



HAL
open science

A Hybrid Parallel Algorithm With Multiple Improved Strategies

Tingting Wang, Jeng-Shyang Pan, Pei-Cheng Song, Shu-Chuan Chu

► **To cite this version:**

Tingting Wang, Jeng-Shyang Pan, Pei-Cheng Song, Shu-Chuan Chu. A Hybrid Parallel Algorithm With Multiple Improved Strategies. 12th International Conference on Intelligent Information Processing (IIP), May 2022, Qingdao, China. pp.228-242, 10.1007/978-3-031-03948-5_19 . hal-04178751

HAL Id: hal-04178751

<https://inria.hal.science/hal-04178751v1>

Submitted on 8 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

A Hybrid Parallel Algorithm with Multiple Improved Strategies

Tingting Wang¹, Jeng-Shyang Pan^{1,2}[0000-0002-3128-9025], Pei-Cheng Song¹,
and Shu-Chuan Chu^{1,3}[0000-0003-2117-0618]*

¹ College of Computer Science and Engineering, Shandong University of Science and
Technology, Qingdao, China

17864198822@163.com, jengshyangpan@gmail.com, spacewe@outlook.com,
scchu0803@gmail.com

² Department of Information Management, Chaoyang University of Technology,
Taichung, Taiwan

jengshyangpan@gmail.com

³ College of Science and Engineering, Flinders University, Bedford Park, Australia
scchu0803@gmail.com

Abstract. This paper proposes a novel hybrid parallel algorithm with multiple improved strategies. The whole population is divided into three subpopulations and each sub-population executes butterfly optimization algorithm, grey wolf optimization algorithm, and marine predator algorithm respectively. Meanwhile, they share information through three different communication strategies. And in order to improve the performance of the algorithm, the text uses the cubic chaotic mapping mechanism in the initialization stage. At the same time, the idea of adaptive parameter strategy is also introduced, so that some hyperparameters are changed along with the iteration. The results show that the algorithm can provide very competitive results, and is superior to the best algorithm in the literature on most test functions.

Keywords: Parallel · Cubic chaotic mapping · Adaptive parameter.

1 Introduction

In recent years, in order to solve a variety of practical problems, a large number of meta heuristic algorithms have been proposed one after another. [21] and [1] use particle swarm optimization (PSO) [19, 34, 8] to find the best deployment of a group of sensors in the monitoring area to achieve the best coverage of the network [17, 38, 4]. [11] and [30] use quasi affine transformation evolutionary algorithm (Quatre) [25, 35, 24]. And an improved binary pigeon inspired optimization algorithm (PIO) [12, 10, 16] to solve the feature selection problem. Pan et al. [29] improved the states of matter search (SMS) [9, 20, 27] algorithm that hides the watermark in QR code [37, 18, 23]. Based on fractional micro product theory, an

* Corresponding author

Email address: scchu0803@gmail.com

improved fish migration optimization (FMO) [31, 15] is proposed to adjust PID controller[36, 33, 22]. There are many such examples, such as butterfly optimization algorithm (BOA)[7, 6, 39, 5], grey wolf optimization algorithm (GWO)[26, 13, 28, 32], and marine predator algorithm (MPA)[14, 3, 2] are also three commonly used intelligent optimization algorithms. The three algorithms are easy to implement in code and have achieved good results in test function, but there is still room for improvement. Therefore, we consider proposing a new algorithm combining the advantages of the three algorithms. The major contributions of this paper are as follows:

1. A population initialization method based on cubic chaotic mapping is proposed to replace the pseudo-random number generator to generate chaotic numbers between 0 and 1.
2. A parameter adaptive method is proposed to make the parameter setting more reasonable for the solution of the optimal value.
3. A parallel strategy combining three communication strategy is proposed to make the algorithm find the optimal solution more quickly and accurately.

The rest of the paper is arranged as follows. In the Sec.2 , we will briefly review the butterfly optimization algorithm, grey wolf algorithm and marine predator algorithm. Then, in Sec.3, we will first describe the concepts of cubic chaotic mapping and adaptive parameter method and then describe how to combine parallel and cubic chaotic mapping and adaptive parameter strategies with three optimization algorithms to improve the performance of the proposed algorithm. The simulation experiment description and experimental results of the improved algorithm based on three improvement strategies will be displayed in the Sec.4. In Sec.5, we will summarize the paper.

2 Related Works

2.1 Butterfly Optimization Algorithm

The algorithm represents a solution in terms of the position of a butterfly and divides the process into three phases. The first phase of the algorithm initializes the position of each butterfly and gives the values of the required hyperparameters in the algorithm. Then the algorithm enters the loop iteration phase.

In this phase, the algorithm divides the evolutionary pattern of the population into two parts, global search and local search, according to the given threshold p area. In the global search mode, the positions of the butterflies are updated according to Eq. 1.

$$x_i^{t+1} = x_i^t + (r^2 \times g^* - x_i^t) \times f_i \quad (1)$$

where x_i^t denotes the position of the i th butterfly at the t th iteration, g^* denotes the global optimal solution that can be found so far, r is a random number

belonging to the interval 0 to 1, and f_i denotes the odor produced by the i th butterfly and also the fitness of the i th solution, which is calculated using Eq. 2.

$$f = cI^a \quad (2)$$

In the local search mode, the position of butterfly i in the $(t+1)$ th generation is determined using Eq.3.

$$x_i^{t+1} = x_i^t + (r^2 \times x_j^t - x_k^t) \times f_i \quad (3)$$

where x_j and x_k are two different individuals in the population distinguished from x_i . The global search mode is switched with the local search mode by switching probability p . The threshold p is set a priori at the initialized individual.

The third stage is used to determine whether the algorithm satisfies the termination condition. If the algorithm satisfies the termination condition, the optimal value is returned. Otherwise, the loop iteration stage continues.

The above is the content of the butterfly optimization algorithm.

2.2 Grey Wolf Optimizer

Grey Wolf Optimizer (GWO) is a novel population intelligence optimization algorithm that simulates the social hierarchy and hunting behavior of the grey wolf population in nature.

In this algorithm, the social hierarchy of grey wolves can be divided into four levels: alpha wolf α , subordinate wolf β , common wolf δ and bottom wolf ω . Among them, α is responsible for leading the wolf pack, β assists α in making decisions, δ follows the instructions of α and β , and can also command the bottom individual ω . α , β and δ in the grey wolf optimization algorithm denote the historical optimal solution, the suboptimal solution and the third optimal solution, respectively, and ω denotes the remaining individuals.

When a prey is found, the iteration begins. There are mainly the best three wolves in each generation of the population (i.e. the three best solutions in the population) to guide the completion. The mathematical model of the behavior of the grey wolf, which gradually approaches the prey and encircles it when it lets the prey go, is represented in Eq. 4 to Eq.8.

$$\vec{D} = \left| \vec{C} \cdot \overrightarrow{X_p(t)} - \overrightarrow{X(t)} \right| \quad (4)$$

$$\overrightarrow{X(t+1)} = \overrightarrow{X_p(t)} - \vec{A} \cdot \vec{D} \quad (5)$$

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (6)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (7)$$

$$\vec{a} = 2 - \frac{2t}{t_{max}} \quad (8)$$

where t is the round of the current iteration, \vec{X} is the position vector of the grey wolf, and the vector \vec{X}_p represents the current position of the prey. \vec{A} and \vec{C} are the coefficient vectors. Eq. 6 and Eq. 7 give the calculation methods of \vec{A} and \vec{C} . \vec{a} is the convergence factor, and \vec{r}_1 and \vec{r}_2 obey a uniform distribution between $[0, 1]$ as the number of iterations decreases linearly from 2 to 0.

For simulating the search behavior of grey wolves(candidate solutions), during every iteration, the best three grey wolves (α, β, δ) in the present population are reserved, and then the locations of other search agents (including ω) are updated on the basis of their location information. We use eq.9 to eq.11 to simulate this behavior.

$$\begin{cases} \vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \\ \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \\ \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \end{cases} \quad (9)$$

$$\begin{cases} \vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \\ \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \\ \vec{X}_3 = \vec{X}_\omega - \vec{A}_3 \cdot \vec{D}_\omega \end{cases} \quad (10)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (11)$$

Where, D_α, D_β and D_δ respectively represent α, β and δ distance from other individuals. $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$ respectively represent the current location of the α, β and δ . \vec{C}_1, \vec{C}_2 and \vec{C}_3 are vectors which elements are composed of random numbers, and X is the current grey wolf position. When $|\vec{A}| > 1$, the algorithm enters the global optimization stage, which also means that ω wolves will escape prey and explore more space. When $|\vec{A}| < 1$, the grey wolf will concentrate on the prey in one or some areas.

2.3 Marine predator algorithm

Marine predator algorithm was proposed by Afshin et al. in 2020. The algorithm simulates the process of catching prey by marine predators.

The marine predator algorithm begins with a random number evenly distributed in the search space. It is expressed by mathematical formula as Eq. 12:

$$X_0 = X_{min} + rand \times (X_{max} - X_{min}) \quad (12)$$

Where X_{min} and X_{max} represent the lower and upper boundaries of the population, and $rand$ represents a random number which the range of is $[0,1]$. Finally, the d-dimensional population position matrix of n individuals is obtained, called the prey matrix:

$$\begin{bmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,d} \\ X_{2,1} & X_{2,2} & \dots & X_{2,d} \\ \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1} & X_{n,2} & X_{n,3} & X_{n,d} \end{bmatrix} \quad (13)$$

where $X_{i,j}$ describes the j th dimension for the i th prey. Then, for each individual $X_i = [X_{i1}, X_{i2}, \dots, X_{id}]$, calculate its fitness, and then use the individual X_i with the best fitness to copy n copies to form the elite matrix:

$$\begin{bmatrix} X_{1,1}^1 & X_{1,2}^1 & \dots & X_{1,d}^1 \\ X_{2,1}^1 & X_{2,2}^1 & \dots & X_{2,d}^1 \\ \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1}^1 & X_{n,2}^1 & X_{n,3}^1 & X_{n,d}^1 \end{bmatrix} \quad (14)$$

where d represents the dimension, X^1 represents the optimal predator vector with n simulation agents to bring the matrix, which is called *Elite*.

Then we began to optimize. There are three steps in the optimization process.

The step 1 (When the number of iterations is less than one third of the maximum number of iterations):

$$\overrightarrow{stepsize}_i = \overrightarrow{R}_B \otimes (\overrightarrow{Elite}_i - \overrightarrow{R}_B \otimes \overrightarrow{P}_i), i = 1, 2, 3, \dots, n \quad (15)$$

$$\overrightarrow{P}_i = \overrightarrow{P}_i + P \times \overrightarrow{R} \otimes \overrightarrow{stepsize}_i \quad (16)$$

Where $\overrightarrow{stepsize}_i$ represents the moving step size, and \overrightarrow{R}_B is a vector composed of random numbers which is generated by Brownian random walk. P is a constant with a value of 0.5. R is a vector of random number which is subject to a uniform distribution between 0 and 1, and its dimension is d . \overrightarrow{R}_B is equivalent to the generalized Gaussian distribution. Each element \overrightarrow{R}_{B_i} can be calculated by the following expression:

$$R_{B_i} = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (17)$$

The step 2 (When the number of iterations is greater than one-third and less than two-thirds of the maximum number of iterations.). The new rules for the first half of the population are as follows:

$$\overrightarrow{stepsize}_i = \overrightarrow{R}_L \otimes (\overrightarrow{Elite}_i - \overrightarrow{R}_L \otimes \overrightarrow{P}_i), i = 1, 2, 3, \dots, \frac{n}{2} \quad (18)$$

$$\overrightarrow{P}_i = \overrightarrow{P}_i + P \times \overrightarrow{R} \otimes \overrightarrow{stepsize}_i \quad (19)$$

where \overrightarrow{R}_L describes a random value that is distributed by the Lévy movement. Each element \overrightarrow{R}_{L_i} of \overrightarrow{R}_L can be calculated from the following formula:

$$R_{L_i} = C \times \frac{x}{y^{\frac{1}{\alpha}}} \quad (20)$$

$$x = Normal(0, \sigma_x^2) \quad (21)$$

$$y = Normal(0, \sigma_y^2) \quad (22)$$

$$\sigma_x = \left[\frac{\Gamma(1 + \alpha) \sin(\frac{\pi\alpha}{2})}{\Gamma(\frac{1+\alpha}{2}) \alpha 2^{\frac{\alpha-1}{2}}} \right]^{\frac{1}{\alpha}} \quad (23)$$

$$\sigma_y = 1 \quad (24)$$

$$\alpha = 1.5 \quad (25)$$

Where, the value of C is 0.05 and the value of a is 1.5, the value of x and y are represented by the Eq. 21 and Eq.22. The new rules for the second half of the population are as follows:

$$\overrightarrow{stepsize}_i = \overrightarrow{R}_B \otimes (\overrightarrow{Elitem}_i - \overrightarrow{R}_B \otimes \overrightarrow{P}_i), i = 1, 2, 3, \dots, \frac{n}{2} \quad (26)$$

$$\overrightarrow{P}_i = \overrightarrow{Elitem}_i + P \times CF \otimes \overrightarrow{stepsize}_i \quad (27)$$

$$CF = \left(1 - \frac{iteration}{Maxiteration}\right)^{\frac{2 \times iteration}{Maxiteration}} \quad (28)$$

The step 3 (When the number of iterations is greater than two-thirds of the maximum number of iterations):

In this stage, the population update rules are as follows:

$$\overrightarrow{stepsize}_i = \overrightarrow{R}_L \otimes (\overrightarrow{R}_L \otimes \overrightarrow{Elitem}_i - \overrightarrow{P}_i), i = 1, 2, 3, \dots, n \quad (29)$$

$$\overrightarrow{P}_i = \overrightarrow{Elitem}_i + P \times CF \otimes \overrightarrow{stepsize}_i \quad (30)$$

Fish aggregation devices (FADS) or eddy current effect usually change the foraging behavior of marine predators. Hence, in order to avoid the algorithm falling into local optimization, jump operation is added in the paper. This is formulated as follows:

$$\begin{cases} \overrightarrow{P}_i = \overrightarrow{P}_i + CF[X_{min} + \overrightarrow{R}_L \otimes (X_{max} - X_{min})] \otimes \overrightarrow{U} & r \leq FADs \\ \overrightarrow{P}_i + [FADs(1 - r) + r](\overrightarrow{P}_{r_1} - \overrightarrow{P}_{r_2}) & r > FADs \end{cases} \quad (31)$$

where the (FADs) is the influence probability, taken as 0.2. U is the binary vector. r is the random value within $[0, 1]$. r_1, r_2 are the random indices of the prey matrix, respectively. The lower and upper bounds of a dimension are represented by X_{min} and X_{max} .

3 Our proposed hybrid parallel algorithm with multiple improved strategies

In this section, we propose a hybrid algorithm called ACHBGM that combines the advantages of the butterfly optimization algorithm, the grey wolf optimization algorithm, and the marine predator optimization algorithm, and incorporates the idea of parallel mechanisms. At the same time, for improving the performance of the algorithm, we use cubic chaotic mapping in the initialization stage. In the iterative stage, some hyperparameters used in the algorithm are adaptively improved. The specific three strategies are as follows.

Table 1. The test function we use.

Function	Example	Range	f_{min}
F1	$\sum_{i=1}^{Dim} x_i^2$	$[-100, 100]$	0
F2	$\sum_{i=1}^{Dim} x_i + \prod_{i=1}^{Dim} x_i $	$[-10, 10]$	0
F3	$\sum_{i=1}^{Dim} (\sum_{j=1}^i x_j)^2$	$[-100, 100]$	0
F4	$\max\{ x_i , 1 \leq i \leq Dim\}$	$[-10, 10]$	0
F5	$\sum_{i=1}^{Dim} (x_i + 0.5)^2$	$[-10, 10]$	0
F6	$\sum_{i=1}^{Dim} Dim \cdot x_i^2 + rand(0, 1)$	$[-1.28, 1.28]$	0
F7	$\exp(0.5 \sum_{i=1}^{Dim} x_i)$	$[-10, 10]$	0
F8	$\sum_{i=1}^{Dim} x_i (i + 1)$	$[-1, 1]$	0
F9	$\sum_{i=1}^{Dim} (Dim \cdot x_i^2)$	$[-10, 10]$	0
F10	$\sum_{i=1}^{Dim} (100(x_{i+1}x_i^2) + (x_i - 1)^2)$	$[-5, 10]$	0
F11	$\sum_{i=1}^{Dim} x_i^2 + (\sum_{i=1}^{Dim} 0.5ix_i)^2 + (\sum_{i=1}^{Dim} 0.5ix_i)^4$	$[-5, 10]$	0
F12	$(x_i - 1)^2 + \sum_{i=1}^{Dim} i \cdot (2x_i^2 - x_{i-1})^2$	$[-10, 10]$	0
F13	$\sum_{i=1}^{Dim} (10^6)^{(i-1)/(Dim-1)} \cdot x_i^2$	$[-100, 100]$	0
F14	$x_i^2 + 10^6 \sum_{i=2}^{Dim} x_i^2$	$[-100, 100]$	0
F15	$\sum_{i=1}^{Dim} [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	0
F16	$\sum_{i=1}^{Dim} [y_i^2 - 10 \cos(2\pi y_i) + 10]$	$[-5.12, 5.12]$	0
-	$y_i = \begin{cases} x_i, & x_i < 0.5 \\ round(2x_i)/2, & x_i > 0.5 \end{cases}$	000	0
F17	$-20 \exp(-0.2 \sqrt{\frac{1}{Dim} \sum_{i=1}^{Dim} x_i^2}) + \exp(\frac{1}{Dim} \sum_{i=1}^{Dim} \cos(2\pi x_i)) + 20 + \exp(1)$	$[-50, 50]$	0
F18	$\frac{1}{4000} \sum_{i=1}^{Dim} x_i^2 - \prod_{i=1}^{Dim} \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$	0
F19	$\sum_{i=1}^{Dim} x_i \cdot \sin(x_i) + 0.1x_i$	$[-10, 10]$	0
F20	$\frac{\pi}{Dim} \{ \sum_{i=1}^{Dim-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_{Dim-1})^2 + 10 \sin^2(\pi y_1) \} + \sum_{i=1}^{Dim} u(x_i, 10, 100, 4)$ $\begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < a \end{cases}$	$[-100, 100]$	0
-	$y_i = 1 + (x_i + 1)/4, u_{y_i, a, k, m} = \begin{cases} 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < a \end{cases}$	000	0
F21	$\frac{1}{10} \{ \sin^2(\pi x_1) + \sum_{i=1}^{Dim-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_{Dim-1})^2 (1 + \sin^2(2\pi x_{i+1})) \} + \sum_{i=1}^{Dim} u(x_i, 5, 100, 4)$	$[-100, 100]$	0
F22	$\sum_{i=1}^{Dim} x_i \cdot \sin(\sqrt{ x_i }) $	$[-100, 100]$	0
F23	$\sin^2(3\pi x_i) + \sum_{i=1}^{Dim-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + x_{Dim} - 1 \cdot [1 + \sin^2(2\pi x_{Dim})]$	$[-10, 10]$	0
F24	$\sum_{i=1}^{Dim} (\sum_{k=0}^{k_{max}} [\alpha^k \cos(2\pi b^k(x_i + 0.5))]) - Dim \cdot \sum_{k=0}^{k_{max}} [\alpha^k \cos(2\pi b^k \cdot 0.5)], a = 0.5, b = 3, k_{max} = 20$	$[-1, 1]$	0
F25	$1 - \cos(2\pi \sqrt{\sum_{i=1}^{Dim} x_i^2}) + 0.1 \sqrt{\sum_{i=1}^{Dim} x_i^2}$	$[-100, 100]$	0
F26	$\sum_{i=1}^{Dim} [x_i^2 + 2x_{i+1}^2 - 0.3 \cdot \cos(3\pi x_i)]$	$[-10, 10]$	0

3.1 Cubic chaotic mapping

The cubic mapping method [40] with faster iteration speed to improve the coverage space of the initial solution. The calculation method is shown in Eq.32.

$$z_{n+1} = \alpha z_n^3 - \beta z_n \quad (32)$$

where α and β represent chaotic factors. When β belongs to interval $(2.3, 3)$, cubic mapping shows chaotic phenomenon. When $\alpha = 1$, the cubic map is located in the interval $(-2, 2)$, and when $\alpha = 4$, the chaotic sequence is located in the interval $(-1, 1)$. Therefore, the chaotic sequence can also be written as:

$$z_{n+1} = \rho z_n(1 - z_n^2) \quad (33)$$

where ρ is the control parameter. In Eq.32, the cubic mapping is located in the interval $(0, 1)$. When $\rho = 2.595$, the generation of chaotic variable Z_n has better ergodicity.

3.2 Parameter adaptive strategy

At the beginning of the iteration of butterfly optimization algorithm, the attraction of fragrance to each butterfly should be relatively large, and this attraction should decrease with the iteration. Therefore, in order to reflect the dynamic impact of fragrance on each butterfly, in this paper, the parameter a in Eq.2 is processed adaptively. The mathematical expression of the improved parameter a is as Eq.34:

$$a = 0.05 + 0.05 \times \left(1 - \frac{t}{T}\right) \quad (34)$$

where, t and T represent the current and the maximum value of iteration round of the algorithm respectively.

3.3 Parallel strategy

Similar to most evolutionary calculations, at the beginning, we initialize the population $P = \{p_1, p_2, \dots, p_n\}$. Then, the population was divided into three parts, namely $P_{butterfly}$, P_{wolf} and P_{prey} . The subpopulation $P_{butterfly}$ uses the evolutionary mechanism of BOA, i.e., Eq.1, Eq.2, and Eq.3, to perform iterations to find the best solution. Similarly, subpopulations P_{wolf} and P_{prey} use the evolutionary approach of GWO and MPA to find the optimal solution, respectively. Then, the algorithm performs communication between different subpopulations every 50 iterations, and a total of three main communication strategies are used in this algorithm.

Algorithm 1 Proposed algorithm

```

1: Population  $P$  is initialized by cubic mapping using Eq.33
2: Population  $P$  was divided into three subpopulations,  $P_{butterfly}$ ,  $P_{wolf}$  and  $P_{prey}$ 
3: repeat
4:    $n \leftarrow n + 1$ 
5:   Update the hyperparameter  $a$  in the BOA algorithm according to Eq.34
6:   Update  $P_{butterfly}$  based on BOA
7:   Update  $P_{wolf}$  based on GWO
8:   Update  $P_{prey}$  based on MPA
9:   if  $n = R$  then
10:    Algorithm 2, algorithm 3 and algorithm 4 are used for parallel communication
    among populations respectively
11:    Strategy 1:
12:       $P_{butterfly}^{worse} \leftarrow P_{wolf}^{best}$ 
13:       $P_{wolf}^{worse} \leftarrow P_{prey}^{best}$ 
14:       $P_{prey}^{worse} \leftarrow P_{butterfly}^{best}$ 
15:    Strategy 2:
16:       $P_{butterfly}^{worse} \leftarrow (P_{wolf}^{best} + P_{prey}^{best} + P_{butterfly}^{best})/3$ 
17:       $P_{wolf}^{worse} \leftarrow (P_{wolf}^{best} + P_{prey}^{best} + P_{butterfly}^{best})/3$ 
18:       $P_{prey}^{worse} \leftarrow (P_{wolf}^{best} + P_{prey}^{best} + P_{butterfly}^{best})/3$ 
19:    Strategy 3:
20:       $P_{butterfly}^{worse} \leftarrow (\alpha \cdot P_{wolf}^{best} + \beta \cdot P_{prey}^{best} + \gamma \cdot P_{butterfly}^{best})/3$ 
21:       $P_{wolf}^{worse} \leftarrow (\alpha \cdot P_{wolf}^{best} + \beta \cdot P_{prey}^{best} + \gamma \cdot P_{butterfly}^{best})/3$ 
22:       $P_{prey}^{worse} \leftarrow (\alpha \cdot P_{wolf}^{best} + \beta \cdot P_{prey}^{best} + \gamma \cdot P_{butterfly}^{best})/3$ 
23:    end if
24:  until The termination conditions are met.
25: Update  $P^{best} \leftarrow \max\{P_{butterfly}^{best}, P_{wolf}^{best}, P_{prey}^{best}\}$ 
Output:  $P^{best}$ 

```

Complementary circle communication strategy After the algorithm enters the iterative stage, three different subpopulations evolve independently according to their own evolution mechanism. All subpopulations share information every R generation. The communication strategy of complementary circle communication advantages aims to make the advantages of one sub-population compensate for the disadvantages of another subpopulation, which means that the best solution found in subpopulation $P_{butterfly}$ is used to improve the poor solution of P_{wolf} , and the individual with the best fitness in P_{wolf} is used to enhance the individual with the poor fitness in P_{prey} . Similarly, the best individual in P_{prey} replaces the best individual in $P_{butterfly}$.

Local average communication strategy Similar to the complementary circle communication strategy, the three subpopulations independently evolved to the R generation and communicated among the subpopulations. Different from the advantage complementary communication strategy, this strategy uses the

Table 2. The result of proposed algorithms on F1-F13.

F		BOA	GWO	MPA	ACHBGM-1	ACHBGM-2	ACHBGM-3
F1	AVG	8.20267E-11	4.05245E-15	3.2765E-19	7.70177E-45	0	3.40456E-69
	STD	5.6956E-12	4.7473E-15	2.57326E-19	1.34914E-44	0	1.5152E-68
	MIN	6.90818E-11	4.5932E-16	3.95248E-20	2.74006E-55	0	9.61134E-92
F2	AVG	4.31622E+49	1.60445E-09	2.05062E-11	2.15129E-39	0	2.0729E-108
	STD	1.11712E+50	5.86795E-10	1.52432E-11	2.36608E-39	0	9.0859E-108
	MIN	1.05202E+44	8.63671E-10	6.99273E-13	6.48284E-42	0	3.4981E-146
F3	AVG	5.84402E-11	0.920417496	0.046566631	1.54608E-44	0	7.47584E-66
	STD	5.34101E-12	1.112003472	0.086763663	2.11056E-44	0	2.68326E-65
	MIN	4.95193E-11	0.011445579	0.000958058	1.03969E-46	0	2.1551E-96
F4	AVG	3.00213E-08	0.015269914	2.96669E-08	6.25619E-39	1.0499E-262	2.61409E-91
	STD	2.36673E-09	0.025413454	1.00889E-08	6.30972E-39	0	8.04148E-91
	MIN	2.63649E-08	0.001234923	1.41054E-08	5.79104E-40	2.3903E-293	1.4696E-114
F5	AVG	22.10515943	7.464370166	1.239254488	4.12658E-08	4.22239E-08	4.48592E-08
	STD	0.79360213	0.952738108	0.409717154	1.44504E-08	1.13744E-08	1.60334E-08
	MIN	20.46727449	6.102047055	0.417215339	1.32913E-08	2.67848E-08	2.28834E-08
F6	AVG	0.001680654	0.004421525	0.001480682	1.47114E-05	3.11909E-05	6.47698E-06
	STD	0.000597359	0.002106161	0.000839343	1.06965E-05	1.99503E-05	5.9769E-06
	MIN	0.000666478	0.002321779	0.000540058	1.14752E-07	2.39528E-07	1.2051E-07
F7	AVG	1.53893E-23	4.558E-156	4.0478E-210	7.1246E-218	7.1246E-218	7.1246E-218
	STD	6.84765E-23	2.0176E-155	0	0	0	0
	MIN	4.28688E-72	6.799E-175	7.2796E-218	7.1246E-218	7.1246E-218	7.1246E-218
F8	AVG	6.17797E-14	2.56164E-86	1.08858E-62	6.18467E-54	2.6644E-186	5.13798E-50
	STD	2.51687E-14	7.5821E-86	3.25285E-62	1.16419E-53	0	2.20014E-49
	MIN	1.41682E-14	2.8828E-103	7.46253E-74	2.46334E-57	1.8133E-218	2.54353E-71
F9	AVG	8.11856E-11	1.64016E-15	1.17197E-19	1.42712E-44	0	3.01594E-68
	STD	5.42261E-12	1.4832E-15	8.49929E-20	2.04293E-44	0	1.24699E-67
	MIN	7.13377E-11	3.13488E-16	6.83108E-21	1.62396E-49	0	1.8251E-119
F10	AVG	98.88238924	97.32754713	96.31538166	90.7727665	91.83529037	90.81797196
	STD	0.03178052	1.050564781	0.578949312	0.251197016	0.515038682	0.21490587
	MIN	98.7901049	95.23559467	95.56895692	90.25036324	90.8265583	90.47588804
F11	AVG	7.89151E-11	1.86528E-15	9.86759E-20	2.35466E-44	0	4.21125E-64
	STD	6.23245E-12	1.20477E-15	1.80937E-19	3.78616E-44	0	1.79905E-63
	MIN	6.7448E-11	3.18422E-16	1.94037E-21	2.3268E-49	0	5.8041E-118
F12	AVG	0.997333524	0.666697243	0.666667641	0.666667904	0.666674412	0.666667555
	STD	0.000795827	2.97512E-05	6.01674E-07	3.97028E-07	2.24791E-05	3.23679E-07
	MIN	0.995990272	0.666682002	0.666666989	0.666667168	0.666667734	0.666667169
F13	AVG	8.08043E-22	0	1.5117E-247	3.2352E-127	0	3.9263E-121
	STD	2.37844E-21	0	0	1.0608E-126	0	1.2596E-120
	MIN	2.52239E-28	0	0	1.4376E-142	0	2.6542E-152
Sum of ranks		56	68	59	44	30	16

Table 3. The result of proposed algorithms on F14-F26.

F		BOA	GWO	MPA	ACHBGM-1	ACHBGM-2	ACHBGM-3
F14	AVG	1.1317E-17	1.6562E-248	6.62447E-92	2.03679E-66	0	1.62301E-64
	STD	1.44449E-17	0	2.94516E-91	8.2691E-66	0	7.25832E-64
	MIN	2.1006E-19	0	4.7309E-122	2.7606E-90	0	3.3786E-124
F15	AVG	7.14459E-20	0	5.4967E-114	2.51671E-94	0	9.38033E-92
	STD	1.21227E-19	0	2.458E-113	1.12545E-93	0	4.19377E-91
	MIN	9.26157E-24	0	8.6133E-145	1.3261E-116	0	1.1679E-111
F16	AVG	0	7.131474111	0	0	0	0
	STD	0	5.253730816	0	0	0	0
	MIN	0	3.22871E-10	0	0	0	0
F17	AVG	6.96471E-11	17.52327955	0	0	0	0
	STD	2.50725E-10	15.88307303	0	0	0	0
	MIN	0	3.000008061	0	0	0	0
F18	AVG	2.81675E-08	4.42485E-09	4.32296E-11	8.88178E-16	8.88178E-16	8.88178E-16
	STD	1.42709E-09	1.16068E-09	2.16022E-11	0	0	0
	MIN	2.52924E-08	2.67826E-09	9.40847E-12	8.88178E-16	8.88178E-16	8.88178E-16
F19	AVG	6.52919E-11	0.000733862	0	0	0	0
	STD	1.87636E-11	0.003281931	0	0	0	0
	MIN	3.00487E-11	2.9976E-15	0	0	0	0
F20	AVG	1.32828E-09	0.001499502	3.90313E-12	3.13406E-39	1.9055E-306	3.32752E-93
	STD	8.37764E-10	0.001208067	3.93902E-12	2.60546E-39	0	1.45374E-92
	MIN	2.45999E-10	7.97156E-09	1.3387E-13	1.38863E-41	0	6.7941E-114
F21	AVG	0.940142747	0.155537907	0.019995853	8.87375E-10	8.65685E-10	8.87682E-10
	STD	0.084449957	0.033668749	0.006444893	3.04724E-10	3.52543E-10	2.40168E-10
	MIN	0.779564979	0.110261225	0.007964981	4.40047E-10	4.70989E-10	5.38688E-10
F22	AVG	9.987075198	5.02108932	3.334885606	0.036033378	0.012635109	0.025118958
	STD	0.006545466	0.38362152	2.33191317	0.049367779	0.031738192	0.043405838
	MIN	9.969609571	4.227549961	0.738291234	2.10663E-08	2.87195E-08	2.19167E-08
F23	AVG	64.62313396	13.85256999	2.063384707	0.777461711	1.289235972	1.672479425
	STD	3.806873315	3.238347912	0.868292625	0.791874457	0.859140769	2.013159769
	MIN	57.91316802	6.155450601	0.926621119	0.05087255	0.162635606	0.075521182
F24	AVG	5.472455543	12.5501013	0	0	0	0
	STD	6.184780647	5.547453413	0	0	0	0
	MIN	0	3.870796175	0	0	0	0
F25	AVG	0.424870337	0.771090873	0.338285676	2.02948E-54	3.053E-176	3.33919E-81
	STD	0.110850249	0.221009959	0.122496568	9.0761E-54	0	1.4825E-80
	MIN	0.397983695	0.397983119	0.099495906	3.1523E-158	0	1.942E-163
F26	AVG	8.19317E-11	3.10862E-16	0	0	0	0
	STD	8.182E-12	4.10063E-16	0	0	0	0
	MIN	6.87764E-11	0	0	0	0	0
Sum of ranks		41	57	60	51	47	17

arithmetic mean of the optimal solution in each subpopulation to improve the worse individuals in the three subpopulations. Considering that there is still a small probability in the process of evolution, the local average communication strategy can effectively avoid this situation.

Global average communication strategy For solving the problem that the fitness gap between populations is too large, this paper introduces the global average communication strategy. The strategy no longer uses the arithmetic mean of the three best solutions, but gives weight to the optimal solution of each subpopulation, weights and sums the three solutions, and finally replaces the worse solution in each subpopulation. It can be seen that the local average communication strategy is a special case of this strategy.

4 Experiments and Results

In order to test the effectiveness of our proposed algorithm, we use 26 test functions to verify the performance of the algorithm, which are listed in Tab. 1. The running results of all algorithms on 26 test functions are sorted, and then the sorted sum of each algorithm is accumulated, and the accumulated results are compared. In order to show the experimental data more clearly, we divided 26 test functions into two groups. The first group contains functions $F1 \sim F13$, and the second group contains functions $F14 \sim F26$. We use two sets of test functions to test all algorithms and count the final results respectively. This paper makes three groups of comparative experiments on 26 test functions.

4.1 Comparative experiment based on adaptive strategy

In this subsection, the BOA algorithm with parameter adaptive strategy is compared with the original BOA algorithm. The experimental results show that the BOA algorithm with parameter adaptive strategy performs well on 22 functions.

4.2 Comparative experiment based on cubic chaotic mapping

In this subsection, in order to prove the effectiveness of cubic chaotic mapping initialization, the three original algorithms BOA, GWO and MPA are compared with the three algorithms after cubic chaotic map initialization. Experimental results show that the algorithm after cubic mapping initialization performs well on 20 test functions. In order to make the results more visual, then we sort the running results of all algorithms on 26 test functions, and accumulate the sorting and of each algorithm to get the ranking.

4.3 Comparative experiments of proposed algorithms

In this subsection, we run all the algorithms involved in the comparison 20 times independently. Firstly, the optimal fitness value is used as the evaluation

standard. ACHBGM with elite negotiation strategy has advantages over other functions in 19 test functions. If the average value of fitness value is taken as the evaluation standard, ACHBGM with elite negotiation strategy has achieved good results in 15 test functions. The experimental data are shown in Tab. 2 and 3. ACHBGM-1 represents the hybrid algorithm with complementary communication strategy, ACHBGM-2 represents that the hybrid algorithm adopts elite negotiation strategy, and ACHBGM-3 means that the communication strategy in the algorithm uses elite team leader strategy.

Overall, we rank the results of all algorithms on each test function, and judge the advantages and disadvantages of different algorithms by adding the ranking of each algorithm on two sets of test functions separately. Similarly, I also discuss from two indicators, one is the average fitness value, and the other is the best fitness value. Firstly, from the perspective of average fitness value, the best is ACHBGM with elite team leader strategy, the second best is ACHBGM with elite negotiation strategy, and the third is ACHBGM with complementary advantage strategy. Secondly, the best fitness value is used as the evaluation standard, ACHBGM-3 is still the top algorithm, followed by ACHBGM-1 and then ACHBGM-2. Experimental results show that the proposed hybrid algorithm is better than the relevant original algorithm under the two evaluation criteria.

5 Conclusion

This work proposes a novel hybrid algorithm ACHBGM, which integrates the idea of parallelism and combines the advantages of BOA, GWO and MPA. The introduction of cubic chaotic mapping initialization and adaptive parameter method also improves the performance of the algorithm. During the iterative process, three different populations share information with three communication strategies, so that the algorithm can jump out of the local optimization. Finally, their effectiveness of proposed algorithm is illustrated by experiments.

References

1. Aziz, N.A.B.A., Moheemmed, A.W., Alias, M.Y.: A wireless sensor network coverage optimization algorithm based on particle swarm optimization and voronoi diagram. In: 2009 international conference on networking, sensing and control. 602–607. (2009)
2. Abdel-Basset, M., Mohamed, R., Chakraborty, R.K., Ryan, M., Mirjalili, S.: New binary marine predators optimization algorithms for 0–1 knapsack problems. *Computers & Industrial Engineering* **151**, 106949. (2021)
3. Abdel-Basset, M., Mohamed, R., Elhoseny, M., Chakraborty, R.K., Ryan, M.: A hybrid covid-19 detection model using an improved marine predators algorithm and a ranking-based diversity reduction strategy. *IEEE Access* **8**, 79521–79540. (2020)
4. Abdollahzadeh, S., Navimipour, N.J.: Deployment strategies in the wireless sensor network: A comprehensive review. *Computer Communications* **91**, 1–16. (2016)

5. Arora, S., Singh, S.: An effective hybrid butterfly optimization algorithm with artificial bee colony for numerical optimization. *International Journal of Interactive Multimedia & Artificial Intelligence* **4**(4), 14–21. (2017)
6. Arora, S., Singh, S.: An improved butterfly optimization algorithm with chaos. *Journal of Intelligent & Fuzzy Systems* **32**(1), 1079–1088. (2017)
7. Arora, S., Singh, S.: Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing* **23**(3), 715–734. (2019)
8. Bratton, D., Kennedy, J.: Defining a standard for particle swarm optimization. In: 2007 IEEE swarm intelligence symposium. 120–127. (2007)
9. Cuevas, E., Echavarría, A., Ramírez-Ortegón, M.A.: An optimization algorithm inspired by the states of matter that improves the balance between exploration and exploitation. *Applied intelligence* **40**(2), 256–272. (2014)
10. Cui, Z., Zhang, J., Wang, Y., Cao, Y., Cai, X., Zhang, W., Chen, J.: A pigeon-inspired optimization algorithm for many-objective optimization problems. *Science China Information Sciences* **62**(7), 1–3. (2019)
11. Du, Z.G., Pan, T.S., Pan, J.S., Chu, S.C.: Quasi-affine transformation evolutionary algorithm for feature selection. In: *Advances in Smart Vehicular Technology, Transportation, Communication and Applications*, 147–156. (2022)
12. Duan, H., Qiao, P.: Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. *International journal of intelligent computing and cybernetics* **7**(1), 24–37. (2014)
13. Emary, E., Zawbaa, H.M., Hassanien, A.E.: Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **172**, 371–381. (2016)
14. Faramarzi, A., Heidarinejad, M., Mirjalili, S., Gandomi, A.H.: Marine predators algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications* **152**, 113377. (2020)
15. Guo, B., Zhuang, Z., Pan, J.S., Chu, S.C.: Optimal design and simulation for pid controller using fractional-order fish migration optimization algorithm. *IEEE Access* **9**, 8808–8819. (2021)
16. Hu, Y., Wang, J., Liang, J., Yu, K., Song, H., Guo, Q., Yue, C., Wang, Y.: A self-organizing multimodal multi-objective pigeon-inspired optimization algorithm. *Science China Information Sciences* **62**(7), 1–17. (2019)
17. Huang, C.F., Tseng, Y.C.: The coverage problem in a wireless sensor network. *Mobile networks and Applications* **10**(4), 519–528. (2005)
18. Kan, T.W., Teng, C.H., Chou, W.S.: Applying qr code in augmented reality applications. In: *Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry*. 253–257. (2009)
19. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of ICNN'95-international conference on neural networks*. **4**, 1942–1948. IEEE (1995)
20. Khanduja, N., Bhushan, B.: Chaotic state of matter search with elite opposition based learning: A new hybrid metaheuristic algorithm. *Optimal Control Applications and Methods* **2021**, 1–16. (2021) <https://doi.org/10.1002/oca.2810>
21. Li, Z., Lei, L.: Sensor node deployment in wireless sensor networks based on improved particle swarm optimization. In: *2009 International Conference on Applied Superconductivity and Electromagnetic Devices*. 215–217. (2009)
22. Mann, G.K., Hu, B.G., Gosine, R.G.: Analysis of direct action fuzzy pid controller structures. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **29**(3), 371–388. (1999)
23. Masalha, F., Hirzallah, N., et al.: A students attendance system using qr code. *International Journal of Advanced Computer Science and Applications* **5**(3), 75–79. (2014)

24. Meng, Z., Chen, Y., Li, X., Yang, C., Zhong, Y.: Enhancing quasi-affine transformation evolution (quatre) with adaptation scheme on numerical optimization. *Knowledge-Based Systems* **197**, 105908. (2020)
25. Meng, Z., Pan, J.S., Xu, H.: Quasi-affine transformation evolutionary (quatre) algorithm: A cooperative swarm based algorithm for global optimization. *Knowledge-Based Systems* **109**, 104–121. (2016)
26. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. *Advances in engineering software* **69**, 46–61. (2014)
27. Mohamed, A.A.A., El-Gaafary, A.A., Mohamed, Y.S., Hemeida, A.M.: Multi-objective states of matter search algorithm for tcsc-based smart controller design. *Electric Power Systems Research* **140**, 874–885. (2016)
28. Niu, P., Niu, S., Chang, L., et al.: The defect of the grey wolf optimization algorithm and its verification method. *Knowledge-Based Systems* **171**, 37–43. (2019)
29. Pan, J.S., Sun, X.X., Chu, S.C., Abraham, A., Yan, B.: Digital watermarking with improved sms applied for qr code. *Engineering Applications of Artificial Intelligence* **97**, 104049. (2021)
30. Pan, J.S., Tian, A.Q., Chu, S.C., Li, J.B.: Improved binary pigeon-inspired optimization and its application for feature selection. *Applied Intelligence* **51**, 8661–8679. (2021)
31. Pan, J.S., Tsai, P.W., Liao, Y.B.: Fish migration optimization based on the fishy biology. In: 2010 Fourth International Conference on Genetic and Evolutionary Computing. 783–786. (2010)
32. Pradhan, M., Roy, P.K., Pal, T.: Grey wolf optimization applied to economic load dispatch problems. *International Journal of Electrical Power & Energy Systems* **83**, 325–334. (2016)
33. Rivera, D.E., Morari, M., Skogestad, S.: Internal model control: Pid controller design. *Industrial & engineering chemistry process design and development* **25**(1), 252–265. (1986)
34. Shi, Y., et al.: Particle swarm optimization: developments, applications and resources. In: *Proceedings of the 2001 congress on evolutionary computation* **1**, 81–86. IEEE (2001)
35. Sung, T.W., Zhao, B., Zhang, X.: Quasi-affine transformation evolutionary with double excellent guidance. *Wireless Communications and Mobile Computing* **2021**, 5591543. (2021)
36. Tang, K.S., Man, K.F., Chen, G., Kwong, S.: An optimal fuzzy pid controller. *IEEE transactions on industrial electronics* **48**(4), 757–765. (2001)
37. Tiwari, S.: An introduction to qr code technology. In: 2016 international conference on information technology (ICIT). 39–44. (2016)
38. Wang, B., Lim, H.B., Ma, D.: A survey of movement strategies for improving network coverage in wireless sensor networks. *Computer Communications* **32**(13–14), 1427–1436. (2009)
39. Yıldız, B.S., Yıldız, A.R., Albak, E.İ., Abderazek, H., Sait, S.M., Bureerat, S.: Butterfly optimization algorithm for optimum shape design of automobile suspension components. *Materials Testing* **62**(4), 365–370. (2020)
40. Zhang, M., Long, D., Qin, T., Yang, J.: A chaotic hybrid butterfly optimization algorithm with particle swarm optimization for high-dimensional optimization problems. *Symmetry* **12**(11), 1800 (2020)