

Received November 3, 2019, accepted November 15, 2019, date of publication November 20, 2019, date of current version December 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2954500

Hybridizing Teaching-Learning-Based Optimization With Adaptive Grasshopper Optimization Algorithm for Abrupt Motion Tracking

HUANLONG ZHANG¹, ZENG GAO¹, XIAOYANG MA¹, JIE ZHANG¹,
AND JIANWEI ZHANG²

¹College of Electric and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

²Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou 450002, China

Corresponding author: Jianwei Zhang (ing@zzuli.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61873246, Grant 61703373, Grant 61672471, and Grant 61501407, and in part by the Graduate Innovation Foundation of Zhengzhou University of Light Industry under Grant 2018024.

ABSTRACT Aiming at the problem that conventional tracking algorithms are difficult to deal with abrupt motion efficiently, an optimization algorithm called hybrid Teaching-learning-based optimization with Adaptive Grasshopper Optimization Algorithm (TLGOA) is proposed in this paper. Firstly, the non-linear strategy based on tangent function is used to replace the linear mechanism in the standard Grasshopper Optimization Algorithm (GOA). The improved adaptive GOA (AGOA) can avoid the local trapping problem and enhance the global optimization ability, which can handle the problem of abrupt motion. Secondly, considering that Teaching-learning-based optimization (TLBO) has obviously local exploitation operator and fast convergence, a hybrid TLGOA tracker is designed by combining the advantages of both AGOA and TLBO. The approach can enable better tracking accuracy and efficiency. Finally, extensive experimental results show that the proposed algorithm has obvious advantages over other algorithms, and also prove that TLGOA tracker is very competitive compared to other state-of-the-art trackers, especially for abrupt motion tracking.

INDEX TERMS Visual tracking, abrupt motion, grasshopper optimization algorithm, teaching-learning-based optimization.

I. INTRODUCTION

Visual tracking is a fundamental problem in computer vision due to its important role in a wide range of applications such as video surveillance, human-computer interaction, medical imaging, to name a few. Although it has attracted increasing interests in recent decades and significant progress have been achieved, visual tracking remains a challenging task due to various factors such as partial occlusion, motion blur, illumination variation, abrupt motion, camera motion, and many more. To further improve tracking performance, many methods have been proposed. These methods are usually divided to two categories: the generative

method [1], [2] and the discriminative method [3], [4]. In recent years, numerous methods based on correlation filter [5], [6] and deep learning [7], [8] have been proposed, which promotes the rapid development of visual tracking.

Although achieving state-of-the-art performance, most of these methods assume that the target has the condition of smooth motion. However, the abrupt motion of targets often occur due to some reasons in real world scenarios, such as fast motion, camera switching, low-frame-rate, etc. In this case, the conventional tracker cannot track effectively. To address the issue, many methods have been proposed subsequently, including detection based tracking methods [9], [10], stochastic sampling based tracking methods [11], [12], optimization based tracking methods [13], [14], etc.

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

In the above methods, the detection based tracking methods need pre-training operation, and the stochastic sampling based tracking methods are to find the target by stochastic sampling rather than a certain trajectory in the search area. These methods will cause a lot of time consumption. Therefore, in order to obtain more intelligent search mechanism to solve the visual tracking, swarm optimization algorithms have attracted more researchers' attention because of its robustness and flexibility. These tracking methods have achieved good results. Zhang *et al.* [15] combined sequence information and multi-layer importance sampling strategy to propose a sequential Particle Swarm Optimization (PSO) based tracking framework, which was more robust and effective in arbitrary motion or large appearance changes. Nguyen and Bhanu [16] presented the modified Bacterial Foraging Optimization algorithm and utilized it for changing of real-time tracking. The method was better than tracker based on PSO and particle filter in speed and accuracy. Gao *et al.* [17] proposed bat algorithm to solve tracking problem, which outperformed the other trackers in accuracy and speed. Although these tracking methods work well, they adopt a single swarm optimization algorithm to solve visual tracking. As we all know, exploration and exploitation are two important components of swarm optimization algorithm. However, an optimization algorithm is difficult to balance exploration and exploitation to deal with visual tracking. No one can recommend a method to solve all problems according to no-free-lunch (NFL) theorem [18]. Therefore, this theorem allows researchers to propose new optimization algorithms or improve existing algorithms to solve a wider range of problems.

Hybridizing optimization algorithm is the latest research trend for overcoming the poor exploration ability of one algorithm and poor exploitation ability of the other algorithm. There are many hybrid optimization algorithms proposed such as Genetic Learning PSO (GL-PSO) [19], hybrid Gravitational Search Algorithm with Dynamic Multi swarm PSO (GSADMSPSO) [20], hybrid Whale Optimization Algorithm with Simulated Annealing (WOA-SA) [21], hybrid Bat Algorithm with Harmony Search (BHS) [22], hybrid Cuckoo Search(HCS) [23], Sine Cosine Water Wave Optimization (SCWWO) [24], etc. These literatures show that hybrid optimization algorithms are designed and utilized for complex optimization problems. Of course, these hybrid optimization algorithms also have some advantages for handling visual tracking. Ljouad *et al.* [25] proposed a new hybrid Kalman Cuckoo Search (CS) tracker, which utilized an improved CS algorithm combined with the Kalman Filter to enhance the initial population's quality. The method was superior to the PSO-based tracker in computational time. Chen *et al.* [26] introduced a Euclid distance based hybrid quantum PSO (HQPSO) and a fast moving object tracking based on HQPSO mean shift algorithm was also designed, which improved tracking efficiency and decreased detection time cost. Nenavath *et al.* [27], [28] a proposed hybrid sine-cosine algorithm with PSO (SCA-PSO) and SCA with Differential

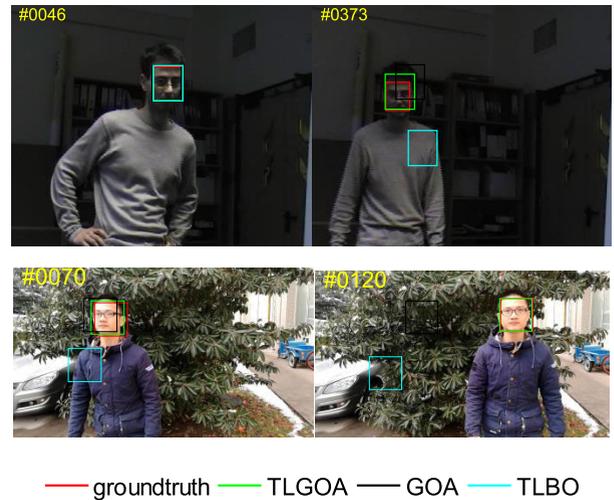


FIGURE 1. The abrupt motion tracking results.

Evolution (SCA-DE) for global optimization and visual tracking. These hybrid optimization algorithms can effectively balance exploration and exploitation for dealing with visual tracking. In addition, we have been focusing on hybrid algorithm to solve abrupt motion tracking [13], [14], [29]. Zhang *et al.* [29] proposed an extended kernel correlation filter (KCF) tracker based on simulated annealing method and designed a unified framework to track smooth or abrupt motion simultaneously. Subsequently, Zhang *et al.* [14] presented an extended CS (ECS) based KCF for abrupt motion tracking. This method utilized ECS to evaluate motion states and designed a model switch to obtain smooth and abrupt motion, which improved the real-time and accuracy of target tracking. In these literatures, the performance of the hybrid algorithm is better than that of the single algorithm.

Recently, Saremi *et al.* [30] and Rao *et al.* [31]–[33] proposed GOA and TLBO, respectively. These two algorithms had been applied in lots of fields successfully [34]–[39] and some improved methods were proposed [40]–[43]. Therefore, a novel optimization algorithm called hybrid TLGOA is proposed in the paper, for solving abrupt motion tracking. The approach leverages the complementary properties between exploratory stage of AGOA and the exploitation operator of TLBO to improve both accuracy and efficiency. As shown Fig. 1, the tracked object undergoes abrupt motion between the consecutive image frames.

The main contribution of our work includes three folds:

(1) A novel optimization algorithm called AGOA is proposed on the basis of GOA, which has stronger searching ability and can avoid local optimum effectively. The AGOA can adapt the problem of abrupt motion.

(2) A novel hybrid TLGOA was proposed based on AGOA with good exploratory stage and obvious exploitation operator of TLBO. 5 benchmark functions and 30 CEC 2014 benchmark problems are conducted to verify the effectiveness of the proposed algorithm.

(3) The hybrid TLGOA is applied to visual tracking, which can improve both robustness and efficiency. The extensive experimental results show that the proposed tracker has much stronger merit compared to GOA tracker, TLBO tracker and 6 state-of-the-art trackers, especially for abrupt motion tracking.

II. TEACHING-LEARNING-BASED OPTIMIZATION AND GRASSHOPPER OPTIMIZATION ALGORITHM

A. TEACHING-LEARNING-BASED OPTIMIZATION (TLBO)

TLBO is a population-based optimization algorithm and it simulates the teaching-learning process in the class [31]–[33]. In this paper, only the two main phases of the algorithm are considered (i.e., ‘teacher phase’ and ‘learner phase’). In the teacher phase, the teacher distributes his knowledge to all learners. In learner phase, a learner knows about knowledge from other learners.

1) TEACHER PHASE

It is first part of the algorithm where the learner with the highest marks acts as a teacher, and the task of the teacher is to increase the mean marks of the class. The update process of i -th learner in teacher phase is formulated as:

$$X_{i,new} = X_i + random \times (X_{teacher} - T_F \times X_{ave}) \quad (1)$$

where X_i is the solution (mark) of the i -th learner, $X_{teacher}$ represents the teacher’s solution (mark), X_{ave} means the average(mark) of all learners, $random$ is a random number in $[0,1]$, and T_F is the teaching factor that decides the value of mean to be changed. The value of T_F can be either 1 or 2 (1 corresponds to no increase in the knowledge level and 2 corresponds to complete transfer of knowledge), which is again a heuristic step and decided randomly with equal probability as $T_F = round[1 + rand(0, 1)\{2 - 1\}]$. In addition, the new solution $X_{i,new}$ is accepted only if it is better than the previous solution, it can be formulated as:

$$X_i = \begin{cases} X_{i,new}; & f(X_{i,new}) > f(X_i) \\ X_i; & otherwise \end{cases} \quad (2)$$

where f represents the fitness function.

2) LEARNER PHASE

It is second part of the algorithm where the learner updates its knowledge through the interaction with other learners. In each iteration, two learners interact with X_m and X_n , in which the smarter learner improves the marks of other learners. In learner phase, a learner learns new things if the other learner has more knowledge than him. The phenomenon is described as follows:

$$X_{m,new} = \begin{cases} X_m + random \times (X_m - X_n); & f(X_m) > f(X_n) \\ X_m + random \times (X_n - X_m); & f(X_m) \leq f(X_n) \end{cases} \quad (3)$$

The temporary solution is accepted only if it is better than the previous solution, it can be formulated as:

$$X_m = \begin{cases} X_{m,new}; & f(X_{m,new}) > f(X_m) \\ X_m; & otherwise \end{cases} \quad (4)$$

Compared with other optimization algorithms, TLBO does not require any parameters to be tuned, thus making the implementation of TLBO simpler. In addition, TLBO exhibits obvious convergence in [31]–[33].

B. GRASSHOPPER OPTIMIZATION ALGORITHM (GOA)

The GOA is a new population-based optimization algorithm that it simulates behavior of grasshopper swarms in nature [30]. The following subsections discuss the mathematical model in details.

1) INITIAL STAGE

To simulate the swarming behavior of grasshopper in this stage, the following mathematical model can be written as:

$$P_p = S_p + G_p + A_p \quad (5)$$

where P_p is the position of the p -th grasshopper, S_p and G_p show the social interaction and the gravity force on the p -th grasshopper, and A_p indicates the wind advection. The value of S_p , G_p and A_p in Eq. (5) are calculated as follows:

$$\begin{cases} S_p = \sum_{q=1 \neq p}^N s(d_{pq}) \widehat{d}_{pq} \\ G_p = -g \widehat{e}_g \\ A_p = u \widehat{e}_w \end{cases} \quad (6)$$

where $d_{pq} = |P_q - P_p|$ is the distance between the p -th grasshopper and the q -th grasshopper. $\widehat{d}_{pq} = (P_q - P_p) / d_{pq}$ is a unit vector from the p -th grasshopper to the q -th grasshopper, g is the gravitational constant, \widehat{e}_g indicates a unity vector towards the center of earth, u is a constant drift, and \widehat{e}_w is a unity vector in the direction of wind. $s(\cdot)$ is a function to define the strength of social forces, it is obtained as follows:

$$s(r) = f e^{-lr} - e^{-r} \quad (7)$$

where f indicates the intensity of attraction, and l is the attractive length scale.

2) UPDATING STAGE

In this stage, Saremi *et al.* [30] proposed the following mathematical model search while grasshoppers are interacting. Substituting S_p , G_p and A_p in Eq. (5), this equation can be expanded as follows:

$$P_p = \sum_{q=1 \neq p}^N s(|P_q - P_p|) \frac{P_q - P_p}{d_{pq}} - g \widehat{e}_g + u \widehat{e}_w \quad (8)$$

Since the grasshoppers quickly reach the comfort zone and the swarm does not converge to a specified point, Eq. (8)

cannot be used directly to solve optimization problems. The resultant modified equation can be expressed as:

$$P_p^d = c \left(\sum_{q=1 \neq p}^N c \frac{ub_d - lb_d}{2} s(|P_q^d - P_p^d|) \frac{P_q - P_p}{d_{pq}} \right) + \hat{T}_d \quad (9)$$

where P_p^d is value of the $d - th$ dimension in the $p - th$ grasshopper, ub_d and lb_d are the upper and lower bound in the $d - th$ dimension, \hat{T}_d is the value of the $d - th$ dimension in the target (best solution found so far). The parameter c is a decreasing coefficient to reduce the range of comfort area, exclusion area and attraction area.

3) SHRINKING FACTOR

For balancing exploration and exploitation, the parameter c is linearly decreased from 1 to 0 over the course of iterations and is calculated as follows:

$$c = c_{max} - \frac{t(c_{max} - c_{min})}{T} \quad (10)$$

where c_{max} and c_{min} are the maximum value and the minimum value, respectively. t is the current iteration, and T is the maximum number of iterations.

III. THE PROPOSED HYBRID TLGOA METHOD

A. THE ADAPTIVE GOA (AGOA)

1) MOTIVATION OF AGOA

In the basic GOA search process, the shrinking factor c is utilized to control the motion step of grasshoppers. This means that once the algorithm falls into local optimum in the early stage of search, it will be difficult to jump out of local optimum and achieve global optimum in the later stage. Therefore, the parameter c should be considered for improvement.

2) IMPLEMENTATION OF AGOA

Based on the above problems, the searching process of GOA is very complicated. The searching strategy of parameter c , decreasing linearly with the number of iterations, cannot fully reflect the real searching process. Therefore, the non-linear strategy is employed to replace the linear mechanism representing the variation of exploring step in the basic GOA. In addition, the location update scheme of GOA is similar to PSO. Therefore, this paper is inspired by the inertia weight in PSO algorithm. The parameter c is calculated as follows:

$$c = c_{max} - (c_{max} - c_{min}) \tan \left(k \frac{t}{T} \pi \right) \quad (11)$$

where k is a constant. In this section, we compare the improved parameter c with the original parameter c , and analyze the value of the constant k as shown in Fig. 2.

We first observe the improved parameter c . For $k = 0.3$, the convergence rate of the improved parameter c and the original parameter c are very similar in the early stage of the algorithm, but the exploration rate converges to 0 in advance when the number of iterations is 250. In the later stage of the

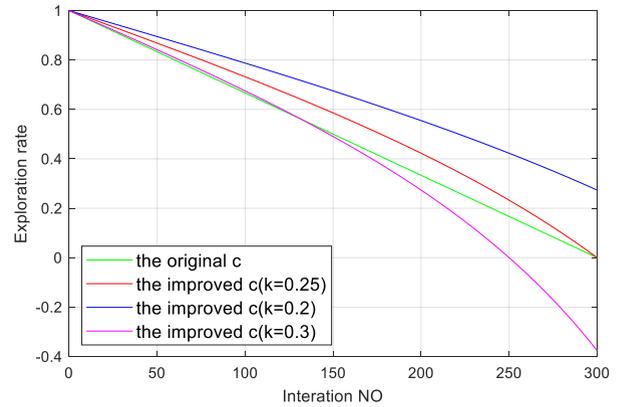


FIGURE 2. Exploration rate with the increase of iterations.

algorithm, the search rate increases gradually so that the algorithm cannot converge to the global optimum. For $k = 0.2$, although the exploration performance is very good at the early stage, the exploration rate is still very high at the later stage of the algorithm, which cannot guarantee the exploitation performance of the algorithm. Then we compare the improved parameter c ($k = 0.25$) and the original parameter c . For $k = 0.25$, the decay degree of the improved parameter c is very slow in the early stage of the algorithm in order to search for the global optimal solution. However, in the later stage of the algorithm, the attenuation degree of the improved parameter c increases to find the local optimal solution more accurately. In a word, the nonlinear parameter c ($k = 0.25$) has better performance of exploration and exploitation than the original parameter c . Therefore, the constant k is set to be 0.25.

A new optimization algorithm which we call as AGOA, is generated by replacing Eq. (10) in basic GOA with Eq. (11), to control the step size of grasshoppers movement, and ultimately achieves global optimization.

B. HYBRID TLGOA

1) PHILOSOPHY

Balancing exploration and exploitation is crucial since AGOA and TLBO belong to population-based optimization algorithms. The purpose of global exploration is to find a more promising area in whole search area. Local exploitation is to find a better solution in some local areas based on previous knowledge or new information during the search process.

Although AGOA adopts non-linear strategy on c to improve the global search performance and avoid the algorithm falling into local optimum, the same update scheme (random walk) is utilized for exploration and exploitation. This means that AGOA can always find the optimal solution under sufficient evaluation times, but this will not guarantee fast convergence. However, TLBO does not require any parameters to be set and has obvious local exploitation performance. Therefore, a hybrid TLGOA is proposed.

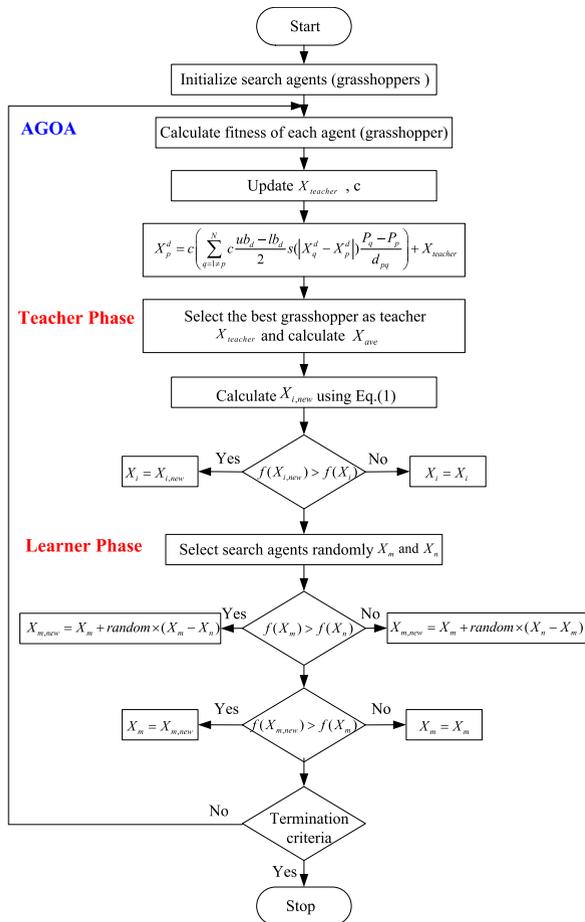


FIGURE 3. Flow chart of hybrid TLGOA.

In hybrid TLGOA, we utilize AGOA and TLBO to compensate for each other’s deficiencies, where the AGOA is mainly utilized for exploring the search space, and the TLBO has obviously local searching ability. In the beginning stage, the AGOA can perform more performance for exploring the whole regions and making most solution move toward a more promising area. In the later stage, the TLBO will obtain more opportunity for exploiting high-precision solution. As a word, the hybrid method can deal with the conflict global and local search effectively. Moreover, the flow chart of hybrid TLGOA is shown in Fig. 3.

2) COMPUTATIONAL COMPLEXITY

The computational complexity of the proposed hybrid TLGOA mainly depends on TLBO and GOA. However, the computational complexity of basic optimization algorithms is mainly determined by population size (n), the maximum number of iterations (T) and dimensions of search space (D). The time complexity of the basic GOA is $O(n \cdot T \cdot D)$, and the fitness ranking of individuals costs $O(n \cdot \log n)$ computations. For non-linear adaptive parameters, the computational complexity is almost the same as that of linearity (i.e., $O(AGO A) \approx O(GOA)$). Since TLBO

includes two stages of teaching and learning, time complexity is $O(2n \cdot T \cdot D)$. However, in hybrid TLGOA, the AGOA is utilized to global exploration to make most solution move toward a more promising area. After exploration stage, TLBO algorithm is utilized to local exploitation to get the optimal value. Therefore, the time complexity of the proposed method is $O(TLGOA) = O(3n \cdot T \cdot D)$.

IV. PERFORMANCE EVALUATION OF HYBRID TLGOA

In order to verify the efficiency of the proposed algorithm, 5 benchmark functions and 30 CEC 2014 benchmark problems are conducted. These 5 benchmark functions employed by many researchers [30], [41]–[43]. All the experiments are performed on the same machine using MATLAB 2017b.

A. EXPERIMENTAL SETUP AND PARAMETER SETTINGS

In order to verify the performance of the proposed algorithm, 5 optimization algorithms were adopted to compare, namely AGOA, GOA [30], Modified TLBO (MTLBO) [42], TLBO [32] and Differential Evolution (DE) [44], respectively. DE is a classical evolutionary algorithm, which has been studied by many researchers [45], [46]. The detailed information regarding 5 benchmark functions used is provided in [30]. The 30 CEC2014 benchmark problems are detailed in [47].

The hybrid TLGOA and other compared algorithms have been simulated on each 5 benchmark functions for comparison. For DE, the convergence time and dimension of the problem should be considered while choosing the value of population size (n). Large n values may affect the calculation time. However, low n may lead to premature convergence and local optimum. According to [44], [48], [49], the reasonable value of population size should be chosen to be in between $5D$ and $8D$ (D is the dimension variable. In this experiment, population size $n = 5D, n = 8D$). For other algorithms, the population size was set to 50. The maximum iterations are set as 1000 for all these algorithms.

Similarly, to illustrate the effectiveness of the proposed algorithm, it was tested on 30 CEC 2014 benchmark problems with $D = 10$. For DE, population size should be chosen to be in between $5D$ and $8D$ (in this experiment, population size $n = 50, n = 80$). For other algorithms, the population size was set to 50. For CEC 2014, the termination criteria is set to be the maximum function evaluation (FE, i.e., $5000D$).

Other initial parameters for 6 optimization algorithms are set as follows:

- GOA [30]: $l = 1.5; f = 0.5; c \in (0, 1)$.
- AGOA: $l = 1.5; f = 0.5; c \in (0, 1); k = 0.25$.
- TLBO [32]: The value of TF can be either 1 or 2.
- TLGOA: The parameters of hybrid TLGOA are consistent with those of AGOA and TLBO.
- MTLBO [42]: The parameters of MTLBO are consistent with TLBO.
- DE [48]: The scaling factor is set at 0.5; The crossover constant is set at 0.9.

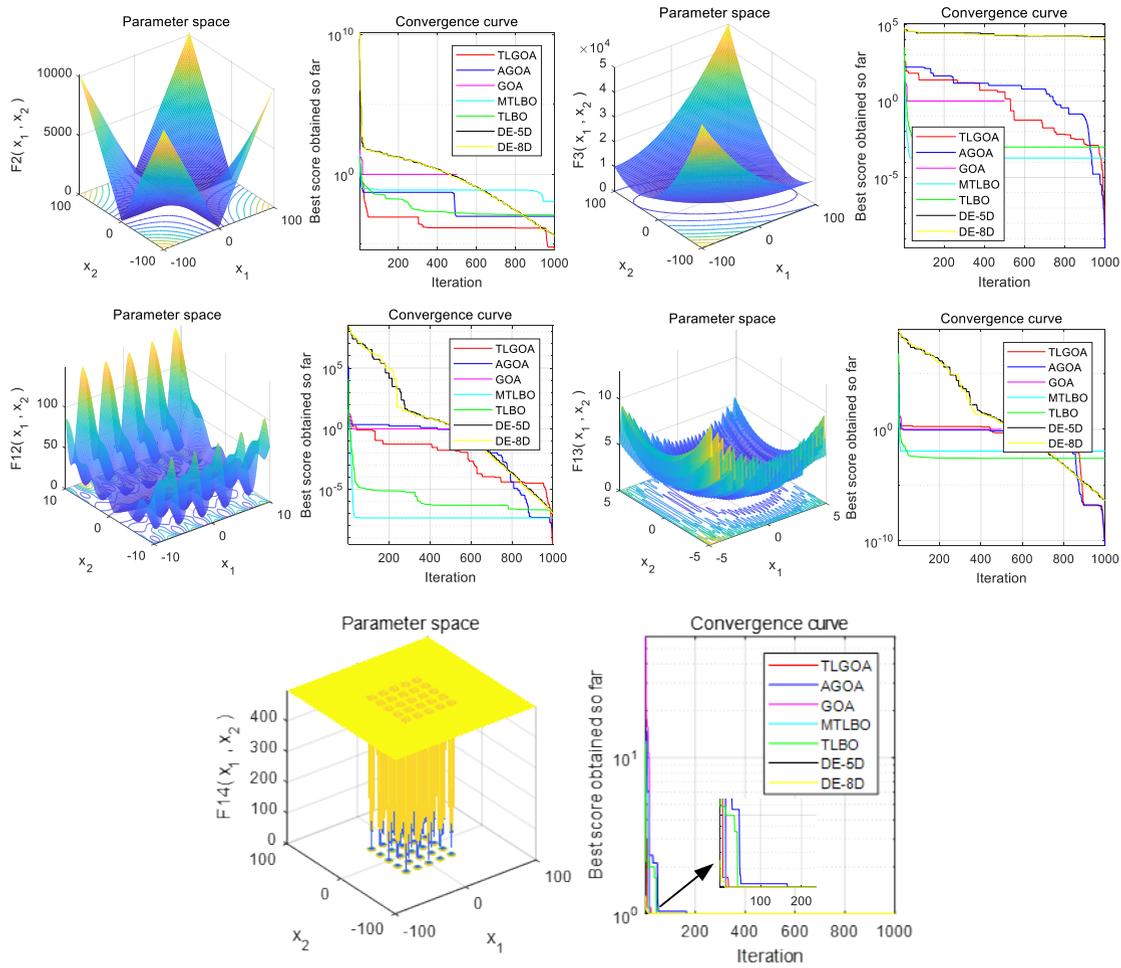


FIGURE 4. Convergence curves of five functions.

TABLE 1. Five functions used in the experimental studies (D: Dimension, F_{min} : Global minima).

| Function | Name of the function | D | Range | F_{min} |
|----------|----------------------|----|------------------|-----------|
| F_2 | Schwefel 2.22 | 30 | [-10,10] | 0 |
| F_3 | Schwefel 1.2 | 30 | [-100,100] | 0 |
| F_{12} | Penalized | 30 | [-50,50] | 0 |
| F_{13} | Penalize 2 | 30 | [-50,50] | 0 |
| F_{14} | Foxholes | 2 | [-65.536,65.536] | 0.998004 |

In addition, in order to compare the results of each run and make the results statistically significant, Wilcoxon’s rank-sum test [50], a nonparametric statistical test, was carried out at 5% significance level. Considering that the best algorithm cannot be compared with itself, *N/A* is written for each function to represent the best algorithm that does Not Applicable.

B. EXPERIMENTS FOR 5 BENCHMARK FUNCTIONS

Hybrid TLGOA and other five algorithms were tested and compared 30 times for each of 5 functions (listed in Table 1). The experimental results are represented by the statistical data of the best solution of the last iteration, such as average,

standard deviation and worst. The results of the algorithms on all test functions are presented in Table 2. The best outcomes are marked in bold type in tables. As can be seen from Table 2, the proposed algorithm has certain advantages. It is also clear from Table 3 that this advantage is statistically significant since the P value is far less than 0.05.

Convergence behavior analysis

Convergence rate is used to verify the convergence of hybrid TLGOA algorithm. The convergence rate is to preserve the fitness of the best solution for each iteration and draw the convergence curve as shown in Fig. 4. Fig. 4 shows the convergence curves of hybrid TLGOA, AGOA, GOA, MTLBO, TLBO and DE for the five functions. For these test functions, the convergence curve of the hybrid TLGOA has an obviously downward trend. This strongly proves that the hybrid TLGOA has a better ability to converge to the global optimum in the process of iteration.

C. EXPERIMENTS FOR FUNCTIONS IN CEC2014

To further illustrate the effectiveness of the hybrid TLGOA, it was tested on 30 benchmark functions from the CEC

TABLE 2. Comparison of optimization results obtained for five functions (ave: average, std: standard deviation). Wilcoxon's rank-sum test at a 0.05 significance level is performed between TLGOA and other algorithms (listed in Table 3). "+", "-", "≈" denote that the performance of the corresponding algorithm is statistically better than, worse than, and similar to that of TLGOA, respectively.

| Function | | TLGOA | AGOA | GOA | MTLBO | TLBO | DE-5D | DE-8D |
|----------|-------|-----------------|----------|----------|----------|----------|---------------|---------------|
| F_2 | ave | 3.66E-05 | 3.20E-04 | 1.30E-04 | 0.0211 | 0.0091 | 4.01E-05 | 4.28E-05 |
| | std | 5.91E-06 | 8.58E-04 | 3.81E-04 | 0.0363 | 0.0169 | 5.45E-06 | 4.74E-06 |
| | worst | 4.88E-05 | 0.0037 | 0.0014 | 0.1818 | 0.0836 | 5.38E-05 | 5.43E-05 |
| F_3 | ave | 4.31E-09 | 2.40E-08 | 7.71E-08 | 0.0049 | 0.0147 | 1.53E+04 | 1.46E+04 |
| | std | 5.02E-09 | 4.09E-08 | 9.53E-08 | 0.0104 | 0.0420 | 1.95E+03 | 1.73E+03 |
| | worst | 1.94E-08 | 1.90E-07 | 4.17E-07 | 0.0394 | 0.2115 | 1.94E+04 | 1.83E+04 |
| F_{12} | ave | 1.68E-09 | 7.17E-09 | 6.02E-09 | 5.22E-07 | 8.35E-07 | 1.78E-07 | 1.78E-07 |
| | std | 3.18E-09 | 9.30E-09 | 1.96E-08 | 7.53E-07 | 3.30E-06 | 4.10E-08 | 6.09E-08 |
| | worst | 1.33E-08 | 3.89E-08 | 1.06E-07 | 2.93E-06 | 1.82E-05 | 2.65E-07 | 4.21E-07 |
| F_{13} | ave | 1.24E-09 | 2.87E-09 | 3.25E-08 | 0.0185 | 0.0103 | 4.76E-07 | 4.61E-07 |
| | std | 2.55E-09 | 7.75E-09 | 7.50E-08 | 0.0309 | 0.0118 | 1.40E-07 | 8.23E-08 |
| | worst | 1.32E-08 | 4.23E-08 | 3.40E-07 | 0.1226 | 0.0444 | 7.10E-07 | 5.97E-07 |
| F_{14} | ave | 0.9980 | 0.9980 | 0.9980 | 1.5258 | 0.9980 | 0.9980 | 0.9980 |
| | std | 0 | 3.94E-16 | 4.11E-16 | 1.2094 | 1.01E-16 | 0 | 0 |
| | worst | 0.9980 | 0.9980 | 0.9980 | 5.9288 | 0.9980 | 0.9980 | 0.9980 |
| +/-/≈ | | 0/5/0 | 0/5/0 | 0/5/0 | 0/5/0 | 0/4/1 | 0/4/1 | |

TABLE 3. P-values of the Wilcoxon ranksum test over five functions.

| | TLGOA | AGOA | GOA | MTLBO | TLBO | DE-5D | DE-8D |
|----------|-------|----------|----------|----------|----------|----------|----------|
| F_2 | N/A | 0.0271 | 2.00E-06 | 3.02E-11 | 1.11E-06 | 0.0261 | 1.49E-04 |
| F_3 | N/A | 0.0044 | 5.54E-08 | 3.02E-11 | 9.92E-11 | 3.02E-11 | 3.02E-11 |
| F_{12} | N/A | 0.0013 | 0.0281 | 4.94E-05 | 1.55E-09 | 3.02E-11 | 3.02E-11 |
| F_{13} | N/A | 0.0392 | 6.06E-07 | 3.02E-11 | 4.98E-11 | 3.34E-11 | 3.02E-11 |
| F_{14} | N/A | 6.67E-13 | 6.29E-13 | 2.15E-06 | 0.0109 | N/A | N/A |

2014 with $D = 10$. For the CEC2014 test function, it is not listed here due to space constrain. The detail description of these function in [47]. It was also compared between TLGOA and AGOA, GOA, MTLBO, TLBO, DE-5D (population size 5D), DE-8D (population size 8D). Table 4 lists the statistical results of the average, standard deviation of CEC2014 functions and the optimal solutions of different algorithms. All the experimental results are based on 51 runs. The best result is shown in bold.

According to Wilcoxon's rank-sum test it is clear that TLGOA performs best in terms of average values for functions C01, C04, C05, C06, C07, C09, C11, C17, C25, C28, C29 and C30. The DE-8D performs best in terms of average values for functions C02, C03 and C18. At the same time, DE-5D and DE-8D are the first parallel algorithms in terms of average values for functions C08, C10, C21 and C22 compared with other algorithms. Note that all algorithms perform best in terms of the average values for functions C12, C13, C16 and C26. For functions C14, C15 and C19, TLBO and MTLBO have the biggest mean among other algorithms. For function C27, TLGOA and DE-8D outperform the other algorithms. Considering the best solutions, TLGOA, DE-5D and DE-8D outperform the other algorithms for functions C20, C23, C24. In addition, it is clear from the last line of Table 4 that our algorithm is very competitive compared to other algorithms on the whole.

Considering the above, it can be concluded that hybrid TLGOA can make an appropriate trade-off between

exploration and exploitation and solve complex optimization problems.

V. THE PROPOSED TRACKING METHOD

Suppose there is target (food) in the image being searched. And a group of target candidates (grasshoppers) are randomly generated in the image (state space). The purpose of TLGOA tracker is to find the "best" candidate using the proposed hybrid algorithm. Based on this, the flowchart of proposed hybrid TLGOA tracker is shown in Fig. 5.

It can be seen from Fig. 5 that TLGOA tracker was executed in the t frame of DEER video sequence. We first manually mark the target image of the first frame in the video sequence as a template. Secondly, target candidates are generated using AGOA and TLBO in the whole search space (image) and target candidate area respectively. Then the features of candidate image and template image are extracted. Finally, visual tracking is implemented by similarity measure function.

A. THE FITNESS FUNCTION

For hybrid TLGOA method, we need to design a fitness function. The fitness function measures the similarity between the target and target candidate. In this paper, we utilize correlation coefficient to measure the similarity between target and target candidate. As we all known, the Histogram of Oriented Gradient (HOG) feature can capture edge or gradient structure well, and has the invariance to local geometric and photometric transformations. Therefore, their similarity is computed as:

$$\rho(X, Y) = \frac{cov(X, Y)}{\sqrt{D(X)}\sqrt{D(Y)}} \quad (12)$$

where $D(\cdot)$ denotes the variance and $Cov(\cdot)$ denotes covariance. X and Y are the HOG feature of the target and target candidate respectively. The fitness function is introduced as

TABLE 4. Comparison of optimization results obtained for CEC2014. (ave: average, std: standard deviation). Wilcoxon’s rank-sum test at a 0.05 significance level is performed between TLGOA and other algorithms. “+”, “-”, “≈” denote that the performance of the corresponding algorithm is statistically better than, worse than, and similar to that of TLGOA, respectively.

| Function | | TLGOA | AGOA | GOA | MTLBO | TLBO | DE-5D | DE-8D |
|----------|-----|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| C01 | ave | 4.21E+05 | 5.17E+06 | 6.39E+06 | 3.26E+07 | 1.86E+07 | 8.70E+05 | 7.23E+05 |
| | std | 2.19E+05 | 9.10E+06 | 1.39E+07 | 2.37E+07 | 1.99E+07 | 2.76E+03 | 2.68E+03 |
| C02 | ave | 7.53E+05 | 2.19E+07 | 6.45E+03 | 2.08E+09 | 1.37E+09 | 200.0025 | 200.0016 |
| | std | 3.17E+05 | 1.56E+08 | 4.95E+03 | 1.24E+09 | 1.36E+09 | 0.0031 | 0.0017 |
| C03 | ave | 301.5343 | 1.64E+04 | 1.90E+04 | 1.42E+04 | 1.18E+04 | 300.0028 | 300.0023 |
| | std | 0.4532 | 1.61E+04 | 2.21E+04 | 6.08E+03 | 6.81E+03 | 0.0026 | 0.0021 |
| C04 | ave | 401.0231 | 444.7856 | 429.5833 | 688.1673 | 547.3109 | 401.6667 | 401.3700 |
| | std | 4.4536 | 58.71 | 18.6706 | 311.9474 | 148.6945 | 1.2171 | 0.8400 |
| C05 | ave | 515.0157 | 520.1048 | 520.0797 | 520.1547 | 520.1158 | 518.7432 | 518.4764 |
| | std | 0.0609 | 0.0891 | 0.0849 | 0.0826 | 0.5209 | 2.6872 | 2.8655 |
| C06 | ave | 600.0513 | 603.3778 | 603.2124 | 607.8566 | 606.0210 | 600.0544 | 600.0595 |
| | std | 2.2322 | 1.9179 | 2.2045 | 1.4328 | 1.3446 | 0.0714 | 0.0506 |
| C07 | ave | 700.0631 | 702.0037 | 701.73487 | 758.0630 | 736.6787 | 700.1117 | 700.9988 |
| | std | 0.1174 | 3.2147 | 2.0709 | 31.9292 | 25.7145 | 0.0260 | 0.0250 |
| C08 | ave | 814.6807 | 828.3898 | 830.5214 | 849.3356 | 835.3225 | 800 | 800 |
| | std | 6.0318 | 14.5040 | 12.7864 | 12.4324 | 10.0532 | 0 | 0 |
| C09 | ave | 904.5752 | 930.1229 | 929.4819 | 949.0630 | 931.8619 | 907.7367 | 907.5863 |
| | std | 3.2748 | 13.0740 | 14.8610 | 12.7341 | 8.2956 | 1.3986 | 1.3856 |
| C10 | ave | 1.55E+03 | 1.96E+03 | 1.99E+03 | 2.16E+03 | 1.93E+03 | 1.00E+03 | 1.00E+03 |
| | std | 296.1871 | 390.6395 | 372.8334 | 260.2960 | 278.1183 | 0.0401 | 0.0087 |
| C11 | ave | 1.41E+03 | 2.01E+03 | 2.13E+03 | 2.32E+03 | 2.11E+03 | 1.45E+03 | 1.44E+03 |
| | std | 165.4170 | 346.1199 | 315.6753 | 263.5104 | 245.6470 | 109.8317 | 79.2825 |
| C12 | ave | 1.20E+03 |
| | std | 0.2965 | 0.2225 | 0.3229 | 0.2440 | 0.1924 | 0.0782 | 0.0571 |
| C13 | ave | 1.30E+03 |
| | std | 0.0753 | 0.0855 | 0.0911 | 1.0863 | 0.9379 | 0.0333 | 0.0309 |
| C14 | ave | 1.40E+03 | 1.40E+03 | 1.40E+03 | 1.41E+03 | 1.41E+03 | 1.40E+03 | 1.40E+03 |
| | std | 0.1727 | 0.3129 | 0.3206 | 8.1613 | 7.0475 | 0.0380 | 0.0283 |
| C15 | ave | 1.50E+03 | 1.50E+03 | 1.50E+03 | 2.27E+03 | 1.59E+03 | 1.50E+03 | 1.50E+03 |
| | std | 0.5109 | 0.7519 | 6.7850 | 2.57E+03 | 148.3307 | 0.2230 | 0.1993 |
| C16 | ave | 1.60E+03 |
| | std | 0.6616 | 0.5704 | 0.7269 | 0.3413 | 0.2908 | 0.2816 | 0.2208 |
| C17 | ave | 3.88E+03 | 1.83E+04 | 9.29E+03 | 5.61E+04 | 4.84E+03 | 4.63E+03 | 3.89E+03 |
| | std | 7.41E+03 | 4.26E+04 | 1.24E+04 | 9.93E+04 | 4.85E+03 | 1.86E+03 | 1.34E+03 |
| C18 | ave | 2.22E+03 | 1.24E+04 | 1.49E+04 | 1.09E+04 | 1.95E+03 | 1.81E+03 | 1.80E+03 |
| | std | 2.46E+03 | 1.01E+04 | 1.33E+04 | 9.16E+03 | 81.7944 | 1.6393 | 1.4406 |
| C19 | ave | 1.90E+03 | 1.90E+03 | 1.90E+03 | 1.91E+03 | 1.91E+03 | 1.90E+03 | 1.90E+03 |
| | std | 1.5180 | 1.6673 | 2.0714 | 2.4363 | 1.9004 | 0.0857 | 0.0834 |
| C20 | ave | 2.00E+03 | 1.20E+04 | 1.81E+04 | 8.64E+03 | 3.29E+03 | 2.00E+03 | 2.00E+03 |
| | std | 49.3835 | 3.79E+04 | 3.71E+04 | 1.08E+04 | 3.09E+03 | 0.1558 | 0.1452 |
| C21 | ave | 2.45E+03 | 9.57E+03 | 8.51E+03 | 1.15E+04 | 2.85E+03 | 2.10E+03 | 2.10E+03 |
| | std | 292.1667 | 1.17E+04 | 1.07E+04 | 9.23E+03 | 440.7851 | 1.7277 | 1.1327 |
| C22 | ave | 2.28E+03 | 2.37E+03 | 2.35E+03 | 2.34E+03 | 2.26E+03 | 2.20E+03 | 2.20E+03 |
| | std | 73.6405 | 110.9569 | 112.9710 | 89.4415 | 46.7791 | 0.0760 | 0.0095 |
| C23 | ave | 2.63E+03 | 2.64E+03 | 2.64E+03 | 2.71E+03 | 2.71E+03 | 2.63E+03 | 2.63E+03 |
| | std | 0.0608 | 13.9695 | 22.2166 | 5.1354 | 4.6835 | 9.19E-13 | 12.9110 |
| C24 | ave | 2.52E+03 | 2.53E+03 | 2.53E+03 | 2.58E+03 | 2.56E+03 | 2.52E+03 | 2.52E+03 |
| | std | 6.6114 | 12.0299 | 12.5225 | 20.3253 | 27.0113 | 2.3388 | 1.9453 |
| C25 | ave | 2.63E+03 | 2.68E+03 | 2.69E+08 | 2.70E+03 | 2.69E+03 | 2.64E+03 | 2.64E+03 |
| | std | 7.1554 | 30.8946 | 21.6764 | 9.6997 | 15.1421 | 7.2481 | 7.1899 |
| C26 | ave | 2.70E+03 |
| | std | 0.0798 | 0.0678 | 13.9853 | 1.1836 | 0.9402 | 0.0326 | 0.0337 |
| C27 | ave | 2.71E+03 | 3.05E+03 | 3.09E+03 | 2.96E+03 | 2.93E+03 | 2.74E+03 | 2.71E+03 |
| | std | 114.6481 | 149.7156 | 125.2119 | 88.3726 | 78.5861 | 88.2891 | 44.6296 |
| C28 | ave | 3.05E+03 | 3.32E+03 | 3.32E+03 | 3.13E+03 | 3.20E+03 | 3.17E+03 | 3.17E+03 |
| | std | 102.6226 | 158.9403 | 120.2374 | 92.1039 | 77.2239 | 10.5456 | 6.8280 |
| C29 | ave | 3.21E+03 | 6.18E+05 | 5.17E+05 | 5.84E+03 | 3.50E+03 | 3.25E+03 | 3.23E+03 |
| | std | 8.88E+03 | 1.11E+06 | 9.00E+05 | 7.62E+03 | 529.5587 | 54.1758 | 46.3620 |
| C30 | ave | 3.92E+03 | 4.20E+03 | 4.18E+03 | 5.76E+03 | 4.41E+03 | 3.93E+03 | 4.41E+03 |
| | std | 697.3231 | 435.9482 | 510.0619 | 1.51E+03 | 821.0216 | 238.1817 | 821.0216 |
| +/-/≈ | | | 0/23/7 | 0/23/7 | 0/26/4 | 2/24/4 | 7/13/10 | 7/12/11 |

follows:

$$E = 2 + 2 * \rho(X, Y) \tag{13}$$

The fitness values affect how new solutions are generated.

B. THE TRACKING IMPLEMENT

In the TLGOA tracker, we first locate the target object in the first frame manually. Secondly, target candidates are generated by TLGOA. Then features of target candidate and

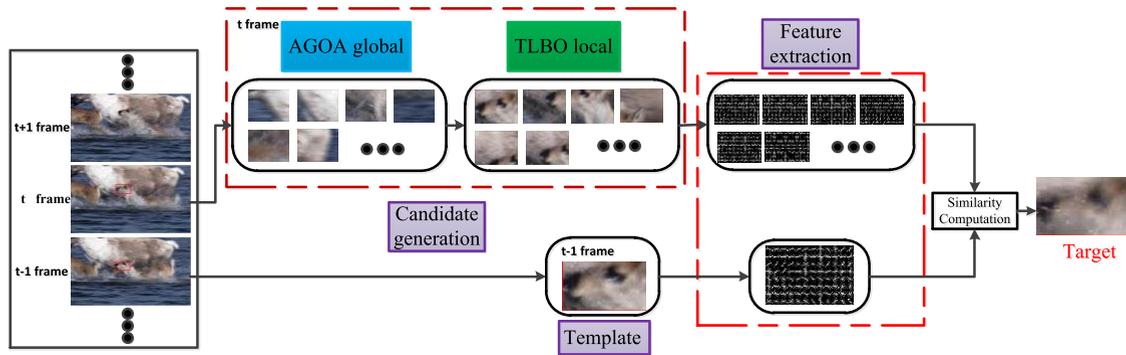


FIGURE 5. The flowchart of the TLGOA tracker.

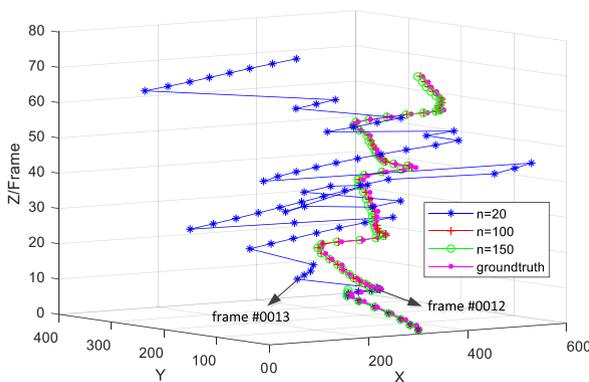


FIGURE 6. Performance comparison with different population size n .

target are extracted. Finally, the fitness function is utilized to find the best target candidate as the output of the current frame and the target of the next frame. The pseudo-code of proposed hybrid TLGOA tracker is shown in Algorithm 1.

C. PARAMETERS’ SENSITIVITY AND ADJUSTMENT

It is well-known that parameter adjustment is vital for optimization algorithm. The speed and accuracy should be considered simultaneously during the parameter tuning. There are four main parameters in TLGOA tracker, namely n (population size), f (the intensity of attraction), l (the attractive length scale), and T (the numbers of iteration), respectively. In this paper, we discuss parameters n, f and l , then the parameter T is designed to be $T = 300$. In addition, DEER video sequence, coming from the website <http://visual-tracking.net>, is selected as the testing source material because the video sequence has similar target and abrupt motion.

We first analyze the population size n , and other parameters are fixed. We test the performance of the TLGOA tracker with different population size n and observe the tracking results of $n = 20, n = 100, n = 150$, respectively. Then the tracking results are compared with the coordinates of groundtruth as shown in Fig. 6.

It can be seen from Fig. 6 that the tracking result at frame #0013 deviates from the target when the population size is

TABLE 5. The tracking result with different parameter l and f .

| l, f | Average fitness value | Average overlap rate | Average center error rate |
|--------------------|-----------------------|----------------------|---------------------------|
| $l = 1.8, f = 0.2$ | 0.8938 | 0.6887 | 8.4984 |
| $l = 1.5, f = 0.5$ | 0.8940 | 0.7378 | 6.7353 |
| $l = 1.2, f = 0.8$ | 0.8932 | 0.6945 | 8.1748 |
| $l = 1, f = 1$ | 0.8546 | 0.0156 | 222.0234 |

$n = 20$. However, when the population size is $n = 100$ and $n = 150$, the tracking results are very close to the coordinates of groundtruth. However, there is no doubt that the increasing population size makes the tracker more time-consuming. Thereby, comprehensive considering the tracking accuracy and speed, we set the best value of the population size to be $n = 100$.

Then we analyze the parameters l and f . Among them, the parameter l is the attractive length scale and the parameter f indicates the intensity of attraction. In order to display the tracking results of different parameters, we calculate the average fitness value, the average overlap rate and the average center error rate of each set of parameters separately. The best result is shown in red fonts. It can be seen from Table 5 that the parameter $l = 1$ and $f = 1$ have the worst performance, whereas $l = 1.5$ and $f = 0.5$ show the best result compared to other parameters. Therefore, the parameters l and f are set to be 1.5 and 0.5.

VI. EXPERIMENTS AND DISCUSSIONS

We implemented the proposed tracker in MATLAB R2017b. The experiments were conducted on a PC with Intel Core i5-7500 3.40GHz and 8GB RAM. To verify the feasibility of visual tracking system based on hybrid TLGOA optimization algorithm, the experiments included the following: (1) Performance evaluation of AGOA; (2) Comparison of tracking performance among TLGOA and AGOA; (3) Comparison of the state-of-the-art trackers.

A. PERFORMANCE EVALUATION OF AGOA

To investigate the performance of proposed AGOA, the first frame of FACE1 video sequence is selected as experimental

Algorithm 1 The Pseudo-Code of TLGOA Tracker

Input: Image sequence

Initialization: Locate the target object in the first frame manually, the population of grasshoppers (target candidates), the number of iterations (T), constant (k), the intensity of attraction (l), the attractive length scale (f)

Calculate the fitness of grasshoppers (target candidates).
Find the best grasshopper (target candidate).

Tracking:

for i from 2 to the last frame **do**

while Current_iter < $T + 1$ **do**

Update c using Eq. (11)

for every grasshopper **do**

Normalize the distances between grasshoppers in [1,4]

Update the position of ant using Eq. (9)

end for

Calculate the fitness of all grasshoppers

Replace the best grasshopper with its corresponding grasshopper if it becomes fitter

%Teacher Phase%

for every grasshopper **do**

Update the position of grasshopper using Eq. (1) and Eq. (2)

end for

%Learner Phase%

Select any two solutions randomly

Update the position of grasshopper using Eq. (3) and Eq. (4)

%The Best Grasshopper%

Calculate the fitness of all grasshoppers

Replace the best grasshopper with its corresponding grasshopper if it becomes fitter

Current_iter = Current_iter+1

end while

Return the best grasshopper (target)

end for

Output: Target state X_{best} in the frame

material. In addition, the same motion model (social behavior of grasshoppers) and the feature (HOG) are adopted, GOA and AGOA are set the same initial position in the first iteration. In the experiment, the parameters were consistent (i.e., population size $n = 50$, $T = 50$, $l = 1.5$, $f = 0.5$, $k = 0.25$). The swarming behavior of GOA and AGOA is shown in Fig. 7.

As can be seen from Fig. 7, for GOA method, the diversity of population decreases with the increase of iterations. When the number of iterations $t = 0.5T$, the population position distribution has deviated from the target. However, the standard GOA utilizes an adaptive linear mechanism representing the variation of exploring step. It is difficult for this mechanism to jump out of local optimum and achieve global optimum in the subsequent search process, which eventually

TABLE 6. The video sequences (pre-processing/post-processing).

| Video | Frame | Max displacement | X Max displacement | Y Max displacement |
|----------|----------|------------------|--------------------|--------------------|
| MHYANG | 1490/995 | 7/35 | 7/35 | 4/8 |
| FISH | 476/320 | 15/56 | 15/56 | 13/28 |
| MAN | 134/69 | 5/22 | 5/22 | 3/3 |
| JUMPING | 313/210 | 36/41 | 18/41 | 36/72 |
| DEER | 71/36 | 38/134 | 38/134 | 34/115 |
| FACE1 | 380/255 | 39/102 | 22/102 | 39/95 |
| ZXJ | 118/60 | 70/121 | 70/121 | 18/62 |
| BLURFACE | 493/330 | 71/134 | 64/134 | 71/129 |
| FHC | 123/63 | 188/571 | 188/571 | 104/504 |
| ZT | 115/60 | 256/637 | 256/637 | 149/151 |

leads to tracking failure. The AGOA method adopts non-linear strategy instead of linear mechanism. It is obvious from Fig. 7 that the diversity of the population is guaranteed during the search process and the target is successfully tracked. In general, the AGOA method has stronger searching ability, and the local best can be avoided better.

B. COMPARISON OF TRACKING PERFORMANCE AMONG TLGOA AND AGOA

To verify the robustness and efficiency of TLGOA tracker, just as the Sec. VI-A, the same environment (feature, video sequence and parameter) is selected. In addition, population size and the iteration number are investigated by dividing the range [10,50] and [20,100] into equal parts, with each a space 10 and 20, respectively. Other parameters are fixed (i.e., $l = 1.5$, $f = 0.5$, $k = 0.25$). During the testing process, each group of parameters was run 30 times to record fitness values and calculate their average values. The experimental results are shown in Fig. 8.

It is obvious from Fig. 8 that the average fitness of TLGOA is always higher than that of AGOA under the same population size and the iteration number. That is to say, compared with AGOA, TLGOA can track targets successfully with fewer population size and the iteration number. In addition, the average fitness of TLGOA increased uniformly with the increase of population size and the iteration number. Therefore, TLGOA tracker has unique advantages in accuracy and efficiency.

C. COMPARISON OF THE STATE-OF-THE-ART TRACKERS

To prove the proposed method can track object successfully, we selected 10 video sequences in the experiment (listed in Table 6). The source of the FACE1 is the dataset AVSS2007. ZT, FHC and ZXJ are our own. Other sequences are available on the website <http://visual-tracking.net>. In this study, for showing the different tracking results with abrupt motion, all video sequences are dealt with in advance in this study, i.e., delete 5 frames for every 5 frames below 300 frames, and delete 5 frames for every 10 frames above 300 frames (listed in Table 6). The experiments are performed on both pre-processing video sequences and post-processing video sequences, respectively.

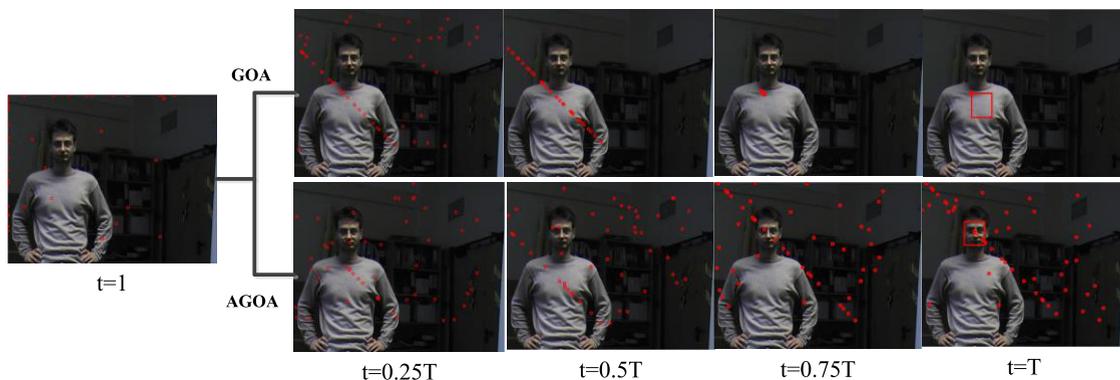


FIGURE 7. Swarming behavior of GOA and AGOA.

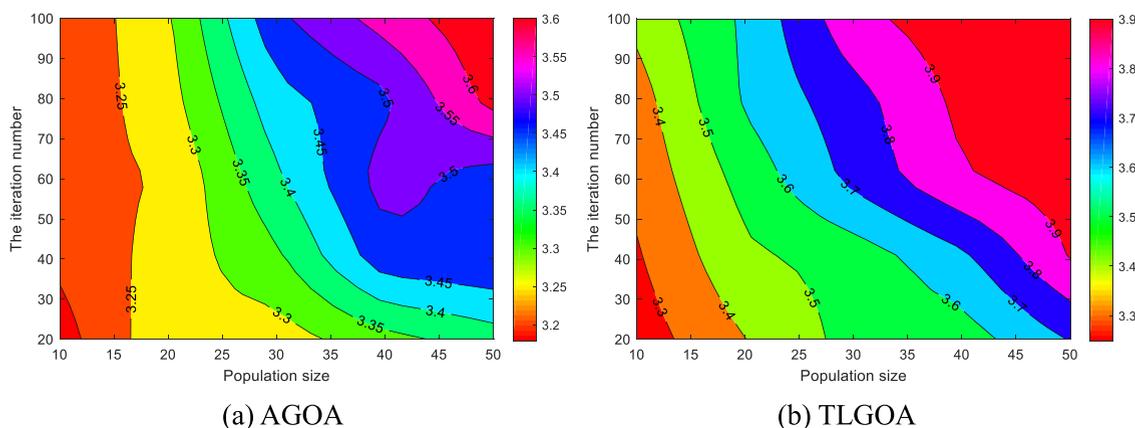


FIGURE 8. Performance comparison using different population size and the iteration number.

We compared TLGOA with GOA, TLBO and 6 state-of-the-art trackers(Fast Compressive Tracking(FCT) [51], High-Speed Tracking with Kernelized Correlation Filters(KCF) [52], Accurate Scale Estimation for Robust Visual Tracking (DSST) [53], Fast Tracking via Spatio-Temporal Context Learning (STC) [54], Least soft-threshold squares tracking(LSST) [55] and Context-Aware Correlation Filter Tracking(CACF) [56]). In the course of the experiment, all parameters are fixed(i.e., $n = 100, T = 300, l = 1.5, f = 0.5$ and $k = 0.25$). In addition, we adopt both qualitative and quantitative methods to test the pre-processing and the post-processing video sequences, and observe the experimental results. In qualitative analysis, we manually label by identifying the upper left corner of the tracked target in each frame of the image sequence. In quantitative analysis, the tracking results are evaluated by using distance precision (DP), center location error (CLE) and overlap precision (OP) in [57].

1) QUALITATIVE ANALYSIS

The tracking results with pre-processing video sequences and post-processing video sequences are shown in Fig. 9(a) and Fig. 9(b).

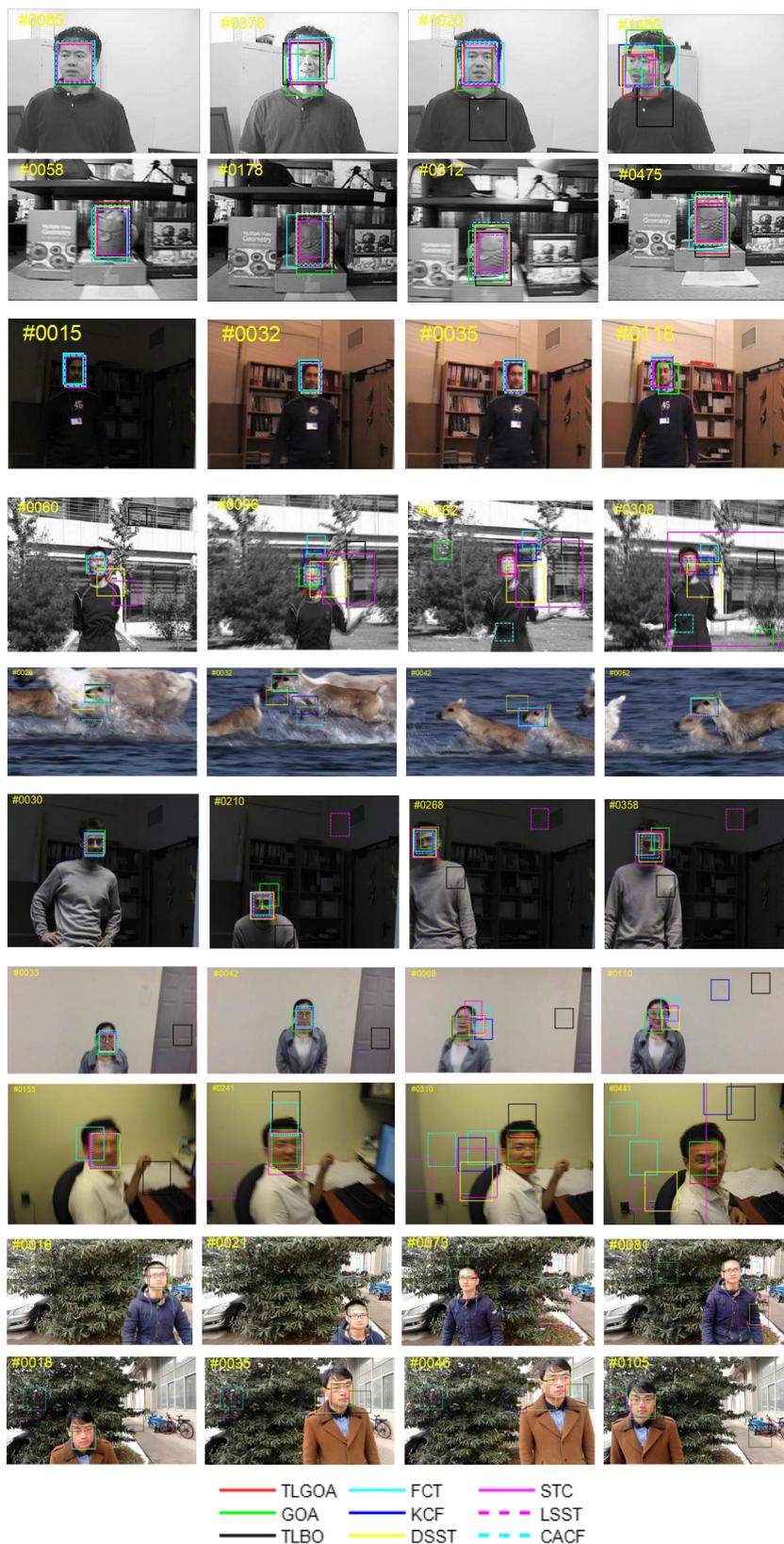
In MHYANG video sequence, it is obvious that there is a change in the brightness of the lights. FCT, TLBO and GOA

have slightly worse tracking results, while others have similar performance with pre-processing video sequence. However, for post-processing video sequence, STC is failure at frame #0333. TLBO has lost its target, GOA and FCT deviate the target at frame #0800 because of the larger illumination changing. Other trackers complete the whole video sequence.

FISH video sequence has camera shake and illumination change. For pre-processing video sequence, obviously, there is camera shake at frame #0058 and #0312, and the brightness is dimmed at frame #0178. Although all trackers can successfully capture targets, TLGOA and KCF show a better performance. For post-processing video sequence, STC, FCT and LSST lose the target unfortunately before frame #0236.

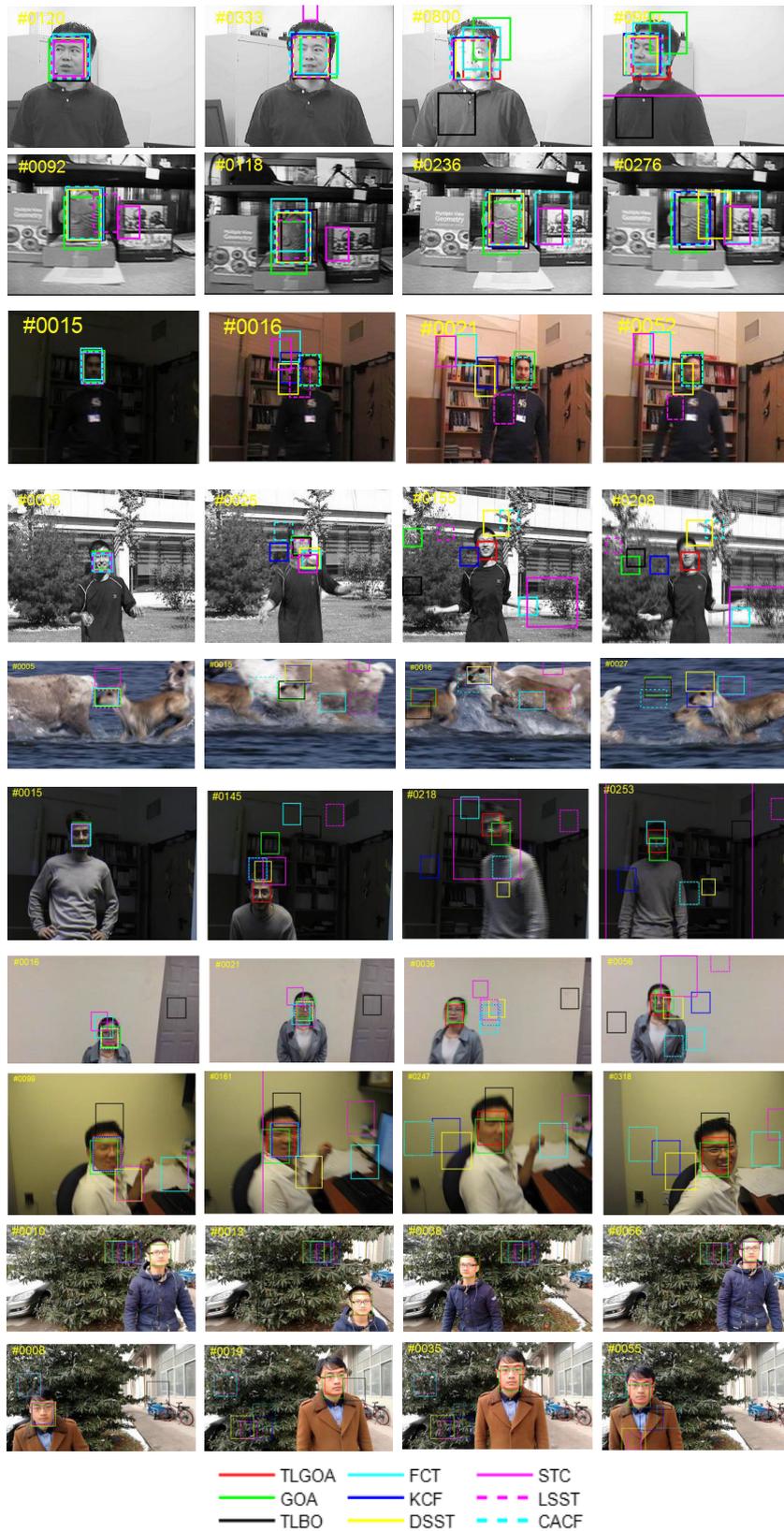
MAN video sequence undergoes a process from dark to bright. All trackers can track the target successfully in the pre-processing video sequence. For post-processing video sequence, all trackers can track the target successfully before #0015 frame. In addition, we can clearly see that the MAN video sequence has great illumination changes from frame #0015 to frame #0016. This caused all trackers to lose their targets except for CACF, GOA, TLBO and TLGOA. And the four trackers complete the whole video sequence.

It is very clear that JUMPING video sequence has been seriously blurred because of the camera shaking.



(a) The pre-processing video sequences

FIGURE 9. A visualization of tracking results.



(b) The post-processing video sequences

FIGURE 9. (Continued.) A visualization of tracking results.

TABLE 7. Average overlap rate (pre-processing/post-processing).

| Sequences | TLGOA | GOA | TLBO | FCT | KCF | DSST | STC | LSST | CACF |
|-----------|-------------------|-------------------|-------------------|-----------|-------------------|-------------------|-----------|-------------------|------------------|
| MHYANG | 0.60/0.75 | 0.54/0.41 | 0.35/0.60 | 0.59/0.56 | 0.80/0.80 | 0.81/0.78 | 0.69/0.13 | 0.78/ 0.81 | 0.78/0.77 |
| FISH | 0.83/0.80 | 0.66/0.63 | 0.74/ 0.80 | 0.66/0.47 | 0.84/0.83 | 0.80/0.66 | 0.58/0.16 | 0.63/0.34 | 0.83/0.79 |
| MAN | 0.70/ 0.78 | 0.63/0.77 | 0.70/ 0.78 | 0.71/0.15 | 0.84/0.26 | 0.84/0.22 | 0.83/0.20 | 0.79/0.20 | 0.84/0.86 |
| JUMPING | 0.60/0.65 | 0.54/ 0.44 | 0.09/0.12 | 0.20/0.07 | 0.27/0.05 | 0.14/0.05 | 0.07/0.06 | 0.60/0.35 | 0.50/0.08 |
| DEER | 0.74/0.33 | 0.72/0.33 | 0.70/0.33 | 0.66/0.14 | 0.62/ 0.43 | 0.64/ 0.42 | 0.04/0.08 | 0.71/0.10 | 0.63/0.30 |
| FACE1 | 0.65/ 0.66 | 0.38/0.23 | 0.24/0.25 | 0.63/0.27 | 0.72/0.39 | 0.79/0.41 | 0.65/0.34 | 0.26/0.24 | 0.72/0.38 |
| ZXJ | 0.84/0.84 | 0.57/ 0.56 | 0.24/0.19 | 0.45/0.24 | 0.45/0.46 | 0.48/0.45 | 0.46/0.09 | 0.79/0.37 | 0.83/0.47 |
| BLURFACE | 0.68/0.66 | 0.56/0.56 | 0.03/0.08 | 0.23/0.13 | 0.51/0.49 | 0.53/0.19 | 0.30/0.10 | 0.51/0.16 | 0.51/0.48 |
| FHC | 0.85/0.82 | 0.42/0.80 | 0.09/0.09 | 0.26/0.09 | 0.28/0.37 | 0.22/0.08 | 0.20/0.08 | 0.28/0.10 | 0.78/0.08 |
| ZT | 0.80/0.80 | 0.64/ 0.70 | 0.03/0.03 | 0.09/0.08 | 0.59/0.15 | 0.65/0.15 | 0.35/0.14 | 0.09/0.17 | 0.84/0.15 |

TABLE 8. Average center error rate (pre-processing/post-processing).

| Sequences | TLGOA | GOA | TLBO | FCT | KCF | DSST | STC | LSST | CACF |
|-----------|--------------------|----------------------|---------------|---------------|---------------------|--------------------|--------------------|--------------------|---------------------|
| MHYANG | 14.77/6.50 | 17.68/27.84 | 50.75/30.10 | 14.72/16.19 | 3.61/3.79 | 2.43/2.32 | 4.22/126.73 | 2.60/2.75 | 6.71/6.83 |
| FISH | 4.87/6.16 | 12.17/14.09 | 9.78/6.16 | 11.99/35.75 | 4.08/4.28 | 4.36/13.56 | 4.64/52.97 | 3.97/20.69 | 4.32/6.29 |
| MAN | 4.48/2.89 | 6.79/3.84 | 4.48/2.89 | 4.04/50.96 | 2.21/27.31 | 1.68/30.72 | 2.11/62.09 | 2.01/44.89 | 2.18/ 1.96 |
| JUMPING | 7.48/6.10 | 36.21/44.81 | 110.54/132.48 | 37.23/75.26 | 26.26/49.92 | 36.55/57.62 | 66.70/110.09 | 6.05/54.33 | 33.84/71.44 |
| DEER | 6.74/105.21 | 7.39/126.49 | 8.12/115.71 | 10.68/121.87 | 21.13/ 61.55 | 16.65/46.19 | 509.55/256.66 | 7.20/184.37 | 22.96/135.50 |
| FACE1 | 10.04/ 8.99 | 32.53/64.37 | 139.14/200.10 | 12.09/95.90 | 5.82/92.87 | 5.30/109.38 | 6.71/ 57.18 | 183.77/218.38 | 5.08/80.90 |
| ZXJ | 4.49/4.37 | 13.95/14.60 | 186.03/169.31 | 26.74/69.12 | 88.46/74.64 | 21.19/50.55 | 23.68/86.64 | 5.06/99.57 | 4.84/61.60 |
| BLURFACE | 15.32/16.47 | 26.02/24.55 | 179.77/121.71 | 116.33/160.60 | 84.83/75.30 | 74.93/111.30 | 89.75/276.39 | 162.05/180.66 | 111.94/111.96 |
| FHC | 13.86/17.32 | 330.11/ 18.48 | 596.11/680.26 | 371.37/398.70 | 364.79/298.07 | 616.05/415.22 | 576.47/392.28 | 392.42/392.32 | 23.04/412.69 |
| ZT | 19.37/19.44 | 46.06/37.77 | 476.84/413.83 | 642.35/668.89 | 127.53/525.57 | 54.01/672.06 | 99.65/663.27 | 684.85/613.50 | 16.71/483.08 |

In this case, for pre-processing video sequence, TLBO, FCT, KCF, DSST and STC lose the target unfortunately before frame #0096. Although TLGOA drifts the target slightly, it recovers quickly. And TLGOA shows excellent performance. All trackers are failures except for TLGOA before the frame #0055 in post-processing video sequence.

DEER video sequence experiences fast motion, motion blur and similar targets. For pre-processing video sequence, DSST, KCF and CACF lose the target at frame #0032. Although they recover tracking in the next few frames, they still cannot track the entire video sequence. On the whole, TLGOA obtains the best tracking results. On the other hand, TLGOA, GOA and TLBO can successfully track the target before frame #0015 in post-processing video sequence. Although KCF can recover quickly, it still cannot track the whole video sequence. However, at frame #0016, TLGOA, GOA and TLBO lost their target because of severe blur, distortion or other factors. In addition, the feature extraction of this paper may also lead to tracking failure. In a word, all trackers are failures.

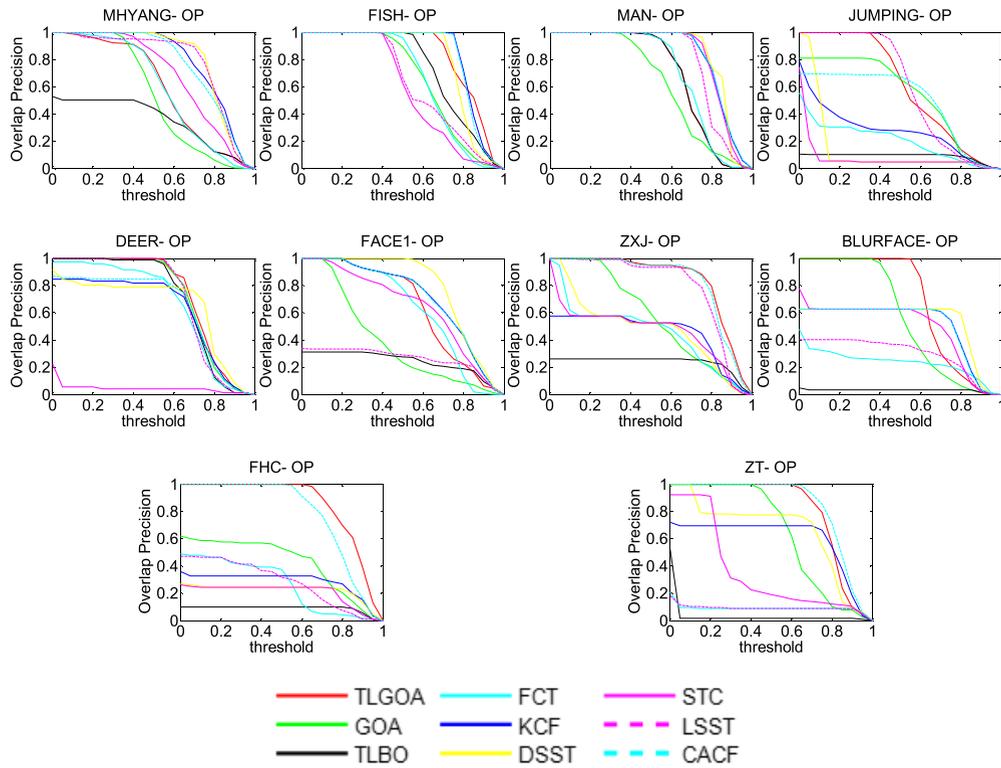
For FACE1 video sequence, there are scale changes. Evidently, GOA, TLBO and LSST fail to track the target at frame #0210 in pre-processing video sequence. Other trackers can complete the whole video sequence successfully. Among them, DSST has the best performance. For post-processing video sequence, all trackers have lost their target before frame #0145 except for TLGOA.

For pre-processing video sequences, the maximum displacements of ZXJ, BLURFACE, FHC and ZT video

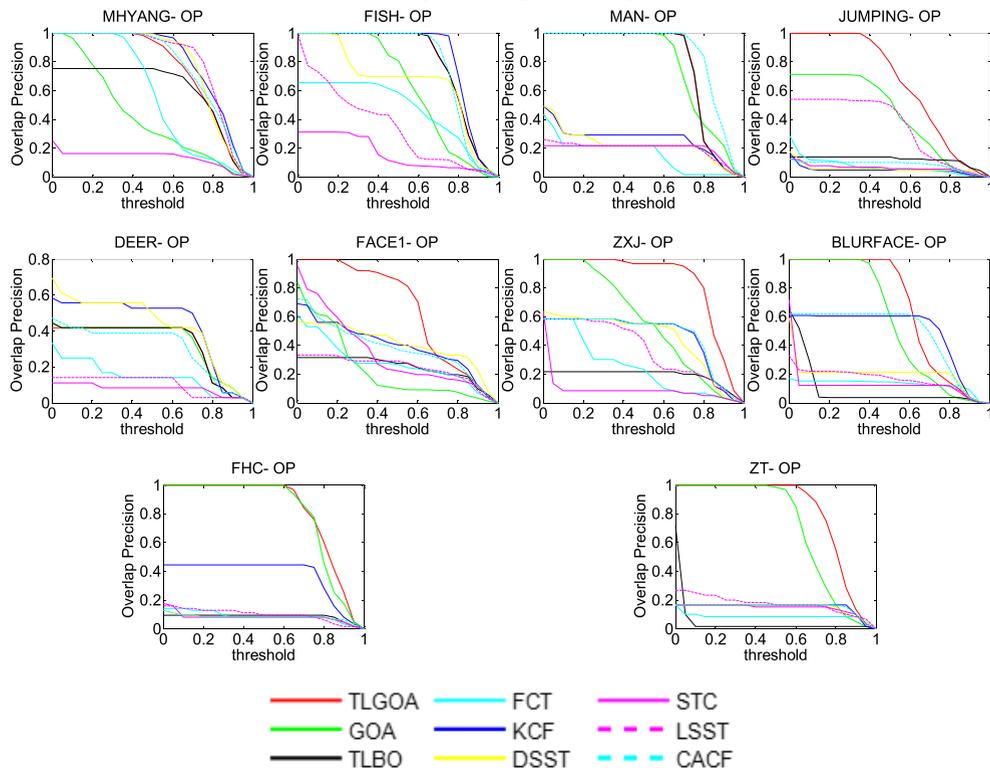
sequences are 70, 71, 188 and 256 pixels, respectively. Among them, ZXJ video sequence shows that the target occurs slight motion blur because of camera shaking continuously. Before the frame #0042, all trackers perform well except for TLBO for pre-processing video sequence. However, other methods all lose the target at frame #0069 except for CACF, GOA and TLGOA. Additionally, LSST deviates the target due to abrupt motion, but it can recover quickly. TLGOA shows the best performance. BLURFACE video sequence was designed the problem of the frame dropping. There has a severe motion blur since the abrupt motion at the frame #0153, #0241 and #0310. Other trackers lost their targets, but TLGOA still can successfully track the target of the whole video sequence. In FHC video sequence, other trackers fail at frame #0073 besides CACF and TLGOA. TLGOA performs much better than 8 other trackers. In ZT video sequence, the motion displacement between the consecutive video frames reaches 256 pixels. Before frame #0035, all trackers are failures except for CACF, GOA and TLGOA. Note that CACF obtains the best results and TLGOA gets the second best.

For post-processing video sequence, the maximum displacements of ZXJ, BLURFACE, FHC and ZT video sequences are 121, 134, 571 and 637 pixels, respectively. Since these video sequences undergo abrupt motion. All trackers lose track targets except for GOA and TLGOA, and TLGOA gets the best performance.

To sum up, compared with other methods, our tracker can keep the better performance and track smooth or abrupt motion simultaneously.

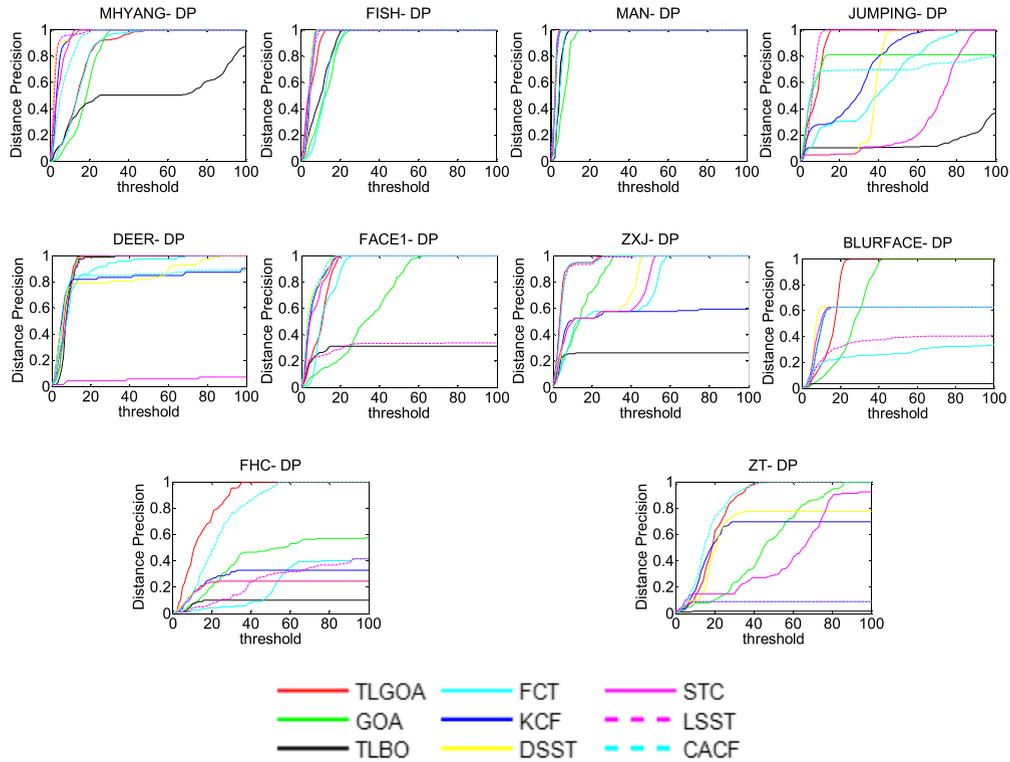


(a) The pre-processing video sequences

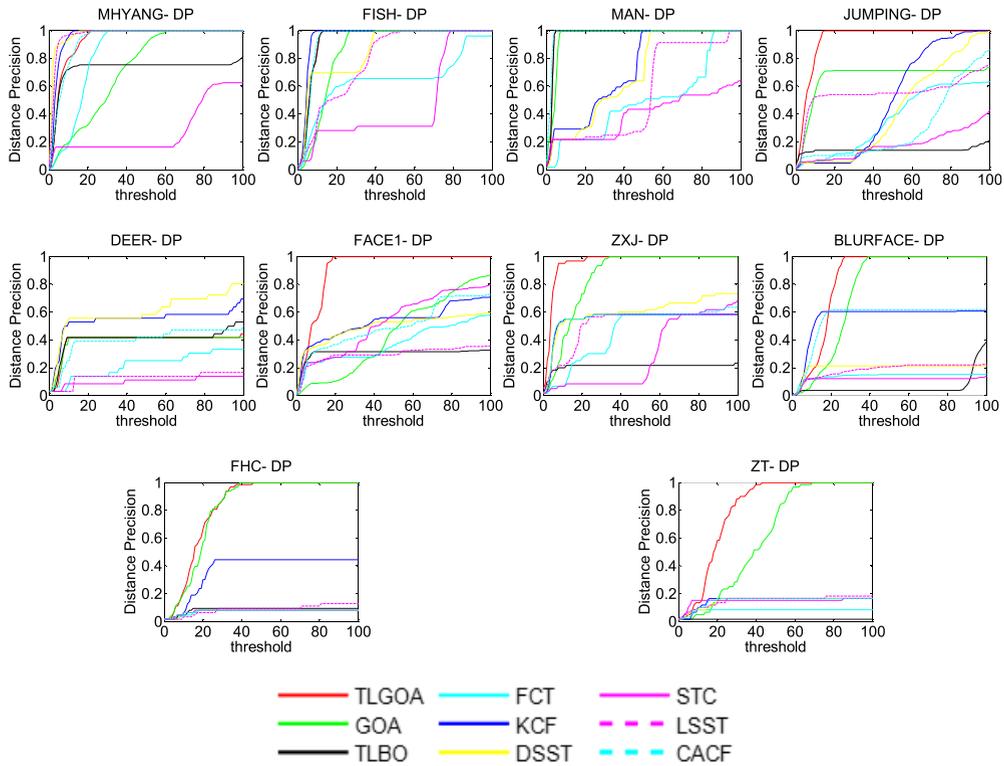


(b) The post-processing video sequences

FIGURE 10. The average precision of OP.



(a) The pre-processing video sequences



(b) The post-processing video sequences

FIGURE 11. The average precision of DP.

2) QUANTITATIVE ANALYSIS

In order to directly observe the tracking results of different trackers, we calculate the average overlap rate and average center error rate of all trackers. Table 7 and Table 8 list a per-sequence comparison of our tracker to GOA, TLBO, FCT, KCF, DSST, STC, LSST, and CACF. Among them, Table 7 refers to average overlap rate and Table 8 is concerned with average error center rate. Red fonts obtain the best results and blue fonts get the second best.

In addition, Fig. 10 and Fig. 11 show the OP and DP of 10 different sequences respectively. Among them, Fig. 10(a) and Fig. 11(a) report the OP of the pre-processing video sequences; Fig. 10(b) and Fig. 11(b) report the DP of the post-processing video sequences. DP is the relative number of frames in the sequence where the center location error is smaller than a certain threshold. OP is defined as the percentage of frames where the bounding box overlap exceeds a threshold. Fig. 10 and Fig. 11 can vividly show the performance of different tracking methods, including TLGOA, GOA, TLBO, FCT, KCF, DSST, STC, LSST and CACF.

Based on the above discussion, it is obviously seen in Fig. 10, Fig. 11, Table 7 and Table 8 that our tracker performs much better than 8 other trackers whether it is pre-processing video sequences or post-processing video sequences. In a word, the proposed method has a good merit for abrupt motion compared with other trackers.

VII. CONCLUSION

During the past 20 years, nature-inspired algorithms have become a hotspot in the optimization computation. In this paper, a hybrid TLGOA tracker is proposed to improve tracking accuracy and efficiency. Among them, a non-linear strategy based on tangent function is introduced into GOA, which is called AGOA. The proposed AGOA have stronger searching ability to handle abrupt motion. 5 benchmark functions and 30 CEC 2014 benchmark problems are conducted to verify the effectiveness of the hybrid TLGOA. At the same time, qualitative and quantitative experiments verify that TLGOA tracker outperforms GOA tracker, TLBO tracker and 6 state-of-the-art trackers, especially for abrupt motion tracking. Choosing the very strong deep features to solve the visual tracking problem will be the focus of our future work.

REFERENCES

- [1] H. Zhang, S. Hu, and J. Yu, "Visual tracking via robust multitask sparse prototypes," *Proc. SPIE*, vol. 24, no. 2, 2015, Art. no. 023025.
- [2] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja, "Robust visual tracking via exclusive context modeling," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 51–63, Jan. 2016.
- [3] T. Zhou, F. Liu, H. Bhaskar, J. Yang, H. Zhang, and P. Cai, "Online discriminative dictionary learning for robust object tracking," *Neurocomputing*, vol. 275, pp. 1801–1812, Jan. 2018.
- [4] H. Zhang, Y. Wang, L. Luo, and M. Zhang, "Sift flow for abrupt motion tracking via adaptive samples selection with sparse representation," *Neurocomputing*, vol. 249, pp. 235–262, Aug. 2017.
- [5] C. Wang, Y. Xu, H. Liu, and X. Jing, "Retrospection of correlation filters for discriminative visual object tracking," *Proc. SPIE*, vol. 27, no. 6, 2018, Art. no. 063010.
- [6] T. Zhang, C. Xu, and M.-H. Yang, "Multi-task correlation particle filter for robust object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4335–4343.
- [7] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 749–765.
- [8] F. Tang, X. Zhang, S. Hu, and H. Zhang, "Convolutional features selection for visual tracking," *Proc. SPIE*, vol. 27, no. 3, Jun. 2018, Art. no. 033031.
- [9] K. Lebeda, S. Hadfield, J. Matas, and R. Bowden, "Texture-independent long-term tracking using virtual corners," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 359–371, Jan. 2016.
- [10] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specific proposals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle WA, USA, Jun. 2016, pp. 943–951.
- [11] J. Kwon and K. M. Lee, "Wang-Landau Monte Carlo-based tracking methods for abrupt motions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 1011–1024, Apr. 2013.
- [12] X. Zhou and Y. Lu, "Abrupt motion tracking via adaptive stochastic approximation Monte Carlo sampling," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1847–1854.
- [13] H. Zhang, X. Zhang, Y. Wang, Y. Wang, K. Shi, J. Zhang, and C. Li, "An experimental comparison of swarm optimization based abrupt motion tracking methods," *IEEE Access*, vol. 6, pp. 75383–75394, 2018.
- [14] H. Zhang, X. Zhang, Y. Wang, X. Qian, and Y. Wang, "Extended cuckoo search-based kernel correlation filter for abrupt motion tracking," *IET Comput. Vis.*, vol. 12, no. 6, pp. 763–769, Sep. 2018.
- [15] X. Zhang, W. Hu, S. Maybank, X. Li, and M. Zhu, "Sequential particle swarm optimization for visual tracking," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.
- [16] H. T. Nguyen and B. Bhanu, "Real-time pedestrian tracking with bacterial foraging optimization," in *Proc. IEEE 9th Int. Conf. Adv. Video Signal-Based Surveill.*, Sep. 2012, pp. 37–42.
- [17] M.-L. Gao, J. Shen, L.-J. Yin, W. Liu, G.-F. Zou, H.-T. Li, and G.-X. Fu, "A novel visual tracking method using bat algorithm," *Neurocomputing*, vol. 177, pp. 612–619, Feb. 2016.
- [18] D. H. Wolper and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
- [19] Y.-J. Gong, J.-J. Li, Y. Zhou, Y. Li, H. S.-H. Chung, Y.-H. Shi, and J. Zhang, "Genetic learning particle swarm optimization," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2277–2290, Oct. 2016.
- [20] A. A. Nagar, F. Han, Q.-H. Ling, and S. Mehta, "An improved hybrid method combining gravitational search algorithm with dynamic multi swarm particle swarm optimization," *IEEE Access*, vol. 7, pp. 50388–50399, 2019.
- [21] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neuro Comput.*, vol. 260, pp. 302–312, Oct. 2017.
- [22] G. Wang and L. Guo, "A novel hybrid bat algorithm with harmony search for global numerical optimization," *J. Appl. Math.*, vol. 2013, pp. 1–21, 2013.
- [23] J. Wei and Y. Yu, "An effective hybrid cuckoo search algorithm for unknown parameters and time delays estimation of chaotic systems," *IEEE Access*, vol. 6, pp. 6560–6571, 2017.
- [24] J. Zhang, Y. Zhou, and Q. Luo, "An improved sine cosine water wave optimization algorithm for global optimization," *J. Intell. Fuzzy Syst.*, vol. 34, no. 4, pp. 2129–2141, Apr. 2018.
- [25] T. Ljouad, A. Amine, and M. Rziza, "A hybrid mobile object tracker based on the modified cuckoo search algorithm and the Kalman filter," *Pattern Recognit.*, vol. 47, pp. 3597–3613, Nov. 2014.
- [26] J. Chen, Y. Zhen, D. Yang, and D. Yang, "Fast moving object tracking algorithm based on hybrid quantum PSO," *WSEAS Trans. Comput.*, vol. 12, no. 10, pp. 375–383, Oct. 2013.
- [27] H. Nenavath, D. R. K. Jatoth, and D. S. Das, "A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking," *Swarm Evol. Comput.*, vol. 43, pp. 1–30, Dec. 2017.
- [28] H. Nenavath and R. K. Jatoth, "Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking," *Appl. Soft Comput.*, vol. 62, pp. 1019–1043, Jan. 2018.
- [29] H. Zhang, J. Zhang, Q. Wu, T. Zhou, and F. U. Hengcheng, "Extended kernel correlation filter for abrupt motion tracking," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 9, pp. 4438–4460, 2017.
- [30] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017.

- [31] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching learning based optimization: A novel method for constrained mechanical design optimization problems," *Comput. Aided Des.*, vol. 43, no. 3, pp. 303–315, Mar. 2011.
- [32] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: An optimization method for continuous non-linear large scale problems," *Inf. Sci.*, vol. 183, no. 1, pp. 1–15, Jan. 2012.
- [33] R. V. Rao *Teaching-Learning-Based Optimization Algorithm*. Cham, Switzerland: Springer, 2016, doi: 10.1007/978-3-319-22732-0_2.
- [34] J. Wu, H. Wang, N. Li, P. Yao, Y. Huang, Z. Su, and Y. Yu, "Distributed trajectory optimization for multiple solar-powered UAVs target tracking in urban environment by adaptive grasshopper optimization algorithm," *Aerosp. Sci. Technol.*, vol. 70, pp. 497–510, Nov. 2017.
- [35] X. Zhang, Q. Miao, H. Zhang, and L. Wang, "A parameter-adaptive VMD method based on grasshopper optimization algorithm to analyze vibration signals from rotating machinery," *Mech. Syst. Signal Process.*, vol. 108, pp. 58–72, Aug. 2018.
- [36] J. Huang, X. Li, and L. Gao, "A new hybrid algorithm for unconstrained optimisation problems," *Int. J. Comput. Appl. Technol.*, vol. 46, no. 3, pp. 187–194, Jan. 2013.
- [37] S. Shahbeig, M. S. Helfroush, and A. Rahideh, "A fuzzy multi-objective hybrid TLBO-PSO approach to select the associated genes with breast cancer," *Signal Process.*, vol. 131, pp. 58–65, Feb. 2017.
- [38] H. Nenavath and R. K. Jatoth, "Hybrid SCA-TLBO: A novel optimization algorithm for global optimization and visual tracking," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 5497–5526, Sep. 2019.
- [39] E. Natarajan, V. Kaviarasan, W. H. Lim, S. S. Tiang, and T. H. Tan, "Enhanced multi-objective teaching-learning-based optimization for machining of delrin," *IEEE Access*, vol. 6, pp. 51528–51546, 2018.
- [40] A. A. Ewees, M. A. Elaziz, and E. H. Houssein, "Improved grasshopper optimization algorithm using opposition-based learning," *Expert Syst. Appl.*, vol. 112, pp. 156–172, Dec. 2018.
- [41] S. Arora and P. Anand, "Chaotic grasshopper optimization algorithm for global optimization," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 4385–4405, Aug. 2019.
- [42] S. C. Satapathy and A. Naik, "Modified teaching-learning-based optimization algorithm for global numerical optimization—A comparative study," *Swarm Evol. Comput.*, vol. 16, pp. 28–37, Jun. 2014.
- [43] K. Yu, X. Wang, and Z. Wang, "An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems," *J. Intell. Manuf.*, vol. 27, no. 4, pp. 831–843, Aug. 2016.
- [44] J. Ronkkonen, S. Kukkonen, and K. V. Price, "Real-parameter optimization with differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, vol. 1, Sep. 2005, pp. 506–513.
- [45] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 71–78.
- [46] R. Tanabe and A. S. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2014, pp. 1658–1665.
- [47] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization," *Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Nanyang Technol. Univ., Singapore, Tech. Rep.*, 2013, vol. 635.
- [48] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
- [49] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
- [50] F. Wilcoxon, S. Katti, and R. A. Wilcox, "Critical values and probability levels for the wilcoxon rank sum test and the wilcoxon signed rank test," *Select. Tables Math. Statist.*, vol. 1, pp. 171–259, Jun. 1970.
- [51] K. Zhang, L. Zhang, and M. Yang, "Fast compressive tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2002–2015, Oct. 2014.
- [52] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.
- [53] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vision Conf. BMVC*, Sep. 2014, pp. 1–5.
- [54] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense Spatio-temporal context learning," in *Proc. Eur. Conf. Comput. Vis. ECCV*, vol. 8693, 2014, pp. 127–141.
- [55] D. Wang, H. Lu, and M.-H. Yang, "Robust visual tracking via least soft-threshold squares," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1709–1721, Sep. 2016.
- [56] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1396–1404.
- [57] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2411–2418.



HUANLONG ZHANG was born in Lingbao, Henan, China, in 1981. He received the M.S. degree from Henan University, Kaifeng, China, in 2007, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2015. He is currently a Teacher with the Zhengzhou University of Light Industry. His research interests include pattern recognition, machine learning, image processing, and computer vision.



ZENG GAO was born in Kaifeng, Henan, China, in 1991. He is currently pursuing the degree with the Zhengzhou University of Light Industry, Zhengzhou, China. His research interests include machine learning and visual tracking.



XIAOYANG MA is currently pursuing the bachelor's degree in automation specialty with the Zhengzhou University of Light Industry, Henan, China. He is also currently pursuing the degree with the College of Electric and Information Engineering, Zhengzhou University of Light Industry. His research interests include machine vision and pattern recognition.



JIE ZHANG was born in 1987. He is currently pursuing the Ph.D. degree. He is currently a Lecturer with the Zhengzhou University of Light Industry. His research interests include image processing and compressed sensing.



JIANWEI ZHANG was born in Nanyang, Henan, China, in 1971. He received the M.S. degree from the Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree from PLA Information Engineering University, Zhengzhou, China, in 2010. He is a Professor. He is currently with the Zhengzhou University of Light Industry, Zhengzhou. His research interests include broadband information networks, distributed systems, and information security.

• • •