

集成随机惯性权重和差分变异操作的樽海鞘群算法



张志强 鲁晓锋 隋连升 李军怀

西安理工大学计算机科学与工程学院 西安 710048

摘要 为了提高樽海鞘群算法(Salp Swarm Algorithm, SSA)的收敛速度、计算精度和全局优化能力,在分析总结粒子群优化(Particle Swarm Optimization, PSO)和差分进化(Differential Evolution, DE)算法相关研究成果后,提出了一种集成 PSO 算法随机惯性权重和 DE 算法差分变异操作的改进 SSA 算法——iSSA。首先,将 PSO 算法的随机惯性权重引入 SSA 算法的追随者位置更新公式中,用于增强和平衡 SSA 算法的勘探与开发能力;其次,用 DE 算法的变异操作替代 SSA 算法的领导者位置更新操作,以提高 SSA 算法的收敛速度和计算精度。为了检验随机惯性权重和差分变异操作对 SSA 算法的改进效果,在多个高维基准函数上进行了仿真实验,并与其他改进 SSA 算法进行了比较。实验结果及分析表明,与 SSA 算法和两个典型的改进 SSA 算法(ESSA 和 CASSA)相比,集成随机惯性权重和差分变异操作的 iSSA 算法,在没有增加算法时间复杂度的情况下,显著地提高了 SSA 算法的收敛速度、计算精度和全局优化能力,并且优于 ESSA 算法和 CASSA 算法。

关键词: 樽海鞘群算法;群体智能;粒子群优化;随机惯性权重;差分进化;变异操作

中图法分类号 TP183

Salp Swarm Algorithm with Random Inertia Weight and Differential Mutation Operator

ZHANG Zhi-qiang, LU Xiao-feng, SUI Lian-sheng and LI Jun-huai

School of Computer Science & Engineering, Xi'an University of Technology, Xi'an 710048, China

Abstract After analyzing and summarizing the related research achievements of particle swarm optimization (PSO) and differential evolution (DE) algorithms, this paper proposes an improved salp swarm algorithm (iSSA) with random inertia weight and differential mutation operator for enhancing the convergence rate, computational accuracy and global optimization of SSA algorithm. Firstly, the random inertia weight of particle swarm optimization is introduced in the followers' position update equation of SSA algorithm in order to enhance and balance the exploration and exploitation ability of SSA algorithm. Secondly, the leader position update of SSA algorithm is replaced with the mutation operator of differential evolution algorithm DE, which is used to improve the convergence rate and computational accuracy of SSA algorithm. For the purpose of checking the improvement effect on SSA algorithm by the random inertia weight and differential mutation operator, simulation experiments on some high-dimension benchmark functions are performed and comparisons between the proposed iSSA and two typical improved SSA algorithms named by enhanced salp swarm algorithm (ESSA) and crazy and adaptive salp swarm algorithm (CASSA) are performed. The experimental results and analysis show that, the proposed iSSA with random inertia weight and differential mutation operator obviously improves the convergence rate, computational accuracy and global optimization when compared with two typical improved ESSA and CASSA algorithms in the case of no any increase in the time complexity of SSA algorithm.

Keywords Salp swarm algorithm, Swarm intelligence, Particle swarm optimization, Random inertia weight, Differential evolution, Mutation operator

收稿日期:2019-07-07 返修日期:2019-10-04 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:陕西省教育厅自然科学研究项目(18JK0557);陕西省科技计划项目(2018HJCG-05);西安市科技计划项目(201805037YD15CG21(4))

This work was supported by the Natural Science Research Program of the Educational Department of Shaanxi Province (18JK0557), Scientific Research Program of Shaanxi Province (2018HJCG-05) and Project of Xi'an Science and Technology Planning Foundation (201805037YD15CG21(4)).

通信作者:张志强(chinazzq@126.com)

1 引言

樽海鞘群算法(SSA)^[1]是澳大利亚 Griffith 大学学者 Mirjalili 于 2017 年提出的一种群体智能(Swarm Intelligence)算法,最初被用于解决工程设计中的最优化问题。SSA 算法具有控制参数少、简单易行、内含能避免陷入局部最优的自适应机制等优点,已被应用于特征选择、电力系统、电气工程、自动控制等领域^[2]。

然而,在解决高维复杂优化问题时,SSA 算法存在收敛速度慢、计算精度差和易于陷入局部最优等问题。为此,近年来一些学者开始研究改进 SSA 算法。例如,Sayed 等提出将混沌理论融入 SSA 算法中,即通过 10 种不可逆的混沌映射调节 SSA 算法的控制参数 c_2 ,使作为领导者的樽海鞘个体按照混沌映射的方式更新最佳位置,提高了算法的收敛速度和计算精度^[3]。Wang 等提出通过单纯形法更新最差追随者的位置,来改善 SSA 算法的种群多样性和局部搜索能力^[4]。Qais 等重构了 SSA 算法控制参数 c_1 的计算公式,并将追随者位置更新公式中的固定系数 0.5 改为均匀分布于 $[0,1]$ 区间的随机数,用于平衡算法的勘探(exploration)和开发(exploitation)^[5]。Poli 等提出将粒子群优化(PSO)算法^[6]的动态线性递减惯性权重引入 SSA 算法的追随者位置更新公式中,以平衡 SSA 算法的勘探和开发。为避免 SSA 算法陷入局部最优,Wu 等提出针对最优解执行自适应变异操作^[7]。Mirjalili 开发了一种 SSA 算法和正弦-余弦算法^[8]的混合算法,采用内含正弦和余弦函数的领导者位置更新公式,促使 SSA 算法获得最优解,将最优解信息引入追随者位置更新公式中,以提高 SSA 算法的勘探和开发能力^[9]。Majhi 等提出根据 QIF(Quadratic Integrate and Fire)模型产生混沌映射,并将其融入 SSA 算法中的追随者位置更新公式中,以改善算法勘探与开发之间的平衡,提高种群多样性和避免算法陷入局部最优^[10]。Yang 等提出采用多个独立的樽海鞘链扩展 SSA 算法,并由此构建 Memetic 计算框架,以改善算法的勘探和开发能力;并将多个樽海鞘群重组为一个虚拟种群,用于不同樽海鞘链之间的全局协作,以增强算法收敛的稳定性^[11]。Wang 等提出采用自适应评估移动策略和基于冯·诺依曼拓扑结构的邻域最优引领策略,两次更新 SSA 算法中追随者的位置,加强个体间的信息交流与协作,加快了算法收敛速度;并采用反向学习策略并以一定变异概率对个体位置进行扰动,促使算法跳出局部最优^[12]。Zhang 等提出引入 Tent 混沌序列生成初始樽海鞘群,增加了樽海鞘个体的多样性;在食物源位置引入疯狂算子,增加了樽海鞘群的多样性;在追随者位置更新公式中引入自适应线性递减惯性权重,使 SSA 算法的全局搜索和局部搜索能力得到了更好的平衡^[13]。

上述改进的 SSA 算法在不同方面、不同程度上弥补了 SSA 算法的缺点,但其仍存在以下问题:文献[3-4,7,10]和文献[11]的改进 SSA 算法分别仅在 10 维和 5 维的基准函数上进行了仿真测试,文献[12]的改进 SSA 算法仅在两个基准函数上进行了仿真测试,这些改进的 SSA 算法的有效性还有待商榷;尽管文献[5,9]和文献[13]分别在最高 30 维和 200 维

的基准函数上进行了仿真测试,但是从其实验结果来看,改进算法的收敛速度、计算精度、全局优化能力和稳定性等仍有待提高。

本文提出将 PSO 算法的随机惯性权重引入 SSA 算法的追随者位置更新公式中,以平衡 SSA 算法的勘探和开发;用差分进化(DE)算法^[14]的变异操作替代 SSA 算法的领导者位置更新操作,以提高 SSA 算法的计算精度和收敛速度。在高维基准函数上的实验结果表明,本文算法非常有效。

2 樽海鞘群算法简介

樽海鞘是一种透明的桶状海洋生物,它通过将海水吸入和排出体外使身体向前游动。海洋生物学家通过研究发现,樽海鞘在导航和觅食的过程中会自发地聚集成链状并快速游动以捕获食物^[1]。正是受这种樽海鞘群体智能行为的启发,Mirjalili 等提出了 SSA 算法。

2.1 樽海鞘群算法的基本原理

樽海鞘链是 SSA 算法的数学模型的基础,它将樽海鞘个体分为两类:唯一的领导者(Leader)和若干追随者(Followers)^[1]。领导者带领追随者在 n 维空间中移动以寻找食物源(即最优解), n 表示给定问题的解包含的变量总数;领导者和追随者的位置均用二维矩阵 \mathbf{x} 表示。领导者按照式(1)更新其位置:

$$x_j^i = \begin{cases} F_j + c_1((ub_j - lb_j)c_2 + lb_j), & \text{if } c_3 \geq 0 \\ F_j - c_1((ub_j - lb_j)c_2 + lb_j), & \text{if } c_3 < 0 \end{cases} \quad (1)$$

其中, x_j^i 表示编号为 1 的樽海鞘(领导者)在第 j 维空间的位置; F_j 是食物源在第 j 维空间的位置; ub_j 和 lb_j 分别表示第 j 维位置取值的上界和下界; c_2 和 c_3 是均匀分布于 $[0,1]$ 的随机数,用于调节领导者位置的变化趋势;系数 c_1 是 SSA 算法最重要的参数,用于平衡算法的勘探和开发,其定义如式(2)所示:

$$c_1 = 2e^{-\frac{4l}{L}} \quad (2)$$

其中, l 表示当前迭代次数, L 为算法最大迭代次数。

在领导者位置更新之后,追随者按照式(3)更新其位置:

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

其中, $2 \leq i \leq NP$, NP 为种群大小; x_j^i 表示编号为 i 的追随者在第 j 维空间的位置。因为编号越大的追随者的位置对应的解越差,所以 x_j^2, \dots, x_j^i 表示第 1 个到第 $i-1$ 个追随者在第 j 维空间的位置。

2.2 樽海鞘群算法的执行过程

综上所述,SSA 算法的执行过程的伪代码表示如下^[1]。

随机初始化每一个樽海鞘的位置;

While (终止条件不满足)

 计算所有樽海鞘位置对应解的适应度;

 食物源 F = 适应度最高的解;

 根据式(2)更新参数 c_1 ;

 For (每个樽海鞘 x_i)

 If ($i = 1$)

 根据式(1)更新领导者位置;

 Else if ($i > 1$ and $i \leq NP$)

根据式(3)更新追随者位置;

End

End

End

将食物源 F 的位置作为最优解输出。

由以上内容可知,SSA 算法的时间复杂度= $O(L \times (n \times NP + Cof \times NP))^{[1]}$,其中 L, n, NP 分别表示算法的最大迭代次数、问题解的变量总数或维数、种群大小或解的个数, Cof 则表示优化问题目标函数的成本。

3 本文改进 SSA 算法的基本策略

SSA 算法的核心是领导者和追随者的位置更新操作。PSO 算法和 DE 算法是目前群体智能算法的优秀代表,其丰富的研究成果为 SSA 算法的改进研究提供了大量有益的参考和启发。本文将 PSO 算法的随机惯性权重和 DE 算法的变异操作分别引入 SSA 算法的式(3)和式(1)中,用于弥补 SSA 算法的缺点。

3.1 PSO 算法随机惯性权重的引入

由 SSA 算法中的式(3)可知,追随者 i 的第 j 维位置 x_j^i 的更新操作直接受前一个追随者 $i-1$ 的第 j 维位置 x_j^{i-1} 的影响。然而,在大多数情况下,追随者 $i-1$ 的位置对应的解通常较差,从而导致 SSA 算法易陷入局部最优,收敛速度慢且计算精度低。

在 PSO 算法中,粒子速度更新决定着粒子位置更新。惯性权重是 PSO 算法粒子速度更新公式中的重要参数,它体现了上一代粒子对当前粒子速度更新的影响力,用于平衡 PSO 算法的勘探与开发^[6]。尽管迄今为止已有大量学者提出了各种 PSO 算法惯性权重控制策略,但文献^[15]的研究表明,在总计 18 种 PSO 算法惯性权重控制策略中,除了随机惯性权重以外,常数项惯性权重优于其他所有的惯性权重控制策略。

常数项惯性权重需要繁琐的算法参数调校,因此本文将 PSO 算法的随机惯性权重引入 SSA 算法追随者位置更新操作中,从而得到追随者位置更新公式:

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

其中, w 为随机惯性权重系数,其取值方法如式(5)^[16]所示:

$$w = 0.5 + \text{Random}(0,1)/2 \quad (5)$$

其中, $\text{Random}(0,1)$ 表示均匀分布于 $(0,1)$ 内的随机数,故 w 的取值范围为 $(0.5,1)$ 。显然,此处引入的随机惯性权重没有改变 SSA 算法的时间复杂度。

3.2 集成 DE 算法的变异操作

根据 SSA 算法的式(1),在领导者位置更新操作中,参与者是食物源且没有追随者,缺乏樽海鞘之间的协作和信息共享,容易导致 SSA 算法过早收敛于较差的局部最优解。

变异操作是 DE 算法的重要组成部分,它一般通过 3 个不同的个体来为种群内每一个个体生成一个新个体,通过比例因子可以调节算法开发和勘探之间的平衡^[14]。目前 DE 算法研究中最典型的变异操作有 5 种,其中与 SSA 算法的式(1)一样包含最优解且形式最简单的变异操作一般简称为

“DE/best/1”^[17]。

综上所述,本文提出用 DE 算法的上述变异操作来替代 SSA 算法领导者的位置更新操作,从而得到如式(6)所示的领导者位置更新操作:

$$x_j^i = F_k + r \times (x_j^m - x_j^n) \quad (6)$$

针对式(1)和式(6),需要说明以下 5 点:

(1)根据 SSA 算法官方网站^[18]提供的 Matlab 版源程序可知,式(1)中的 x_j^1 等同于式(6)中的 x_j^i ,式(1)中的食物源 F_j 实际上是领导者 x_j^1 ;只要樽海鞘 i 的适应度高于领导者的适应度,领导者就被樽海鞘 i 替代,食物源或最优解也随之被领导者或樽海鞘 i 替代。

(2)在式(6)中,随机数 m 和 n 是互异的樽海鞘编号,说明新生成的樽海鞘 i 是由最优解、樽海鞘 m 和樽海鞘 n 经过差分变异操作合成的。

(3)在式(6)中,互异的下标 k 和 j 意味着最优解的异维(跨维)分量参与差分变异操作,有益于增强搜索的广度(开发)和新个体的多样性,避免算法过早陷入局部最优。本文算法程序调试证实了此类异维差分变异操作带来的好处。

(4)式(6)中,系数 r 的作用类似于 DE 算法变异操作中的比例因子,经反复测试发现,其是取值范围在 $(0,1/3)$ 的随机数时效果较好。

(5)与式(1)相比,本文提出的式(6)依然是一种一阶线性关系,没有增加 SSA 算法的时间复杂度。

4 本文提出的改进 SSA 算法的仿真实验及分析

为了检验本文提出的改进 SSA 算法,以求解 8 个常见基准函数^[1,5]的最小值为例进行仿真实验。这些基准函数的基本信息如表 1 所列。

表 1 基准函数

Table 1 Benchmark functions

Name	Number	Search Range	Minimum
Sphere	1	$[-100,100]$	0
Rosenbrock	2	$[-30,30]$	0
Step	3	$[-100,100]$	0
Quartic	4	$[-1.28,1.28]$	0
Schwefel	5	$[-500,500]$	-418.9829D
Ackley	6	$[-32,768,32,768]$	0
Rastrigin	7	$[-5.12,5.12]$	0
Griewank	8	$[-600,600]$	0

表 1 中, D 为基准函数的维数; Sphere, Step, Quartic, Rosenbrock 是单峰函数,其定义域内仅有一个最小值,用于检验算法的收敛速度和计算精度; Ackley, Rastrigin, Schwefel, Griewank 是多峰函数,其定义域内有多个局部最小值和唯一的全局最小值,用于检验算法的全局优化能力。

4.1 本文算法惯性权重的仿真实验及分析

为检验 PSO 算法随机惯性权重对 SSA 算法的改进效果,将 SSA 算法和仅含随机惯性权重的本文提出的 SSA(记为 SSA_{rw})算法各自独立运行 30 遍后取平均值和标准差。两种算法的参数设置均为:种群大小 $NP=30$,最大迭代次数 $MC=1000$,基准函数维数 $D=100$ 。实验结果如表 2 所列。

表2 随机惯性权重对 SSA 算法的改进效果

Table 2 Improvement effect on SSA algorithm by random inertia weight

Function Number	SSA		SSA _{riw}	
	Average	Standard Deviation	Average	Standard Deviation
1	2.50	1.97	2.13×10^{-14}	8.65×10^{-15}
2	1.77×10^3	1.58×10^3	9.79×10	2.44×10^{-2}
3	3.11	2.21	9.56×10^{-2}	4.63×10^{-2}
4	1.38	3.34×10^{-1}	8.67×10^{-5}	8.99×10^{-5}
5	-2.32×10^4	1.43×10^3	-2.35×10^4	1.52×10^3
6	7.49	1.80	1.78×10^{-8}	3.05×10^{-9}
7	1.72×10^2	4.82×10	0	0
8	7.06×10^{-1}	1.88×10^{-1}	2.04×10^{-14}	9.46×10^{-15}

从表2可以看出,除了在多峰函数 Schwefel(序号5)上的结果相差无几以外,在求解单峰和多峰函数优化问题时,本文提出的 SSA_{riw} 算法求解结果的平均值和标准差,明显小于 SSA 算法的平均值和标准差,说明引入 PSO 算法的随机惯性权重后,显著地提高了 SSA 算法的收敛速度、计算精度、全局优化能力和稳定性。

4.2 本文算法差分变异操作的仿真实验及分析

为了验证集成 DE 算法的差分变异操作对 SSA 算法的改进效果,本实验将 SSA 算法和仅集成了差分变异操作的本文算法(记为 SSA_{dmo})各自独立运行 30 遍后取平均值和标准差。两种算法的参数设置与上文相同,实验结果如表 3 所列。

表3 差分变异操作对 SSA 算法的改进效果

Table 3 Improvement effect on SSA algorithm

Function Number	SSA		SSA _{riw}	
	Average	Standard Deviation	Average	Standard Deviation
1	2.50	1.97	1.17×10^{-150}	2.85×10^{-150}
2	1.77×10^3	1.58×10^3	7.84×10	3.99×10
3	3.11	2.21	0	0
4	1.38	3.34×10^{-1}	9.76×10^{-3}	1.76×10^{-2}
5	-2.32×10^4	1.43×10^3	-4.19×10^4	7.40×10^{-12}
6	7.49	1.80	2.38×10^{-1}	9.07×10^{-1}
7	1.72×10^2	4.82×10	3.79×10^{-15}	2.08×10^{-14}
8	7.06×10^{-1}	1.88×10^{-1}	0	0

从表3可以看出,无论是单峰还是多峰函数优化问题,本文提出的 SSA_{dmo} 算法求解结果的平均值和标准差明显小于 SSA 算法的平均值和标准差,说明集成 DE 算法的差分变异操作后,显著地提高了 SSA 算法的收敛速度、计算精度、全局优化能力和稳定性。

4.3 各改进 SSA 算法实验结果的对比及分析

为了便于全面考验本文提出的 SSA 算法的改进效果,下文将同时包含随机惯性权重和差分变异操作的本文算法命名为 iSSA,并与文献[5]提出的 ESSA 算法和文献[13]提出的 CASSA 算法分别进行对比实验。用于基准测试的单峰和多峰函数的编号及其定义请见文献[1,5]。文献[5]中的 ESSA 算法的实验参数设置为:种群大小 $NP=30$,最大迭代次数 $MC=500$,算法运行次数 $RT=30$,单模和多模基准函数的维数 $D=30$ 。文献[13]中的 CASSA 算法的实验参数设置为:种群大小 $NP=40$,最大迭代次数 $MC=1000$,算法运行次数 $RT=50$,单模和多模基准函数的维数 $D=60,120,200$ 等。为公平起见,本文算法和对比算法的参数设置完全相同;ESSA

算法和 CASSA 算法的实验结果来自文献[5,13]提供的原始数据。本文算法和 ESSA 算法的实验结果如表 4 所列。

表4 本文算法和 ESSA 算法的实验结果

Table 4 Experimental results of proposed iSSA algorithm and ESSA algorithm

Function Number	SSA		SSA _{riw}	
	Average	Standard Deviation	Average	Standard Deviation
F_1	5.12×10^{-38}	2.44×10^{-37}	3.09×10^{-114}	4.84×10^{-114}
F_2	2.01×10^{-25}	7.74×10^{-25}	4.44×10^{-58}	4.14×10^{-58}
F_3	3.36×10^{-35}	1.79×10^{-34}	1.17×10^{-150}	3.95×10^{-150}
F_4	4.25×10^{-18}	2.1×10^{-17}	2.65×10^{-58}	2.06×10^{-58}
F_5	28.8803	0.0298	28.7204	0.0014
F_6	3.3387	0.8243	1.76×10^{-6}	9.40×10^{-7}
F_7	8.82×10^{-5}	6.94×10^{-5}	8.56×10^{-5}	1.00×10^{-4}
F_8	-4830	709.30	-2569.34	0.06287
F_9	0	0	0	0
F_{10}	8.88×10^{-16}	0	8.88×10^{-16}	0
F_{11}	0	0	0	0
F_{12}	0.2724	0.1555	2.94×10^{-7}	2.28×10^{-7}
F_{13}	2.8525	0.3519	3.07×10^{-6}	1.74×10^{-6}

由表4可知,在求解单峰函数 F_5, F_7 和多峰函数 F_9, F_{10}, F_{11} 时,本文算法与 ESSA 算法的表现同样出色,除此以外本文算法结果的平均值和标准差明显小于 ESSA 算法,说明在收敛速度、计算精度、全局优化能力和稳定性方面,本文算法优于 ESSA 算法。

需要注意,文献[13]中基准函数及其编号与文献[1,5]不同;为便于比较和避免混淆,下文以文献[13]描述的基准函数及其编号为准。文献[13]测试的高维单峰基准函数有 Sphere ($F_3, 60$ 维)、Quartic ($F_4, 60$ 维)、Schwefel 2.22 ($F_5, 120$ 维)、Schwefel 2.21 ($F_6, 200$ 维),高维多峰基准函数有 Ackley ($F_{10}, 60$ 维)、Griewank ($F_{11}, 120$ 维)、Penalized ($F_{12}, 200$ 维)。本文算法和 CASSA 算法的实验结果如表 5 所列。

表5 本文算法和 CASSA 算法的实验结果

Table 5 Experimental results of proposed iSSA algorithm and CASSA algorithm

Function	Algorithm	Best	Average	Standard Deviation
F_3	CASSA	2.07×10^{-95}	2.92×10^{-95}	5.25×10^{-96}
	iSSA	4.20×10^{-235}	6.94×10^{-233}	0
F_4	CASSA	3.71×10^{-8}	1.24×10^{-5}	1.04×10^{-5}
	iSSA	1.26×10^{-7}	2.36×10^{-5}	2.42×10^{-5}
F_5	CASSA	1.07×10^{-47}	1.21×10^{-47}	8.03×10^{-49}
	iSSA	1.67×10^{-118}	1.46×10^{-117}	4.99×10^{-117}
F_6	CASSA	1.71×10^{-48}	2.10×10^{-48}	2.66×10^{-49}
	iSSA	1.97×10^{-119}	8.56×10^{-118}	8.87×10^{-118}
F_{10}	CASSA	8.88×10^{-16}	8.88×10^{-16}	0
	iSSA	8.88×10^{-16}	8.88×10^{-16}	0
F_{11}	CASSA	0	0	0
	iSSA	0	0	0
F_{12}	CASSA	3.50×10^{-2}	5.75×10^{-2}	1.39×10^{-2}
	iSSA	9.53×10^{-9}	4.96×10^{-8}	3.63×10^{-8}

观察表5的结果可知,在测试单峰基准函数时,除了 Quartic (F_4) 以外,本文算法结果的最优值、平均值和标准差明显小于 CASSA 算法的结果,说明本文算法的收敛速度、计算精度和稳定性优于 CASSA 算法;在测试多峰基准函数 Ackley (F_{10}) 和 Griewank (F_{11}) 时,本文算法与 CASSA 算法的

表现同样优秀;在测试多峰基准函数 Penalized 时,本文算法结果的最优值、平均值和标准差明显小于 CASSA 算法,说明在全局优化能力和稳定性方面,本文算法优于 CASSA 算法。

结束语 为弥补 SSA 算法收敛慢、计算精度低和全局优化能力弱等缺点,本文参考借鉴 PSO 和 DE 算法的研究成果,将 PSO 算法的随机惯性权重系数引入 SSA 算法的追随者位置更新公式中,并用 DE 算法的差分变异操作替代 SSA 算法的领导者位置更新操作,用于提高 SSA 算法的收敛速度、计算精度和全局优化能力等。高维基准函数测试结果对比及分析表明,本文算法改进效果非常明显。作为群体智能算法家族的新成员,SSA 算法还亟待有关学者开展系统而深入的理论及应用研究,因此本文提出的 SSA 算法研究还可以扩展到解决各种工程优化领域中的约束优化、组合优化、多目标优化等问题领域,以解决现实中的复杂优化问题。

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: 163-191.
- [2] FARIS H, MIRJALILI S, ALJARAHA I, et al. *Nature-Inspired Optimizers* [M]. Switzerland: Springer International Publishing, 2020: 185-199.
- [3] SAYED G I, KHORIBA G, HAGGAG M H. A novel chaotic salp swarm algorithm for global optimization and feature selection [J]. *Applied Intelligence*, 2018, 48(10): 3462-3481.
- [4] WANG D, ZHOU Y, JIANG S, et al. A Simplex Method-Based Salp Swarm Algorithm for Numerical and Engineering Optimization [C]// *International Conference on Intelligent Information Processing*. Cham: Springer, 2018: 150-159.
- [5] QAIS M H, HASANIEN H M, ALGHUWAINEM S. Enhanced salp swarm algorithm: Application to variable speed wind generators [J]. *Engineering Applications of Artificial Intelligence*, 2019, 80: 82-96.
- [6] POLI R, KENNEDY J, BLACKWELL T. Particle swarm optimization. An overview [J]. *Swarm Intelligence*, 2007, 1(1): 33-57.
- [7] WU J, NAN R, CHEN L. Improved salp swarm algorithm based on weight factor and adaptive mutation [J]. *Journal of Experimental & Theoretical Artificial Intelligence*, 2019, 31(3): 493-515.
- [8] MIRJALILI S. SCA: A Sine Cosine Algorithm for solving optimization problems [J]. *Knowledge-Based Systems*, 2016, 96: 120-133.
- [9] SINGH N, SON L H, CHICLANA F, et al. A new fusion of salp swarm with sine cosine for optimization of non-linear functions [J]. *Engineering with Computers*, 2020, 36: 185-212.
- [10] MAJHI S K, MISHRA A, PRADHAN R. A chaotic salp swarm algorithm based on quadratic integrate and fire neural model for function optimization [J]. *Progress in Artificial Intelligence*, 2019: 1-16.
- [11] YANG B, ZHONG L, ZHANG X, et al. Novel bio-inspired memetic salp swarm algorithm and application to MPPT for PV systems considering partial shading condition [J]. *Journal of Cleaner Production*, 2019, 215: 1203-1222.
- [12] WANG M Q, WANG Y, JI Z C. Permanent Magnet Synchronous Motor Multi-parameter Identification Based on Improved Salp Swarm Algorithm [J]. *Journal of System Simulation*, 2018, 30(11): 4284-4297.
- [13] ZHANG D M, CHEN Z Y, XIN Z Y, et al. Salp Swarm Algorithm Based on Crazyness and Adaptive [J]. *Control and Decision*, 2020, 35(9): 2112-2120.
- [14] PRICE K, STORN R M, LAMPINEN J A. *Differential Evolution: A Practical Approach to Global Optimization* [M]. Berlin: Springer-Verlag, 2005: 37-130.
- [15] HARRISON K R, ENGELBRECHT A P, OMBUKI-BERMAN B M. Inertia weight control strategies for particle swarm optimization [J]. *Swarm Intelligence*, 2016, 10(4): 267-305.
- [16] EBERHART R C, SHI Y. Tracking and optimizing dynamic systems with particle swarms [C]// *Proceedings of the 2001 Congress on Evolutionary Computation*. IEEE, 2001: 94-100.
- [17] DAS S, MULLICK S S, SUGANTHAN P N. Recent advances in differential evolution-An updated survey [J]. *Swarm and Evolutionary Computation*, 2016, 27: 1-30.
- [18] MIRJALILI S. Salp Swarm Algorithm[EB/OL]. <http://www.alimirjalili.com/SSA.html>.



ZHANG Zhi-qiang, born in 1973, Ph.D., lecturer, is a member of China Computer Federation. His main research interests include computational intelligence and optimization technology.