

# 融合 Tent 映射和模拟退火的改进鸽群优化算法

张安玲

(长治学院 数学系, 山西 长治 046011)

**摘要:** 针对鸽群优化算法容易陷入局部最优、求解精度低和局部搜索能力差的问题, 提出了一种融合 Tent 映射和模拟退火的改进鸽群优化算法。首先, 采用 Tent 映射初始化种群, 使初始种群分布更为均匀。然后, 在鸽群优化算法的地标算子运行后再加入模拟退火算法, 利用模拟退火算法以一定的概率跳出局部最优解以及具有的渐进收敛性, 提高了全局优化的能力。基于 16 个基准测试函数对算法性能进行了测试, 实验结果表明, Tent-PIO-SA 算法相比 PIO、Tent-PIO 算法, 在收敛精度上平均提高了 10 个数量级, 特别对于极难优化的 Rosenbrock 函数, Tent-PIO-SA 算法相比最近的经典算法 LECUSSA、SCASL、CML-WOA、APN-WOA 在收敛精度上平均高出了 6 个数量级, 比 TLPSO、SCA-DE 算法高出了 7 个数量级, 证明了所提出的 Tent-PIO-SA 算法具有较强的寻优能力。

**关键词:** 鸽群优化算法; 模拟退火算法; Tent 映射

**中图分类号:** TP301.6 **文献标识码:** A **doi:** 10.62756/jnuc.issn.1673-3193.2023.11.0026

**引用格式:** 张安玲. 融合 Tent 映射和模拟退火的改进鸽群优化算法[J]. 中北大学学报(自然科学版), 2025, 46(1): 53-63.

ZHANG Anling. Improved pigeon swarm optimization algorithm combining Tent mapping and simulated annealing[J]. Journal of North University of China(Natural Science Edition), 2025, 46(1): 53-63.

## Improved Pigeon Swarm Optimization Algorithm Combining Tent Mapping and Simulated Annealing

ZHANG Anling

(Department of Mathematics, Changzhi University, Changzhi 046011, China)

**Abstract:** Aiming at the problems that pigeon swarm optimization algorithm is easy to fall into local optimum, low solution accuracy and the poor local searching ability, improved pigeon swarm optimization algorithm combining Tent mapping and simulated annealing was proposed. Firstly, Tent mapping was used to initialize the population to make the initial population distribution more uniform. Then, the simulated annealing algorithm was added after running the landmark operator of the pigeon swarm optimization algorithm. The simulated annealing algorithm can jump out of the local optimal solution with a certain probability and has the asymptotic convergence to improve the ability of global optimization. The experimental results on the performance of the algorithm based on 16 benchmark functions show that the convergence accuracy of Tent-PIO-SA algorithm is improved by an average of 10 orders of magnitude compared with PIO and Tent-PIO algorithms. Especially for Rosenbrock function, which is extremely difficult to optimize, the convergence accuracy of Tent-PIO-SA algorithm achieves on average 6 orders of magnitude higher than the recent classical algorithms LECUSSA, SCASL, CML-WOA and APN-WOA, and 7 orders of magnitude higher than TLPSO, SCA-DE algorithm.

**收稿日期:** 2023-11-25

**作者简介:** 张安玲(1980—), 女, 副教授, 硕士, 主要从事智能优化算法方面的研究。E-mail: zhanganling1118@163.com。

It is proved that the proposed Tent-PIO-SA algorithm has strong optimization ability.

**Key words:** pigeon swarm optimization algorithm; simulated annealing algorithm; Tent mapping

## 0 引言

鸽群优化(pigeon-inspired optimization, PIO)算法<sup>[1]</sup>是Duan等提出的一种基于鸽子归巢行为型群体智能优化算法。该算法在无人机编队、控制参数优化和图像处理等多个领域<sup>[2-6]</sup>已得到成功应用。但是,该算法存在易早熟、易陷入局部最优、后期搜索能力差等不足<sup>[7]</sup>。针对这些问题,许多学者提出了改进策略。He等<sup>[8]</sup>提出了一种多策略鸽群优化(MSPIO)算法,该算法利用非线性权重因子动态地改变了鸽子在鸽群中的位置,同时用盘旋逼近的搜索方法增强了鸽子的探索能力,并且提出了一种随机对立学习方法解决了可能处于局部最优停滞状态的问题。Sun等<sup>[9]</sup>提出了一种拟仿射变换鸽群优化(QT-PIO)算法,该算法主要将拟仿射变换演化算法中的协同进化 $M$ 和变异 $B$ 应用到PIO算法中,使个体在优化过程中以矩阵的形式更新自己的位置,这种学习策略不仅提高了PIO的多样性,而且有效避免了PIO算法中鸽子陷入局部最优解的问题。Huo等<sup>[10]</sup>提出了一种基于Agent路由和光传感器的自适应变异多目标鸽群优化(AMMOPIO)算法,该算法利用自适应飞行机制获得较好的鸽群分布,增强了种群的多样性和收敛性,并且利用变异机制简化了鸽群优化算法的模型,提高了搜索效率。胡耀龙等<sup>[11]</sup>提出了基于自适应学习策略的改进鸽群优化(ALPIO)算法,该算法引入了自学习策略,增强了种群的多样性,提高了算法全局寻优能力。

本文针对标准PIO算法容易陷入局部最优、求解精度低、局部搜索能力差的特点,引入Tent搜索机制和模拟退火算法,提出了一种改进鸽群优化(Tent-PIO-SA)算法。该算法应用Tent映射初始化种群,使初始个体尽可能分布均匀。在PIO算法的地标算子运行后融入模拟退火算法,利用模拟退火算法能够跳出局部最优和搜索精度高的优点可以弥补PIO算法的不足。

## 1 鸽群优化算法

鸽群优化算法有2种不同的算子模型:

1) 地图和指南针算子(map and compass operator)。

$D$ 维搜索空间的 $N$ 只鸽子的位置和速度分别为

$$X_i = [x_{i1}, x_{i2}, \dots, x_{iD}],$$

$$V_i = [v_{i1}, v_{i2}, \dots, v_{iD}], \quad i = 1, 2, \dots, N.$$

每只鸽子依据式(1)和式(2)更新其速度 $V_i$ 和位置 $X_i$ 。

$$V_i' = V_i^{t-1} \cdot e^{-Rt} + rand \cdot (X_{gbest} - X_i^{t-1}), \quad (1)$$

$$X_i' = X_i^{t-1} + V_i', \quad (2)$$

式中: $R$ 为地图和指南针因数,取值范围设定为 $0 \sim 1$ ;  $rand$ 为 $0 \sim 1$ 的随机数; $t$ 为当前迭代次数; $X_{gbest}$ 为 $t-1$ 次迭代循环中鸽子的全局最优位置。

2) 地标算子(landmark operator)。

当鸽子接近目标时,鸽子会依靠临近它们的地标进行导航,并根据地标算子精确寻找目标。

地标算子

$$N^t = \frac{N^{t-1}}{2}, \quad (3)$$

$$X_{center}^{t-1} = \frac{\sum_{i=1}^{N^{t-1}} X_i^{t-1} \cdot F(X_i^{t-1})}{N^{t-1} \cdot \sum_{i=1}^{N^{t-1}} F(X_i^{t-1})}, \quad (4)$$

$$X_i' = X_i^{t-1} + rand \cdot (X_{center}^{t-1} - X_i^{t-1}), \quad (5)$$

式中: $F(X_i^{t-1})$ 为第 $i$ 个个体在第 $t-1$ 次迭代后的适应度值; $X_{center}^{t-1}$ 为 $t-1$ 次迭代后剩余鸽子的中心位置,将被当作地标。

## 2 改进鸽群优化算法

### 2.1 Tent映射

鸽群优化算法通常用搜索范围内随机产生的数据作为初始种群,这将难以保证种群的多样性,在一定程度上会影响算法的寻优结果。然而,Tent映射具有随机性、均匀性和遍历性,从而使算法维持种群的多样性。因此,本文利用Tent映射初始化种群。

Tent映射的表达式为

$$x_{t+1} = \begin{cases} \frac{x_t}{u}, & 0 \leq x_t < u, \\ \frac{1-x_t}{1-u}, & u \leq x_t \leq 1. \end{cases}$$

当 $u = \frac{1}{2}$ 时,Tent映射具有最典型的形式。

本文使用的 Tent 映射为

$$x_{t+1} = \begin{cases} 2x_t, & 0 \leq x_t < \frac{1}{2}, \\ 2(1-x_t), & \frac{1}{2} \leq x_t \leq 1. \end{cases} \quad (6)$$

Tent 映射产生序列值的步骤如下:

Step 1: 随机产生初始值  $x_0$  (在  $[0, 1]$  之间且避免  $x_0$  落在  $(0.2, 0.4, 0.6, 0.8)$ ), 记入标记组, 即  $y(1) = x_0, i = 1, j = 1$ ;

Step 2: 按式(6)迭代产生一个  $x$  序列,  $i = i + 1$ ;

Step 3: 如果达到最大迭代次数, 则转向 Step 4; 否则, 如果  $x_i = (0, 0.25, 0.5, 0.75)$  或者  $x_i = x_{i-k}, k = \{0, 1, 2, 3, 4\}$ , 则按式  $x(i) = y(j+1) = y(j) + c$  改变迭代初值, 其中,  $c$  是随机数,  $j = j + 1$ ; 否则返回 Step 2;

Step 4: 运行停止, 保留  $x$  序列。

## 2.2 模拟退火算法

模拟退火算法是基于 Monte-Carlo 迭代求解策略的一种随机寻优算法, 它是结合概率突跳特性在解空间中随机寻找目标函数的全局最优解。

模拟退火算法流程如下:

1) 初始化。设置初始温度为  $T_0$ , 令  $T = T_0$ , 任取初始解  $S_1$ , 确定每个  $T$  时的迭代次数  $L$ 。

2) 对当前温度  $T$  和  $k = 1, 2, \dots, L$ , 重复 3)~6)。

3) 给当前解  $S_1$  一个随机扰动, 得到新解  $S_2$ ; 本文使用的扰动方式为

$$\Delta S = \frac{(1 + 10^T)^{|2rand() - 1|} - 1}{10^T} \cdot sign(2rand() - 1) \cdot (u - l),$$

其中,  $T$  表示当前温度,  $rand()$  表示 0 与 1 之间的随机数,  $sign(2rand() - 1)$  表示 -1 与 1 之间随机数的符号。

新解  $S_2 = S_1 + \Delta S$ 。

4) 计算增量  $df = f(S_2) - f(S_1)$ 。

5) 根据 Metropolis 准则, 公式为

$$P = \begin{cases} 1, & df < 0; \\ e^{-\frac{df}{T}}, & df \geq 0. \end{cases}$$

如果  $df < 0$ , 则接受新解; 否则, 以  $e^{-\frac{df}{T}}$  的概率接受新解。

6) 若满足终止条件, 则输出当前解  $S_1$  为最优解, 结束程序; 否则降低温度  $T$  转步骤 2)。

## 2.3 改进鸽群优化算法

针对鸽群优化算法易陷入局部最优、求解精度低、局部搜索能力差的不足, 本文提出了 Tent-PIO-SA 算法。使用 Tent 映射初始化种群, 使初始位置分布更加均匀, 扩大鸽群在空间中的搜索范围, 增加群体位置的多样性, 以提高初始解的质量。在地标算子运行后, 对现有的每只鸽子执行模拟退火算法, 充分利用模拟退火算法具有的跳出局部最优和搜索精度高的优点, 加强算法的局部搜索能力, 保障算法的搜索性能。

Tent-PIO-SA 算法流程如下:

1) 参数初始化。

2) 位置初始化。在搜索空间随机生成位置后, 根据式(6)进行 Tent 映射。

3) 计算适应度值, 找出全局最优位置  $X_{gbest}$ 。

4) 速度初始化。

5) 执行地图和指南针算子, 根据式(1)和式(2)更新鸽子的速度和位置。

6) 计算新位置的适应度, 并计算全局最优位置  $X_{gbest}$ 。

7) 判断是否达到地图和指南针算子最大迭代次数  $N_{c1}$ , 若达到则转 8), 否则  $t = t + 1$  转 3)。

8) 根据式(3)更新种群数目, 并根据式(4)计算位置中心  $X_{center}$ 。

9) 执行地标算子。根据式(5)更新鸽群位置, 并计算适应度值。

10) 对每只鸽子执行模拟退火算法。

11) 计算鸽子的适应度, 并更新当前搜索到的全局最优位置  $X_{gbest}$ 。

12) 判断是否达到地标算子最大迭代次数  $N_{c2}$ , 若达到则输出最优位置和对应的适应度值, 否则  $t = t + 1$  转 8)。

## 3 实验仿真与分析

### 3.1 测试函数

引入 16 个经典基准测试函数来验证本文提出的 Tent-PIO-SA 算法处理复杂函数问题的性能, 其中  $f_1 \sim f_6$  为单模态基准测试函数,  $f_7 \sim f_{12}$  为多模态基准测试函数,  $f_{13} \sim f_{16}$  为复合基准测试函数, 如表 1 所示。

表1 基准测试函数  
Tab.1 Benchmark functions

函数名	表达式	维度	搜索区间	理论最优值
Sphere	$f_1 = \sum_{i=1}^n x_i^2$	30	$[-100, 100]$	0
Schwefel 2.22	$f_2 = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30	$[-10, 10]$	0
Schwefel 1.2	$f_3 = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2$	30	$[-100, 100]$	0
Schwefel 2.21	$f_4 = \max \{  x_i , 1 \leq i \leq n \}$	30	$[-100, 100]$	0
Rosenbrock	$f_5 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2) + (x_i - 1)^2]$	30	$[-30, 30]$	0
Step	$f_6 = \sum_{i=1}^n (x_i + 0.5)^2$	30	$[-100, 100]$	0
Schwefel 2.26	$f_7 = \sum_{i=1}^n \left[ -x_i \sin \left( \sqrt{ x_i } \right) \right]$	30	$[-500, 500]$	$-418.9829 \times \text{dim}$
Rastrigin	$f_8 = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0
Ackley	$f_9 = -20 \exp \left( -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left( \frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$	30	$[-32, 32]$	0
Griewangk	$f_{10} = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$	30	$[-600, 600]$	0
Penalized	$f_{11} = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$	30	$[-50, 50]$	0
Generalized Penalized	$f_{12} = 0.1 \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	$[-50, 50]$	0
Shekel's Foxholes	$f_{13} = \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	2	$[-65, 65]$	0.998
Goldstein Price	$f_{14} = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]$	3
Shekel's Family	$f_{15} = -\sum_{i=1}^7 \left[ (X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	$[0, 10]$	-10.402 8
	$f_{16} = -\sum_{i=1}^{10} \left[ (X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	$[0, 10]$	-10.536 3

注: dim 为维度。

### 3.2 实验结果及分析

使用基本的PIO、用Tent映射初始化的PIO算法Tent-PIO以及用Tent映射初始化和加入模拟退火算法的Tent-PIO-SA算法求解16个经典基准测试函数。表2~表5中列出了这3种算法在最大迭代次数 $T$ 分别为500和1000,种群规模为30,独立运行30次条件下的实验结果,使用算法搜索到的最优值、平均值和标准差来评估算法的优劣。所有仿真都在Intel® Core(TM) i5-8500T CPU@2.10 GHz 8 GB RAM电脑平台的Matlab(R2014a)中运行。

测试函数中 $f_1 \sim f_6$ 为单峰函数,这类函数只有一个全局最优解。由表2可以看出:对单峰函数 $f_1 \sim f_6$ ,Tent-PIO算法相比PIO算法并不具有优势,在PIO算法初始化时加入Tent映射只能增强种群的多样性,并不能提高算法的全局寻优能力。对 $f_1 \sim f_4$ ,本文提出的Tent-PIO-SA算法比PIO、Tent-PIO算法平均精度分别提高了20和19个数量级,9和12个数量级,22和21个数量级,3和4个数量级。 $f_5$ 是一个极复杂的病态函数,全局最优解位置在一条极其狭窄的峡谷上,而本文算法在求解 $f_5$ 时达到 $1.15E-06$ ,比PIO

的16.995 395 45、Tent-PIO的1.54E+01高出了7个数量级。对于 $f_6$ ，Tent-PIO-SA算法比PIO、Tent-PIO算法也高出了6个数量级。这些说明Tent-PIO-SA算法在处理单峰测试函数上比PIO、Tent-PIO算法有更高的寻优能力。

表 2 3种算法求解函数 $f_1 \sim f_{12}$ 的运行结果对比( $T=500$ )  
Tab. 2 Comparison of experimental results solving  $f_1 \sim f_{12}$  with 3 algorithms ( $T=500$ )

函数	算法	最优值	平均值	标准差
$f_1$	PIO	2.00E-196	3.14E-151	1.692E-150
	Tent-PIO	3.59E-198	7.50E-150	4.036 4E-149
	Tent-PIO-SA	6.83E-196	1.22E-171	0
$f_2$	PIO	2.09E-95	1.06E-77	5.678E-77
	Tent-PIO	2.49E-97	3.27E-74	1.761 86E-73
	Tent-PIO-SA	5.72E-103	2.54E-86	1.366 31E-85
$f_3$	PIO	2.78E-183	4.54E-149	2.445 6E-148
	Tent-PIO	4.73E-192	3.78E-150	2.035 8E-149
	Tent-PIO-SA	5.24E-202	4.87E-171	0
$f_4$	PIO	3.55E-100	6.44E-80	3.378 23E-79
	Tent-PIO	1.49E-101	5.59E-79	2.597 94E-78
	Tent-PIO-SA	2.42E-05	8.56E-83	4.608 8E-82
$f_5$	PIO	0.000 772 57	16.995 395 45	12.812 21
	Tent-PIO	7.26E-03	1.54E+01	13.430 22
	Tent-PIO-SA	9.32E-11	1.15E-06	1.942 93E-06
$f_6$	PIO	1.020 7E-07	0.324 232 058	0.812 901
	Tent-PIO	1.265 2E-08	0.302 873 384	0.828 241
	Tent-PIO-SA	1.57E-12	2.09E-07	3.353 35E-07
$f_7$	PIO	-12 569.484 7	-9 511.504 05	2 591.076 9
	Tent-PIO	-12 569.486 3	-10 445.951 43	2 029.787 82
	Tent-PIO-SA	-12 569.486 6	-12 569.486 6	1.819E-12
$f_8$	PIO	0	0	0
	Tent-PIO	0	0	0
	Tent-PIO-SA	0	0	0
$f_9$	PIO	8.881 8E-16	8.881 8E-16	3.944E-31
	Tent-PIO	8.881 8E-16	8.881 8E-16	3.944E-31
	Tent-PIO-SA	8.881 8E-16	8.881 8E-16	3.944E-31
$f_{10}$	PIO	0	0	0
	Tent-PIO	0	0	0
	Tent-PIO-SA	0	0	0
$f_{11}$	PIO	5.10E-09	8.03E-02	0.234 897
	Tent-PIO	5.177 7E-08	0.012 429 586	0.027 078 31
	Tent-PIO-SA	1.68E-12	1.66E-09	2.444 45E-09
$f_{12}$	PIO	3.764E-08	0.079 508	0.156 362
	Tent-PIO	1.227 5E-07	0.128 501 398	0.232 099
	Tent-PIO-SA	5.839 4E-13	1.45E-08	2.178 58E-08

测试函数中 $f_7 \sim f_{12}$ 为多峰函数，这类函数有多个局部最优解，寻优算法在求解时易于陷入局部最优。由表2中求解多峰函数结果的均值分析发现：在求解 $f_8, f_9, f_{10}$ 时，3种算法寻优性能相当，而在求解 $f_7$ 时，Tent-PIO算法略优于算法PIO，Tent-PIO-SA算法每次都找到最优解-12 569.486 6，求解精度和稳定性极好。对于 $f_{11}$ 和 $f_{12}$ ，Tent-PIO算法相比PIO算法并不具有优势，而Tent-PIO-SA算法比PIO、Tent-PIO算法的平均精度分别都提高了7个数量级。这充分说明，在Tent-PIO-SA算法中模拟退火算法充分发挥了它具有的跳出局部最优和搜索

精度高的特性，这恰好解决了鸽群优化算法在搜索过程中易陷入局部最优的不足，从而使Tent-PIO-SA算法具有了更好的寻优能力。

表 3 3种算法对于函数 $f_{13} \sim f_{16}$ 的运行结果对比( $T=500$ )  
Tab. 3 Comparison of experimental results solving  $f_{13} \sim f_{16}$  with 3 algorithms ( $T=500$ )

函数	算法	最优值	平均值	标准差
$f_{13}$	PIO	0.998	0.998 01	2.657 48E-05
	Tent-PIO	0.998	0.998 007	2.352 3E-05
	Tent-PIO-SA	0.998	0.998	4.440 89E-16
$f_{14}$	PIO	3.001 3	3.028 157	0.018 729
	Tent-PIO	3.000 1	3.008 607	0.008 07
	Tent-PIO-SA	3.000 1	3.006 279	0.006 726
$f_{15}$	PIO	-10.135	-5.465 98	1.256 172
	Tent-PIO	-10.398	-5.814 2	1.627 722
	Tent-PIO-SA	-10.402 8	-10.402 8	8.881 78E-15
$f_{16}$	PIO	-10.174 6	-5.82E+00	1.410 056
	Tent-PIO	-10.536 3	-6.192 236 7	2.019 421
	Tent-PIO-SA	-10.536 3	-10.536 3	0

表 4 3种算法求解函数 $f_1 \sim f_{12}$ 的运行结果对比( $T=1 000$ )  
Tab. 4 Comparison of experimental results solving  $f_1 \sim f_{12}$  with 3 algorithms ( $T=1 000$ )

函数	算法	最优值	平均值	标准差
$f_1$	PIO	0	0	0
	Tent-PIO	0	0	0
	Tent-PIO-SA	0	0	0
$f_2$	PIO	9.27E-232	1.99E-166	0.00E+00
	Tent-PIO	2.70E-225	9.61E-168	0.00E+00
	Tent-PIO-SA	6.78E-232	1.38E-199	0.00E+00
$f_3$	PIO	0	0	0
	Tent-PIO	0	0	0
	Tent-PIO-SA	0	0	0
$f_4$	PIO	1.10E-216	5.17E-171	0.00E+00
	Tent-PIO	1.65E-232	6.94E-162	0.00E+00
	Tent-PIO-SA	2.19E-239	9.39E-201	0.00E+00
$f_5$	PIO	0.036 114	16.333 25	13.123 25
	Tent-PIO	0.002 983	16.872 8	12.295 83
	Tent-PIO-SA	9.00E-11	2.21E-07	2.49E-07
$f_6$	PIO	5.836E-08	0.255 433	0.650 171 54
	Tent-PIO	1.311 9E-07	0.138 118	0.634 742 453
	Tent-PIO-SA	1.16E-10	9.54E-09	9.24E-09
$f_7$	PIO	-12 569.5	-10 318.5	2 089.664
	Tent-PIO	-12 569.5	-10 333	2 065.601
	Tent-PIO-SA	-12 569.486 6	-12 569.486 6	1.818 99E-12
$f_8$	PIO	0	0	0
	Tent-PIO	0	0	0
	Tent-PIO-SA	0	0	0
$f_9$	PIO	8.881 8E-16	8.881 8E-16	3.944E-31
	Tent-PIO	8.881 8E-16	8.881 8E-16	3.944E-31
	Tent-PIO-SA	8.881 8E-16	8.881 8E-16	3.944E-31
$f_{10}$	PIO	0	0	0
	Tent-PIO	0	0	0
	Tent-PIO-SA	0	0	0
$f_{11}$	PIO	1.38E-08	9.29E-02	3.08E-01
	Tent-PIO	1.08E-09	0.058 441	0.277 719 6
	Tent-PIO-SA	6.58E-16	9.44E-11	8.91E-11
$f_{12}$	PIO	2.527E-08	0.086 947	0.207 709 91
	Tent-PIO	2.998 1E-07	0.105 731	0.251 810 277
	Tent-PIO-SA	6.15E-14	1.20E-09	1.22E-09

测试函数中 $f_{13} \sim f_{16}$ 为复合基准测试函数，是固定维多峰函数，这类函数有多个局部最优解。由表3

可以看出,对于函数 $f_{13} \sim f_{16}$ , Tent-PIO算法无论在最优值还是平均值上都优于PIO算法,优化效果更好一些。Tent-PIO-SA算法除对 $f_{14}$ 外,对函数 $f_{13}, f_{15}, f_{16}$ 每次运行时都收敛到了其理论最优值,表现出更强的寻优能力和稳定性。特别对于函数 $f_{15}$ 和 $f_{16}$ , PIO和Tent-PIO算法极易陷入局部最优解(1, 1, 1),而Tent-PIO-SA算法能以100%的成功率找到全局最优值-10.402 8和-10.536 3,说明模拟退火算法充分发挥了它跳出局部最优的优势。

由表4可以看出,当最大迭代次数提高到1 000时,Tent-PIO算法相比PIO算法并不具有明显优势,Tent-PIO-SA算法找到了函数 $f_1$ 和 $f_3$ 的理论最优解0, $f_2$ 和 $f_4$ 获得的平均最优解的精度也有了大幅度的提高。Tent-PIO-SA算法比PIO、Tent-PIO算法的平均精度都提高了30多个数量级,并且标准差达到了0.00E+00。对比表2,Tent-PIO-SA算法在运算函数 $f_5, f_6, f_{11}$ 和 $f_{12}$ 上的平均精度也有了一定的提高,但PIO、Tent-PIO算法在加大迭代次数时平均值基本没有变化。这说明Tent-PIO-SA算法比PIO、Tent-PIO算法有更强的寻优能力,模拟退火算法在Tent-PIO-SA

算法中发挥了较大的作用。

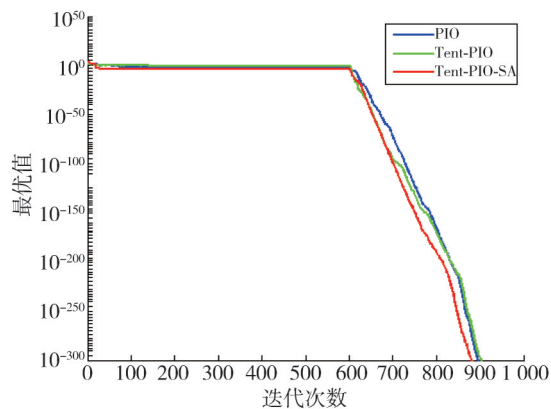
表5 3种算法求解函数 $f_{13} \sim f_{16}$ 的运行结果对比( $T=1\ 000$ )

Tab. 5 Comparison of experimental results solving  $f_{13} \sim f_{16}$  with 3 algorithms ( $T=1\ 000$ )

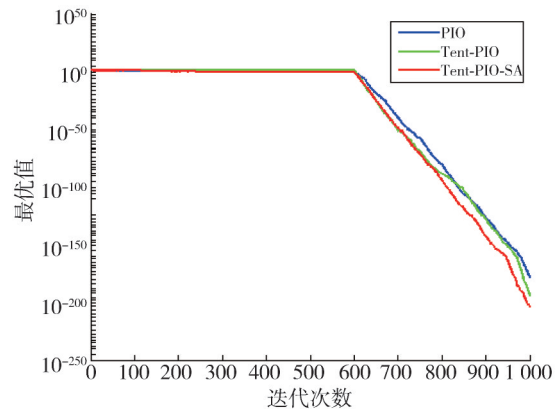
函数	算法	最优值	平均值	标准差
$f_{13}$	PIO	0.998	0.998	4.440 89E-16
	Tent-PIO	0.998	0.998	4.440 89E-16
	Tent-PIO-SA	0.998	0.998	4.440 89E-16
$f_{14}$	PIO	3.000 5	3.005 65	0.006 88
	Tent-PIO	3.000 4	3.001 833 33	0.002 508 43
	Tent-PIO-SA	3.000 3	3.000 533	0.000 236
$f_{15}$	PIO	-5.087 7	-5.087 636 364	9.791 21E-05
	Tent-PIO	-10.356 2	-5.764 764	1.580 099 9
	Tent-PIO-SA	-10.402 8	-10.402 8	0
$f_{16}$	PIO	-10.500 7	-6.202 94	2.148 88
	Tent-PIO	-10.536 3	-8.372 22	2.648 486 767
	Tent-PIO-SA	-10.536 3	-10.536 3	0

由表5可以看出,当最大迭代次数提高到1 000时,PIO、Tent-PIO算法对于函数 $f_{13} \sim f_{16}$ 的平均精度都有一定提高,但仍找不到 $f_{15}$ 和 $f_{16}$ 的最优值。Tent-PIO-SA算法在500代时就找到了函数 $f_{13}, f_{15}$ 和 $f_{16}$ 的理论最优值,说明Tent-PIO-SA具有更好的寻优能力和更快的收敛速度。

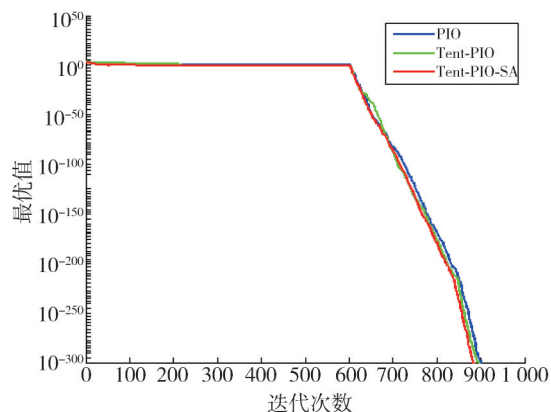
图1给出了16个基准函数在PIO、Tent-PIO、Tent-PIO-SA这3种优化算法下的收敛曲线。



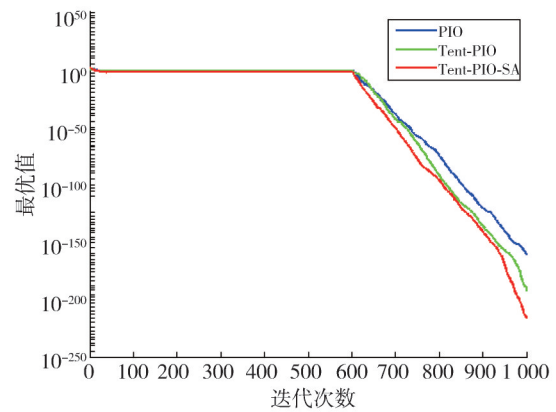
(a)  $f_1$ 收敛曲线



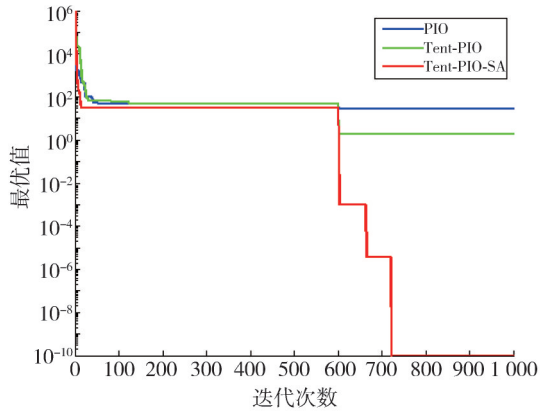
(b)  $f_2$ 收敛曲线



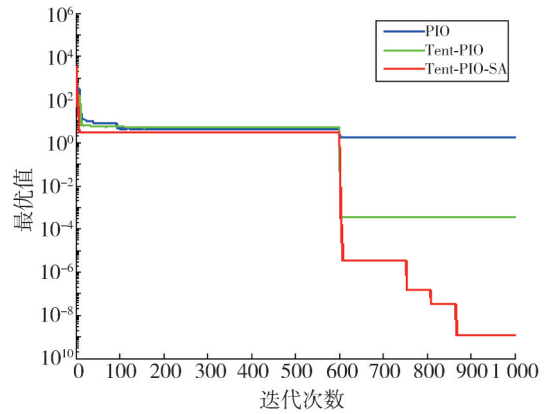
(c)  $f_3$ 收敛曲线



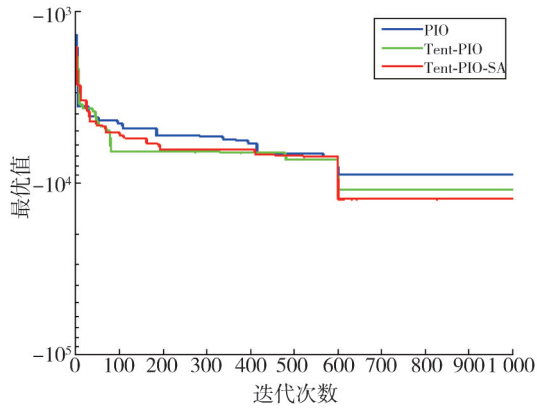
(d)  $f_4$ 收敛曲线



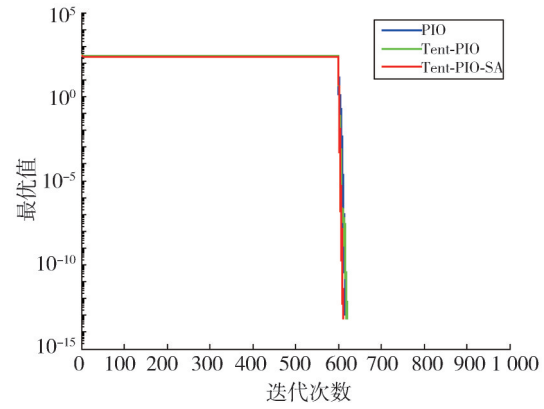
(e)  $f_5$ 收敛曲线



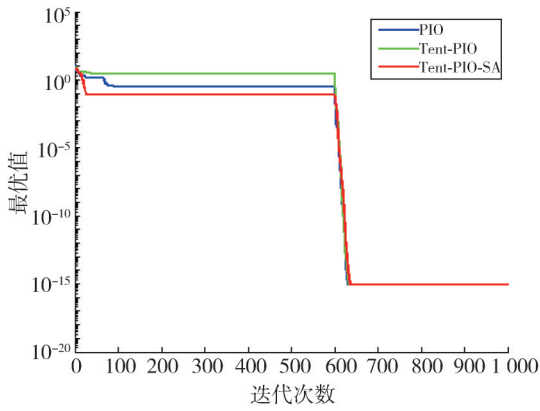
(f)  $f_6$ 收敛曲线



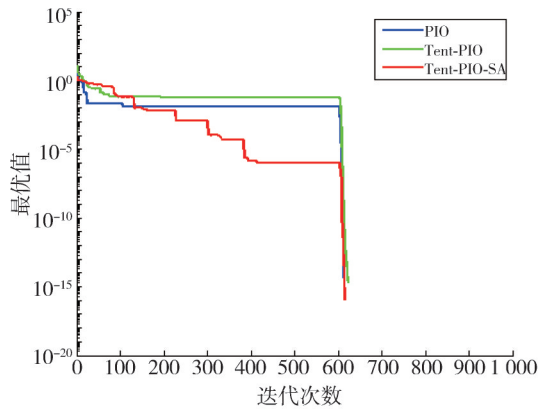
(g)  $f_7$ 收敛曲线



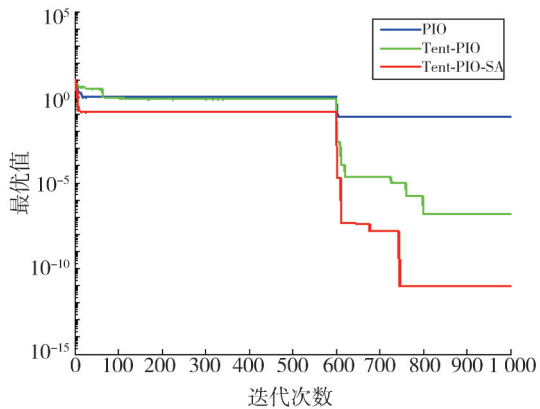
(h)  $f_8$ 收敛曲线



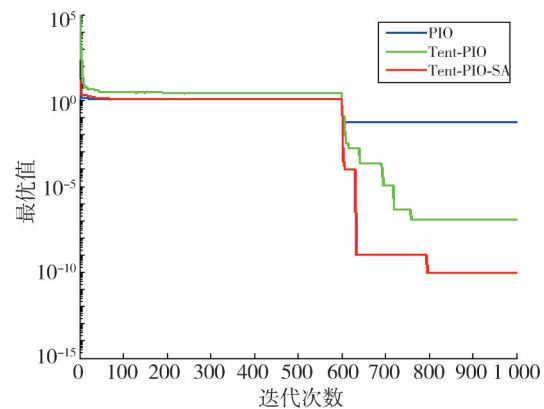
(i)  $f_9$ 收敛曲线



(j)  $f_{10}$ 收敛曲线



(k)  $f_{11}$ 收敛曲线



(l)  $f_{12}$ 收敛曲线

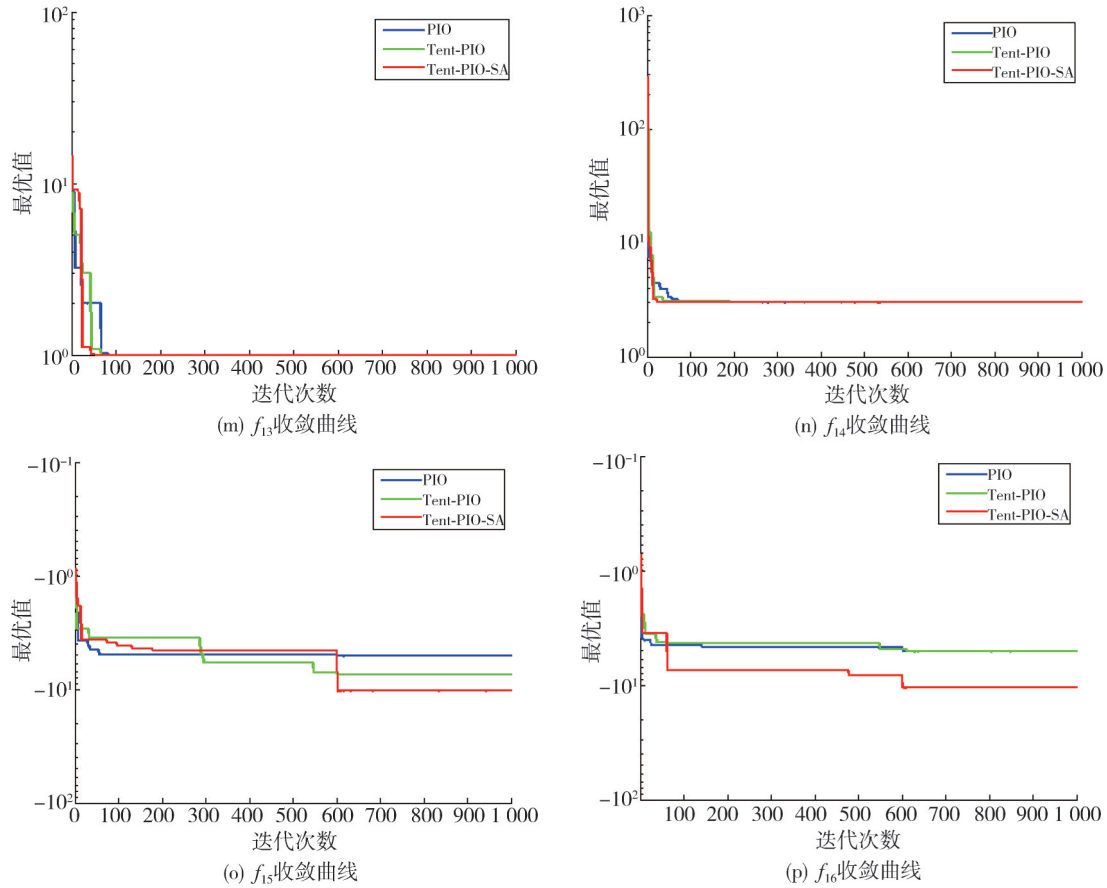
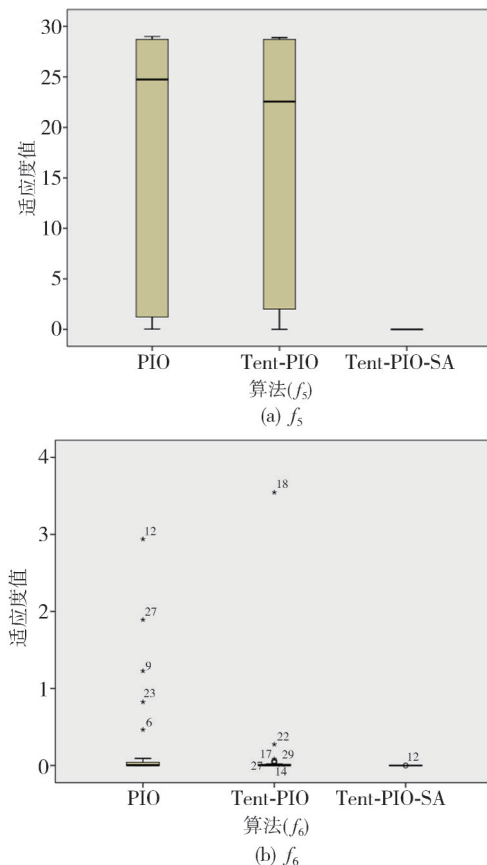


图1 3种算法在16个函数上的收敛曲线  
Fig. 1 Convergence curves of three algorithms on sixteen functions

由图1可以看出,对于函数 $f_1 \sim f_4$ , Tent-PIO-SA算法比PIO、Tent-PIO算法有更快的收敛速度和更高的收敛精度。对于更为复杂的函数 $f_5, f_6, f_7, f_{11}, f_{12}, f_{15}$ 和 $f_{16}$ , PIO、Tent-PIO在迭代到600代左右时就丧失了寻优能力,不再更新最优解,而Tent-PIO-SA算法在前600代的优势并不明显,但600代后还在不断更新最优解,表现出了强大的寻优能力。这正是因为在600代后进入鸽群优化算法的地标算子阶段,而在此阶段融入了模拟退火算法,极大地增强了整个算法的寻优能力,使算法在后期可以不断更新全局最优值,所以Tent-PIO-SA算法在收敛精度上有很大优势。

图2为PIO、Tent-PIO、Tent-PIO-SA这三种优化算法求解5个具有代表性的基准测试函数时,每个函数独立运行30次所得最优解的箱式图。

由图2可知,Tent-PIO-SA算法在求解 $f_5, f_6, f_7, f_{11}$ 和 $f_{12}$ 时,均无异常点出现。并且Tent-PIO-SA算法在求解这5个函数时,收敛值的分布非常集中,明显优于其他算法,这说明Tent-PIO-SA算法在求解问题时具有较好的稳定性和鲁棒性。



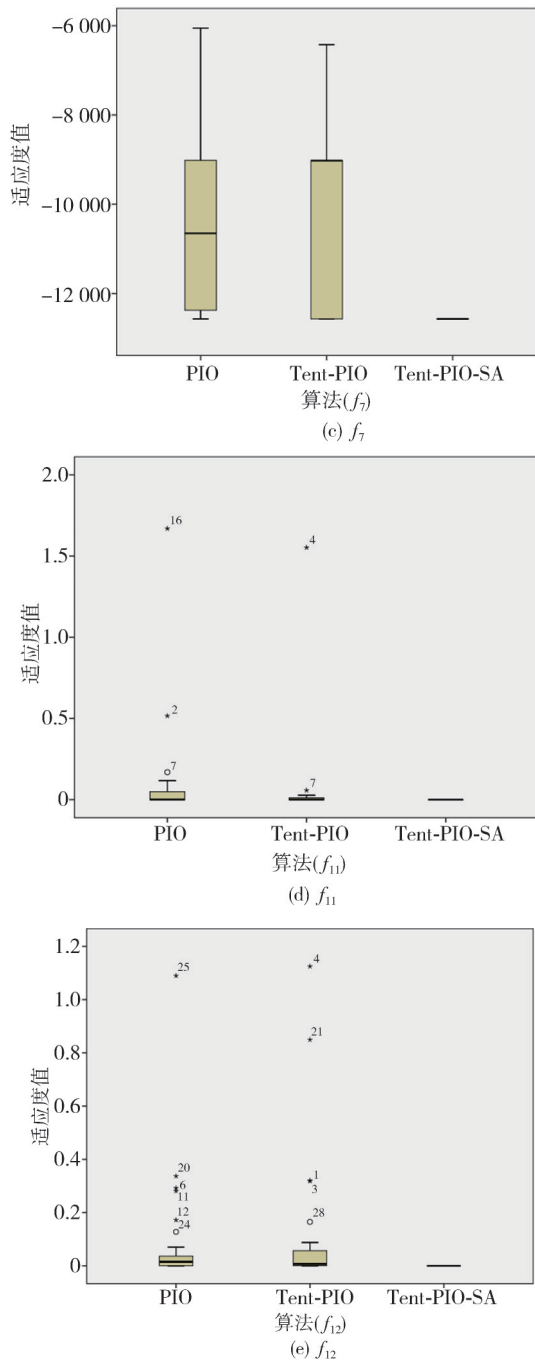


图 2 3种算法在5个函数上的收敛箱式图

Fig. 2 Convergence box diagrams of three algorithms on five functions

为了更好地验证所提出的改进算法的优越性,对比以下经典算法:一种改进的采用莱维飞行策略的条件化更新的樽海鞘群算法(Levy Flight Based Conditional Updating Salp Swarm Algorithm, LECUSSA)<sup>[12]</sup>、自学习策略和Levy飞行的正弦余弦优化算法(Sine Cosine Optimization Algorithm with Self-Learning Strategy and Lévy Flight, SCASL)<sup>[13]</sup>、混沌多精英鲸鱼优化算法(Chaotic Multi-Leader Whale Optimization Algorithm, CML-

WOA)<sup>[14]</sup>、基于自适应参数及小生境技术的改进鲸鱼优化算法(Whale Optimization Algorithm Based on Adaptive Parameters and Niche, APN-WOA)<sup>[15]</sup>、基于鸽群算法的Fuch混沌蝗虫算法(Fuch Chaotic Grasshopper Algorithm Based on Pigeon Swarm Algorithm, PFGOA)<sup>[16]</sup>、基于Tent映射的混合灰狼优化的改进算法(Grey Wolf Optimization Algorithm Based on Particle Swarm Optimization, PSO-GWO)<sup>[17]</sup>、教与学信息交互粒子群优化(Teaching and Learning Information Interactive PSO, TLPSO)算法<sup>[18]</sup>、Hybridizing Sine Cosine Algorithm with Differential Evolution(SCA-DE)<sup>[19]</sup>、闫群民等提出的算法<sup>[20]</sup>。为了保证算法对比结果的公平性,本文算法与参考文献设置相同的最大迭代次数和种群规模。表6中,种群规模均为30,最大迭代次数均为500。表7中,种群规模均为30,最大迭代次数均为1000。

由表6可知,对于高维单峰函数 $f_1 \sim f_4$ ,本文提出的Tent-PIO-SA算法在寻优能力上优于LECUSSA,而劣于SCASL、CML-WOA、APN-WOA、PFGOA。对于函数 $f_8, f_9, f_{10}$ , Tent-PIO-SA算法优于LECUSSA,与其他算法寻优性能相当,都具有很好的稳定性。但是,对于更复杂的函数 $f_6, f_7, f_{11}$ 和 $f_{12}$ ,本文算法的平均值和标准差比其他算法至少高出3个数量级,相较于其他算法有明显优势。尤其对于极难优化的函数 $f_5$ ,本文算法平均值达到了 $1.15E-06$ ,比LECUSSA高出了6个数量级,比SCASL高出了4个数量级,比CML-WOA高出了5个数量级,比APN-WOA高出了7个数量级。因此说明在鸽群优化算法的地标算子中加入模拟退火算法,加强了整个算法的局部搜索能力,且有效利用了模拟退火算法具有的概率突跳的特性,使Tent-PIO-SA算法具有了更好的寻优能力。

由表7可以看出,对于单峰函数 $f_1$ 和 $f_3$ ,本文的Tent-PIO-SA算法找到了理论最优值0, $f_2$ 和 $f_4$ 虽然劣于SCA-DE,但平均值也分别达到了 $1.38E-199$ 和 $9.39E-201$ ,标准差达到了0,说明Tent-PIO-SA算法具有较好的搜索能力和稳定性。对于函数 $f_6$ , Tent-PIO-SA算法劣于TLPSO,而优于SCA-DE。对于函数 $f_7 \sim f_{12}$ , Tent-PIO-SA算法的精度都高于其他算法。在函数 $f_5$ 上,本文算法具有绝对优势,比TLPSO、SCA-DE、文献[20]算法高出了7个数量级。

综合表2~表7可以看出,本文提出的Tent-

PIO-SA算法对于复杂函数 $f_5 \sim f_{16}$ 具有良好的寻优能力, 特别对于极其难以优化的病态函数 $f_5$ 优化效果

显著, 说明加入模拟退火算法能有效提高鸽群优化算法的寻优能力。

表6 各算法运行结果比较( $T=500$ )

Tab. 6 Comparison of experimental results of different algorithms( $T=500$ )

函数	结果	Tent-PIO-SA	LECUSSA	SCASL	CML-WOA	APN-WOA	PFGOA
$f_1$	平均值	1.22E-171	2.65E-13	0.00E+00	0	0.00E+000	0.0E+00
	标准差	0	3.94E-13	0.00E+00	0	0.00E+000	0.0E+00
$f_2$	平均值	2.54E-86	5.36E-08	0.00E+00	2.60E-160	2.27E-245	0.00E+00
	标准差	1.366 31E-85	5.76E-08	0.00E+00	0	0.00E+000	0.00E+00
$f_3$	平均值	4.87E-171	1.30E-12	0.00E+00	3.55E-297	0.00E+000	0.00E+00
	标准差	0	3.13E-12	0.00E+00	0	0.00E+000	0.00E+00
$f_4$	平均值	8.56E-83	1.10E-07	0.00E+00	0	3.34E-244	0.00E+00
	标准差	4.608 8E-82	1.11E-07	0.00E+00	0	0.00E+000	0.00E+00
$f_5$	平均值	1.15E-06	8.949 2	6.39E-02	2.93E-01	2.78E+001	—
	标准差	1.942 93E-06	4.01E-02	3.44E-02	1.02	3.13E-001	—
$f_6$	平均值	2.09E-07	1.03E-03	2.49E-04	8.60E-03	—	6.8E-04
	标准差	3.353 35E-07	1.55E-03	4.37E-04	1.89E-02	—	1.3E-03
$f_7$	平均值	-12 569.486 6	-3 005.39	-1.26E+04	—	-1.23E+004	-12 569.39
	标准差	1.819E-12	0.131	2.41E+00	—	3.25E+002	5.3E-02
$f_8$	平均值	0	2.98E-14	0.00E+00	0	0.00E+000	0.0E+00
	标准差	0	5.26E-14	0.00E+00	0	0.00E+000	0.0E+00
$f_9$	平均值	8.881 8E-16	8.75E-09	8.88E-16	8.88E-16	8.88E-016	8.9E-16
	标准差	3.944E-31	8.33E-08	0.00E+00	0	0.00E+000	3.9E-31
$f_{10}$	平均值	0	4.11E-13	0.00E+00	0	0.00E+000	0.0E+00
	标准差	0	8.46E-13	0.00E+00	0	0.00E+000	0.0E+00
$f_{11}$	平均值	1.66E-09	1.90E-04	2.30E-05	4.25E-05	1.82E-002	1.3E-05
	标准差	2.444 45E-09	2.04E-04	4.63E-05	2.97E-04	6.60E-003	1.2E-05
$f_{12}$	平均值	1.45E-08	0.932 77	1.22E-04	1.34E-32	2.45E-001	—
	标准差	2.178 58E-08	0.132 04	4.02E-04	5.56E-48	7.37E-002	—

注: “—”表示参考文献未给出。

表7 与其他算法运行结果比较( $T=1\ 000$ )

Tab. 7 Comparison of experimental results with other algorithms( $T=1\ 000$ )

函数	结果	Tent-PIO-SA	TLPSO	SCA-DE	文献[20]
$f_1$	平均值	0	2.39E-167	0.00E+00	0.00E+00
	标准差	0	0	0.00E+00	0.00E+00
$f_2$	平均值	1.38E-199	3.92E-83	0.00E+00	—
	标准差	0	1.93E-83	0.00E+00	—
$f_3$	平均值	0	8.31E-34	0.00E+00	—
	标准差	0	1.04E-33	0.00E+00	—
$f_4$	平均值	9.39E-201	5.30E-67	0.00E+00	—
	标准差	0	2.12E-67	0.00E+00	—
$f_5$	平均值	2.21E-07	5.75E+00	2.34E+01	2.87E+01
	标准差	2.49E-07	8.98E-01	3.17E-01	4.03E-01
$f_6$	平均值	9.54E-09	4.55E-31	8.39E-08	—
	标准差	9.24E-09	1.90E-31	9.28E-08	—
$f_7$	平均值	-12 569.486 6	—	-1.24E+04	4.05E+03
	标准差	1.819E-12	—	3.18E+02	6.62E+02
$f_8$	平均值	0	0	0.00E+00	3.63E+00
	标准差	0	0	0.00E+00	8.68E+00
$f_9$	平均值	8.881 8E-16	4.44E-15	5.57E-15	8.88E-16
	标准差	3.944E-31	0	1.83E-15	5.07E-22
$f_{10}$	平均值	0	0	0.00E+00	4.83E-02
	标准差	0	0	0.00E+00	2.64E-01
$f_{11}$	平均值	9.44E-11	4.04E-07	4.07E-09	—
	标准差	8.91E-11	7.56E-07	2.64E-09	—
$f_{12}$	平均值	1.20E-09	—	4.33E-08	—
	标准差	1.22E-09	—	6.01E-08	—

注: “—”表示参考文献未给出。

## 4 结 论

本文针对PIO算法易陷入局部最优、求解精度低和局部搜索能力差的缺陷, 提出了一种融合Tent映射和模拟退火的改进鸽群优化算法。利用Tent映射初始化种群, 比随机产生的种群分布更均匀。在鸽群优化算法的地标算子阶段加入模拟退火算法, 增强了算法的局部搜索能力, 利用模拟退火算法的优势弥补了鸽群优化算法的不足。16个基准函数的仿真实验结果、收敛曲线以及收敛箱式图都表明, 改进鸽群优化算法的稳定性、收敛精度都有大幅度提高, 充分说明这些改进策略能提高算法的寻优能力。

### 参考文献:

- [1] DUAN H B, QIAO P X. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning [J]. International Journal of Intelligent Computing and Cybernetics, 2014, 7(1): 24-37.
- [2] ZHANG X M, DUAN H B, YANG C. Pigeon-inspired optimization approach to multiple UAVs formation reconfiguration controller design [C] // Proceedings of

- 2014 IEEE Chinese Guidance, Navigation and Control Conference. Piscataway: IEEE Press, 2014: 2707-2712.
- [ 3 ] SUN Y B, XIAN N, DUAN H B. Linear-quadratic regulator controller design for quadrotor based on pigeon-inspired optimization[J]. Aircraft Engineering and Aerospace Technology, 2016, 88(6): 761-770.
- [ 4 ] DOU R, DUAN H B. Pigeon inspired optimization approach to model prediction control for unmanned air vehicles[J]. Aircraft Engineering and Aerospace Technology, 2016, 88(1): 108-116.
- [ 5 ] DENG Y M, DUAN H B. Control parameter design for automatic carrier landing system via pigeon-inspired optimization[J]. Nonlinear Dynamics, 2016, 85(1): 97-106.
- [ 6 ] ZHANG D F, DUAN H, YANG Y J, et al. Active disturbance rejection control for small unmanned helicopters via levy flightbased pigeon-inspired optimization[J]. Aircraft Engineering and Aerospace Technology, 2017, 89(8): 946-952.
- [ 7 ] DUAN H B, WANG X H. Echo state networks with orthogonal pigeon-inspired optimization for image restoration[J]. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(11): 2413-2425.
- [ 8 ] HE H X, DUAN H B. A multi-strategy pigeon-inspired optimization approach to active disturbance rejection control parameters tuning for vertical take-off and landing fixed-wing UAV[J]. Chinese Journal of Aeronautics, 2022, 35(1): 19-30.
- [ 9 ] SUN X X, PAN J S, CHU S C, HU P, and TIAN A Q. A novel pigeon-inspired optimization with quasi-affine transformation evolutionary algorithm for DV-Hop in wireless sensor networks[J]. International Journal of Distributed Sensor Networks, 2020, 16(6): 1-15.
- [10] HUO M Z, DUAN H B. An adaptive mutant multi-objective pigeon-inspired optimization for unmanned aerial vehicle target search problem [J]. Control Theory and Applications, 2020, 37(3): 584-591.
- [11] 胡耀龙, 冯强, 海星朔, 等. 基于自适应学习策略的改进鸽群优化算法[J]. 北京航空航天大学学报, 2020, 46(12): 2348-2355.
- HU Yaolong, FENG Qiang, HAI Xingshuo, et al. Improved pigeon-inspired optimization algorithm based on adaptive learning strategy [J]. Journal of Beijing University of Aeronautics and Astronautics, 2020, 46(12): 2348-2355. (in Chinese)
- [12] 张严, 秦亮曦. 基于 Levy 飞行策略的改进樽海鞘群算法[J]. 计算机科学, 2020, 47(7): 154-160.
- ZHANG Yan, QIN Liangxi. Improved salp swarm algorithm based on levy flight strategy [J]. Computer Science, 2020, 47(7): 154-160. (in Chinese)
- [13] 李银通, 韩统, 赵辉, 等. 自学习策略和 Levy 飞行的正弦余弦优化算法[J]. 重庆大学学报, 2019, 42(9): 56-66.
- LI Yintong, HAN Tong, ZHAO Hui, et al. An improved sine cosine optimization algorithm with self-learning strategy and Levy flight [J]. Journal of Chongqing University, 2019, 42(9): 56-66. (in Chinese)
- [14] 汤安迪, 韩统, 徐登武, 等. 混沌多精英鲸鱼优化算法[J]. 北京航空航天大学学报, 2021, 47(7): 1481-1494.
- TANG Andi, HAN Tong, XU Dengwu, et al. Chaotic multi-leader whale optimization algorithm [J]. Journal of Beijing University of Aeronautics and Astronautics, 2021, 47(7): 1481-1494. (in Chinese)
- [15] 刘亮, 何庆. 一种求解函数优化问题的改进鲸鱼优化算法[J]. 计算机应用研究, 2020, 37(4): 1004-1009.
- LIU Liang, HE Qing. Improved whale optimization algorithm for solving function optimization problems [J]. Application Research of Computers, 2020, 37(4): 1004-1009. (in Chinese)
- [16] 尹德鑫, 张达敏, 蔡朋宸, 等. 基于鸽群算法的 Fuch 混沌蝗虫算法[J]. 计算机应用研究, 2021, 38(7): 2013-2017.
- YIN Dexin, ZHANG Damin, CAI Pengchen, et al. Fuch chaotic grasshopper algorithm based on pigeon swarm algorithm[J]. Application Research of Computers, 2021, 38(7): 2013-2017. (in Chinese)
- [17] 腾志军, 吕金玲, 郭力文, 等. 一种基于 Tent 映射的混合灰狼优化的改进算法[J]. 哈尔滨工业大学学报, 2018, 50(11): 40-49.
- TENG Zhijun, LÜ Jinling, GUO Liwen, et al. An improved hybrid grey wolf optimization algorithm based on Tent mapping[J]. Journal of Harbin Institute of Technology, 2018, 50(11): 40-49. (in Chinese)
- [18] 聂方鑫, 王宇嘉, 贾欣. 教与学信息交互粒子群优化算法[J]. 计算机应用, 2022, 42(3): 874-882.
- NIE Fangxin, WANG Yujia, JIA Xin. Teaching and learning information interactive particle swarm optimization algorithm [J]. Journal of Computer Applications, 2022, 42(3): 874-882. (in Chinese)
- [19] NENAVATH H, JATOTH R K. Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking [J]. Applied Soft Computing, 2018, 62: 1019-1043.