Journal of Computers Vol. 30 No. 4, 2019, pp. 63-77
doi:10.3966/199115992019083004006

Application of Adaptive Hybrid Teaching-learning-based Optimization Algorithm in Flatness Error Evaluation

Yang Yang[1*], Ming Li[1], Jing-Jun Gu[1]

[1] Key Laboratory of Intelligent Manufacturing and Robotics, School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, 200072, China
mryyfi@i.shu.edu.cn, robotlib@shu.edu.cn, sdulmhsh@163.com

Received 2 October 2017; Revised 17 March 2018; Accepted 3 May 2018

Abstract. To improve the computational accuracy of the flatness evaluation under the minimum zone method, an artificial intelligence optimization algorithm called teaching-learning based optimization (TLBO) was applied to the evaluation of the flatness. On the basis of the TLBO algorithm, an adaptive hybrid teaching-learning-based optimization (AHTLBO) algorithm was designed by introducing an adaptive factor and shuffled frog leaping algorithm (SFLA) that are used to improve the search ability of the algorithm, in order to further improve the precision of the algorithm. Finally, the AHTLBO algorithm was verified by seven sets of experiments, and the calculation results were compared with several other common algorithms. The experimental results show that the AHTLBO has higher precision and faster convergence speed in the evaluation of flatness, and it is suitable for high-precision flatness evaluation.

Keywords: adaptive factor, adaptive hybrid teaching-learning-based optimization (AHTLBO), flatness, minimum zone method, shuffled frog leaping algorithm (SFLA)

1 Introduction

With the development of precision manufacturing technology, the improvement of the geometric quality of machine parts has become one of the most important research projects in the field of mechanical engineering. As important technical parameters for judging the quality of parts, form and position errors have been studied extensively and deeply, and a series of geometric specifications have been established in the world [1-2]. In form and position errors, the flatness error is one of the most basic geometric evaluation elements, and it is widely used in the evaluation of mechanical parts. Therefore, it is very important to ensure the evaluation accuracy of the flatness of the parts.

In the methods of flatness error evaluation, the least squares method (LSM) and the minimum zone method (MZM) are the most common algorithms [3]. The LSM is a relatively simple evaluation method that is mainly used in the engineering field, but its evaluation result is not accurate. For the minimum zone method, it is an accurate algorithm that can meet the related standards, but there is no uniform formula for its computation. Therefore, scholars have conducted many studies on evaluation algorithms for MZM in recent years.

For the evaluation of the flatness error based on the MZM algorithm, Mark T introduced the convex hull (CH) into the evaluation of the flatness error. The precision of its calculation result is higher than that of the LSM [4]. Kanada applied the downhill simplex method (DSM) to the evaluation of flatness, and the bracketing method and the least-squared method were also discussed in the paper [5]. Carr proposed an error evaluation method for solving flatness and straightness using linear programming (LP) which can satisfy the minimum zone principle [6]. Cheraghi developed a method of flatness and straightness errors evaluation based on a linear search algorithm (LSA) [7]. Lee used convex hull theory to analyze to the flatness error, which enhanced the evaluation accuracy [8]. Samuel introduced computational geometry into the form and position errors evaluation such as flatness and straightness [9].

* Corresponding Author

Wen applied the improved genetic algorithm (IGA) to the flatness error evaluation and achieved good results [10-11]. Tseng proposed the application of GA in form and position errors, which included flatness [12]. Cho utilized the data envelopment analysis (DEA) for flatness and straightness errors evaluation [13]. Luo improved the artificial bee colony (ABC) algorithm and applied it to the evaluation of the flatness error [14]. Hermann introduced computational geometry (CG) into form and position errors evaluation such as flatness and straightness [15]. Tu built a unified mathematical model of shape error that includes flatness, but there was no experiment to test the model [16]. Muhammad proposed a new flatness error algorithm called efficient genetic algorithm (EGA) [17]. Li combined the nonlinear optimization method with the computational geometry method and presented a new method for evaluating the flatness error [18]. Radlovacki established a one-point plane bundle (OPPB) method for flatness evaluation [19]. Pathak V K designed a constriction factor-based particle swarm optimization (CFPSO) and applied it to the shape error, which included flatness [20]. In summary, the related works for the evaluation of the flatness are shown in Table. 1.

Table 1. Related work of the flatness evaluation

Number	Year	Method	Number	Year	Method
1	1989	CH	7	2003	GA
2	1993	DSM	8	2006	IGA
3	1995	LP	9	2012	IGA/DEA/DEA
4	1996	LSA	10	2015	CG
5	1997	CH	11	2016	EGA/OPPB/CG
6	1999	CH	12	2017	CFPSO

Overall, a variety of methods have been applied to flatness error evaluation such as the operational research method [5-6] and computational geometry [9, 15]. Although the above methods have improved the evaluation accuracy of flatness, it is still difficult to satisfy the minimum zone principle. In recent years, the development of intelligent optimization algorithms has provided a good method for solving such complex nonlinear problems. They have been applied to the flatness error evaluation process because of their high precision and efficiency, such as GA [10-11]. In addition, compared with computational geometry and operations research, intelligent optimization algorithms have the advantages of simple mathematical model. However, these algorithms are more dependent on the control parameters, and the algorithm accuracy can be further improved. Therefore, how to improve the accuracy of the algorithm to obtain more accurate flatness error information is the focus of the current research.

The teaching-learning-based optimization (TLBO) algorithm [21] proposed in 2012 is a novel swarm intelligence optimization algorithm. It mainly simulates the interaction and learning between teachers and students in the class and has been widely applied in the engineering optimization field [22-23]. With a few control parameters, the TLBO algorithm has good calculation precision and convergence speed. However, the TLBO algorithm also easily fall into local optimal problems in the iterative process. Therefore, in order to further improve the solving ability of TLBO and the accuracy of the flatness evaluation, an adaptive hybrid teaching-learning-based algorithm (AHTLBO) that is integrated into the shuffled frog leaping algorithm (SFLA) and the adaptive factor method is proposed.

Combined with the advantages of the SFLA algorithm, first, a class grouping strategy is designed in the class initialization of TLBO. It ranks and groups the students according to their scores, to avoid the premature convergence to the initial optimal solution. Second, during the iterative process of TLBO, the adaptive teaching factor and local updating strategy are introduced to adjust the information updating ability and improve the worst solution, which can further improve the quality of the solution. Finally, after each iteration, the students of each class are mixed and regrouped to achieve the exchange of global information between students, and further improve the global search ability of the algorithm. Through the test functions and the flatness data in the related literature, the results show that the AHTLBO algorithm is superior to the existing algorithms with respect the optimization accuracy and iterative speed in flatness error evaluation.

2 TLBO Algorithm

In the basic teaching-learning based algorithm, all students in the class are the population that can be represented as $X = (x_1, x_2, \ldots, x_N)$, where N is the number of the class students. For each student x_i, it can be represented as $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D})$, where D is the number of the courses and the dimensions of the problem. For each dimension, there is an upper limit $x_{i,j}^u$ and a lower limit $x_{i,j}^l$. In each class, the best student will be the teacher, which is represented as x_t. On the basis of the initial parameters, the basic TLBO algorithm consists of two stages. The first stage is mainly the teacher's teaching process. In the teaching stage, the students get the knowledge by communicating with the teacher. The second stage is the students' learning process. In this stage, students get knowledge from each other. The detailed process of the basic TLBO algorithm is as follows.

2.1 Teaching Stage

First, all students should be ranked according to the fitness value of each student in the class, from which the best student will be the teacher, and the students will be guided by the teacher. In the teaching stage, the teacher passes his knowledge to the students to improve the overall performance of the whole class students. The teaching phase can be expressed as formulas (1)-(3):

$$x(i,j)_{new}^{k+1} = x(i,j)_{old}^{k} + rand \times (x_t(j)^k - t_f(j)^k),\ j = 1, 2, ..., D. \tag{1}$$

$$t_f = round[1 + rand(0,1)]. \tag{2}$$

$$M(j)^k = \frac{1}{N}\sum_{1}^{N} x(j). \tag{3}$$

Where k is the number of iterations; i is the number of class students; j is the number of courses; $x(i,j)_{old}^{k}$ is the result of the student before teaching; $x(i,j)_{new}^{k+1}$ is the result of the student after teaching; $x_t(j)^k$ is the score of the teacher's j course when the number of iterations is k; t_f is the teaching factor which value is 1 or 2; is a random function which value is 0 to 1; $M(j)^k$ is the average score of the subject j in class; In the iterative process, the results of the students are updated by comparing the fitness function value between $x(i,j)_{old}^{k}$ and $x(i,j)_{new}^{k+1}$.

2.2 Learning Stage

In the learning stage, the students in the class acquire new knowledge by communicating with each other to improve their scores, and the scores can be adjusted by randomly selecting two students to communicate. For the minimization problems, the specific expression is shown in formula (4):

$$x(i,j)_{new}^{k+1} = \begin{cases} x(i,j)_{old}^{k+1} + rand \times (x(i,j)_{old}^{k+1} - x_f) & f(x(i,j)_{old}^{k} < f(x_f) \\ x(i,j)_{old}^{k} + rand \times (x_f - x(i,j)_{old}^{k}) & f(x(i,j)_{old}^{k} > f(x_f) \end{cases}. \tag{4}$$

Where $x(i,j)_{old}^{k}$ is the score of the students before learning; $x(i,j)_{new}^{k+1}$ is the score of the students after learning; x_f is the student who communicates with student $x(i,j)_{old}^{k}$;The learning process is to determine the object of the learning by comparing the fitness value of $x(i,j)_{old}^{k}$ and x_f.

If the number of iterations reaches the requirement, the algorithm terminates. Otherwise, returns to section 2.1. The simple flow chart of the TLBO algorithm is shown in Fig. 1.

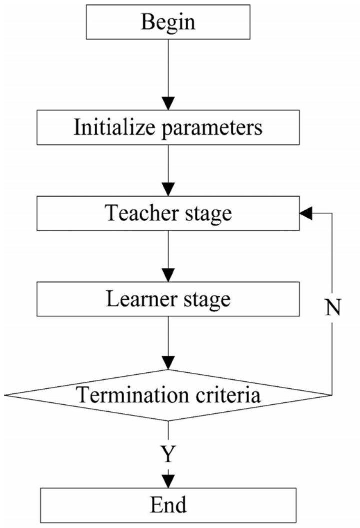

Fig. 1. Flow chart of the TLBO algorithm

3 SFLA and Adaptive Teaching Factor

For the basic TLBO, the algorithm has a few set parameters and higher stability. However, as other intelligent optimization algorithms, the TLBO algorithm also prematurely converges and easily falls into the local optimum. Therefore, through the design of the related strategy, the solution accuracy and convergence speed of the algorithm are further improved.

3.1 SFLA

The SFLA was proposed in 2003 by Eusuff M and Lansey [24]. The SFLA has a few control parameters, a simple principle and a strong global search capability. The frog group is composed of several subgroups, which can obtain the global optimal solution through the in-depth search of the subgroups and the information exchange between the individual frog groups. In each of the subgroups, individual frogs were updated through meta-evaluation strategies until the worst frog in the subgroups was replaced. At the end of each iteration, the individuals were reordered according to the fitness value, in order to construct new subgroups and conduct the next iteration, so that the global information could be fully communicated. When the number of iterations or the accuracy of iteration reaches the requirements, the SFLA solution is completed and the global optimal solution is solved. The specific steps of the SFLA algorithm are shown below:

3.1.1 Grouping Strategy

There are $F = M \times N$ initial solutions to the different problems, where F is the total number of the initial solutions, M is the number of subgroups, and N is the number of individuals per subgroup. The fitness function of all individuals is calculated and sorted according to the fitness value. The sort method is described as follows. Place the number one in the 1st subgroup, and then place the number two into 2nd subgroups. Then, place the number M into Mth subgroups, and place the number M+1 into M+1th subgroups until all individuals are assigned into individual subpopulations.

3.1.2 Local Search

In the SFLA, the local search process updates the worst solution in each subgroup. In each subpopulation, the worst solution was replaced by using deep search. The detailed calculation formulas (5)-(7) are shown below:

$$x'_w = x_w + rand \times (x_b - x_w). \tag{5}$$

$$x'_w = x_w + rand \times (x_g - x_w). \tag{6}$$

$$x'_w = x_w + rand \times (x_{\max} - x_{\min}). \tag{7}$$

Where x'_w is the worst individua l after the update; x_w is the worst individual in the sub group; x_b is the optimal individual in the subgroup; x_g is the optimal individual in the global; *rand* is a random number which is 0 to 1; is the optimal solution in the search space; is the worst solution in the search space. In the depth search process, the solution is update according to formula (5) first, and if $f(x'_w) < f(x_w)$, then replace x_w with x'_w; Else the x_w will be updated again by formula (6), and if $f(x'_w) < f(x_w)$, then replace x_w with x'_w; Else the x_w will be updated again by formula (7), and then as an iterative loop until the number of local iterations is reached.

3.1.3 Population Mixing

When the local iteration number is completed, the subpopulations are merged to a new population. Determine whether the global iteration number meet the termination criteria, and if so, the optimal solution is output. If not, returns to section 3.1.1. The simple flow chart of the SFLA algorithm is shown in Fig. 2.

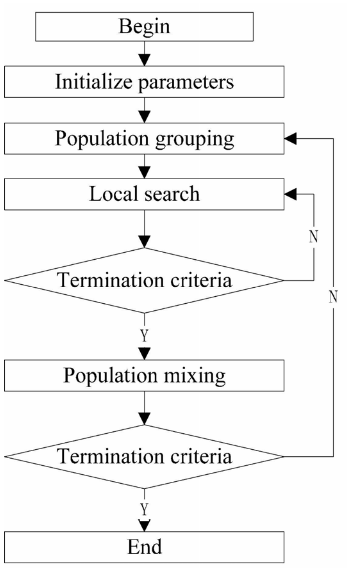

Fig. 2. Flow chart of the SFLA algorithm

3.2 Adaptive Teaching Factor

In the basic TLBO algorithm, the teaching factor t_f is a very important parameter that determines the learning situation of the class students from the teacher. The value of the teaching factor is 1 or 2. Its means that the students in the class have only two result cases. One is to get all of the knowledge from the teacher, and the other is to get none of the knowledge from the teacher. Therefore, it is not a good learning effect. In the actual teaching process, the students who study in different stages have different learning ability, in the early stage of the learning, the knowledge is relatively simple, and the acceptance ability of students is strong that is easy to get the knowledge from the teacher. With the increase of the difficulty of the knowledge, the acceptance ability of the students will decrease, therefore, the teaching factor t_f need to be adjusted according to the learning stage. A new t'_f is proposed by Rao, the adaptive teaching factor is shown as formula [25] (8):

$$t'_f = \frac{M_j^k}{M_{t,j}^k} \tag{8}$$

Where M_j^k is the average score of the course j in class students ; $M_{t,j}^k$ is the score of the teacher.

4 AHTLBO Algorithm

In the basic TLBO algorithm, there is only one teacher in the class. Thus, at the beginning of the iteration, all the students in the class learn from the same teacher. Therefore, in the iterative process, the population diversity is rapidly lost and the algorithm falls into a local optimum. In the SFLA, the frog group information was fully exchanged in the whole world by grouping the frog groups. In each group of frogs, the quality of solution of the frog group was further improved through the individual selection of the frog population. In addition, an adaptive teaching factors is introduced into the teaching process to improve the learning efficiency of the students in class. The solution precision of the TLBO algorithm is further improved by these methods.

The specific process of the AHTLBO algorithm is shown below:

(1) Parameter initialization. The initial parameters of AHTLBO are class size F, the number of students in each group M, the dimension of the problem D, the variable upper bound U and the lower bound L, the number of iterations in AHTLBO is I_1, and the number iterations of the local search is I_2. Then, go to step (2).

(2) Grouping strategy. The class grouping strategy is conducted according to the method described in section 3.1.1. Then, go to step (3).

(3) Teaching stage. In each group, the teaching process is described in section 2.1, which is updated according to formula (1)-(3). The teaching factor is replaced by formula (8). Then, go to step (4).

(4) Learning stage. According to section 2.2, the students learn from each other as in formula (4). And record the best solution x_g in the class, the best solution x_b in each group, and the worst individual in the subgroup x_w. Then go to step (5).

(5) Local search. According to section 3.1.2, the worst solution in each group is updated until the iteration of the local search is reached. Then, go to step (6).

(6) Population mixing. According to section 3.1.3, the subpopulations are merged to a new population. Then, go to step (7).

(7) Determine whether the number of iterations required for termination is satisfied. If the requirement is satisfied and the process is terminated, the global optimal solution x_g is output. If not, then return to the step (2) and repeat the calculation again. The simple flow chart of the AHTLBO algorithm is shown in Fig. 3.

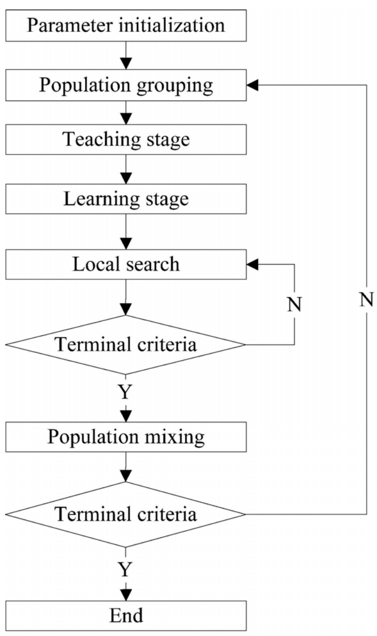

Fig. 3. Flow chart of AHTLBO

5 Mathematical Model of Flatness

According to the relevant geometric standard, the flatness error that is based on the minimum principle is essentially the least distance between the two ideal parallel planes that contain the measured points. Therefore, the equation for any plane in space is represented by the formula $z = Ax - By - C$. The formula of the distance between any point and the plane is shown in formula (9). As is shown in Fig. 4, f is the flatness error of the measured plane.

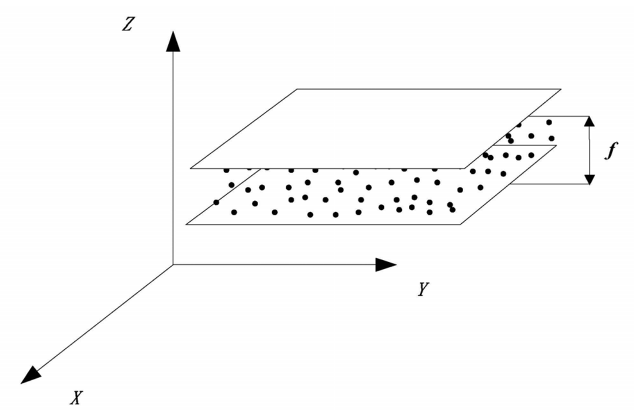

Fig. 4. Diagram of flatness

$$L_i = \frac{z_i - Ax_i - By_i - C}{\sqrt{A^2 + B^2 + 1}} \quad \textbf{(9)}$$

Where $P_i(x_i, y_i, z_i)(i = 1, 2, 3 \ldots n)$ is the measured point; *A*, *B* are the direction parameters of the plane; *C* is the positional parameter of the plane; L_i is the distance between $P_i(x_i, y_i, z_i)$ and the plane *S*;

As is seen in formula (9), even if *C* takes different values, the relative position in space for two parallel planes is fixed. Thus, *C* can take any value, but *A* and *B* are the main influential factors. Formula (10) shows the process of solving the two values of *A* and *B*. Therefore, according to the definition of the minimum zone principle method and the distance formula of the point to the plane, the objective function of the measured points is shown as formula (10).

$$f(A, B) = \min(\max(L_i) - \min(L_i)) \quad \textbf{(10)}$$

6 Experimental Verification

6.1 Performance Test

To measure the performance of AHTLBO, the standard test functions f_1 (formula (11)), f_2 (formula (12)), f_3 (formula (13)) and f_4 (formula (14)) are adopted. The Sphere (f_1) function is a simple one-peak function and mainly verifies the convergence speed of the algorithm. The Rosenbrock (f_2) function is also a single-peak function but it is relatively more complex. It is often used to test the optimal performance. Both the Rastrign (f_3) function and the Griewank (f_4) function have multiple peaks and have multiple local optimal solutions. However, the Griewank function is relatively more complex than the Rastrign function, and they are both commonly used to test the global search capability of the algorithm.

$$f_1 = \sum_{i=1}^{n} x_i^2 \quad (-30, 30). \quad \textbf{(11)}$$

$$f_2(x)=\sum_{i=1}^{n-1}(100*(x_{i+1}-x_1^2)^2+(x_i-1)^2 \quad (-30,30). \tag{12}$$

$$f_3(x)=\sum_{i=1}^{30}[x_i^2-10\cos 2\pi x_i \ +10] \ \ (-5.12,5.12). \tag{13}$$

$$f_4(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}(\frac{x_i}{\sqrt{i}})+1 \ \ (-600,600). \tag{14}$$

In order to test the calculation performance of the AHTLBO, the GA, PSO and TLBO are also tested as contrasting algorithms. The parameters of the AHTLBO are as follows: the initial population number is 50, the number of AHTLBO subgroups is 5, the dimension of the problem is 2, the global iteration number is 500, the local iteration number of AHTLBO is 20, and the algorithm is tested 50 times. The iterative curves of the test functions are shown in Fig. 5 to Fig. 8.

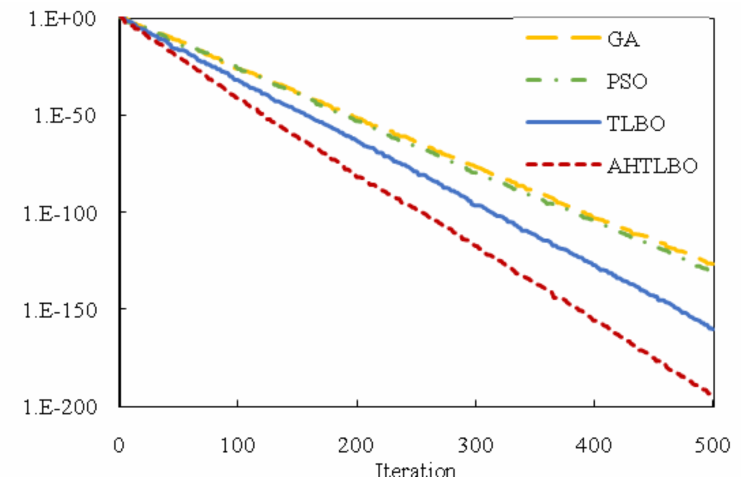

Fig. 5. Iteration curve of Sphere

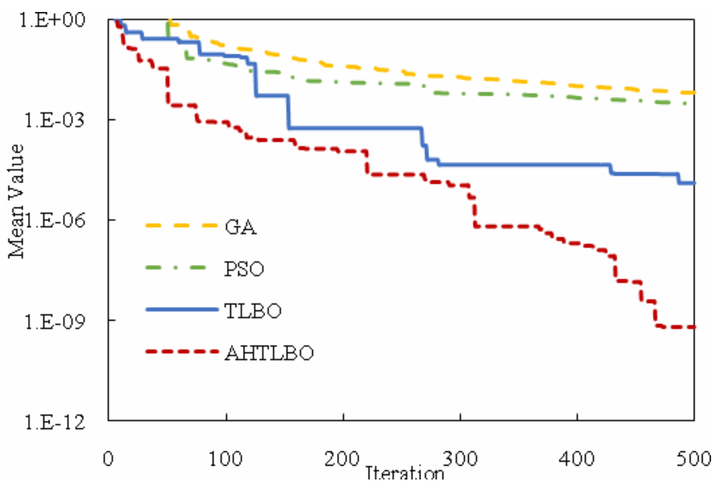

Fig. 6. Iteration curve of Rosenbrock

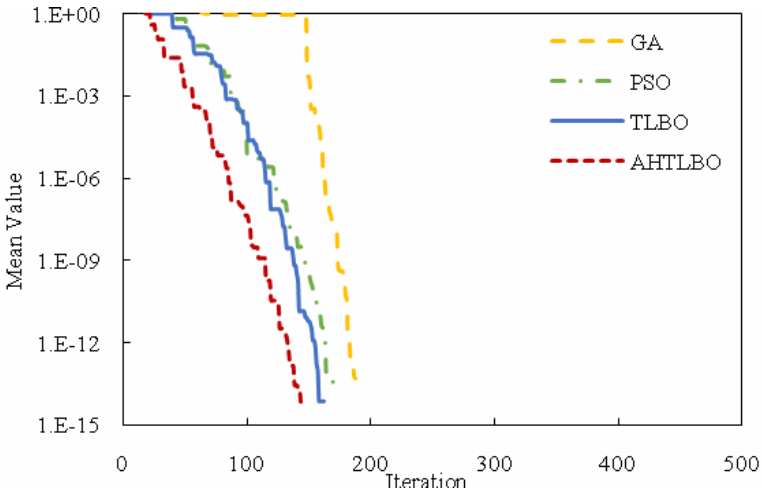

Fig. 7. Iteration curve of Rastrign

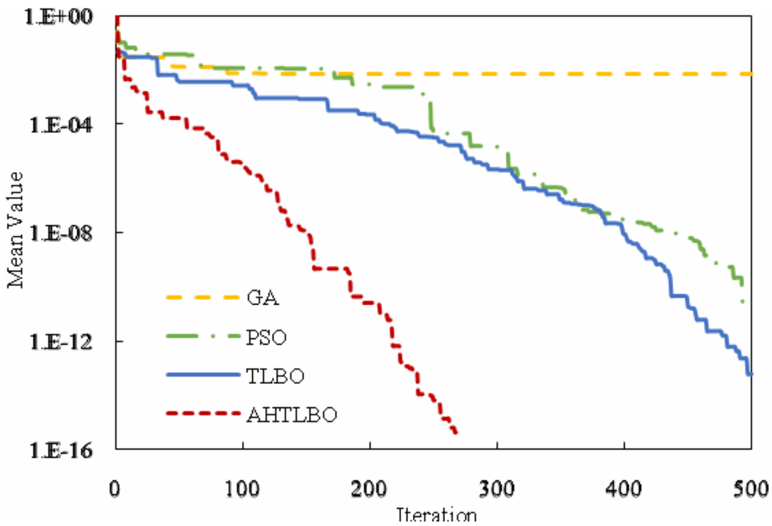

Fig. 8. Iteration curve of Griewank

The results of the test functions are shown in Table 2. For the test results of the Sphere and Rosenbrock functions, compared with the GA, PSO and TLBO algorithms, the AHTLBO algorithm has the lowest mean value, the highest calculation precision, the lowest standard deviation and the high computational stability. For Rastrign function, because of the low test dimension, four algorithms can find the global optimal solution in 500 iterations. However, it can see from Fig. 7 that the AHTLBO algorithm converges faster and converges to the optimal solution on average at 150 iterations. For the more complex Griewank function, in the two-dimensional test case, the GA, PSO, and TLBO algorithms have not found the global optimal solution, and only AHTLBO can find the global optimal solution. Based on the results of the test function of the four algorithms, it can been seen that the AHTLBO algorithm is better than the other three algorithms from the comparison of the experimental results in the accuracy (mean value) and the stability (standard deviation).

6.2 Example 1 of Flatness

In order to verify the validity of the algorithm in the flatness evaluation, the data of the measured plane was obtained by the coordinates measured machine, as is shown in Table 3. The flatness of the measured data is calculated by the LSM and the four different intelligent optimization algorithms.

Table 2. Test results of test functions

Algorithm	Function	Mean Value	Standard Deviation
GA	f_1	3.42E-128	9.22E-128
PSO		1.12E-132	4.31E-133
TLBO		6.15E-152	2.75E-153
AHTLBO		**1.15E-171**	**3.52E-171**
GA	f_2	6.18E-3	4.28E-3
PSO		2.29E-3	1.77E-3
TLBO		1.28E-5	6.45E-6
AHTLBO		**6.2E-10**	**4.53E-11**
GA	f_3	**0**	**0**
PSO		**0**	**0**
TLBO		**0**	**0**
AHTLBO		**0**	**0**
GA	f_4	8.20E-2	7.34E-3
PSO		1.26E-10	6.88E-11
TLBO		6.15E-14	1.33E-15
AHTLBO		**0**	**0**

Table 3. Measurement data mm

No	*x*	*y*	*z*	No	*x*	*y*	*z*
1	25.129	53.237	16.1324	17	25.129	29.237	16.1712
2	37.129	53.237	16.1959	18	37.129	29.237	16.2369
3	49.129	53.237	16.1854	19	49.129	29.237	16.2341
4	61.129	53.237	16.1499	20	61.129	29.237	16.1139
5	25.129	47.237	16.1509	21	25.129	23.237	16.2714
6	37.129	47.237	16.2504	22	37.129	23.237	16.2231
7	49.129	47.237	16.2359	23	49.129	23.237	16.2954
8	61.129	47.237	16.1611	24	61.129	23.237	16.1821
9	25.129	41.237	16.2101	25	25.129	17.237	16.2254
10	37.129	41.237	16.2314	26	37.129	17.237	16.1154
11	49.129	41.237	16.2084	27	49.129	17.237	16.1769
12	61.129	41.237	16.1934	28	61.129	17.237	16.1431
13	25.129	35.237	16.2789	29	25.129	11.237	16.1571
14	37.129	35.237	16.1311	30	37.129	11.237	16.1679
15	49.129	35.237	16.1269	31	49.129	11.237	16.2154
16	61.129	35.237	16.1414	32	61.129	11.237	16.2518

Table 4 shows the flatness results of Table 3, where the flatness error calculated by the LSM is 0.19021 mm and the parameter of the plane is (-0.00068,-0.000335). In the evaluation results of intelligent optimization algorithm, the result of the GA is 0.180175 mm, and the control parameters of the plane are -0.00014 and 0.00033. For the PSO and TLBO, the calculation results are 0.18093 mm and 0.18075 mm, respectively. The result of AHTLBO is 0.18070 mm, and the control parameters are -0.00016 and 0.00019. The above results show that the results of the intelligent optimization algorithms are higher than that of the LSM and the highest precision is the AHTLBO in intelligent optimization algorithm.

Table 4. Calculation results of flatness error mm

Method	*A*	*B*	*f*
LSM	-0.00068	-0.000335	0.19021
GA	-0.00014	0.00033	0.18175
PSO	0.00031	-0.00072	0.18093
TLBO	-0.00008	0.00003	0.18075
AHTLBO	-0.00016	0.00019	0.18070

Fig. 9 is the iterative curve of the different intelligent optimization algorithms. As seen from Fig. 9, the the convergence speed of AHTLBO is the fastest. The AHTLBO algorithm converges at the 45th iteration.

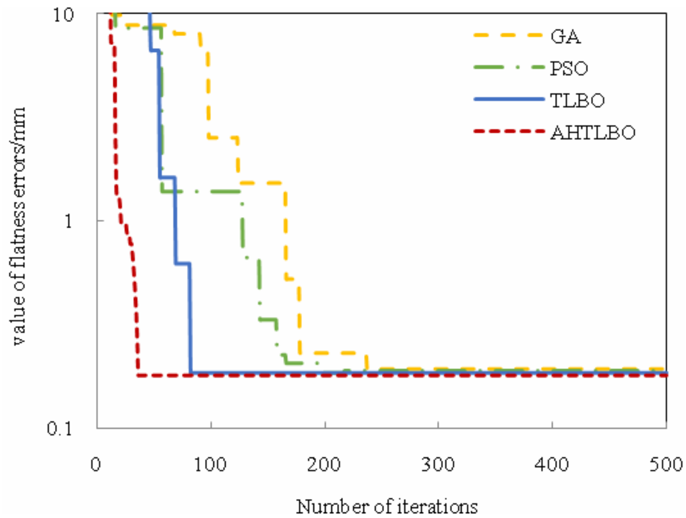

Fig. 9. Iteration curve of different algorithms

6.3 Example 2 of Flatness

In order to compare the effectiveness of the AHTLBO algorithm in the flatness evaluation, the correlation literature data are used to experiment. The calculation results of the TLBO and AHTLBO algorithms are compared with the relevant literatures in 2016 [7-8, 13, 18-19]. The measurement data can be seen in Table 5.

Table 5. The literature [7] measurement data mm

No	*X*	*Y*	*Z*	No	*X*	*Y*	*Z*
1	0	0	2	14	50	75	9
2	0	25	5	15	50	100	11
3	0	50	6	16	75	0	7
4	0	75	8	17	75	25	7
5	0	100	9	18	75	50	6
6	25	0	5	19	75	75	7
7	25	25	7	20	75	100	9
8	25	50	8	21	100	0	7
9	25	75	9	22	100	25	6
10	25	100	12	23	100	50	6
11	50	0	6	24	100	75	6
12	50	25	7	25	100	100	8
13	50	50	8				

The calculation results of the flatness error corresponding to the different evaluation methods are shown in Table 6. As with the results in the relevant literature, the calculation results of the LSA, convex hull are 4.8573 mm. Although their results are relatively small numerically, it can not meet the high-precision measurement field. To obtain more detailed flatness information, the OPPB method and the CH method were used to calculate the flatness error in 2016, and their results are 4.87260821 mm and 4.85733795080 mm, respectively. In this paper, the calculation results of the TLBO and AHTLBO algorithms are 4.857496 mm and 4.8573379507999 mm respectively, and the corresponding AHTLBO planar parameters of *A* and *B* are 0.0018181818181818 and 0.0509090909090909. Therefore, AHTLBO has the highest accuracy compared to the other methods.

Table 6. Calculation results mm

Methods	*A*	*B*	Error
LSA [7]	—	—	4.8573
Convex hull [8]	—	—	4.8573
DEA [13]	—	—	4.8573
CH [18]	0.0018158272713	0.0508431635972	4.85733795080
OPPB [19]	—	—	4.87260821
TLBO	0.00181824	0.0509101	4.857496
AHTLBO	0.0018181818181818	0.0509090909090909	4.8573379507999

Fig. 10 illustrates the iterative curves of the flatness error of the AHTLBO and TLBO algorithms. It can be seen that in the iterative process, the AHTLBO algorithm converges at 35 iterations, and its iterative speed is faster than that of the TLBO, it has better computational performance than the basic TLBO.

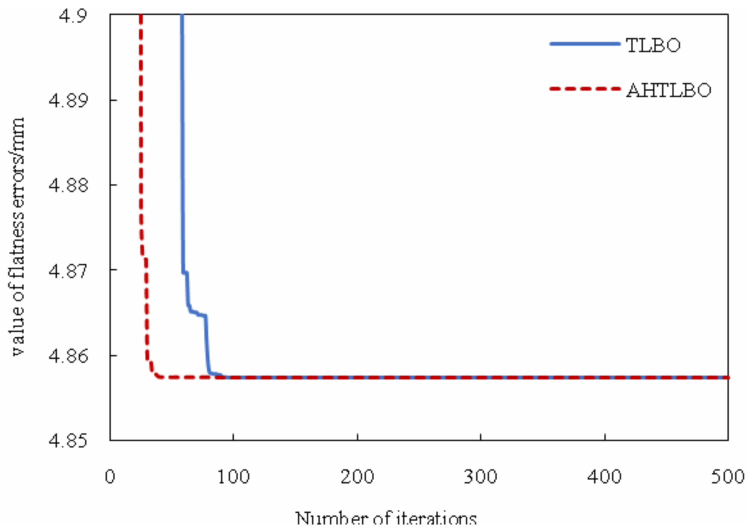

Fig. 10. Iteration curve of AHTLBO

6.4 Example 3 of Flatness

In order to further compare the effectiveness of AHTLBO algorithm in flatness evaluation, the second correlation literature data are used to experiment [11]. The calculation results of TLBO and AHTLBO algorithm are compared with the relevant literatures [11, 18-19]. The measurement data can be seen in Table 7.

Table 7. The literature [11] measurement data mm

No	*X*	*Y*	*Z*	No	*X*	*Y*	*Z*
1	9.9995	5	4.0103	14	60.001	80.0019	4.0048
2	10.0002	29.9981	4.0138	15	59.9978	105.0005	4.0024
3	10.0015	55.0007	4.0009	16	84.9988	5.001	4.0126
4	9.9996	79.9988	4.0095	17	84.9981	30	4.0114
5	9.9997	104.9981	4.0161	18	85.0007	55.0008	4.0021
6	35.0003	4.9986	4.0029	19	85	80.0003	4.0054
7	35.0003	30.0012	4.0044	20	84.9988	105.002	4.0056
8	34.9974	54.9996	3.9987	21	109.9984	4.9981	4.0161
9	35.0013	80.0012	4.0025	22	110	29.9992	4.0119
10	34.9993	104.9982	4.0201	23	109.9986	55.0004	4.0122
11	59.9989	4.9982	4.005	24	109.998	79.9988	4.0115
12	60.0003	29.9987	4.01	25	110.0003	104.9984	4.0056
13	60.0013	54.9986	4.0025				

The calculation results are recorded as shown in Table 8, as is shown in literature [11, 18-19]. The flatness evaluation results of IGA, CG and OPPB are 0.1847463 mm, 0.01838007809 mm and 0.018382568 mm respectively. In this paper the calculation results of TLBO and AHTLBO algorithm are 0.018384 mm and 0.0183800780872 mm respectively, and the corresponding plane parameters of AHTLBO are 0.000027199200331 and 0.00006039909506 by AHTLBO, the AHTBLO has the highest accuracy compared to other algorithms in related literatures.

Table 8. Calculation results mm

Methods	*A*	*B*	Error
IGA [11]			0.01847463
CG [18]	0.0000271992003	0.0000603990949	0.01838007809
OPPB [19]			0.018382568
TLBO	0.000027806	0.000061378	0.018384
AHTLBO	0.000027199200331	0.00006039909506	0.0183800780872

Fig. 11 illustrates iterative curves of the flatness error of the AHTLBO and TLBO algorithms, it can be seen that in the iterative process, the AHTLBO algorithm converges at 42 iterations, and its iterative speed is faster than that of TLBO, so it has better computational performance than basic TLBO.

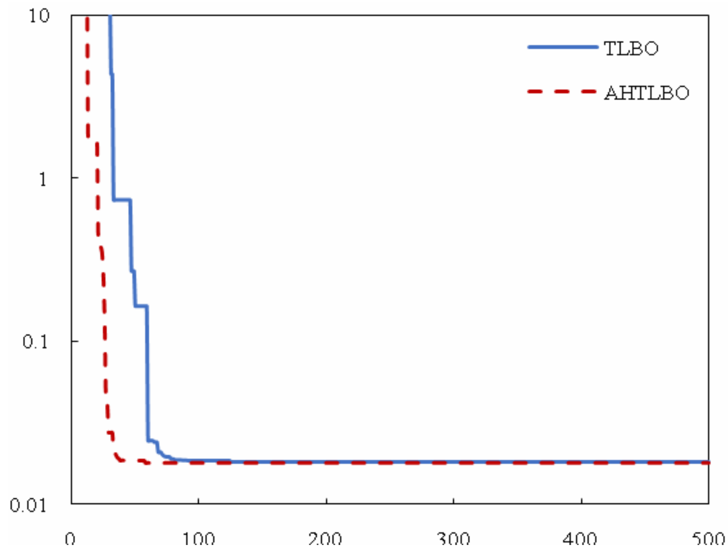

Fig. 11. Iteration curve of AHTLBO

7 Conclusion

Flatness error is one of the most important parts technical parameters. In order to improve the accuracy of flatness error evaluation under the minimum zone method, the teaching-learning based algorithm is applied to the evaluation process. For the basic teaching-learning-based algorithm, a hybrid teaching-learning based algorithm is proposed with the method of SFAL and an adaptive factor. The precision and convergence speed of the AHTLBO are improved compared to TLBO. The AHTLBO is applied to the evaluation of the flatness error under the minimum zone principle. The calculation results are improved compared to the LSM, and the algorithm also has higher precision and faster convergence in the flatness evaluation compared to the PSO, GA. It is very suitable for the evaluation of the flatness error. In addition, the shape error evaluation of parts is a systematic project. It needs the algorithm to realize the data analysis, and in order to obtain more accurate and reliable error information of the parts, the uncertainty theory should be introduced into the final evaluation result. In addition, how to estimate the shape error, conducting more extensive data analysis, and the transition to the error prediction of batch parts, also need further research.

Acknowledgements

This paper is supported by national high technology research and development program of china (No. 2015AA043003).

References

[1] ANSI/ASME Y14.5. National standard on dimensioning and tolerancing, 2009.

[2] ISO 1101-2012. Geometrical Product Specifications (GPS)-Tolerances of form, orientation, location and run-out, 2012.

[3] ISO/TS 12781-1-2012. Geometrical product specifications (GPS)-Flatness-Part 1: Vocabulary and parameters of flatness, 2012.

[4] M.-T. Traband, S. Joshi, R. Wysk, T.-M. Cavalier, Evaluation of straightness and flatness tolerance using the minimum zone, Manufacturing Review 2(3)(1989) 189-195.

[5] T. Kanada, S. Suzuki, Evaluation of minimum zone flatness by means of nonlinear optimization techniques and its verification, Precision Engineering 15(2)(1993) 93-99.

[6] K. Carr, P. Ferreira, Verification of form tolerances part I: Basic issues, flatness, and straightness, Precision Engineering 17(2)(1995) 131-143.

[7] S.-H. Cheraghi, H.-S. Lim, S. Motavalli, Straightness and flatness tolerance evaluation: an optimization approach, Precision Engineering 18(1)(1996) 30-37.

[8] M. Lee, A new convex-hull based approach to evaluating flatness tolerance, Computer Aided Design 29(12)(1997) 861-868.

[9] G.-L. Samuel, M.-S. Shunmugam, Evaluation of straightness and flatness error using computational geometric techniques, Computer-Aided Design 31(3)(1999) 829-843.

[10] X.-L. Wen, A.-G. Song, An improved genetic algorithm for planar and spatial straightness error evaluation, International Journal of Machine Tools & Manufacture 43(11)(2003) 1157-1162.

[11] X.-L. Wen, X.-C Zhu, Y.-B. Zhao, D.-X. Wang, F.-L. Wang, Flatness error evaluation and verification based on new generation geometrical product specification (GPS), Precision Engineering 36(1)(2012) 70-76.

[12] H.-Y. Tseng, A genetic algorithm for assessing flatness in automated manufacturing systems, Journal of Intelligent Manufacturing 17(3)(2006) 301-306.

[13] S. Cho, J.-Y. Kim, Straightness and flatness evaluation using data envelopment analysis, The International Journal of Advanced Manufacturing Technology 63(5)(2012) 731-740.

[14] J. Luo, Q. Wang, L. Fu, Application of modified artificial bee colony algorithm to flatness error evaluation, Optics and Precision Engineering 20(2)(2012) 422-430.

[15] G. Hermann, Application of computational geometry in coordinate measurement, in: Proc. 2015 IEEE Jubilee International Symposium on Applied Computational Intelligence and Informatics, 2015.

[16] X. Tu, X.-Q. Lei X, Evaluation reference and error function unification for shape error, in: Proc. 2015 International Conference on Material, Mechanical and Manufacturing Engineering, 2015.

[17] M.-I. Khan, S. Ma, New generation geometrical product specification (GPS) backed flatness error estimation and uncertainty Analysis, Open Mechanical Engineering Journal 10(1)(2016) 66-78.

[18] P. Li, X.-M. Ding, J.-B, Tan, J.-W. Cui, A hybrid method based on reduced constraint region and convex-hull edge for flatness error evaluation, Precision Engineering 45(2016) 168-175.

[19] V. Radlovački, M. Hadžistević, B. Štrbac, M. Delić, B. Kamberović, Evaluating minimum zone flatness error using new method-bundle of plains through one point, Precision Engineering 43(2016) 554-562.

[20] V.-K. Pathak, A.-K. Singh. Form error evaluation of noncontact scan data using constriction factor particle swarm optimization, Journal of Advanced Manufacturing Systems 16(3)(2017) 205-226.

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

[22] R.-V. Rao, K.-C. More, Advanced optimal tolerance design of machine elements using teaching-learning-based optimization algorithm, Production & Manufacturing Research 2(1)(2014) 71-94.

[23] G. Tuncel, D. Aydin. Two-sided assembly line balancing using teaching-learning based optimization algorithm, Computers & Industrial Engineering 74(1)(2014) 291-299.

[24] M.-M. Eusuff, K. E. Lansey, Optimization of water distribution network design using the shuffled frog leaping algorithm, Journal of Water Resources Planning & Management 129(3)(2003) 210-225.

[25] R.-V. Rao, V. Patel, An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems, Scientia Iranica 20(3)(2013) 710-720.