



# A Modified Brain Storm Optimization Based Distributed Model Predictive Control Method for Formation Control of Multiple UAVs

Tian Zhang<sup>(✉)</sup>, Xiaodong Wang, Xun Song, and Meiliwen Wu

Beijing Institute of Electronic System Engineering, Beijing 100854, China  
tzhang\_april@163.com

**Abstract.** In this paper, a distributed Model Predictive Control (MPC) method based on modified Brain Storm Optimization (BSO) algorithm is proposed to solve the control problem of multiple Unmanned Aerial Vehicles (UAVs). The modified BSO algorithm considers all cluster centers, and the chaos mechanism is introduced into the solution replacement mechanism. Based on the distributed model predictive control framework, the cost functions and trigger mechanism are designed for formation control. Simulation results show that the designed trigger mechanism can reduce the amount of calculation, and verifies the feasibility and effectiveness of the proposed method.

**Keywords:** Unmanned Aerial Vehicle (UAV) · Modified Brain Storm Optimization (BSO) · Distributed Model Predictive Control (MPC)

## 1 Introduction

In an increasingly complex combat environment, a single small Unmanned Aerial Vehicle (UAV) is gradually unable to meet mission requirements. UAV swarm has attracted more attention. When performing complex tasks such as reconnaissance, offense, and defense, different formation shapes such as horizontal, column, echelon, and V formation shape can be used. A reasonable formation can improve aerodynamic efficiency, reduce energy consumption, and extend flight distance [1].

The typical control methods of formation control include leader-follower strategy, behavior-based method, virtual structure, Model Predictive Control (MPC), consensus theory, etc. SASKA [2] realized formation flying with the leader-follower control method based on the on-board visual perception equipment. Duan [3, 4] proposed an improved multi-objective pigeon-inspired optimization algorithm to design a multi-UAV obstacle avoidance control algorithm. Based on the back-stepping design, they proposed a consensus control algorithm for multi-rotor formation control. Research on MPC includes strategies to reduce the amount of calculation, nonlinear predictive control theory, and so on. Compared with the centralized model predictive control to solve the multi-constraint optimization problem as a whole, the distributed model predictive control can decompose the complexity of the overall optimization. A trigger mechanism can be combined

into the MPC scheme to decrease the update frequency of control inputs and reduce the computational burden.

Brain Storming Optimization (BSO) algorithm is inspired by the brain storm conference [5]. The algorithms for improving BSO can be divided into two types. One is to change the cluster method and replace the k-means cluster with other clustering methods. The other is to change the update formula of the solution. The predator and prey mechanism can be introduced, and then perform a chaotic search on each updated solution [6]. Chaotic theory can be used to change the random one-dimensional solution in a randomly selected cluster. Multi-branch chaotic mutation can be used to generate a chaotic mutation operator [7, 8]. These methods mainly focus on the cluster method and the update method of the solution, and pay less attention to the replacement mechanism of the solution and the generation mechanism of the auxiliary solution in BSO algorithm.

## 2 Modified BSO Based Distributed MPC Method

### 2.1 Model

Assuming that there is a UAV  $i$  in a certain height horizontal plane, it can be regarded as a mass point, and its state  $z_i = [x_i, y_i, \theta_i, v_i]^T$  is

$$\begin{cases} \dot{x}_i = v_i \cos \theta_i \\ \dot{y}_i = v_i \sin \theta_i \\ \dot{\theta}_i = \omega_i \\ \dot{v}_i = a_i \end{cases} \quad i = 1, 2, \dots, N_{uav} \quad (1)$$

where the  $x_i, y_i$  is the position of UAV  $i$ ,  $v_i$  indicates the linear velocity and  $a_i$  is the acceleration.  $\theta_i$  indicates the yaw angle and  $\omega_i$  is the angular velocity.

### 2.2 Distributed MPC

Compared with centralized model predictive control, distributed model predictive control has stronger ability to deal with multi-input and multi-output systems with state and control constraints. It transforms the control problem into a group of control problems of sub-systems, and constructs multiple distributed predictive platforms with information interaction to solve the optimization problem. Establish a distributed model predictive control framework for the UAV formation system, the UAV system  $i \in V = \{1, 2, \dots, N_{uav}\}$  can be expressed as

$$\dot{z}_i(t) = f_i(z_i(t), u_i(t)), \quad t \geq t_0, \quad z_i(t_0) = z_{i0} \quad (2)$$

where  $z_i(t) \in \mathcal{R}^n$  is the state of UAV  $i$  at time  $t$ , and  $u_i(t) \in \mathcal{R}^m$  is the control input of the UAV  $i$ .  $z_{i0}$  is the initial state,  $t_0$  is the initial time, and  $N_{uav}$  is the number of UAVs.

The neighbors of the UAV  $i$  in the set  $\mathcal{N}_i$  can be expressed as

$$\dot{z}_{-i}(t) = f_{-i}(z_{-i}(t), u_{-i}(t)), \quad t \geq t_0 \quad (3)$$

where  $u_{-i}(t) = \{u_j(t)\}$ ,  $z_{-i}(t) = \{z_j(t)\}$ ,  $j \in \mathcal{N}_i$ . And the optimal predictive control input and cost function of the UAV  $i$  are

$$u_i^*(s; t_c) = \arg \min_{u_i^*(s; t_c)} J_i(z_i(s; t_c), z_{-i}(s; t_c), u_i(s; t_c)) \quad (4)$$

$$J_i = \int_{t_c}^{t_c+T_p} F_i(z_i^p(s; t_c), \hat{z}_{-i}(s; t_c), u_i^p(s; t_c)) ds \quad (5)$$

where the prediction horizon  $T_p \in (0, \infty)$  is a constant, and the control update period  $\delta_T \in (0, T_p]$  is selected based on requirements. The update time is defined as  $t_c = t_0 + \delta_T c$ ,  $c \in \{0, 1, 2, \dots\}$ . At each update instant,  $J_i$  is a distributed cost function,  $F_i$  is a function determined by task requirements,  $z_i^p(s; t_c)$  is the predictive state of the UAV  $i$  and  $\hat{z}_{-i}(s; t_c)$  is the estimated state of its neighbors. The estimated state is composed of two parts, the optimal prediction state  $z_j^*(s; t_{c-1})$  in  $s \in [t_c, t_{c-1} + T_p)$  and the optimal prediction state  $z_j^*(t_{c-1} + T_p; t_{c-1})$  in  $s \in [t_{c-1} + T_p, t_c + T_p]$ .

### 2.3 Modified BSO

Firstly, an initial solution is randomly generated in the solution space, and each solution can be expressed as  $X_i = \{X_i^1, X_i^2, \dots, X_i^D\}$ ,  $i \in \{1, 2, 3, \dots, N\}$ ,  $D$  is the dimension of solution space,  $X_i^1, X_i^2, \dots, X_i^D$  are coordinates of the solution in each dimension of the solution space. The range of the solution is limited by the solution space  $X_i \in [X_{\min}, X_{\max}]$ , where  $X_{\min}$  is the lower limit of the solution space and  $X_{\max}$  is the upper limit of the solution space.

K-means cluster is used to divide  $N$  individuals into  $m$  clusters. In each cluster  $C_j$ ,  $j \in \{1, 2, 3, \dots, m\}$ , individuals are sorted according to the value of cost function, and the optimal individual is the cluster center  $X_{C_j}$ .

Replace the solution with probability  $p_r$ , and randomly generate a random number  $p_{r0}$  between  $[0, 1]$ . If  $p_{r0} < p_r$ , select a cluster center  $X_{C_{j_r}}$  randomly and replace it with chaotic solution, where  $j_r \in \{1, 2, 3, \dots, m\}$  is the randomly selected. A  $D$ -dimension vector  $cr(0)$  is randomly generated as the seed of the sequence. A chaotic map  $f_{rmap}$  is selected for  $n_r$  times mapping, and the chaotic sequence  $\{cr(1), cr(2), \dots, cr(n_r)\}$  is generated by formula

$$cr(i_r) = f_{rmap}(cr(i_r - 1)), i_r = 1, 2, \dots, n_r \quad (6)$$

The corresponding chaotic solution sequence  $\{cx(1), cx(2), \dots, cx(n_r)\}$  is obtained by formula

$$cx(i_r) = cr(i_r)(X_{\max} - X_{\min}) + X_{\min} \quad (7)$$

The cost function value of each chaotic solution is calculated. The chaotic solution  $cx_{best}$  with the optimal cost function value is to replace the selected cluster center  $X_{C_{j_r}} = cx_{best}$ .

Generate an auxiliary solution  $X_{ih}$  for each solution  $X_i$ . In the generation of auxiliary solution  $\{X_{C_1}, X_{C_2}, \dots, X_{C_m}\}$  all cluster centers are considered. A  $m$ -dimension vector

$ch(0) = [ch_1(0), ch_2(0), \dots, ch_m(0)]$  is randomly generated as the seed of the sequence, and a chaotic map  $f_{hmap}$  is selected for secondary mapping. The chaotic weight coefficient sequence  $\{ch(1), ch(2), \dots, ch(n_h)\}$  is generated by

$$ch(i_h) = f_{hmap}(cr(i_h - 1)), i_h = 1, 2, \dots, n_h \tag{8}$$

Use  $ch(n_h) = [ch_1(n_h), ch_2(n_h), \dots, ch_m(n_h)]$  and cluster centers to generate auxiliary solutions

$$X_{ih} = \sum_{j=1}^m (ch_j(n_h) \times X_{C_j}) \tag{9}$$

A new solution  $X_{inew}$  is generated based on each auxiliary solution  $X_{ih}$ . The update formula is

$$X_{inew} = X_{ih} + \text{logsig}((0.5 \times T_{\max} - T_{now})/k) \times X_{rand} \times \text{norm}(0, 1, D) \tag{10}$$

$\text{logsig}()$  is the sigmoid function, and the output of it is between (0, 1).  $k$  is the parameter that affects the slope of the function.  $T_{\max}$  is the maximum number of iterations,  $T_{now}$  is the current number of iterations.  $\text{norm}(0, 1, D)$  generates a normal distribution of random numbers to form a D-dimensional vector. If the value of cost function of  $X_{inew}$  is better than that of  $X_i$ , update the solution  $X_i = X_{inew}$ .

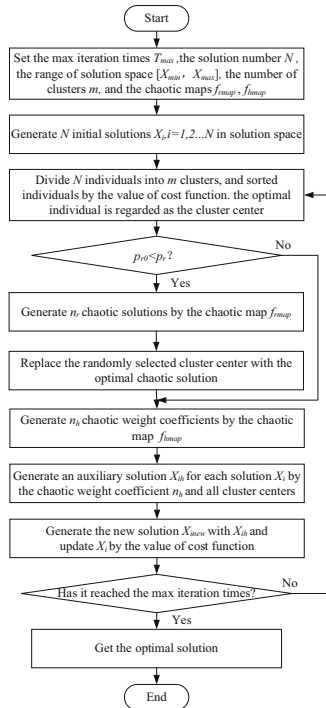


Fig. 1. The flow chart of the modified BSO algorithm

It is checked whether the maximum number of iterations is reached, and if reached, the optimal solution is output as the optimization result, otherwise, iterative optimization is performed until the maximum number of iterations is reached. The flow chart of the modified BSO algorithm is shown in Fig. 1.

### 2.4 Cost Functions and Trigger Mechanism

#### Cost Functions

The cost function for multiple UAVs to maintain the desired formation according to the desired trajectory are

$$F_i(z_i(t), u_i(t)) = w_{idst}F_{idst}(z_i(t), u_i(t)) + w_{itrj}F_{itrj}(z_i(t), u_i(t)) \tag{11}$$

$$F_{idst}(z_i(t), u_i(t)) = \sum_{j \in \mathcal{N}_i} \|p_{ij}(t) - p_{ij}^d(t)\|^2 \tag{12}$$

$$F_{itrj}(z_i(t), u_i(t)) = \|p_c(t) - p_c^d(t)\|^2 \tag{13}$$

where  $F_{idst}$  is the formation distance cost function of UAVs.  $p_{ij}$  is the distance between UAV  $i, j$ , and  $p_{ij}^d$  is the desired distance between UAV  $i, j$ .  $F_{idst}$  is the cost function of tracking the reference trajectory.  $p_c = \sum_{i \in V} p_i / N_{uav}$  is the center position of the formation.  $p_i$  is the position of the drone, and  $p_c^d$  is the desired position of the formation center.  $w_{idst}, w_{itrj}$  are the weight constants. The distributed cost function of UAV  $i$  in  $s \in [t_c, t_c + T_p]$  is

$$\begin{aligned} &F_i(z_i^p(s; t_c), \hat{z}_{-i}(s; t_c), u_i^p(s; t_c)) \\ &= w_{idst} \sum_{j \in \mathcal{N}_i} \|p_i^p(s; t_c) - \hat{p}_j(s; t_{c-1}) - p_{ij}^d(s; t_c)\|^2 \\ &+ w_{itrj} \left\| \frac{1}{N_{uav}} \left( p_i^p(s; t_c) + \sum_{j \in \mathcal{N}_i} \hat{p}_j(s; t_{c-1}) \right) - p_c^d(s; t_c) \right\|^2, \end{aligned} \tag{14}$$

$s \in [t_c, t_c + T_p]$

where  $p_i^p(s; t_c)$  is the predictive position of the UAV, and  $\hat{p}_j(s; t_{c-1})$  is the estimated position of the UAV  $j$ .  $p_{ij}^d(s; t_c)$  represents the desired distance between the UAV  $i$  and  $j$ , and  $p_c^d(s; t_c)$  is the desired position of the formation center, determined by the reference trajectory.

#### Trigger Mechanism

In order to reduce the burden of calculation, a trigger mechanism is introduced. The control inputs are solved and updated when the trigger condition is met. The mechanism

of triggering the solution and update is mainly determined by the error of formation distance, the error of trajectory tracking, control input and prediction horizon.

$$\Delta p_{ij}(t_c) = \left\| p_i^p(t_c) - \hat{p}_j(t_{c-1}) - p_{ij}^d(t_c) \right\| - \varepsilon_p, j \in \mathcal{N}_i \quad (15)$$

$$\Delta p_O(t) = \left\| \frac{1}{N} \left( p_i^p(t) + \sum_{j \in \mathcal{N}_i} \hat{p}_j(t) \right) - p_c^d(t) \right\| - \varepsilon_O \quad (16)$$

$$\Delta u_i(t_c) = u_i^p(t_c) - \eta u_{\max}, u_{\max} = [a_{\max}, \omega_{\max}]^T \quad (17)$$

$$\Delta T_i(t_c) = t_c - (t_{trg}^i + T_p) \quad (18)$$

where  $\varepsilon_p$  is the threshold of predictive formation distance error, and  $\varepsilon_O$  is the threshold of formation tracking error.  $u_{\max} = [a_{\max}, \omega_{\max}]^T$  is the limitation of the control input.  $u_i^p(t_c)$  is the predictive control input of the UAV  $i$ , and the range of  $\eta$  is from 0 to 1.  $t_{trg}^i$  is the last update instant of the UAV  $i$ .  $\Delta p_{ij}(t_c)$  is the difference between the formation distance error and the corresponding threshold.  $\Delta p_O(t)$  is the difference between the trajectory tracking error and the corresponding threshold.  $\Delta u_i(t_c)$  is the difference between the control input and the corresponding threshold.  $\Delta T_i(t_c)$  is the difference between  $t_c$  and the prediction horizon  $t_{trg}^i + T_p$  at the last update instant  $t_{trg}^i$ . The update condition can be described as

$$\Delta p_{ij}(t_c) \geq 0 \parallel \Delta p_O(t) \geq 0 \parallel \Delta u_i(t_c) \geq 0 \parallel \Delta T_i(t_c) \geq 0 \quad (19)$$

When the condition is met, the solution is updated. otherwise the last optimal control input is used. The flow chart of modified BSO based distributed MPC method is shown in Fig. 2.

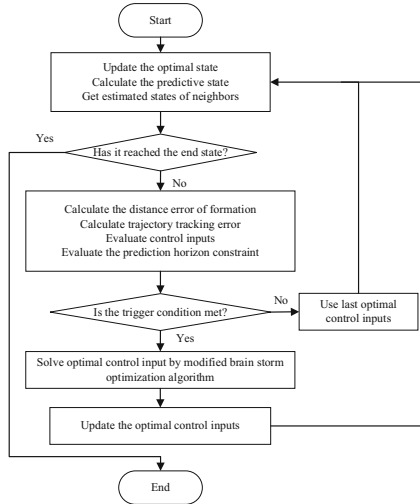


Fig. 2. The framework of modified BSO based distributed MPC method

### 3 Numerical Simulations

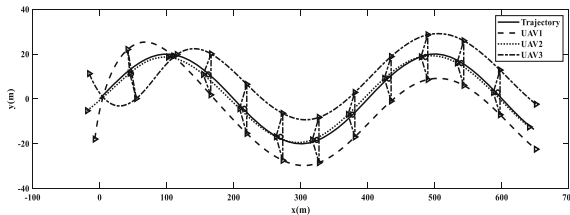
In this section, numerical simulations are used to verify the performance of the designed method. The formation control of three UAVs are considered in the simulation scenario. The initial states of UAVs are shown in Table 1.

**Table 1.** Initial states of UAVs

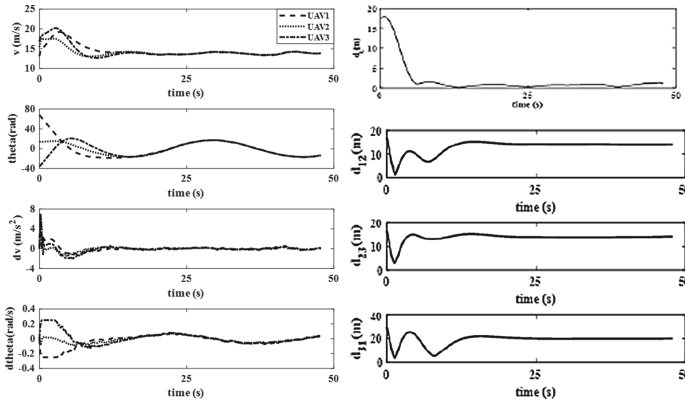
UAV	x (m)	y (m)	v (m/s)	$\theta$ ( $^\circ$ )
1	-7.08	-17.96	13.12	68.02
2	-18.48	-5.25	15.43	14.74
3	-15.46	11.29	16.83	-35.97

The prediction horizon is set as  $T_p = 4$  s, and the update period is  $\delta_T = 0.2$  s. The acceleration range is  $[-7 \text{ m/s}^2, +7 \text{ m/s}^2]$ , and the angular velocity range is  $[-0.25 \text{ rad/s}, +0.25 \text{ rad/s}]$ . The desired formation shape is an isosceles right triangle and the length of its hypotenuse is 20 m.

The numerical simulation is carried out without trigger mechanism. The trajectory of formation is shown in Fig. 3. It can be seen that the UAV can form the desired formation, and the formation center can track the reference trajectory. The states of each UAV and the distances of formation are shown in Fig. 4. As the formation is formed, the velocity and yaw angle of each UAV gradually converge, and the acceleration and angular velocity also tend to be the same. The formation center moves to the reference trajectory quickly in the early stage. The trajectory tracking error is greatly reduced, and then fluctuated around 0. The maximum error is within 1.25 m. After the formation is formed, the distance  $d_{31}$  between UAV 1 and UAV 3 is kept near 20 m, and the distance  $d_{12}, d_{23}$  are kept near 14.14 m. These errors are within 0.2 m.

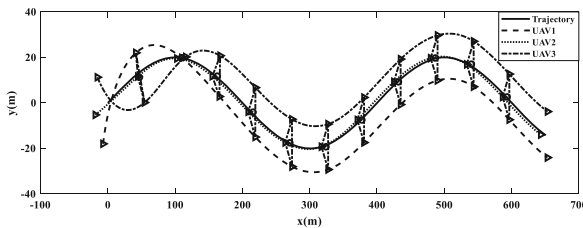


**Fig. 3.** The trajectory of formation



**Fig. 4.** States and distances of formation

The numerical simulation is carried out with trigger mechanism, and the thresholds are set as  $\varepsilon_p = 2$  m,  $\varepsilon_O = 2$  m,  $\eta = 1$ . The trajectory of formation with trigger mechanism is shown in Fig. 5. It can be seen that the UAV can form the desired formation, and the formation center can track the reference trajectory. The states of each UAV and the distances of formation with trigger mechanism are shown in Fig. 6. As the formation is formed, the velocity and yaw angle of each UAV gradually converge, and the acceleration and angular velocity also tend to be the same. The formation center moves to the reference trajectory quickly in the early stage. The trajectory tracking error is greatly reduced, and then fluctuates around 0. The maximum error is within 1.4 m. After the formation is formed, the distance is kept near 20 m, and the error is within 0.25 m. The distance  $d_{12}$ ,  $d_{23}$  are kept near 14.14 m. These errors are within 0.3 m. The trigger instants of each UAV are shown in Fig. 7. It can be seen that the control inputs of each UAV is solved asynchronously. When the formation shape has not been formed in the early stage, the control inputs are solved and updated frequently, and then the numbers of triggering times are reduced. There are 240 update instants, and UAV 1, UAV 2 and UAV 3 trigger 157, 155 and 160 times respectively, reducing the number of solving the control problem by more than 33.3%, which lessens the computational burden.



**Fig. 5.** The trajectory of formation with trigger mechanism



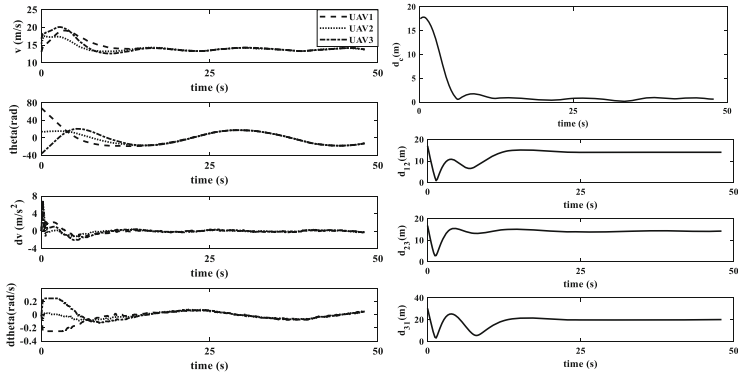


Fig. 6. The states of each UAV and the distances of formation with trigger mechanism

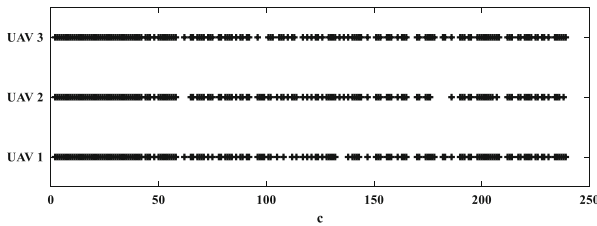


Fig. 7. The trigger instants of each UAV

## 4 Conclusions

This paper proposes a modified BSO algorithm, which introduces the chaos theory to the replacement mechanism of the solution and considers all cluster centers in the update mechanism of the solution. Based on the distributed MPC, the cost function and trigger mechanism are designed, and the proposed algorithm is used to solve the control inputs of each UAV, forming a distributed MPC method based on the modified BSO. The cost function and trigger mechanism are designed, and the proposed method is used to solve the control problem of UAV formation. Numerical simulation results show the feasibility and effectiveness of the method. The designed trigger mechanism reduces the number of update times to less than two-thirds, and reduces the amount of calculation. Further research will improve the trigger mechanism and carry out simulation analysis on control problems of distributed heterogeneous UAVs.

**Acknowledgements.** This work was supported by the Science and Technology Innovation 2030-Key Project of New Generation Artificial Intelligence under Grant 2020AAA0108200.

## References

1. Seiler, P., Pant, A., Hedrick, K.: Analysis of bird formations. *Decis. Control* **1**, 118–123 (2002)
2. Saska, M., et al.: System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization. *Auton. Robot.* **41**(4), 919–944 (2016)
3. Qiu, H., Duan, H.: A multi-objective pigeon-inspired optimization approach to UAV distributed flocking among obstacles. *Inf. Sci.* **509**, 515–529 (2020)
4. Du, H., Zhu, W., Wen, G., Duan, Z., Lü, J.: Distributed formation control of multiple quadrotor aircraft based on non-smooth consensus algorithms. *IEEE Trans. Cybern.* **49**(1), 342–353 (2019)
5. Shi, Y.: Brain storm optimization algorithm. In: Tan, Y., Shi, Y., Chai, Yi., Wang, G. (eds.) *ICSI 2011*. LNCS, vol. 6728, pp. 303–309. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-21515-5\\_36](https://doi.org/10.1007/978-3-642-21515-5_36)
6. Qiu, H., Duan, H., et al.: Chaotic predator-prey brain storm optimization for continuous optimization problems. In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–7. IEEE (2017)
7. Tuba, E., Dolicanin, E., Tuba, M.: Chaotic brain storm optimization algorithm. In: Yin, H., et al. (eds.) *IDEAL 2017*. LNCS, vol. 10585, pp. 551–559. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-68935-7\\_60](https://doi.org/10.1007/978-3-319-68935-7_60)
8. Cheng, S., Zhang, M., et al.: Brain storm optimization algorithm for solving knowledge spillover problems. *Neural Comput. Appl.* 1–14 (2021)