Ball curves and surfaces can be easily achieved by manipulating their control balls directly. Some scholars have done some work on Ball curves and surfaces. For instance, Wu et al. studied the properties [37], intersection [38] and extension [39] of Ball B-Spline curves. Hu and Wang [40] studied the boundary of Ball Bézier surface. Liu [41] proposed the fitting of scattered data with Ball B-spline curves/surfaces.

In the design and manufacture procedure of a product, the product data is usually transferred between several design and manufacturing systems, Since different systems may adopt different standards, these systems usually have different requirements on the form of curves and surfaces, such as the degree of curves and surfaces. As a result, for the efficient transmission a Ball curve/surface between different CAD/CAM systems, the multi-degree reduction of the Ball curve/surface is a crucial step. Therefore, Wu [42] presented the degree reduction of Ball Bézier surfaces over rectangular domain. Chen [43] proposed the degree reduction of Ball Bézier surfaces over triangular domain.

In the degree reduction definition of Ball surfaces, the inclusion constrain and the objective for minimum thickness are complex and not easy to deal with. Thus, for each kind of specific interpolation constrain, concrete analysis is required for the multi-degree reduction of Ball surfaces in the previous works. These tedious and complicated analysis and derivation lead to the lack of universal approach for the multi-degree reduction of Ball surfaces under different interpolation constrains. To solve this problem, we propose a new method for the multi-degree reduction of Ball Bézier surfaces by introducing metaheuristic methods, which can be used to deal with all the interpolation constrains, simplifying the degree reduction procedure significantly.

The arrangements for this paper are as follows: Section 2 describes the cooperation search algorithm; Section 3 presents the main inspiration for this paper and proposes the improved cooperation search algorithm combined with a crossover operation (CCSA); In Section 4, conceptual comparative analysis of CCSA with other metaheuristic algorithms are provided. In Section 5, the performance of CCSA is analyzed based on experimental results; In Section 6, the CCSA is applied to the optimal multi-degree reduction of Ball Bézier surfaces; Finally, conclusions and future work are discussed in section 7.

2. Cooperation search algorithm

As a novel social human behavior based algorithm, CSA algorithm simulates the efficient team cooperation behaviors and dynamic position updating mechanism in modern enterprises to realize the process of optimization.

2.1. Team cooperation behaviors in modern enterprises

The team cooperation behaviors and position updating mechanism in modern enterprises are introduced as follows. In general, an enterprise usually offers four different kinds of positions, including the board of directors, the board of supervisors, the chairman and the staff. The board of directors is in charge of the enterprise, whose members are elected from the individuals in the enterprise. The board of supervisors is supposed to exercise responsible supervision over executive directors for promoting the interests of shareholders. The chairman on duty is elected form the members in the board of directors, which is mainly responsible for the smooth and ordered running of the company and has very important influence on the enterprise. The staffs shall do specific jobs under the guidance of the leaders. The staffs are empowered to elect the leaders in the board of directors and the board of supervisors.

As we all know, human beings plays a key role in improving productivity. Thus, in order to achieve the scientific development of an enterprise and maintain its market competitiveness, it is essential to improve staff's strength, i.e., to help every staff acquire knowledge as much as possible. The knowledge of a staff is mainly affected by the chairman on duty for its highest position in the team. Besides, the leaders in the board of directors and the board of supervisors can also offer extensive information to staff. After a period of time, ordinary staffs are encouraged to promote their job positions through their good performance while the under-performing staffs and leaders may be replaced by staffs with better performance. This kind of team cooperation behaviors and dynamic position updating mechanism can help the company to maintain the initiative and thus to realize its sustainable development. Fig. 1 gives a schematic diagram to illustrate the team cooperation relationship in enterprises.

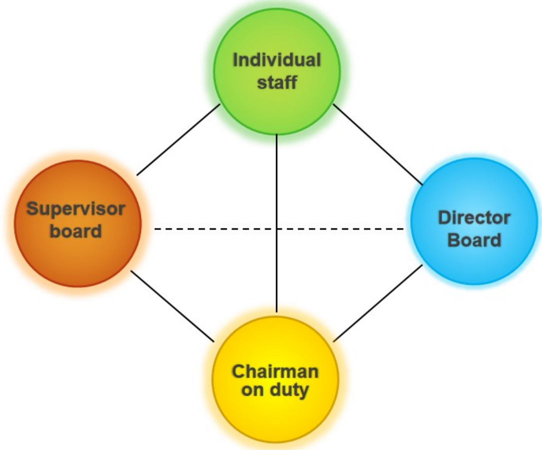

Fig. 1. Sketch map of the team relationship in modern enterprises.

2.2. Search principle of the CSA method

In the cooperation search algorithm, the optimization process of the problem under consideration is regarded as the development of an enterprise. In the optimization process, every solution is treated as a staff and a group of staffs constitute a company team. The fitness value of a solution for the target problem represents the performance of the corresponding staff. The board of directors and the board of supervisors are composed of M global best solutions the group has found by far (an external elite set) and N personal best solutions respectively. The chairman is randomly selected from the board of directors. The implementation process of the CSA is described as follows.

(1) **Team building phase**. For conducting uniform search in the initial phase, all the staffs in the team are randomly produced by Eq.(1). Then evaluate the fitness value of each solution, store the $M \in [1, N]$ number of global best solutions to constitute the external elite set $\{gbest_i^1\}_{i=1}^{M}$ and record the personal best state $pbest_i^1$ of each staff.

$$x_{i,j}^t = \phi(\underline{x}_j, \overline{x}_j), \quad i \in [1, N], \quad j \in [1, D], \quad t = 1, \tag{1}$$

where N denotes the population size, D is the number of decision variables, $x_{i,j}^t$ is the jth decision variable of the ith solution at tth iteration. $\phi(\underline{x}, \overline{x})$ represents a uniformly distributed random number in the interval $[\underline{x}, \overline{x}]$.

(2) **Team communication operator**. Staff could obtain new knowledge through exchanging information with the leaders. There are three sources of information for a staff: the knowledge A from the chairman on duty (namely a randomly selected member in the board of directors) and the collective knowledge B and C form the board of directors and supervisors respectively.

$$y_{i,j}^{t+1} = x_{i,j}^t + A_{i,j}^t + B_{i,j}^t + C_{i,j}^t, \; i \in [1, N], \; j \in [1, D], \; t \in [1, T], \tag{2}$$

$$A_{i,j}^t = \log(1/\phi(0,1)) \cdot (gbest_{cha,j}^t - x_{i,j}^t), \tag{3}$$

$$B_{i,j}^t = \alpha \cdot \phi(0,1)\big[\frac{1}{M}\sum_{m=1}^{M} gbest_{m,j}^t - x_{i,j}^t\big], \tag{4}$$

$$C_{i,j}^t = \beta \cdot \phi(0,1)\big[\frac{1}{N}\sum_{i=1}^{N} pbest_{i,j}^t - x_{i,j}^t\big], \tag{5}$$

where $y_{i,j}^{t+1}$ denotes the jth value of the ith communication solution at $t+1$th iteration. $gbest_{m,j}^t$ denotes the jth value of the mth global best solution that the population has found by far. $pbest_{i,j}^t$ is the jth value of the ith personal best solution at tth iteration. $A_{i,j}^t$ is the knowledge obtained from the chairman on duty and cha is a randomly selected index from the set $\{1, 2, \ldots, M\}$. $B_{i,j}^t$ is the mean knowledge obtained from the M number of global best solutions. $C_{i,j}^t$ is the mean knowledge obtianed from the N personal best solutions. α and β are learning coefficients, which are used to control the relevant effects of the corresponding subpopulations.

(3) **Reflective learning operator**. In addition to exchanging information with leaders, each staff could also obtain new message by summing its own experience in the opposite direction, which is represented as follows

$$z_{i,j}^{t+1} = \begin{cases} r_{i,j}^{t+1} & y_{i,j}^{t+1} \geq c_j \\ p_{i,j}^{t+1} & \text{otherwise} \end{cases} \quad i \in [1,N],\ j \in [1,D],\ t \in [1,T]. \tag{6}$$

$$r_{i,j}^{t+1} = \begin{cases} \phi(\bar{x}_j + \underline{x}_j - y_{i,j}^{t+1}, c_j) & |y_{i,j}^{t+1} - c_j| < \phi(0,1) \cdot |\bar{x}_j - \underline{x}_j| \\ \phi(\underline{x}_j, \bar{x}_j + \underline{x}_j - y_{i,j}^{t+1}) & \text{otherwise} \end{cases} \tag{7}$$

$$p_{i,j}^{t+1} = \begin{cases} \phi(c_j, \bar{x}_j + \underline{x}_j - y_{i,j}^{t+1}) & |y_{i,j}^{t+1} - c_j| < \phi(0,1) \cdot |\bar{x}_j - \underline{x}_j| \\ \phi(\bar{x}_j + \underline{x}_j - y_{i,j}^{t+1}, \bar{x}_j) & \text{otherwise} \end{cases} \tag{8}$$

$$c_j = \frac{\bar{x}_j + \underline{x}_j}{2}. \tag{9}$$

where $z_{i,j}^{t+1}$ is the jth value of the ith reflective solution at $t+1$th iteration.

(4) **Internal competition operator**. In order to guarantee that staffs with better performance could be conserved to promote the company's market competitiveness, the following competition operator is applied

$$\mathbf{x}_i^{t+1} = \begin{cases} \mathbf{y}_i^{t+1} & F(\mathbf{y}_i^{t+1}) \leq F(\mathbf{z}_i^{t+1}) \\ \mathbf{z}_i^{t+1} & \text{otherwise} \end{cases} \quad i \in [1,N], \quad t \in [1,T], \tag{10}$$

where $F(\mathbf{y})$ is the fitness value of the solution $\mathbf{y}$.

The pseudo-code of the CSA algorithm is shown in Table 1 and the sketch map of team communication mechanism is given in Fig. 2.

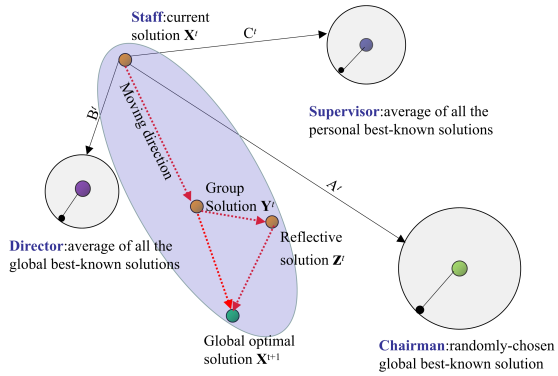

Fig. 2. Sketch map of the team communication mechanism in the CSA method.

Table 1: Pseudo-code of the CSA method.

Algorithm 1. Cooperation Search Algorithm (CSA).
Begin:
1. Initialize population $\mathbf{x}^1$ by Eq.(1).
2. Evaluate the fitness value of individuals.
While (the termination condition is not satisfied)
3. Update M global best solutions $\mathbf{gbest}_m^t$.
4. Update N personal best solutions $\mathbf{pbest}_i^t$.
5. Generate N communication solutions $\mathbf{y}_i^{t+1}$ by Eq.(4.3)-Eq.(5) for global exploitation.
6. Generate N reflective solutions $\mathbf{z}_i^{t+1}$ by Eq.(6)-Eq.(9) to increase population divesity.
7. Evaluate the fitness value of the communication and reflective solutions.
8. Apply Eq.(10) to select N better solutions for the next iteration.
end
9. Output the global best individual as the final optimal solution for the target problem.
End

3. Mutation and crossover based cooperation search algorithm (CCSA)

3.1. Motivation of the work

For an optimization problem with D decision variables, the fitness value of a solution is determined by the solution structure in all the D dimensions, that is

$$F(\mathbf{x}_i) = F(x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,D}). \tag{11}$$

It is natural to think that individuals whose fitness values are slightly worse may have good solution structure in some particular dimensions, and thus learning from these individuals can help a individual gain knowledge as much as possible [18, 44]. The search mechanism of CSA implies the dependency of the search directions on the leading individuals, which will cause the good solution structure that some slightly worse individuals hold in some dimensions to be neglected and cause the population to lose diversity, increasing the possibility of getting trapped at local optima and decreasing the convergence accuracy of population. Though the reflective learning operator can increase population diversity and thus can alleviate this issue to some extent, it still has some blindness in the search process.

In order to make full use of every individual's excellent solution structure during the search process and enhance the exploration and exploitation capability of the CSA, the CSA algorithm is modified by incorporating a DE/best/1 mutation operator and a binomial crossover operator, which is named CCSA.

3.2. Search principle of the CCSA algorithm

The search principle of the CCSA algorithm is described as follows.

(1) **Team building phase**. Generate a initial population to form the team of an enterprise by Eq.(1). Determine the leaders in the board of directors (M number of global best solutions $\{\mathbf{gbest}_m^1\}_{m=1}^M$) and the board of supervisors (N number of personal best solutions $\{\mathbf{pbest}_i^1\}_{i=1}^N$) as well as the chairman (a randomly selected solution $\mathbf{gbest}_{cha,j}^1$ from the M global best solutions).

(2) **Team communication operator**. Ordinary staffs learn from leaders in the board of directors and the board of supervisors as well as the chairman simultaneously to gain new knowledge by Eq.(4.3)-Eq.(5).

(3) **Mutation operator**. In order to strengthen the collaboration among individual staffs and explore the neighborhood potential areas of the personal best states of individual staffs, we utilize the personal best solution $\mathbf{pbest}_i^1$ of each individual and modify the DE/best/1 mutation operator in differential evolution algorithms as follows to help individual staffs gain new knowledge.

$$\mathbf{s}_i^{t+1} = \mathbf{pbest}_i^t + k(\mathbf{x}_{r_1}^t - \mathbf{x}_{r_2}^t), \quad i \in [1, N], \quad t \in [1, T]. \tag{12}$$

where $\mathbf{s}_i^{t+1}$ is the ith mutation solution, $\mathbf{x}_{r_1}$ and $\mathbf{x}_{r_2}$ are two randomly selected individuals in the current population. The learning coefficient k controls the influence of the difference vector. The high value of k supports extensive exploration and the small value of k encourages local exploitation. In this study, we set k as a variable, which is linearly decreased from the initial value 1.5 to the final value 0.

(4) **Crossover operator**. To make full use of the good solution structure of individual staffs, the binomial crossover operator in DE is utilized to merge the messages obtained from team communication and mutation operator, which is expressed as

$$g_{i,j}^{t+1} = \begin{cases} y_{i,j}^{t+1} & \phi(0,1) < CR \\ s_{i,j}^{t+1} & \text{otherwise} \end{cases} \tag{13}$$

where $g_{i,j}^{t+1}$ is the jth value of the ith crossover solution, CR is the crossover probability fixed as 0.5 in our study. $\mathbf{y}_i^{t+1}$ is the team communication solution obtained from Eq.(4.3) and $\mathbf{s}_i^{t+1}$ is the mutation solution obtained from Eq.(12).

(5) **Reflective learning operator**. Each individual gain new knowledge by summing its experience in the opposite direction, which is expressed as below:

$$z_{i,j}^{t+1} = \begin{cases} r_{i,j}^{t+1} & g_{i,j}^{t+1} \geq c_j \\ p_{i,j}^{t+1} & \text{otherwise} \end{cases} \quad i \in [1, N],\ j \in [1, D],\ t \in [1, T]. \tag{14}$$

$$r_{i,j}^{t+1} = \begin{cases} \phi(\bar{x}_j + \underline{x}_j - g_{i,j}^{t+1}, c_j) & |g_{i,j}^{t+1} - c_j| < \phi(0,1) \cdot |\bar{x}_j - \underline{x}_j| \\ \phi(\underline{x}_j, \bar{x}_j + \underline{x}_j - g_{i,j}^{t+1}) & \text{otherwise} \end{cases} \tag{15}$$

$$p_{i,j}^{t+1} = \begin{cases} \phi(c_j, \bar{x}_j + \underline{x}_j - g_{i,j}^{t+1}) & |g_{i,j}^{t+1} - c_j| < \phi(0,1) \cdot |\bar{x}_j - \underline{x}_j| \\ \phi(\bar{x}_j + \underline{x}_j - g_{i,j}^{t+1}, \bar{x}_j) & \text{otherwise} \end{cases} \tag{16}$$

$$c_j = \frac{\bar{x}_j + \underline{x}_j}{2}. \tag{17}$$

where $g_{i,j}^{t+1}$ is jth value of the ith crossover solution obtained from Eq.(13) and $z_{i,j}^{t+1}$ is the jth value of the ith reflective solution in $t+1$th iteration.

(6) **Greedy selection**. In an enterprise, staffs with better performance are usually conserved for promoting the company's market competitiveness. To realize this goal, the following greedy selection is applied

$$\mathbf{x}_i^{t+1} = \begin{cases} \mathbf{y}_i^{t+1} & F(\mathbf{y}_i^{t+1}) < min(F(\mathbf{g}_i^{t+1}), F(\mathbf{z}_i^{t+1})) \\ \mathbf{g}_i^{t+1} & F(\mathbf{g}_i^{t+1}) < min(F(\mathbf{y}_i^{t+1}), F(\mathbf{z}_i^{t+1})) \\ \mathbf{z}_i^{t+1} & F(\mathbf{z}_i^{t+1}) \leq min(F(\mathbf{y}_i^{t+1}), F(\mathbf{g}_i^{t+1})) \end{cases} \tag{18}$$

Table 2 gives the pseudo-code of the proposed CCSA algorithm.

Table 2: Pseudo-code of the CCSA algorithm.

Algorithm 2. CCSA Algorithm

Begin:

1. Initialize population $\mathbf{x}^1$ by Eq.(1).
2. Evaluate the fitness value of individuals.

While (the termination condition is not satisfied)

3. Update M global best solutions $\mathbf{gbest}_m^t$.
4. Update N personal best solutions $\mathbf{pbest}_i^t$.
5. Generate N communication solutions $\mathbf{y}_i^{t+1}$ by Eq.(4.3)-Eq.(5) for global exploitation.
6. Generate N mutation solutions $\mathbf{s}_i^{t+1}$ by Eq.(12) to strengthen collaboration.
7. Merge the solutions between $\mathbf{y}_i^{t+1}$ and $\mathbf{s}_i^{t+1}$ by Eq.(13) to generate N crossover solutions $\mathbf{g}_i^{t+1}$ to make full use of the good solution structure of individuals.
8. Generate N reflective solutions $\mathbf{z}_i^{t+1}$ by Eq.(14)-Eq.(17) to increase population diversity.
9. Evaluate the fitness of the team communicaiton, crossover and reflective solutions.
10. Use Eq.(18) to select N best solutions to form a new population.

end

12. Output the global best solution as the final optimal solution for the target problem.

End

The traits of the CCSA algorithm are summarized as below:

(1) In the team communication phase of CCSA, new direction for movement of a individual is guided by M Gbest and N Pbest and a randomly selected Gbest solution, i.e. cumulative effect of all three is considered, which can help the population find the promising regions and accelerate convergence effectively.

(2) The mutation operator implemented on personal best solution and the crossover operator can significantly increase population diversity, guaranteeing the exploration capability of the algorithm.

(3) The reflective learning operator enables the search of a individual to be transferred from one region to another, which can help population jump out of local optima. Compared with the well known Quasi-opposition learning strategy, in which the value range of the Quasi-opposite point of a point $g_{i,j}$ is $(\bar{x}_j + \underline{x}_j - g_{i,j}, \frac{\bar{x}_j+\underline{x}_j}{2})$ or $(\frac{\bar{x}_j+\underline{x}_j}{2}, \bar{x}_j + \underline{x}_j - g_{i,j})$, the value range of the reflective point is wider, and thus the reflective points of the population can be more dispersed. Consequently the CCSA possesses an excellent exploration ability.

(4) The greedy selection enables the population to seek out high-quality solutions for the next iteration to accelerate the convergence.

(5) The swarm is able to achieve an appropriate balance between global exploitation and local exploration via the operators above, increasing the probability of approximating the global optimal solution.

3.3. Time complexity analysis of CCSA

Time complexity is a key factor to evaluate the efficiency of an intelligent algorithm in solving problems. It is supposed that the population size N and T iterations are involved in the optimization procedure of a target problem with D decision variables. Besides, it is assumed that the fitness value evaluation per solution is much more complicated than that of other calculations. In the optimization process of the target problem with the CCSA algorithm, N solutions are evaluated in the original team building phase, then N team communication solutions, N crossover solutions and N reflective solutions are evaluated in each iteration. Therefore the total number of evaluations with the CCSA algorithm is $N + 3NT$.

4. Conceptual comparative analysis of CCSA with other metaheuristic algorithms

In general, all nature-inspired metaheuristics present two features i.e. adaptability and choice of the fittest, so nature-inspired metaheuristics apparently looks quite similar. In fact, their solution updating mechanism differs from each other. In this section, the updating mechanism of the proposed CCSA algorithm is compared with that of the particle swarm optimization (PSO), the genetic algorithms(GA), and the squirrel search algorithm (SSA).

4.1. CCSA versus PSO

Both the CCSA and PSO utilize the cumulative effect of global best information and personal best information to guide the evolution of the population, however technically they present several differences in their formulation and updating mechanism. Some of the major differences are described as follows:

(1) The PSO utilizes one global best solution while the CCSA utilizes M number of global best solutions to guide the search direction.

(2) In PSO, the solution updating of a individual $\mathbf{x}_i$ just involves its own personal best solution $\mathbf{pbest}_i$ and the global best solution $\mathbf{gbest}$, while in CCSA, the solution updating involves the collective knowledge form all personal best solutions $\{\mathbf{pbest}_i\}_{i=1}^{N}$ and M global best solutions $\{\mathbf{gbest}_i\}_{i=1}^{M}$. More importantly, a randomly selected global best solution $\mathbf{gbest}_{cha}$ plays an important and even conclusive role in the solution updating.

(3) The mutation and crossover operator as well as the reflective learning operator are employed in the CCSA to maintain population diversity, whereas PSO does not utilize them.

4.2. CCSA versus GA

The mutation and crossover operators are involved in both the CCSA and GA algorithms, but the meaning of the two operators in CCSA and the meaning of the two operators in GA are totally different.

(1) In GA, all the updating operators are implemented on chromosomes encoded by an array of parameter values. In contrary, for CCSA all the updating operators are implemented on the parameter values directly.

(2)The crossover operator in GA swaps a subsequence of two of the chosen chromosomes to create two offspring. In CCSA, for two given solutions, a crossover solution is generated by randomly choose each of its parameter value form two candidate solutions.

(3) The mutation operator in GA means that randomly flips individual bits in the chromosomes (turning a 0 into a 1 and vice versa). While in CCSA, the mutation operator means probabilistic neighbourhood selection, i.e. basically replaces one solution vector by an arithmetic recombination of the solution vector and the difference vector of two randomly selected solution vectors.

4.3. CCSA versus SSA

Squirrel search algorithm is a novel nature-inspired metaheuristic algorithm, inspired by the dynamic foraging behaviour of flying squirrels and their efficient gliding flight. The typical assumptions of the SSA algorithm are:

(a) There are N number of flying squirrels in forest, each of which is considered to be on a tree and searches for food individually.

(b) There are three kinds of trees in the forest: one hickory tree FS_{ht} (hickory nuts food source), three acorn trees FS_{at} (acorn nuts food source) and the remaining normal trees FS_{nt}.

The position updating mechanism of the squirrel search algorithm include the following three cases:

Case 1. Flying squirrels on acorn trees (FS_{at}^t) may move towards the hickory tree (FS_{ht}^t), which can be represented as:

$$FS_{at}^{t+1} = \begin{cases} FS_{at}^t + d_g \times G_c \times (FS_{ht}^t - FS_{at}^t), & R_1 \geq P_{dp}, \\ \text{Random location}, & \text{otherwise.} \end{cases}$$

where d_g is a gliding distance, G_c is a gliding constant that controls the balance between exploration and exploitation of SSA, R_1 is a random number uniformly distributed in [0,1],

Table 3: Unimodal benchmark functions.

Function	Dim	Lb	Ub	f_{min}
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	-100	100	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	-10	10	0
$F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_i)^2$	30	-100	100	0
$F_4(x) = \max_i \lvert x_i \rvert, 1 \leq i \leq n$	30	-100	100	0
$F_5(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	30	-30	30	0
$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	-100	100	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0, 1)$	30	-1.28	1.28	0

P_{dp} is the predator presence probability.

Case 2. Some of the flying squirrels on normal trees (FS_{nt}^t) may move towards the acorn trees for food, which can be represented as:

$$FS_{nt}^{t+1} = \begin{cases} FS_{nt}^t + d_g \times G_c \times (FS_{at}^t - FS_{nt}^t), & R_2 \geq P_{dp}, \\ \text{Random location}, & \text{otherwise}. \end{cases}$$

where R_2 is a random number uniformly distributed in [0,1].

Case 3. The remaining flying squirrels on normal trees (FS_{nt}^t) may move towards the hickory tree (FS_{ht}^t) for food, which can be represented as:

$$FS_{nt}^{t+1} = \begin{cases} FS_{nt}^t + d_g \times G_c \times (FS_{ht}^t - FS_{nt}^t), & R_3 \geq P_{dp}, \\ \text{Random location}, & \text{otherwise}. \end{cases}$$

where R_3 is a random number uniformly distributed in [0,1].

Both the CCSA and SSA work on effective division of labour, which gives them a similar appearance superficially, however their population updating mechanisms are entirely different.

(1)In SSA, the pattern matrix is divided into three regions and SSA employs three strategies in different regions of pattern matrix. On the contrary, in CCSA, the whole pattern matrix utilizes a universal updating strategy, which is much easier to implement.

(2)In the mutation operator of CCSA, the difference information of two randomly selected individuals is used to increase population diversity. While in SSA, this goal is achieved by introducing predator presence, which is modelled using a probabilistic behaviour.

5. Numerical experiments

5.1. Numerical experiments on 23 classic benchmark functions

In this section, experimental studies are carried out on 23 well known benchmark functions [22] to verify the performance of the CCSA algorithm. These benchmark functions include the unimodal functions F1-F7 in Table 3, the multimodal functions F8-F13 in Table 4, and the fixed dimensional multimodal functions F14-F23 in Table 5.

The optimization result of the proposed CCSA algorithm is compared with those of 6 existing metaheuristic algorithms, including the PSO [4], GWO [5], mGWO [18], QISCA [22], SSA[7] and the CSA algorithm. For the sake of fairness, each algorithm will run on each

Table 4: Multimodal benchmark functions.

Function	Dim	Lb	Ub	f_{min}
$F_8(x) = \sum_{i=1}^{n} -x_i sin(\sqrt{\lvert x_i \rvert})$	30	-500	500	-418.9829*d
$F_9(x) = \sum_{i=1}^{n}(x_i^2 - 10\cos 2\pi x_i + 10)$	30	-5.12	5.12	0
$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	30	-32	32	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30	-600	600	0
$F_{12}(x) = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\}$ $+\sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i+1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	-100	100	0
$F_{13}(x) = 0.1\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]$ $+\sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	-50	50	0

Table 5: Fixed-dimension multimodal benchmark functions.

Function	Dim	Lb	Ub	f
$F_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	2	-65	65	1
$F_{15}(x) = \sum_{i=1}^{11}[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]$	3	-5	5	0.0003
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	-5	5	-1.0316
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	-5	5	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 16x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2$ $\times(18 - 32x_1 + 12x^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	-2	2	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i\exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	0	1	-3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i\exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	0	1	-3.2
$F_{21}(x) = -\sum_{i=1}^{5}[(\mathbf{X} - a_i)(\mathbf{X} - a_i)^T + c_i]^{-1}$	4	0	10	-10.1532
$F_{22}(x) = -\sum_{i=1}^{7}[(\mathbf{X} - a_i)(\mathbf{X} - a_i)^T + c_i]^{-1}$	4	0	10	-10.4028
$F_{23}(x) = -\sum_{i=1}^{10}[(\mathbf{X} - a_i)(\mathbf{X} - a_i)^T + c_i]^{-1}$	4	0	10	-10.5363

Table 6: Parametric settings of the 10 algorithms.

Algorithms	Parameters	Value
PSO	w, C_1, C_2	0.9, 2, 2
GWO	$a_{\min}, a_{\max}$	0, 2
mGWO	$a_{\min}, a_{\max}, k$	0, 2, 1→0
QISCA	a, δ	2, [0.001, 0.01]
SSA	n_1, n_2, G_c, P_{dp}	3, 9, 1.9, 0.1
CSA	α, β, M	0.1, 0.15, 3
OBPSO	w, C_1, C_2	0.72984, 1.49618, 1.49618
ISSA	$n_1, n_2, G_c, P_{dp}, max_{seed}, min_{seed}$	3, 9, 1.9, 0.1, 5, 2
SCA-PSO	$w, C_1, C_2, r_2, r_3, r_4, \alpha$	0.9 → 0.4, 2, 2, 2π·rand, 2·rand, 2
CCSA	α, β, M, k	0.1, 0.15, 3, 1.5→0

benchmark function independently for 20 times and the values of these parameters are set as suggested by their authors. Table 6 displays the parameter configuration of these algorithms. Simulation experiment configuration is Matlab-2016b, Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz, 16GB.

In the comparative experiments, the population size N and the maximum iteration number T in these algorithms are set as 50 and 500 respectively. The statistical results of these algorithms in experiments are displayed Table 7. Here, we mark the best and the second best result obtained by those algorithms on each test function with bold font and with underlined respectively. To evaluate the advantages and disadvantages of these algorithms, the last row of Table 7 displays the Wilcoxon rank-sum-test results of the comparison algorithms, where the "$+/=/-$" in each comparison algorithm represents its number of functions "superior/comparable/inferior" to the CCSA method.

From Table 7, it can be found that the CCSA method achieves the best or sub-best result on 21 out of 23 benchmark functions, 15 of which are the best result. Besides, the CCSA uniformly converges to the theoretical optimum in 20 independent experiments in F1, F3, F9, F11, F17. Table 8 provides the detection value P in the Wilcoxon rank-sum-test, where the boldface data implies that there is no obvious difference between the CCSA and the corresponding comparison algorithm when $\alpha = 0.05$.

In order to reflect the search efficiency of each algorithm more intuitively, the convergence trajectory of the 7 methods for several benchmark functions are shown in Fig. 3. It can be seen from Fig. 3 that the CCSA algorithm can quickly seek out the most satisfying solutions during the early iterations compared with the other algorithms; At the end of iteration, the CCSA algorithm can usually yield a solution of the best quality among these algorithms. This might because the team communication operator enables the algorithm to conduct global exploration and determine the promising search area quickly. While the crossover and reflective learning operator enable the algorithm to do exploration in different areas within the search space, which are helpful for the algorithm to jump out of local optima as well as to enhance convergence speed and solution accuracy. These results show that the performance of the CCSA algorithm is superior to that of the other comparison algorithms in terms of convergence speed and solution accuracy.

To explore the distributions of optimal values obtained in different runs, Fig. 4 shows the Box plot of the 7 algorithms for several benchmark functions. As shown in Fig. 4 that the objective distribution range of the CCSA algorithm is smaller than the other algorithms,

Table 7: Statistical results of 7 metaheuristic methods for 23 classic benchmark functions.

Function		Algorithm						
		PSO	SSA	GWO	mGWO	QISCA	CSA	CCSA
F1	Ave	7.7836E+00	5.4296E-05	2.1329E-59	1.4127E-26	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Std	3.3028E+00	3.3612E-05	4.5923E-59	3.0036E-26	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
F2	Ave	3.9247E+00	7.3560E+01	2.3177E-35	4.5441E-16	7.7740E-188	3.1952E-216	**6.6372E-220**
	Std	2.0989E+00	6.0385E+01	3.0479E-35	3.4568E-16	0.0000E+00	0.0000E+00	**0.0000E+00**
F3	Ave	2.8314E+02	2.8885E+01	4.2538E-15	2.3232E-02	2.0487E-298	**0.0000E+00**	**0.0000E+00**
	Std	1.0976E+02	1.3363E+01	1.5697E-14	2.7145E-02	0.0000E+00	**0.0000E+00**	**0.0000E+00**
F4	Ave	8.7519E+00	7.2140E+00	6.9181E-16	5.0644E-04	2.0241E-175	3.2451E-205	**1.8149E-205**
	Std	2.7523E+00	2.2830E+00	1.1455E-15	5.6985E-04	0.0000E+00	0.0000E+00	**0.0000E+00**
F5	Ave	3.0636E+02	6.4232E+01	2.8068E+01	**2.6197E+01**	2.7495E+01	2.8612E+01	2.7291E+01
	Std	1.8358E+02	5.6696E+01	6.6778E-01	**4.8425E-01**	5.6708E-01	1.8116E-01	4.2872E-01
F6	Ave	8.8800E+01	1.9000E+01	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	2.9333E+00	1.3333E-01
	Std	5.1813E+01	5.6821E+00	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	1.3345E+00	3.5187E-01
F7	Ave	4.8963E-02	3.6341E-02	3.0317E-04	3.7144E-03	**1.6463E-05**	2.7970E-04	2.2548E-04
	Std	2.2267E-02	1.3369E-02	3.4566E-04	1.8799E-03	**2.0098E-05**	2.5527E-04	2.1723E-04
F8	Ave	-6.0999E+03	-7.3678E+03	-5.0509E+03	-5.0958E+03	**-8.3591E+03**	-4.3425E+03	-8.1156E+03
	Std	6.6475E+02	1.3540E+03	9.1746E+02	3.9819E+02	**8.3722E+02**	1.0639E+03	5.5188E+02
F9	Ave	3.7309E+01	1.4781E+02	**0.0000E+00**	2.5090E+01	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Std	1.2738E+01	4.1612E+01	**0.0000E+00**	1.1142E+01	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
F10	Ave	4.4481E+00	9.6723E+00	6.3400E-15	5.6800E-01	**8.8800E-16**	**8.8800E-16**	**8.8800E-16**
	Std	9.0549E-01	6.4586E+00	1.8300E-15	1.1900E+00	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
F11	Ave	1.0955E+00	1.4279E-02	**0.0000E+00**	5.2064E-03	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Std	5.9029E-02	8.2055E-03	**0.0000E+00**	8.0194E-03	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
F12	Ave	4.3787E+00	7.8472E+00	2.0821E-01	1.6587E-02	2.4410E-02	3.0744E-01	**3.7500E-04**
	Std	1.9901E+00	2.9781E+00	7.5396E-02	1.3047E-02	6.2998E-03	6.7840E-02	**7.7840E-04**
F13	Ave	3.2444E+01	2.0035E+01	2.0774E+00	**1.8312E-01**	9.5163E-01	2.3310E+00	6.5051E-01
	Std	1.6755E+01	1.2957E+01	3.8634E-01	**1.7186E-01**	1.6426E-01	4.2832E-01	1.6527E-01
F14	Ave	2.5803E+00	9.9800E-01	8.2294E+00	9.9800E-01	3.0885E+00	5.1962E+00	**9.9800E-01**
	Std	1.8173E+00	2.3700E-16	4.9016E+00	8.8600E-13	3.6423E+00	3.8378E+00	**1.4500E-16**
F15	Ave	**3.0749E-04**	8.7100E-04	3.0952E-03	1.6972E-03	7.4920E-04	1.8598E-03	7.2750E-04
	Std	**3.5900E-18**	3.1986E-04	7.0184E-03	5.1678E-03	2.2510E-04	4.3296E-03	2.1800E-04
F16	Ave	**-1.0316E+00**	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0314E+00	-1.0316E+00
	Std	**5.9300E-17**	2.1400E-16	1.0400E-08	1.8000E-09	7.4500E-10	8.1300E-04	5.6300E-07
F17	Ave	**3.9789E-01**	**3.9789E-01**	3.9789E-01	3.9789E-01	3.9789E-01	3.9789E-01	**3.9789E-01**
	Std	**0.0000E+00**	**0.0000E+00**	2.6100E-07	4.2500E-08	6.7400E-08	2.7300E-06	**0.0000E+00**
F18	Ave	**3.0000E+00**	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00
	Std	**1.4200E-15**	4.6100E-15	8.4300E-06	2.4000E-08	1.5700E-07	1.3700E-08	2.1100E-15
F19	Ave	-3.8628E+00	-3.8628E+00	-3.8611E+00	-3.8628E+00	-3.8612E+00	-3.8628E+00	**-3.8628E+00**
	Std	7.1200E-16	5.2900E-15	2.9708E-03	3.9700E-07	2.7809E-03	9.6900E-06	**6.5000E-16**
F20	Ave	**-3.2903E+00**	-3.2583E+00	-3.2619E+00	-3.2545E+00	-3.2233E+00	-3.2299E+00	**-3.2903E+00**
	Std	**5.4422E-02**	6.1659E-02	9.3191E-02	5.8808E-02	6.4620E-02	6.9585E-02	**5.4422E-02**
F21	Ave	-6.4560E+00	-7.3018E+00	-9.4759E+00	-1.0153E+01	-4.3085E+00	-9.5361E+00	**-1.0153E+01**
	Std	2.8289E+00	3.2689E+00	1.7858E+00	2.6809E-04	2.0183E+00	1.9400E+00	**1.7100E-15**
F22	Ave	-9.0321E+00	-8.3298E+00	-1.0402E+01	-1.0402E+01	-4.8248E+00	-9.5066E+00	**-1.0403E+01**
	Std	2.8846E+00	3.1126E+00	5.5575E-04	3.1476E-04	3.0357E+00	2.3477E+00	**2.9600E-15**
F23	Ave	-7.3785E+00	-9.1006E+00	-1.0536E+01	-1.0536E+01	-3.0255E+00	-8.4707E+00	**-1.0536E+01**
	Std	3.5980E+00	2.4647E+00	6.0761E-04	2.8625E-04	6.6967E-01	3.5551E+00	**3.1800E-15**
+/=/−		1/5/17	1/3/19	0/4/19	2/1/20	1/8/14	0/8/15	

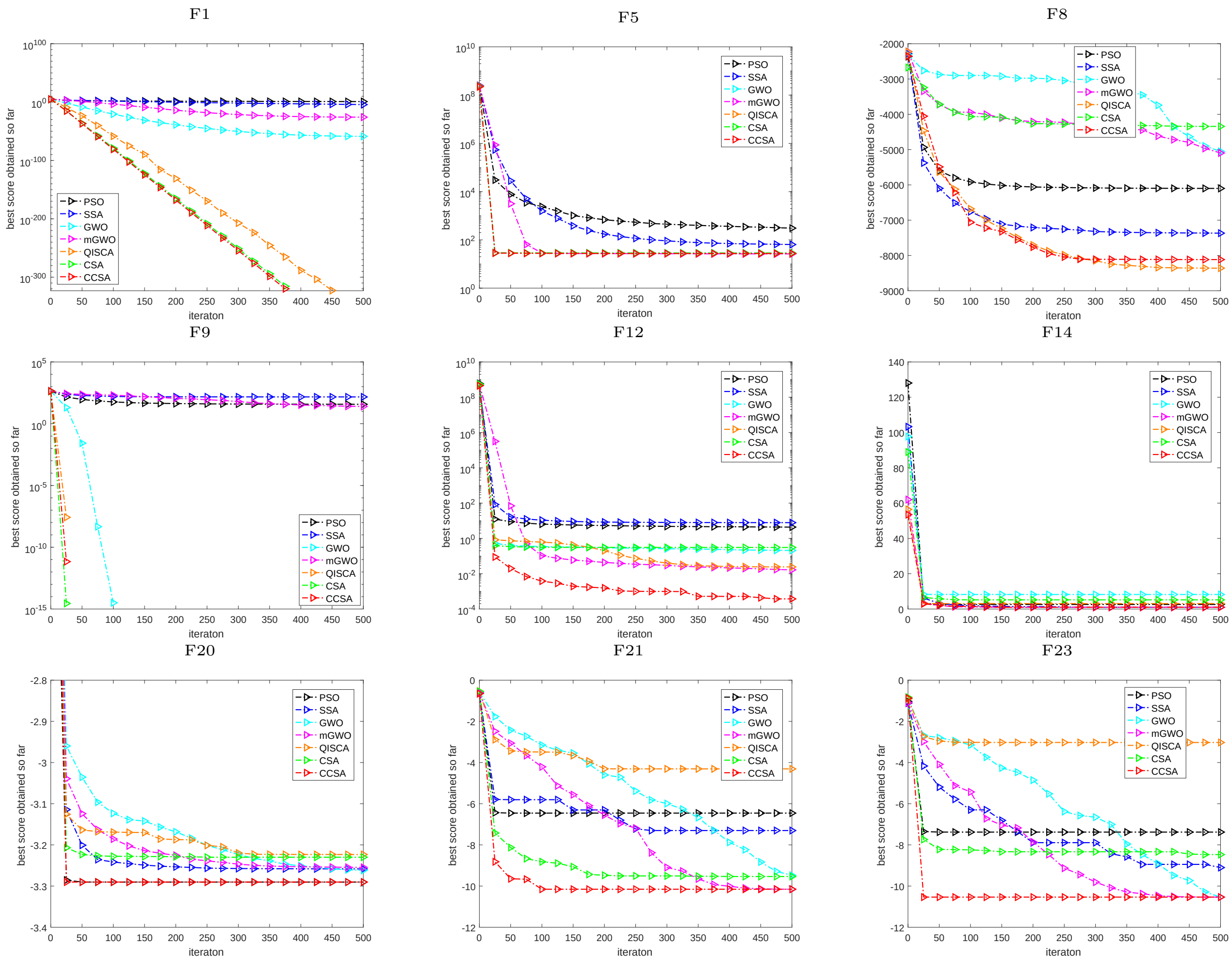

Fig. 3. Comparison of convergence curves of 7 algorithms for some of the 23 benchmark functions.

Table 8: P-value obtained from Wilcoxon rank-sum test on 23 classic benchmark functions.

Function	PSO	SSA	GWO	mGWO	QISCA	CSA
	P	P	P	P	P	P
F1	6.9E-07	6.9E-07	6.9E-07	6.9E-07	**NaN**	**NaN**
F2	3.4E-06	3.4E-06	3.4E-06	3.4E-06	3.4E-06	3.4E-06
F3	6.9E-07	6.9E-07	6.9E-07	6.9E-07	**3.5E-01**	**NaN**
F4	3.4E-06	3.4E-06	3.4E-06	3.4E-06	3.4E-06	3.1E-02
F5	3.4E-06	1.1E-03	1.9E-03	3.4E-05	8.9E-02	2.8E-05
F6	1.3E-06	1.2E-06	**1.6E-01**	**1.6E-01**	**1.6E-01**	6.2E-06
F7	3.4E-06	3.4E-06	**4.6E-01**	3.4E-06	7.5E-06	**4.1E-01**
F8	2.5E-04	**2.4E-01**	1.8E-04	1.8E-04	**1.9E-01**	1.8E-04
F9	6.9E-07	6.9E-07	**NaN**	6.9E-07	**NaN**	**NaN**
F10	6.9E-07	6.9E-07	4.3E-07	6.8E-07	**NaN**	**NaN**
F11	6.9E-07	6.9E-07	**NaN**	1.8E-02	**NaN**	**NaN**
F12	3.4E-06	3.4E-06	3.4E-06	5.7E-05	3.4E-06	3.4E-06
F13	3.4E-06	3.4E-06	3.4E-06	7.5E-06	1.6E-04	3.4E-06
F14	6.0E-04	1.9E-02	1.6E-06	1.6E-06	1.6E-06	1.7E-06
F15	3.4E-06	**2.5E-01**	7.0E-03	5.8E-04	**9.7E-01**	**6.2E-01**
F16	**1.0E+00**	8.5E-05	2.2E-05	2.2E-05	2.2E-05	1.0E-02
F17	**NaN**	**NaN**	6.9E-07	6.9E-07	6.9E-07	**3.5E-01**
F18	**8.0E-01**	1.1E-04	2.9E-06	2.9E-06	2.9E-06	6.8E-03
F19	**2.9E-01**	1.2E-05	2.1E-06	2.1E-06	2.1E-06	1.4E-05
F20	**6.0E-01**	3.5E-04	2.2E-03	6.6E-04	4.4E-05	8.9E-05
F21	3.5E-05	3.1E-06	1.3E-06	1.3E-06	1.3E-06	1.3E-06
F22	6.0E-03	1.7E-05	2.4E-06	2.4E-06	2.4E-06	2.4E-06
F23	2.1E-03	1.0E-04	2.1E-06	2.1E-06	2.1E-06	2.1E-06

demonstrating the strong robust of the CCSA algorithm.

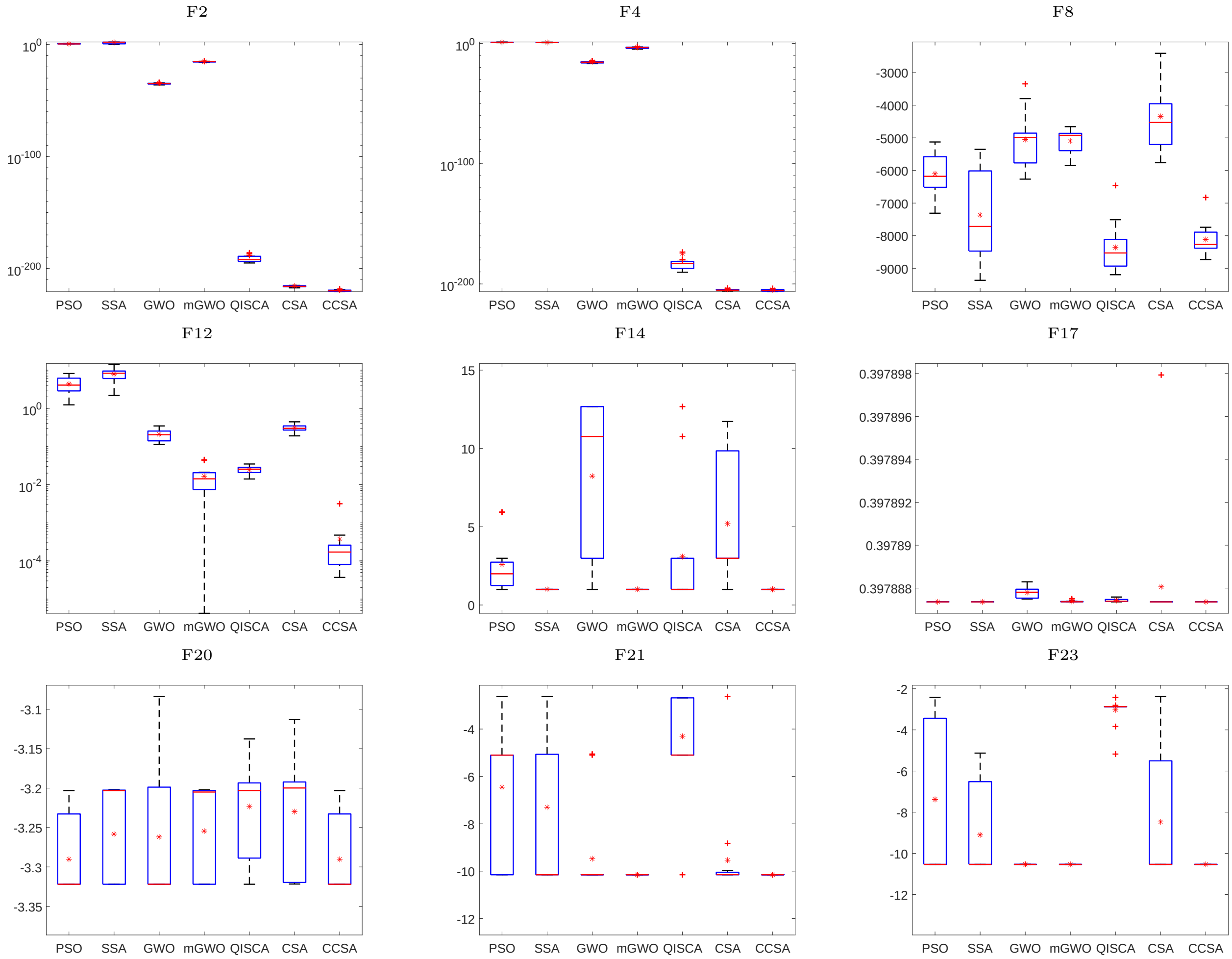

Fig. 4. Box plot of 7 algorithms for some of the 23 benchmark functions.

5.2. Numerical experiments on Benchmark suite: CEC2017

To fully verify the performance of the CCSA algorithm on solving high-dimensional complex optimization problems, we use the CCSA algorithm to solve the latest collection of benchmarks CEC 2017 [45] and compares the results with 6 latest metaheuristic algorithms, including the OBPSO[26], SCA-PSO [27], QISCA [22], mGWO [18], ISSA [25] and the CSA algorithm. In the CEC 2017 test, the population size N and the maximum iteration number T are set as 50 and 1000 respectively. Table 9 shows the average Ave, standard deviation and the sorting rank of the 7 algorithms running independently for 20 independent runs in the 50 dimensions. The second row to the bottom of Table 9 provides the total ranking of the 29 test functions for each algorithm and the last row provides the total running time of each algorithm. Table 10 shows the detection value P in the Wilcoxon rank-sum-test with the significance level $\alpha = 0.05$. Besides, Fig. 5 and Fig. 6 display the comparison of convergence curves and Box plot of 9 test functions solved by the 7 algorithms.

As shown in Table 9, the CCSA algorithm performs best on 19 out of 29 test functions with a total ranking 47. While the total ranking of the comparison algorithm OBPSO, SCA-PSO, QISCA, mGWO, NOSSA and CSA are 174, 100, 88, 117, 100, 186 respectively. Besides, the Wilcoxon rank-sum-test result of the 6 comparison algorithms are 0/0/29, 3/0/26, 5/0/24, 4/0/25, 6/0/23, 0/0/29 respectively. Fig. 5 and Fig. 6 reveal that the CCSA has a fast convergence rate and a high convergence accuracy. It is clear from the obtained results that the CCSA outperforms other comparison algorithms significantly.

The tests results on 23 classic benchmark functions and 29 CEC 2017 benchmark functions demonstrate that the CCSA has excellent local exploitation performance in dealing with unimodal problems and has the ability to jump out of local optima in dealing with multimodal problems. Besides, as shown in Table 7-Table 9, the results obtained by the CCSA algorithm are much better than those obtained by the CSA in solving multimodal problems, indicating that the exploration capability of the CSA algorithm is greatly improved after incorporating these modifications. This may because the probabilistic neighbourhood selection in mutation operator and the merging of different dimensional information of various solutions in crossover operator can help population to find more excellent solutions in different areas, and thus the population has a high probability of jumping out of local optima. Consequently, the CCSA can yield better optimization result than other comparison algorithms in general. Given the excellent optimization performance of the proposed CCSA algorithm, we apply the CCSA algorithm to solve the multi-degree reduction problems of Ball Bézier surfaces.

Table 9: Statistical results of CCSA and 6 metaheuristic methods for CEC 2017 benchmark suite.

F		OBPSO	SCA-PSO	QISCA	SSA	ISSA	CSA	CCSA
F1	Ave	4.797E+10	7.038E+09	5.052E+08	7.368E+04	**4.470E+03**	4.794E+10	3.444E+07
	STD	9.218E+09	3.216E+09	6.584E+08	5.243E+04	**6.172E+03**	6.430E+09	1.124E+08
	Rank	7	5	4	2	1	6	3
F3	Ave	1.974E+05	5.722E+04	9.256E+04	2.722E+04	**1.328E+03**	2.234E+05	1.396E+05
	STD	5.029E+04	1.382E+04	1.538E+04	9.672E+03	**1.155E+03**	5.397E+04	4.701E+04
	Rank	6	3	4	2	1	7	5
F4	Ave	9.318E+03	8.435E+02	7.106E+02	6.340E+02	6.090E+02	1.124E+04	**5.412E+02**
	STD	3.332E+03	1.177E+02	7.940E+01	3.978E+01	3.604E+01	2.296E+03	**4.063E+01**
	Rank	6	5	4	3	2	7	1
F5	Ave	1.055E+03	8.306E+02	7.975E+02	9.907E+02	1.037E+03	9.826E+02	**7.263E+02**
	STD	7.137E+01	5.632E+01	4.699E+01	8.882E+01	9.160E+01	4.441E+01	**2.964E+01**
	Rank	7	3	2	5	6	4	1
F6	Ave	6.766E+02	6.437E+02	**6.357E+02**	6.787E+02	6.815E+02	6.776E+02	6.360E+02
	STD	6.454E+00	7.678E+00	**6.405E+00**	1.073E+01	1.063E+01	7.252E+00	7.662E+00
	Rank	4	3	1	6	7	5	2
F7	Ave	2.063E+03	1.237E+03	1.311E+03	1.615E+03	1.613E+03	1.588E+03	**1.185E+03**
	STD	1.860E+02	7.913E+01	9.060E+01	1.351E+02	2.250E+02	1.082E+02	**1.065E+02**
	Rank	7	2	3	6	5	4	1
F8	Ave	1.355E+03	1.151E+03	1.118E+03	1.316E+03	1.320E+03	1.313E+03	**1.049E+03**
	STD	4.320E+01	3.595E+01	4.508E+01	7.761E+01	8.025E+01	4.152E+01	**4.375E+01**
	Rank	7	3	2	5	6	4	1
F9	Ave	2.072E+04	1.569E+04	1.701E+04	2.025E+04	1.958E+04	2.376E+04	**7.195E+03**
	STD	5.086E+03	7.200E+03	2.953E+03	4.691E+03	4.950E+03	5.135E+03	**1.970E+03**
	Rank	6	2	3	5	4	7	1
F10	Ave	1.138E+04	9.625E+03	8.274E+03	9.511E+03	8.971E+03	1.322E+04	**7.915E+03**
	STD	1.065E+03	1.012E+03	8.717E+02	1.058E+03	1.141E+03	1.155E+03	**1.854E+03**
	Rank	6	5	2	4	3	7	1
F11	Ave	1.032E+04	2.282E+03	3.654E+03	1.495E+03	1.460E+03	1.667E+04	**1.335E+03**
	STD	4.698E+03	3.540E+02	8.699E+02	9.491E+01	8.942E+01	3.212E+03	**9.391E+01**
	Rank	6	4	5	3	2	7	1
F12	Ave	8.065E+09	7.461E+08	8.488E+07	7.902E+07	6.242E+07	1.891E+10	**1.467E+07**
	STD	4.737E+09	6.183E+08	4.544E+07	3.903E+07	3.156E+07	6.020E+09	**8.234E+06**
	Rank	6	5	4	3	2	7	1
F13	Ave	2.625E+09	1.933E+08	1.124E+06	1.488E+05	2.555E+05	7.886E+09	**1.263E+05**
	STD	4.097E+09	2.649E+08	5.849E+05	8.016E+04	1.555E+05	4.027E+09	**6.222E+04**
	Rank	6	5	4	2	3	7	1
F14	Ave	2.525E+06	4.268E+05	7.798E+05	3.166E+05	**9.379E+04**	2.415E+07	2.357E+05
	STD	3.500E+06	2.940E+05	4.806E+05	3.033E+05	**5.676E+04**	1.326E+07	2.471E+05
	Rank	6	4	5	3	1	7	2
F15	Ave	4.557E+07	1.314E+07	1.504E+05	1.129E+05	9.292E+04	5.814E+08	**4.222E+04**
	STD	1.287E+08	2.009E+07	6.497E+04	9.969E+04	8.443E+04	6.439E+08	**1.919E+04**
	Rank	6	5	4	3	2	7	1
F16	Ave	4.934E+03	3.637E+03	4.096E+03	4.821E+03	4.424E+03	5.975E+03	**3.551E+03**
	STD	4.816E+02	4.079E+02	5.789E+02	1.220E+03	5.602E+02	8.029E+02	**4.451E+02**
	Rank	6	2	3	5	4	7	1
F17	Ave	4.018E+03	3.285E+03	3.530E+03	3.907E+03	4.010E+03	4.105E+03	**3.209E+03**
	STD	4.673E+02	2.886E+02	4.293E+02	3.724E+02	4.174E+02	5.326E+02	**3.863E+02**
	Rank	6	2	3	4	5	7	1
F18	Ave	7.140E+06	4.147E+06	4.338E+06	2.558E+06	**9.530E+05**	5.112E+07	1.674E+06
	STD	1.519E+07	2.610E+06	2.930E+06	1.638E+06	**7.836E+05**	3.803E+07	1.182E+06
	Rank	6	4	5	3	1	7	2
F19	Ave	6.823E+07	8.083E+06	1.579E+05	2.955E+06	1.941E+06	1.409E+08	**7.170E+04**
	STD	1.906E+08	9.817E+06	9.589E+04	1.921E+06	9.987E+05	1.765E+08	**4.743E+04**
	Rank	6	5	2	4	3	7	1
F20	Ave	3.607E+03	3.408E+03	3.043E+03	3.804E+03	3.614E+03	3.905E+03	**2.987E+03**
	STD	3.375E+02	3.037E+02	2.867E+02	2.995E+02	3.748E+02	3.597E+02	**3.420E+02**
	Rank	4	3	2	6	5	7	1
F21	Ave	2.923E+03	2.641E+03	2.641E+03	2.882E+03	2.918E+03	2.888E+03	**2.537E+03**
	STD	7.636E+01	5.417E+01	6.515E+01	9.167E+01	1.195E+02	6.764E+01	**4.358E+01**
	Rank	7	2	3	4	6	5	1
F22	Ave	1.326E+04	1.071E+04	**8.855E+03**	1.180E+04	1.034E+04	1.502E+04	9.602E+03
	STD	9.650E+02	1.183E+03	**2.836E+03**	1.803E+03	1.116E+03	1.493E+03	1.937E+03
	Rank	6	4	1	5	3	7	2
F23	Ave	3.809E+03	3.078E+03	3.120E+03	3.767E+03	3.781E+03	4.125E+03	**3.014E+03**
	STD	1.724E+02	5.882E+01	6.740E+01	2.215E+02	2.267E+02	2.463E+02	**5.353E+01**
	Rank	6	2	3	4	5	7	1
F24	Ave	4.064E+03	3.216E+03	3.307E+03	4.055E+03	3.885E+03	4.456E+03	**3.154E+03**
	STD	2.039E+02	6.574E+01	6.894E+01	2.389E+02	1.385E+02	1.708E+02	**6.264E+01**
	Rank	6	2	3	5	4	7	1
F25	Ave	7.675E+03	3.401E+03	3.234E+03	3.104E+03	**3.058E+03**	7.919E+03	3.120E+03
	STD	1.898E+03	1.784E+02	6.788E+01	3.438E+01	**3.335E+01**	9.966E+02	2.640E+01
	Rank	6	5	4	2	1	7	3
F26	Ave	1.444E+04	7.241E+03	**6.590E+03**	1.572E+04	1.537E+04	1.358E+04	8.620E+03
	STD	1.754E+03	4.241E+02	**2.045E+03**	2.606E+03	2.565E+03	1.148E+03	1.609E+03
	Rank	5	2	1	7	6	4	3
F27	Ave	4.474E+03	**3.404E+03**	3.487E+03	4.158E+03	3.841E+03	6.131E+03	3.679E+03
	STD	5.314E+02	**5.715E+01**	9.799E+01	3.574E+02	2.216E+02	6.384E+02	1.196E+02
	Rank	6	1	2	5	4	7	3
F28	Ave	7.732E+03	3.546E+03	3.682E+03	3.367E+03	**3.336E+03**	8.174E+03	3.380E+03
	STD	1.361E+03	1.413E+02	6.115E+02	3.561E+01	**3.932E+01**	7.004E+02	5.216E+01
	Rank	6	4	5	2	1	7	3
F29	Ave	8.028E+03	5.023E+03	4.796E+03	7.211E+03	6.738E+03	1.215E+04	**4.757E+03**
	STD	1.658E+03	3.767E+02	3.270E+02	9.525E+02	9.229E+02	4.462E+03	**3.522E+02**
	Rank	6	3	2	5	4	7	1
F30	Ave	2.241E+08	9.650E+07	5.180E+06	9.010E+07	6.323E+07	6.675E+08	**4.642E+06**
	STD	1.627E+08	3.353E+07	2.072E+06	2.904E+07	1.910E+07	3.036E+08	**1.509E+06**
	Rank	6	5	2	4	3	7	1
+/=/−		0/0/29	3/0/26	5/0/24	4/0/25	6/0/23	0/0/29	
Total Rank		174	100	88	117	100	186	47
CPUtime(s)		7.51E+02	1.63E+03	1.91E+03	5.81E+02	1.89E+03	1.02E+03	1.86E+03

Table 10: P-value obtained from Wilcoxon rank-sum test on CEC 2017 benchmark suite.

Function	OBPSO	SCA-PSO	QISCA	mGWO	NOSSA	CSA
	P	P	P	P	P	P
F1	6.80E-08	6.80E-08	6.01E-07	6.80E-08	6.80E-08	6.80E-08
F3	2.47E-04	6.80E-08	4.17E-05	6.80E-08	6.80E-08	2.92E-05
F4	6.80E-08	6.80E-08	6.80E-08	1.05E-06	9.75E-06	6.80E-08
F5	6.80E-08	9.13E-07	6.67E-06	6.80E-08	6.80E-08	6.80E-08
F6	6.80E-08	3.97E-03	**7.76E-01**	6.80E-08	6.80E-08	6.80E-08
F7	6.80E-08	4.68E-02	5.09E-04	9.17E-08	2.22E-07	9.17E-08
F8	6.80E-08	9.13E-07	8.29E-05	6.80E-08	7.90E-08	6.80E-08
F9	6.80E-08	6.01E-07	7.90E-08	9.17E-08	7.90E-08	6.80E-08
F10	2.36E-06	5.25E-05	3.15E-02	1.29E-04	2.14E-03	1.20E-06
F11	6.80E-08	6.80E-08	6.80E-08	1.81E-05	2.22E-04	6.80E-08
F12	6.80E-08	6.80E-08	6.92E-07	6.80E-08	1.20E-06	6.80E-08
F13	6.80E-08	6.80E-08	6.80E-08	**4.25E-01**	1.78E-03	6.80E-08
F14	8.35E-03	1.14E-02	9.28E-05	**2.29E-01**	2.07E-02	6.80E-08
F15	3.29E-05	6.80E-08	1.06E-07	**5.31E-02**	3.05E-04	6.80E-08
F16	1.23E-07	**6.75E-01**	2.56E-03	1.60E-05	8.60E-06	6.80E-08
F17	7.58E-06	**5.98E-01**	1.93E-02	1.10E-05	3.07E-06	3.07E-06
F18	**5.61E-01**	3.38E-04	1.01E-03	**7.64E-02**	4.68E-02	6.80E-08
F19	6.80E-08	7.90E-08	2.75E-04	9.17E-08	1.66E-07	6.80E-08
F20	1.81E-05	4.16E-04	**6.55E-01**	2.96E-07	3.29E-05	2.96E-07
F21	6.80E-08	2.06E-06	5.17E-06	6.80E-08	6.80E-08	6.80E-08
F22	6.80E-08	**5.31E-02**	**5.43E-01**	6.22E-04	**1.81E-01**	6.80E-08
F23	6.80E-08	1.78E-03	1.25E-05	6.80E-08	6.80E-08	6.80E-08
F24	6.80E-08	5.56E-03	7.95E-07	6.80E-08	6.80E-08	6.80E-08
F25	6.80E-08	6.80E-08	9.17E-08	**1.02E-01**	2.36E-06	6.80E-08
F26	6.80E-08	3.38E-04	2.56E-03	1.43E-07	7.90E-08	6.80E-08
F27	9.17E-08	1.43E-07	1.81E-05	1.38E-06	4.70E-03	6.80E-08
F28	6.80E-08	8.60E-06	8.29E-05	**9.25E-01**	4.60E-04	6.80E-08
F29	6.80E-08	3.15E-02	**6.17E-01**	6.80E-08	5.23E-07	6.80E-08
F30	6.80E-08	6.80E-08	**5.25E-01**	6.80E-08	6.80E-08	6.80E-08

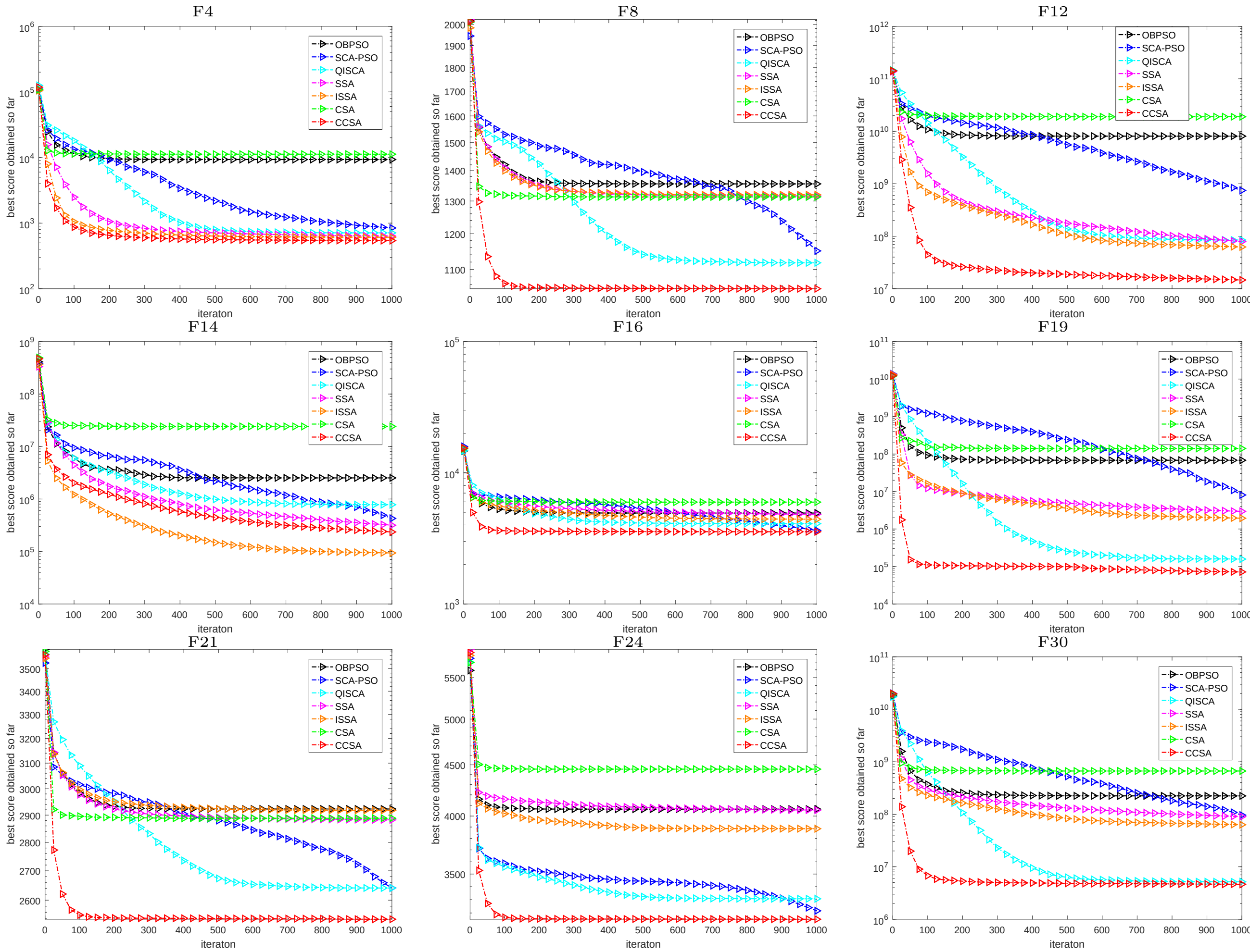

Fig. 5. Comparison of convergence curves of 7 algorithms for some of the CEC 2017 benchmark suite.

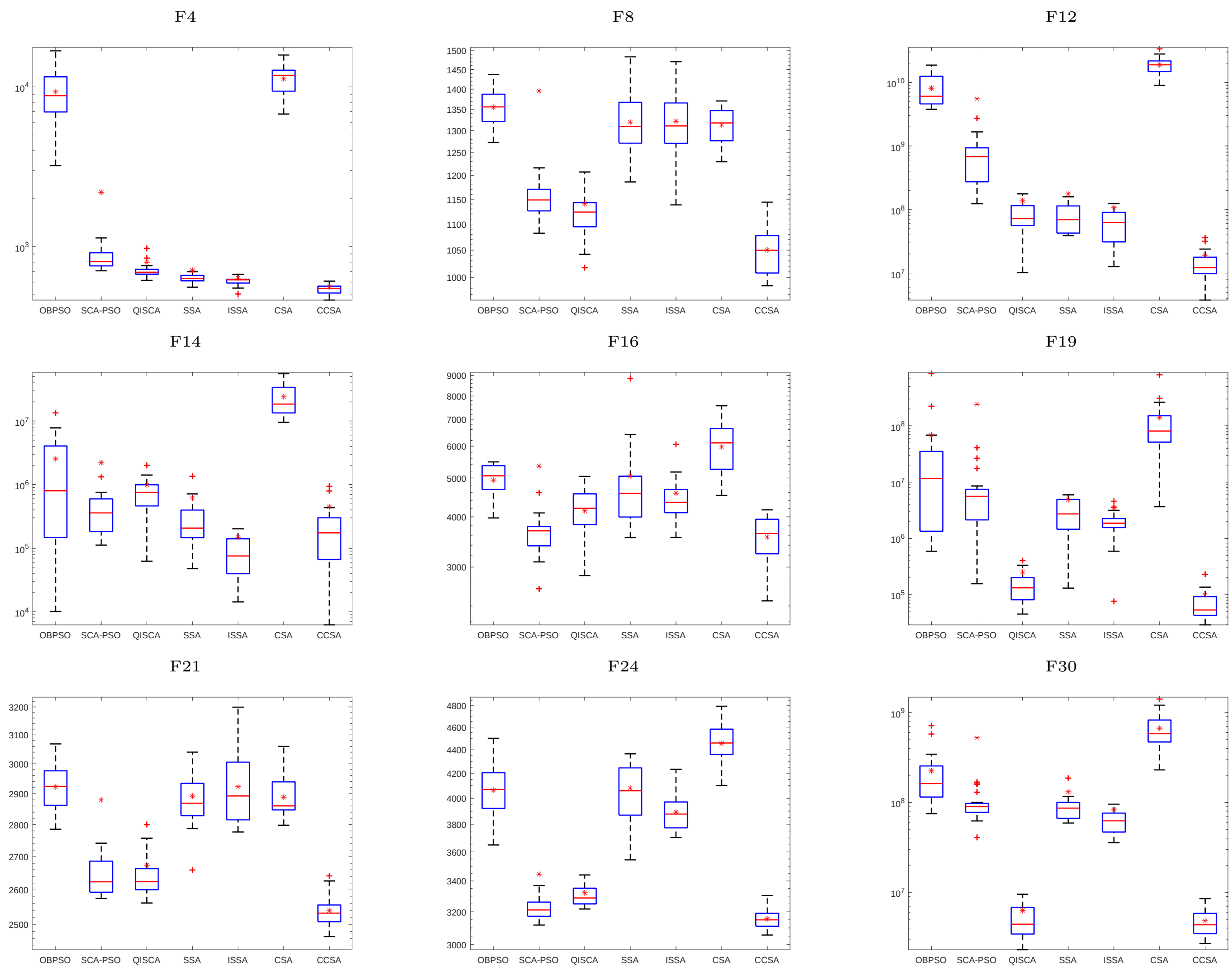

Fig. 6. Box plot of 7 algorithms for some of the CEC 2017 benchmark suite.

6. The optimal multi-degree reduction of Ball Bézier surfaces

6.1. Ball Bézier surfaces and their degree reduction

Definition 1. A Ball is defined as the following point set

$$\langle \mathbf{p}_o, r_o \rangle = \langle (x_o, y_o, z_o), r_o \rangle = \{x, y, z \in R^3 | (x - x_o)^2 + (y - y_o)^2 + (z - z_o)^2 \leq r_o^2\}, \tag{19}$$

where $\mathbf{p}_o = (x_o, y_o, z_o)$ is the center point and $r_o > 0$ is the radius of the ball.

Definition 2. Given $(m+1) \times (n+1)$ control balls $\{\langle \mathbf{p}_{i,j}, r_{i,j} \rangle\}_{i,j=0}^{m,n}$, a Ball Bézier surface of degree $(m \times n)$ is defined as [37]

$$\langle \mathbf{P} \rangle (u, v) = \sum_{i,j=0}^{m,n} B_i^m(u) B_j^n(v) \langle \mathbf{p}_{i,j}, r_{i,j} \rangle, \quad (u, v) \in [0, 1]^2, \tag{20}$$

where $B_i^m(u)$ is the ith Bernstein basis function of degree m and $B_j^n(v)$ is the jth Bernstein basis function of degree n.

Since a Ball surface $\langle \mathbf{P} \rangle(u,v)$ can be reformulated into

$$\langle \mathbf{P} \rangle(u,v) = \Big\langle \sum_{i,j=0}^{m,n} B_i^m(u)B_j^n(v)\mathbf{p}_{i,j}, \sum_{i,j=0}^{m,n} B_i^m(u)B_j^n(v)r_{i,j} \Big\rangle, \quad (u,v) \in [0,1]^2, \tag{21}$$

a Ball surface can be viewed as consisting of two parts:
(1) the center surface

$$\mathbf{S}(u,v) = \sum_{i,j=0}^{m,n} B_i^m(u)B_j^n(v)\mathbf{p}_{i,j}, \tag{22}$$

(2) the radius function

$$R(u,v) = \sum_{i,j=0}^{m,n} B_i^m(u)B_j^n(v)r_{i,j}. \tag{23}$$

Definition 3. The degree reduction of a Ball Bézier surface $\langle \mathbf{P} \rangle(u,v)$ of degree $(m \times n)$ means to search for a Ball Bézier surface of degree $(m_1 \times n_1)$ $(m_1 \leq m; n_1 \leq n)$

$$\begin{aligned} \langle \mathbf{Q} \rangle(u,v) =& \langle \mathbf{C}(u,v), G(u,v) \rangle \\ =& \Big\langle \sum_{i,j=0}^{m_1,n_1} B_i^{m_1}(u)B_j^{n_1}(v)\mathbf{q}_{i,j}, \sum_{i,j=0}^{m_1,n_1} B_i^{m_1}(u)B_j^{n_1}(v)g_{i,j} \Big\rangle, \quad (u,v) \in [0,1]^2, \end{aligned} \tag{24}$$

such that

$$\{\langle \mathbf{P} \rangle(u,v) | 0 \leq u,v \leq 1\} \subset \{\langle \mathbf{Q} \rangle(u,v) | 0 \leq u,v \leq 1\}, \tag{25}$$

and the radius of $\langle \mathbf{Q} \rangle(u,v)$ is as small as possible [42].

Since the inclusion condition in Eq.(25) is not easy to deal with, it is usually substituted by

$$G(u,v) \geq R(u,v) + \|\mathbf{C}(u,v), \mathbf{S}(u,v)\|, \quad (u,v) \in [0,1]^2, \tag{26}$$

where $\mathbf{S}(u,v)$ and $R(u,v)$ are the center surface and radius function of the original Ball Bézier surface $\langle \mathbf{P} \rangle(u,v)$ respectively, and $\mathbf{C}(u,v)$ and $G(u,v)$ are the center surface and radius function of the lower-degree Ball Bézier surface $\langle \mathbf{Q} \rangle(u,v)$ respectively, and for a fixed set of parameters (u,v),

$$\|\mathbf{C}(u,v), \mathbf{S}(u,v)\| = \Big[\big(C_x(u,v) - S_x(u,v)\big)^2 + \big(C_y(u,v) - S_y(u,v)\big)^2 + \big(C_z(u,v) - S_z(u,v)\big)^2\Big]^{\frac{1}{2}}.$$

In the previous literature, the constrain in Eq.(26) is usually further simplified by

$$G(u,v) - R(u,v) \geq d \geq \|\mathbf{C}(u,v), \mathbf{S}(u,v)\|, \quad (u,v) \in [0,1]^2. \tag{27}$$

Simplifying the constrain condition Eq.(26) to Eq.(27) will result in the radius of the obtained Ball Bézier surface being greater than that of the theoretical optimum. More importantly, the degree reduction problems under different interpolation constrains, such as degree reduction without interpolation constrains, degree reduction with endpoints interpolation or with boundaries interpolation, need to be analyzed and solved using different methods respectively and the consequent derivation and calculation are also quite complicated, , which makes the degree reduction procedure complex and thus hinders the data transmission of Ball Bézier surface among different CAD/CAM systems.

6.2. The algorithm for the optimal multi-degree reduction of Ball Bézier surfaces

In some cases, there are no interpolation constraints on the degree-reduced Ball surfaces, while in some cases the degree-reduced Ball surfaces are required to interpolate some endpoints or boundaries of the original one, which means that some of the control balls of the degree-reduced Ball surfaces need to be fixed and the others are movable (or unknown). In order to determine the optimal positions and radiuses of the unknown control balls, we separate the fixed and unknown control balls first. Let $\mathcal{I}_1 = \{0, 1, \cdots, m\} \times \{0, 1, \cdots, n\}$, $\mathcal{I}_2 = \{0, 1, \cdots, m_1\} \times \{0, 1, \cdots, n_1\}$, and $\mathcal{M} \subset \mathcal{I}_2$ be a subset which contains pairs formed by row and column indices of the unknown control balls of the Ball Bézier surface $\langle \mathbf{Q} \rangle(u, v)$ or is empty.

Similar to the works in [42], the center surface and radius function of a Ball Bézier surface are regarded as two independent parts. Therefore the procedure of the multi-degree reduction of a Ball Bézier surface consists of the following two steps:

Step 1. The optimal multi-degree reduction of center surface.

Step 2. The optimal multi-degree reduction of radius function.

In this study, for a Ball Bézier surface, the optimal multi-degree reduction of its center surface and radius function are formulated as two optimization problems. The change of different interpolation constrains are treated as the change of decision variables while the objective function and constraint conditions keeping unchanged, and thus we can use an optimization method to deal with all these cases.

6.2.1. The optimal multi-degree reduction of center surface

As for the optimal multi-degree reduction of center surface in Step 1, we formulate the problem as an optimization one in which the objective function is defined based on the distance between the original $(m \times n)$th-degree center Bézier surface and the low-$(m_1 \times n_1)$th-degree center Bézier surface as follows:

$$F_1[\mathbf{C}; \mathbf{S}] = \int_0^1 \int_0^1 \|\mathbf{C}(u, v) - \mathbf{S}(u, v))\|^2 \mathrm{d}u\mathrm{d}v. \tag{28}$$

By separating the fixed and unknown control balls of the low-$(m_1 \times n_1)$th-degree Ball Bézier surface $\langle \mathbf{Q} \rangle(u, v)$, its center surface $\mathbf{C}(u, v)$ can be represented as [46]

$$\mathbf{C}(u, v) = \mathbf{m}_{\mathcal{M}} + \mathbf{r}_{\mathcal{M}} \tag{29}$$

where

$$\mathbf{m}_{\mathcal{M}}(u, v) = \sum_{(i,j)\in\mathcal{M}} B_i^{m_1}(u) B_j^{n_1}(v) \mathbf{q}_{i,j}, \quad \mathbf{r}_{\mathcal{M}}(u, v) = \sum_{(k,l)\in\mathcal{I}_2\setminus\mathcal{M}} B_k^{m_1}(u) B_l^{n_1}(v) \mathbf{q}_{k,l}.$$

Then the objective function $F_1[\mathbf{C}; \mathbf{S}]$ in Eq.(28) can be rewritten as

$$\begin{aligned} F_1[\mathbf{C}; \mathbf{S}] &= \int_0^1 \int_0^1 (\mathbf{m}_{\mathcal{M}} + \mathbf{r}_{\mathcal{M}} - \mathbf{S}, \mathbf{m}_{\mathcal{M}} + \mathbf{r}_{\mathcal{M}} - \mathbf{S}) \mathrm{d}u\mathrm{d}v \\ &= \int_0^1 \int_0^1 (\mathbf{m}_{\mathcal{M}}, \mathbf{m}_{\mathcal{M}}) + 2(\mathbf{m}_{\mathcal{M}}, \mathbf{r}_{\mathcal{M}} - \mathbf{S}) + (\mathbf{r}_{\mathcal{M}} - \mathbf{S}, \mathbf{r}_{\mathcal{M}} - \mathbf{S}) \mathrm{d}u\mathrm{d}v \end{aligned} \tag{30}$$

Since the Bernstein bases have the following properties

$$B_i^n(u)B_j^m(u) = \frac{\binom{n}{i}\binom{m}{j}}{\binom{n+m}{i+j}} B_{i+j}^{n+m}(u), \qquad \int_0^1 B_i^n(u)du = \frac{1}{n+1}, \tag{31}$$

we can obtain

$$\begin{aligned} &\int_0^1 \int_0^1 (\mathbf{m}_{\mathcal{M}}, \mathbf{m}_{\mathcal{M}}) \mathrm{d}u\mathrm{d}v \\ &= \int_0^1 \int_0^1 \Big(\sum_{(i,j)\in\mathcal{M}} B_i^{m_1}(u) B_j^{n_1}(v) \mathbf{q}_{i,j}, \sum_{(i,j)\in\mathcal{M}} B_i^{m_1}(u) B_j^{n_1}(v) \mathbf{q}_{i,j} \Big) \mathrm{d}u\mathrm{d}v \\ &= \sum_{(i,j)\in\mathcal{M}} \sum_{(i_1,j_1)\in\mathcal{M}} A_{iji_1j_1} (\mathbf{q}_{i,j}, \mathbf{q}_{i_1,j_1}) \end{aligned} \tag{32}$$

where

$$\begin{aligned} A_{iji_1j_1} &= \int_0^1 \int_0^1 B_i^{m_1}(u) B_j^{n_1}(v) B_{i_1}^{m_1}(u) B_{j_1}^{n_1}(v) \mathrm{d}u\mathrm{d}v \\ &= \frac{1}{(2m_1+1)(2n_1+1)} \frac{\binom{m_1}{i}\binom{m_1}{i_1}}{\binom{2m_1}{i+i_1}} \frac{\binom{n_1}{j}\binom{n_1}{j_1}}{\binom{2n_1}{j+j_1}} \end{aligned}$$

Analogously,

$$\begin{aligned} &\int_0^1 \int_0^1 (\mathbf{m}_{\mathcal{M}}, \mathbf{r}_{\mathcal{M}} - \mathbf{S}) \mathrm{d}u\mathrm{d}v \\ &= \sum_{(i,j)\in\mathcal{M}} \sum_{(k,l)\in\mathcal{I}_2\setminus\mathcal{M}} B_{ijkl}(\mathbf{q}_{i,j}, \mathbf{q}_{k,l}) - \sum_{(i,j)\in\mathcal{M}} \sum_{(s,t)\in\mathcal{I}_1} C_{ijst}(\mathbf{q}_{i,j}, \mathbf{p}_{s,t}) \end{aligned} \tag{33}$$

where

$$B_{ijkl} = \frac{1}{(2m_1+1)(2n_1+1)} \frac{\binom{m_1}{i}\binom{m_1}{k}}{\binom{2m_1}{i+k}} \frac{\binom{n_1}{j}\binom{n_1}{l}}{\binom{2n_1}{j+l}},$$

$$C_{ijst} = \frac{1}{(m+m_1+1)(n+n_1+1)} \frac{\binom{m_1}{i}\binom{m}{s}}{\binom{m+m_1}{i+s}} \frac{\binom{n_1}{j}\binom{n}{t}}{\binom{n+n_1}{j+t}}.$$

Let

$$\phi = \int_0^1 \int_0^1 (\mathbf{r}_{\mathcal{M}} - \mathbf{S}, \mathbf{r}_{\mathcal{M}} - \mathbf{S}) \mathrm{d}u\mathrm{d}v \tag{34}$$

Thus, the objective function in Eq.(28) can be expressed as

$$\begin{aligned} F_1[\mathbf{C}; \mathbf{S}] = &\sum_{(i,j)\in\mathcal{M}} \sum_{(i_1,j_1)\in\mathcal{M}} A_{iji_1j_1}(\mathbf{q}_{i,j}, \mathbf{q}_{i_1,j_1}) + 2 \sum_{(i,j)\in\mathcal{M}} \sum_{(k,l)\in\mathcal{I}_2\setminus\mathcal{M}} B_{ijkl}(\mathbf{q}_{i,j}, \mathbf{q}_{k,l}) \\ &- 2 \sum_{(i,j)\in\mathcal{M}} \sum_{(s,t)\in\mathcal{I}_1} C_{ijst}(\mathbf{q}_{i,j}, \mathbf{p}_{s,t}) + \phi, \end{aligned} \tag{35}$$

To solve the quadratic minimization problem $F_1[\mathbf{C};\mathbf{S}] \to \min$, we take the partial derivatives of Eq.(35) with respect to the unfixed control points and establish the following linear system

$$\frac{\partial F_1[\mathbf{C};\mathbf{S}]}{\partial \mathbf{q}_{i,j}} = 0, \quad (i,j) \in \mathcal{M}, \tag{36}$$

the ith row of which can be represented as

$$\sum_{(i_1,j_1)\in\mathcal{M}} A_{iji_1j_1}\mathbf{q}_{i_1,j_1} = \sum_{(s,t)\in\mathcal{I}_1} C_{ijst}\mathbf{p}_{s,t} - \sum_{(k,l)\in\mathcal{I}_2\setminus\mathcal{M}} B_{ijkl}\mathbf{q}_{k,l}, \quad (i,j) \in \mathcal{M}. \tag{37}$$

Thus the positions of the unfixed control points $\{\mathbf{q}_{i,j}\}_{(i,j)\in\mathcal{M}}$ of the degree reduced center surface $\mathbf{C}(u,v)$ could be determined by solving the linear system Eq.(37).

In previous works, the degree reduction problem of center curves were usually solved based on Chebyshev polynomials, which can only achieve 1-degree reduction one time[42]. For the problem of multi-degree reduction, repeated operation is required, greatly increasing the error. While in our method, the positions of the unknown control points of the approximate center curve can be determined by solving a linear system directly, simplifying the degree reduction procedure of center surface remarkably.

When the center surface of $\langle\mathbf{Q}\rangle(u,v)$ is determined, it is essential to computer the distance between the original center surface and the approximate one. Take sufficient uniform sampling points $\{(u_i,v_j)\}_{i,j=1}^{K_1,K_2} \in [0,1]^2$, compute and store the discrete distance for $\{(u_i,v_j)\}$ between the center curves before and after degree reduction

$$d_{i,j} = \|\mathbf{C}(u_i,v_j), \mathbf{S}(u_i,v_j)\|, \quad i = 1,2,\ldots,K_1; j = 1,2,\ldots,K_2. \tag{38}$$

for the next step, here we let $K_1 = K_2 = 20$.

6.2.2. The optimal multi-degree reduction of radius function

As for the optimal multi-degree reduction of the radius function of Ball Bézier surface $\langle\mathbf{P}\rangle(u,v)$ in Step 2, we formulate the problem as the following optimization one.

Objective function

$$\text{Minimiz} \quad F_2[G;R] = \sum_{i,j=1}^{K_1,K_2}\Big[G^2(u_i,v_j) - \lambda\min\Big(G(u_i,v_j) - R(u_i,v_j) - d_{i,j}, 0\Big)\Big], \tag{39}$$

where $R(u,v)$ and $G(u,v)$ denote the radius functions of the Ball Bézier surface before and after degree reduction respectively, $d_{i,j}$ is the discrete distance between the original and the approximate center surface for (u_i,v_j).

Variable ranges

$$r_{\min} \le g_{i,j} \le d_{\max} + kr_{\max}, \quad (i,j) \in \mathcal{M}, \tag{40}$$

where the decision variables $\{g_{i,j}\}_{(i,j)\in\mathcal{M}}$ of the optimization problem are the unknown control radiuses of the low-$(m_1 \times n_1)$th-degree Ball Bézier surface $\langle\mathbf{Q}\rangle(u,v)$, and

$$r_{\min} = \min_{(i,j)\in\mathcal{I}_1}\{r_{i,j}\},\ r_{\max} = \max_{(i,j)\in\mathcal{I}_1}\{r_{i,j}\},\ d_{\max} = \max_{i,j=1,1}^{K1,K2}\{d_{i,j}\}.$$

In Eq.(39), $\sum_{i,j=1}^{K_1,K_2} G^2(u_i, v_j)$ is employed to ensure that the approximate radius function $G(u, v)$ is as small as possible, while $\sum_{i,j=1}^{K_1,K_2} \Big[\min \Big(G(u_i, v_j) - R(u_i, v_j) - d_{i,j}, 0 \Big) \Big]$ is applied to guarantee $G(u, v) \geq R(u_i, v_j) + d_{i,j}$ for any (u_i, v_j) for fulfilling the inclusion condition in Eq.(25). λ is a weighting coefficient, and after trial and error we set $\lambda = 100$. In Eq.(40), k is a parameter determined by the reduced degree.

The optimization problem can be solve using an metaheuristic method, with which we search within the variable range for the optimal set of control radiuses of the lower-degree Ball Bézier surface $\langle \mathbf{Q} \rangle(u, v)$ to determine the approximate radius function. Combining the approximate center surface $\mathbf{C}(u, v)$, the optimal low-$(m_1 \times n_1)$th-degree Ball Bézier surface $\langle \mathbf{Q} \rangle(u, v)$ can be obtained. Table 11 gives the implementation steps of the proposed degree reduction algorithm for Ball Bézier surface.

Table 11: Implementation steps of the optimal multi-degree reduction of Ball Bézier surface.

Algorithm 3: Algorithm of the Optimal Multi-degree Reduction of Ball Bézier Surface
Step 1: Input the original Ball Bézier surface $\langle \mathbf{P} \rangle(u, v)$ and the degree it need to be reduced.
Step 2: Determine the approximate center surface $\mathbf{C}(u, v)$ by solving the linear system Eq.(37).
Step 3: Take sufficient uniform sampling points in $\{(u_i, v_j)\}_{i,j=1,1}^{K_1,K_2}$ in $[0, 1]^2$, calculate and restore $\{R(u_i, v_j) + \lVert \mathbf{C}(u_i, v_j), \mathbf{S}(u_i, v_j) \rVert\}_{i,j=1,1}^{K_1,K_2}$.
Step 4: Solve problem Eq.(39) using the proposed CCSA algorithm with the data obtained in Step 3, determine
Step 5: Output the optimal degree reduced Ball Bézier surface according to $\mathbf{C}(u, v)$ and $G(u, v)$.

6.3. Experiments to verify the optimal multi-degree reduction algorithm

To verify the effectiveness of the proposed algorithm, the degree reduction results of different Ball Bézier surfaces without interpolation constrains, with four endpoints interpolation and with two boundaries interpolation are provided in Fig. 7-Fig. 9 respectively. In these examples, the parameter k in Eq.(40) is set as 3.

Example 1. Given a Ball Bézier surface of degree (5×5), Fig. 7 shows the optimal approximate Ball Bézier surface of degree (4×4) under no interpolation constrains.

Example 2. Given a Ball Bézier surface of degree (7×7), Fig. 8 shows the optimal approximate Ball Bézier surface of degree (5×5) with the interpolation of the four endpoints of the original Ball surface.

Example 3. Given a Ball Bézier surface of degree (7×7), Fig. 9 shows the optimal approximate Ball Bézier surface of degree (7×5) with the interpolation of two boundaries of the original Ball surface.

6.4. Results discussion

(1). As shown in Fig. 7(b)-Fig. 9(b), the center surfaces after degree reduction are very close to the input ones. The boundary curves of the center surface before and after degree reduction basically coincide, and there are quite a bit of overlapping regions between the degree reduced center surface and the original one. All these indicate that the proposed multi-degree reduction method works well under different interpolation conditions.

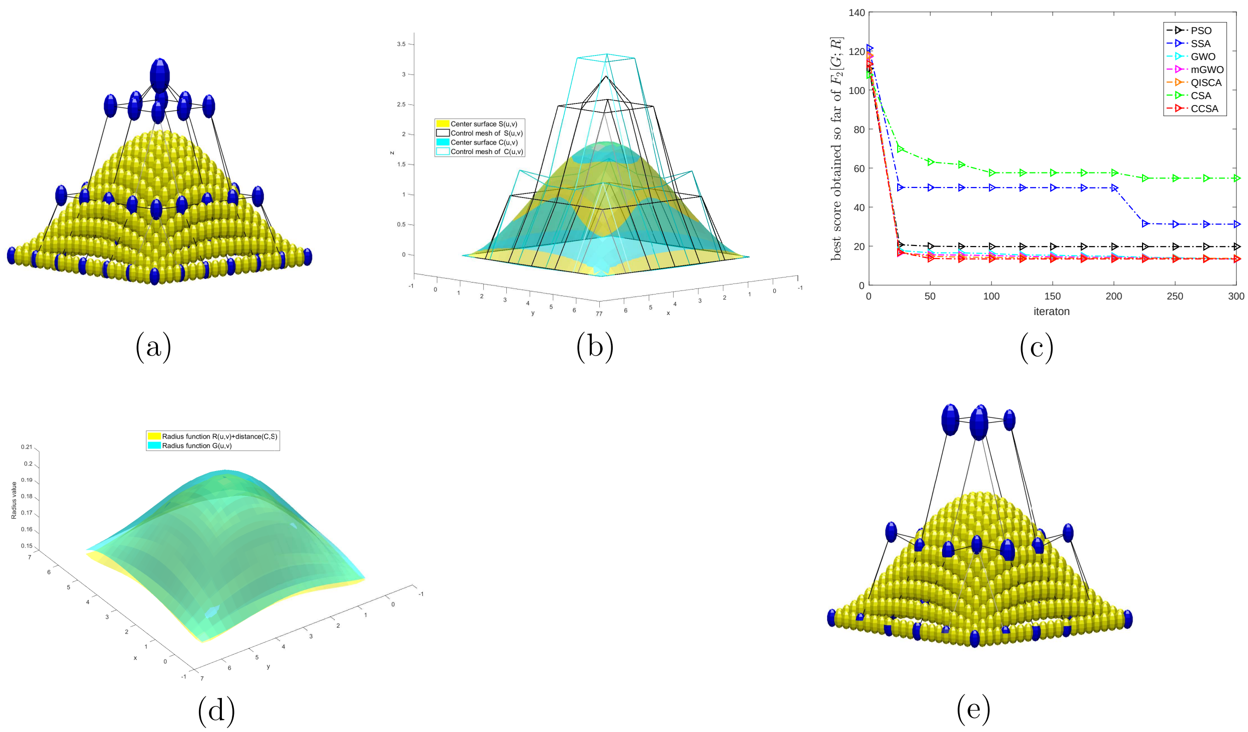

Fig. 7. The optimal multi-degree reduction from a Ball Bézier surface of degree (5×5) to one of degree (4×4) without interpolation constrains. (a) The original Ball Bézier surface. (b) The original center surface and the optimal approximate center surface obtained by the proposed method. (c) Convergence curves of the objective function $F_2[G; R]$ in Eq.(39) with the proposed CCSA and 6 existing metaheuristic algorithms. (d) Comparison between the function $R(u, v) + \|\mathbf{C}(u, v), \mathbf{S}(u, v)\|$ and the optimal radius function $G(u, v)$ obtained using the CCSA. (e) The optimal low-(4×4)th-degree Ball Bézier surface obtained using the CCSA.

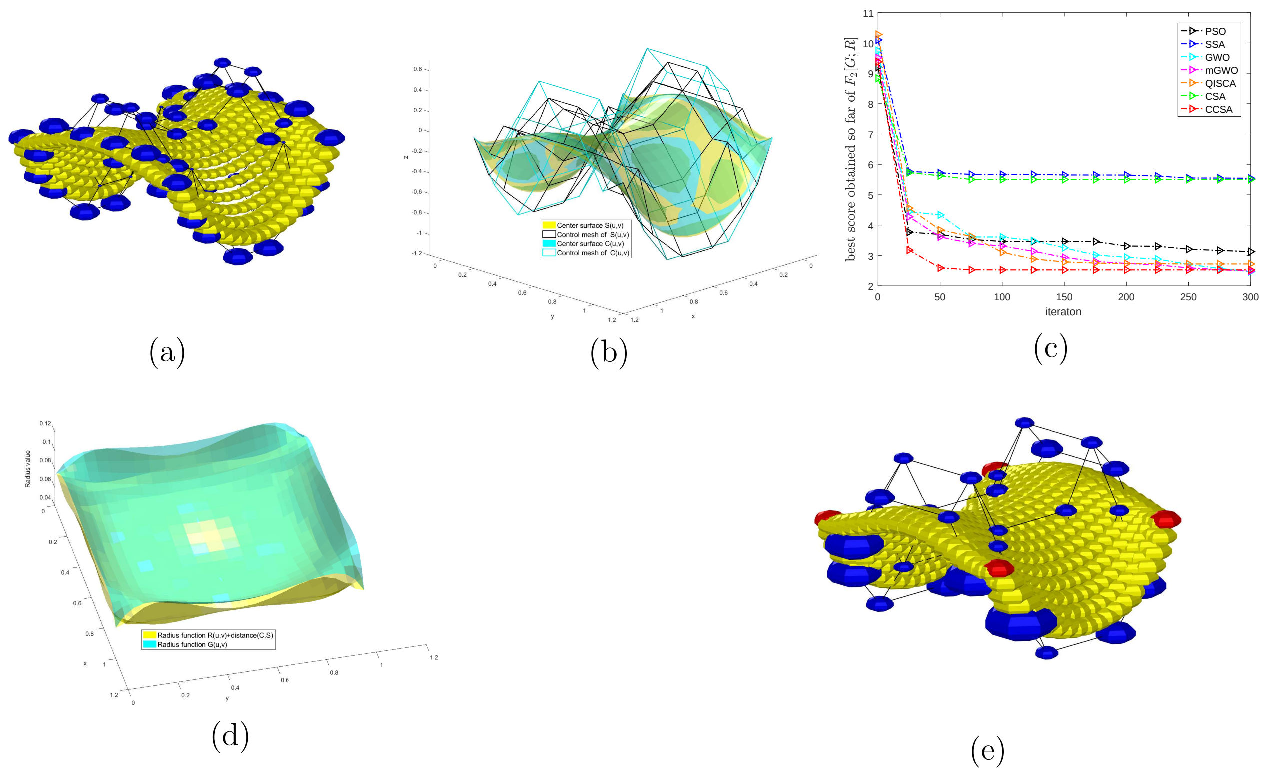

Fig. 8. The optimal multi-degree reduction from a Ball Bézier surface of degree (7×7) to one of degree (5×5) with four endpoints interpolation. (a) The original Ball Bézier surface. (b) The original center surface and the optimal approximate center surface obtained by the proposed method. (c) Convergence curves of the objective function $F_2[G; R]$ in Eq.(39) with the proposed CCSA and 6 existing metaheuristic algorithms. (d) Comparison between the function $R(u, v) + \|\mathbf{C}(u, v), \mathbf{S}(u, v)\|$ and the optimal radius function $G(u, v)$ obtained using the CCSA. (e) The optimal low-(5×5)th-degree Ball Bézier surface obtained by the CCSA, where the four endpoints are those of the original Ball surface and are drawn in red.

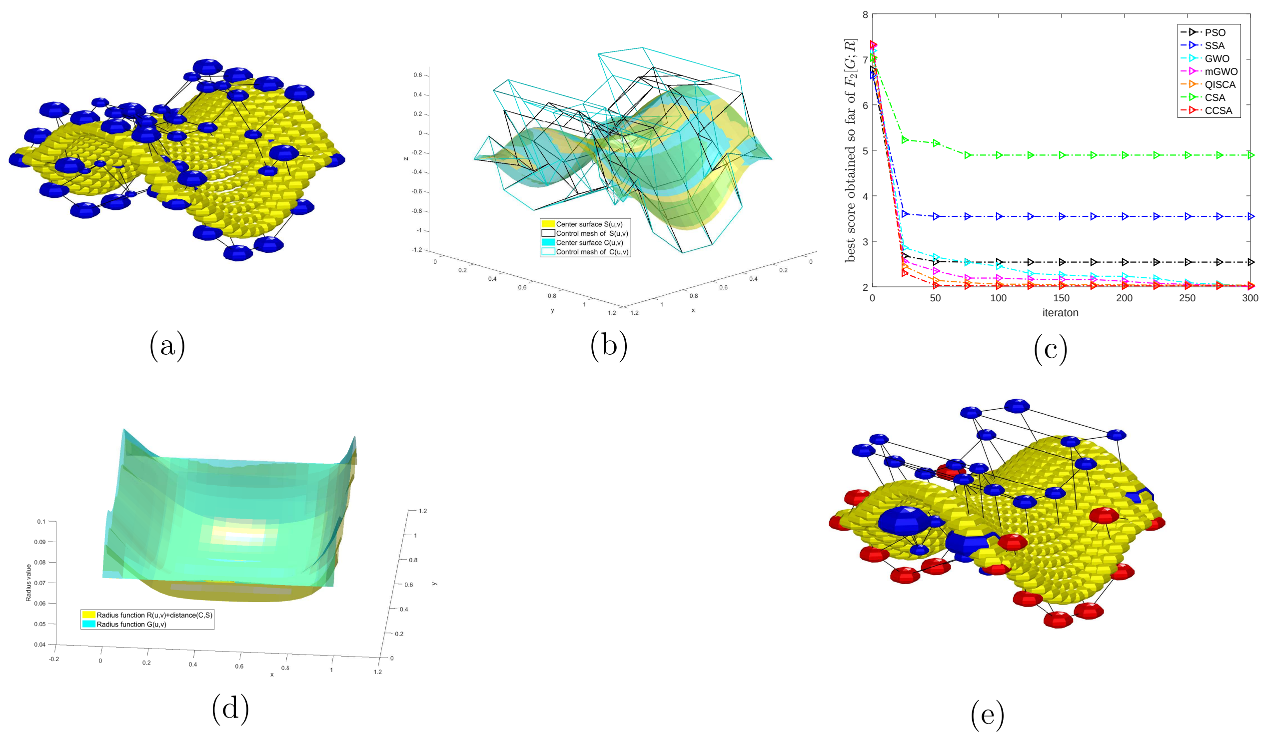

Fig. 9. The optimal multi-degree reduction from a Ball Bézier surface of degree (7×7) to one of degree (7×5) with two boundaries interpolation. (a) The original Ball Bézier surface. (b) The original center surface and the optimal approximate center surface obtained by the proposed method. (c) Convergence curves of the objective function $F_2[G; R]$ in Eq.(39) with the proposed CCSA and 6 existing metaheuristic algorithms. (d) Comparison between the function $R(u, v) + \|\mathbf{C}(u, v), \mathbf{S}(u, v)\|$ and the optimal radius function $G(u, v)$ obtained using the CCSA. (e) The optimal low-(7×5)th-degree Ball Bézier surface obtained by the CCSA, where the control balls of the two unchanged boundaries are drawn in red.

(2). Form Fig. 7(d)-Fig. 9(d), we can see that the green surface i.e. the radius function after degree reduction, is always above the yellow surface i.e. the original radius function, indicating that for any $(u, v) \in [0, 1]^2$ the condition $G(u, v) \geq R(u, v) + \|\mathbf{C}(u, v), \mathbf{S}(u, v)\|$ is satisfied. Under this condition, $G(u, v)$ approximates $R(u, v) + \|\mathbf{C}(u, v), \mathbf{S}(u, v)\|$ as close as possible for different original Ball Bézier surfaces, which demonstrates that both the inclusion condition in Eq.(25) and the constrain that the radius of the lower-degree Ball Bézier surface should be as small as possible can be fully satisfied by using the proposed multi-degree reduction method for radius function.

(3). It is obvious from the convergence curves of the CCSA and other existing meta-heuristic algorithms in Fig. 7(c)-Fig. 9(c) that the CCSA algorithm has better performance both on convergence speed and accuracy in solving the degree reduction problems of radius functions.

(4). It can be seen in each example that the Ball Bézier surfaces before and after degree reduction are similar both in shape and thickness.

The experiment results show that the proposed degree reduction method for Ball Bézier surfaces can deal with various interpolation conditions flexibly and can achieve good degree reduction effect.

7. Conclusion

This study proposes an improved cooperation search algorithm (CCSA) by incorporating a mutation and a crossover operator. The mutation operator is adopted to enhance population diversity and the crossover is employed to avoid the ignorance of obtained potential message of the search space. These operators can help enhance the collaborative strength of the team and maintain an appropriate balance between exploration and exploitation. Numerical experiments on 23 classic benchmark functions and 29 CEC 2017 benchmark functions demonstrate the superior performance of the proposed CCSA algorithm in terms of search-efficiency, solution accuracy. Beside, we propose a new universal method for the multi-degree reduction of Ball Bézier surfaces under different interpolation constrains by applying the proposed CCSA algorithm. The simulation results under different interpolation constrains show that the proposed method is effective and flexible, which achieves the automatic and intelligent multi-degree reduction of Ball Bézier surfaces.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (Grant No. 51875454).

Author Contributions

The authors contributed to each part of this paper equally. The authors read and approved the final manuscript.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Human and animal rights This article does not contain any studies with human participants performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

[1] D. Molina, J. Poyatos, J.D. Ser, S. García, A. Hussain, F. Herrera. Comprehensive taxonomies of nature-and bio-inspired optimization: inspiration versus algorithmic behavior, critical analysis and recommendations. Cognitive computation. 12(5), 897-939.

[2] J.H. Holland. Adaptation in Natural and Artificial Systems. University of Michigan Press. Ann Arbor, MI (1975).

[3] S. Das, P.N. Suganthan. Differential evolution: a survey of the state-of-the-art. IEEE Trans. Evol. Comput. 15 (2011), 4-31.

[4] J. Robinson, Y. Rahmat-Samii. Particle swarm optimization in electromagnetic. IEEE Trans. Antennas Propag. 52 (2004), 397-407.

[5] S. Mirjalili, S.M. Mirjalili. A. Lewis, Grey wolf optimizer. Adv. Eng. Softw. 69 (2014), 46-61.

[6] X. Yang, S. Deb. Cuckoo search via Lévy flights. In: World Congress on Nature Biologically Inspired Computing. NaBIC (2009), 210-214.

[7] M. Jain, V. Singh, A. Rani. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. Swarm Evol. Comput. 44 (2019), 148-175.

[8] G. Hu, X. Zhu, G. Wei, C.-T Chang. An improved marine predators algorithm for shape optimization of developable Ball surfaces. Eng. Appl. Artif. Intel. 105 (2021), 104417.

[9] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi. GSA: a gravitational search algorithm. Inf. Sci. 179 (2009), 2232-2248.

[10] S. Mirjalili, S.M. Mirjalili, A. Hatamlou. Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput. Appl. 27 (2016), 495-513.

[11] A. Lam, V. Li. Chemical-reaction-inspired metaheuristic for optimization. IEEE Trans. Evol. Comput. 14(2010), 381-399.

[12] T. Huan, A. Kulkarni, J. Kanesan, C. Huang, A. Abraham. Ideology algorithm: a socio-inspired optimization methodology. Neural Comput. Appl. 28(2017), 845-876.

[13] H. Khattab, A. Sharieh, B.A. Mahafzah. Most valuable player algorithm for solving minimum vertex cover problem. Int. J. Adv. Comput. Sci. Appl. 10 (2019), 159-167.

[14] M. Ghaemi, MR. Feizi-Derakhshi. Forest optimization algorithm, Expert Syst. Appl. 41(2014), 6676-6687.

[15] V. Punnathanam, P. Kotecha. Yin-Yang-pair optimization: A novel lightweight optimization algorithm. Eng. Appl. Artif. Intel. 54(2016),62-79.

[16] M.A. Elaziz, D. Oliva, S. Xiong. An improved opposition-based sine cosine algorithm for global optimization. Expert Syst. Appl. 90 (2017), 484-500.

[17] D.H. Wolpert, W.G. Macready. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1 (1997), 67-82.

[18] S. Gupta, K. Deep. A memory-based grey wolf optimizer for global optimization tasks. Appl. Soft Comput. 93 (2020), 106367.

[19] M.A. Saxena, R. Kumar, S. Das. β-chaotic map enabled grey wolf optimizer. Appl. Soft Comput. 75 (2019), 84-105.

[20] R.A. Ibrahim, M.A. Elaziz, S. Lu. Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. Expert Syst. Appl. 108 (2018), 1-27.

[21] S. Mirjalili. A sine cosine algorithm for solving optimization problems. Knowl.-Based Syst. 96 (2016), 120-133.

[22] W. Guo, Y. Wang, F. Dai, P. Xu. Improved sine cosine algorithm combined with optimal neighborhood and quadratic interpolation strategy. Eng. Appl. Artif. Intel. 94 (2020), 103779.

[23] H. Chen, C. Yang, A.A. Heidari, X. Zhao. An efficient double adaptive random spare reinforced whale optimization algorithm. Expert Syst. Appl. 154 (2020), 113018.

[24] A.R. Mehrabian, C. Lucas. A novel numerical optimization algorithm inspired from weed colonization. Ecol. Inform. 1 (2006), 355-366.

[25] H. Hu, L. Zhang, Y. Bai, P. Wang, X. Tan. A hybrid algorithm based on squirrel search algorithm and invasive weed optimization for optimization. IEEE Access. 7 (2019), 105652-105668.

[26] H. Wang, Y. Liu, S. Zeng. Opposition-based particle swarm algorithm with Cauchy mutation. 2007 IEEE congress on evolutionary computation, 4425095, 4750-4756.

[27] H. Nenavath, R.J. Kumar, S. Das. A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking. Swarm Evol. Comput. 43 (2018), 1-30.

[28] Z. Feng, W. Niu, S. Liu. Cooperation search algorithm: A novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems. Appl. Soft Comput. 98 (2021), 106734.

[29] M.R. Tanweer, S. Suresh, N. Sundararajan. Self regulating particle swarm optimization algorithm. Inform. Sci. 294 (2015), 182-202.

[30] A. Pasko, V. Adzhiev, P. Comninos. Heterogeneous Objects Modelling and Applications: Collection of Papers on Foundations and Practice. Springer 4889 (2008).

[31] Y. Song, Z. Yang, Y. Liu, J. Deng. Function representation based slicer for 3D printing. Comput. Aided Geom. Design. 62 (2018), 276-293.

[32] X. Wang, Z. Wu, J. Shen, et al. Repairing the cerebral vascular through blending Ball B-Spline curves with G^2 continuity. Neurocomputing. 173 (2016), 768-777.

[33] Z. Wu, M. Zhou, X. Wang, et al. An interactive system of modeling 3D trees with ball

b-spline curves. In: 2007 10th IEEE International Conference on Computer-Aided Design and Computer Graphics. (2007), 259-265.

[34] T. Zhu, F. Tian, Y. Zhou, et al. Plant modeling based on 3D reconstruction and its application in digital museum. Int. J. Virt. Real. 7 (2008), 81-88.

[35] X. Xu, C. Leng, Z. Wu. Rapid 3D human modeling and animation based on sketch and motion database. In: 2011 Workshop on Digital Media and Digital Content Management (DMDCM). (2011), 121-124.

[36] C. Leng, Z. Wu, M. Zhou. Reconstruction of tubular object with ball b-spline curve. In: Proceedings of Computer Graphics International (2011).

[37] Z. Wu, H. Seah , M. Zhou. Skeleton based parametric solid models: Ball B-Spline curves. In: 2007 10th IEEE International Conference on Computer-Aided Design and Computer Graphics. (2007), 421-424.

[38] Q. Fu, Z. Wu, X. Wang, M. Zhou, J. Zheng, X. Wang, et al. An algorithm for finding intersection between ball B-spline curves. J. Comput. Appl. Math. 327 (2018), 260-273.

[39] X. Liu, X. Wang, Z. Wu, D. Zhang, X. Liu. Extending Ball B-spline by B-spline. Comput. Aided Geom. Design. 82 (2020), 101926.

[40] Q. Hu, G. Wang. Exact boundary of ball Bézier surface and its approximation by polynomial form. J. Zhejiang. Univ. (Eng. Sci.) 42 (2008), 1906-1909, (in Chinese).

[41] H. Liu, J. Deng. Fitting scattered data with Disk/Ball Bézier and B-Spline curves/surfaces. J. Univ. Sci..Technol. Chin. 38 (2008), 113-120, (in Chinese).

[42] H. Wu, J. Deng. Degree reduction of Bézier surfaces with ball control points. J. Univ. Sci. Technol. China. 36 (2006), 582-589,(in Chinese).

[43] Y. Chen, H. Wu, J. Deng. Degree reduction of ball-control-point Bézier surfaces over triangular domain. J. Univ. Sci. Technol. Chin. 37 (2007), 777-784, (in Chinese).

[44] F. Wang, H. Zhang, K. Li, et al. A hybrid particle swarm optimization algorithm using adaptive learning strategy. Inform. Sci. 436 (2018), 162-177.

[45] N.H. Awad, M.Z. Ali, J.J. Liang, B.Y. Qu, P.N. Suganthan. Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-Parameter numerical optimization. Technical report, Nanyang Technological University Singapore, 2016.

[46] I. Juhász, Á. Róth. Adjusting the energies of curves defined by control points. Comput.-Aided Des. 107 (2019), 77-88.