Article

Collaborative Target Search Algorithm for UAV Based on Chaotic Disturbance Pigeon-Inspired Optimization

Linlin Li ^{1,2} , Shufang Xu ^{1,2,*}, Hua Nie ³, Yingchi Mao ^{1,2} and Shun Yu ^{1,2}

- ¹ School of Computer and Information, Hohai University, Nanjing 211100, China; 201307040034@hhu.edu.cn (L.L.); yingchimao@hhu.edu.cn (Y.M.); yushun15295790708@163.com (S.Y.)
² Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University, Nanjing 211100, China
³ Suma Technology Co., Ltd., Kunshan 215300, China; nh@cancon.com.cn
* Correspondence: xushufang@hhu.edu.cn

Abstract: Unmanned aerial vehicles (UAVs) have shown their superiority in military and civilian missions. In the face of complex tasks, many UAVs are usually needed to cooperate with each other. Therefore, multi-UAV cooperative target search has attracted more and more scholars' attention. At present, there are many bionic algorithms for solving the cooperative search problem of multi-UAVs, including particle swarm optimization algorithm (PSO) and differential evolution (DE). Pigeon-inspired optimization (PIO) is a new swarm intelligence optimization algorithm proposed in recent years. It has great advantages over other algorithms in convergence, robustness, and accuracy, and has few parameters to be adjusted. Aiming at the shortcomings of the standard pigeon colony algorithm, such as poor population diversity, slow convergence speed, and the ease of falling into local optimum, we have proposed chaotic disturbance pigeon-inspired optimization (CDPIO) algorithm. The improved tent chaotic map was used to initialize the population and increase the diversity of the population. The disturbance factor is introduced in the iterative update stage of the algorithm to generate new individuals, replace the individuals with poor performance, and carry out disturbance to increase the optimization accuracy. Benchmark functions and UAV target search model were used to test the algorithm performance. The results show that the CDPIO had faster convergence speed, better optimization precision, better robustness, and better performance than PIO.

Keywords: pigeon-inspired optimization; tent map; disturbance mechanism; cooperative target search

Citation: Li, L.; Xu, S.; Nie, H.; Mao, Y.; Yu, S. Collaborative Target Search Algorithm for UAV Based on Chaotic Disturbance Pigeon-Inspired Optimization. *Appl. Sci.* **2021**, *11*, 7358. <https://doi.org/10.3390/app11167358>

Academic Editor: David Megías

Received: 12 July 2021

Accepted: 4 August 2021

Published: 10 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, with the development of UAV-related technology, the application of UAVs has penetrated into all aspects of society. It is an indispensable tool for search and rescue in critical and time-sensitive missions due to its strong adaptability, low risk, low cost, and no casualties to the physical environment. However, due to the increasingly complex mission environment, it is more and more difficult for a single UAV to search the target area, and it is often unable to complete all the aerial search tasks in a short time. Therefore, the study of cooperation mechanism among multiple UAVs is helpful to solve more complex and valuable application problems. Multi-UAVs cooperation can carry out multiple tasks in military and civil fields, including regional reconnaissance, search and rescue, terrain survey, etc., which has higher task execution efficiency and fault tolerance compared with single UAV. The key problem of multi-UAVs cooperative search is coverage and task assignment. One or a set of ordered tasks are assigned to each UAV to maximize the overall operational efficiency of the UAV. Obviously, multi-UAVs' cooperative target search can be turned into a multi-objective optimization problem (MOP).

Recently, increasingly more scholars are using intelligent algorithms to solve MOPs [1–7]. Intelligent algorithms include particle swarm algorithm [2], genetic algorithm [3], and sim-

ulated annealing algorithm [4]. Reference [5] improves the updating strategy of particles and enhances the robustness of multi-objective particle swarm optimization algorithm in solving permanent magnet optimization problems of electromagnetic devices. In Reference [6], a hybrid multi-objective optimization algorithm combining differential evolution and particle swarm optimization (PSO) is proposed for the contactor (electromagnetic device) optimization problem to accelerate the convergence rate and obtain the limit optimal solution. In Reference [7], a multi-objective optimization method with adaptive weight determination scheme and tunnel method is proposed. By using the concept of hyperplane, the weights of different objective functions are determined adaptively, so that new solutions can be found gradually in the target space. The pigeon-inspired optimization (PIO) is a new swarm intelligence optimization algorithm proposed in recent years. It is widely used in neural network, path planning, and other fields because of its fast convergence speed and easy implementation. PIO algorithm can be divided into map and compass operator and landmark operator. In the map and compass section, the pigeon updates its position and speed according to the global optimal position during each iteration. In the landmark section, the pigeons update their positions with the best position for each iteration. Two operators are employed to guide the pigeons to find the destination. Although PIO algorithm is superior to other intelligent optimization algorithms, such as PSO and DE, it still has the common precocious convergence problem.

In the present study, PIO algorithm was optimized. Optimization is mainly divided into two parts: initialization optimization and optimization in the iteration update phase. In the initial part, there are the following studies: Duan et al. added the orthogonal design strategy in the initial stage to improve the diversity of individuals [8]. In Reference [9], Pareto sort is added into the initialization stage to apply to the multi-objective optimization algorithm to improve the efficiency of the algorithm. The optimization of iterative update mainly has the following three optimization objectives: (1) In order to increase the population diversity, the predator strategy was introduced in the studies of [10–12] to increase the diversity of pigeons and overcome the local optimal solution; Deng introduced the concept of dynamics to improve population diversity [13]. (2) In order to solve the problem of the pigeons being easily trapped in the local optimal solution, Duan added the global update operator at the beginning of the iteration, and on this basis, added the population mutation operator further, which can expand the search scope and make the population out of the local optimal solution [14]. Reference [15] was able to effectively reduce the probability of optimization results falling into local optimum by implementing compass and landmark operators with Cauchy variation. (3) To solve the convergence speed, efficiency, and robustness of the algorithm, References [16,17] introduced the parallel competition and hierarchical search strategy and the pigeon-group hierarchy network mechanism, respectively, to improve the global search ability and local convergence ability of the algorithm. Xu et al. added interference factors in the map and compass stages, which could improve efficiency, and selected classification learning strategy in the landmark stage [18]. Hai et al. proposed an optimization algorithm based on evolutionary game theory to improve the adaptability of the original PIO algorithm. Meanwhile, the robustness and efficiency of the algorithm are enhanced [19]. In Reference [20], mutation mechanism was added in the iterative update stage to enhance the exploration ability of the pigeons and improve the efficiency of the algorithm.

It can be seen from the above that the optimization of PIO algorithm is mainly focused on the iterative update stage only, while there are not many studies on the simultaneous optimization of initialization and iterative update stage. Therefore, a target search algorithm based on chaotic disturbance pigeon-inspired optimization is proposed in this paper. The chaos mapping is used to initialize the population, which can enrich the diversity of the population and enhance the global search ability. In addition, this paper also adds disturbance factors in the traditional iteration update stage to avoid the algorithm easily falling into local optimum and premature convergence, so that the algorithm has a better global search ability.

The remainder of this paper is organized as follows. The second section describes the basic idea and process of PIO algorithm and points out the problems to be optimized at each stage. The third section illustrates CDPIO algorithm. Simulation validation, together with comparison with the traditional approach, is presented in Section 4. In the fifth section, the performance of CDPIO algorithm will be verified by combining the practical problem of UAV cooperative target search. Section 6 provides conclusions and some possible paths for future work.

2. Related Work

2.1. PIO Algorithm

Pigeon-inspired optimization (PIO) is a new swarm intelligence algorithm based on pigeon behavior recently proposed by Professor Duan [21]. By sensing the earth's magnetic field and following their familiar landmarks, pigeons can always find their way to their destinations as quickly as possible.

The process of PIO algorithm can be divided into three main parts:

(1) The population initialization

Before algorithm execution, parameters need to be initialized. Initialization parameters include the population size N_p , solution space dimension D , map and compass factor R , the total number of iterations N_c , the number of iterations $N_{C1 \text{ max}}$ and $N_{C2 \text{ max}}$ for two operators, and $N_{C1 \text{ max}} < N_{C2 \text{ max}}$. Although PIO algorithm has a fast convergence speed, it also has a fatal shortcoming: it is easy to fall into local optimum. As you can imagine, the whole searching process is towards the optimal or the best individual; therefore, when the solution of initialization is dense, or when the optimal initial position is not in the vicinity of a truly global optimal, the error of orientation will make the space of whole other individual search mistakes along the way, and the optimal solution cannot be found.

Aiming at this defect, we optimized PIO algorithm in the population initialization stage. In the process of initialization, adding chaotic mapping makes the initial sequence generated more random, and the solution space covers a wider area, which can obtain a better distribution of pigeons so that the population has diversity and convergence.

(2) Map and compass section

The main principle that pigeons use to find their destination is to sense the earth's magnetic field to create a map in their minds. At the same time, they can adjust their orientation according to the height of the sun. In the d dimensional search space, each pigeon P_i has two attributes: position X_i and speed V_i . Pigeons use maps and compasses to update their position and speed. The updating equation of the n -th iteration is as follows:

$$V_i(n) = V_i(n-1)e^{-Rn} + \text{rand}(X_g(n) - X_i(n-1)) \quad (1)$$

$$X_i(n) = V_i(n) + X_i(n-1) \quad (2)$$

where X_g is the global optimal position, which can obtain the best position in the whole search space, and R is the map and compass operator determined by the actual situation and experience, and $\text{rand} \in [0, 1]$.

In the late iteration of the map and compass part of PIO algorithm, all the pigeons gradually gather together, and the optimal position of the pigeon colony only changes very little over a long period of time. In this case, the position update of pigeons is very slow, and each pigeon will slowly stop moving until it finally converges to a certain position in the search space. In this case, it will fall into local optimum. To solve the above problems, this paper proposes an improved PIO algorithm—CDPIO. Tent chaotic disturbance is introduced into iterative updating to increase the global search ability of the algorithm. Firstly, the chaotic variable Z_d carrier generated by the population initialization is put into the solution space to be solved, and then the chaotic disturbance of the individual is carried out according to certain rules. Finally, enhanced individual performance and better searching effect can be achieved by the simulations in Sections 4 and 5.

(3) Landmark section

Pigeons use landmarks as a tool to optimize their position as they approach their destination. Pigeons familiar with the landmark flew directly to their destination, while those unfamiliar followed them. Assuming that every pigeon could fly directly to its destination, half of the pigeons familiar with the landmark could find a nearby center that would guide them to their destination as quickly as possible. In each iteration, the number of pigeons N_p is reduced by half. The updated equation is as follows.

$$N_p(n) = \frac{N_p(n-1)}{2} \quad (3)$$

$$X_c(n) = \frac{\sum X_i(n) \text{fitness}(X_i(n))}{N_p \sum \text{fitness}(X_i(n))} \quad (4)$$

$$X_i(n) = X_i(n-1) + \text{rand}(X_c(n) - X_i(n-1)) \quad (5)$$

where X_c is the average center of the pigeons in each n -th iteration, and $\text{fitness}()$ is the quality of the pigeon individual. For the maximizing optimization problem, we have the expression of $\text{fitness}(X_i(n))$.

$$\text{fitness}(X_i(n)) = f_{\max}(X_i(n)) \quad (6)$$

For the minimization optimization problem,

$$\text{fitness}(X_i(n)) = \frac{1}{f_{\min}(X_i(n)) + \varepsilon} \quad (7)$$

where ε is a positive value close to 0.

Finally, we can obtain the global optimal position X_p after N_c iterations.

$$X_p = \max(X_g(1), X_g(2), \dots, X_g(n)) \quad (8)$$

where X_p represents the global best location.

2.2. Analysis of Traditional Mapping Approaches

Chaos is a seemingly random motion that determines the dynamic system, and it is ubiquitous in nonlinear systems. Chaotic motion has three remarkable characteristics: (1) it is highly sensitive to initial values; (2) ergodicity of motion trajectory; (3) randomness. It can traverse every orbit according to its own rules within a certain range, not repeating itself nor crossing itself. Because of the randomness as well as the ergodicity and regularity of chaotic variables, they have been applied to the optimization search problem by many scholars, which can not only effectively maintain the diversity of the population, but also help the algorithm to jump out of the local optimum and improve the global searching ability.

Logistic mapping and tent mapping are the most commonly used at present. Most studies use logistic mapping to carry out chaotic operation. Xu et al. used Logistic mapping in the initial stage to make individual search more accurate and avoid falling into local optimal. Meanwhile, chaotic processing was only performed on some currently superior individuals to speed up the convergence rate [22].

The mathematical expression of Logistic mapping is expressed as:

$$x(t+1) = \mu x(t)(1-x(t)) \quad (9)$$

where t is the iteration time steps, $x(t) \in [0, 1]$, μ is a tunable parameter to ensure that the mapped $x(t)$ is always within $[0, 1]$, $\mu \in [0, 4]$. The results show that when the value of μ meets the condition of $3.5699456 < \mu \leq 4$, especially when it is close to 4, the value generated by iteration is in a random distribution state. However, the value probability of

Logistic mapping is high at both ends of the interval, and thus the optimization speed of the algorithm will be affected by the unevenness of Logistic traversal.

Tent mapping has better traversal uniformity and convergence speed than logistic mapping and can be used to optimize the search ability of the algorithm. In view of the decrease of population diversity, it is easy to fall into the problem of local optimal solution. However, most of the current literature studies are focused on PSO algorithm. There are few studies on PIO algorithm. In the existing literature, most of the studies use tent mapping to initialize the population, which effectively improves the diversity of the population. By chaotic search, the adaptive tent mapping optimization is used to help our algorithm to jump out of the local optimum and enhance the global search ability. Among them, the advantage of tent mapping can make up for the disadvantage of PIO algorithm, and therefore chaotic mapping can be considered to apply to the optimization of PIO algorithm.

3. Proposed Algorithm

3.1. Population Initialization Based on Tent Mapping

Before the execution of PIO algorithm, the parameters and pigeon cluster sequence need to be initialized. The parameters include: $N_p, N_{C1}, N_{C2}, X_i(1), V_i(1)$, where N_{C1} and N_{C2} represent the maximum iteration times of the map and compass part and the landmark part, respectively, and $N_{C1} < N_{C2}$.

The pigeon cluster initialization sequence is generated using tent mapping.

There are small periods and unstable periodic points in traditional tent chaotic sequences. According to the ideas in [23], the traditional tent mapping is improved in this paper. In order to avoid it falling into small periodic points or unstable periodic points and at the same time not destroying the three characteristics of chaotic variables, we optimized the original tent mapping expression as follows.

Random variable $rand(0, 1) \times \frac{1}{NT}$ is introduced into the original tent mapping expression, and the improved expression is shown as follows.

$$z_{i+1} = \begin{cases} 2z_i + rand(0, 1) \times \frac{1}{NT}, & 0 \leq z \leq \frac{1}{2} \\ 2(1 - z_i) + rand(0, 1) \times \frac{1}{NT}, & \frac{1}{2} \leq z \leq 1 \end{cases} \quad (10)$$

The expression after the Bernoulli transformation [24] can show that

$$z_{i+1} = (2z_i) \bmod 1 + rand(0, 1) \times \frac{1}{NT} \quad (11)$$

where, z_{i+1} is the generated chaotic sequence, NT is the number of particles in the chaotic sequence, and $rand(0, 1)$ is the random number between $[0, 1]$.

First, generate the initial value z_0 in the interval $[0, 1]$, denoted by $i = 0$; then, according to the above Equation (11), loop iteration is carried out to generate Z sequence, with i increasing by 1. When the number of iterations reaches the maximum, the program is over, and the generated Z sequence will be saved as the initialization pigeon cluster sequence of this algorithm.

In the process of initialization, adding chaotic mapping makes the initial sequence generated more random and covers a wider area, which can get a better distribution of pigeons.

Implementation steps of population initialization are shown in Algorithm 1.

Algorithm 1 Population Initialization

Step1: Initialize the population to generate the initial value z_0 in the interval $[0, 1]$, denoted by $i = 0$.

Step2: Equation (13) is used for iteration to generate Z sequence, with i increasing by 1.

Step3: If the maximum number of iterations is reached, the program will stop and the resulting Z sequence will be saved.

3.2. Iterative Updating Based on Disturbance Mechanism

In the late iteration of the map and compass part of PIO algorithm, all the pigeons gradually gather together, and the optimal position of the pigeon colony only changes very little over a long period of time. In this case, the position updating of pigeons is very slow, and each pigeon will slowly move until it finally converges to a certain position in the search space. In this case, it will fall into local optimum. To solve the above problem, this paper proposes an optimization algorithm (CDPIO), which introduces tent chaotic disturbance in the iterative update phase to increase the global search ability of the algorithm.

In the iterative update stage, the pigeon mainly adjusts its direction through a compass to find its destination. In the d dimensional search space, each pigeon P_i has two attributes: position X_i and speed V_i . Pigeons will use a map and compass and update their position and speed according to the appropriate strategy. Pigeons familiar with the landmark flew directly to their destination, while those unfamiliar followed them. Each pigeon will reach its destination as fast as possible through the center pigeon. After the completion of one iteration of the algorithm, the fitness f_i of each pigeon and the average fitness f_{avg} of the flock were calculated. When $f_i \geq f_{avg}$, it indicates that the pigeons are in the “divergence” trend, and tent chaotic disturbance should be added to optimize the pigeons.

Firstly, the chaotic variable Z_d carrier generated in the initial stage of the population in Section 3.3.2 is added to the solution space to be solved, which is defined as follows.

$$newX_d = min_d + (max_d - min_d) \cdot Z_d \quad (12)$$

where max_d and min_d are the maximum and minimum values of the d -dimensional variable $newX_d$, respectively.

If it is necessary to disturb the current sequence of pigeons, the individuals will be disturbed according to the following disturbance rules.

$$newX' = (X' + newX) / 2 \quad (13)$$

where X' is the individual requiring chaotic disturbance, $newX$ is the amount of chaotic disturbance generated, and $newX'$ is the new individual after chaotic disturbance. If the performance of the disturbed individual is better, the new individual is used to replace the original individual; otherwise, the original individual remains unchanged.

By introducing chaotic disturbance, the algorithm can avoid falling into local optimum and improve the global searching ability and searching precision. Implementation steps of Tent chaotic disturbance mechanism are shown in Algorithm 2.

Algorithm 2 Tent Chaotic Disturbance

- Step1: The chaos variable Z_d is generated by Equation (11).
 - Step2: Carrier the chaotic variable into the solution space of the problem to be solved.
 - Step3: According to Equation (12), the chaotic disturbance is performed on the individual, and the replacement and update operation is decided according to the performance comparison between the new and the old individual.
-

3.3. CDPIO

The PIO algorithm is well known for its fast convergence speed, and the optimal solution is stable and satisfactory in the search process. However, too high search speed will reduce the traversal ability of PIO algorithm in the whole search space, leading to premature convergence of the algorithm. Therefore, on the basis of the above two aspects of chaotic mapping and chaotic disturbance mechanism, we supplemented and optimized the PIO algorithm.

3.3.1. Population Initialization

In the initialization stage, initial values are set by the initialization mechanism in Section 3.1 to set the parameters required by the algorithm, and the improved tent chaotic mapping expression is used to generate tent chaotic sequence in the feasible domain; then, the fitness f_i of each pigeon is calculated, and the current optimal fitness f_g and its corresponding position x_b are selected, as well as the current worst fitness f_w and its corresponding position. The population initialization is thus completed.

3.3.2. Iteration Updating Stage

After the initial stage, the algorithm will enter the iterative update stage. In this section, each pigeon will be updated according to the existing speed and position update strategy. When the algorithm completes an iteration, it will judge whether the pigeons are in a “divergent” trend. If so, the individuals will be disturbed according to the tent disturbance mechanism described in Section 3.2 of this paper to generate new individuals. After the disturbance, the optimal position x_b and its fitness f_g experienced by the whole population are updated according to the current status of the flock, as well as the current worst fitness f_w and its corresponding position. After updating, the algorithm will judge whether the number of iterations has reached the maximum number of iterations. If not, the cycle of updating iteration will continue. Otherwise, the cycle will end, and the optimal sequence of pigeons will be output as the optimal result of the algorithm.

CDPIO algorithm introduces tent chaotic mapping, which increases the population diversity, improves the performance of the algorithm, and avoids falling into local optimum. The concrete implementation steps of CDPIO are shown in Algorithm 3.

Algorithm 3 Chaotic Disturbance Pigeon-inspired Optimization Algorithm

Step 1: Initialize the parameters in CDPIO, including the number of pigeons N_p , the maximum iteration times of compass part N_{C1} , and the landmark part N_{C2} ($N_{C1} < N_{C2}$), and the state of each pigeon, the position X_i , and the velocity V_i .

Step 2: The population is initialized according to Algorithm 1.

Step 3: Calculate the fitness f_i of each pigeon, select the current best fitness f_g and its corresponding position x_b , along with the current worst fitness f_w and its corresponding position.

Step 4: Update X_i and V_i according to Equations (1) and (2).

Step 5: If $n > N_{C1}$, jump to Step 6, or $n = n + 1$, and return Step 4.

Step 6: Update N_p , X_c , X_i according to Equations (3)–(5), and update and preserve X_g .

Step 7: After the completion of one iteration, it is judged as to whether the pigeons need to be disturbed according to Algorithm 3.

Step 8: According to the current status of the pigeons, the optimal position x_b and its fitness f_g experienced by the whole population are updated, as well as the current worst fitness f_w and its corresponding position.

Step 9: If $n > N_{C2}$, jump to Step 10; or if $n = n + 1$, return to Step 6.

Step 10: According to Equation (8), X_p is obtained and retained. The loop ends and the optimization result is output as the output of the algorithm.

4. Performance Evaluation of CDPIO

4.1. Simulation Setup

This section is mainly to verify the feasibility and superiority of CDPIO algorithm. In order for the performance and characteristics of the algorithm to be fully tested, many standard functions with different characteristics are usually introduced. Here, nine complex functions [25], as shown in Table 1, are used as experimental objects. The functions selected involve multiple complex functions such as single peak, multi-peak, low dimension, and high dimension. The optimization ability of CDPIO algorithm can be fully investigated through various types of reference functions.

Table 1. Benchmark functions.

No.	Function	Scope	Best
f1	<i>Easom</i>	$[-100,100]$	−1
f2	<i>Matyas</i>	$[-10,10]$	0
f3	<i>Trid6</i>	$[-d^2, d^2]$	−50
f4	<i>Booth</i>	$[-10,10]$	0
f5	<i>Bohachevsky1</i>	$[-100,100]$	0
f6	<i>Schaffer</i>	$[-10,10]$	0
f7	<i>Six Hump Camel Back</i>	$[-3,3] \& [-2,2]$	−1.0316
f8	<i>Bohachevsky3</i>	$[-100,100]$	0
f9	<i>Rastrigin</i>	$[-5.12, 5.12]$	0

Experimental setting parameters are shown in Table 2:

Table 2. Experimental parameters.

Parameter	Value
Number of pigeons N_p	Dimension $D * 5$
Compass factor C	0.3
Parameter of position x_0	0
Parameter of size γ	1
Number of iterations of map and compass N_{C1max}	6
Number of iterations of landmark N_{C2max}	4
The total number of iterations N_c	100

4.2. Performance of CDPIO

The population size of each algorithm and the total number of iterations were the same, and each individual was evaluated for individual fitness only once during each iteration. The average value, standard deviation, and optimal value of each algorithm were used as the final evaluation index for nine different types of benchmark functions. The correlation performance of each test function was tested and compared with that of PIO algorithm. The optimization results are shown in the table below.

The best value and average value in Table 3 can be expressed as the convergence precision and optimization ability of the algorithm. For f1, f4, f5, f6, and f7 functions, the convergence accuracy was very high, while for f5, f6, and f8 functions, the average value was very close to the optimal value, and compared with PIO algorithm, the ability of CDPIO has been significantly improved.

The standard deviation in Table 3 reflects the robustness of the algorithm. Low standard deviation proves that there is little difference in the optimization data and the optimization process is more stable. For f5, f6, f8, and f9, the standard deviation is 0, which is a great improvement compared with the standard deviation of PIO algorithm, indicating that CDPIO has better robustness.

As shown in Figure 1, the four pictures (a–d) represent the low-dimensional unimodal and low-dimensional multimodal, high-dimensional unimodal, and iterative graph of high-dimensional multimodal function. The blue curve shows the test curve of the benchmark functions using the PIO algorithm, and the red curve represents the test curve of functions using CDPIO algorithm. From the four figures, it can be seen that the efficiency and optimization effect of CDPIO algorithm is obviously better than that of PIO algorithm, no matter whether it is low-dimensional single peak, low-dimensional multi-peak, high-dimensional single peak, or high-dimensional multi-peak function.

Table 3. Optimization results comparison of benchmark functions.

No.	Algorithm	Best	Mean	Std
f1	PIO	−1.0	−0.9799999901948055	0.14070528128524876
	CDPIO	−1.0	−0.9999999994216735	$5.3662409510967924 \times 10^{-9}$
f2	PIO	$1.685843817620635 \times 10^{-64}$	$4.342683102611767 \times 10^{-24}$	$4.3418063892324814 \times 10^{-23}$
	CDPIO	$1.046667523548257 \times 10^{-101}$	$2.9866863790417878 \times 10^{-95}$	$1.541578753912777 \times 10^{-94}$
f3	PIO	−49.86852199484861	−29.455532099162753	18.60338945611166
	CDPIO	−49.99888611582929	−45.89495109500953	5.080090216176373
f4	PIO	0.0	$1.3492636488024638 \times 10^{-10}$	$1.3237224957292601 \times 10^{-9}$
	CDPIO	0.0	$1.8185977389425053 \times 10^{-10}$	$1.2932245827776857 \times 10^{-9}$
f5	PIO	0.0	$3.3306690738754695 \times 10^{-18}$	$2.4724833664181837 \times 10^{-17}$
	CDPIO	0.0	0.0	0.0
f6	PIO	0.0	$3.64456242962774 \times 10^{-13}$	$2.6108790831418787 \times 10^{-12}$
	CDPIO	0.0	0.0	0.0
f7	PIO	−1.0316284528850663	−1.0316284534828164	$6.121029484034265 \times 10^{-11}$
	CDPIO	−1.0316281801364018	−1.03162845074708	$2.7334550254776607 \times 10^{-8}$
f8	PIO	0.0	$2.7755575615628915 \times 10^{-18}$	$1.6503143019113236 \times 10^{-17}$
	CDPIO	0.0	0.0	0.0
f9	PIO	−3.0053818284590452	−3.0053818284590452	$4.463264492478967 \times 10^{-17}$
	CDPIO	−3.0053818284590452	−3.0053818284590452	0.0

It can be seen that compared with PIO algorithm, the CDPIO algorithm has higher accuracy, closer to the theoretical value of the function, and better optimization accuracy, robustness, and optimization efficiency.

(a)

(b)

Figure 1. Cont.

Figure 1. Comparison of convergence curves of 2 algorithms obtained on the benchmark functions. (a) is the iterative graph of the low-dimensional unimodal function; (b) is the iterative graph of low-dimensional multimodal function; (c) is the iterative graph of the high-dimensional unimodal function; (d) is the iterative graph of high-dimensional multimodal function.

4.3. Nonlinear Correlation Performance of CDPIO

The performance of PIO algorithm for finding the optimal value of the function with x_i , x_{i-1} will gradually decline with the increase of the dimension of the search space. In order to test the nonlinear correlation of the CDPIO algorithm and the ability of jumping out of the local optimal solution, we selected three test functions. The three test functions [26] have the same characteristics and all have the form of x_i , x_{i-1} . The two algorithms mentioned above are compared and analyzed on the basis of the three test functions. It is concluded that the CDPIO algorithm proposed in this paper has better performance than PIO algorithm. Figure 2 shows the iterative convergence curve of the algorithm. Figure 2a–d respectively represent the convergence curves of 2 algorithms obtained on TRID, DIXON-PRICE and ROSENBROCK function. The abscissa represents the number of iterations, and the ordinate represents the optimal adaptive value.

The formula of the three test functions is as follows.

(1) TRID:

$$f(x) = \sum_{i=1}^d (x_i - 1)^2 - \sum_{i=1}^d x_i x_{i-1} \quad (14)$$

(2) DIXON-PRICE:

$$f(x) = (x_1 - 1)^2 + \sum_{i=2}^d i (2x_i^2 - x_{i-1})^2 \quad (15)$$

(3) ROSENBROCK:

$$f(x) = \sum_{i=1}^{d-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right] x_i \in [-5, 10] \quad (16)$$

The CDPIO algorithm and PIO algorithm proposed are used for testing on the basis of the three functions, and the result analysis diagram is shown below.

Figure 2. Comparison of convergence curves of 2 algorithms obtained on the three test functions. (a) is the convergence curves of 2 algorithms obtained on TRID function; (b) is the convergence curves of 2 algorithms obtained on DIXON-PRICE function; (c) is the convergence curves of 2 algorithms obtained on ROSENBROCK function.

It can be concluded from the results of the above three simulation experiments that the convergence accuracy of CDPIO algorithm is significantly improved compared with the PIO algorithm, which is closer to the optimal value and the curve is smoothed. Among them, the closer the optimal fitness value is to 1.0, the higher the optimization precision; the inflection point of the curve appears first, and the convergence speed becomes faster. The smoother the curve is, the stronger the ability of the algorithm to escape the local optimal solution.

To sum up, CDPIO algorithm has good ability to resist local extreme values for most functions, especially for simple low-dimensional functions, wherein it has good robustness, efficiency, and global search ability.

5. Application in Coordinated Target Search of UAVs

UAVs play a key role in search missions, as they can find targets in complex environments. The searching problem of UAVs is a relatively complex multi-objective optimization problem under multiple constraints. Most search algorithms cannot meet the requirements of high efficiency and low power consumption in the search process. In order to further verify the superiority of CDPIO algorithm, not only in reference function optimization but also in the process of solving practical problems, this paper used the CDPIO algorithm to solve the cooperative search problem of UAVs.

5.1. Environment Initialization

The search area was divided into discrete areas of $D_x \times D_y$ to construct a relatively real environment map. The grid (a, b) is numbered as $v = a + (b - 1) \times D_y$, $v \in \{1, 2, 3, \dots, D_x \times D_y\}$; D_x and D_y represent the number of grids in two dimensions of length and width. For each network $v \in \{1, 2, 3, \dots, D_x \times D_y\}$, a certain value $Z_v(l)$ is given, where $Z_v(l)$ is the information structure, which comprehensively describes the environment and target information in the current grid, as shown below: $Z_v(l) = (z_v(l), x_v(l), \omega_v(l), v \in \{1, 2, 3, \dots, D_x \times D_y\})$, including $z_v(l) \in [0, 1]$, as the target probability exists, and describing the moment l the v th grid has the possibility of searching for the target. The value of $x_v(l)$ is 0 or 1, indicating whether the UAV fully understands the environmental information at the place; $\omega_v(l)$ is the function of UAV position and time. When the UAV moves to the current grid, the signal of pheromone information will be generated at the corresponding position, and the information will be transmitted to the nearby UAVs through network communication.

5.2. Cost Function

The cooperative target search problem of UAV is a very complex multi-objective optimization problem. It is very important to select the relevant multi-objective cost function. The cost function set is introduced as follows:

- F_1 : describes the probability of finding a target;
- F_2 : describes the environmental certainty of UAV;
- F_3 : describes the time cost and fuel cost between two consecutive way points;
- F_4 : used to estimate whether the UAV's trajectory is within the search area.

A better search task requires a larger cost function: F_1 , F_2 , F_3 and a lower cost function F_4 . Therefore, the inverse values of the cost functions F_1 , F_2 , and F_3 are used for unified optimization.

5.3. Experimental Result

In this paper, CDPIO and PIO algorithms were used to solve the problem. It was stipulated that 10 UAVs should be used to search for targets, among which 10 UAVs should successfully avoid two obstacle threat areas and successfully search for three targets. The experimental results are shown in Figure 3.

As shown in Figure 3a, CDPIO was used for the simulation experiment. Figure 3b shows an experimental study using PIO. In the figure above, no matter whether PIO or improved CDPIO was used, 10 UAVs can successfully avoid the threat areas and find the target together. However, when CDPIO algorithm was used for target search, the UAV was more sensitive to the threat area and its average distance from the threat area was larger. In essence, CDPIO was shown to have a better effect on solving practical UAV cooperative target search.

Figure 3. Experimental results of multi-unmanned aerial vehicle cooperative target search. (a) is the result of UAV target search using CDPIO algorithm; (b) is the result of UAV target search using PIO algorithm.

6. Conclusions

Aiming at the defects of the standard PIO algorithm, we proposed an improved Pigeon-inspired algorithm (CDPIO) in this paper. The improved tent chaotic map was used to initialize the population, and the tent disturbance strategy was introduced in the iteration update stage to improve the performance of the algorithm. The main conclusions of this paper are as follows:

(1) The initial location of the pigeons is very important for the convergence rate and solution precision of the PIO algorithm. The improved tent chaotic mapping has high ergodic, random, uniform, and initial value sensitivity. Using improved tent mapping to initialize the population, the population can be distributed evenly in the whole space effectively. The performance of CDPIO algorithm was greatly improved compared with that of the PIO algorithm by introducing tent disturbance in the iteration update stage to avoid falling into the local optimal solution.

(2) The optimization test results of nine benchmark functions and three special test functions showed that CDPIO has a strong performance advantage over the PIO algorithm in terms of convergence speed, optimization precision, and robustness, which verifies the effectiveness of the improved algorithm.

(3) In the next step, the optimization mechanism and algorithm structure of CDPIO can be further improved and optimized in the optimization time. The intelligent algorithm with better performance was proposed, and the CDPIO algorithm was applied to more complex optimization problems, so as to expand the application field of the algorithm.

Author Contributions: Conceptualization, S.X., L.L., H.N. and Y.M.; methodology, S.X., L.L. and S.Y.; validation, L.L. and S.Y.; formal analysis, L.L. and S.X.; investigation, S.X. and L.L.; writing—original draft preparation, L.L. and S.X.; writing—review and editing, L.L.; supervision, S.X.; project administration, S.X. and Y.M.; funding acquisition, Y.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Gusu Innovation Leading Talents Special Project, grant number ZX12020210, and the 2020 Kunshan Zu Chong Key Tack Project.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dobes, J.; Michal, J.; Biolkova, V. Multiobjective optimization for electronic circuit design in time and frequency domains. *Radioengineering* **2013**, *22*, 136–152.
2. Hirano, H.; Yoshikawa, T. A study on two-step search using global-best in PSO for multi-objective optimization problems. In Proceedings of the 6th International Conference on Soft Computing and Intelligent Systems/13th International Symposium on Advanced Intelligent Systems (SCIS/ISIS), Kobe, Japan, 20–24 November 2012.
3. Shorbagy, M.; Mousa, A.A.A.; Fathi, W. *Hybrid Particle Swarm Algorithm for Multiobjective Optimization: Integrating Particle Swarm Optimization with Genetic Algorithms for Multiobjective Optimization*; LAP (Lambert Academic Publishing): Saarbrücken, Germany, 2011.
4. Fakhfakh, M.; Sallem, A.; Boughariou, M.; Bennour, S.; Bradai, E.; Gaddour, E.; Loulo, U.M. Analogue Circuit Optimization through a Hybrid Approach. In *Intelligent Computational Optimization in Engineering*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 366, pp. 297–327.
5. Ye, X.; Chen, H.; Liang, H.; Chen, X.; You, J. Multi-Objective Optimization Design for Electromagnetic Devices with Permanent Magnet Based on Approximation Model and Distributed Cooperative Particle Swarm Optimization Algorithm. *IEEE Trans. Magn.* **2018**, *54*, 1–5.
6. Yang, W.; Guo, J.; Liu, Y.; Zhai, G. Multi-Objective Optimization of Contactor's Characteristics Based on RBF Neural Networks and Hybrid Method. *IEEE Trans. Magn.* **2019**, *55*, 1–4. [CrossRef]
7. Ryu, N.; Song, W.S.; Jung, Y.; Min, S. Multi-Objective Topology Optimization of a Magnetic Actuator Using an Adaptive Weight and Tunneling Method. *IEEE Trans. Magn.* **2019**, *55*, 1–4. [CrossRef]
8. Duan, H.; Wang, X. Echo State Networks With Orthogonal Pigeon-Inspired Optimization for Image Restoration. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 2413–2425. [CrossRef] [PubMed]
9. Ruan, W.; Duan, H. Obstacle avoidance control for multi-UAV based on multi-objective social learning pigeons swarm optimization. *Front. Inf. Technol. Electron. Eng.* **2020**, *21*, 740–749. [CrossRef]
10. Duan, H.; Yang, Y.S. Predator-Prey Pigeon-Inspired Optimization for UAV ALS Longitudinal Parameters Tuning. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 2347–2358. [CrossRef]
11. Zhang, B.; Duan, H. Three-Dimensional Path Planning for Uninhabited Combat Aerial Vehicle Based on Predator-Prey Pigeon-Inspired Optimization in Dynamic Environment. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2017**, *14*, 97–107. [CrossRef]
12. Duan, H.; Qiu, H.X. Tight formation cooperative control of unmanned aerial vehicle based on predation and escape pigeons swarm optimization. *Sci. China Technol. Sci.* **2015**, *45*, 559–572.
13. Duan, H.; Zhao, J. Dynamic Discrete Pigeon-Inspired Optimization for Multi-UAV Cooperative Search-Attack Mission Planning. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 706–720. [CrossRef]
14. Hua, B.; Liu, R.N. A spacecraft cluster orbit planning method based on adaptive population mutation pigeons swarm optimization. *Sci. China Technol. Sci.* **2020**, *50*, 453–460.
15. Duan, H.; Yang, Y. Rolling time domain control of large civil aircraft based on Cauchy mutation flock optimization. *Sci. China Technol. Sci.* **2018**, *48*, 277–288.
16. Wang, J.; Song, X. Design of Self-Shielded Uniform Magnetic Field Coil via Modified Pigeon-Inspired Optimization in Miniature Atomic Sensors. *IEEE Sens. J.* **2021**, *21*, 475–482.
17. Qu, M.; Wei, C. Clustering optimization algorithm for large scale UAV based on intelligent behavior of pigeons. *Sci. China Technol. Sci.* **2020**, *50*, 475–482.
18. Xu, X.; Deng, Y. UAV Power Component—DC Brushless Motor Design with Merging Adjacent-Disturbances and Integrated-Dispatching Pigeon-Inspired Optimization. *IEEE Trans. Magn.* **2018**, *54*, 1–7. [CrossRef]
19. Hai, X.; Wang, Z.; Feng, Q.; Ren, Y.; Xu, B.; Cui, J.; Duan, H. Mobile Robot ADRC With an Automatic Parameter Tuning Mechanism via Modified Pigeon-Inspired Optimization. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 2616–2626. [CrossRef]
20. Duan, H.; Shi, Y. Limit-Cycle-Based Mutant Multi-objective Pigeon-Inspired Optimization. *IEEE Trans. Evol. Comput.* **2020**, *24*, 948–959. [CrossRef]
21. Duan, H.; Qiao, P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *Int. J. Intell. Comput. Cybern.* **2014**, *7*, 24–37. [CrossRef]
22. Li, P.; Xu, Z. Stochastic Optimal Operation of Microgrid Based on Chaotic Binary Particle Swarm Optimization. *IEEE Trans. Smart Grid* **2016**, *7*, 66–73. [CrossRef]
23. Lv, X.; Mu, X.D. Chaotic Sparrow Search Optimization Algorithm. *J. Beijing Univ. Aeronaut. Astronaut.* **2020**, 1–10. [CrossRef]
24. Solving Bernoulli Equation Transformation. Available online: <https://math.stackexchange.com/questions/1252916/solving-bernoulli-equation-transformation> (accessed on 3 August 2021).

-
25. Wu, H.; Zhang, F.; Wu, L. A new swarm intelligence algorithm—Wolf pack algorithm. *Syst. Eng. Electron.* **2013**, *35*, 2430–2438.
 26. Virtual Library of Simulation Experiments: Test Functions and Datasets. Available online: <https://www.sfu.ca/~jssurjano/index.html> (accessed on 3 August 2021).