Applied Soft Computing Journal 92 (2020) 106314

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Enhancing tree-seed algorithm via feed-back mechanism for optimizing continuous problems ☆

Jianhua Jiang [a,b,*], Xianqiu Meng [a], Yunjun Chen [a], Chunyan Qiu [a], Yang Liu [a], Keqin Li [c]

[a] Department of Data Science, Jilin University of Finance and Economics, Changchun 130117, PR China
[b] Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun, 130012, PR China
[c] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Article history:
Received 5 January 2019
Received in revised form 14 March 2020
Accepted 13 April 2020
Available online 21 April 2020

MSC:
68W20
68T20

Keywords:
Optimization algorithm
Tree-seeds algorithm
TSA
Continuous problems

ABSTRACT

Tree-Seed Algorithm (TSA) is a novel population-based random search algorithm with its advantages in continuous optimization problems. However, there are some problems in its searching procedure. **Problem (1)**: its balance mechanism of exploration and exploitation is implemented with a constant *ST*, and this fixed value is unreasonable in the random search procedure; **Problem (2)**: the seed generation mechanism is achieved randomly without considering different searching phases based on function evaluations. To overcome these two problems, the feedback mechanism should be enhanced. Firstly, the st_TSA is proposed to solve the Problem (1); secondly, the ns_TSA is proposed to further solve the Problem (2); finally, in order to inherit these feedback mechanisms, a novel fb_TSA has been proposed and verified by standard 30 test benchmark functions from IEEE CEC 2014 with the basic TSA and its variants, such as STSA. In addition, GWO, ABC, SCA, DE, PSO and CLPSO are adopted for some comparative experiments with different dimensions. The computational results demonstrate that the enhanced feedback mechanism on *ST* and *ns* parameters can improve the optimization capability of the basic TSA significantly, especially in global optimum. The applicability of the proposed fb_TSA is proved by the 4 real engineering problems when compared with TSA, SCA, ABC and PSO.

1. Introduction

Optimization problems refer to determining the value of some optional variables under certain constraints to optimize the selected objective functions [1]. It is the inherent characteristic of achieving the best or the most advantageous (minimum or maximum) in a given situation [2]. Optimization problems contain discrete [3,4] or continuous [5,6] structured solution space and multi-objective optimization problems [7–9], etc. Hence, optimization applies in engineering [10–12], industrial design [13,14], design analysis and so on [15]. Meanwhile, nature-inspired computation also becomes increasingly popular in engineering [16]. Many of these technologies have been inspired by the evolution of biology that has produced extremely complex living systems [16].

Since the 1970s, many heuristic algorithms have begun to imitate natural phenomena [17,18]. In the past few years, many nature-/bio-inspired optimization techniques were proposed. Some of the recent and popular algorithms to solve continuous optimization problems are the following :

- Evolutionary techniques: Genetic Algorithms (GA) [19], Differential Evolution (DE) [20–22], Biogeography-Based Optimization algorithm (BBO) [23], and Evolution Strategy (ES) [24] etc.
- Swarm intelligence techniques: Ant Colony Optimization (ACO) [25], Particle Swarm Optimization (PSO) [26], Comprehensive Learning Particle Swarm Optimization (CLPSO) [27], Artificial Bee Colony (ABC) [28,29] algorithm and Firefly Algorithm (FA) [30] etc.
- Physics-based techniques: Gravitational Search Algorithm (GSA) [31], Colliding Bodies Optimization (CBO) [32], and Black Hole (BH) [33] etc.
- Human-related techniques: League Championship Algorithm (LCA) [34], Mine Blast Algorithm (MBA) [35], and Teaching–Learning-Based Optimization (TLBO) [36] etc.

To solve optimization problems, three main directions are proposed including: improving the current techniques, hybridizing

☆ The authors are grateful to the financial support by the Natural Science Foundation of Jilin Province, China (no. 20180101044JC), Jilin Province Philosophy and Social Science Planning Fund Office, China (nos. 2019B68, 2017BS28), the Foundation of the Education Department of Jilin Province, China (nos. JJKH20180465KJ, JJKH20190111KJ) and the Foundation of Jilin University of Finance and Economics, PR China (no. 2018Z05).
* Corresponding author at: Department of Data Science, Jilin University of Finance and Economics, Changchun 130117, PR China.
E-mail addresses: jianhuajiang@yahoo.com (J. Jiang), 1134430458@qq.com (X. Meng), lik@newpaltz.edu (K. Li).

https://doi.org/10.1016/j.asoc.2020.106314
1568-4946/

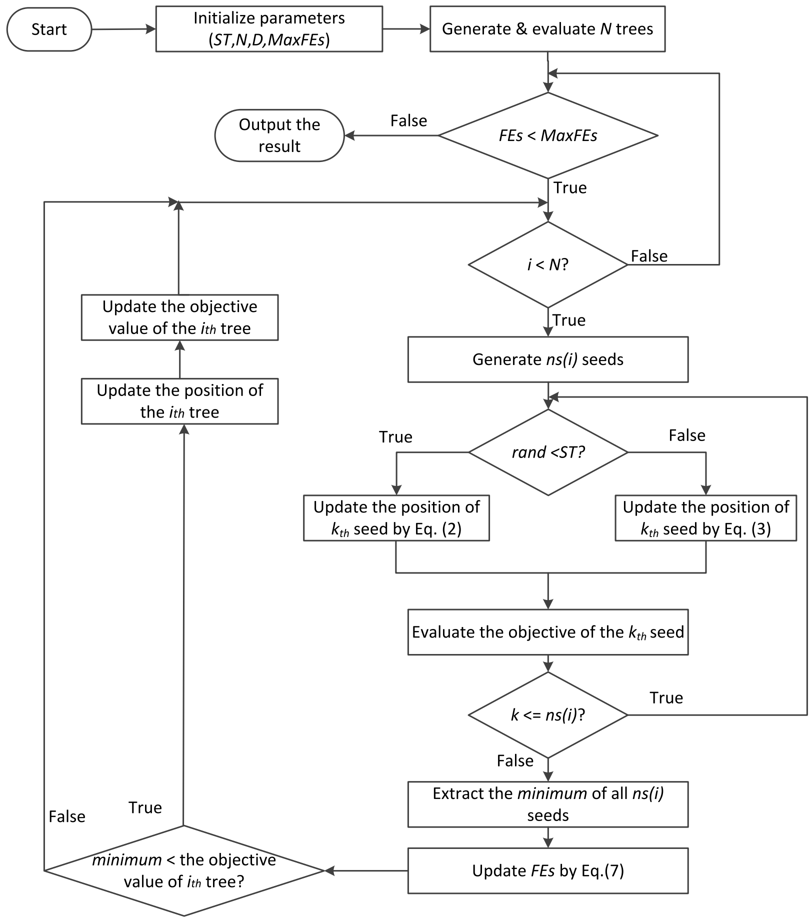


Fig. 1. Flow chart of the basic TSA.

Table 1
Definitions of the basic functions.

Definitions of the basic functions	
High Conditioned Elliptic Function	$f_1(X)=\sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}}X_i^2$
Bent Cigar Function	$f_2(X)=X_1^2+10^6\sum_{i=1}^{D}X_i^2$
Discus Function	$f_3(X)=10^6X_i^2+\sum_{i=1}^{D}X_i^2$
Rosenbrock's Function	$f_4(X)=\sum_{i=1}^{D-1}(100(X_i^2-X_{i+1})^2+(X_i-1)^2)$
Ackley's Function	$f_5(X)=-20exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}X_i^2})-exp(\frac{1}{D}\sum_{i=1}^{D}cos(2\pi X_i))$
Weierstrass Function	$f_6(X)=\sum_{i=1}^{D}(\sum_{k=0}^{kmax}[a^k cos(2\pi b^k(X_i+0.5))])-D\sum_{k=0}^{kmax}[a^k cos(2\pi b^k\bullet 0.5); a=0.5, b=3, kmax=20]$;
Griewank's Function	$f_7(X)=\sum_{i=1}^{D}\frac{X_i^2}{4000}-\prod_{i=1}^{D}cos(\frac{X_i}{\sqrt{i}})+1$
Rastrigin's Function	$f_8(X)=\sum_{i=1}^{D}(X_i^2-10cos(2\pi X_i)+10)$
Modified Schwefel's Function	$f_9=418.9829\ timesD-\sum_{i=1}^{D}g(z_i);\ z_i=x_i+4.209687462275036e{+}002$;
Katsuura Function	$f_{10}(X)=\frac{10}{D^2}\prod_{i=1}^{D}(1+i\sum_{j=1}^{32}\frac{\lvert 2^jX_i-round(2^jX_i)\rvert}{2^j})^{\frac{10}{D^{12}}}-\frac{10}{D^2}$
HappyCat Function	$f_{11}(X)=\left\lvert(\sum_{i=1}^{D}X_i^2)^2-D\right\rvert^{1/4}+(0.5\sum_{i=1}X_i^2+\sum_{i=1}^{D})/D+0.5$
HGBat Function	$f_{12}(X)=\left\lvert(\sum_{i=1}^{D}X_i^2)^2-(\sum_{i=1}^{D}X_i^2)\right\rvert^{1/2}+(0.5\sum_{i=1}X_i^2+\sum_{i=1}^{D})/D+0.5$
Expanded Griewank's plus Rosenbrock's Function	$f_{13}=f_7(f_4(X_1,X_2))+f_7(f_4(X_2,X_3))+\ldots.f_7(f_4(X_{D-1},X_D))+f_7(f_4(X_D,X_1))$
Expanded Scaffer's F6 Function	$g(x,y)=0.5+\frac{(sin^2(\sqrt{X^2+y^2})-0.5)}{(1+0.001(X^2+y^2))^2};\ f_{14}=g(X_1,X_2)+g(X_2,X_3)+\ldots.g(X_{D-1},X_D)+g(X_D,X_1)$

different algorithms, and proposing new algorithms. The second popular research direction deals with hybridizing different algorithms to improve the performance [37–41]. There are some hybrid meta-heuristics in literature such as: BA-CGSA [42], STSA [43], MABC [44], PSO-GA [45], PSO-ACO [46], ACO-GA [47].

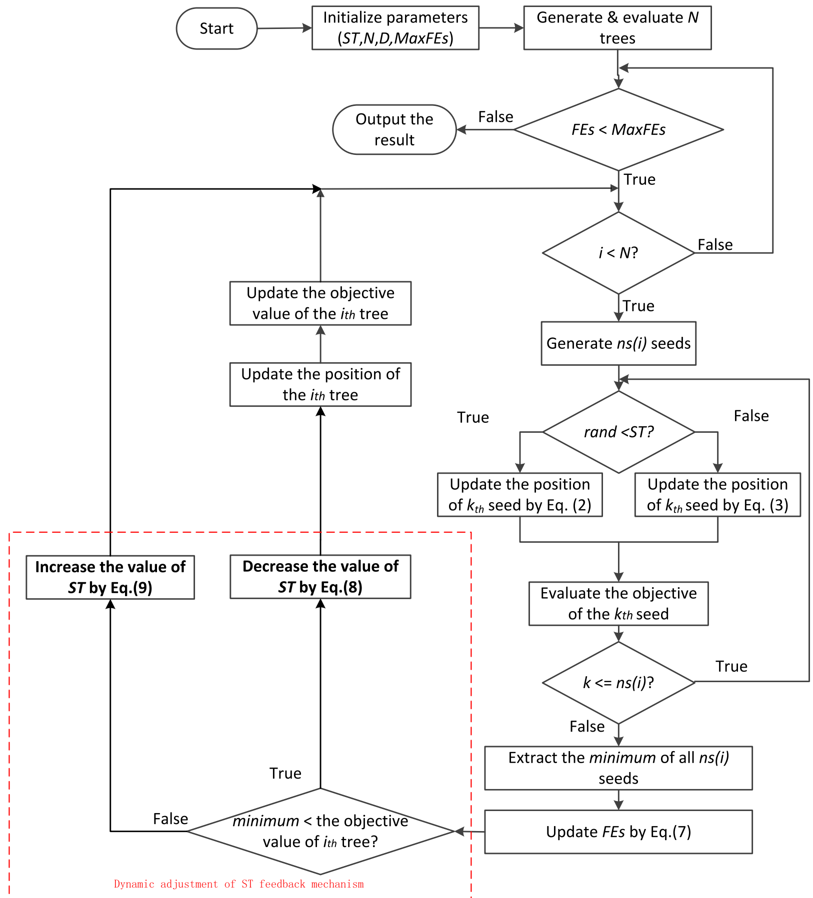


Fig. 2. Flow chart of st_TSA.

Although a lot of hybrid algorithms have been put forward in this field recently, there is no a concrete optimization algorithm to achieve best optima for all optimization problems. Based on no free lunch (NFL) theorem [48], a new hybrid optimization algorithm needs to be proposed.

Tree-Seed Algorithm (TSA) [49] was put forward by Kiran in 2015, which is a population-based on evolutionary approaches inspired by the relationship between trees and seeds, trees send their seeds to the surface in propagation. And TSA has some important **advantages**, such as : (1) it is simple and easy to be implemented; (2) due to the seed generating mechanism, it has strong capability of exploitation. It is proved that the TSA has excellent optimization ability to solve continuous problems [49–52]. Many variants of TSA are proposed to enhance the capability of global searching. The major enhancements can be summarized as follows:

- For unconstrained optimization problems, Ahmet Babalika et al. (2017) [50,51] adopts Deb's rules to enhance the search capability of TSA.
- For complex optimization problems, Murat Aslan et al. (2018) [52] proposed the Improved Tree-Seed Algorithm (ITSA) to solve it.
- For continuous optimization problems, Jianhua Jiang et al. has proposed the Sine Tree-Seed Algorithm (STSA) with the inspiration of SCA to enhance the seed generation mechanism in TSA [43]. Even more, Jianhua Jiang et al. have proposed a new balance mechanism between exploration and exploitation in EST-TSA [53].

Currently, variants of TSA focus on different optimization problems, such as unconstrained optimization problems [50,51], complex optimization problems [52] and continuous problems [43,53]. However, the feedback mechanism of TSA is not enhanced. For instance, the parameters of *ST* (search tendency) and *ns* (number of seeds) are given in a constant or random value without dynamic adjustment via feedback mechanism.

For the paper, the **motivations** are as following:

- When parameter *ST* takes different values, the optimal values are different. That is to say, when the *ST* value is not fixed, the optimal value also changes, so the *ST* value is dynamically adjusted by feed-back mechanism.
- The parameter *ns* determines the local optimum, and the seed changes constantly update the tree, so the parameter *ns* is dynamically adjusted by the feedback mechanism.

The feedback mechanism of TSA is a good design for the evolution of trees and seeds. Seeds are generated by a tree and a tree will be replaced by a better seed that has a minimal value of all seeds generated by the tree. However, the definition of *ST* and *ns* is very important to balance exploration and exploitation in order to find the global optimal solution. It is obvious that the *ns* value should be increased when better seeds are found around a tree. That is to say, the *ns* value given with the random principle is not scientific. Even more, the *ST* is a balancing factor to determine Eqs. (2) and (3) based on TSA [49]. Nevertheless, it is unreasonable because it is common sense that the *ST* value is fixed through the whole procedure of searching for optimal solutions. Hence, **two hypothesizes** are given in this paper.

- Hypothesis 1 : Dynamic *ST* value determined by the feedback mechanism helps to find global optimal value through the whole searching process.

Table 2
Benchmark functions of CEC 2014.

A. Unimodal functions:	
Rotated High Conditioned Elliptic Function	$F_1(x) = f_1(M(x - o_1)) + 100$
Rotated Bent Cigar Function	$F_2(x) = f_2(M(x - o_2)) + 200$
Rotated Discus Function	$F_3(x) = f_3(M(x - o_3)) + 300$
B. Multimodal functions:	
Shifted and Rotated Rosenbrock's Function	$F_4(x) = f_4(M(\frac{2.048(x - o_4)}{100}) + 1) + 400$
Shifted and Rotated Ackley's Function	$F_5(x) = f_5(M(x - o_5)) + 500$
Shifted and Rotated Weierstrass Function	$F_6(x) = f_6(M(\frac{0.5(x - o_6)}{100})) + 600$
Shifted and Rotated Griewank's Function	$F_7(x) = f_7(M(\frac{600(x - o_7)}{100})) + 700$
Shifted Rastrigin's Function	$F_8(x) = f_8(M(\frac{5.12(x - o_8)}{100})) + 800$
Shifted and Rotated Rastrigin's Function	$F_9(x) = f_8(M(\frac{5.12(x - o_9)}{100})) + 900$
Shifted Schwefel's Function	$F_{10}(x) = f_9(M(\frac{1000(x - o_{10})}{100})) + 1000$
Shifted and Rotated Schwefel's Function	$F_{11}(x) = f_9(M(\frac{1000(x - o_{11})}{100})) + 1100$
Shifted and Rotated Katsuura Function	$F_{12}(x) = f_{10}(M(\frac{5(x - o_{12})}{100})) + 1200$
Shifted and Rotated HappyCat Function	$F_{13}(x) = f_{11}(M(\frac{5(x - o_{13})}{100})) + 1300$
Shifted and Rotated HGBat Function	$F_{14}(x) = f_{12}(M(\frac{5(x - o_{14})}{100})) + 1400$
Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	$F_{15}(x) = f_{13}(M(\frac{5(x - o_{15})}{100}) + 1) + 1500$
Shifted and Rotated Expanded Scaffer's F6 Function	$F_{16}(x) = f_{14}(M((x - o_{16}))1) + 1600$
Hybrid functions	
$F_{17} = f_9(M_1Z_1) + f_8(M_2Z_2) + f_3(M_3Z_3) + 1700$	p=[0.3,0.3,0.4]
$F_{18} = f_2(M_1Z_1) + f_8(M_2Z_2) + f_3(M_3Z_3) + 1800$	p=[0.3,0.3,0.4]
$F_{19} = f_7(M_1Z_1) + f_8(M_2Z_2) + f_3(M_3Z_3) + f_8(M_4Z_4) + 1900$	p=[0.2,0.2,0.3,0.3]
$F_{20} = f_{12}(M_1Z_1) + f_3(M_2Z_2) + f_{13}(M_3Z_3) + f_8(M_4Z_4) + 2000$	p=[0.2,0.2,0.3,0.3]
$F_{21} = f_{14}(M_1Z_1) + f_{12}(M_2Z_2) + f_4(M_3Z_3) + f_9(M_4Z_4) + f_1(M_5Z_5) + 2100$	p=[0.1,0.2,0.2,0.2, 0.3]
$F_{22} = f_{10}(M_1Z_1) + f_{11}(M_2Z_2) + f_{13}(M_3Z_3) + f_9(M_4Z_4) + f_5(M_5Z_5) + 2200$	p=[0.1,0.2,0.2,0.2, 0.3]
Notes:	
$Z_1 = [y_{s_1}, y_{s_1} \ldots\ldots, y_{s_{n1}}]$	
$Z_2 = [y_{s_{n1+1}}, y_{s_{n1+2}} \ldots\ldots, y_{s_{n1+n2}}]$	
$Z_N = [y_{s_{\sum_{i=1}^{N-1} n1+1}}, y_{s_{\sum_{i=1}^{N-1} n1+2}} \ldots\ldots, y_{s_{5D}}]$	
$y = x - o_i$, $S = randperm(1 : D)$, $percentageofg_i(x)$	
$n_1 = [p_1D]$, $n_2 = [p_2D]$, …, $n_{N-1} = [p_{N-1}D]$, $n_N = D - \sum_{i=1}^{N-1} n + i$	
Composition functions	
$F_{23} = w_1 * F'_4(x) + w_2 * [1e^{-6}F'_1(x) + 100] + w_3 * [1e^{-26}F'_2(x) + 200]$ $+ w_4 * [1e^{-6}F'_3(x) + 300] + w_5 * [1e^{-6}F'_1(x) + 400] + 2300$	$\sigma = [10, 20, 30, 40, 50]$
$F_{24} = w_1 * F'_{10}(x) + w_2 * [F'_9(x) + 100] + w_3 * [F'_{14}(x) + 200] + 2400$	$\sigma = [20, 20, 20]$
$F_{25} = w_1 * 0.25F'_{11}(x) + w_2 * [F'_9(x) + 100] + w_3 * [1e^{-7}F'_1(x) + 200] + 2500$	$\sigma = [10, 30, 50]$
$F_{26} = w_1 * 0.25F'_{11}(x) + w_2 * [F'_{13}(x) + 100] + w_3 * [1e^{-7}F'_1(x) + 200]$ $+ w_4 * [2.5F'_6(x) + 300] + w_5 * [1e^{-6}F'_{13}(x) + 400] + 2700$	$\sigma = [10, 10, 10, 10, 10]$
$F_{27} = w_1 * 10F'_{14}(x) + w_2 * [10F'_9(x) + 100] + w_3 * [2.5F'_1 1(x) + 200]$ $+ w_4 * [25F'_{16}(x) + 300] + w_5 * [1e^{-6}F'_1(x) + 400] + 2700$	$\sigma == [10, 10, 10, 20, 20]$
$F_{28} = w_1 * 2.5F'_{15}(x) + w_2 * [10F'_9(x) + 100] + w_3 * [2.5F'_1 1(x) + 200]$ $+ w_4 * [5e^{-4}F'_{16}(x) + 300] + w_5 * [1e^{-6}F'_1(x) + 400] + 2800$	$\sigma == [10, 20, 30, 40, 50]$
$F_{29} = w_1 * F'_{17}(x) + w_2 * [F'_{18}(x) + 100] + w_3 * [F'_{19}(x) + 200] + 2900$	$\sigma = [10, 30, 50]$
$F_{30} = w_1 * F'_{20}(x) + w_2 * [F'_{21}(x) + 100] + w_3 * [F'_{22}(x) + 200] + 3000$	$\sigma = [10, 30, 50]$
Notes:	
$w_i = \frac{1}{\sqrt{\sum_{j=1}^{D}(x_j - o_{ij})}} exp(-\frac{\sum_{j=1}^{D}(x_i - o_{ij}^2)}{2D\sigma_i^2})$	

- Hypothesis 2 : *ns* value determined by feedback mechanism enhances the capability of finding a global optimal solution.

In the paper, we propose st_TSA and ns_TSA to validate the above two assumptions respectively. Furthermore, fb_TSA algorithm is proposed to aggregate advantages from st_TSA and

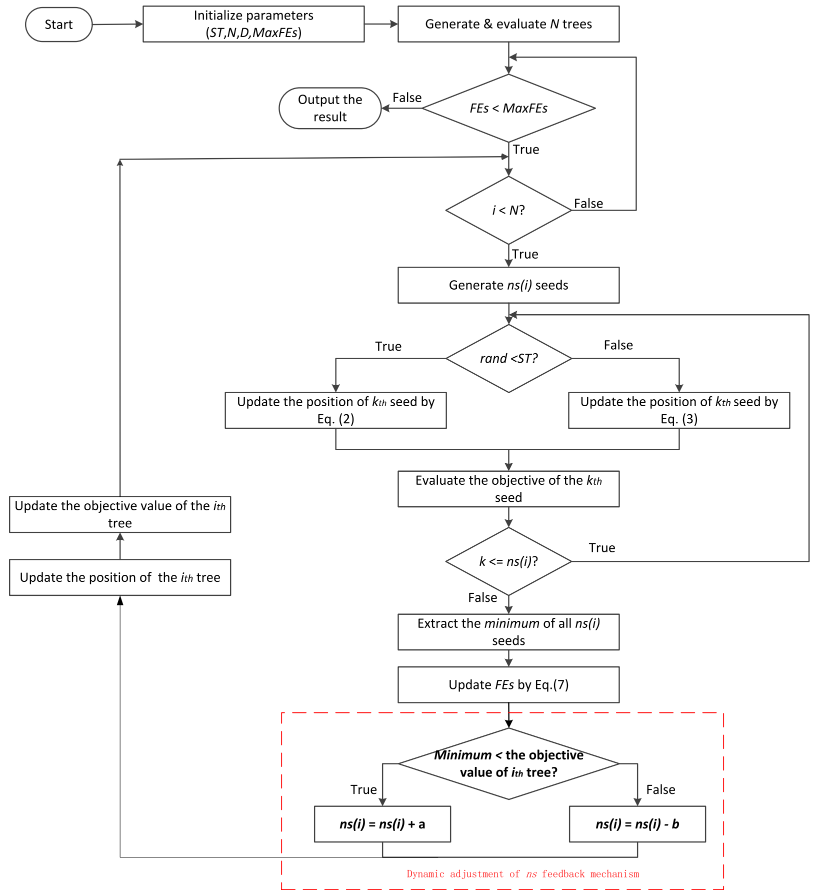


Fig. 3. Flow chart of ns_TSA.

ns_TSA with an integrated feedback mechanism to achieve a better global optimal solution. Therefore, the major **contributions** of this paper can be summarized as:

- The capability of balancing exploration and exploitation is enhanced by the feed-back mechanism in TSA.
- Search tendency (*ST*) is redefined to improve the capability of finding global optimal solutions.
- The seeds generation mechanism is redesigned with the heuristic inspiration from the result of the last seeds.

This paper is presented as follows: In Section 2, firstly, the theory and principle of the basic TSA is depicted and explained in Section 2.1; secondly, how to redefine *ST* is explained in the proposed st_TSA in Section 2.2; thirdly, how to redesign the *ns* generation mechanism is proposed in ns_TSA in Section 2.3; lastly, the enhanced feedback mechanism with *ST* and *ns* is proposed in Section by 2.4. In Section 3, the advantage of the proposed st_TSA, ns_TSA, fb_TSA are proved by experiments in detail. In Section 4, discussions are given to analyse the underlying reasons. In Section 5, the proposed algorithm is applied and the result is listed. Finally, conclusion and future works are given in the last Section 6.

2. Method

The efficiency of optimization algorithms is usually determined by their ability to find a global optimal solution, and the feedback mechanism can make the algorithm more effective [54, 55]. The feedback mechanism can balance the exploration and exploitation better [56,57]. Meanwhile, the feed-back mechanism is used in some algorithms [58], such as PSO [59], ABC [60], ACO [61] and others [62]. In TSA, *ST* and *ns* are two vital parameters to achieve the global optimum. Hence, it is necessary to combine feed-back with *ST* and *ns* to get better optimum.

2.1. TSA: Tree-Seed Algorithm

In the light of the metaheuristic algorithm, TSA is proposed by Kiran (2015) as a novel algorithm for resolving of continuous optimization problems [49]. TSA mimics the appearance that trees send their seeds to the surface in the propagation and it is a combination of trees and seeds to solve continuous optimization problems. In its initialization phase, trees are generated in the search space by using the Eq. (1).

$$T_{i,j} = L_{j,min} + r_{i,j} \times (H_{j,max} - L_{j,min}) \tag{1}$$

where $T_{i,j}$ is the j_{th} dimension of the i_{th} tree, $r_{i,j}$ is a random number in range of [0,1] produced for each dimension and location, $H_{j,max}$ is the higher bound of the search space, and $L_{j,min}$ is the lower bound of the search space. Any seed comes from a tree. In TSA, seeds are generated through Eqs. (2) and (3).

$$S_{i,j} = T_{i,j} + \alpha_{i,j} \times (B_j - T_{r,j}) \tag{2}$$

$$S_{i,j} = T_{i,j} + \alpha_{i,j} \times (T_{i,j} - T_{r,j}) \tag{3}$$

where $S_{i,j}$ is the j_{th} dimension of the i_{th} seed which is produced by the i_{th} tree, $\alpha_{i,j}$ which is the scaling factor that is produced in range of [-1, 1] randomly, B_j is the j_{th} dimension of the best tree location obtained so far (B), $T_{r,j}$ is the j_{th} dimension of a tree randomly selected from the population, the number of the seeds produced from a tree is controlled with principle of Kiran (2015) [49]. The minimum (S_{min}) (10%) and maximum (S_{max}) (25%)

values of the number of seeds are generated by using the Eqs. (4) and (5).

$$S_{min} = 10\% \times N \tag{4}$$

$$S_{max} = 25\% \times N \tag{5}$$

For all experiments, the termination condition depends on the maximum times of function evaluations (*MaxFEs*) by Eq. (6), and function evaluations (*FEs*) are updated by Eq. (7).

$$MaxFEs = D \times 10000 \tag{6}$$

$$FEs = FEs + ns \tag{7}$$

where *ns* is the number of seeds produced by a tree. The detailed algorithmic framework of TSA is given in Algorithm 1 and its flow chart is given in Fig. 1.

Algorithm 1: The basic Tree-Seed Algorithm

Step 1: Initialize parameters:
1.1 Set up the number of tree population (N)
1.2 Put up the search tendency *ST* parameter (***ST*= 0.1**)
1.3 Set the problem dimensions (D)
1.4 Initial the *FEs* to record the number of function evaluations (*FEs*)
1.5 Determine the termination condition (*MaxFEs*)
1.6 i and k are indicated the sequence of any tree and seed respectively (i,k)
Step 2:
While $FEs < MaxFEs$
2.1 For i : N
2.1.1 Generate the number of seeds for each tree ($ns(i)$) randomly
2.1.2 For $k <= ns(i)$
Update the position of the k_{th} seed;
Evaluate the objective of the k_{th} seed;
End For
2.1.3 Extract the minimum objective value (*minimum*) of all *ns(i)* seeds
2.1.4 Update *FEs* by Eq. (7)
2.1.5 If $minimum <$ the objective value of the i_{th} tree
Update the position of the i_{th} tree
Update the objective value of the i_{th} tree
End If
End For
End While

Problems of TSA concerned in this paper is as follows :

Problem [1] It is unreasonable to set the value of *ST* as a constant [49]. As shown in Tables 3–6, when the value of *ST* changes, the optimum will change accordingly. When the value of *ST* is smaller, the optimal value will be smaller. Hence, the value of *ST* being a constant is not a good choice in the basic TSA.

Problem [2] It is unscientific to generate seeds randomly. According to Kiran (2015) [49], the setting of parameters *ns* is random. As shown in Tables 7–10, experiments and observation demonstrate that the optimal value will change when *ns* changes. Hence, the optimum is influenced by the number of seeds. And the relation between the number of seeds and the better number of seeds affects the algorithm's optimum, so the setting of parameter *ns* is unreasonable. By summarizing the above disadvantages and problems, the following three subsections will solve them pertinently.

2.2. st_TSA: Tree-Seed Algorithm with ST feed-back mechanism

According to Kiran (2015) [49], parameter *ST* controls the Eqs. (2) and (3) to generate the new location of seeds from a tree in range of [0,1]. According to Tables 3–6, the dynamic adjustment of *ST* influences the optimum value. Nevertheless, when the value of *ST* gets lower, the convergence is slower but the global search is powerful. In other words, the exploration and exploitation capabilities of the TSA are controlled by the *ST* parameter [49]. In the proposed st_TSA, a *ST* feedback mechanism is proposed to change *ST* value dynamically in the search procedure. In TSA, there is no feed-back mechanism for *ST* value. In the whole search procedure, the *ST* value is fixed. However, as the illustration in Tables 3–6, it has been proved unreasonable,

The feedback mechanism of *ST* is proposed since the *ST* can balance the exploration and exploitation to find the optimum. According to Tables 3–6, illustrate that the *ST* happens to change and the optimum changes. The main step of the st_TSA is shown in Algorithm 2, and the workflow is expressed in Fig. 2. When the *minimum* is smaller than the objective, *ST* value is updated by Eq. (8). It is used to reduce local search opportunities and increase global search opportunities. Conversely, Eq. (9) is used to increase the chance of local search and improve the efficiency of finding the optimum.

$$ST = ST - x \tag{8}$$

$$ST = ST + y \tag{9}$$

where, through constant experimentation and common sense that the value of x is 0.02 and the value of y is 0.04 in this paper.

Algorithm 2 :st_TSA: Tree-Seed Algorithm with *ST* feed-back mechanism

Step 1: Initialize parameters:
1.1 Set up the number of tree population (N)
1.2 Put up the search tendency *ST* parameter (*ST*)
1.3 Set the problem dimensions (D)
1.4 Initial the *FEs* to record the number of function evaluations (*FEs*)
1.5 Determine the termination condition (*MaxFEs*)
1.6 i and k are indicated the sequence of any tree and seed respectively (i,k)
Step 2:
While $FEs < MaxFEs$
2.1 For i : N
2.1.1 Generate the number of seeds for each tree ($ns(i)$) randomly
2.1.2 For $k <= ns(i)$
Update the position of the k_{th} seed;
Evaluate the objective of the k_{th} seed;
End For
2.1.3 Extract the minimum objective value (*minimum*) of all $ns(i)$ seeds
2.1.4 Update *FEs* by Eq. (7)
2.1.5 **If *minimum* < the objective value of the i_{th} tree**
Update the value of *ST* by Eq. (8)
Else
Update the value of *ST* by Eq. (9)
End If
End For
End While

By enhancing feedback mechanism, the value of *ST* can be updated dynamically and revised continuously to achieve relative good results, the better performance of the proposed st_TSA is verified in Tables 7–10.

2.3. ns_TSA: Tree-Seed Algorithm with ns feed-back mechanism

The seed plays a role in spreading to find optimal values, but the number of seeds generated from trees randomly is unreasonable. When the number of seeds generated is random, many situations may happen, such as falling into the local optimum situation or ignoring local optimum. These will reduce the efficiency of the algorithm. So we propose a new seed generation mechanism to improve its efficiency. The main step is expressed in Algorithm 3, and the flow chart is shown in Fig. 3. The number of seeds is initialized by Eq. (10).

$$ns_i = (S_{min} + S_{max}) \times 0.5 \tag{10}$$

where ns_i refers to the number of seeds generated by the i_{th} tree. As shown Tables 7–10 shows, when the value of seed is better than the objective value of trees, the number of seeds will be increased to enhance the local search and the number of seeds is updated by Eq. (11) to form i_{th} tree. By contraries, when the value of seed is worse than the objective value of trees, the number of seeds will be reduced to increase the global search and Eq. (12) is used to update the number of seeds for i_{th} tree. Hence, the ns_i should be considered to have dynamic changes to choose a reasonable value. Through experiments, a and b are determined as 2 and 2 in this paper, respectively

$$ns_i = ns_i + a \tag{11}$$

$$ns_i = ns_i - b \tag{12}$$

Algorithm 3 ns_TSA: Tree-Seed Algorithm with *ns* feed-back mechanism

Step 1: Initialize parameters:
1.1 Set up the number of tree population (N)
1.2 Put up theST parameter (ST)
1.3 Set the problem dimensions (D)
1.4 Initialize the FEs to record the number of function evaluations (FEs)
1.5 Determine the termination condition ($MaxFEs$)
1.6 i and k are indicated the sequence of any tree and seed respectively (i,k)
Step 2:
While $FEs < MaxFEs$
1.7 Calculate the number of seeds initialized through Eq. (10)
2.1 For $i : N$
2.1.1 Generate the number of seeds ($ns(i)$) for each tree randomly
2.1.2 For $k <= ns(i)$
Update the position of the k_{th} seed;
Evaluate the objective of the k_{th} seed;
End For
2.1.3 Extract the minimum objective value (*minimum*) of all $ns(i)$ seeds
2.1.4 Update FEs by Eq. (7)
2.1.5 **If *minimum* < the objective value of the i_{th} tree**
Update the number of *ns* by Eq. (11)
Else
Update the number of *ns* by Eq. (12)
End If
End For
End While

On the basis of Kiran (2015) [49], *ns* affects the procession of searching to the optimal value. Once the seed is the optimal value, it is updated immediately. The feed-back mechanism is used to update the optimal value continuously. When the optimal value is the seed, the possibility of local searching increases. Otherwise, the possibility of local searching will be reduced.

2.4. fb_TSA: Tree-Seed Algorithm with feedback mechanism of updating ns *and* ST *values*

Exploration and exploitation are two important parts for the evolutionary algorithm, but exploration and exploitation are unbalanced in TSA. According to Sections 2.2 and 2.3, the dynamic adjustment of ST can make the exploration and exploitation achieve better balance. The number of seeds changes by Eqs. (11) and (12), it can balance the opportunity to find the local and global optimum. *ns* constantly changes to decrease the unnecessary search and increase the chance of seeds to become local optimum. So we can combine Algorithm 2 and Algorithm 3 to balance, and the integration is called Tree-Seed Algorithm with Feed-Back mechanism (fb_TSA). fb_TSA can make the balance of exploration and exploitation. By using the Eqs. (11) and (12) to change the number of seeds and using the Eqs. (8) and (9) to adjust the value of ST dynamically. The constant adjustment for the value of ST and *ns* is helpful to balance exploration and exploitation. When the number of seeds is equal with the low number of seeds, the way of generating seeds will be changed, it is generated by Eq. (13).

$$S_{i,j} = T_{r,j} + \alpha_{i,j} \times (T_{i,j} - T_{r,j}) \tag{13}$$

The flow chart of fb_TSA is shown in Fig. 4, and the process is shown in detail in Algorithm 4. The experiment shows that the proposed algorithm can achieve a better balance. The results are shown in Tables 11 and 12.

Algorithm 4 fb_TSA: Tree-Seed Algorithm with Feed-Back mechanism of updating *ns* and *ST* values

Step 1: Initialize parameters:
1.1 Set up the number of tree population (N)
1.2 Put up the ST parameter (ST)
1.3 Set the problem dimensions (D)
1.4 Initial the FEs metric to record the times of function evaluations (FEs)
1.5 Determine the termination condition ($MaxFEs$)
1.6 i and k are indicated the sequence of any tree and seed respectively (i,k)
1.7 Calculate the number of seeds initialized through the Eq. (10)
Step 2:
While $FEs < MaxFEs$
2.1 For $i : N$
2.1.1 Generate the number of seeds for each tree ($ns(i)$) randomly
2.1.2 For $k <= ns(i)$
If ***ns* ==*nSeedLow***
Update the position of the k_{th} seed by Eq. (13)
else
If *rand* <ST
Update the position of the k_{th} seed by Eq. (2);
Else
Update the position of the k_{th} seed by Eq. (3);
End If
Evaluate the objective of the k_{th} seed ;
End If
End For
2.1.3 Extract the minimum objective value (*minimum*) of all $ns(i)$ seeds
2.1.4 Update FEs by Eq. (7)
2.1.5 If *minimum* < the objective value of the i_{th} tree
Update the number of *ns* by Eq. (11)
Update the value of *ST* by Eq. (8)
Else
Update the number of *ns* by Eq. (12)
Update the value of *ST* by Eq. (9)
End If
End For
End While

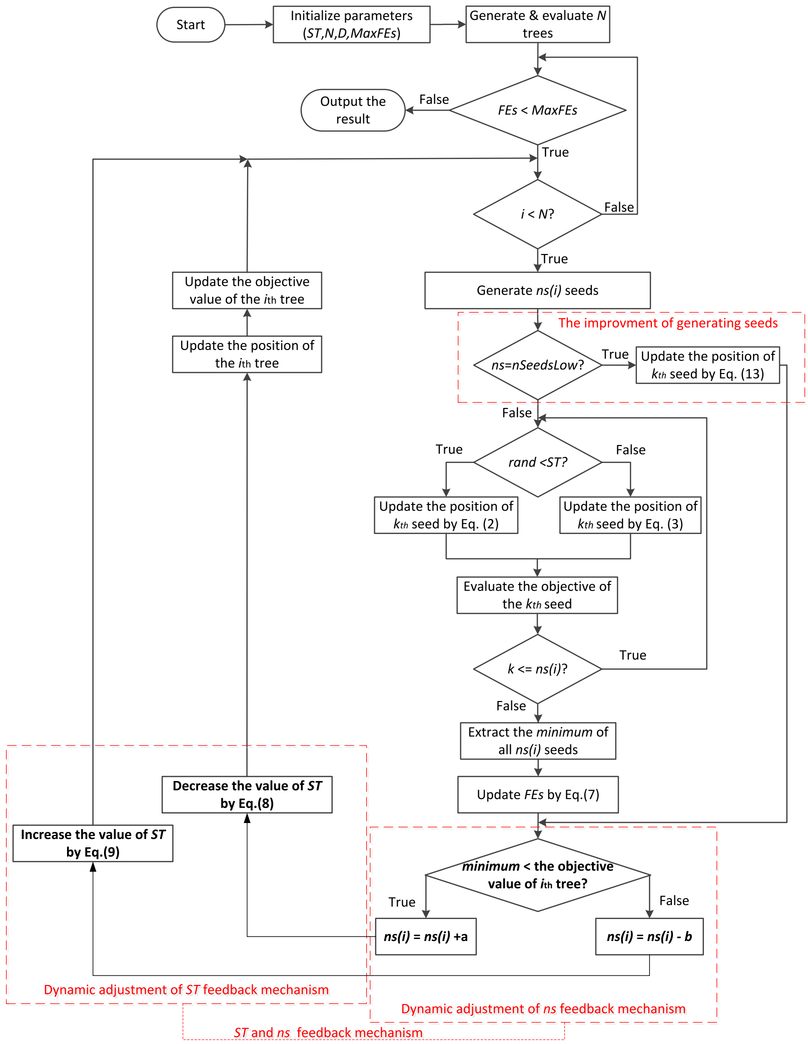


Fig. 4. Flow chart of fb_TSA.

3. Experiments and results

3.1. Experimental fundamentals

The quality of the capability of the proposed algorithm is tested by CEC2014 and it contains 30 test benchmark functions, such as unimodal functions, simple multimodal functions, hybrid functions, and composition functions. They are listed in Tables 1 and 2. 30 random experiments are performed by CEC 2014 in four different dimensions, including 10, 30, 50 and 100.

3.2. Experimental environment

For these experiments, the variants are coded in Matlab R2015b environment under the Windows 10 operating system, all simulations are running on the computer with Intel Core(TM) i3-6100 CPU @ 3.70 GHz and its memory is 8G.

3.3. Specific parameters

To better demonstrate the advantages of fb_TSA, it is compared to TSA [49], STSA [43], GWO [63], ABC [28,29], SCA [64], DE [20], PSO [26], CLPSO [27]. The population size of all methods is set to 30 and max iteration is 500. The *c1* (coefficient of the cognitive component) and *c2* (coefficient of the social component) as PSO algorithm control parameters are set to 2 and 2 respectively. ABC algorithm has one control parameter named as a limit specific to the method which is set to 180. CLPSO algorithm is a variant of PSO and *c1* and *c2* for CLPSO are 1.49445 and 1.49445, respectively.

3.4. Test the influence factor of ST parameter in four different dimensions

The performance and efficiency of the proposed st_TSA are tested on thirty well-known benchmark functions and listed in Tables 1 and 2. The results are listed in Tables 3–6. Based on different dimensions, dynamic changes for the value of *ST* can change the optimum. When the *x* and *y* vary in a certain range, the optimum also has slight changes. In the light of four tables, dynamic changes for *ST* can change the balance of the exploration and exploitation and the Eqs. (2) and (3) are used effectively.

3.5. Test the influence factor of ns parameter in four different dimensions

The performance of proposed ns_TSA is also tested on 30 benchmark functions, and the relationship of the number of better seeds and the number of seeds has a certain correlation. And

Table 3
st_TSA: Dynamic changes of the value of *ST* for 30 benchmark functions in D = 10.

Function	x/y			
	0.02/0.04	0.001/0.04	0.1/0.04	0.02/0.1
F_1	120579.69	143366.65	156127.86	157701.37
F_2	890.40	607.86	877.78	944.92
F_3	2014.66	1905.68	2076.17	1866.52
F_4	400.74	401.25	402.93	402.27
F_5	519.65	519.91	519.80	519.79
F_6	600.00	600.00	600.00	600.00
F_7	700.21	700.22	700.22	700.23
F_8	807.04	806.16	805.81	805.92
F_9	921.86	920.33	919.70	920.15
F_{10}	1104.32	1122.47	1129.80	1112.74
F_{11}	1986.17	2023.81	2041.67	2054.22
F_{12}	1201.21	1201.16	1201.12	1201.24
F_{13}	1300.18	1300.19	1300.19	1300.17
F_{14}	1400.21	1400.20	1400.21	1400.21
F_{15}	1502.23	1502.26	1502.40	1502.21
F_{16}	1602.67	1602.69	1602.65	1602.75
F_{17}	4759.82	4697.38	4155.88	4290.16
F_{18}	3090.61	2911.49	3533.58	3238.33
F_{19}	1900.97	1900.96	1900.90	1900.92
F_{20}	2193.95	2215.25	2237.43	2171.47
F_{21}	2438.99	2445.43	2451.44	2491.99
F_{22}	2211.66	2213.39	2212.76	2211.58
F_{23}	2629.46	2629.46	2629.46	2629.46
F_{24}	2521.92	2522.37	2522.04	2523.69
F_{25}	2664.35	2668.27	2664.63	2666.96
F_{26}	2700.19	2700.18	2700.18	2700.19
F_{27}	2793.44	2818.00	2781.00	2782.21
F_{28}	3149.82	3157.18	3186.16	3176.42
F_{29}	3672.45	3732.17	3727.34	3850.37
F_{30}	4307.48	4211.99	4304.75	4226.16

Table 4
st_TSA: Dynamic changes of the value of *ST* for 30 benchmark functions in D = 30.

Function	x/y			
	0.02/0.04	0.001/0.04	0.1/0.04	0.02/0.1
F_1	8.55E+07	9.50E+07	9.48E+07	8.79E+07
F_2	3.52E+05	4.31E+05	4.61E+05	3.61E+05
F_3	42403.19	41006.05	38807.03	42721.43
F_4	555.40	533.72	547.91	547.96
F_5	520.98	521.00	520.99	521.02
F_6	622.11	622.98	623.11	623.72
F_7	700.03	700.02	700.04	700.02
F_8	966.29	964.94	975.14	969.45
F_9	1116.42	1122.02	1119.76	1115.61
F_{10}	6246.49	5942.79	6152.42	6232.02
F_{11}	8258.63	8142.86	8172.20	8236.16
F_{12}	1202.86	1202.74	1202.81	1202.76
F_{13}	1300.50	1300.50	1300.51	1300.50
F_{14}	1400.33	1400.32	1400.32	1400.32
F_{15}	1519.96	1519.54	1519.81	1519.91
F_{16}	1612.74	1612.76	1612.73	1612.70
F_{17}	1.85E+06	2.31E+06	1.68E+06	2.01E+06
F_{18}	2162.65	2276.98	2330.89	2263.44
F_{19}	1907.92	1907.82	1908.00	1908.24
F_{20}	17677.63	16709.29	16917.73	17038.99
F_{21}	4.16E+05	3.87E+05	3.66E+05	3.49E+05
F_{22}	2742.69	2699.92	2719.30	2699.70
F_{23}	2615.27	2615.27	2615.27	2615.27
F_{24}	2627.58	2627.52	2627.41	2627.47
F_{25}	2723.20	2722.21	2722.92	2723.05
F_{26}	2700.60	2700.56	2700.59	2700.72
F_{27}	3218.27	3172.02	3210.21	3192.01
F_{28}	4041.63	4039.32	4033.98	4028.66
F_{29}	24793.31	27046.46	35228.97	26886.21
F_{30}	13072.70	13581.42	14091.94	13446.83

the number of seeds changes dynamically by Eqs. (11) and (12).

In Tables 7–10, the results are presented.

Table 5
st_TSA: Dynamic changes of the value of *ST* for 30 benchmark functions in D = 50.

Function	x/y			
	0.02/0.04	0.001/0.04	0.1/0.04	0.02/0.1
F_1	3.49E+08	3.19E+08	3.34E+08	3.48E+08
F_2	2.43E+08	1.69E+08	2.07E+08	1.86E+08
F_3	125461.52	126847.52	119272.47	124440.71
F_4	956.34	973.85	972.36	958.76
F_5	521.18	521.17	521.19	521.19
F_6	654.60	653.98	654.68	652.87
F_7	702.18	701.84	702.11	702.01
F_8	1205.95	1217.27	1189.97	1203.77
F_9	1368.10	1363.22	1363.41	1361.35
F_{10}	12556.59	12561.87	12665.82	12571.19
F_{11}	14839.77	14735.92	14735.53	14835.90
F_{12}	1203.72	1203.76	1203.85	1203.75
F_{13}	1300.76	1300.72	1300.76	1300.74
F_{14}	1400.44	1400.44	1400.41	1400.42
F_{15}	1856.99	1831.26	1857.27	1828.71
F_{16}	1622.51	1622.53	1622.48	1622.56
F_{17}	1.78E+07	1.55E+07	1.65E+07	1.54E+07
F_{18}	2677.38	2796.17	2765.09	2864.93
F_{19}	1968.51	1969.98	1970.53	1966.26
F_{20}	42144.92	46176.33	46070.98	37967.16
F_{21}	6.53E+06	6.16E+06	6.77E+06	7.14E+06
F_{22}	3877.58	3914.31	3905.72	3855.75
F_{23}	2646.22	2646.07	2646.17	2645.97
F_{24}	2694.78	2693.53	2694.29	2694.55
F_{25}	2769.92	2772.40	2771.72	2771.99
F_{26}	2761.75	2776.64	2747.59	2746.11
F_{27}	4192.30	4203.48	4198.80	4219.09
F_{28}	5833.91	5847.92	5770.75	5873.97
F_{29}	1.21E+06	1.05E+06	1.11E+06	1.17E+06
F_{30}	1.34E+05	1.36E+05	1.13E+05	1.34E+05

Table 6
st_TSA: Dynamic changes of the value of *ST* for 30 benchmark functions in D = 100.

Function	x/y			
	0.02/0.04	0.001/0.04	0.1/0.04	0.02/0.1
F_1	2.38E+09	2.39E+09	2.32E+09	2.29E+09
F_2	5.34E+10	5.28E+10	5.28E+10	5.17E+10
F_3	342324.14	350008.12	356023.33	356126.63
F_4	8912.98	8590.09	8781.36	8850.82
F_5	521.36	521.36	521.37	521.37
F_6	743.33	742.81	743.85	743.64
F_7	1175.70	1136.10	1158.04	1158.93
F_8	1894.81	1889.46	1863.41	1871.78
F_9	2098.22	2076.13	2103.53	2096.92
F_{10}	29825.47	29610.81	29706.01	29453.49
F_{11}	32372.35	32183.21	32027.41	32128.58
F_{12}	1204.64	1204.55	1204.49	1204.58
F_{13}	1303.38	1303.29	1303.42	1303.41
F_{14}	1532.31	1533.55	1535.27	1532.10
F_{15}	711200.83	599522.18	613187.23	579062.88
F_{16}	1646.90	1646.92	1646.95	1646.90
F_{17}	2.21E+08	2.11E+08	2.17E+08	2.08E+08
F_{18}	2925.30	3873.09	3091.23	3181.64
F_{19}	2146.09	2142.54	2142.39	2149.12
F_{20}	258120.80	256586.28	272802.09	259771.47
F_{21}	9.04E+07	8.54E+07	9.58E+07	9.48E+07
F_{22}	7043.48	6975.28	6972.78	7093.97
F_{23}	2767.61	2765.08	2765.53	2763.77
F_{24}	2992.75	2991.42	2992.32	2989.50
F_{25}	3045.71	3044.33	3042.34	3039.12
F_{26}	2995.37	2991.78	2985.54	2989.51
F_{27}	6408.97	6368.61	6341.93	6366.62
F_{28}	20230.98	20392.11	20018.54	20096.33
F_{29}	2.40E+07	2.21E+07	2.28E+07	2.53E+07
F_{30}	2.55E+06	2.39E+06	2.40E+06	2.30E+06

3.6. *Test the influence from the feedback mechanism of* ns *and* ST *values in four different dimensions*

The optimum of fb_TSA is better than st_TSA, ns_TSA, and TSA. It is seen in Tables 11 and 12. The convergence curve is better

Table 7
ns_TSA: Dynamic changes of the value of *ns* for 30 benchmark functions in D = 10.

Function	a/b			
	2/2	1/3	3/1	1/5
F_1	4.53E+05	3.77E+05	4.76E+05	3.69E+05
F_2	1167.49	1348.81	1442.26	980.32
F_3	3389.85	3730.56	3893.18	3962.28
F_4	404.70	403.04	404.97	402.29
F_5	520.20	520.18	520.06	520.23
F_6	600.02	600.00	600.11	600.00
F_7	700.33	700.34	700.35	700.26
F_8	810.86	811.17	811.31	810.93
F_9	925.89	924.52	926.36	926.18
F_{10}	1383.53	1361.23	1341.03	1398.61
F_{11}	2244.74	2215.40	2176.11	2237.92
F_{12}	1201.33	1201.26	1201.32	1201.29
F_{13}	1300.21	1300.20	1300.21	1300.21
F_{14}	1400.24	1400.25	1400.23	1400.22
F_{15}	1502.46	1502.59	1502.35	1502.48
F_{16}	1602.89	1602.89	1602.97	1602.88
F_{17}	6901.81	7405.80	6389.38	6703.02
F_{18}	3519.87	3210.12	2907.60	3484.34
F_{19}	1901.25	1901.46	1901.31	1901.24
F_{20}	2447.32	2438.07	2655.50	2493.25
F_{21}	2925.59	2844.71	2965.28	2727.24
F_{22}	2220.20	2221.32	2222.52	2219.10
F_{23}	2629.46	2629.46	2629.46	2629.46
F_{24}	2530.06	2529.01	2527.88	2527.68
F_{25}	2676.94	2680.38	2674.62	2672.52
F_{26}	2700.21	2700.21	2700.23	2700.21
F_{27}	2773.53	2792.80	2760.24	2770.00
F_{28}	3186.38	3196.32	3187.29	3194.02
F_{29}	3988.99	3925.19	3746.83	3848.65
F_{30}	4446.20	4488.96	4519.55	4500.36

Table 8
ns_TSA: Dynamic changes of the value of *ns* for 30 benchmark functions in D = 30.

Function	a/b			
	2/2	1/3	3/1	1/5
F_1	1.29E+08	1.25E+08	1.32E+08	1.25E+08
F_2	3.02E+06	1.57E+06	4.13E+06	1.25E+06
F_3	54845.99	55310.10	55723.87	51264.88
F_4	598.21	584.76	590.19	588.08
F_5	521.03	521.02	521.03	521.02
F_6	628.61	627.74	628.96	627.55
F_7	700.36	700.17	700.46	700.09
F_8	987.08	993.75	989.12	989.44
F_9	1130.89	1122.07	1129.25	1127.57
F_{10}	6839.34	6677.79	6663.64	6878.79
F_{11}	8362.33	8385.34	8355.41	8390.63
F_{12}	1202.98	1203.01	1202.91	1202.97
F_{13}	1300.55	1300.57	1300.56	1300.56
F_{14}	1400.34	1400.35	1400.35	1400.33
F_{15}	1522.77	1521.87	1522.30	1521.52
F_{16}	1612.83	1612.83	1612.90	1612.78
F_{17}	3.08E+06	3.00E+06	3.50E+06	3.25E+06
F_{18}	2407.15	2370.39	2657.56	2356.37
F_{19}	1908.80	1908.57	1909.00	1908.74
F_{20}	24833.39	26193.80	22964.87	22457.05
F_{21}	619146.60	617581.81	627699.21	453094.70
F_{22}	2777.40	2827.62	2819.96	2811.21
F_{23}	2615.75	2615.49	2615.83	2615.42
F_{24}	2630.15	2629.74	2630.51	2629.14
F_{25}	2726.79	2726.93	2728.12	2727.77
F_{26}	2700.76	2700.70	2704.45	2700.68
F_{27}	3317.09	3324.98	3316.00	3292.98
F_{28}	4106.59	4107.76	4100.24	4092.95
F_{29}	71539.07	79267.32	81491.19	83470.96
F_{30}	21713.52	19938.71	21734.66	20602.55

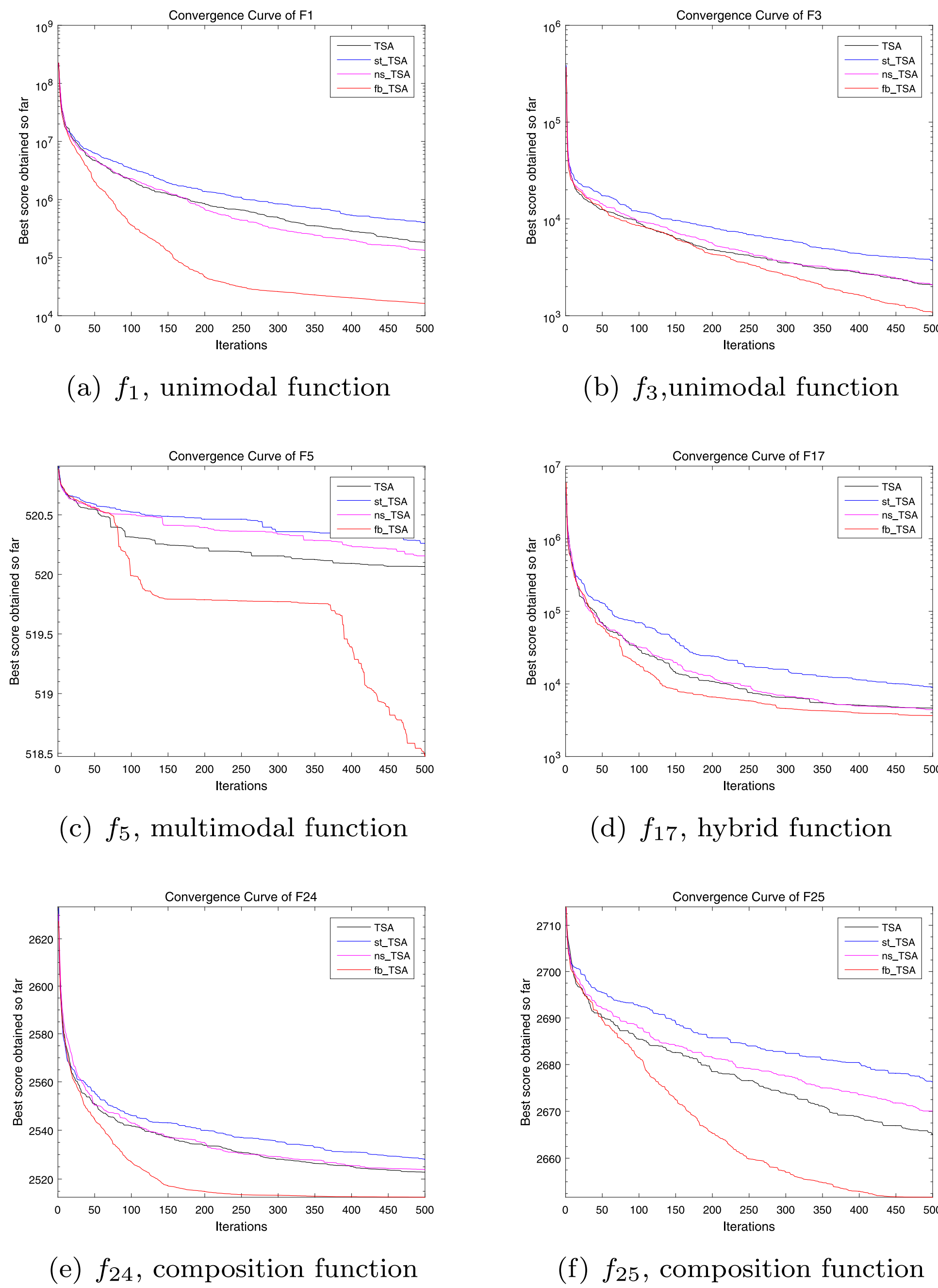
Table 9
ns_TSA: Dynamic changes of the value of *ns* for 30 benchmark functions in D = 50.

Function	a/b			
	2/2	1/3	3/1	1/5
F_1	4.25E+08	4.24E+08	4.31E+08	4.25E+08
F_2	1.45E+09	9.36E+08	1.83E+09	6.37E+08
F_3	1.53E+05	1.50E+05	1.52E+05	1.49E+05
F_4	1452.39	1295.75	1392.69	1233.90
F_5	521.20	521.21	521.21	521.20
F_6	658.16	657.93	659.05	658.62
F_7	711.57	707.49	713.87	705.12
F_8	1229.58	1231.09	1235.46	1226.39
F_9	1385.03	1382.00	1383.39	1387.94
F_{10}	13451.24	13296.01	13262.92	13285.88
F_{11}	14970.15	14918.83	14992.60	14958.87
F_{12}	1204.01	1203.97	1203.86	1203.91
F_{13}	1300.82	1300.82	1300.82	1300.78
F_{14}	1400.60	1400.49	1400.56	1400.53
F_{15}	4543.16	4209.01	4512.64	3326.49
F_{16}	1622.65	1622.69	1622.66	1622.64
F_{17}	2.36E+07	2.42E+07	2.53E+07	2.22E+07
F_{18}	2681.37	2767.31	2641.96	2745.88
F_{19}	1972.86	1969.45	1975.67	1974.08
F_{20}	62106.92	60657.03	62519.63	61252.65
F_{21}	8.37E+06	8.23E+06	9.26E+06	9.36E+06
F_{22}	4027.00	3978.57	4006.82	4127.05
F_{23}	2651.18	2649.53	2652.23	2648.50
F_{24}	2708.30	2706.79	2710.06	2705.20
F_{25}	2782.98	2781.64	2781.75	2779.01
F_{26}	2779.78	2777.31	2798.06	2777.65
F_{27}	4356.85	4348.34	4364.52	4361.49
F_{28}	6406.88	6503.46	6981.24	6461.94
F_{29}	2.67E+06	2.09E+06	2.56E+06	2.57E+06
F_{30}	2.45E+05	2.19E+05	2.36E+05	2.21E+05

Table 10
ns_TSA: Dynamic changes of the value of *ns* for 30 benchmark functions in D = 100.

Function	a/b			
	2/2	1/3	3/1	1/5
F_1	2.70E+09	2.42E+09	2.92E+09	2.70E+09
F_2	5.24E+10	6.76E+10	8.60E+10	cfv8.05E+10
F_3	332440.41	387565.90	353786.25	393337.03
F_4	7829.39	12248.54	11841.54	11681.93
F_5	521.36	521.36	521.37	521.37
F_6	740.21	748.19	747.33	744.73
F_7	1175.98	1349.55	1322.44	1397.61
F_8	1916.14	1946.06	1916.84	1928.23
F_9	2091.93	2138.19	2147.27	2135.88
F_{10}	30569.22	30767.82	30975.29	30827.49
F_{11}	32292.32	32361.47	32079.33	32685.85
F_{12}	1204.45	1204.99	1204.81	1204.53
F_{13}	1303.51	1304.30	1304.40	1304.17
F_{14}	1520.04	1582.62	1600.50	1579.50
F_{15}	8.21E+05	1.50E+06	1.23E+06	8.57E+05
F_{16}	1647.04	1647.06	1647.14	1646.82
F_{17}	2.23E+08	2.45E+08	2.94E+08	2.26E+08
F_{18}	3183.29	3146.85	38452.80	3199.31
F_{19}	2151.88	2286.38	2210.69	2167.21
F_{20}	2.44E+05	3.07E+05	2.41E+05	4.50E+05
F_{21}	6.51E+07	1.22E+08	1.04E+08	1.26E+08
F_{22}	6835.99	7186.81	6828.34	7018.42
F_{23}	2772.80	2819.86	2834.53	2812.32
F_{24}	3010.45	3024.71	3049.61	3030.09
F_{25}	3112.19	3066.19	3092.88	3093.21
F_{26}	2990.29	2972.90	2998.84	3018.60
F_{27}	6369.76	6358.10	6428.84	6465.73
F_{28}	19151.57	23152.07	23080.88	21043.86
F_{29}	1.92E+07	5.06E+07	4.89E+07	4.93E+07
F_{30}	2.15E+06	3.69E+06	4.47E+06	3.37E+06

than st_TSA, ns_TSA and TSA, they are presented in Figs. 5–8. The optimum of fb_TSA is better than TSA, STSA, GWO, ABC, SCA, DE, PSO and CLPSO, the results are shown in Tables 13–16. In the third experiment, compared fb_TSA with TSA, STSA, GWO, ABC, SCA, DE, PSO and CLPSO, the convergence curve are shown in Figs. 9, 10, 11 and 12.



(a) f_1, unimodal function (b) f_3,unimodal function

(c) f_5, multimodal function (d) f_{17}, hybrid function

(e) f_{24}, composition function (f) f_{25}, composition function

Fig. 5. Convergence curve of TSA, st_TSA, ns_TSA and fb_TSA, D=10.

4. Discussion

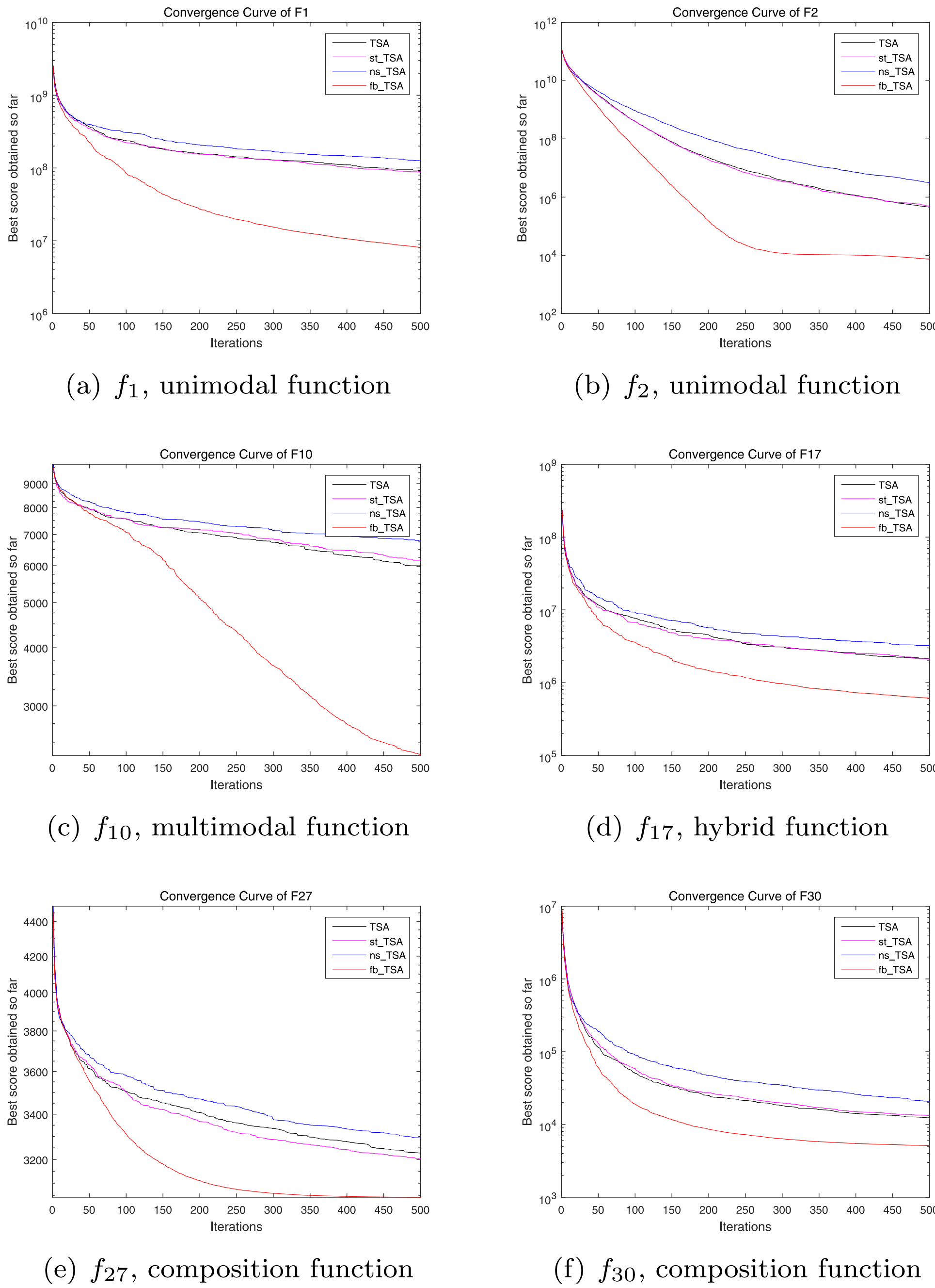
4.1. Discuss computational complexity

As for the TSA algorithm, its complexity is determined by the number of populations (N), the average number of seeds generated by each tree ($\overline{ns}$), and its maximal iterations (M), hence its complexity is O($N \times \overline{ns} \times M$). However, based on the analysis of the basic procedure of the proposed st_TSA, ns_TSA and fb_TSA, they do not increase any circles in their workflows when compared with TSA. Therefore, the computing complexity of proposed ns_TSA, st_TSA and fb_TSA is the same as the basic TSA with O($N \times \overline{ns} \times M$). Tables 17–20 provide the mean CPU execution time results (in seconds) between the TSA, STSA, GWO, ABC, SCA, DE, PSO and CLPSO on 10D, 30D, 50D and 100D benchmark test functions in CEC 2014. From Tables 17–20, compared with the TSA, STSA, GWO, ABC, SCA, DE, PSO and CLPSO algorithms, the proposed fb_TSA requires less CPU execution time (in seconds) than STSA.

4.2. Discuss the influence of ST and ns

The statistical results, numerical results, and convergence of the new hybrid method is tested, and the numerical solutions obtained are compared with the recent algorithms, such as TSA, STSA, GWO, ABC, SCA, DE, PSO and CLPSO. For all the experiments mentioned above, the following analysis for two hypotheses for this paper can be obtained:

Discussion [1] According to Tables 3–6. It can be seen that the *ST* plays a vital role in regulating balance. When the value of *ST* is high, the Eq. (2) is used to carry on powerful local search. By contraries, Eq. (3) is used to implement a powerful global search.



(a) f_1, unimodal function

(b) f_2, unimodal function

(c) f_{10}, multimodal function

(d) f_{17}, hybrid function

(e) f_{27}, composition function

(f) f_{30}, composition function

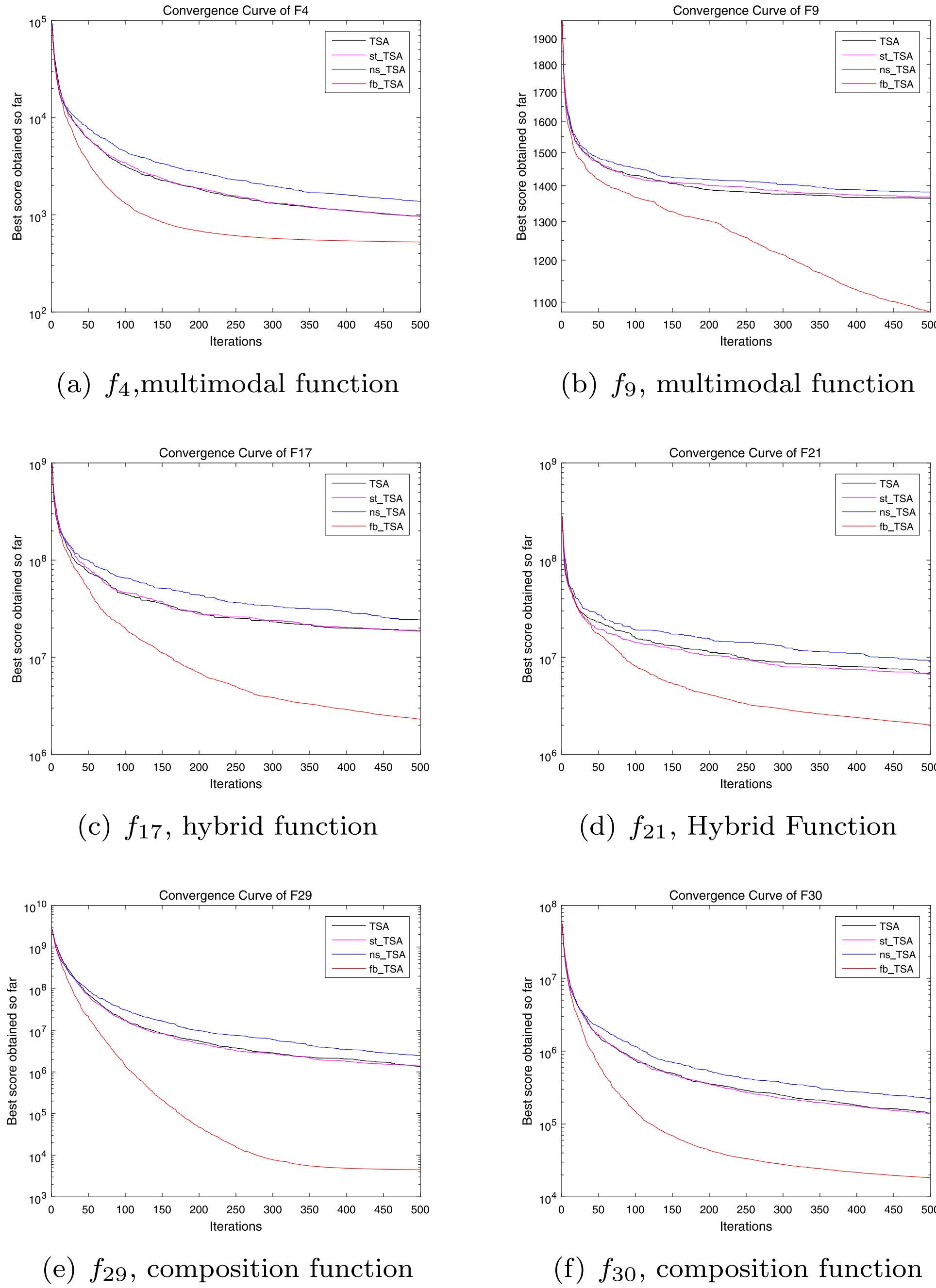
Fig. 6. Convergence curve of TSA, st_TSA, ns_TSA and fb_TSA, D=30.

The feedback mechanism is enhanced by updating the value of *ST* in Eqs. (8) and (9). As can be seen from Tables 3–6, continuous revisions of *ST* value and dynamic adjustments can achieve a good balance between exploration and exploitation.

Discussion [2] When the number of seeds is random, it is unreasonable. According to Eqs. (11) and (12), it can increase the opportunity to find local optimum. It fits with the common sense and can decrease the unnecessary search of seeds. However, when one over ten number of seeds is smaller than the number of better seeds, the optimum is better. There is a relatively good result and a balance between exploration and exploitation can be achieved. According to the number of seeds, it shows that the feed-back mechanism plays an important role. The number of seeds is updated and finally, achieves good condition. The mechanism adjusts the local and global optimum to update continuously.

Discussion [3] fb_TSA is tested by benchmark functions, the results present that fb_TSA has better optimum than TSA. According to the experimental results, we can find some advantages of fb_TSA: the optimum can get better. It shows that updating the *ST* value and adjusting the number of seeds can balance exploration and exploitation well. The feed-back mechanism can work by dynamic *ST* and *ns*, exploration and exploitation can be adjusted, the optimum is not ignored. fb_TSA is compared with other recent algorithms, such as TSA, STSA, GWO, ABC, SCA, DE, PSO and CLPSO. According to Tables 13–16, the st_TSA, ns_TSA, and fb_TSA have better results than TSA. And the fb_TSA has a better result than ABC, SCA, PSO and CLPSO.

fb_TSA is compared with TSA, STSA, GWO, ABC, SCA, DE, PSO and CLPSO and the statistic results are shown in Tables 21–24 from 4 different dimensions, the value of p is small. But in Tables 21 and 25, the p of GWO exceeds 0.1. Hence, there



(a) f_4,multimodal function (b) f_9, multimodal function

(c) f_{17}, hybrid function (d) f_{21}, Hybrid Function

(e) f_{29}, composition function (f) f_{30}, composition function

Fig. 7. Convergence curve of TSA, st_TSA, ns_TSA and fb_TSA, D=50.

is no difference between GWO and fb_TSA. For two hypothesis in the introduction, they are true through Wilcoxon's test in different dimensions. In addition, Friedman tests also prove the two hypothesis from Table 25. From all the above results, we can see that fb_TSA has TSA with more than half of the function optimal value is better than TSA, STSA, GWO, ABC, SCA, DE, PSO and CLPSO. For fb_TSA with 30, 50 and 100 dimensions, more than 20 test functions can get smaller optimal values. Specially, fb_TSA has worse than GWO for 10 dimension. In summary, we can see that the result of the proposed algorithm is prominent and the proposed algorithm is acceptable.

5. Application

5.1. Example 1: the tension/compression spring design problem

The Tension/Compression Spring Design problem (T/CSD) is a continuous constrained problem and this problem was described by Belegundu in 1982 [65]. The design variables are the mean coil diameter D, the diameter d and the number of active coils N, as shown in Fig. 13. The problem can be described as:

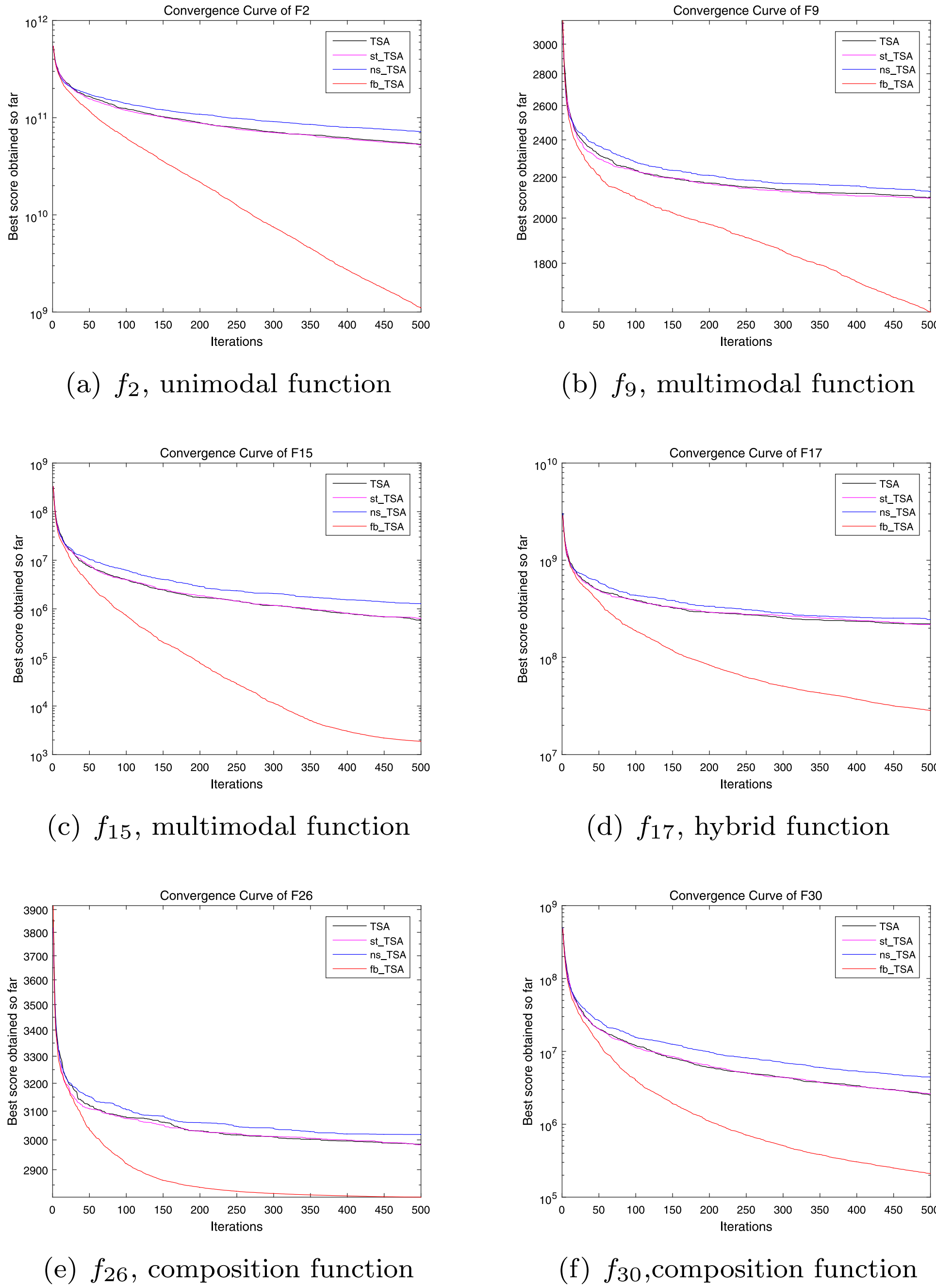
Consider : $x=[x_1, x_2, x_3]=[d,D,N]$

Minimize : $f(x) = (N+2)Dd^2$

Subject to :

$$g_1(x) = 1 - \frac{D^3N}{71785D^4} \leq 0$$

$$g_2(x) = \frac{4D^2 - dD}{12566(Dd^3 - d^4)} + \frac{1}{5108d^2} - 1 \leq 0$$



(a) f_2, unimodal function (b) f_9, multimodal function

(c) f_{15}, multimodal function (d) f_{17}, hybrid function

(e) f_{26}, composition function (f) f_{30},composition function

Fig. 8. Convergence curve of TSA, st_TSA, ns_TSA and fb_TSA, D=100.

$$g_3(x) = 1 - \frac{140.45d}{D^2N} \leq 0$$

$$g_4(x) = \frac{D+d}{1.5} - 1 \leq 0$$

The problem consists of the design variables is the following: $0.050000 \leq d \leq 2.000000$, $0.25000 \leq D \leq 1.300000$ and $2.000000 \leq N \leq 15.000000$. The solution shown for the technique proposed here is the best produced after 30 runs, the optimal results of the problem are presented in Table 26. The result for the tension/compression spring design problem, the fb_TSA is the best among these optimization algorithms.

5.2. Example 2: The pressure vessel design — PVD problem

Pressure vessel design pressure refers to the pressure used to determine the thickness of the pressure vessel shell at the corresponding design temperature. It is given by Kannan and Kramer [66] and can be describe :

Find:

x= $[x_1, x_2, x_3, x_4]$= $[T_s, T_h$, R, L];

to minimize: $f(x) = 0.6224x_1x_2x_3 + 1.7781x_2x_3^2 + 3.1611x_1^2x_4 + 19.84x_1^2 + x_3$

subject to : $g_1(x) = -x_1 + 0.0193x_3$

$$g_2(x) = -x_2 + 0.00954$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000$$

$$g_4(x) = x_4 - 240$$

where, the x_1 is the thickness of the shell(T_s), the x_2 presents the thickness of the head (T_h), x_3 is the inner radius (R) and the length of the cylindrical section without considering the head is described by $x_4(L)$, it is shown in Fig. 14.

Table 11
The minimal values of TSA, st_TSA, ns_TSA and fb_TSA for 30 benchmark functions.

Function		D=10				Function		D=30			
		fb_TSA	TSA	st_TSA	ns_TSA			fb_TSA	TSA	st_TSA	ns_TSA
Unimodal functions	F_1	**1.62E+04**	1.84E+05	1.33E+05	4.04E+05	Unimodal functions	F_1	**8.12E+06**	9.21E+07	8.81E+07	1.26E+08
	F_2	845.82	798.63	892.01	1585.97		F_2	**7469.75**	4.51E+05	4.96E+05	3.12E+06
	F_3	**1091.53**	2087.10	2095.41	3646.59		F_3	**1.64E+04**	3.90E+04	3.92E+04	5.36E+04
Simple multimodal functions	F_4	412.85	403.71	402.50	402.98	Simple multimodal functions	F_4	**491.88**	549.31	545.88	597.23
	F_5	**518.47**	520.07	520.16	520.26		F_5	521.02	520.99	521.00	521.01
	F_6	**600.0000**	600.0026	600.0003	600.0061		F_6	**602.77**	624.06	622.59	627.85
	F_7	**700.02**	700.24	700.22	700.32		F_7	**700.00**	700.04	700.02	700.31
	F_8	**802.92**	806.44	805.95	812.00		F_8	**835.64**	968.81	966.13	991.60
	F_9	**905.94**	921.11	920.12	926.16		F_9	**967.05**	1119.13	1118.87	1128.84
	F_{10}	**1056.23**	1172.12	1143.54	1379.15		F_{10}	**2348.76**	5982.69	6162.52	6781.05
	F_{11}	**1361.40**	2005.83	2048.98	2252.72		F_{11}	**7445.33**	8240.83	8186.64	8417.79
	F_{12}	1201.30	1201.09	1201.19	1201.28		F_{12}	1202.86	1202.75	1202.81	1202.90
	F_{13}	**1300.15**	1300.19	1300.18	1300.23		F_{13}	**1300.42**	1300.52	1300.50	1300.55
	F_{14}	1400.23	1400.21	1400.20	1400.24		F_{14}	1400.3343	1400.3255	1400.3341	1400.3316
	F_{15}	**1501.24**	1502.30	1502.19	1502.46		F_{15}	**1513.36**	1520.04	1520.04	1521.95
	F_{16}	**1602.07**	1602.70	1602.64	1602.92		F_{16}	**1612.49**	1612.73	1612.61	1612.88
Hybrid Functions	F_{17}	**3653.52**	4637.28	4415.29	8960.17	Hybrid functions	F_{17}	**6.11E+05**	2.11E+06	2.13E+06	3.22E+06
	F_{18}	4120.20	2765.73	3291.12	3284.26		F_{18}	**2110.16**	2315.58	2377.05	2333.83
	F_{19}	**1900.36**	1900.96	1901.03	1901.26		F_{19}	**1906.41**	1907.80	1907.91	1908.71
	F_{20}	2258.33	2274.95	2225.82	2317.25		F_{20}	**15152.24**	17545.10	17896.82	23007.79
	F_{21}	**2281.59**	2448.27	2478.25	2850.28		F_{21}	**149688.49**	377670.77	403700.91	658805.81
	F_{22}	**2208.58**	2209.40	2211.30	2222.07		F_{22}	**2410.90**	2652.08	2730.69	2784.62
Composition functions	F_{23}	**2629.46**	2629.46	2629.46	2629.46	Composition functions	F_{23}	**2615.24**	2615.27	2615.27	2615.69
	F_{24}	**2512.59**	2522.82	2523.70	2527.91		F_{24}	**2624.86**	2627.56	2627.48	2629.64
	F_{25}	**2651.72**	2664.99	2670.07	2676.30		F_{25}	**2709.02**	2723.03	2723.01	2727.51
	F_{26}	**2700.13**	2700.18	2700.18	2700.22		F_{26}	2703.72	2700.64	2700.76	2700.65
	F_{27}	2821.42	2782.43	2766.52	2772.97		F_{27}	**3042.33**	3227.23	3202.37	3293.18
	F_{28}	3215.77	3164.51	3193.25	3211.16		F_{28}	**3713.24**	4028.10	4015.56	4112.72
	F_{29}	**3339.13**	3695.82	3737.72	3946.06		F_{29}	**3951.30**	34427.90	27191.99	81119.41
	F_{30}	**3853.93**	4306.83	4334.18	4578.13		F_{30}	**5126.36**	12409.05	13326.32	20432.43

Table 12
The minimal values of TSA, st_TSA, ns_TSA and fb_TSA for 30 benchmark functions.

Function		D=50				Function		D=100			
		fb_TSA	TSA	st_TSA	ns_TSA			fb_TSA	TSA	st_TSA	ns_TSA
Unimodal functions	F_1	**1.86E+07**	3.28E+08	3.60E+08	4.38E+08	Unimodal functions	F_1	**2.07E+08**	2.34E+09	2.28E+09	2.71E+09
	F_2	**2.93E+04**	2.09E+08	2.14E+08	1.53E+09		F_2	**1.10E+09**	5.33E+10	5.24E+10	7.21E+10
	F_3	**1.15E+05**	1.27E+05	1.21E+05	1.52E+05		F_3	**3.09E+05**	3.44E+05	3.51E+05	3.86E+05
Simple multimodal functions	F_4	**525.54**	970.74	961.84	1361.77	Simple multimodal functions	F_4	**1033.26**	9006.34	8968.72	12274.59
	F_5	521.21	521.20	521.19	521.21		F_5	521.37	521.36	521.36	521.37
	F_6	**616.97**	655.00	654.12	659.46		F_6	**687.56**	743.48	742.57	746.97
	F_7	**700.10**	702.34	702.18	711.76		F_7	**710.67**	1161.24	1156.40	1336.84
	F_8	**898.56**	1202.28	1202.20	1238.81		F_8	**1185.48**	1886.86	1892.33	1947.27
	F_9	**1077.78**	1364.44	1367.69	1381.94		F_9	**1606.41**	2096.90	2094.57	2127.63
	F_{10}	**5399.50**	12759.60	12462.56	13238.15		F_{10}	**18285.00**	29862.58	29876.56	30442.81
	F_{11}	**14380.59**	14667.61	14705.81	14909.93		F_{11}	**31866.54**	32207.28	32083.01	32386.99
	F_{12}	1203.97	1203.79	1203.79	1203.93		F_{12}	1204.61	1204.40	1204.45	1204.56
	F_{13}	**1300.62**	1300.77	1300.75	1300.80		F_{13}	**1300.70**	1303.47	1303.47	1304.19
	F_{14}	1400.45	1400.43	1400.45	1400.55		F_{14}	**1400.52**	1538.69	1536.59	1586.14
	F_{15}	**1530.79**	1858.67	1849.04	4504.95		F_{15}	**1877.44**	5.81E+05	6.34E+05	1.28E+06
	F_{16}	**1622.39**	1622.56	1622.45	1622.69		F_{16}	1646.85	1646.92	1647.02	1647.04
Hybrid functions	F_{17}	**2.30E+06**	1.87E+07	1.85E+07	2.43E+07	Hybrid Function	F_{17}	**2.85E+07**	2.21E+08	2.18E+08	2.45E+08
	F_{18}	**2602.67**	2777.78	2798.35	2874.87		F_{18}	**3022.81**	3221.26	3117.11	14123.25
	F_{19}	**1940.33**	1967.94	1963.78	1972.69		F_{19}	**2018.65**	2153.73	2145.66	2235.49
	F_{20}	**38638.58**	43312.74	46112.05	63462.04		F_{20}	**1.92E+05**	2.45E+05	2.66E+05	3.65E+05
	F_{21}	**1.98E+06**	6.73E+06	6.83E+06	9.21E+06		F_{21}	**1.05E+07**	8.88E+07	9.19E+07	1.18E+08
	F_{22}	**3399.96**	3872.96	3885.01	4054.24		F_{22}	**6627.95**	7006.32	7002.74	7208.65
Composition functions	F_{23}	**2644.01**	2646.26	2646.03	2651.54	Composition functions	F_{23}	**2655.28**	2764.41	2762.80	2829.13
	F_{24}	**2672.45**	2695.27	2694.79	2709.03		F_{24}	**2816.78**	2990.70	2993.04	3034.82
	F_{25}	**2720.89**	2772.63	2773.33	2781.80		F_{25}	**2802.07**	3047.84	3047.48	3087.92
	F_{26}	2772.50	2765.93	2738.24	2766.86		F_{26}	**2810.72**	2985.41	2983.03	3018.36
	F_{27}	**3417.30**	4205.53	4202.94	4341.28		F_{27}	**4895.74**	6366.75	6395.92	6532.39
	F_{28}	**4377.07**	6205.37	5881.29	6607.40		F_{28}	**9620.58**	20832.84	20601.91	22392.43
	F_{29}	**4498.71**	1.36E+06	1.40E+06	2.49E+06		F_{29}	**3.38E+04**	2.34E+07	2.25E+07	6.31E+07
	F_{30}	18299.25	1.41E+05	1.37E+05	2.25E+05		F_{30}	**2.10E+05**	2.53E+06	2.64E+06	4.45E+06

The solution to design of a pressure vessel is the best produced after 30 runs, the optimal results of the problem is presented in Table 27. And through Table 27, we can see that the fb_TSA is the best.

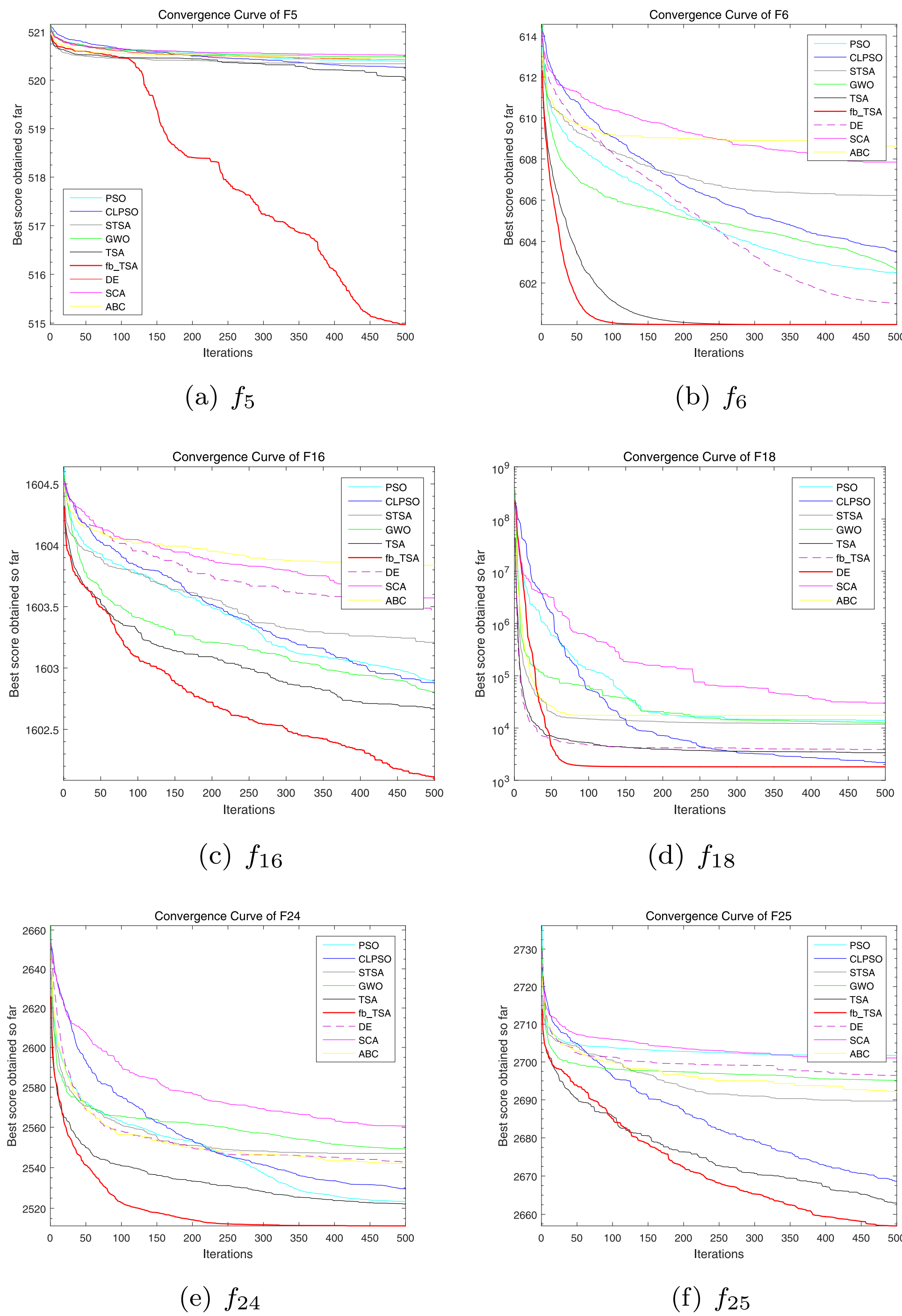


Fig. 9. Convergence curve of fb_TSA and other algorithms, D=10.

5.3. Example 3: Himmelblau's non-linear optimization problem

Himmelblau's non-linear optimization problem was proposed by Himmelblau [66]. After that many researchers solved this problem using different methods [67–69]. And it has been widely used as a benchmark non-linear constrained optimization problem [70]. The problem can be stated as follows:

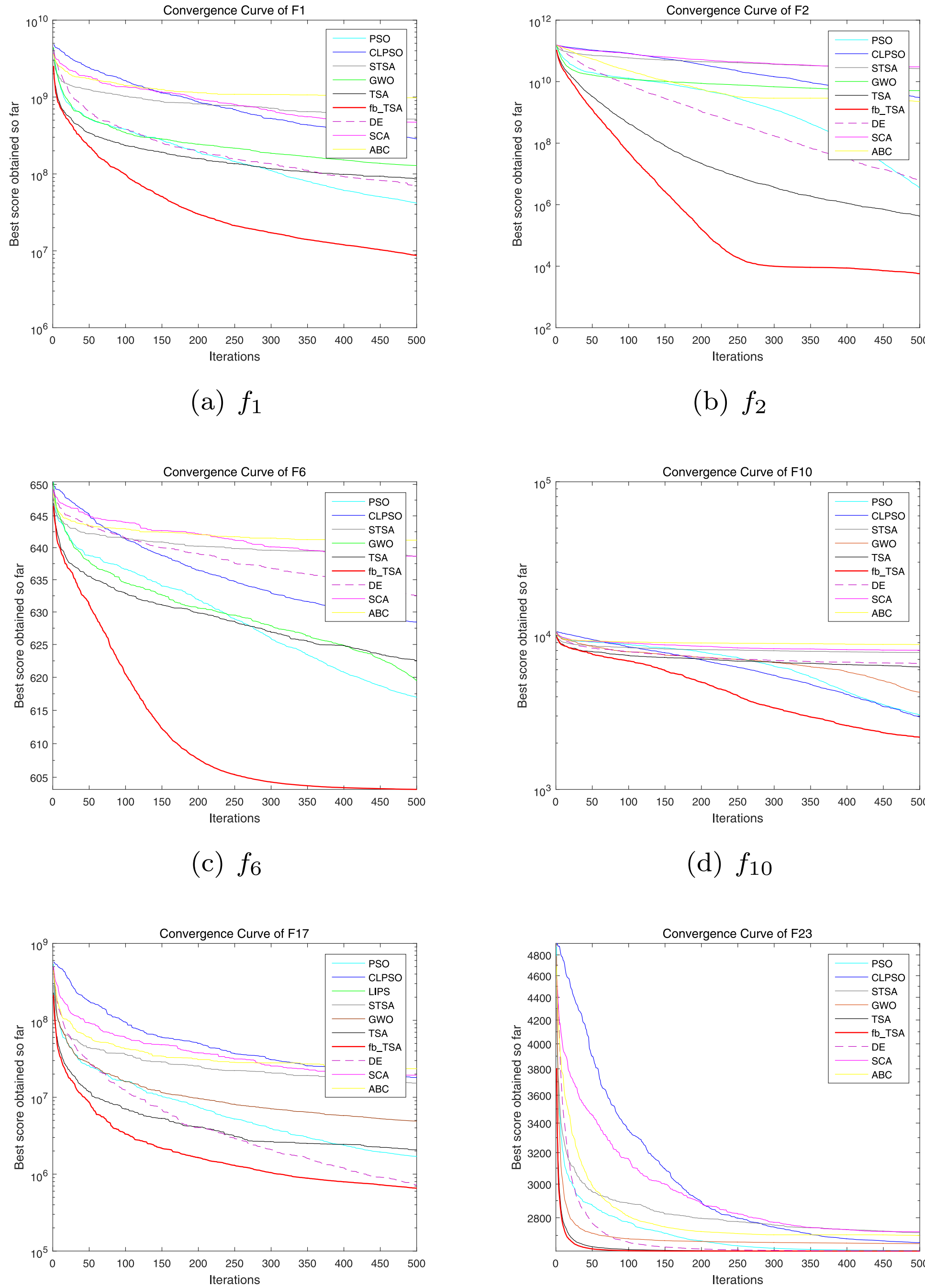
Minimize: $f(x) = 5.378547x_3^2 + 0.8356891x_1x_5 + 37.29329x_1 - 40792.141$

Subject to : $g_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.00026x_1x_4 - 0.0022053x_3x_5$

$g_2(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2$

$g_3(x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4$

In this problem, there are five design variables (x_1, x_2, x_3, x_4, x_5), six non-linear inequality constraints and 10 boundary conditions. The value range of variables is as follows: $0 \le g_1(x) \le 92$, $90 \le g_2(x) \le 110$, $20 \le g_3(x) \le 25$ $78 \le x_1 \le 102$, $33 \le x_2 \le 45$, $27 \le x_3 \le 45$, $27 \le x_4 \le 45$, $27 \le x_5 \le 45$ After 30 experiments, the optimal value can be obtained. It is presented in Table 28. Through the results, we can get the fb_TSA is better.



(a) f_1 (b) f_2 (c) f_6 (d) f_{10} (e) f_{17} (f) f_{23}

Fig. 10. Convergence curve of fb_TSA and other algorithms, D=30.

5.4. Example 4: Welded beam design problem

This problem is to minimize the manufacturing cost of welded beams [71] as shown in Fig. 15. The constraints for welded beams are as follows [66]:

- shear stress (τ)
- bending stress in the beam (σ)
- buckling load on the bar (p_c)
- end deflection of the beam (δ)
- side constraints

This problem has four variables such as thickness of weld ($h(x_1)$), length of attached part of bar ($l(x_2)$), the height of the bar ($t(x_3)$), and thickness of the bar ($b(x_4)$). The problem can be stated as follows:

Minimize:
$F(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$
Subject to :
$g_1(x) = \tau(x) - \tau_{max} \leq 0$
$g_2(x) = \sigma(x) - \sigma_{max} \leq 0$
$g_3(x) = x - 1 - x_4 \leq 0$
$g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0$
$g_5(x) = 0.125 - x_1 \leq 0$
$g_6(x) = \delta(x) - \delta_{max} \leq 0$
$g_7(x) = P - P_c(x) \leq 0$
where

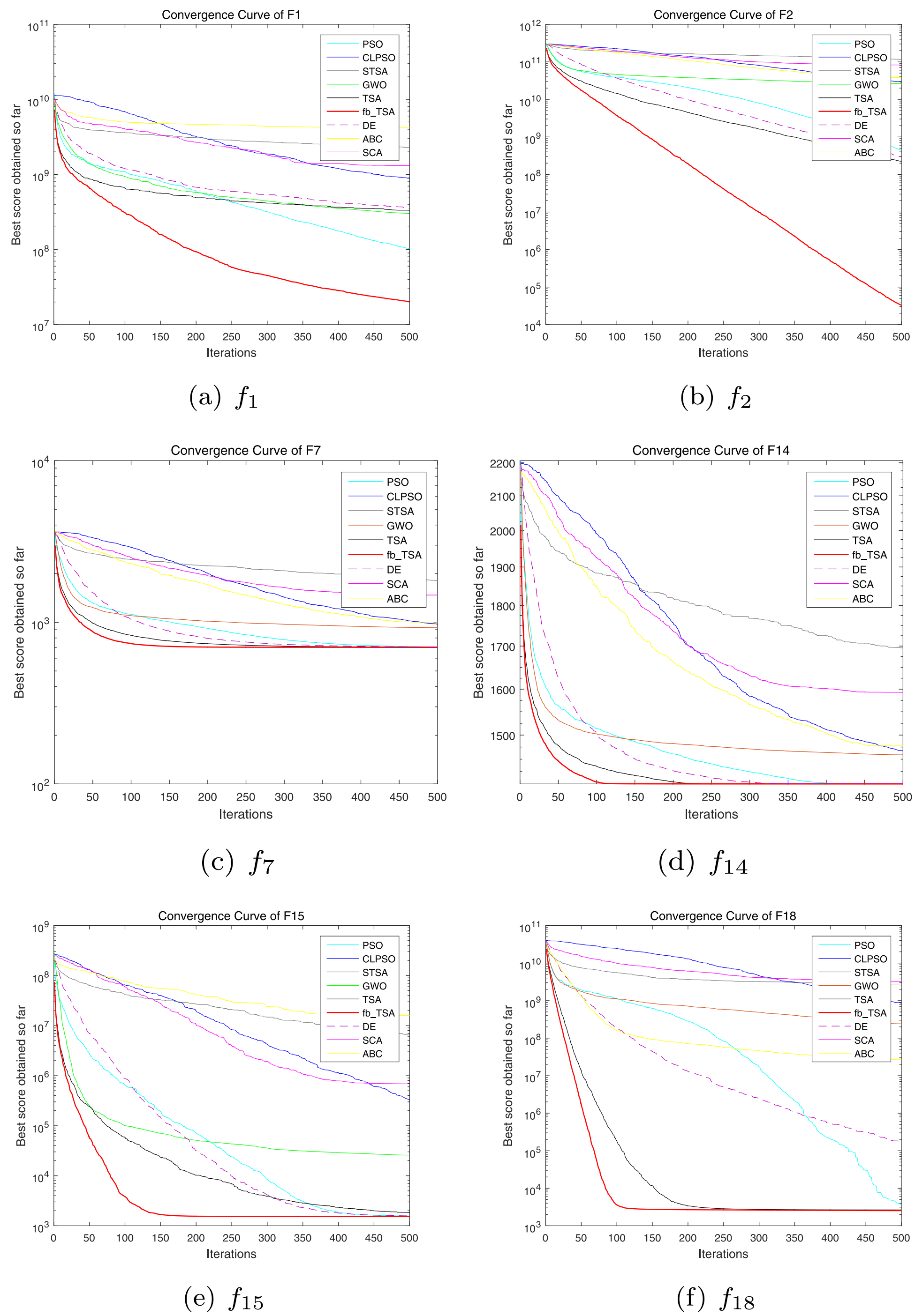


Fig. 11. Convergence curve of fb_TSA and other algorithms, D=50.

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$$
$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = p(L + \frac{X_2}{2})$$
$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2}$$
$$J = 2\sqrt{2}x_1x_2[\frac{x_2^2}{12} + (\frac{x_1 + x_3}{2})^2]$$
$$\sigma(x) = \frac{6PL}{x_4x_3^2}, \delta(x) = \frac{4PL^3}{EX_3^3x_4}$$

$$P_c(x) = \frac{4.013E\sqrt{\frac{X_3^2X_4^6}{36}}}{L^2}(1 - \frac{X_3}{2L}\sqrt{\frac{E}{4G}})$$
$P = 6000\text{lb}, L = 14\text{in}, E = 30 \times 10^6\text{psi}, G = 12 \times 10^6\text{psi}$
$\tau_{max} = 13600\text{psi}, \sigma_{max} = 3000\text{psi}, \delta_{max} = 0.25\text{in}$

After 30 experiments, the optimal value can be got in Table 29. Obviously the results of fb_TSA is the best.

Based on the these above four real engineering problems, the proposed fb_TSA is the best optimization and when compared with the basic TSA, ABC, SCA, PSO algorithms. The enhanced feedback mechanism has a better optimization performance when it is used to process some constrained continuous engineering optimization problems.

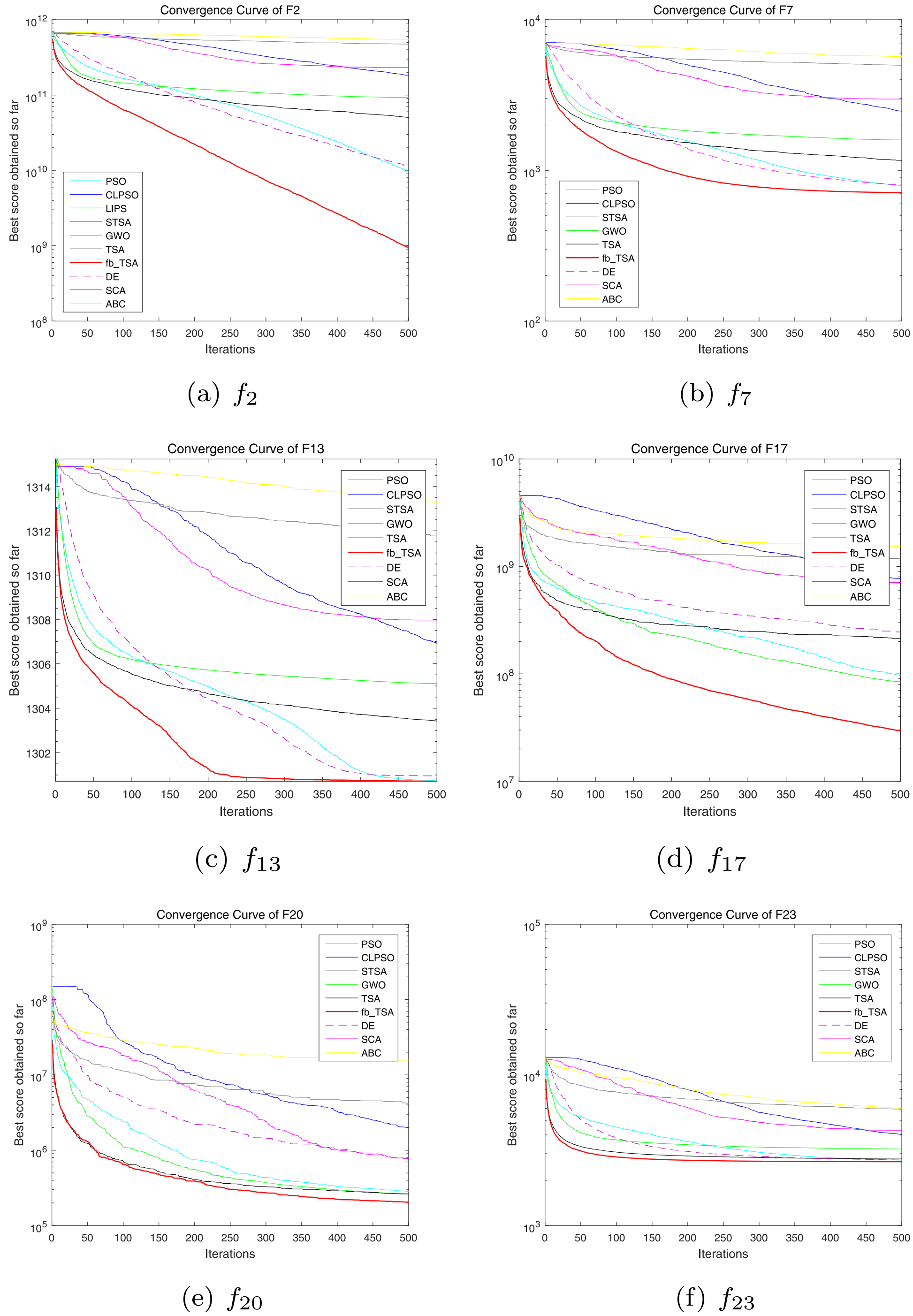


Fig. 12. Convergence curve of fb_TSA and other algorithms, D=100.

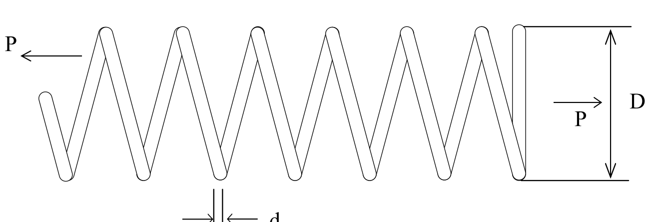


Fig. 13. The Tension/Compression Spring Design problem (T/CSD).

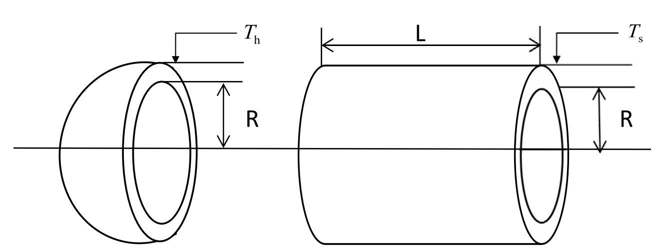


Fig. 14. Pressure vessel design — PVD.

Table 13
The minimal values of fb_TSA and other algorithms for 30 benchmark functions, D=10.

Function		fb_TSA		TSA		STSA		GWO		ABC		SCA		PSO		CLPSO		DE	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Unimodal functions	F_1	16100	1.70E+04	182000	1.33E+05	1.16E+07	2.93E+06	1.05E+07	9.81E+06	7.78E+06	3.91E+06	1.39E+07	6.02E+06	1.09E+05	1.09E+05	1.40E+06	1.04E+06	1.91E+03	2.13E+03
	F_2	855.52	682.05	927.97	748.65	6.11E+08	1.62E+08	4.67E+07	1.33E+08	4.18E+06	2.12E+06	1.03E+09	3.55E+08	5.13E+03	3.16E+03	1.05E+05	2.03E+03	2.00E+02	1.86E−01
	F_3	1185.33	847.27	1917.57	859.22	6.04E+03	3.07E+03	8.94E+03	4.77E+03	4.39E+04	1.77E+04	1.09E+04	6.90E+03	6.10E+03	3.79E+03	1.29E+03	6.76E+02	3.00E+02	6.54E−05
Simple multimodal functions	F_4	412.71	16.09	402.53	6.98	4.64E+02	1.37E+01	4.17E+02	1.58E+01	4.07E+02	4.50E−01	4.90E+02	3.67E+01	4.29E+02	7.95E+00	4.21E+02	1.08E+01	4.35E+02	1.22E+01
	F_5	**517.16**	7.35	519.65	2.21	5.20E+02	9.59E−02	5.20E+02	1.07E−01	5.21E+02	9.00E−02	5.20E+02	9.00E−02	5.20E+02	1.10E−01	5.20E+02	7.00E−02	5.20E+02	1.01E−01
	F_6	**600**	0.00	600	0.01	6.07E+02	1.05E+00	6.03E+02	1.22E+00	6.09E+02	8.00E−01	6.08E+02	1.24E+00	6.02E+02	1.55E+00	6.04E+02	9.70E−01	6.00E+02	1.32E+00
	F_7	**700.02**	0.01	700.23	0.13	7.15E+02	3.78E+00	7.03E+02	1.04E+01	7.01E+02	1.00E−01	7.15E+02	4.33E+00	7.00E+02	8.00E−02	7.00E+02	1.10E−01	7.01E+02	8.25E−02
	F_8	803.01	1.43	807.51	3.62	8.43E+02	5.51E+00	8.14E+02	8.32E+00	8.23E+02	4.08E+00	8.46E+02	1.01E+01	8.04E+02	2.24E+00	8.00E+02	4.00E−01	8.28E+02	4.29E+00
	F_9	**905.57**	2.24	921.44	5.76	9.46E+02	4.12E+00	9.16E+02	7.62E+00	9.48E+02	5.28E+00	9.48E+02	8.05E+00	9.12E+02	5.07E+00	9.16E+02	3.28E+00	9.41E+02	5.62E+00
	F_{10}	1058.54	66.66	1167.21	162.71	2.04E+03	1.62E+02	1.35E+03	2.01E+02	2.47E+03	1.33E+02	2.14E+03	1.85E+02	1.18E+03	1.32E+02	1.02E+03	6.55E+00	1.27E+03	1.27E+02
	F_{11}	**365.06**	250.88	2041.31	194.39	2.55E+03	1.54E+02	1.73E+03	4.09E+02	2.37E+03	1.73E+02	2.60E+03	2.20E+02	1.66E+03	3.16E+02	1.80E+03	1.27E+02	2.72E+03	2.62E+02
	F_{12}	1201.26	0.31	1201.18	0.25	1.20E+03	2.38E−01	1.20E+03	6.58E−01	1.20E+03	2.50E−01	1.20E+03	3.40E−01	1.20E+03	4.00E−01	1.20E+03	1.20E−01	1.20E+03	4.07E−01
	F_{13}	**1300.14**	0.03	1300.18	0.04	1.30E+03	1.20E−01	1.30E+03	6.77E−02	1.30E+03	6.00E−02	1.30E+03	1.80E−01	1.30E+03	7.00E−02	1.30E+03	5.00E−02	1.30E+03	5.54E−02
	F_{14}	1400.23	0.05	1400.2	0.05	1.40E+03	3.96E−01	1.40E+03	1.84E−01	1.40E+03	5.00E−02	1.40E+03	7.30E−01	1.40E+03	0.07	1.40E+03	6.00E−02	1.40E+03	4.80E−02
	F_{15}	**1501.32**	0.4	1502.18	0.46	1.51E+03	1.08E+00	1.50E+03	1.01E+00	1.50E+03	4.80E−01	1.51E+03	3.80E+00	1.50E+03	6.90E−01	1.50E+03	5.20E−01	1.50E+03	4.76E−01
	F_{16}	**1602.13**	0.36	1602.68	0.20	1.60E+03	1.41E−01	1.60E+03	5.27E−01	1.60E+03	2.50E−01	1.60E+03	2.50E−01	1.60E+03	4.70E−01	1.60E+03	3.00E−01	1.60E+03	1.59E−01
Hybrid functions	F_{17}	3555.68	1729.88	4772.66	1779.64	2.81E+04	1.50E+04	9.92E+04	1.79E+05	1.06E+04	6.89E+04	6.05E+04	9.59E+04	6.73E+03	3.18E+03	8.31E+04	8.87E+04	1.90E+03	8.12E+01
	F_{18}	3859.58	1963	3094.25	1832.44	1.27E+04	2.30E+03	1.01E+04	7.01E+03	3.15E+04	9.58E+03	5.74E+04	9.28E+04	1.02E+04	1.06E+04	2.06E+03	3.10E+02	1.81E+03	2.13E+00
	F_{19}	**1900.47**	0.39	1900.97	0.27	1.91E+03	6.35E−01	1.90E+03	1.03E+00	1.90E+03	2.70E−01	1.91E+03	1.08E+00	1.90E+03	9.50E−01	1.90E+03	2.60E−01	1.90E+03	4.52E−01
	F_{20}	2283.95	283.05	2217.24	150.34	5.58E+03	2.11E+03	1.11E+04	7.14E+03	9.93E+03	4.84E+03	8.93E+03	5.63E+03	4.43E+03	4.11E+03	2.11E+03	1.43E+02	2.00E+03	1.29E+00
	F_{21}	2304.5	119.49	2450.18	135.95	7.96E+03	2.61E+03	1.88E+04	3.68E+04	3.24E+04	1.30E+04	1.48E+04	9.26E+03	3.14E+03	1.66E+03	1.00E+04	7.29E+03	2.10E+03	2.92E+01
	F_{22}	**2209.58**	8.86	2209.81	9.07	2.26E+03	7.70E+00	2.30E+03	7.21E+01	2.25E+03	8.92E+00	2.28E+03	2.00E+01	2.22E+03	1.48E+01	2.22E+03	2.22E+01	2.22E+03	9.46E+00
	F_{23}	2629.46	0	2629.46	0.00	2.64E+03	1.95E+01	2.64E+03	5.82E+00	2.63E+03	1.95E+00	2.65E+03	6.72E+00	2.63E+03	0.00E+00	2.62E+03	28.59	2.63E+03	2.95E−10
Composition functions	F_{24}	**2511.7**	4.46	2523.49	7.47	2.55E+03	6.33E+00	2.56E+03	3.67E+01	2.55E+03	5.72E+00	2.56E+03	7.34E+00	2.52E+03	5.83E+00	2.53E+03	5.54E+00	2.54E+03	5.97E+00
	F_{25}	**2655.41**	30.87	2666.44	15.78	2.69E+03	9.47E+00	2.70E+03	9.52E+00	2.70E+03	6.85E+00	2.70E+03	5.78E+00	2.69E+03	2.66E+01	2.67E+03	1.58E+01	2.70E+03	1.06E+01
	F_{26}	**2700.13**	0.04	2700.18	0.04	2.70E+03	1.40E−01	2.70E+03	9.39E−02	2.70E+03	6.00E−02	2.70E+03	1.70E−01	2.70E+03	6.00E−02	2.70E+03	7.00E−02	2.70E+03	4.93E−02
	F_{27}	2794.36	143.08	2785.27	126.41	2.91E+03	1.91E+02	3.06E+03	1.04E+02	2.89E+03	8.84E+01	3.00E+03	1.82E+02	3.03E+03	9.72E+01	2.85E+03	1.80E+02	2.70E+03	1.58E+02
	F_{28}	3200.21	46.19	3179.72	60.93	3.27E+03	2.38E+01	3.14E+03	4.75E+01	3.11E+03	4.60E−01	3.31E+03	8.28E+01	3.31E+03	8.27E+01	3.26E+03	4.64E+01	3.18E+03	2.32E+01
	F_{29}	3329.45	112.38	3799.53	404.91	1.26E+04	7.75E+03	3.11E+03	9.51E+00	3.11E+03	2.20E+00	1.95E+04	2.27E+04	4.15E+05	1.25E+05	3.36E+03	9.75E+01	3.12E+03	2.24E+03
	F_{30}	3914.1	232.30	4322.01	301.99	4.70E+03	6.32E+02	3.38E+03	1.70E+02	3.36E+03	4.47E+01	5.37E+03	1.46E+03	3.98E+03	3.98E+03	3.97E+03	1.89E+02	3.49E+03	2.42E+01
Average ranking		**2.23**		3.31		7.03		5.7		6.4		8.33		4.16		4.06		3.8	
Total ranking		**1**		2		8		6		9		5		4		3		7	

Table 14
The minimal values of fb_TSA and other algorithms for 30 benchmark functions, D=30.

Function		fb_TSA		TSA		STSA		GWO		ABC		SCA		PSO		CLPSO		DE	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Unimodal functions	F_1	**6.990E+06**	4.220E+06	9.450E+07	2.030E+07	8.732E+08	1.932E+08	2.866E+09	1.940E−06	1.380E+09	1.240E+10	5.290E+08	2.360E+08	3.150E+07	5.190E+11	2.360E+08	9.240E+10	4.807E+07	2.219E+07
	F_2	**6.565E+03**	4.030E+03	4.520E+05	3.760E+05	4.123E+10	3.941E+09	1.028E+11	0.000E+00	2.910E+09	1.870E+12	2.890E+10	4.940E+09	3.380E+06	2.530E+06	3.160E+09	8.380E+08	4.493E+06	2.890E+06
	F_3	1.602E+04	1.173E+04	4.157E+04	9.123E+03	1.354E+05	2.009E+04	3.555E+07	2.273E−08	4.610E+05	8.296E+04	7.070E+04	1.752E+04	2.345E+04	1.080E+04	3.266E+04	9.686E+03	4.490E+02	1.314E+02
Simple multimodal functions	F_4	**4.936E+02**	3.184E+01	5.418E+02	2.961E+01	6.003E+03	1.011E+03	2.583E+04	7.400E−12	7.150E+02	7.790E+04	2.830E+03	9.577E+02	6.545E+02	5.946E+01	1.245E+03	2.280E+02	5.793E+02	2.112E+01
	F_5	5.210E+02	7.000E−02	5.210E+02	5.000E−02	5.210E+02	6.431E−02	5.217E+02	3.469E−13	5.211E+02	8.000E−02	5.211E+02	5.000E−02	5.210E+02	1.000E−01	5.207E+02	6.000E−02	5.210E+02	7.090E−02
	F_6	**6.039E+02**	2.950E+00	6.231E+02	4.420E+00	6.399E+02	1.259E+00	6.521E+02	1.156E−13	6.399E+02	1.070E+00	6.376E+02	2.380E+00	6.169E+02	4.170E+00	6.280E+02	1.820E+00	6.350E+02	6.440E+00
	F_7	**7.000E+02**	0.000E+00	7.000E+02	4.000E−02	1.078E+03	4.369E+01	1.771E+03	1.156E−12	7.045E+02	1.960E+00	9.580E+02	4.604E+01	7.010E+02	1.400E−01	7.275E+02	5.740E+00	7.009E+02	5.610E−02
	F_8	**8.349E+02**	8.930E+00	9.672E+02	1.966E+01	1.133E+03	2.689E+01	1.331E+03	6.938E−13	1.056E+03	1.581E+01	1.084E+03	2.431E+01	8.524E+02	1.444E+01	8.802E+02	8.710E+00	1.001E+03	1.362E+01
	F_9	**9.662E+02**	4.694E+01	1.118E+03	1.300E+01	1.277E+03	2.400E+01	1.380E+03	2.313E−13	1.198E+03	1.342E+01	1.226E+03	2.186E+01	1.022E+03	4.083E+01	1.114E+03	1.849E+01	1.120E+03	1.474E+01
	F_{10}	**2.390E+03**	9.219E+02	5.971E+03	8.877E+02	8.006E+03	2.663E+02	1.178E+04	7.400E−12	9.279E+03	2.535E+02	7.890E+03	3.833E+02	3.154E+03	6.750E+02	2.879E+03	3.515E+02	6.203E+03	2.658E+02
	F_{11}	**7.560E+03**	9.023E+02	8.253E+03	2.616E+02	8.710E+03	2.681E+02	1.390E+04	5.550E−12	8.931E+03	2.990E+02	8.919E+03	3.671E+02	6.401E+03	1.447E+03	6.179E+03	3.616E+02	8.664E+03	3.761E+02
	F_{12}	1.203E+03	3.200E−01	1.203E+03	4.000E−01	1.203E+03	2.596E−01	1.208E+03	2.313E−13	1.204E+03	4.400E−01	1.203E+03	4.000E−01	1.202E+03	6.800E−01	1.201E+03	1.900E−01	1.203E+03	4.455E−01
	F_{13}	**1.300E+03**	7.000E−02	1.301E+03	6.000E−02	1.305E+03	4.460E−01	1.311E+03	6.938E−13	1.301E+03	1.000E−01	1.304E+03	3.900E−01	1.301E+03	1.400E−01	1.301E+03	1.700E−01	1.301E+03	7.718E−02
	F_{14}	1.400E+03	4.000E−02	1.400E+03	4.000E−02	1.525E+03	1.837E+01	1.810E+03	6.938E−13	1.401E+03	2.100E−01	1.485E+03	1.315E+01	1.400E+03	8.000E−02	1.414E+03	3.550E+00	1.400E+03	1.207E−01
	F_{15}	**1.514E+03**	3.960E+00	1.520E+03	1.350E+00	2.981E+05	1.881E+05	1.052E+06	0.000E+00	9.834E+03	1.181E+04	3.912E+04	3.616E+04	1.521E+03	3.960E+00	5.272E+03	2.606E+03	1.523E+03	1.700E+00
	F_{16}	1.613E+03	2.800E−01	1.613E+03	2.600E−01	1.613E+03	1.347E−01	1.616E+03	4.625E−13	1.614E+03	1.300E−01	1.613E+03	2.700E−01	1.613E+03	3.200E−01	1.612E+03	2.600E−01	1.613E+03	2.186E−01
Hybrid functions	F_{17}	6.180E+05	3.320E+05	1.900E+06	7.380E+05	2.575E+07	8.365E+06	9.796E+08	2.425E−07	2.160E+07	7.780E+06	1.780E+07	7.560E+06	2.490E+06	1.350E+06	2.170E+07	1.360E+07	3.723E+05	3.877E+05
	F_{18}	**2.258E+03**	5.958E+02	2.378E+03	6.047E+02	2.444E+08	1.247E+08	1.545E+10	9.700E−06	1.120E+06	4.010E+08	4.010E+08	2.180E+08	8.540E+05	3.190E+06	3.280E+07	3.150E+07	1.250E+04	6.889E+03
	F_{19}	**1.906E+03**	6.400E−01	1.908E+03	6.200E−01	2.027E+03	1.686E+01	2.805E+03	9.250E−13	1.922E+03	2.041E+03	2.041E+03	2.940E+01	1.917E+03	1.870E+01	2.004E+03	2.826E+01	1.908E+03	1.296E+00
	F_{20}	1.485E+04	7.258E+03	1.901E+04	6.653E+03	1.334E+05	5.895E+04	3.199E+09	0.000E+00	5.124E+05	6.665E+04	6.665E+04	3.230E+04	2.701E+04	1.249E+04	2.760E+04	1.127E+04	2.150E+03	4.915E+02
	F_{21}	1.850E+05	1.490E+05	3.730E+05	1.620E+05	6.913E+06	2.958E+06	2.759E+09	0.000E+00	1.220E+07	5.850E+06	5.850E+06	3.980E+06	4.350E+05	3.920E+05	6.230E+06	3.550E+06	1.922E+04	1.124E+04
	F_{22}	**2.442E+03**	1.478E+02	2.670E+03	1.193E+02	3.430E+03	1.376E+02	5.839E+06	9.472E−10	3.268E+03	3.429E+03	3.249E+03	1.364E+02	2.606E+03	2.250E+01	2.953E+03	2.006E+02	2.721E+03	1.790E+02
	F_{23}	2.615E+03	0.000E+00	2.615E+03	2.000E−02	2.775E+03	2.729E+01	2.500E+03	0.000E+00	2.694E+03	2.733E+03	2.733E+03	3.830E+01	2.618E+03	1.180E+00	2.660E+03	1.611E+01	2.616E+03	1.678E−01
Composition functions	F_{24}	2.625E+03	1.350E+00	2.628E+03	7.500E−01	2.692E+03	1.546E+01	2.600E+03	0.000E+00	2.655E+03	2.627E+03	2.627E+03	1.876E+01	2.642E+03	4.480E+00	2.663E+03	4.230E+00	2.635E+03	3.668E+00
	F_{25}	2.710E+03	3.420E+00	2.723E+03	2.870E+00	2.775E+03	1.214E+01	2.700E+03	0.000E+00	2.771E+03	2.748E+03	2.748E+03	1.261E+01	2.714E+03	3.240E+00	2.728E+03	5.020E+00	2.715E+03	4.720E+00
	F_{26}	2.717E+03	3.784E+01	2.701E+03	8.900E−01	2.705E+03	4.367E−01	2.800E+03	0.000E+00	2.701E+03	2.704E+03	2.704E+03	5.800E−01	2.711E+03	3.037E+01	2.704E+03	1.610E+00	2.701E+03	4.287E+01
	F_{27}	3.056E+03	5.617E+01	3.225E+03	9.589E+01	3.974E+03	1.556E+02	2.900E+03	0.000E+00	4.048E+03	3.947E+03	3.947E+03	2.209E+02	3.336E+03	1.675E+02	3.307E+03	5.152E+01	3.464E+03	1.415E+02
	F_{28}	3.716E+03	4.725E+01	4.035E+03	5.132E+01	5.317E+03	2.593E+02	3.000E+03	0.000E+00	3.574E+03	5.867E+03	5.867E+03	6.527E+02	4.313E+03	5.507E+02	5.440E+03	4.302E+02	3.753E+03	9.354E+01
	F_{29}	3.917E+03	1.734E+02	3.373E+04	2.382E+04	3.012E+07	9.853E+06	3.100E+03	0.000E+00	3.469E+03	3.880E+07	3.880E+07	1.270E+07	3.640E+06	9.280E+06	7.630E+06	6.289E+06	1.252E+04	3.518E+06
	F_{30}	5.058E+03	7.405E+02	1.375E+04	2.158E+03	6.118E+05	1.600E+05	3.200E+03	0.000E+00	5.525E+03	6.276E+05	6.286E+05	2.278E+05	1.721E+04	8.848E+03	2.468E+05	9.111E+04	6.779E+03	5.050E+03
Average ranking		**2.03**		3.23		7.43		7.13		6.67		6.17		3.63		4.87		3.83	
Total ranking		**1**		2		9		8		6		7		3		5		4	

Table 15
The minimal values of fb_TSA and other algorithms for 30 benchmark functions, D=50.

Function		fb_TSA		TSA		STSA		GWO		ABC		SCA		PSO		CLPSO		DE	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
	F_1	**1.770E+07**	7.950E+06	3.320E+08	5.610E+08	2.700E+09	5.606E+08	1.665E+10	0.000E+00	2.820E+09	8.200E+08	1.420E+09	3.830E+08	1.020E+08	5.020E+07	8.200E+08	1.980E+08	4.113E+08	8.926E+07
Unimodal functions	F_2	**2.870E+04**	1.830E+04	2.260E+08	5.800E+08	1.453E+11	1.553E+10	1.996E+11	3.104E−05	2.570E+10	1.050E+10	8.710E+10	1.140E+10	3.790E+08	1.820E+08	2.990E+10	5.950E+09	2.563E+08	6.860E+07
	F_3	1.157E+05	1.473E+04	1.234E+05	1.284E+04	3.433E+05	3.700E+04	6.963E+08	3.637E−07	6.634E+05	2.077E+05	1.580E+05	2.754E+04	1.585E+05	3.059E+04	1.390E+05	2.387E+04	4.927E+04	1.369E+04
	F_4	**5.303E+02**	3.085E+01	9.732E+02	7.165E+01	4.002E+04	8.432E+03	7.299E+04	1.480E−11	1.017E+04	4.178E+03	1.558E+04	3.043E+03	8.281E+02	8.172E+01	7.268E+03	1.431E+03	5.976E+02	4.892E+01
	F_5	5.212E+02	5.000E−02	5.212E+02	4.000E−02	5.212E+02	3.422E−02	5.217E+02	3.469E−13	5.212E+02	4.000E−02	5.213E+02	4.000E−02	5.212E+02	7.000E−02	5.209E+02	5.000E−02	5.212E+02	4.242E−02
	F_6	**6.151E+02**	5.950E+00	6.535E+02	3.500E+00	6.753E+02	1.586E+00	6.907E+02	0.000E+00	6.780E+02	1.870E+00	6.713E+02	2.950E+00	6.398E+02	4.670E+00	6.601E+02	1.680E+00	6.715E+02	7.569E+00
	F_7	**7.001E+02**	6.000E−02	7.022E+02	3.300E−01	2.087E+03	1.377E+02	2.579E+03	2.313E−12	1.116E+03	5.976E+01	1.487E+03	9.107E+01	7.046E+02	1.260E+00	9.668E+02	6.153E+01	7.047E+02	8.968E−01
	F_8	8.921E+02	2.131E+01	1.167E+02	3.360E+01	1.524E+03	3.895E+01	1.709E+03	0.000E+00	1.344E+03	3.472E+01	1.380E+03	3.060E+01	9.422E+02	3.397E+01	1.079E+03	2.611E+01	1.228E+03	2.268E+01
Simple multimodal functions	F_9	9.107E+04	9.354E+01	1.365E+03	1.726E+01	1.744E+03	3.882E+01	1.911E+03	4.625E−13	1.570E+03	3.705E+01	1.543E+03	4.335E+01	1.214E+03	7.671E+01	1.445E+03	4.042E+01	1.335E+03	2.799E+01
	F_{10}	**5.364E+03**	1.636E+03	1.274E+04	9.180E+02	1.490E+04	3.798E+02	1.943E+04	0.000E+00	1.576E+04	4.490E+02	1.470E+04	4.679E+02	6.404E+03	1.088E+03	7.596E+03	6.023E+02	1.349E+04	5.950E+02
	F_{11}	1.461E+04	5.133E+02	1.470E+04	4.128E+02	1.537E+04	5.265E+02	1.943E+04	0.000E+00	1.646E+04	4.470E+02	1.565E+04	4.586E+02	1.415E+04	1.539E+03	1.163E+04	4.377E+02	1.566E+04	4.623E+02
	F_{12}	1.204E+03	2.800E−01	1.204E+03	3.300E−01	1.204E+03	4.599E−01	1.214E+03	4.625E−13	1.204E+03	3.500E−01	1.204E+03	4.200E−01	1.203E+03	7.500E−01	1.202E+03	1.800E−01	1.204E+03	3.456E−01
	F_{13}	1.301E+03	1.000E−01	1.301E+03	5.000E−02	1.308E+03	4.434E−01	1.310E+03	2.313E−13	1.303E+03	3.200E−01	1.306E+03	3.800E−01	1.301E+03	1.000E−01	1.304E+03	3.400E−01	1.301E+03	1.278E−01
	F_{14}	**1.400E+03**	4.000E−02	1.400E+03	8.000E−02	1.773E+03	4.128E+01	1.880E+03	6.938E−13	1.489E+03	1.569E+01	1.593E+03	2.547E+01	1.401E+03	2.900E−01	1.473E+03	1.331E+01	1.400E+03	3.743E−01
	F_{15}	1.533E+03	5.330E+00	1.848E+03	2.108E+02	1.471E+07	6.387E+06	2.740E+07	0.000E+00	1.980E+07	1.210E+07	6.907E+05	5.735E+05	1.572E+03	2.445E+01	3.586E+05	1.999E+05	1.636E+03	6.335E+01
	F_{16}	1.622E+03	0.000E+00	1.623E+03	1.600E−01	1.623E+03	1.767E−01	1.625E+03	9.250E−13	1.624E+03	0.000E+00	1.623E+03	1.700E−01	1.622E+03	4.100E−01	1.622E+03	3.200E−01	1.623E+03	1.869E−01
	F_{17}	**2.620E+06**	1.820E+06	1.740E+07	5.680E+06	2.219E+08	4.326E+07	3.878E+09	1.940E−06	1.470E+08	7.040E+07	1.170E+08	5.320E+07	1.040E+07	6.330E+06	1.650E+08	6.950E+07	1.557E+07	6.883E+06
	F_{18}	**2.561E+03**	7.025E+02	2.677E+03	6.078E+02	3.282E+09	8.024E+08	3.821E+10	7.760E−06	2.460E+07	1.310E+07	3.170E+09	8.660E+08	2.825E+04	1.350E+06	9.420E+08	4.560E+08	7.019E+04	2.678E+05
	F_{19}	1.941E+03	2.076E+01	1.967E+03	1.863E+01	2.504E+03	1.051E+02	1.083E+04	0.000E+00	1.938E+03	1.460E+00	2.317E+03	6.700E+01	1.981E+03	1.689E+01	2.205E+03	8.367E+01	1.934E+03	1.090E+01
Hybrid functions	F_{20}	3.864E+04	1.159E+04	3.936E+04	1.158E+04	6.537E+05	3.485E+05	3.218E+09	0.000E+00	7.053E+05	9.363E+05	1.507E+05	1.006E+05	7.091E+04	2.522E+04	1.268E+05	9.194E+04	1.963E+04	1.771E+04
	F_{21}	**2.700E+06**	1.990E+06	7.540E+06	1.950E+06	5.835E+07	1.603E+07	1.867E+09	7.275E−07	9.620E+07	2.090E+07	3.360E+07	1.490E+07	6.010E+06	3.530E+06	3.70+E07	1.800E+07	2.837E+06	1.906E+06
	F_{22}	**3.420E+03**	4.103E+02	3.891E+03	1.693E+02	5.427E+03	2.878E+02	6.111E+06	3.789E−09	5.464E+03	1.901E+02	5.341E+03	3.112E+02	3.679E+03	4.403E+02	5.185E+03	5.327E+02	3.919E+03	1.620E+02
	F_{23}	2.644E+03	2.400E−03	2.646E+03	5.500E−01	3.564E+03	1.243E+02	2.500E+03	0.000E+00	2.762E+03	2.782E+01	3.311E+03	1.133E+02	2.657E+03	4.090E+00	2.964E+03	5.816E+01	2.645E+03	8.211E−01
	F_{24}	2.672E+03	3.120E+00	2.695E+03	3.100E+00	2.977E+03	3.979E+01	2.600E+03	0.000E+00	2.729E+03	1.963E+01	2.788E+03	1.997E+01	2.698E+03	7.250E+00	2.782E+03	1.202E+01	2.688E+03	4.438E+00
	F_{25}	2.721E+03	4.160E+00	2.770E+03	8.640E+00	2.948E+03	4.262E+01	2.700E+03	0.000E+00	3.023E+03	4.985E+01	2.816E+03	3.843E+01	2.735E+03	7.890E+00	2.803E+03	1.561E+01	2.768E+03	2.037E+01
	F_{26}	2.711E+03	4.703E+01	2.753E+03	5.894E+01	2.709E+03	4.742E−01	2.800E+03	0.000E+00	2.712E+03	1.790E+00	2.720E+03	7.158E+01	2.778E+03	7.999E+01	2.740E+03	5.345E+01	2.701E+03	1.435E+02
Composition functions	F_{27}	3.414E+03	1.681E+02	4.204E+03	1.083E+01	5.031E+03	5.287E+01	2.900E+03	1.388E−12	4.904E+03	4.301E+01	4.949E+03	9.324E+01	4.030E+03	1.269E+02	4.722E+03	6.621E+01	4.377E+03	1.800E+02
	F_{28}	4.369E+03	1.915E+02	5.923E+03	5.880E+02	9.677E+03	5.299E+02	3.000E+03	1.388E−12	8.034E+03	9.187E+02	1.067E+04	9.839E+02	6.529E+03	1.176E+03	1.049E+04	9.291E+02	4.266E+03	1.292E+02
	F_{29}	4.465E+03	4.111E+02	1.170E+06	6.480E+05	3.008E+08	5.079E+07	3.100E+03	0.000E+00	1.006E+04	2.102E+03	3.540E+08	6.850E+07	2.270E+07	5.790E+07	1.940E+08	5.360E+07	2.523E+04	1.947E+07
	F_{30}	1.837E+04	2.612E+03	1.349E+05	4.263E+04	4.871E+06	1.312E+06	3.200E+03	0.000E+00	8.422E+03	8.216E+02	4.982E+06	1.413E+06	9.644E+04	2.402E+04	3.081E+06	1.350E+06	2.253E+04	1.003E+04
Average ranking		**2.3**		3.6		7.2		7.1		6.2		6.7		3.4		5		5	
Total ranking		**1**		4		9		8		6		7		2		5		3	

Table 16
The minimal values of fb_TSA and other algorithms for 30 benchmark functions, D=100.

Function		fb_TSA		TSA		STSA		GWO		ABC		SCA		PSO		CLPSO		DE	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Unimodal functions	F_1	**2.110E+08**	7.150E+07	2.250E+09	2.700E+08	1.291E+10	1.486E+09	1.793E+10	0.000E+00	5.110E+09	8.650E+08	1.850E+10	2.090E+09	5.670E+08	1.970E+08	4.720E+09	1.060E+09	2.860E+09	5.965E+08
	F_2	**1.120E+09**	3.870E+08	5.170E+10	5.520E+09	4.881E+11	3.435E+10	3.416E+11	0.000E+00	2.330E+11	1.390E+10	6.090E+11	4.240E+10	1.070E+10	2.250E+09	1.730E+11	1.740E+10	1.468E+10	3.198E+09
	F_3	**3.196E+05**	3.413E+04	3.573E+05	2.410E+04	8.621E+05	5.037E+04	2.401E+08	1.819E−07	3.933E+05	4.411E+04	8.670E+05	3.571E+05	4.458E+05	6.034E+04	4.625E+05	4.423E+04	4.210E+05	4.952E+04
Simple multimodal functions	F_4	**1.028E+03**	7.984E+01	8.556E+03	9.872E+02	1.704E+05	1.697E+04	1.420E+05	0.000E+00	5.611E+04	9.746E+03	1.639E+05	4.316E+04	2.327E+03	5.511E+02	4.183E+04	8.228E+03	1.892E+03	4.122E+02
	F_5	5.214E+02	3.000E−02	5.214E+02	3.000E−02	5.214E+02	2.811E−02	5.217E+02	2.313E−13	5.214E+02	2.000E−02	5.214E+02	3.000E−02	5.214E+02	6.000E−02	5.211E+02	5.212E+02	5.214E+02	3.718E−02
	F_6	**6.864E+02**	1.299E+01	7.429E+02	3.120E+00	7.656E+02	2.530E+00	7.814E+02	3.469E−13	7.594E+02	5.190E+00	7.463E+02	2.680E+00	7.102E+02	7.380E+00	7.446E+02	4.050E+00	7.599E+02	4.884E+00
	F_7	**7.099E+02**	2.860E+00	1.151E+03	3.547E+01	5.135E+03	2.567E+02	4.167E+03	0.000E+00	2.975E+03	1.651E+02	5.890E+03	5.190E+02	7.899E+02	2.064E+01	2.431E+03	1.913E+02	8.324E+02	2.122E+01
	F_8	**1.198E+03**	8.725E+01	1.879E+03	4.709E+01	2.619E+03	7.130E+01	2.567E+03	1.388E−12	2.172E+03	4.954E+01	2.817E+03	8.357E+01	1.338E+03	7.119E+01	1.870E+03	5.716E+01	1.814E+03	3.088E+01
	F_9	**1.571E+03**	2.090E+02	2.089E+03	3.972E+01	3.081E+03	7.391E+01	2.601E+03	4.625E−13	2.392E+03	5.446E+01	3.259E+03	1.053E+02	1.939E+03	1.134E+02	2.417E+03	6.363E+01	1.991E+03	3.682E+01
	F_{10}	**1.785E+04**	3.810E+02	2.944E+04	1.176E+03	3.323E+04	5.014E+02	3.761E+04	2.960E−11	3.167E+04	1.056E+03	3.426E+04	5.593E+02	1.800E+04	1.967E+03	2.267E+04	7.740E+02	2.875E+04	7.386E+02
	F_{11}	3.174E+04	1.146E+03	3.216E+04	4.572E+02	3.346E+04	4.791E+02	3.775E+04	2.220E−11	3.364E+04	6.072E+02	3.498E+04	5.220E+02	3.112E+04	1.814E+03	2.716E+04	7.562E+02	3.347E+04	7.269E+02
	F_{12}	1.205E+03	3.600E−01	1.205E+03	2.700E−01	1.205E+03	2.803E−01	1.214E+03	0.000E+00	1.205E+03	2.700E−01	1.205E+03	2.800E−01	1.204E+03	4.600E−01	1.202E+03	1.800E−01	1.205E+03	2.811E−01
	F_{13}	**1.301E+03**	1.000E−01	1.303E+03	2.400E−01	1.312E+03	4.810E−01	1.310E+03	0.000E+00	1.308E+03	3.100E−01	1.314E+03	7.400E−01	1.301E+03	1.200E−01	1.307E+03	4.200E−01	1.301E+03	9.988E−02
	F_{14}	**1.401E+03**	1.100E−01	1.538E+03	1.296E+01	2.690E+03	8.883E+01	2.442E+03	9.250E−13	2.042E+03	5.070E+01	2.928E+03	1.005E+02	1.427E+03	7.790E+00	1.915E+03	4.904E+01	1.422E+03	1.010E+01
	F_{15}	**1.810E+03**	2.180E+02	6.000E+05	1.950E+05	2.691E+08	7.361E+07	6.099E+07	2.273E−08	9.080E+06	3.840E+06	4.350E+08	1.460E+08	1.780E+04	1.280E+04	1.520E+07	7.460E+06	1.696E+04	3.242E+04
	F_{16}	1.647E+03	2.000E−01	1.647E+03	2.100E−01	1.648E+03	1.900E−01	1.650E+03	1.156E−12	1.648E+03	3.000E−01	1.648E+03	2.000E−01	1.647E+03	4.300E−01	1.646E+03	4.900E−01	1.648E+03	2.391E−01
Hybrid functions	F_{17}	**2.310E+07**	1.240E+07	2.250E+08	4.340E+07	1.445E+09	2.001E+08	3.590E+09	0.000E+00	6.710E+08	1.960E+08	1.510E+09	4.240E+08	8.890E+07	3.350E+07	7.840E+08	2.310E+08	2.138E+08	7.895E+07
	F_{18}	**2.790E+03**	6.520E+02	2.910E+03	2.540E+02	2.724E+10	3.717E+09	5.828E+10	2.328E−05	1.400E+10	3.240E+09	1.200E+09	2.730E+08	1.300E+07	3.110E+07	1.290E+10	3.310E+09	1.724E+06	1.462E+06
	F_{19}	**2.016E+03**	2.945E+01	2.147E+03	2.195E+01	6.991E+03	1.227E+03	1.788E+04	1.110E−11	4.508E+03	4.730E+02	3.630E+03	3.777E+02	2.146E+03	3.491E+01	4.725E+03	6.465E+02	2.067E+03	1.467E+01
	F_{20}	**2.130E+05**	6.650E+04	2.570E+05	5.500E+04	5.973E+06	2.816E+06	1.313E+09	2.425E−07	8.700E+05	3.890E+05	1.130E+07	9.770E+06	3.150E+07	1.080E+05	1.780E+06	8.650E+05	4.556E+05	3.917E+05
	F_{21}	**8.160E+06**	3.640E+06	9.190E+07	1.960E+07	6.617E+08	1.511E+08	1.409E+09	2.425E−07	2.420E+08	8.170E+07	7.370E+08	1.730E+08	3.220E+07	1.880E+07	3.470E+08	1.230E+08	4.070E+07	3.264E+07
	F_{22}	6.740E+03	3.968E+02	7.119E+03	1.983E+02	1.268E+04	2.845E+03	2.169E+06	1.894E−09	9.306E+03	1.083E+03	1.167E+04	1.542E+03	6.139E+03	8.803E+02	1.564E+04	5.806E+03	7.705E+03	2.715E+02
	F_{23}	2.656E+03	2.610E+00	2.766E+03	1.299E+01	6.116E+03	4.126E+02	2.500E+03	0.000E+00	4.252E+03	2.320E+02	5.179E+03	4.339E+02	2.719E+03	2.146E+01	4.046E+03	2.534E+02	2.740E+03	1.970E+01
Composition functions	F_{24}	2.814E+03	9.620E+00	2.994E+03	1.502E+01	3.972E+03	1.094E+02	2.600E+03	0.000E+00	3.179E+03	4.536E+01	4.345E+03	1.218E+02	2.919E+03	1.857E+01	3.240E+03	3.818E+01	2.907E+03	1.393E+01
	F_{25}	2.807E+03	1.766E+01	3.053E+03	2.169E+01	3.890E+03	1.477E+02	2.700E+03	0.000E+00	3.086E+03	9.200E+01	4.262E+03	1.687E+02	2.866E+03	2.317E+01	3.175E+03	5.539E+01	3.169E+03	7.637E+01
	F_{26}	2.811E+03	4.090E+00	2.992E+03	2.547E+01	2.755E+03	1.190E+02	2.800E+03	0.000E+00	3.062E+03	2.439E+02	2.744E+03	9.810E+00	2.854E+03	5.659E+01	3.130E+03	5.349E+01	2.701E+03	2.252E+02
	F_{27}	4.942E+03	2.361E+02	6.388E+03	1.026E+02	7.502E+03	9.737E+01	2.900E+03	2.313E−12	7.394E+03	1.045E+02	7.357E+03	6.332E+01	5.812E+03	2.323E+02	7.142E+03	1.130E+02	6.767E+03	1.351E+02
	F_{28}	1.066E+04	2.043E+03	2.073E+04	1.352E+03	2.374E+04	1.462E+03	3.000E+03	2.313E−12	2.578E+04	1.911E+03	2.350E+04	2.474E+03	1.424E+04	1.725E+03	2.644E+04	1.567E+03	6.336E+03	1.021E+03
	F_{29}	3.240E+04	2.780E+04	2.480E+07	6.610E+06	1.596E+09	1.414E+08	3.100E+03	0.000E+00	1.950E+09	2.600E+08	7.010E+04	2.750E+04	3.790E+06	6.940E+06	1.690E+09	3.220E+08	1.233E+07	4.517E+07
	F_{30}	2.210E+05	7.690E+04	2.630E+06	6.240E+05	7.094E+07	1.732E+07	3.200E+03	0.000E+00	5.560E+07	1.940E+07	6.610E+04	2.960E+04	9.630E+05	5.300E+05	5.610E+07	2.480E+07	1.319E+06	4.478E+05
Average ranking		**1.8**		4.03		7.33		6.4		6.06		7.13		2.97		5.43		3.93	
Total ranking		**1**		4		9		7		6		8		2		5		3	

Table 17
The time (in seconds) for 30 benchmark functions in D = 10.

	fb_TSA	TSA	STSA	GWO	SCA	ABC	DE	PSO	CLPSO
F_1	0.0453	0.212	2.0439	0.18	0.083	0.057	0.011	0.001	0.017
F_2	0.122	0.101	1.9143	0.177	0.125	0.128	0.009	0.075	0.028
F_3	0.129	0.173	2.3447	0.18	0.223	0.198	0.015	0.155	0.039
F_4	0.265	0.243	2.3776	0.18	0.295	0.270	0.009	0.226	0.048
F_5	0.338	0.318	2.2793	0.17	0.372	0.346	0.010	0.298	0.060
F_6	0.796	0.728	4.5092	0.58	1.164	1.115	0.087	0.385	0.334
F_7	1.210	1.189	2.1056	0.20	1.245	1.219	0.010	1.167	0.349
F_8	1.287	1.266	2.6322	0.19	1.318	1.293	0.006	1.248	0.360
F_9	1.36	1.340	2.6968	0.18	1.392	1.367	0.010	1.321	0.374
F_{10}	1.448	1.425	2.4370	0.19	1.491	1.466	0.009	1.395	0.395
F_{11}	1.549	1.527	2.4328	0.19	1.594	1.568	0.012	1.495	0.417
F_{12}	1.697	1.672	3.2277	0.26	1.788	1.759	0.025	1.599	0.476
F_{13}	1.830	1.81	2.8816	0.17	1.861	1.837	0.010	1.792	0.486
F_{14}	1.903	1.883	3.4255	0.17	1.935	1.910	0.009	1.863	0.498
F_{15}	1.977	1.957	2.7061	0.18	2.01	1.985	0.010	1.937	0.512
F_{16}	2.055	2.034	3.6422	0.18	2.0903	2.0653	0.010	2.0126	0.526
F_{17}	2.139	2.118	3.2146	0.20	2.18	2.152	0.011	2.09	0.543
F_{18}	2.226	2.205	2.6405	0.19	2.263	2.236	0.010	2.182	5.557
F_{19}	2.381	0.251	3.6594	0.27	2.48	2.45	0.024	2.267	0.627
F_{20}	2.529	2.509	3.6343	0.18	2.563	2.541	0.012	2.486	0.641
F_{21}	2.616	2.595	2.1241	0.19	2.657	2.630	0.011	2.569	0.659
F_{22}	2.714	2.692	3.7018	0.2	2.757	2.731	0.012	2.661	0.681
F_{23}	2.887	2.855	3.5648	0.26	2.990	2.96	0.032	2.762	0.758
F_{24}	3.092	3.065	4.6831	0.23	3.177	3.148	0.022	2.994	0.814
F_{25}	3.292	3.262	2.7127	0.427	3.385	3.355	0.034	3.182	0.874
F_{26}	3.909	3.845	7.0639	0.67	4.371	4.314	0.120	3.402	1.217
F_{27}	4.855	4.791	6.8732	0.68	5.317	0.122	5.261	4.387	1.55
F_{28}	5.450	5.419	4.0686	0.28	5.575	5.542	0.037	5.322	1.623
F_{29}	5.730	5.696	4.3060	0.32	5.886	5.85	0.040	5.582	1.709
F_{30}	5.779	5.968	4.2759	0.25	6.094	6.063	0.026	5.891	1.763

Table 18
The time (in seconds) for 30 benchmark functions in D = 30.

	fb_TSA	TSA	STSA	GWO	SCA	ABC	DE	PSO	CLPSO
F_1	0.0803	0.055	6.9072	0.323	0.118	0.146	0.0324	0.003	0.038
F_2	0.215	0.184	5.8745	0.357	0.241	0.272	0.0314	0.15	0.171
F_3	0.335	0.311	6.3700	0.35	0.354	0.381	0.0276	0.276	0.297
F_4	0.441	0.416	6.1904	0.307	0.458	0.484	0.0271	0.384	0.403
F_5	0.551	0.527	6.7096	0.367	0.575	0.60	0.0300	0.488	0.512
F_6	1.701	1.555	5.8673	1.762	2.662	2.757	0.2568	0.635	1.3706
F_7	2.834	2.806	5.98	0.3839	0.324	2.891	0.0307	2.761	2.792
F_8	2.947	2.923	6.0104	0.276	2.961	2.987	0.0192	2.893	2.91
F_9	3.063	3.037	5.9477	0.293	3.087	3.114	0.0313	2.99	3.02
F_{10}	3.206	3.181	6.2083	0.349	3.254	3.284	0.0283	3.118	3.161
F_{11}	3.399	3.37	6.3992	0.394	3.464	3.5	0.0370	3.29	3.347
F_{12}	3.762	3.718	7.4727	0.669	3.946	3.984	0.0688	3.507	3.67
F_{13}	4.051	4.026	5.8627	0.307	4.0226	4.094	0.0346	3.988	4.011
F_{14}	4.154	4.131	5.9190	0.299	4.172	4.201	0.0279	4.098	4.118
F_{15}	4.276	4.251	5.8262	0.305	4.303	4.332	0.0401	4.204	4.236
F_{16}	4.41	4.383	5.9694	0.311	4.443	4.47	0.0326	4.336	4.366
F_{17}	4.556	4.530	6.0263	0.332	4.603	4.641	0.0338	4.474	4.512
F_{18}	4.711	4.685	6.2250	0.306	4.735	4.762	0.0286	4.646	4.671
F_{19}	5.054	5.003	9.3341	0.563	5.267	5.31	0.0709	4.772	4.951
F_{20}	5.38	5.357	6.2154	0.317	5.406	5.433	0.0299	5.316	5.342
F_{21}	5.512	5.487	6.6253	0.326	5.548	5.575	0.0330	5.436	5.47
F_{22}	5.673	5.646	6.7705	0.356	5.722	5.751	0.0513	5.58	5.626
F_{23}	6.06	6.00	7.8047	0.547	6.276	6.317	0.0939	5.762	1.949
F_{24}	6.551	6.51	7.1754	0.46	6.71	6.74	0.0648	6.326	6.471
F_{25}	7.015	6.963	8.4691	0.539	7.212	7.253	0.0796	6.759	6.916
F_{26}	8.663	8.487	22.9369	2.056	9.885	9.99	0.3445	7.297	8.256
F_{27}	11.381	11.2	18.4765	1.864	12.578	12.692	0.3407	10.049	10.97
F_{28}	13.056	12.981	8.8313	0.592	13.33	13.383	0.1140	12.706	12.919
F_{29}	13.79	13.718	9.2659	0.732	14.142	14.196	0.1204	13.4	13.644
F_{30}	14.432	14.385	7.7002	0.483	14.612	14.652	0.0762	14.207	14.341

6. Conclusion and future works

TSA is a novel continuous optimization approach with an excellent performance. In the basic TSA, the feed-back mechanism is its basic and essential mechanism to achieve the evolution process by trees and seeds. However, there are two problems with TSA. For instance, *ST* is fixed with a constant in TSA, and the number of seeds (*ns*) is generated without any heuristics. As hypothesized in this paper, these two factors should be redefined and redesigned in the feedback mechanism in the TSA.

To achieve a better global optimal solution with an enhanced TSA, three novel algorithms of st_TSA, ns_TSA, and fb_TSA are proposed to enhance its optimization capability. The redefined *ST*

Table 19
The time (in seconds) for 30 benchmark functions in D = 50.

	fb_TSA	TSA	STSA	GWO	SCA	ABC	DE	PSO	CLPSO
F_1	0.131	0.0973	11.4898	0.489	0.205	0.237	0.0620	0.006	0.0683
F_2	0.336	0.304	10.1065	0.503	0.379	0.408	0.0485	0.243	0.283
F_3	0.504	0.472	9.6419	0.444	0.546	0.574	0.0507	0.414	0.449
F_4	0.672	0.641	9.6344	0.467	0.714	0.743	0.0489	0.58	0.619
F_5	0.852	0.823	9.8118	0.439	0.911	0.940	0.0530	0.749	0.797
F_6	2.754	2.535	25.1522	3.251	4.378	4.531	0.4357	0.995	2.205
F_7	4.6556	4.621	9.9557	0.55	4.720	4.753	0.0549	4.537	4.594
F_8	4.841	4.815	9.5494	0.447	4.863	4.89	0.0364	4.758	4.792
F_9	4.998	4.968	9.8105	0.503	5.052	5.081	0.0551	4.895	4.943
F_{10}	5.225	5.183	10.4284	0.503	5.307	5.338	0.0503	5.087	5.153
F_{11}	5.513	5.471	11.1059	0.525	5.627	5.662	0.0646	5.346	5.434
F_{12}	6.028	5.974	12.7706	0.849	6.343	6.0404	0.1179	5.675	5.901
F_{13}	6.505	6.472	10.3258	0.425	6.549	6.576	0.0555	6.41	6.449
F_{14}	6.672	6.643	12.3563	0.429	6.714	6.743	0.0505	6.581	6.621
F_{15}	6.865	6.833	12.0144	0.44	6.931	6.962	0.0552	6.749	6.80
F_{16}	7.085	7.053	12.8986	0.448	7.149	7.179	0.0534	6.968	7.024
F_{17}	7.319	7.285	13.3899	0.474	7.403	7.435	0.0587	7.186	7.254
F_{18}	7.55	7.516	11.3199	0.442	7.6	7.63	0.0565	7.441	7.488
F_{19}	8.113	8.036	15.6015	0.84	8.509	8.572	0.1220	7.646	7.943
F_{20}	8.678	8.649	11.5144	0.442	8.733	8.764	0.0538	8.576	8.623
F_{21}	8.895	8.86	12.2169	0.472	8.972	9.008	0.0557	8.769	8.831
F_{22}	9.181	9.131	11.2937	0.517	9.281	9.315	0.0663	9.015	9.097
F_{23}	9.857	9.774	15.7780	0.874	10.27	10.33	0.1563	9.335	9.684
F_{24}	10.72	10.654	16.4257	0.682	11.017	11.066	0.1159	10.345	10.587
F_{25}	11.514	11.44	15.7896	0.776	11.873	11.924	0.1356	11.083	11.359
F_{26}	14.264	13.99	33.2329	3.457	16.332	16.513	0.5870	11.998	13.594
F_{27}	18.71	18.427	41.7209	3.312	20.793	20.977	0.5666	16.59	18.042
F_{28}	21.607	21.515	20.2447	1.173	22.140	22.206	0.1901	20.999	21.396
F_{29}	22.916	22.809	17.5021	1.215	23.544	23.618	0.2056	22.233	22.679
F_{30}	24.015	23.943	14.7954	0.772	24.355	24.407	0.1330	23.634	23.866

Table 20
The time (in seconds) for 30 runs of partial benchmark functions in D = 100.

	fb_TSA	TSA	STSA	EST_TSA	SCA	ABC	DE	PSO	CLPSO
F_1	0.346	0.286	24.0072	0856	0.589	0.635	0.1179	0.0133	0.204
F_2	0.901	0.456	21.8802	0.745	1.077	1.118	0.1047	0.647	0.785
F_3	1.378	1.331	21.1197	0.919	1.591	1.4216	0.1041	1.13	1.272
F_4	1.851	1.794	22.1800	0.825	2.021	2.063	0.1030	1.603	1.738
F_5	2.344	2.296	21.6156	0.834	2.543	2.584	0.1098	2.0786	2.213
F_6	5.903	5.504	53.3709	4.887	9.167	9.438	0.8795	2.699	4.885
F_7	9.757	9.697	22.8632	0.746	9.979	10.021	0.1251	9.453	9.632
F_8	10.138	10.103	19.1246	0.588	10.185	10.221	0.0874	10.029	10.069
F_9	10.506	10.453	22.8163	0.869	10.712	10.753	0.1071	10.235	10.389
F_{10}	10.981	10.934	21.1495	0.792	11.139	11.177	0.1040	10.765	5.440
F_{11}	11.568	11.506	26.0309	0.855	11.882	11.931	0.1358	11.195	11.416
F_{12}	12.719	12.615	28.4855	1.427	13.442	13.518	0.2461	11.960	12.456
F_{13}	13.783	13.733	22.8660	0.687	13.951	13.989	0.1027	13.529	3.672
F_{14}	14.248	14.191	26.0382	0.697	14.418	14.455	0.1026	14.001	14.132
F_{15}	14.748	14.691	22.3108	0.726	14.96	15.002	0.1124	14.468	14.628
F_{16}	15.29	15.238	21.8285	0.732	15.503	15.544	0.1326	15.015	15.166
F_{17}	15.889	15.821	24.2055	0.792	16.146	16.196	0.1534	15.558	15.743
F_{18}	16.52	16.45	22.0553	0.724	16.736	16.782	0.1407	16.211	16.365
F_{19}	17.780	17.640	32.9438	1.346	18.615	18.707	0.2664	18.816	17.466
F_{20}	18.972	18.924	28.5300	0.677	19.166	19.206	0.1410	18.720	18.862
F_{21}	19.511	19.455	24.8771	0.78	19.733	19.776	0.1291	19.22	19.388
F_{22}	20.131	20.072	22.2520	0.821	20.407	20.454	0.1355	19.790	19.995
F_{23}	21.821	21.635	29.2674	1.712	23.028	23.136	0.3197	20.502	21.395
F_{24}	24.013	23.886	26.2333	3.6568	24.764	24.873	0.2351	23.172	23.722
F_{25}	25.976	25.828	29.6840	1.523	26.948	27.045	0.2814	24.916	25.623
F_{26}	32.06	31.475	63.9164	5.931	36.765	37.121	1.2602	27.207	30.585
F_{27}	41.824	41.229	71.4691	5.853	46.597	47.006	1.1433	37.288	40.341
F_{28}	48.631	48.422	42.7324	2.053	50.183	50.335	0.3866	47.066	48.113
F_{29}	51.921	51.691	37.6615	2.379	53.462	53.607	0.4196	50.395	51.384
F_{30}	54.625	54.473	27.8951	1.914	55.586	55.687	0.2999	53.645	54.27

in st_TSA has a better performance when compared with TSA. At the same time, the redesigned seed number generation mechanism in ns_TSA has its superiority proved by the experiment. To enhance the capability of TSA, both *ST* and *ns* are combined in the feedback mechanism with the best performance when compared with TSA, st_TSA and ns_TSA. Hence, there are some **findings** in this paper.

- Change *ST* dynamically in the whole searching process will help to find the global optimal solution in TSA.
- Determine *ns* by seed evaluation has a positive effect on finding the global optimal solution.
- Enhance the feedback mechanism in TSA will increase the continuous optimization capability of TSA significantly.

Table 21
Results of the Wilcoxon's test for fb_TSA and other algorithms on 30 test functions with D=10.

Algorithm	Better	Worst	W^+	W^-	p-value	$\alpha = 0.1$	$\alpha = 0.05$
fb_TSA VS TSA	23	7	371	65	0.00976	Yes	Yes
fb_TSA VS GWO	19	11	340	125	0.1064	No	No
fb_TSA VS STSA	24	6	391	74	1.73E−06	Yes	Yes
fb_TSA VS SCA	30	0	500	0	1.734E−06	Yes	Yes
fb_TSA VS ABC	26	4	395	70	8.307E−05	Yes	Yes
fb_TSA VS PSO	28	2	445	8	4.801E−06	Yes	Yes
fb_TSA VS DE	30	0	465	0	1.73E−06	Yes	Yes
fb_TSA VS CLPSO	24	6	371	95	0.0047	Yes	Yes

Table 22
Results of the Wilcoxon's test for fb_TSA and other algorithms on 30 test functions with D=30.

Algorithm	Better	Worst	W^+	W^-	p-value	$\alpha = 0.1$	$\alpha = 0.05$
fb_TSA VS TSA	26	4	444	21	1.36E−05	Yes	Yes
fb_TSA VS GWO	25	5	394	71	0.0041	Yes	Yes
fb_TSA VS STSA	28	2	459	6	3.18E−06	Yes	Yes
fb_TSA VS SCA	29	1	459	6	3.18E−06	Yes	Yes
fb_TSA VS ABC	29	1	428	36	6.319E−05	Yes	Yes
fb_TSA VS DE	30	0	465	0	1.73E−06	Yes	Yes
fb_TSA VS PSO	26	4	445	8	5.306E−05	Yes	Yes
fb_TSA VS CLPSO	26	4	435	30	3.72E−05	Yes	Yes

Table 23
Results of the Wilcoxon's test for fb_TSA and other algorithms on 30 test functions with D=50.

Algorithm	Better	Worst	W^+	W^-	p-value	$\alpha = 0.1$	$\alpha = 0.05$
fb_TSA VS TSA	27	3	451	14	6.98E−06	Yes	Yes
fb_TSA VS GWO	22	8	353	112	0.0132	Yes	Yes
fb_TSA VS STSA	27	3	458	7	2.60E−05	Yes	Yes
fb_TSA VS SCA	29	1	460	5	2.87E−06	Yes	Yes
fb_TSA VS ABC	26	4	430	35	4.86E−05	Yes	Yes
fb_TSA VS DE	29	1	449	16	8.47E−06	Yes	Yes
fb_TSA VS PSO	26	4	454	11	2.84E−05	Yes	Yes
fb_TSA VS CLPSO	25	5	436	29	2.84E−05	Yes	Yes

Table 24
Results of the Wilcoxon's test for fb_TSA and other algorithms on 30 test functions with D=100.

Algorithm	Better	Worst	W^+	W^-	p-value	$\alpha = 0.1$	$\alpha = 0.05$
fb_TSA VS TSA	28	2	461	4	2.603E−06	Yes	Yes
fb_TSA VS GWO	24	6	383	82	0.0166	Yes	Yes
fb_TSA VS STSA	29	1	465	5	2.88E−06	Yes	Yes
fb_TSA VS SCA	30	0	465	0	1.734E−06	Yes	Yes
fb_TSA VS ABC	28	2	439	26	2.163E−05	Yes	Yes
fb_TSA VS DE	29	1	461	5	2.88E−06	Yes	Yes
fb_TSA VS PSO	26	4	425	40	8.918E−05	Yes	Yes
fb_TSA VS CLPSO	27	3	459	6	1.49E−05	Yes	Yes

Table 25
Results of the Friedman test for fb_TSA and other algorithms on 30 test functions.

	Algorithm	p-value	$\alpha = 0.05$		Algorithm	p-value	$\alpha = 0.05$
D=10	fb_TSA VS TSA	0.008151	Yes	D=30	fb_TSA VS TSA	5.90E−06	Yes
	fb_TSA VS GWO	0.273	No		fb_TSA VS GWO	0.001	Yes
	fb_TSA VS STSA	4.3205E−08	Yes		fb_TSA VS STSA	2.00E−06	Yes
	fb_TSA VS SCA	4.3205E−08	Yes		fb_TSA VS SCA	3.19E−06	Yes
	fb_TSA VS ABC	5.90E−05	Yes		fb_TSA VS ABC	1.20E−05	Yes
	fb_TSA VS DE	4.3205E−08	Yes		fb_TSA VS DE	4.32E−08	Yes
	fb_TSA VS PSO	3.00E−06	Yes		fb_TSA VS PSO	5.90E−05	Yes
	fb_TSA VS CLPSO	0.001	Yes		fb_TSA VS CLPSO	2.61E−04	Yes
D=50	fb_TSA VS TSA	5.90E−05	Yes	D=100	fb_TSA VS TSA	2.00E−06	Yes
	fb_TSA VS GWO	0.011	Yes		fb_TSA VS GWO	0.011	Yes
	fb_TSA VS STSA	1.90E−05	Yes		fb_TSA VS STSA	3.19E−07	Yes
	fb_TSA VS SCA	3.19E−08	Yes		fb_TSA VS SCA	2.00E−06	Yes
	fb_TSA VS ABC	5.90E−05	Yes		fb_TSA VS ABC	4.32E−08	Yes
	fb_TSA VS DE	3.19E−07	Yes		fb_TSA VS DE	4.19E−08	Yes
	fb_TSA VS PSO	4.18E−04	Yes		fb_TSA VS PSO	2.61E−04	Yes
	fb_TSA VS CLPSO	1.05E−03	Yes		fb_TSA VS CLPSO	5.90E−05	Yes

In this paper, we propose two hypotheses, which are verified in the paper. The **verifications** are as follows:

- **Do not to reject hypothesis 1**: dynamic *ST* value determined by the feed-back mechanism helps to find global optimal value through the whole searching process.

Table 26
The result for the Tension/Compression Spring Design problem (T/CSD).

Design variables	Optimal values for variables									
	fb_TSA		TSA		ABC		SCA		PSO	
	Mean		Mean		Mean		Mean		Mean	
x_1	0.05		0.05		0.05		0.06		0.06	
x_2	0.38		0.37		0.34		0.61		0.49	
x_3	8.50		8.55		13.08		8.04		7.46	
$g_1(x)$	−5.23E−04		−1.01E−08		−0.01		−0.13		−1.88E−08	
$g_2(x)$	−3.01E−04		−5.54E−08		−0.12		−0.15		−0.11	
$g_3(x)$	−4.86		−4.86		−2.76		−3.47		−4.23	
$g_4(x)$	−0.72		−0.72		−0.74		−0.55		−0.64	
$f(x)$	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
	0.0098	1.93E−05	0.00987	4.01285E−09	0.0132	0.0001	0.0175	0.0037	0.0133	0.0008

Table 27
The result for pressure vessel design — PVD problem.

Design variables	Optimal values for variables									
	fb_TSA		TSA		ABC		SCA		PSO	
	Mean		Mean		Mean		Mean		Mean	
$x_1(T_s)$	0.91		1.10		1.10		1.34		1.12	
$x_2(T_h)$	0.48		0.60		0.60		0.64		0.6	
$x_3(R)$	46.89		56.99		56.79		59.54		57.93	
$x_4(L)$	129.01		51.00		52.62		58.14		46.07	
$g_1(x)$	−0.00456		−4.88E−08		−4.89E−03		−0.19		−7.19236E−10	
$g_2(x)$	−0.02769		−0.59		−0.59		−0.63		−0.59	
$g_3(x)$	−3480.96		−61166.19		−4207.72		−132089.08		0	
$g_4(x)$	−110.99		−183.01		−187.38		−181.86		−193.93	
$f(x)$	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
	6302.61	217.52	7019.34	0.0011	7057.22	19.423	8810.03	1202.26	7036.57	36.39

Table 28
The result for Himmelblau's non-linear optimization problem.

Design variables	Optimal values for variables									
	fb_TSA		TSA		ABC		SCA		PSO	
	Mean		Mean		Mean		Mean		Mean	
x_1	78.011		78.07		78.02		78.01		78.00	
x_2	33.0294		33.14		33.01		33.00		33.36	
x_3	27.097		27.21		27.22		27.23		27.26	
x_4	44.982		44.86		44.70		45.00		45.00	
x_5	44.898		44.60		44.81		44.98		44.39	
$g_1(x)$	92.00		91.97		91.96		91.99		91.99	
$g_2(x)$	100.41		100.42		100.39		100.43		100.48	
$g_3(x)$	20.00		20.00		20.02		20.06		20.00	
$f(x)$	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
	−31021.608	**2.616**	−31003.15	7.6803	−30990.131	15.943	−30977.94	32.26606	−31006.988	33.437

Table 29
The result for welded beam design.

Design variables	Optimal values for variables									
	fb_TSA		TSA		ABC		SCA		PSO	
	Mean		Mean		Mean		Mean		Mean	
x_1	0.21		0.21		0.20		0.20		0.21	
x_2	2.27		2.27		2.58		2.73		2.18	
x_3	9.04		9.04		9.03		9.26		8.87	
x_4	0.21		0.21		0.22		0.21		0.21	
$g_1(x)$	−0.0906		−1.43E−01		−705.39		−734.24		−1.41E−06	
$g_2(x)$	−0.0475		−3.25E−01		−1731.98		−2345.67		−3.97E−04	
$g_3(x)$	−6.26E−07		−3.56E−06		−0.02		−0.02		−1.01E−06	
$g_4(x)$	−3.5412		−3.54		−3.42		−3.41		−3.52	
$g_5(x)$	−0.0804		−8.04E−02		−0.08		−0.07		−0.09	
$g_6(x)$	−0.2356		−2.36E−01		−0.24		−0.24		−0.24	
$g_7(x)$	−0.0388		−1.86E−01		−1389.72		−880.11		−931.09	
$f(x)$	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
	1.5603	**2.496E−05**	1.5603	6.526E−05	1.6977	0.0395	1.7026	0.0432	1.58764	0.06249

- **Do not to reject hypothesis 2**: *ns* is changed dynamically by feedback mechanism that can balance the exploration and exploitation.

As tested in CEC 2014 and 4 real engineering optimization problems, the proposed fb_TSA is demonstrated and proved to

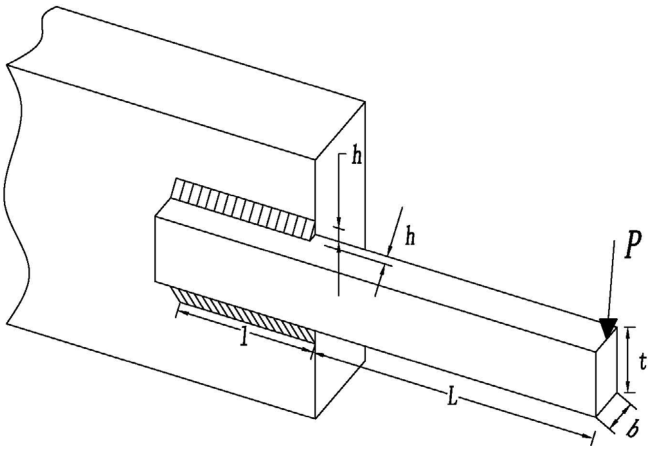


Fig. 15. Welded beam design.

be an excellent TSA variant that should be explored in the future. However, there are some **limitations** that should be explored in the future as the following.

- Its *ST* updating mechanism is a linear method, but different non linear *ST* updating mechanisms should be proposed to be a more adapter feedback mechanism;
- Its *ns* generation mechanism is implemented with some simple equations, but different adaptive *ns* generation mechanisms based on different searching phases or heuristics should be proposed in the future;
- Only 4 real engineering problems are validated, but many other optimization problems should be tested by the proposed fb_TSA or its variants.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Jianhua Jiang: Code modification, Writing - review & editing, Validation, Formal analysis, Supervision, Final review. **Xianqiu Meng:** Writing - review & editing, Code modification, Simulation experiment. **Yunjun Chen:** Simulation experiment, Document collection. **Chunyan Qiu:** Simulation experiment, Document collection. **Yang Liu:** Simulation experiment. **Keqin Li:** Supervision, Final review.

References

[1] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[2] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[3] C. Blum, Ant colony optimization: Introduction and recent trends, Phys. Life Rev. 2 (4) (2005) 353–373.
[4] G.N. Elnagar, M.A. Kazemi, M. Razzaghi, The pseudospectral Legendre method for discretizing optimal control problems, IEEE Trans. Automat. Control 40 (10) (1995) 1793–1796.
[5] R. Storn, K.V. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[6] A. Afroomand, S. Tavakoli, Vector-based swarm optimization algorithm, Appl. Soft Comput. 37 (2015) 911–922.
[7] N. Wang, W. Zhao, N. Wu, D. Wu, Multi-objective optimization, Expert Syst. Appl. 74 (2017) 96–104.
[8] M. Farina, K. Deb, P. Amato, Dynamic multiobjective optimization problems: test cases, approximations, and applications, IEEE Trans. Evol. Comput. 8 (5) (2004) 425–442.
[9] C.A. Coello, Evolutionary multi-objective optimization: a historical view of the field, IEEE Comput. Intell. Mag. 1 (1) (2006) 28–36.
[10] J. Du, Y. Yuan, T. Si, J. Lian, H. Zhao, Customized optimization of metabolic pathways by combinatorial transcriptional engineering, Nucleic Acids Res. 40 (18) (2012) 177–209.
[11] S.S. Raghavan, C. Woon, A. Kraus, K. Megerle, H. Pham, J. Chang, Optimization of human tendon tissue engineering: synergistic effects of growth factors for use in tendon scaffold repopulation, Plast. Reconstruct. Surg. 129 (2) (2012) 479–489.
[12] R.G. Bodade, C.N. Khobragade, S. Arfeen, Optimization of culture conditions for glucose oxidase production by a penicillium chrysogenum SRT 19 strain, Eng. Life Sci. 10 (1) (2010) 35–39.
[13] C. Nianyi, L. Wencong, C. Ruiliang, L. Chonghe, Q. Pei, Chemometric methods applied to industrial optimization and materials optimal design, Chemometr. Intell. Lab. Syst. 45 (1) (1999) 329–333.
[14] S. Jang, H. Ryoo, S. Ahn, J. Kim, G.H. Rim, Development and optimization of high-voltage power supply system for industrial magnetron, IEEE Trans. Ind. Electron. 59 (3) (2012) 1453–1461.
[15] J.M. Gordon, A. Rabl, Design, analysis and optimization of solar industrial process heat plants without storage, Sol. Energy 28 (6) (1982) 519–530.
[16] A. Arvay, J. French, J.C. Wang, X. Peng, A.M. Kannan, Nature inspired flow field designs for proton exchange membrane fuel cell, Int. J. Hydrogen Energy 38 (9) (2013) 3717–3726.
[17] M. Nawaz, E. Enscore, I. Ham, A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, Omega 11 (1) (1983) 91–95.
[18] K.S. Lee, Z.W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Comput. Methods Appl. Mech. Engrg. 194 (3638) (2005) 3902–3933.
[19] J.H. Holland, J.S. Reitman, Cognitive systems based on adaptive algorithms, ACM SIGART Bull. (1977) 49.
[20] Y. Wang, H. Li, T. Huang, L. Li, Differential evolution based on covariance matrix learning and bimodal distribution parameter setting, Appl. Soft Comput. 18 (2014) 232–247.
[21] Z.C.Y. Wang, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evol. Comput. 15 (2011) 55–66.
[22] Z.C.Y. Wang, Q. Zhang, Enhancing the search ability of differential evolution through orthogonal crossover, Inform. Sci. 185 (2012) 153–177.
[23] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (2008) 702–713.
[24] K. Deb, An efficient constraint handling method for genetic algorithms, Comput. Methods Appl. Mech. Engrg. 186 (2) (2000) 311–338.
[25] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Trans. Evol. Comput. 1 (1) (1997) 53–66.
[26] C. Tsai, K. Huang, C. Yang, M. Chiang, A fast particle swarm optimization for clustering, Soft Comput. 19 (2) (2015) 321–338.
[27] K. Mahadevan, P. Kannan, Comprehensive learning particle swarm optimization for reactive power dispatch, Appl. Soft Comput 10 (2) (2010) 641–652.
[28] D. Karaboga, C. Ozturk, A novel clustering approach: Artificial bee colony (ABC) algorithm, Appl. Soft Comput. 11 (1) (2011) 652–657.
[29] J. Jiang, Y. Feng, J. Zhao, K. Li, DateABC: A fast ABC based energy-efficient live VM consolidation policy with data-intensive energy evalation model, Future Gener. Comput. Syst. 74 (2017) 132–141.
[30] X. Yang, Firefly algorithm, stochastic test functions and design optimisation, Int. J. Bio-Inspired Comput. 2 (2) (2010) 78–84.
[31] E. Rashedi, H. Nezamabadipour, S. Saryazdi, GSA: A gravitational search algorithm, Inform. Sci. 179 (13) (2009) 2232–2248.
[32] A. Kaveh, V. Mahdavi, A hybrid CBO-PSO algorithm for optimal design of truss structures with dynamic constraints, Appl. Soft Comput. 34 (2015) 260–273.
[33] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, Inform. Sci. 222 (2013) 175–184.
[34] A.H. Kashan, League championship algorithm (LCA): An algorithm for global optimization inspired by sport championships, Appl. Soft Comput. 16 (2014) 171–200.
[35] H.E.A. Sadollah, A. Bahreininejad, M. Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Inform. Sci. 13 (2013) 2592–2612.
[36] R.V. Rao, V. Savsani, D.P. Vakharia, Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems, Comput.-Aided Des. 43 (3) (2011) 303–315.
[37] C. Blum, J. Puchinger, G.R. Raidl, A. Roli, Hybrid metaheuristics in combinatorial optimization: a survey, Appl. Soft Comput. J. 11 (6) (2011) 4135–4151.
[38] M. Ehrgott, X. Gandibleux, Hybrid Metaheuristics for Multi-objective Combinatorial Optimization, Springer Berlin Heidelberg, 2008.

[39] G. Wang, L. Guo, A novel hybrid bat algorithm with harmony search for global numerical optimization, J. Appl. Math. (2013) 1–21.
[40] G.G. Wang, A.H. Gandomi, A.H. Alavi, G.S. Hao, Hybrid krill herd algorithm with differential evolution for global numerical optimization, Neural Comput. Appl. 25 (2) (2014) 297–308.
[41] G. Wang, L. Guo, H. Duan, H. Wang, L. Liu, M. Shao, Hybridizing harmony search with biogeography based optimization for global numerical optimization, J. Comput. Theor. Nanosci. 10 (10) (2013) 2312–2322.
[42] J. Jiang, X. Yang, X. Meng, K. Li, Enhance chaotic gravitational search algorithm (CGSA) by balance adjustment mechanism and sine randomness function for continuous optimization problems, Physica A 537 (2020) 1–20.
[43] J. Jiang, M. Xu, X. Meng, K. Li, STSA: A sine Tree-Seed Algorithm for complex continuous optimization problems, Physica A (2019) 1–20.
[44] W. Gao, S. Feng, S. yang Liu, A modified artificial bee colony algorithm, Comput. Oper. Res. 39 (3) (2012) 687–697.
[45] X.H. Shi, Y. Liang, H. Lee, C. Lu, L. Wang, An improved GA and a novel PSO-GA-based hybrid algorithm, 93 (5) 255–261.
[46] N. Holden, A.A. Freitas, A hybrid PSO/ACO algorithm for discovering classification rules in data mining, J. Artif. Evol. Appl. (2008) 1–11.
[47] S. Nemati, M.E. Basiri, N. Ghasem-Aghaee, M.H. Aghdam, A novel ACO–GA hybrid algorithm for feature selection in protein function prediction, Expert Syst. Appl. 36 (10) (2009) 12086–12094.
[48] D.H. Wolpert, W.G. Macready, A hybrid artificial bee colony algorithm for numerical function optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82.
[49] M.S. Kiran, TSA: Tree-seed algorithm for continuous optimization, Expert Syst. Appl. 42 (19) (2015) 6686–6698.
[50] A. Babalik, A.C. Cinar, M.S. Kiran, A modification of tree-seed algorithm using Deb's rules for constrained optimization, Appl. Soft Comput. 63 (2018) 289–305.
[51] K. Deb, An efficient constraint handling method for genetic algorithms, Comput. Methods Appl. Mech. Engrg. 186 (2) (2000) 311–338.
[52] M. Aslan, M. Beskirli, H. Kodaz, M.S. Kiran, An improved tree seed algorithm for optimization problems, Int. J. Mach. Learn. Comput. 8 (1) (2018) 20–25.
[53] J. Jiang, S. Jiang, X. Meng, C. Qiu, EST-TSA: An effective search tendency based to tree seed algorithm, Physica A 534 (2019).
[54] N. Westerhof, P. Sipkema, G. Den Bos, G. Elzinga, Forward and backward waves in the arterial system, Cardiovasc. Res. 6 (6) (1972) 648–656.
[55] J.C. Duchi, Y. Singer, Efficient online and batch learning using forward backward splitting, J. Mach. Learn. Res. 10 (2009) 2899–2934.
[56] J.G. March, Exploration and exploitation in organizational learning, Organ. Sci. 2 (1) (1991) 71–87.
[57] T. Hinamoto, K. Higashi, W. Lu, Separate/joint optimization of error feedback and coordinate transformation for roundoff noise minimization in two-dimensional state-space digital filters, IEEE Trans. Signal Process. 51 (9) (2003) 2436–2445.
[58] D. Moerder, A. Calise, Convergence of a numerical algorithm for calculating optimal output feedback gains, IEEE Trans. Automat. Control 30 (9) (1985) 900–903.
[59] H.A. Hashim, S. Elferik, M.A. Abido, A fuzzy logic feedback filter design tuned with PSO for L 1 adaptive controller, Expert Syst. Appl. 42 (23) (2015) 9077–9085.
[60] J. Gordon, ABC of learning and teaching in medicine one to one teaching and feedback, BMJ 326 (7388) (2003) 543–545.
[61] K. Sundareswaran, P.S.R. Nayak, Ant colony based feedback controller design for soft-starter fed induction motor drive, Appl. Soft Comput. 12 (5) (2012) 1566–1573.
[62] J. Jiang, Y. Feng, M. Parmar, K. Li, FP-ABC: fast and parallel ABC based energy-efficiency live VM allocation policy in data centers, Sci. Program. 2016 (2016) 1–9.
[63] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[64] Mirjalili, Seyedali, Sca: a sine cosine algorithm for solving optimization problems, Knowl.-Based Syst (2016).
[65] A.D. Belegundu, A study of mathematical programming methods for structural optimization, Internat. J. Numer. Methods Engrg. 21 (9) (1985) 1601–1623.
[66] C.A.C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 (2000) 113–127.
[67] G.G. Dimopoulos, Mixed-variable engineering optimization based on evolutionary and social metaphors, Comput. Methods Appl. Mech. Eng. 196 (4) (2007) 803–817.
[68] H. Garg, A hybrid PSO-GA algorithm for constrained optimization problems, Appl. Math. Comput. 274 (2016) 292–305.
[69] K. Deb, An efficient constraint handling method for genetic algorithms, Comput. Methods Appl. Mech. Eng. 186 (2) (2000) 311–338.
[70] H. Garg, A hybrid GSA-GA algorithm for constrained optimization problems, Inform. Sci. 478 (2019) 499–523.
[71] C.A.C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 (2) (2000) 113–127.