

# 基于阶梯式Tent混沌和模拟退火的樽海鞘群算法

周 鹏<sup>1,2</sup>,董朝轶<sup>1,2</sup>,陈晓艳<sup>1,2</sup>,齐咏生<sup>1,2</sup>,赵肖懿<sup>1,2</sup>,王启来<sup>1,2</sup>

(1. 内蒙古工业大学,内蒙古呼和浩特010080;2. 内蒙古机电控制重点实验室(内蒙古工业大学),内蒙古呼和浩特010051)

**摘 要:** 针对樽海鞘群算法寻优迭代过程中存在容易陷入局部最优、收敛速度慢的问题,提出一种改进的樽海鞘群算法.引入Tent混沌映射初始化种群来提高算法迭代前期的收敛速度,通过惯性权值“阶梯式”调整策略来更好地兼顾算法全局探索能力和局部开发能力,通过模拟退火增强樽海鞘群算法迭代后期跳出局部最优解的能力,以基准测试函数和磁导航自动导引车模糊控制器参数寻优问题为例测试了算法性能.仿真结果表明,对于单峰和多峰测试函数,改进后的樽海鞘群算法具有更快的收敛速度和更强的全局寻优能力.相比较标准樽海鞘群算法的参数调节法,改进后的樽海鞘群算法所设计的磁导航自动导引车模糊控制器对磁偏差值控制性能更为优化,在控制器设计方面具有潜在的应用价值.

**关键词:** 樽海鞘群算法; Tent混沌映射; 阶梯式权值; 模拟退火; 系统辨识; 自动导引车

**中图分类号:** TP301.6 **文献标识码:** A **文章编号:** 0372-2112(2021)09-1724-12

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.12263/DZXB.20200593

## A Salp Swarm Algorithm Based on Stepped Tent Chaos and Simulated Annealing

ZHOU Peng<sup>1,2</sup>, DONG Chao-yi<sup>1,2</sup>, CHEN Xiao-yan<sup>1,2</sup>, QI Yong-sheng<sup>1,2</sup>, ZHAO Xiao-yi<sup>1,2</sup>, WANG Qi-lai<sup>1,2</sup>

(1. Inner Mongolia University of Technology, Hohhot, Inner Mongolia 010080, China;

2. Inner Mongolia Electromechanical Control Laboratory, Hohhot, Inner Mongolia 010051, China)

**Abstract:** To solve the problems of local optimization and slow convergence in the process of optimization and iteration, the paper proposed an improved optimization algorithm for a salp swarm, i.e. step-by-step tent chaos simulated annealing salp swarm algorithm (STSA-SSA). Firstly, an initial population of Tent chaotic map was introduced to enhance the algorithm convergence at the early stage of iteration, and a step adjustment strategy of inertia weights was employed to improve the global and local exploring ability of the STSA-SSA. Then, the ability of escaping local optimal solutions of the STSA-SSA at the later stage of iterations was increased by a simulated annealing policy. Finally, the performance of the STSA-SSA was tested in the processes of optimizing the parameters of benchmark functions and a fuzzy controller a magnetic navigation automated guided vehicle (AGV). The results show that for the single peak and multi peak test functions, the STSA-SSA has faster convergence speed and stronger global optimization ability. Compared with SSA, the fuzzy controller of the magnetic navigation AGV designed by STSA-SSA is more optimized. Therefore, the STSA-SSA has a potential engineering application value in controller designs.

**Key words:** salp swarm algorithm; tent chaotic mapping; step weight; simulated annealing; system identification; AGV

### 1 引言

近些年以来,研究人员通过观察并模拟自然界生物的捕食行为或物理现象,提出了众多的启发式优化算法.与传统的优化算法相比,因其精度高、不受求解

对象约束等特点而广泛地被应用于各种工业领域.樽海鞘群算法(Salp Swarm Algorithms, SSA)作为一种新的启发式优化算法,2017年由Mirjalili教授等<sup>[1]</sup>人提出.与遗传算法、萤火虫优化算法、粒子群算法和引力搜索

收稿日期:2020-06-18;修回日期:2021-01-23;责任编辑:覃怀银

基金项目:国家自然科学基金(No.61863029, No.61364018, No.61763037);内蒙古科技成果转化项目(No.CGZH2018129);内蒙古自治区科技计划项目申报书(关键技术攻关计划项目);内蒙古自然科学基金(No.2020MS06020, No.2016JQ07)

算法等 6 种优化算法相比,SSA 算法具有调节参数少和准确性高等优势,但在寻优迭代过程中依然存在容易陷入局部最优、收敛速度慢的问题. 针对上述问题,学者们提出了一些全局搜索策略应于 SSA 算法之外的其他智群算法:例如,褚鼎立等<sup>[2]</sup>通过引入模拟退火增强鲸鱼优化算法的全局探索能力,防止陷入局部最优解;滕志军等<sup>[3]</sup>提出一种基于粒子群的混合灰狼优化的改进算法,将个体自身经历过的最优值与种群最优值相结合来更新灰狼个体的位置信息,提高了局部和全局探索能力;另外,还将改进种群策略应用于传统的 SSA 算法或其他智群算法: Mohammad Tubishat 等<sup>[4]</sup>引入反学习因子初始化樽海鞘种群,增强个体的多样性;张达敏等<sup>[5]</sup>提出基于疯狂自适应的樽海鞘群算法,通过引入疯狂算子增强种群的多样性,引入自适应惯性权重增强全局探索能力;方群等<sup>[6]</sup>在粒子群算法速度更新公式中,采用惯性权值“阶梯式”调整策略提高收敛速度;王依柔等<sup>[7]</sup>提出基于自适应扰动的疯狂蝴蝶算法,在最优解位置引入扰动策略,避免算法陷入局部最优;龙文等<sup>[8]</sup>提出一种改进的灰狼算法,对灰狼最优个体执行透视镜成像学习策略来增强种群的多样性,提高了寻优的效率.

针对传统 SSA 算法的局限性,提出一种基于阶梯式 Tent 混沌的模拟退火樽海鞘群算法 (Step-by-step Tent Chaos Simulated Annealing Salp Swarm Algorithm, STSA-SSA). 算法前期,采用 Tent 混沌映射初始化种群,增强种群的均匀性和遍历性,提高前期的收敛速度;在追随者更新公式中引入使用惯性权值“阶梯式”调整策略,不同的优化阶段取不同的惯性权值,更好的兼顾全局探索能力和局部开发能力;但仅靠对于惯性权值的优化调整,不能完全解决樽海鞘群算法容易陷入局部最优的问题,所以在算法的后期,引入模拟退火中的 Boltzmann 选择机制,以一定的概率接受适应度值较差的解来提升跳出局部最优的能力. 通过经典函数测试和磁导航自动导引车模糊控制器参数寻优来验证 STSA-SSA 算法的有效性.

## 2 樽海鞘群算法

Mirjalili 等人根据樽海鞘特殊的聚集行为提出一种樽海鞘链模型,模型中将樽海鞘种群分为两组:领导者和追随者,领导者是链中最前面的个体,其余为追随者. 在 SSA 算法中,樽海鞘链中的个体的位置矢量  $F$  定义在  $N$  维搜索空间中,是由  $N$  个维度为  $d$  的个体组成. 迭代过程中,一种被称为  $F$  的食物源是樽海鞘链移动的目标位置,领导者通过更新自己的位置逐渐向食物源靠近,其更新公式为:

$$x_j^1 = \begin{cases} F_j + c_1 \left( (ub_j - lb_j) c_2 + lb_j \right), & c_3 \geq 0.5 \\ F_j - c_1 \left( (ub_j - lb_j) c_2 + lb_j \right), & c_3 < 0.5 \end{cases} \quad (1)$$

式中,  $x_j^1$  为樽海鞘链中第一只个体在  $j$  维空间的位置,  $F_j$  为  $j$  维空间中食物源的位置,  $ub_j$  为  $j$  维空间中搜索的上限,  $lb_j$  为  $j$  维空间中搜索的下限,  $c_2$  与  $c_3$  是区间内的随机数,它们指示第  $j$  维中的下一个位置是否应该朝向正无穷大或负无穷大以及步长,  $c_1$  为 SSA 算法重要的自适应调整参数,其定义公式如下:

$$c_1 = 2e^{-\left(\frac{L_{\max}}{L_{\max}}\right)^2} \quad (2)$$

式中,  $L$  是当前迭代次数,  $L_{\max}$  是定义的最大迭代次数,随着迭代次数的变化,  $c_1$  可以动态的权衡 SSA 算法的全局探索能力和局部开发能力.

使用以下公式来更新追随者位置

$$x_j^i = \frac{1}{2} at^2 + v_0 t \quad (3)$$

式中,  $x_j^i$  为第  $i$  只追溯者在  $j$  维空间中的位置,其中  $i \geq 2$ ,  $v_0$  为初始速度,  $t$  为时间,  $a$  为加速度,  $x_j^{i-1}$  为樽海鞘链中第  $i-1$  只个体在  $j$  维空间的位置,式(3)可转化为:

$$x_j^i = \frac{1}{2} (x_j^i + x_j^{i-1}) \quad (4)$$

## 3 基于阶梯式 Tent 混沌的模拟退火樽海鞘群算法

### 3.1 Tent 混沌映射序列

混沌映射具有随机性、遍历性和有序性等特点,很多学者将混沌映射引入到人工蜂群算法<sup>[9]</sup>、灰狼算法<sup>[10]</sup>等启发式优化算法,利用其求解经典函数的最优解,可以增加种群的多样性,加快算法前期的收敛速度. 因此,本文选用 Tent 混沌产生的序列初始化种群来改进 SSA 算法<sup>[11]</sup>,表达式如下:

$$x_{i+1} = \begin{cases} 2x_i, & 0 \leq x_i \leq 0.5 \\ 2(1-x_i), & 0.5 \leq x_i \leq 1 \end{cases} \quad (5)$$

式(5)经过伯努利移位变换后的表达式为:

$$x_{i+1} = (2x_i) \bmod 1 \quad (6)$$

根据以上 Tent 映射的表达式,在可行域内产生 Tent 映射序列的步骤如下:

**步骤 1** 在  $(0, 1)$  区间之间随机产生  $x_0$  (避免  $x_0$  在小周期内  $\{0.2, 0.4, 0.6, 0.8\}$ ),  $Z(1) = x_0, i = j = 1$ ;

**步骤 2** 根据式(6)迭代,得  $x(i+1)$ , 继续  $i = i+1$ ;

**步骤 3** 若  $x(i) = \{0, 0.25, 0.5, 0.75\}$ , 或  $x(i) = x(i-k), k = \{0, 1, 2, 3, 4\}$ , 按  $x(i) = z(j+1) = z(j) + \varepsilon$  更新初值,  $\varepsilon$  为随机数,  $j = j+1$ , 否则返回步骤 2;

**步骤 4** 直到最大迭代数,程序结束,生成初始化序列,否则返回步骤 2.

### 3.2 “阶梯式”惯性权值

针对 SSA 算法容易陷入局部最优的缺陷,本文借鉴文献[6]提出的“阶梯式”惯性权值调整策略,惯性权值在算法前期取固定最大值用于提高算法的全局探索能力,在算法后期取固定最小值用来提高算法的局部开发能力,其表达式为:

$$x_j^i = \frac{1}{2}(x_j^i + \omega(t)x_j^{i-1}) \quad (7)$$

$$\omega(t) = \begin{cases} \omega_{\max}, & t/T_{\max} \leq \lambda \\ \omega_{\min}, & t/T_{\max} > \lambda \end{cases} \quad (8)$$

式中,  $\omega_{\max}$  为最大惯性权值,  $\omega_{\min}$  为到最小惯性权值,  $t$  为当前迭代次数,  $T_{\max}$  为最大迭代次数.  $\lambda \in (0, 1)$  表示迭代前期时间占整个周期的比例:取  $\lambda > 0.5$  时,算法前期分配了更多的时间比例进行全局搜索,进而扩大寻优范围;取  $\lambda < 0.5$  时,算法后期占用更多的时间进行局部开发,进而寻求更高的收敛精度. 这样,通过调整  $\lambda$ ,可有效地兼顾算法的全局探索能力和局部开发能力. 通过多次尝试  $(0, 1)$  范围内不同的  $\lambda$  的取值,观察对比寻优效果,最终确定  $\lambda$  的取值.

### 3.3 模拟退火算法

模拟退火算法 (Simulated Annealing, SA) 由 Metropolis 在 1953 年提出<sup>[12]</sup>,其特点为在一定的概率情况下保留劣质群体,增加种群的多样性,在一定程度上提高了跳出局部最优的能力. 在全局寻优方面,本文采取模拟退火算法与 SSA 算法结合的方式,在 SSA 算法求出食物源位置之后再行模拟退火处理,对食物源位置进行调整,其接受劣质群体概率的公式<sup>[13]</sup>为:

$$P = \begin{cases} 1, & f_{\text{new}}(X_j) < f(X_j) \\ \exp\left(-\left(f_{\text{new}}(X_j) - f(X_j)\right)/T\right), & f_{\text{new}}(X_j) \geq f(X_j) \end{cases} \quad (9)$$

式中,  $f_{\text{new}}(X_j)$  是模拟退火阶段产生的新种群中第  $j$  个樽海鞘个体的适应度值,  $T$  表示第  $j$  次迭代温度,  $f_{\text{new}}(X_j)$  的值越小,表示适应度值越好,当  $f_{\text{new}}(X_j) < f_{\text{new}}(X_j)$  时,完全接受  $f_{\text{new}}(X_j)$  (替换概率  $P$  为 1), 否则以概率  $P$  接受劣质解  $X_j$ . 引入模拟退火算法后的 SSA 算法,在食物源位置周围范围内初始化新的种群,以一定概率下接受劣质解来更新食物源位置,通过更新式(1)、式(7)增加了领导者和追随者个体的多样性.

### 3.4 STST-SSA 算法实现步骤

改进的 SSA 算法 (STSA-SSA) 具体实现步骤如下:

(1) 设置算法参数并初始化种群:种群个数  $N$ , 最大迭代次数  $T_{\max}$ , 可调参数  $\lambda$ , 惯性权值  $\omega_{\max}$  和  $\omega_{\min}$ ; 引入 Tent 混沌映射序列初始化种群,生成一个  $N \times d$  的维矩阵.

(2) 计算适应度值:计算每个樽海鞘个体的适应度值,选取适应度值最小的个体位置作为食物源位置.

(3) 领导者和追随者位置更新:选取一半的樽海鞘个体根据式(1)更新领导者位置,可调参数  $\lambda$  的变化,选定不同的惯性权值,根据式(7)更新追随者位置.

(4) 食物源位置更新:根据更新后的樽海鞘群个体计算适应度值,若小于原食物源位置的适应度值,则更新食物源位置.

(5) 引入模拟退火算法:在原食物源位置周围随机产生一个新的种群,计算新种群个体的适应度值. 若新种群中樽海鞘个体适应度值小于食物源适应度值,则更新食物源位置,否则以式(9)中的概率  $P$  接受新种群中樽海鞘个体的位置,进行退温操作  $t = t \times 0.99$ ,记录下整个迭代过程中适应度值最小的食物源位置及其适应度值.

(6) 若当前迭代次数小于最大迭代次数,重复步骤 3~5 的迭代过程,直到达到设定的精度要求或最大迭代次数,输出最优个体位置及其适应度值.

## 4 仿真结果与分析

### 4.1 基准测试函数及测试环境

为验证提出的 STSA-SSA 算法的性能,本文选取单峰、多峰合计 14 个 Benchmark 基准函数进行算法寻优测试<sup>[14]</sup>,根据测试函数不同的性能来验证算法寻优能力,单峰、多峰函数见表 1、表 2. 单峰函数主要用来检验算法的收敛速度,多峰函数主要用来检验算法的全局探索能力和跳出局部最优解的能力.

测试环境与参数设置:测试环境采用微软 64 位操作系统,所有的算法性能检验均采用 MATLAB 编程来实现;种群规模设置为 100,最大迭代次数设置为 1000,重复运算 50 次,取最优收敛值、平均值和标准差.

表 1 单峰测试函数

函数表达式	维度	取值范围	全局最小值
$f_1(x) = \sum_{i=1}^n x_i^2$	30	[-100, 100]	0
$f_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	10	[-10, 10]	0
$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j\right)$	30	[-100, 100]	0
$f_4(x) = \max\{ x_i , 1 \leq i \leq n\}$	30	[-100, 100]	0
$f_5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_{i+1} - 1)^2\right]$	30	[-30, 30]	0
$f_6(x) = \sum_{i=1}^n \left([x_i + 0.5]\right)^2$	30	[-100, 100]	0
$f_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$	30	[-1.28, 1.28]	0

表 2 多峰测试函数

函数表达式	维度	取值范围	全局最小值
$f_8(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-100, 100]	0
$f_9(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	10	[-10, 10]	0
$f_{10}(x) = \frac{1}{4000}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-100, 100]	0
$f_{11}(x) = \frac{\pi}{n}\left[10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + \sin^2(\pi y_{i+1})] + (y_{n-1})^2\right] + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$	30	[-100, 100]	0
$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$			
$f_{12}(x) = 0.1\left[\sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)]\right] + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	[-30, 30]	0
$f_{13}(x) = \sum_{i=1}^n [a_i - x_i(b_1^2 + b_1 x_2)] / (b_1^2 + b_1 x_2 + x_4)$	4	[-5, 5]	0.0003
$f_{14}(x) = -\sum_{i=1}^4 q_i \exp\left(-\sum_{j=1}^3 q_j (x_i - p_j)^2\right)$	3	[1, 3]	-3.86

4.2 基准测试函数及测试环境

4.2.1 λ 参数多个取值的寻优性能对比分析

STSA-SSA 算法中, λ 决定了不同惯性权值在整个迭代周期中所占有的时间比例, 以调整算法的全局探索能力和局部开发能力. 本文测试了不同的 λ (0.1, 0.2, 0.4, 0.5, 0.6, 0.7, 0.9 等 7 种取值) 情况下,  $f_1 \sim f_{14}$  测试函数寻优结果, 其最优收敛值、平均值和标准差如表 3 所示.

表 3 中, 最佳值、平均值反映了算法的收敛精度和寻优能力, 方差反映算法的稳定性. 由表 3 结果可知: λ 取 0.4 时, STSA-SSA 算法对于  $f_1 \sim f_4, f_6, f_7, f_{11}, f_{12}, f_{14}$  达到了最好的寻优效果; 对于函数  $f_8, f_9$  和  $f_{10}$ , 7 种取值下获得了相同的寻优结果; 对于函数  $f_8, f_9$ , λ 取 0.4 时, 虽然寻优稳定性略差, 但仍可达到全局最小值. 综合来看, 基于 λ 取值为 0.4 的 STSA-SSA 算法对于大多数测试函数而言, 相比其他取值, 具有显著的优势. 同时, 考虑到原樽海鞘群算法的后期收敛精度不高, 为提高这一性能, 最终选取 λ = 0.4.

4.2.2 与其他智群算法的寻优性能对比分析

将 STSA-SSA 算法与传统 SSA 算法、蚁狮优化算法 (ALO)<sup>[15]</sup>、粒子群优化算法 (PSO)<sup>[16]</sup>、飞蛾扑火优化算法 (MFO)<sup>[17]</sup>、蜻蜓优化算法 (DA)<sup>[18]</sup> 和多神经优化算法

(MVO)<sup>[19]</sup> 进行函数测试比较, 仿真结果见表 4. 对表 4 中的仿真结果进行无重复双因素分析和 Friedman 秩和验证<sup>[2]</sup>, 其显著性水平值设定为 5%, 结果见表 5 和表 6.

测试结果如表 4 所示, 对于单峰测试函数  $f_1 \sim f_7$ , 无论是收敛精度还是稳定性方面, STSA-SSA 算法在函数  $f_1, f_2, f_5, f_7$  计算性能方面均优于其他算法. 相比较于 DA 算法, STSA-SSA 在函数  $f_3, f_4$  计算中其收敛精度并未达到最小值 0, 但 DA 算法函数求解的均值和方差均大于 STSA-SSA 算法, 证明 STSA-SSA 算法的稳定较高; 相比较于 MFO 算法, 除了  $f_6$  函数的收敛精度和稳定性略差, 其他函数的寻优能力均优于 MFO 算法. 对于多峰函数  $f_8 \sim f_{14}$ , STSA-SSA 算法在函数  $f_8, f_9, f_{10}, f_{13}, f_{14}$  计算中收敛精度均达到或最接近全局最小值, 相比较其他算法求解精度和稳定性有较高的提升. 相比较于 PSO、MFO 算法, 函数  $f_{11}, f_{12}$  计算中虽然其收敛精度略差, 但求解稳定性方面明显优于这两种智能算法. 相比较于 SSA 算法, 无论是单峰函数或者多峰函数的求解, 改进后的 STSA-SSA 算法在收敛精度、寻优能力和稳定性方面有了明显的优势, 进一步验证了其能全局探索能力和跳出局部最优解的能力的提升, 具有更好的寻优效果和鲁棒性.

对仿真结果进行无重复双因素分析如表 5 所示, 由

表3 不同λ下,STSA-SSA算法对f<sub>1</sub>~f<sub>14</sub>测试函数的寻优结果

函数值	评价指标	λ = 0.1	λ = 0.2	λ = 0.4	λ = 0.5	λ = 0.6	λ = 0.7	λ = 0.9
f <sub>1</sub>	最佳值	4.31E-49	2.67E-49	<b>1.53E-57</b>	3.96E-47	3.71E-49	3.92E-49	3.51E-49
	均值	5.48E-49	5.69E-49	1.72E-52	5.56E-49	5.17E-49	5.95E-49	6.16E-49
	方差	3.27E-98	2.54E-98	3.94E-103	1.01E-98	2.25E-38	1.59E-98	3.30E-98
f <sub>2</sub>	最佳值	2.79E-25	3.04E-25	<b>4.57E-26</b>	2.07E-25	2.60E-25	2.78E-25	2.72E-25
	均值	3.42E-25	3.34E-25	1.02E-25	3.11E-25	3.30E-25	3.26E-25	3.22E-25
	方差	1.38E-51	9.39E-52	6.22E-52	2.63E-51	1.84E-51	9.93E-52	1.09E-51
f <sub>3</sub>	最佳值	8.72E-50	7.74E-50	<b>5.72E-50</b>	6.88E-50	1.64E-49	7.68E-50	5.91E-50
	均值	3.42E-49	5.26E-49	5.20E-49	4.67E-49	1.04E-48	7.63E-49	6.49E-49
	方差	4.34E-98	3.62E-97	2.35E-97	2.79E-97	7.52E-97	3.66E-97	8.41E-97
f <sub>4</sub>	最佳值	1.23E-25	1.53E-25	<b>1.03E-25</b>	1.16E-25	1.36E-25	1.80E-25	1.49E-25
	均值	2.04E-25	2.35E-25	2.40E-25	2.02E-25	2.09E-25	2.55E-25	2.51E-25
	方差	3.59E-51	6.30E-51	3.86E-51	2.77E-51	1.64E-51	1.53E-51	6.30E-51
f <sub>5</sub>	最佳值	8.26E-09	1.28E-08	5.05E-09	9.21E-09	1.33E-08	1.03E-09	1.01E-09
	均值	1.74E-08	1.86E-08	1.68E-08	1.72E-08	1.78E-08	1.45E-08	1.45E-08
	方差	2.14E-17	2.86E-17	1.99E-17	1.47E-17	8.88E-18	1.43E-17	8.09E-18
f <sub>6</sub>	最佳值	3.74E-10	2.71E-10	<b>2.28E-10</b>	2.99E-10	4.16E-10	4.23E-10	3.90E-10
	均值	5.84E-10	6.31E-10	5.78E-10	5.16E-10	5.94E-10	6.14E-10	6.21E-10
	方差	3.46E-20	3.79E-20	2.26E-20	1.99E-20	1.25E-20	1.12E-20	2.25E-20
f <sub>7</sub>	最佳值	6.18E-07	3.11E-07	<b>2.40E-07</b>	2.52E-06	1.82E-06	3.70E-06	3.66E-06
	均值	9.95E-06	1.38E-05	1.67E-05	1.81E-05	1.66E-05	2.16E-05	2.43E-05
	方差	1.03E-10	1.40E-10	3.57E-10	5.05E-10	2.48E-10	7.08E-10	8.99E-10
f <sub>8</sub>	最佳值	0	0	0	0	0	0	0
	均值	0	0	0	0	0	0	0
	方差	0	0	0	0	0	0	0
f <sub>9</sub>	最佳值	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	均值	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	方差	0	0	0	0	0	0	0
f <sub>10</sub>	最佳值	0	0	0	0	0	0	0
	均值	0	0	0	0	0	0	0
	方差	0	0	0	0	0	0	0
f <sub>11</sub>	最佳值	2.87E-12	3.02E-12	<b>1.73E-12</b>	3.40E-12	2.26E-12	2.83E-12	3.23E-12
	均值	6.41E-12	5.90E-12	5.82E-12	5.79E-12	5.56E-12	5.80E-12	6.82E-12
	方差	4.21E-24	2.62E-24	4.58E-24	3.16E-24	3.82E-24	3.60E-24	4.68E-24
f <sub>12</sub>	最佳值	1.77E-11	1.04E-11	<b>8.72E-12</b>	1.62E-11	1.67E-11	1.89E-11	1.42E-11
	均值	2.53E-11	2.30E-11	2.30E-11	2.58E-11	2.29E-11	2.30E-11	2.51E-11
	方差	2.48E-23	3.03E-23	3.86E-23	7.39E-23	1.94E-23	6.23E-23	4.78E-23
f <sub>13</sub>	最佳值	0.00030	0.00030	0.00030	0.00030	0.00030	0.00030	0.00030
	均值	0.00040	0.00030	0.00040	0.00059	0.00039	0.00031	0.00040
	方差	8.29E-08	2.56E-11	1.09E-07	1.90E-07	8.38E-08	5.02E-19	8.38E-08
f <sub>14</sub>	最佳值	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628
	均值	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628
	方差	9.82E-29	2.25E-28	1.75E-28	1.64E-28	1.86E-28	6.21E-29	1.09E-28

于  $P\text{-value} > 0.05$ , 且  $F < F_{\text{crit}}$ , 所以7种算法在0.05显著性水平值下无明显差异. Friedman 秩和验证中STSA-SSA算法的秩均值最小, 进一步验证其算法性能

优于其他智能算法.

7种算法在单峰和多峰函数计算中收敛精度存在明显的差异, 为更加直观的呈现不同的算法在函数求

表 4 仿真结果

函数值	评价指标	STSA-SSA	SSA	MVO	ALO	DA	PSO	MFO
$f_1$	最佳值	<b>1.53E-57</b>	1.95E-10	0.00034	2.7E-10	0.1362	6.12E-18	1.61E-06
	均值	<b>1.72E-52</b>	1.83E-10	1.16E-03	8.41E-10	4.494	7.44E-12	600
	方差	3.94E-103	1.87E-13	2.65E-07	8.30E-20	119.35	8.36E-28	5.7E-07
$f_2$	最佳值	<b>4.57E-26</b>	3.91E-05	0.1089	0.00089	0.0042	1.49E-09	5.64E-05
	均值	<b>1.02E-25</b>	0.1687	0.1931	22.237	10.223	1.62E-07	32.000
	方差	6.22E-52	0.0884	0.0020	1220.27	31.225	8.79E-14	379.59
$f_3$	最佳值	<b>5.72E-50</b>	0.0191	3.0589	4.0574	<b>0</b>	0.8018	1.55E-15
	均值	<b>5.20E-49</b>	1.0265	7.0964	34.873	4.2E3	2.516	1.67E-11
	方差	2.35E-97	1.8767	8.9943	622.13	2.6E7	1.2863	3.61E-21
$f_4$	最佳值	<b>1.03E-25</b>	0.0069	0.01662	0.667	<b>0</b>	0.08	1.39E-12
	均值	2.40E-25	0.0049	0.376	4.9348	6.5196	0.2041	4.71E-10
	方差	<b>3.86E-51</b>	0.7660	0.0186	6.0292	27.112	0.006	4.72E-19
$f_5$	最佳值	<b>5.05E-09</b>	0.055	26.313	13.312	28.734	0.571	0.0375
	均值	1.68E-08	27.679	215.81	136.1	7.01E03	41.76	101.01
	方差	1.99E-17	6585.9	1.47E7	3.54E5	2.26E8	1049.2	1.91E5
$f_6$	最佳值	2.28E-10	4.29E-09	0.0553	3.24E-08	1.2529	69.52	<b>0</b>
	均值	5.78E-10	7.23E-09	0.1198	7.86E-08	232.26	110.62	2.42E-32
	方差	2.26E-20	2.24E-18	0.00148	2.21E-15	1.44E5	215.46	<b>3.82E-63</b>
$f_7$	最佳值	<b>2.40E-07</b>	0.0102	0.00024	0.0102	3.59E-05	0.0117	0.0004
	均值	1.67E-05	0.0263	0.0069	0.0226	0.0654	0.03222	0.0017
	方差	3.57E-10	9.82E-05	5.40E-06	5.06E-05	0.01481	0.00001	6.02E-07
$f_8$	最佳值	<b>0</b>	11.939	46.833	30.843	0.0023	17.295	0.9949
	均值	<b>0</b>	35.042	100.16	64.195	51.005	32.312	14.155
	方差	<b>0</b>	166.80	584.17	433.07	2049.7	84.286	101.836
$f_9$	最佳值	<b>8.88E-16</b>	1.70E-05	0.0667	4.7E-05	0.0011	9.19E-10	4.44E-15
	均值	8.88E-16	1.1542	0.4199	1.8083	4.5597	3.30E-09	4.44E-15
	方差	0	0.7409	0.2917	0.4163	7.1093	2.78E-15	0
$f_{10}$	最佳值	<b>0</b>	1.48E-08	0.1678	6.06E-06	<b>0</b>	<b>0</b>	0.0394
	均值	<b>0</b>	0.010	0.2863	0.0104	5.4385	0.0097	0.1622
	方差	<b>0</b>	0.0001	0.0050	0.0001	168.64	0.0001	0.0119
$f_{11}$	最佳值	1.73E-12	2.21E-11	0.0004	0.675	0.1465	<b>2.04E-20</b>	<b>4.71E-32</b>
	均值	<b>5.82E-12</b>	2.3169	0.7215	6.5158	26.296	<b>0.002</b>	<b>0.031</b>
	方差	<b>4.58E-24</b>	3.603	0.89	12.627	2.27E04	<b>0.0002</b>	<b>0.0167</b>
$f_{12}$	最佳值	8.72E-12	2.32E-10	0.005	2.12E-08	2.0911	<b>5.63E-18</b>	<b>1.35E-32</b>
	均值	<b>2.3E-11</b>	0.0055	0.022	0.0082	217.23	<b>0.00088</b>	<b>0.0015</b>
	方差	<b>3.86E-23</b>	9.42E-05	0.0002	0.0002	9.67E05	<b>9.07E-06</b>	<b>1.48E-05</b>
$f_{13}$	最佳值	<b>0.00030</b>	0.00030	0.00030	0.00030	0.00031	0.00030	0.00032
	均值	0.00040	0.00070	0.00042	0.0019	0.0014	0.00071	0.0009
	方差	1.09E-07	1.03E-07	5.81E-05	2.22E-05	2.66E-07	2.77E-08	1.17E-07
$f_{14}$	最佳值	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628
	均值	<b>-3.8628</b>	-3.6609	-3.8628	-3.8628	-3.8623	-3.8524	3.8628
	方差	1.75E-28	8.32E-29	3.76E-14	4.84E-29	1.85E-06	5.35E-05	9.86E-30

表 5 无重复双因素分析结果

F	P-value	Fcrit
1.1082	0.3650	1.8478

解过程中的收敛情况,图 1 给出 9 个基准测试函数的收敛图,图例中横坐标为 1000 迭代次数,纵坐标为收敛适应度值.从单峰函数 $f_1$ 、 $f_2$ 、 $f_3$ 、 $f_4$ 、 $f_5$ 收敛图可知,

表 6 Friedman 秩和验证结果

算法	秩均值
STSA-SSA	1.14
SSA	3.71
MVO	4.57
ALO	5.07
DA	6.36
PSO	3.43
MFO	3.71

相比于其他算法, STSA-SSA 算法在迭代前期收敛速度明显加快, 这是由于引入 Tent 混沌映射序列初始化种群, 增加种群多样性进而寻加快优的效率, 随着迭代次数的增加, 各种算法出现了不同程度上的停滞, 陷入局部最优. 在迭代中后期, STSA-SSA 算法引入惯性

权值“阶梯式”调整策略和模拟退火算法, 跳出局部最优并持续收敛到迭代后期, 其他算法除了 MFO 和 PSO 算法有小幅度的持续收敛外, 其余后期均处于不同程度上的停滞状态. 多峰函数  $f_8$ 、 $f_9$ 、 $f_{10}$ 、 $f_{12}$  收敛图已看出, STSA-SSA 算法在函数  $f_8$ 、 $f_9$ 、 $f_{10}$  迭代前期依然具有明显的收敛速度优势, 在迭代后期跳出局部最优, 继续进行寻优, 在函数  $f_8$ 、 $f_{10}$  上收敛精度达到函数理论最小值 0. 在函数  $f_{12}$  寻优过程中, 虽最后的收敛精度不如 MFO 和 PSO 算法精确, 但整体的寻优能力相对于 SSA 算法有明显的提升. 总之, 对于绝大部分的单峰和多峰函数, STSA-SSA 算法在迭代前期能够快速收敛, 中后期惯性权值“阶梯式”调整策略和模拟退火算法的结合增强跳出局部最优的能力并持续全局探索, 达到或接近最高的收敛精度.

#### 4.2.3 与其他改进算法的寻优性能对比分析

为进一步验证 STSA-SSA 算法的优化性能, 将 STSA-SSA 算法与阶梯式 Tent 混沌樽海鞘群算法 (Step-by-step Tent chaos Salp Swarm Algorithm, ST-SSA)、

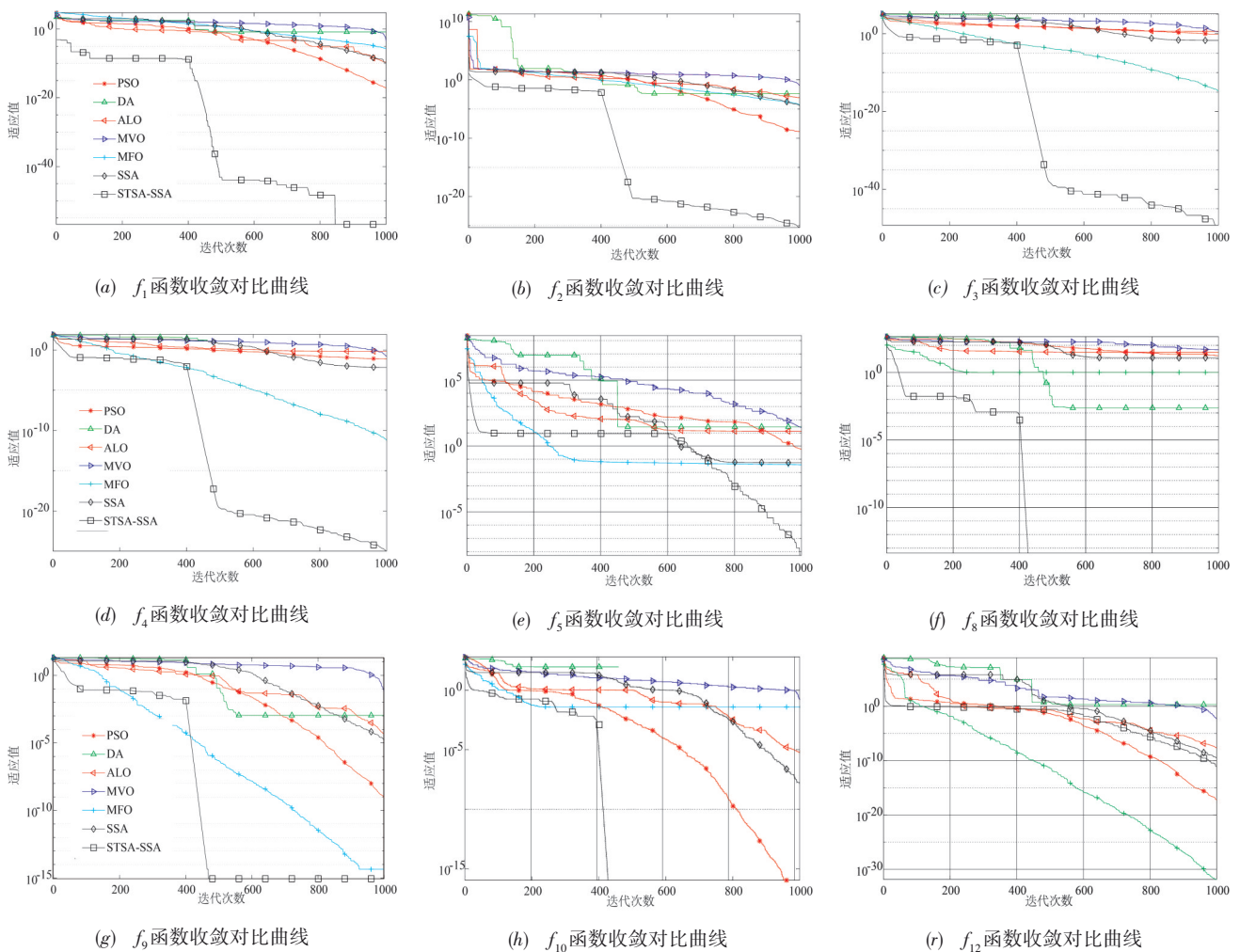


图 1 测试函数收敛曲线

Tent混沌模拟退火樽海鞘群算法(Simulated Annealing Salp Swarm Algorithm, SA-SSA)、Tent混沌自适应权重樽海鞘群算法(Adaptive Weight Salp Swarm Algorithm, AW-SSA)进行寻优性能对比分析,在相同测试环境和参数设置的情况下,各改进算法的测试函数最优收敛值、平均值和标准差见表7,无重复双因素分析和Fried-

表7 仿真结果

函数值	评价指标	STSA-SSA	SSA	SA-SSA	ST-SSA	AW-SSA
$f_1$	最佳值	<b>1.53E-57</b>	1.95E-10	4.47E-09	4.29E-49	4.47E-10
	均值	<b>1.72E-52</b>	1.83E-10	5.65E-09	5.31E-49	6.09E-10
	方差	<b>3.94E-103</b>	1.87E-13	7.76E-19	2.22E-98	2.25E-20
$f_2$	最佳值	<b>4.57E-26</b>	3.91E-05	0.0096	2.48E-25	9.45E-06
	均值	<b>1.02E-25</b>	0.1687	0.1148	3.26E-25	1.09E-05
	方差	<b>6.22E-52</b>	0.0884	0.0026	3.02E-51	9.13E-13
$f_3$	最佳值	<b>5.72E-50</b>	0.0191	1.24E-10	8.72E-50	3.97E-11
	均值	<b>5.20E-49</b>	1.0265	2.42E-10	5.53E-49	1.85E-10
	方差	2.35E-97	1.8767	5.08E-21	2.18E-97	1.71E-20
$f_4$	最佳值	<b>1.03E-25</b>	0.0069	3.91E-06	1.69E-25	4.05E-06
	均值	2.40E-25	0.0049	8.10E-06	2.28E-25	7.26E-06
	方差	3.86E-51	0.7660	6.63E-12	1.97E-57	2.63E-12
$f_5$	最佳值	5.05E-09	0.055	<b>4.86E-09</b>	6.6823	8.8579
	均值	1.68E-08	27.679	<b>1.09E-08</b>	7.0606	8.8887
	方差	1.99E-17	6585.9	<b>1.83E-17</b>	0.0530	0.0002
$f_6$	最佳值	<b>2.28E-10</b>	4.29E-09	2.57E-10	5.03E-10	0.1714
	均值	5.78E-10	7.23E-09	4.76E-10	6.03E-10	0.2234
	方差	2.26E-20	2.24E-18	2.51E-20	6.73E-21	0.0015
$f_7$	最佳值2.40E-07	2.40E-07	0.0102	<b>8.72E-08</b>	1.78E-06	3.61E-07
	均值	<b>1.67E-05</b>	0.0263	<b>1.69E-05</b>	2.07E-05	1.84E-05
	方差	3.57E-10	9.82E-05	<b>6.06E-10</b>	3.29E-10	8.59E-10
$f_8$	最佳值	<b>0</b>	11.939	6.28E-11	<b>0</b>	2.60E-11
	均值	<b>0</b>	35.042	1.19E-10	<b>0</b>	9.2411
	方差	<b>0</b>	166.80	1.91E-21	<b>0</b>	154.09
$f_9$	最佳值	<b>8.88E-16</b>	1.70E-05	3.94E-06	<b>8.88E-16</b>	3.09E-06
	均值	8.88E-16	1.1542	6.31E-06	<b>8.88E-16</b>	5.13E-05
	方差	0	0.7409	1.61E-12	<b>0</b>	4.58E-13
$f_{10}$	最佳值	<b>0</b>	1.48E-08	3.87E-10	<b>0</b>	1.37E-10
	均值	<b>0</b>	0.010	1.08E-09	<b>0</b>	0.2167
	方差	<b>0</b>	0.0001	1.17E-19	<b>0</b>	0.0859
$f_{11}$	最佳值	<b>1.73E-12</b>	2.21E-11	3.21E-12	3.44E-12	0.0073
	均值	<b>5.82E-12</b>	2.3169	5.07E-12	0.0311	0.0357
	方差	<b>4.58E-24</b>	3.603	1.86E-24	0.0097	0.0002
$f_{12}$	最佳值	8.72E-12	2.32E-10	<b>4.62E-12</b>	9.46E-12	0.0066
	均值	2.3E-11	0.0055	<b>1.13E-11</b>	2.98E-11	0.1368
	方差	3.86E-23	9.42E-05	<b>5.38E-24</b>	1.84E-22	0.0028
$f_{13}$	最佳值	<b>0.00030</b>	0.00030	0.00032	0.00031	0.00032
	均值	<b>0.00040</b>	0.00070	0.00076	0.00049	0.00048
	方差	1.09E-07	1.03E-07	1.22E-07	1.48E-07	1.95E-07
$f_{14}$	最佳值	-3.8628	-3.8628	-3.8628	-3.8628	-3.8617
	均值	-3.8628	-3.6609	-3.8628	-3.8628	-3.8576
	方差	1.75E-28	8.32E-29	1.09E-28	1.27E-28	1.28E-05

man 秩和验证结果见表 8 和表 9.

表 8 无重复双因素分析结果

F	P-value	Fcrit
0.6030	0.6621	2.5498

表 9 Friedman 秩和验证结果

算法	秩均值
STSA-SSA	1.14
SSA	3.71
SA-SSA	3.00
ST-SSA	2.50
AW-SSA	4.11

从表 7 测试结果可知,SA-SSA 算法和 ST-SSA 算法经过 1000 次迭代后均能接近全局最小值,相比较于 SSA 算法,除  $f_1, f_2, f_{13}$  函数的收敛精度和稳定性略差,其他函数的计算性能方面均优于 SSA 算法. 相比较于

SA-SSA 算法,除了函数  $f_5, f_7, f_{12}$  之外,STSA-SSA 在其他函数的计算性能方面均优于 SA-SSA 算法. 相比较于 ST-SSA 算法,函数  $f_8, f_9, f_{10}, f_{14}$  两种优化效果相持平,其余测试函数 STSA-SSA 计算性能方面要优于 ST-SSA 算法,验证了 ST-SSA 算法迭代后期引入 SA 算法提升跳出局部最优的能力. 相比较于 AW-SSA 算法,无论单峰还是多峰测试函数,ST-SSA 算法在收敛精度、寻优能力和稳定性方面均优于 AW-SSA 算法.

从图 2 中 9 个基准测试函数的收敛图可看出,相比于 SA-SSA、AW-SSA 算法,STSA-SSA 算法在迭代前期具有更快的收敛速度,迭代后期具有更高的收敛精度. STSA-SSA 算法与 ST-SSA 算法在函数  $f_3, f_4, f_8, f_9, f_{10}, f_{12}$  收敛过程中具有相似的收敛轨迹,但 STSA-SSA 算法在函数  $f_3, f_4, f_{10}$  的迭代后期具有更高的收敛精度;对于函数  $f_1, f_2, f_5$ ,其收敛速度与精度要优于 STSA-SSA 算法. 表 8 无重复双因素分析结果中, $P\text{-value} > 0.05$ ,且  $F < F_{crit}$ ,说明几种改进算法在 0.05 显著性水平值下无明显

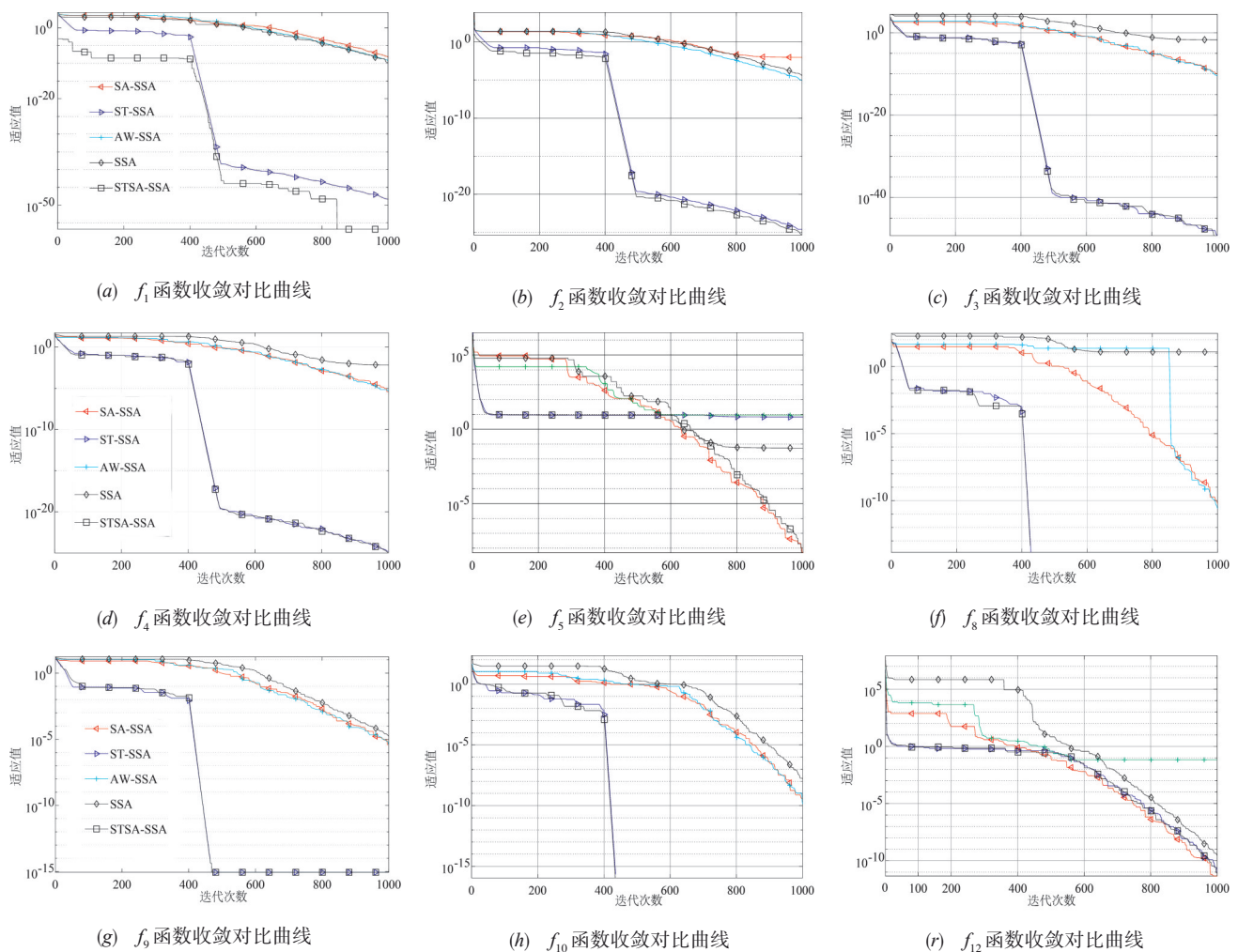


图 2 测试函数收敛曲线

显差异,表 9 中 STSA-SSA 算法的秩均值最小. 总体上, STSA-SSA 算法性能相比于其他改进算法具有明显的优越性.

#### 4.2.4 磁导航 AGV 模糊控制器参数寻优

针对磁导航 AGV 磁偏差值控制问题,由于传统磁导航 AGV 模糊控制器量化因子  $k_e$ 、 $k_{ec}$  和比例因子  $k_u$  依靠人工经验试凑调节,随意性较强. 针对这一不足,以图 3 所示某重点实验室科技成果转化项目中的磁导航 AGV 为研究对象,引入 STSA-SSA 算法对量化因子参数优化.



图 3 磁导航 AGV

对于模型依赖的控制算法应用,良好的控制性能依赖于受控对象动态模型的精确建立. 虽然模糊控制器在一定程度上对磁导航 AGV 数学模型的摄动具有鲁棒性,但对其控制效果的评估和最优控制器参数设计,仍需建立较精确的受控对象数学模型. 本文采用时域辨识法对磁导航 AGV 的自回归各态历经 (AutoRegressive eXogenous, ARX) 模型进行辨识. 对于单输入/单输出系统 (SISO), ARX 模型的 Z 变换形式为<sup>[20]</sup>:

$$A(z^{-1})y(t) = B(z^{-1})u(t-d) + q(t) \quad (10)$$

$$A(z^{-1}) = 1 + a_1z^{-1} + \dots + a_nz^{-n} \quad (11)$$

$$B(z^{-1}) = b_0 + b_1z^{-1} + \dots + b_mz^{-m} \quad (12)$$

式中,  $u(t)$  和  $y(t)$  分别表示系统的输入和输出,  $q(t)$  为白噪声,  $z^{-1}$  为滞后算子,  $d$  为延迟,  $m$  和  $n$  为阶数.

辨识实验设计步骤如下: 本实验通过对磁导航 AGV 的特性分析及实验, 为保证磁导航 AGV 在磁轨道上正常运行避免脱离轨道, 最终采用方波信号作为激励信号为系统的输入, 磁偏差值为系统的输出. 采集实验的数据波形如图 4 所示, 采样周期为 0.36s, 数据长度为 42.84s. 其中方波周期为 10.08s, 幅值为 50mm/s.

借助 MATLAB 软件中系统辨识工具箱, 通过最小二乘法辨识出 ARX 模型为:

$$A(z^{-1}) = 1 - 1.106z^{-1} - 1.1309z^{-2} + 0.2472z^{-3} \quad (13)$$

$$B(z^{-1}) = 0.01002z^{-1} + 0.001831z^{-2} \quad (14)$$

式中, 设定  $q(t)$  为标准白噪声,  $d = 0$ , 输出的翻滚角的传递函数表示为:

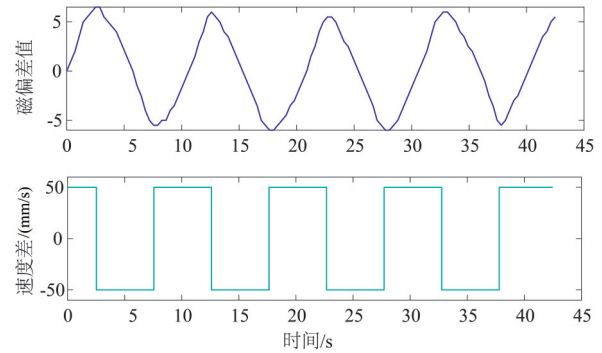


图 4 输入输出数据波形图

$$G(z) = \frac{0.01002z^2 + 0.001813z}{z^4 - 1.106z^3 - 0.1309z^2 + 0.2472z} \quad (15)$$

仿真实验: 为验证 STSA-SSA 算法的参数优化性能, 在无外界环境扰动情况下, 设置量化因子  $k_e$ 、 $k_{ec}$  和比例因子  $k_u$  参数范围分别为  $[0, 100]$ 、 $[0, 100]$  和  $[0, 50]$ , 种群大小为 80, 迭代次数上限为 300,  $\omega_{\max}$  设置为 0.9,  $\omega_{\min}$  设置为 0.4, 适应度值收敛曲线见图 5.

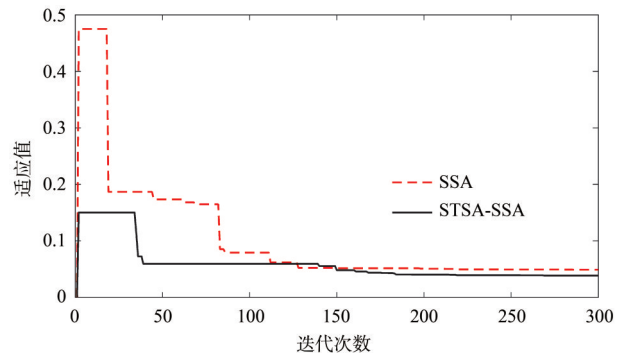


图 5 适应度值收敛曲线

在假设无内、外部扰动时, 通过对图 5、图 6 和表 10 分析可知: 除试凑法之外, 其他两种寻参方法均能实现对阶跃输入信号的无误差跟踪. 相比于试凑算法, 基于 STSA-SSA 算法增益寻优的磁导航 AGV 模糊控制器在上升时间加快 0.07s; 超调量减少 1.9%. 相比于 SSA 算

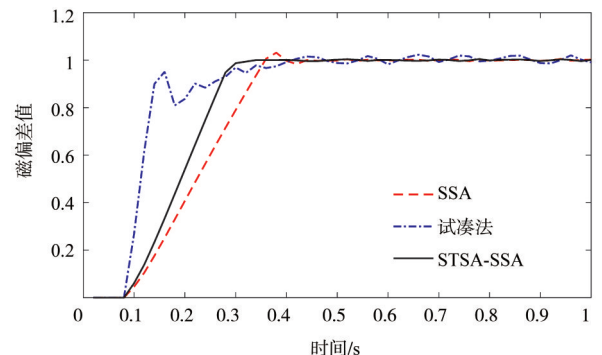


图 6 磁偏差值控制性能对比曲线

法,基于STSA-SSA算法增益寻优的磁导航AGV模糊控制器在上升时间缩短0.03s;超调量减少3.1%;调节时间缩短0.07s。STSA-SSA算法比其他两种算法收敛精度更高,控制效果也得到显著的改善。

表10 翻滚通道算法对比数据

寻参方法	上升时间/s	超调量/%	调节时间/s
试凑法	0.39	1.9	N/A
SSA	0.35	3.1	0.39
STSA-SSA	0.32	0	0.32

## 5 结论

对传统SSA算法收敛速度和易陷入局部最优的缺陷,提出一种基于Tent混沌映射、惯性权值“阶梯式”调整策略和模拟退火的樽海鞘群算法,引入Tent混沌映射初始化种群,增加种群多样性和加快迭代前期的收敛速度,通过策略动态调整惯性权值以权衡全局探索能力和局部开发能力,并且引入模拟退火算法增加跳出局部最优的概率。本文选取14个Benchmark基准函数和磁导航AGV模糊控制器参数寻优进行算法寻优测试,还使用无重复双因素分析和Friedman秩和验证对算法性能进行验证。仿真结果表明:改进后的SSA算法具有更好的全局探索能力和跳出局部最优解的能力,收敛速度加快的同时也提高了收敛精度,对于解决磁导航AGV控制器设计问题具有实际的工程应用价值。

## 参考文献

- [1] MIRJALILI S, GANDOMI A H, MIRJALILI S Z, et al. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems [J]. *Advances in Engineering Software*, 2017, 114(6): 163 – 191.
- [2] 褚鼎立, 陈红, 王旭光. 基于自适应权重和模拟退火的鲸鱼优化算法[J]. *电子学报*, 2019, 47(05): 992 – 999. CHU Ding-li, CHEN Hong, WANG Xu-guang. Whale optimization algorithm based on adaptive weight and simulated annealing [J]. *Acta Electronica Sinica*, 2019, 47(05): 992 – 999. (in Chinese)
- [3] 滕志军, 吕金玲, 郭力文, 许媛媛. 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 ZHI-JUN, LÜ JIN-LING, GUO LI-WEN, XU YUAN-YUAN. 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)
- [4] TUBISHAT M, IDRIS N, SHUIB L, et al. Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection [J]. *Expert Systems with Applications*, 2019, 145: 113 – 122.
- [5] 张达敏, 陈忠云, 辛梓芸, 张绘娟, 闫威. 基于疯狂自适应的樽海鞘群算法[J]. *控制与决策*: 2020, 35(09): 1 – 9. ZHANG Da-min, CHEN Zhong-yun, XIN Zi-yun, Zhang Hui-juan, Yan Wei. Salp swarm algorithm based on craziness and adaptive [J]. *Control and Decision*: 2020, 35(09): 1 – 9. (in Chinese)
- [6] 方群, 徐青. 基于改进粒子群算法的无人机三维航迹规划[J]. *西北工业大学学报*, 2017, 35(01): 66 – 73.
- [7] 王依柔, 张达敏, 徐航, 宋婷婷, 樊英. 基于自适应扰动的疯狂蝴蝶算法[J]. *计算机应用研究*, 2020, 37(11): 1 – 6. WANG Yi-rou, ZHANG Da-min, XU Hang, SONG Ting-ting, FAN Ying. Crazy butterfly algorithm based on adaptive perturbation [J]. *Application Research of Computers*, 2020, 37(11): 1 – 6. (in Chinese)
- [8] 龙文, 伍铁斌, 唐明珠, 徐明, 蔡绍洪. 基于透镜成像学习策略的灰狼优化算法[J]. *自动化学报*, 2020, 46(10): 148 – 164. LONG Wen, WU Tie-bin, TANG Ming-zhu, XU Ming, Cai Shao-hong. Grey wolf optimizer algorithm based on lens imaging learning strategy [J]. *Acta Automatica Sinica*, 2020, 46(10): 148 – 164. (in Chinese)
- [9] 匡芳君, 徐蔚鸿, 金忠. 自适应Tent混沌搜索的人工蜂群算法[J]. *控制理论与应用*, 2014, 31(11): 1502–1509. KUANG Fang-jun, XU Wei-hong, JIN Zhong. Artificial bee colony algorithm based on self-adaptive Tent chaos search [J]. *Control Theory & Applications*, 2014, 31(11): 1502 – 1509. (in Chinese)
- [10] 张贾奎, 崔利杰, 郭庆, 陈浩然. 基于Tent混沌序列的灰狼优化算法[J]. *微电子学与计算机*, 2018, 35(06): 11 – 16. ZHANG Jia-kui, CUI Li-jie, GUO Qing, CHEN Hao-ran. Grey wolf optimizer based on Tent chaotic sequence [J]. *Microelectronics & Computer*, 2018, 35(06): 11 – 16. (in Chinese)
- [11] 张振兴, 杨任农, 房育寰, 赵克新. 自适应Tent混沌搜索的蚁狮优化算法[J]. *哈尔滨工业大学学报*, 2018, 50(05): 152 – 159. ZHANG Zhen-xing, YANG Ren-nong, FANG Yu-huan, ZHAO Ke-xin. Ant lion optimization algorithm based on self-adaptive Tent chaos search [J]. *Journal of Harbin Institute of Technology*, 2018, 50(05): 152 – 159. (in Chinese)

Chinese)

- [12] DUPANLOU I, SCHNEIDER S, EXCOFFIER L. A simulated annealing approach to define the genetic structure of populations [J]. *Molecular Ecology*, 2002, 11(12): 2571 – 2581.
- [13] 孔芝, 李事成, 赵杰. 珊瑚礁算法的改进研究[J]. *东北大学学报(自然科学版)*, 2020, 41(02): 163 – 169.  
KONG Zhi, LI Shi-cheng, ZHAO Jie. Improved coral reef algorithm [J]. *Journal of Northeastern University (Natural Science)*, 2020, 41(02): 163 – 169. (in Chinese)
- [14] YANG X S. Firefly algorithm, stochastic test functions and design optimisation[J]. *Journal of Bio-inspired Computation*, 2010, 2(2): 78 – 84.
- [15] MIRJALILI S. The ant lion optimizer [J]. *Advances in Engineering Software*, 2015, 83(5): 80 – 98.
- [16] KENNEDY J, EBERHART R C. Particle swarm optimization[A]. *Proceedings of IEEE International Conference on Neural Networks*[C]. Perth, Australia, 1995. 1942 – 1948.
- [17] MIRJALILI S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm [J]. *Knowledge-Based Systems*, 2015, 89: 228 – 249.
- [18] MIRJALILI S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems [J]. *Neural Comput-*

ing and Applications, 2016, 27(4): 1053 – 1073.

- [19] MIRJALILI S, HATAMLOU A. Multi-verse optimizer: A nature-inspired algorithm for global optimization [J]. *Neural Computing and Applications*, 2016, 27(2): 495 – 513.
- [20] DINANA B B, VICTOR G, MARIO T G, MARIO G B, JORGE L R. An adaptive ARX model to estimate the RUL of aluminum plates based on its crack growth [J]. *Mechanical Systems and Signal Processing*, 2017, 82: 519 – 536.

#### 作者简介



周 鹏 男, 1991年2月生于山东省菏泽市. 研究生. 主要研究方向为智能优化算法、系统辨识与控制方法.

E-mail: zhoupeng\_hz@163.com



董朝轶(通信作者) 男, 1976年7月生于内蒙古自治区包头市. 现为内蒙古工业大学教授、硕士生导师, 主要研究方向为地面移动机器人的动态建模、导航、复杂生物网络建模与网络结构辨识.

E-mail: dongchaoyi@hotmail.com