

# 控制与决策

Control and Decision

## 基于自适应正态云模型的引力樽海鞘群算法

张铸, 张仕杰, 饶盛华, 王静袁

引用本文:

张铸, 张仕杰, 饶盛华, 等. 基于自适应正态云模型的引力樽海鞘群算法[J]. 控制与决策, 2022, 37(2): 344–352.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2020.1292>

---

## 您可能感兴趣的其他文章

Articles you may be interested in

### 基于自适应惯性权重的樽海鞘群算法

Salp swarm algorithm based on adaptive inertia weight

控制与决策. 2022, 37(1): 237–246 <https://doi.org/10.13195/j.kzyjc.2020.0454>

### 基于正态云模型的状态转移算法求解多目标柔性作业车间调度问题

State transition algorithm based on normal cloud model for solving multi-objective flexible job shop scheduling problem

控制与决策. 2021, 36(5): 1181–1190 <https://doi.org/10.13195/j.kzyjc.2019.1233>

### 基于双种群模糊引力搜索算法的舰载机甲板作业调度

Flight deck operations scheduling based on dual population fuzzy gravitational search algorithm

控制与决策. 2021, 36(11): 2751–2759 <https://doi.org/10.13195/j.kzyjc.2020.0523>

### 基于自适应正态云模型的灰狼优化算法

Grey wolf optimization algorithm based on adaptive normal cloud model

控制与决策. 2021, 36(10): 2562–2568 <https://doi.org/10.13195/j.kzyjc.2020.0233>

### 一种自适应拟牛顿-状态转移混合智能优化算法及应用

A hybrid state transition optimization algorithm based on adaptive quasi-newton method and its application

控制与决策. 2021, 36(10): 2451–2458 <https://doi.org/10.13195/j.kzyjc.2020.0214>

# 基于自适应正态云模型的引力樽海鞘群算法

张铸<sup>†</sup>, 张仕杰, 饶盛华, 王静袁

(湖南科技大学信息与电气工程学院, 湖南湘潭 411201)

**摘要:** 针对樽海鞘群算法 (salp swarm algorithm, SSA) 在求解复合问题时存在收敛速度慢和容易陷入局部最优等缺点, 提出一种结合引力搜索技术与正态云发生器的樽海鞘群算法 (cloud gravitational SSA, CGSSA). 在更新樽海鞘领导者位置阶段引入引力搜索算法 (gravitational search algorithm, GSA) 中的加速度系数, 避免樽海鞘群的无效搜索, 从而加快搜索速度; 使用正态云模型对樽海鞘追随者位置进行更新, 可丰富种群的多样性; 同时正态云模型熵值能随着迭代次数增加而自适应调整, 可有效地提高迭代后期的收敛精度. 在 23 个基准函数上进行 CGSSA 与其他 10 种优化算法的综合比较, 仿真实验的统计结果、箱线图和收敛曲线表明, 改进后的算法在搜索效率、收敛精度和避免局部最优方面具有更好的性能.

**关键词:** 樽海鞘群算法; 引力搜索算法; 加速度系数; 正态云模型; 自适应云模型; 函数优化

中图分类号: TM352

文献标志码: A

DOI: 10.13195/j.kzyjc.2020.1292

开放科学(资源服务)标识码(OSID):



**引用格式:** 张铸, 张仕杰, 饶盛华, 等. 基于自适应正态云模型的引力樽海鞘群算法[J]. 控制与决策, 2022, 37(2): 344-352.

## Gravity salp swarm algorithm based on adaptive normal cloud model

ZHANG Zhu<sup>†</sup>, ZHANG Shi-jie, RAO Sheng-hua, WANG Jing-yuan

(College of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan 411201, China)

**Abstract:** Aiming at the disadvantages of the salp swarm algorithm (SSA), such as slow convergence speed and easy to fall into local optima when solving complex problems, an improved SSA equipped with gravitational search technique and normal cloud generators (CGSSA) is proposed. The acceleration coefficient of the gravitational search algorithm (GSA) is introduced in the stage of updating the position of the leader of salps, which avoids the invalid search of the salp swarm and accelerates the search speed. The normal cloud model is used to update the position of the followers of the salp, which enriches the diversity of the population. At the same time, the entropy value of the normal cloud model can be adaptively adjusted with the increase of iteration times, which effectively improves the convergence accuracy in the later iteration period. A comprehensive comparison between the CGSSA and other 10 optimization algorithms is made on 23 benchmark functions. The statistical result, convergence curve and box-whisker plot of simulation experiment show that the improved algorithm has better performance in search efficiency, convergence accuracy and avoiding local optimum.

**Keywords:** salp swarm algorithm; gravitational search algorithm; acceleration coefficient; normal cloud model; adaptive cloud mode; function optimization

## 0 引言

基于群体的元启发式算法因其具有概念简单、易于实现和不需要梯度信息的优点, 在解决实际工程问题中受到越来越多的关注. 近 10 年来, 各种受到动物群体社会行为启发的群体智能算法被提出, 如布谷鸟搜索算法<sup>[1]</sup>(cuckoo search, CS)、灰狼优化算法<sup>[2]</sup>(grey wolf optimizer, GWO)和鲸鱼优化算法<sup>[3]</sup>(whale optimization algorithm, WOA)等.

通过模拟樽海鞘群的觅食行为, 文献[4]提出了一种新型优化算法——樽海鞘群算法 (salp swarm algorithm, SSA). 该算法具有控制参数少、灵活性好等优点, 同时对于日益复杂的优化问题, SSA 也是一种优异的方法, 并已成功应用于多个领域. 文献[5]利用 SSA 获得了可再生分布式发电机的最佳位置和容量, 其目标是更高的电压水平、最小的功率损耗和最低的运行成本; 文献[6]使用 SSA 求解无源时

收稿日期: 2020-09-15; 录用日期: 2020-11-05.

责任编辑: 陈家伟.

<sup>†</sup>通讯作者. E-mail: eezhuzhang@gmail.com.

差的定位问题,通过仿真对比可知SSA的定位精度最优;文献[7]将SSA应用于集成学习方法的权重优化,以预测移动通信中接收信号强度与高度之间的关系,结果表明,与常规集成方法相比,预测精度令人满意.然而,与其他元启发式的群体智能算法类似,SSA必须克服收敛速度慢和早熟的缺点.为了提高性能,自SSA提出以来,已经有一些文献对SSA进行了改进.文献[8]使用Tent混沌序列和疯狂算子对SSA进行了改进,结果表明,改进算法的收敛速度和收敛精度有所提高,且在求解高维问题时具有很好的效果;文献[9]中应用莱维飞行SSA(Levy flight SSA, LSSA)对灰度共生矩阵进行了优化,提高了图像的分割效果,该算法在探索与开发之间取得了更好的平衡;文献[10]提出了一种量子行为SSA,利用反向学习算法和小波变异机制的特点,优化了SSA的搜索能力和种群多样性,最后通过工程问题进行了验证,结果表明,改进后的SSA在收敛速度和收敛精度上得到了改善.

为了提高SSA的收敛速度和全局搜索能力,本文提出一种结合引力搜索技术与正态云发生器的樽海鞘群算法.在CGSSA中,加速度系数被用来确定朝向食物的运动方向,加速度系数与质量之间的引力成正比,与自身的惯性质量成反比,这有利于避免盲目地随机搜索食物位置.此外,由于樽海鞘群的随机性和模糊性不高,本文引入正态云模型提高其种群多样性.SSA在改进后可有效地提高搜索速度和跳出局部最优的能力.本文用3类共23个基准函数对所提出的算法进行测试,并与如下9种其他元启发式算法进行比较和分析:SSA、CS、GWO、WOA、增强型SSA<sup>[11]</sup>(enhanced SSA, ESSA)、粒子群优化算法<sup>[12]</sup>(particle swarm optimization, PSO)、快速进化规划算法<sup>[13]</sup>(faster evolutionary programming, FEP)、引力搜索算法<sup>[14]</sup>(gravitational search algorithm, GSA)和蜻蜓算法<sup>[15]</sup>(dragonfly algorithm, DA).测试结果表明,该算法能有效加快收敛速度,显著提高避免局部最优的能力.

## 1 基于自适应正态云模型的引力樽海鞘群算法

### 1.1 樽海鞘群算法

樽海鞘是海洋中一种自由漂浮的生物,它像水母一样以抽水为动力进行移动,并且总是形成一个链状的群体.在捕食过程中,樽海鞘链由前端的领导者引导,其余的樽海鞘跟随其先前的个体,而不是独立运动到食物.通过模仿捕食行为可知SSA的形式如下.

随机初始化樽海鞘 $X_j^i$ ( $i = 1, 2, \dots, N, j = 1, 2, \dots, D$ )的位置为

$$X_j^i = \text{rand}(N, D) \times (\text{ub}_j - \text{lb}_j) + \text{lb}_j. \quad (1)$$

其中: $D$ 为空间维数, $N$ 为种群规模, $\text{ub}_j$ 和 $\text{lb}_j$ 分别为第 $j$ 维空间的上限和下限.

樽海鞘领导者的位置更新公式如下:

$$x_j^1 = \begin{cases} F_j + c_1((\text{ub}_j - \text{lb}_j)c_2 + \text{lb}_j), & c_3 > 0.5; \\ F_j - c_1((\text{ub}_j - \text{lb}_j)c_2 + \text{lb}_j), & c_3 \leq 0.5. \end{cases} \quad (2)$$

其中: $F_j$ 为食物源在第 $j$ 维空间的位置; $c_2$ 和 $c_3$ 为[0, 1]中的随机数;系数 $c_1$ 为樽海鞘领导者位置更新公式中重要参数,它的定义如下:

$$c_1 = 2e^{-\left(\frac{4t}{T_{\max}}\right)^2}. \quad (3)$$

其中: $t$ 为当前迭代次数, $T_{\max}$ 为最大迭代次数.

更新樽海鞘追随者位置的公式如下:

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

其中: $x_j^i$ 和 $x_j^{i-1}$ 分别为第 $i$ 个和第 $i-1$ 个樽海鞘追随者在第 $j$ 维空间的位置.

## 1.2 引力樽海鞘群算法

### 1.2.1 GSA

GSA是Rashedi在2009年提出的一种元启发式搜索技术,具有易于理解、收敛速度快和求解精度高等优点.目前GSA已广泛应用于多种工业和科学研究领域<sup>[16-17]</sup>.

在GSA中, $D$ 维空间中第 $i$ 个粒子的位置为 $X_i = [x_i^1, \dots, x_i^d, \dots, x_i^D]$ ( $i = 1, 2, \dots, N$ ), $x_i^d$ 为第 $i$ 个粒子在第 $d$ 维空间的位置.粒子的惯性质量 $M_i(t)$ 由群体的适应度值计算得出,公式如下:

$$m_i = \frac{\text{fit}_i(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)}, \quad M_i = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)}. \quad (5)$$

其中: $\text{fit}_i(t)$ 为第 $i$ 个粒子在第 $t$ 次迭代的适应值; $\text{best}(t)$ 和 $\text{worst}(t)$ 分别为第 $t$ 次迭代时所有粒子中最佳及最差适应度的个体.

在第 $t$ 次迭代中 $d$ 维空间上第 $i$ 与第 $j$ 个粒子之间的引力定义如下:

$$F_{ij}^d = G(t) \frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)). \quad (6)$$

其中: $M_i(t)$ 和 $M_j(t)$ 为第 $i$ 个粒子和第 $j$ 个粒子的惯性质量; $R_{ij}(t)$ 为两个粒子之间的欧氏距离; $\varepsilon$ 为一个很小的常数; $G(t)$ 为第 $t$ 次迭代时的引力常数,定义为

$$G(t) = G_0 e^{-\omega \frac{t}{T_{\max}}}. \quad (7)$$

这里: $G_0$ 为引力系数初值, $\omega$ 为衰减系数, $T_{\max}$ 为最大迭代次数.作用在第 $i$ 个粒子上的合力表示如下:

$$F_i^d(t) = \sum_{j \in K_{\text{best}}, j \neq i}^N \text{rand}_j F_{ij}^d(t). \quad (8)$$

其中:  $\text{rand}_j$  为  $[0, 1]$  中的一个随机数,  $K_{\text{best}}$  为质量最大的前  $K$  个粒子数.

第  $i$  个粒子的加速度可用牛顿第二定律得到, 即

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)}. \quad (9)$$

### 1.2.2 GSSA

在前面所述 SSA 中, 通过等式 (2) 在食物源附近更新了樽海鞘领导者的位置. 从式 (2) 中可看出, 樽海鞘领导者的搜索方向由参数  $c_3$  决定, 但它是一个随机数, 樽海鞘领导者以完全随机的方式改变其搜索方向, 导致搜索速度慢且精度较低. 为了提高 SSA 的搜索性能, 提出一种引力樽海鞘群算法 (gravitational SSA, GSSA). 将参数  $c_3$  由 GSA 中的加速度代替, 改进后的樽海鞘领导者位置更新公式如下:

$$x_j^1 = F_j + c_1 \times a((\text{ub}_j - \text{lb}_j)c_2 + \text{lb}_j). \quad (10)$$

其中:  $a$  为加速度系数, 如果  $a > 0$ , 则领导者会朝着正方向移动; 否则会朝着反方向移动.

图 1 为樽海鞘领导者在改进前后的搜索轨迹, 虚线表示 GSSA 的搜索轨迹, 实线表示 SSA 的搜索轨迹.

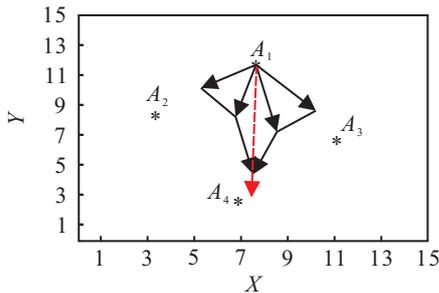


图 1 SSA 和 GSSA 的搜索轨迹

从图 1 中可知, GSSA 的樽海鞘领导者可一步接近可能的食物源位置, 而非 SSA 中的几个随机步骤.

## 1.3 基于自适应正态云模型的 GSSA

### 1.3.1 正态云模型

在众多的不确定性中, 随机性和模糊性无疑是最常见的属性. 为了克服处理不确定性的不足, 文献 [18] 提出了云模型来实现定量描述与定性概念之间的不确定性转换. 云模型的特征在于 3 个数学参数: 期望 ( $\text{Ex}$ )、熵 ( $\text{En}$ ) 和超熵 ( $\text{He}$ )<sup>[19]</sup>.  $\text{En}$  表示  $\text{Ex}$  的熵, 用于度量一个定性概念的随机性, 而  $\text{He}$  则用来度量  $\text{En}$  的随机性.

正态分布是自然界中常见且重要的概率分布, 因此更适合描述搜索粒子的随机性. 正态云模型的定义如下.

设  $C$  是定量论域  $U$  上的定性概念, 若  $\forall x \in U$  是  $C$  的一次随机实现, 则  $x$  对  $C$  的确定度是一个具有稳定倾向的随机数, 且隶属度函数满足

$$\mu = \exp[-(x - \text{Ex})^2 / (2(\text{En})^2)]. \quad (11)$$

其中: 云滴  $x$  在  $U$  中的分布称为正态云模型, 正态云模型的伪代码如下.

- 1) for  $i = 1 : N$
- 2) 生成正态分布随机数  $\text{En}'_i = \text{normrnd}(\text{En}, \text{He})$ ;
- 3) 生成正态分布随机数  $x_i = \text{normrnd}(\text{En}_i, |\text{En}'_i|)$ ,  $x_i$  是云滴;
- 4) 计算  $y_i = \exp[-(x_i - \text{Ex})^2 / (2(\text{En}'_i)^2)]$ ;
- 5) end for

就转换方向而言, 有两种类型的正态云发生器, 即正向和反向, 本文使用正向的正态云发生器. 正态云发生器根据设定的数字特征运行一次即产生一个云滴, 直到运行生成期望数量的云滴. 正态云滴生成过程可定义为如下形式:

$$X[x_1, x_2, \dots, x_{\text{Nd}}] = \text{Gnc}(\text{Ex}, \text{En}, \text{He}, \text{Nd}), \quad (12)$$

其中  $\text{Nd}$  为云滴的期望数量.

图 2 为云滴的分布与正常云模型的数学参数之间的关系. 显而易见, 云滴的分布范围随  $\text{En}$  的增加而扩大, 离散度随  $\text{He}$  的增加而增加, 很好地反映了正态云模型的随机性和模糊性.

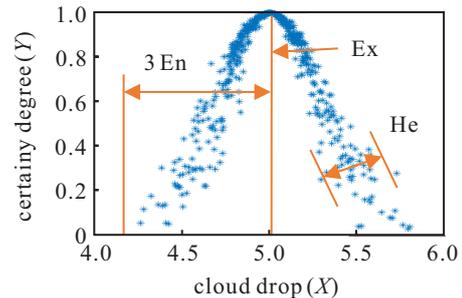


图 2 正态云发生器的云滴分布

### 1.3.2 CGSSA

在 SSA 中, 樽海鞘追随者的分布缺乏多样性, 这限制了 SSA 在解决多模态问题时的探索. 而正态云模型被认为是描述随机性和模糊性的可靠技术, 因此它适用于改进樽海鞘跟随者的位置更新机制. 新的樽海鞘追随者位置更新公式如下:

$$\text{flwr\_SP}_j = \text{Gnc}(\text{ldr\_SP}_j, \text{En}, \text{He}, \text{Nd}), \quad (13)$$

其中  $\text{ldr\_SP}_j$  和  $\text{flwr\_SP}_j$  分别为第  $j$  维空间中樽海鞘领导者和追随者的位置.

对于式 (13), 可通过改变  $\text{En}$  的值调整樽海鞘追随者与领导者的距离, 通过改变  $\text{He}$  的值调整樽海鞘追随者的分散程度. 所以  $\text{En}$  和  $\text{He}$  的取值可随着迭代次数增加而自适应调整, 取值如下:

$$\text{En} = \theta \times \left( \frac{T_{\text{max}} - t}{T_{\text{max}}} \right)^\alpha, \quad (14)$$

$$\text{He} = \text{En} \times 10^\beta. \quad (15)$$

其中:  $\theta$  为  $[0,1]$  中的常数,  $\alpha$  和  $\beta$  为正整数,  $T_{\max}$  为最大迭代次数,  $t$  为当前迭代次数.

综上所述, 本文提出的 CGSSA 步骤可总结如下.

- step 1: 初始化算法参数  $N$ 、 $D$ 、 $ub$ 、 $lb$ 、 $T_{\max}$ 、 $G_0$ 、 $\omega$ 、 $K_{\text{best}}$ 、 $\theta$ 、 $\alpha$ 、 $\beta$ ;
- step 2: 根据式(1)初始化种群;
- step 3: 计算樽海鞘个体的适应度值并进行排序, 选出适应度最优个体为食物源  $F$ ;
- step 4: 计算  $worst$ 、 $best$  的值, 通过式(7)计算  $G$ ;
- step 5: 根据式(9)计算加速度系数  $a$ ;
- step 6: 根据式(3)更新参数  $c_1$ ;
- step 7: 根据式(10)更新樽海鞘领导者的位置;
- step 8: 通过式(14)和(15)计算正态云模型的参数  $En$  和  $He$ ;
- step 9: 根据式(13)更新樽海鞘追随者的位置;
- step 10: 修正超出搜索空间的樽海鞘;

step 11: 计算更新后的樽海鞘个体适应度值, 重新进行排序, 选出最优个体作为新的  $F$ ;

step 12: 判断迭代次数是否达到  $T_{\max}$ , 如果  $t < T_{\max}$ , 则  $t = t + 1$ , 返回 step 4, 否则算法结束.

## 2 仿真实验与结果分析

本文的仿真实验是在 Matlab R2018a 上运行的, 且使用了 3 种类型共 23 个测试函数进行基准测试. 表 1 给出了基准函数的数学公式、范围、维数和最优解. 在这些基准函数中:  $F_1 \sim F_7$  为单峰基准函数, 具有单一最优解, 可对局部开发能力进行测试;  $F_8 \sim F_{13}$  为多峰基准函数, 具有多个局部最优解和一个全局最优解, 可以有效地检验算法的全局寻优能力;  $F_{14} \sim F_{23}$  为固定维多峰基准函数, 该类型函数更能贴近实际的工程优化问题, 需要优化算法的搜索与开发能力更平衡, 以避免陷入局部最优. 因此, 该算法的优化能力可以得到充分的检验.

表 1 基准函数

| 函数   | 维数 | 范围              | 最优解 |
|--|----|-----------------|-----|
| $F_1(x) = \sum_{i=1}^n x_i^2$  | 30 | $[-100, 100]$   | 0   |
| $F_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $  | 10 | $[-10, 10]$     | 0   |
| $F_3(x) = \sum_{i=1}^n \left( \sum_{j=1}^i x_j^2 \right)$  | 10 | $[-100, 100]$   | 0   |
| $F_4(x) = \max\{ x_i , 1 \leq i \leq n\}$  | 10 | $[-100, 100]$   | 0   |
| $F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$   | 10 | $[-100, 100]$   | 0   |
| $F_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$  | 10 | $[-100, 100]$   | 0   |
| $F_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$   | 10 | $[-1.28, 1.28]$ | 0   |
| $F_8(x) = \sum_{i=1}^n -x \sin(\sqrt{ x_i })$  | 10 | $[-500, 500]$   | 0   |
| $F_9(x) = \sum_{i=1}^n [x_i^n - 10 \cos(2\pi x_i) + 10]$   | 10 | $[-5.12, 5.12]$ | 0   |
| $F_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left[\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right] + 20 + e$  | 10 | $[-32, 32]$     | 0   |
| $F_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$  | 10 | $[-600, 600]$   | 0   |
| $F_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1}) + (y_n - 1)^2] \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$<br>$y_i = 1 + \frac{x_i + 1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a; \\ 0, & -a < x_i < a; \\ k(-x_i - a)^m, & x_i < -a. \end{cases}$ | 10 | $[-50, 50]$     | 0   |
| $F_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1) + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)]] \right\} + \sum_{i=1}^n u(x_i, 5100, 4)$  | 10 | $[-50, 50]$     | 0   |

表1 (续)

| 函数  | 维数 | 范围              | 最优解    |
|---|----|-----------------|--------|
| $F_{14}(x) = \left( \frac{1}{500} + \sum_{i=1}^{25} \frac{1}{i + \sum_{j=1}^2 (x_i - a_{ij})} \right)^{-1}$   | 2  | [-65.54, 65.54] | 1      |
| $F_{15}(x) = \sum_{i=1}^{11} \left[ a_i - \frac{x_1(b_i^2 - b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$   | 4  | [-5, 5]         | 0.003  |
| $F_{16}(x) = 4x_1^2 - 2.1x_1^2 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$   | 2  | [-5, 5]         | -1.032 |
| $F_{17}(x) = \left( x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + 10 \left( 1 - \frac{1}{8\pi} \right) \cos x_1 + 10$  | 2  | [-5, 5]         | 0.396  |
| $F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 3x_2^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2](18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)$ | 2  | [-2, 2]         | 3      |
| $F_{19}(x) = -\sum_{i=1}^4 c_i \exp \left( -\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2 \right)$  | 3  | [1, 3]          | -3.86  |
| $F_{20}(x) = -\sum_{i=1}^4 c_i \exp \left( -\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2 \right)$  | 6  | [0, 1]          | -3.32  |
| $F_{21}(x) = -\sum_{i=1}^5 [(x - a_i)(x - a_i)^T + c_i]^{-1}$   | 4  | [0, 10]         | -10.15 |
| $F_{22}(x) = -\sum_{i=1}^7 [(x - a_i)(x - a_i)^T + c_i]^{-1}$   | 4  | [0, 10]         | -10.15 |
| $F_{23}(x) = -\sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$  | 4  | [0, 10]         | -10.15 |

2.1 整体性能对比

为了观察CGSSA的整体性能,使用了8种算法进行比较:SSA、ESSA、PSO、CS、GSA、FEP、GWO和WOA.为了获得可靠的测试结果,每个算法在23个基准函数上都进行30次独立的运算,通过对比平均值

和标准差判断优化算法的性能.优化算法的平均值和标准差越小,性能越佳.表2为单峰基准函数( $F_1 \sim F_7$ )测试的统计结果.从表2中可知,在7个单峰基准函数测试下,CGSSA的统计结果都优于其他8种算法,表明CGSSA的寻优精度高且局部开发能力优异.

表2 单峰基准函数测试的统计结果

| F              | Index | CGSSA       | ESSA        | SSA       | GWO       | WOA       | PSO         | GSA       | FEP       | CS          |
|----------------|-------|-------------|-------------|-----------|-----------|-----------|-------------|-----------|-----------|-------------|
| F <sub>1</sub> | mean  | 1.673 8e-54 | 1.122 5e-38 | 4.313e-09 | 2.452e-14 | 6.257e-15 | 2.325e-05   | 3.125e-14 | 5.521e-04 | 6.25e-03    |
|                | std   | 2.705 4e-54 | 1.254 2e-37 | 1.093e-09 | 4.325e-14 | 5.214e-15 | 3.214 5e-05 | 6.254e-14 | 1.254e-04 | 3.425 8e-03 |
|                | rank  | 1           | 2           | 6         | 4         | 3         | 7           | 5         | 8         | 9           |
| F <sub>2</sub> | mean  | 2.915e-28   | 2.012 3e-25 | 3.226e-05 | 3.256e-14 | 1.212e-15 | 5.612e-03   | 5.41e-02  | 7.26e-03  | 3.256e-01   |
|                | std   | 2.261 2e-28 | 7.741 2e-25 | 1.393e-04 | 2.156e-14 | 2.021e-15 | 2.13e-02    | 2.012e-01 | 5.23e-04  | 2.156e-02   |
|                | rank  | 1           | 2           | 5         | 4         | 3         | 6           | 8         | 7         | 9           |
| F <sub>3</sub> | mean  | 1.853e-45   | 3.361 2e-35 | 2.067e-09 | 3.452e-05 | 4.356e-07 | 5.15e+02    | 7.24e+02  | 1.523e-02 | 2.053e-01   |
|                | std   | 3.044 3e-45 | 1.842 5e-34 | 1.126e-09 | 7.154e-05 | 3.278e-06 | 562.14      | 318.154   | 1.231e-02 | 2.945e-02   |
|                | rank  | 1           | 2           | 3         | 5         | 4         | 8           | 9         | 6         | 7           |
| F <sub>4</sub> | mean  | 1.362 7e-27 | 4.235 6e-18 | 1.639e-05 | 6.782e-07 | 0.042 1   | 18.254      | 0.952     | 0.214     | 1.754e-04   |
|                | std   | 1.481 1e-27 | 2.124 1e-17 | 3.790e-06 | 5.121e-07 | 0.541     | 5.214       | 1.415     | 0.457     | 3.428e-05   |
|                | rank  | 1           | 2           | 4         | 3         | 6         | 9           | 8         | 7         | 5           |
| F <sub>5</sub> | mean  | 1.435 9e-24 | 25.880 3    | 186.397 7 | 30.528    | 27.178    | 120.256     | 72.156    | 5.006     | 6.125e-03   |
|                | std   | 2.984 1e-24 | 0.029 8     | 409.936   | 1.325     | 1.253     | 98.526      | 68.521    | 5.64      | 5.21e-02    |
|                | rank  | 1           | 4           | 9         | 6         | 5         | 8           | 7         | 3         | 2           |
| F <sub>6</sub> | mean  | 0           | 3.125 6     | 6.800e-10 | 0.8526    | 2.351     | 5.46e-05    | 2.42e-16  | 0         | 6.17e-04    |
|                | std   | 0           | 0.834 5     | 1.701e-10 | 0.452     | 0.856     | 1.23e-04    | 1.685e-16 | 0         | 2.814e-05   |
|                | rank  | 1           | 8           | 3         | 6         | 7         | 4           | 2         | 1         | 5           |
| F <sub>7</sub> | mean  | 1.068e-05   | 8.821 2e-05 | 6.045e-03 | 1.56e-03  | 1.425e-03 | 5.89e-02    | 7.86e-02  | 0.152 3   | 2.28e-02    |
|                | std   | 1.413 9e-05 | 6.651 2e-05 | 5.307e-03 | 1.23e-03  | 1.149e-03 | 2.13e-02    | 4.25e-02  | 0.354 6   | 1.347e-03   |
|                | rank  | 1           | 2           | 5         | 4         | 3         | 7           | 8         | 9         | 6           |

表3为多峰基准函数( $F_8 \sim F_{13}$ )测试的统计结果,从表3中可知,在大多数多峰基准函数中,CGSSA的统计结果都是最优的,只在F<sub>8</sub>基准函数测试中排

在第2位,仅次于FEP算法.考虑到多模态函数的特点,依然能表明CGSSA具有优秀的探索能力.表4为固定维多峰基准函数( $F_{14} \sim F_{23}$ )测试的统计结果.

表3 多峰基准函数测试的统计结果

| $F$      | Index | CGSSA      | ESSA       | SSA       | GWO        | WOA       | PSO       | GSA      | FEP       | CS        |
|----------|-------|------------|------------|-----------|------------|-----------|-----------|----------|-----------|-----------|
| $F_8$    | mean  | -8875.1424 | -4830.2142 | -2856.617 | -5842.15   | -4896.98  | -8546.89  | -2563.08 | -12856    | -2289.89  |
|          | std   | 3573.8788  | 708.3124   | 373.9031  | 725.28     | 702.56    | 723.45    | 458.32   | 45.23     | 0.589     |
|          | rank  | 2          | 6          | 7         | 4          | 5         | 3         | 8        | 1         | 9         |
| $F_9$    | mean  | 0          | 0          | 18.274    | 3.5689     | 0         | 1.2546    | 19.256   | 5.2e-02   | 0.589     |
|          | std   | 0          | 0          | 8.9812    | 3.9826     | 0         | 10.2634   | 8.524    | 2.354e-02 | 2.58e-03  |
|          | rank  | 1          | 1          | 6         | 5          | 1         | 4         | 7        | 2         | 3         |
| $F_{10}$ | mean  | 8.8818e-16 | 8.8818e-16 | 0.6336    | 8.958e-12  | 5.452     | 0.9586    | 5.23e-02 | 2.156e-02 | 4.25e-10  |
|          | std   | 0          | 0          | 0.8358    | 7.562e-13  | 8.248     | 1.245     | 0.3156   | 3.124e-03 | 5.36e-09  |
|          | rank  | 1          | 1          | 6         | 2          | 8         | 7         | 5        | 4         | 3         |
| $F_{11}$ | mean  | 0          | 0          | 0.2441    | 6.15e-03   | 2.813e-03 | 1.891e-02 | 20.356   | 1.859e-02 | 0.2481    |
|          | std   | 0          | 0          | 0.1412    | 1.23e-02   | 2.58e-03  | 2.31e-02  | 3.256    | 2.346e-02 | 3.56e-02  |
|          | rank  | 1          | 1          | 6         | 3          | 2         | 4         | 8        | 5         | 7         |
| $F_{12}$ | mean  | 4.7116e-32 | 0.2825     | 0.4673    | 3.1259e-02 | 0.2589    | 0.9123    | 1.8456   | 9.254e-06 | 1.258e-02 |
|          | std   | 1.6702e-47 | 0.15255    | 0.8264    | 1.259e-02  | 0.2546    | 1.3589    | 0.9812   | 3.146e-06 | 5.218e-08 |
|          | rank  | 1          | 6          | 7         | 4          | 5         | 8         | 9        | 2         | 3         |
| $F_{13}$ | mean  | 1.3498e-32 | 2.8433     | 1.098e-03 | 0.6523     | 1.5689    | 0.256     | 7.526    | 1.256e-04 | 0.521     |
|          | std   | 5.5674e-48 | 0.3489     | 3.352e-03 | 0.3546     | 0.3426    | 0.3256    | 5.231    | 7.256e-06 | 6.259e-08 |
|          | rank  | 1          | 8          | 3         | 6          | 7         | 4         | 9        | 2         | 5         |

表4 固定维多峰基准函数测试的统计结果

| $F$      | Index | CGSSA      | ESSA       | SSA       | GWO       | WOA       | PSO       | GSA       | FEP       | CS        |
|----------|-------|------------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| $F_{14}$ | mean  | 0.998      | 3.9852     | 0.998     | 5.0231    | 2.561     | 0.998     | 5.892     | 2.130     | 1.568     |
|          | std   | 0          | 3.6245     | 2.386e-16 | 4.235     | 2.648     | 1.367e-10 | 3.256     | 0.234     | 2.5e-03   |
|          | rank  | 1          | 7          | 2         | 8         | 6         | 3         | 9         | 5         | 4         |
| $F_{15}$ | mean  | 3.261e-05  | 2.4125e-03 | 8.901e-04 | 5.26e-03  | 5.75e-05  | 3.25e-03  | 3.578e-03 | 5.125e-04 | 5.124e-04 |
|          | std   | 8.9776e-06 | 5.3214e-03 | 2.765e-04 | 5.6e-03   | 3.814e-04 | 7.25e-03  | 1.859e-03 | 3.215e-04 | 1.561e-04 |
|          | rank  | 1          | 7          | 5         | 9         | 2         | 6         | 8         | 4         | 3         |
| $F_{16}$ | mean  | -1.0316    | -1.0316    | -1.0316   | -1.0316   | -1.0316   | -1.0316   | -1.0316   | -1.032    | -1.0316   |
|          | std   | 5.151e-16  | 6.5187e-06 | 1.247e-14 | 2.579e-12 | 5.236e-07 | 7.125e-16 | 4.258e-15 | 5.218e-08 | 1.481e-08 |
|          | rank  | 1          | 9          | 4         | 5         | 8         | 2         | 3         | 7         | 6         |
| $F_{17}$ | mean  | 0.3979     | 0.3979     | 0.3978    | 0.3978    | 0.3979    | 0.3979    | 0.3979    | 0.398     | 0.398     |
|          | std   | 0          | 1.1254e-05 | 7.527e-15 | 8.95e-07  | 3.452e-06 | 0         | 0         | 1.423e-06 | 3.25e-06  |
|          | rank  | 1          | 7          | 2         | 3         | 6         | 1         | 1         | 4         | 5         |
| $F_{18}$ | mean  | 3          | 3.0003     | 3         | 3         | 3         | 3         | 3         | 3.021     | 3.01      |
|          | std   | 2.469e-15  | 3.6902e-04 | 8.476e-14 | 4.21e-05  | 4.35e-015 | 1.561e-14 | 4.235e-15 | 0.234     | 2.15e-03  |
|          | rank  | 1          | 7          | 5         | 6         | 3         | 4         | 2         | 9         | 8         |
| $F_{19}$ | mean  | -3.8628    | -3.8261    | -3.8628   | -3.2568   | -3.856    | -3.8625   | -3.8627   | -3.86     | -3.268    |
|          | std   | 2.484e-15  | 5.3312e-02 | 6.686e-14 | 3.21e-03  | 2.71e-03  | 4.243e-15 | 3.256e-15 | 2.531e-07 | 2.121e-05 |
|          | rank  | 1          | 7          | 4         | 9         | 8         | 3         | 2         | 6         | 5         |
| $F_{20}$ | mean  | -3.0237    | -3.1703    | -3.2123   | -3.2568   | -2.982    | -3.2518   | -3.317    | -3.27     | -3.322    |
|          | std   | 8.796e-02  | 0.1194     | 3.741e-02 | 8.85e-02  | 0.3723    | 0.2156    | 2.315e-02 | 523e-02   | 7.214e-03 |
|          | rank  | 8          | 7          | 6         | 4         | 9         | 5         | 2         | 3         | 1         |
| $F_{21}$ | mean  | -10.1532   | -10.1478   | -8.3048   | -9.3954   | -7.045    | -6.3248   | -5.955    | -5.52     | -9.728    |
|          | std   | 2.624e-15  | 1.4325e-02 | 2.7181    | 2.145     | 3.6321    | 2.985     | 3.7524    | 2.15      | 0.325     |
|          | rank  | 1          | 2          | 6         | 4         | 8         | 7         | 9         | 5         | 3         |
| $F_{22}$ | mean  | -10.4029   | -10.4004   | -9.366    | -10.2241  | -8.182    | -7.039    | -9.368    | -5.426    | -9.853    |
|          | std   | 5.234e-14  | 5.0124e-02 | 2.4183    | 0.9521    | 3.829     | 3.567     | 2.526     | 2.356     | 0.3256    |
|          | rank  | 1          | 2          | 6         | 3         | 7         | 8         | 5         | 9         | 4         |
| $F_{23}$ | mean  | -10.5364   | -10.5331   | -9.8445   | -10.0835  | -9.2856   | -7.7241   | -10.5342  | -7.53     | -9.8752   |
|          | std   | 5.237e-15  | 5.1243e-02 | 2.1406    | 1.5246    | 2.4247    | 3.6258    | 3.15e-14  | 3.241     | 0.5246    |
|          | rank  | 1          | 3          | 6         | 4         | 7         | 9         | 2         | 8         | 5         |

从表4中可知,在10个固定维多峰基准函数测试下,CGSSA的统计结果都是最优的,考虑到固定维多峰基准函数的复杂性,验证了CGSSA在避免局部最优方面具有突出的能力。

2.2 适应度值对比

箱线图通常以统计的方式描述变量的离散。

为了进一步观察CGSSA的性能,CGSSA、ALO、DA、GSA、GWO、PSO、SSA和WOA用部分基准函数分别进行了30次独立运行,所得箱线图如图3所示。

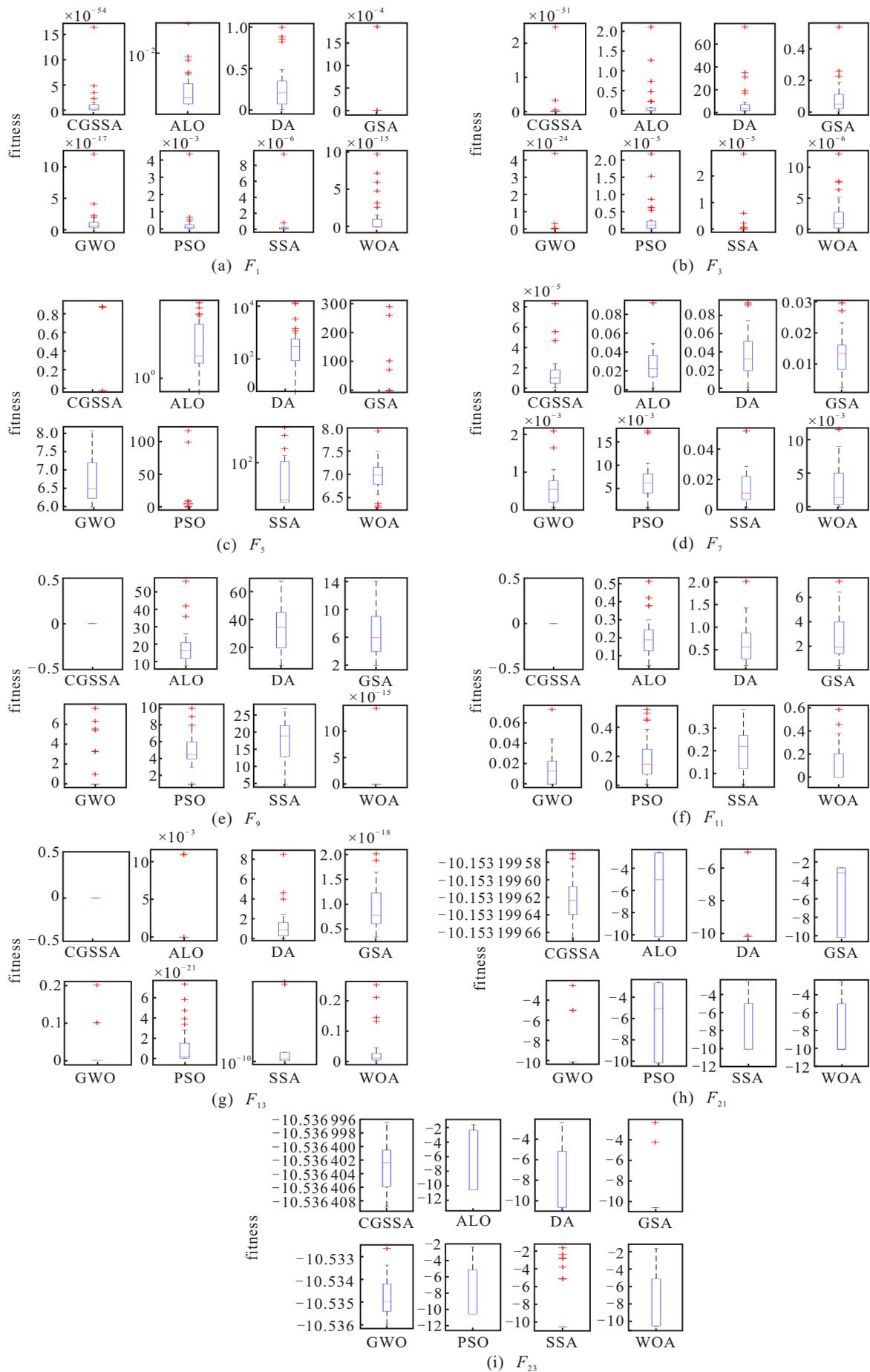


图3 部分基准函数的收敛曲线

通过比较适应度的中值、四分位数、范围和可能的极值,观察到CGSSA具有最佳的全局探索能力,因为它可以在大多数测试函数上达到全局最优,并且比其他算法具有更高的精度.

### 2.3 收敛曲线对比

收敛速度、收敛精度和避免局部最优能力是检验优化算法的重要指标. 图4为部分测试函数的收敛曲线. 从单峰基准函数( $F_1$ 、 $F_3$ 、 $F_5$ 、 $F_7$ )上的曲线可以看出,CGSSA的收敛精度明显高于其他算法,表明在加入自适应正态云模式后,在樽海鞘领导者周围产

生的多样化的樽海鞘追随者,提高了CGSSA的开发能力. 从多峰基准函数( $F_9$ 、 $F_{11}$ 、 $F_{13}$ )上的曲线可以看出,与其他算法相比,CGSSA的收敛速度快,表明在更新樽海鞘领导者位置时引入了加速度系数后,加速度系数为樽海鞘领导者提供了向食物位置移动的方向,避免了无效搜索,加快了搜索速度. 此外,通过观察固定维多峰基准函数( $F_{21}$ 、 $F_{23}$ )的曲线可知,CGSSA停滞的次数比其他算法少,即使CGSSA陷入局部最优,也能随着迭代次数的增加快速跳出,收敛行为验证了CGSSA的搜索与开发能力是更为平衡的,因此,验证了CGSSA的优越性.

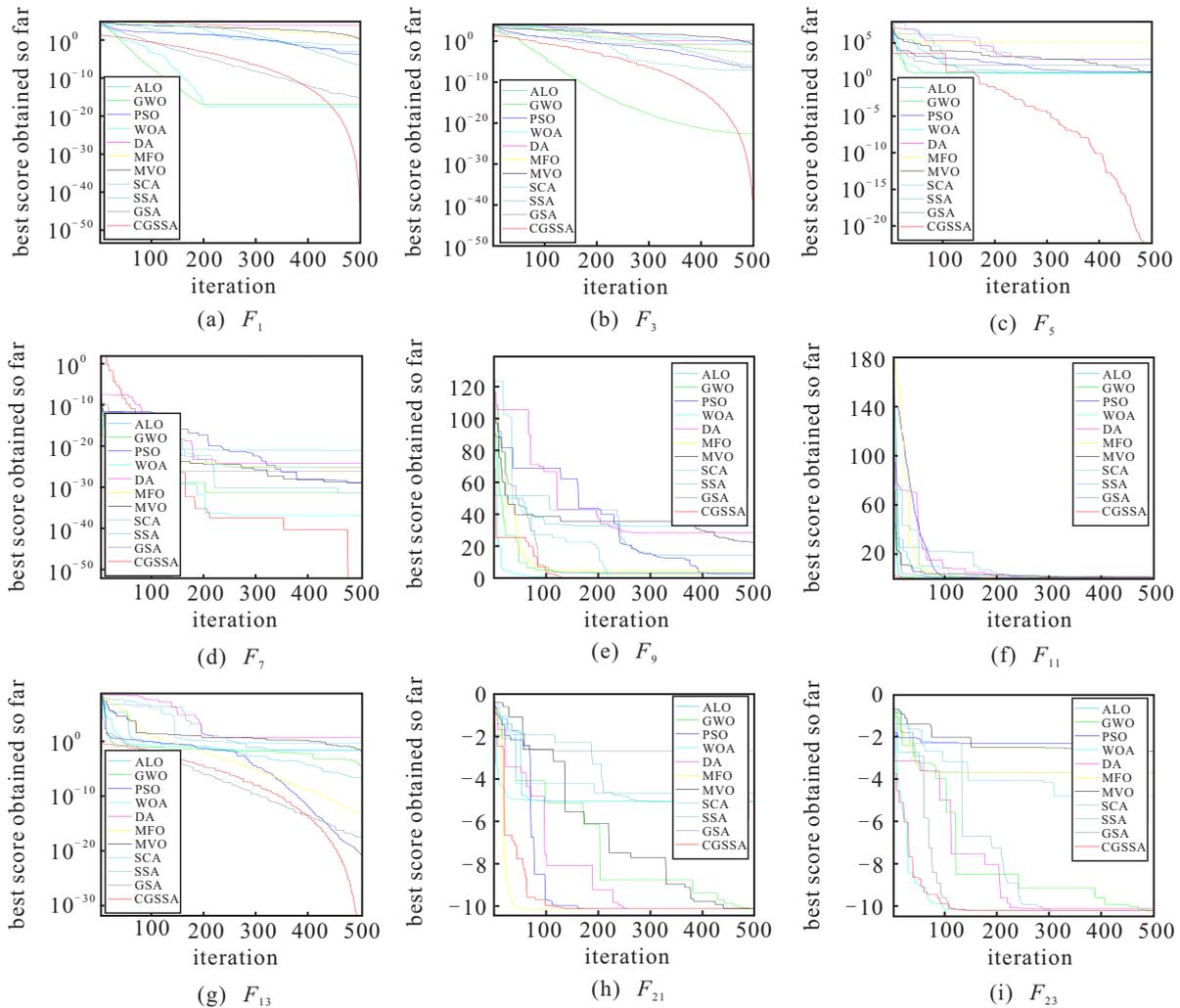


图4 部分基准函数的收敛曲线

### 3 结论

本文提出了一种结合引力搜索技术与正态云发生器的樽海鞘群算法. 通过引入加速系数引导樽海鞘领导者的移动方向,并利用正态云生成器增强樽海鞘群体的种群多样性. 通过3个阶段的实验,对CGSSA的搜索和开发能力进行了评价. 首先,使用23个基准函数获得算法的总体评估,结果表明,该算法

优于其他与之对比的算法. 从单峰和多峰测基准函数的测试结果可以看出,CGSSA具有更好的开发和探索能力. 其次,给出了CGSSA、ALO、DA、GSA、GWO、PSO、SSA和WOA在部分基准函数上的箱线图,从结果可知,CGSSA可以提供更稳定和更精确的探索. 最后,对大部分基准函数的收敛行为进行了比较,收敛曲线验证了CGSSA在开发和探索中的优越

性.

### 参考文献(References)

- [1] Rajabioun R. Cuckoo optimization algorithm[J]. Applied Soft Computing, 2011, 11(8): 5508-5518.
- [2] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46-61.
- [3] Mirjalili S, Lewis A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.
- [4] 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: 163-191.
- [5] Tolba M, Rezk H, Diab A, et al. A novel robust methodology based salp swarm algorithm for allocation and capacity of renewable distributed generators on distribution grids[J]. Energies, 2018, 11(10): 2556.
- [6] 陈涛, 王梦馨, 黄湘松. 基于樽海鞘群算法的无源时差定位[J]. 电子与信息学报, 2018, 40(7): 1591-1597.  
(Chen T, Wang M X, Huang X S. Time difference of arrival passive location based on salp swarm algorithm[J]. Journal of Electronics & Information Technology, 2018, 40(7): 1591-1597.)
- [7] Goudos S K, Athanasiadou G. Application of an ensemble method to UAV power modeling for cellular communications[J]. IEEE Antennas and Wireless Propagation Letters, 2019, 18(11): 2340-2344.
- [8] 张达敏, 陈忠云, 辛梓芸, 等. 基于疯狂自适应的樽海鞘群算法[J]. 控制与决策, 2020, 35(9): 2112-2120.  
(Zhang D M, Chen Z Y, Xin Z Y, et al. Salp swarm algorithm based on craziness and adaptive[J]. Control and Decision, 2020, 35(9): 2112-2120.)
- [9] Xing Z K, Jia H M. Multilevel color image segmentation based on GLCM and improved salp swarm algorithm[J]. IEEE Access, 2019, 7: 37672-37690.
- [10] Chen R Z, Dong C, Ye Y, et al. QSSA: Quantum evolutionary salp swarm algorithm for mechanical design[J]. IEEE Access, 2019, 7: 145582-145595.
- [11] Qais M H, Hasanien H M, Alghuwainem S. Enhanced salp swarm algorithm: Application to variable speed wind feedings of generators[J]. Engineering Applications of Artificial Intelligence, 2019, 80: 82-96.
- [12] Kennedy J, Eberhware R C. Particle swarm optimization[J]. Proceedings of IEEE International Conference on Neural Network. Perth, 1995, 4: 1942-1948.
- [13] Yao X, Liu Y, Lin G M. Evolutionary programming made faster[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82-102.
- [14] Rashedi E, Nezamabadi-Pour H, Saryaidi S. GSA: A gravitational search algorithm[J]. Information Sciences, 2009, 179(13): 2232-2248.
- [15] Mirjalili S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems[J]. Neural Computing and Applications, 2016, 27(4): 1053-1073.
- [16] Askari H, Zahiri S H. Decision function estimation using intelligent gravitational search algorithm[J]. International Journal of Machine Learning and Cybernetics, 2012, 3(2): 163-172.
- [17] Xiao J H, Niu Y Y, Chen P, et al. An improved gravitational search algorithm for green partner selection in virtual enterprises[J]. Neurocomputing, 2016, 217: 103-109.
- [18] 李德毅, 孟海军, 史雪梅. 隶属云和隶属云发生器[J]. 计算机研究与发展, 1995, 32(6): 15-20.  
(Li D Y, Meng H J, Shi X M. Membership clouds and membership cloud generators[J]. Journal of Computer Research and Development, 1995, 32(6): 15-20.)
- [19] Li D Y, Liu C Y, Gan W Y. A new cognitive model: Cloud model[J]. International Journal of Intelligent Systems, 2009, 24(3): 357-375.

### 作者简介

张铸(1981—), 男, 讲师, 博士, 从事电力电子与电力传动、智能控制等研究, E-mail: eezhuzhang@gmail.com;

张仕杰(1995—), 男, 硕士生, 从事电力电子与电力传动、智能控制的研究, E-mail: Zhangsj007@163.com;

饶盛华(1993—), 男, 硕士生, 从事电力传动的研究, E-mail: 1014791899@qq.com;

王静袁(1997—), 男, 硕士生, 从事电力电子与电力传动的研究, E-mail: 2211294225@qq.com.

(责任编辑: 闫妍)