

多策略自适应共生生物搜索算法

周 虎, 赵 辉*, 李牧东, 蔡亚伟

(空军工程大学航空航天工程学院,西安,710038)

摘要 针对共生生物搜索算法搜索速度慢、收敛精度不高且易早熟的缺点,提出了一种多策略自适应改进算法。首先,根据适应度将种群分为 3 个群体,每个群体采用不同的搜索策略以实现不同功能。其次,提出了一种基于实时信息反馈的混合搜索策略,使其搜索策略实现自适应调整。最后,对超边界个体进行变异操作,以增加种群多样性。对 14 个标准测试函数的仿真测试表明改进算法全局优化能力更强,具有更好的搜索速度和收敛精度。

关键词 共生生物搜索算法;多策略;自适应;全局优化

DOI 10.3969/j.issn.1009-3516.2016.04.019

中图分类号 TP301.6 **文献标志码** A **文章编号** 1009-3516(2016)04-0101-06

Multi-strategy Adaptive Symbiotic Organisms Search Algorithm

ZHOU Hu, ZHAO Hui*, LI Mudong, CAI Yawei

(Aeronautics and Astronautics Engineering College, Air Force Engineering University, Xi'an 710038, China)

Abstract: Aimed at the problems that Symbiotic Organisms Search (SOS) algorithm is poor in convergence, low in searching precision and ease of premature convergence, a multi-strategy adaptive algorithm is proposed. Firstly, according to the fitness, populations can be divided into three groups, and each group with different strategies can achieve different functions. Secondly, a hybrid search strategy based on adaptive scaling factor can make its search strategy realization of the adaptive adjustment. Finally, in order to maintain the population diversity, a mutation is utilized when the individual beyond the boundary. Experiments are conducted on the 14 benchmark functions, and the results show that the MSASOS algorithm improves obviously the performance in convergence speed, precision and global optimization.

Key words: symbiotic organisms search (SOS); multi-strategy; adaptive adjustment; global optimization

元启发式优化算法如遗传算法^[1-2],蚁群算法^[3-4]和粒子群算法^[5-6]等,通常从一组初始变量开始,然后运行过程,直到获得全局最优解^[7],以其结构简单、求解效率高特点得到了前所未有的发展,并在网络优化、智能识别、工程应用以及多目标优化等领域中广泛应用^[8]。

共生生物搜索算法^[9]记作 SOS,由 Min-Yuan Cheng 和 Doddy Prayogo 在 2014 提出,具有易操作、控制参数少、稳定性较好的优点,但同时也存在早熟、收敛速度慢、后期搜索迟滞等问题,这些缺陷,也是许多元启发式优化算法的通病,对此,学者们提出了大量的改进算法以及新的优化算法。文献^[10]

收稿日期:2015-10-19

基金项目:国家自然科学基金(71501184)

作者简介:周 虎(1991—),男,四川遂宁人,硕士生,主要从事无人飞行器作战系统与技术研究.E-mail:382207064@qq.com

* **通信作者:**赵 辉(1973—),男,河北任丘人,教授,主要从事无人飞行器作战系统与技术研究.E-mail:kgy_zhao@163.com

引用格式:周虎,赵辉,李牧东,等.多策略自适应共生生物搜索算法[J].空军工程大学学报:自然科学版,2016,17(4):101-106. ZHOU Hu, ZHAO Hui, LI Mudong, et al. Multi-strategy Adaptive Symbiotic Organisms Search Algorithm[J]. Journal of Air Force Engineering University: Natural Science Edition, 2016, 17(4): 101-106.

在差分进化算法中引入3种变异策略和适当的控制参数来提高算法性能;文献[11]利用PSO算法搜索过程中粒子的较优解对搜索方程加以改进,提高了算法的收敛速度;文献[12]在DE算法的基础上通过引入自适应进化策略提高了算法的性能;文献[13~15]通过模拟动物的觅食行为分别提出了布谷鸟搜索算法(Cuckoo Search, CK)、蜂群算法(Artificial Bee Colony, ABC)以及灰狼算法(Grey Wolf Optimize, GWO);此外,许多算法寻优已经不再局限于单一启发原则,而是将多种启发式算法融合,文献[16]基于进化算法和模拟退火的优点,提出了一种混合启发式算法(Hybrid Metaheuristics, HMs)。

1 标准共生生物搜索算法

SOS算法模拟生物共生关系,通过种群内个体间的合作与竞争产生群体智能指导优化搜索,并将共生关系概括为“互利”“共栖”“寄生”3种:“互利”使得双方相互受益;“共栖”使得一方受益,另一方不受影响;“寄生”却使一方受益,另一方受害。依此建立主要搜索原则,主要步骤如下:

步骤1 初始化种群:首先按式(1):

$$\mathbf{X}_i = L_b + \text{rand}(1, D)(U_b - L_b) \quad (1)$$

随机生成 N 个“生物”个体,每个“生物”为一个初始解。式中 \mathbf{X}_i 代表生态系统中第 i ($i=1, 2, \dots, N$)个“生物”, D 为解的维数, $\text{rand}(1, D)$ 为 $1 \times D$ 维的缩放因子向量, U_b 、 L_b 是搜索空间的上界和下界。

步骤2 互利:在此阶段,随机从种群中选择一个“生物” \mathbf{X}_j 与 \mathbf{X}_i 相互作用产生互利,使得各自向最优解学习。 \mathbf{X}_i 与 \mathbf{X}_j 按式(2)生成新的解 $\mathbf{X}_{i\text{new}}$ 、 $\mathbf{X}_{j\text{new}}$ 。

$$\begin{aligned} \mathbf{X}_{i\text{new}} &= \mathbf{X}_i + \text{rand}(0, 1)(\mathbf{X}_{\text{best}} - \mathbf{M}_v B_1) \\ \mathbf{X}_{j\text{new}} &= \mathbf{X}_j + \text{rand}(0, 1)(\mathbf{X}_{\text{best}} - \mathbf{M}_v B_2) \end{aligned} \quad (2)$$

式中: $i, j \in \{1, 2, \dots, N\}$, $i \neq j$, 缩放因子 $\text{rand}(0, 1)$ 为 $[0, 1]$ 之间的随机数。 \mathbf{X}_{best} 为当前最优个体, B_1 、 B_2 是 $\{1, 2\}$ 中的随机数,表示获益因子,互利共生的生物相互间受益并不相等,一方可能受益较多,而另一方受益较少,因此,获益因子大表示获益较多,反之则较少。 \mathbf{M}_v 叫做“互利向量”,表示2个“生物”间的关系特征。

步骤3 共栖:与“互利”阶段相似,从种群中随机选择一个“生物” \mathbf{X}_j 与 \mathbf{X}_i 相互作用,作用的结果使 \mathbf{X}_i 从中受益,而 \mathbf{X}_j 既不受益也不受害, \mathbf{X}_i 通过式(3)进行更新。

$$\mathbf{X}_{i\text{new}} = \mathbf{X}_i + \text{rand}(-1, 1)(\mathbf{X}_{\text{best}} - \mathbf{X}_j) \quad (3)$$

步骤4 寄生:首先对 \mathbf{X}_i 中某个或多个元素进行随机修改,得到一个新个体,叫做“寄生向量”,记作 \mathbf{X}_{pv} ;然后从种群中随机选出一个个体 \mathbf{X}_j ($j \neq i$)作为 \mathbf{X}_{pv} 的“宿主”,计算“寄生向量”和“宿主”的适应度值并进行比较。若“寄生向量”的适应度值更好,那么生物 \mathbf{X}_j 将会被其取代,否则 \mathbf{X}_j 将具有免疫性,被保留下来。

2 多策略自适应共生生物搜索算法

一个好的搜索算法应在初始阶段保持种群多样性,注重算法的探索能力,进行全局搜索,而在算法后期侧重开发能力,进行局部搜索,以提高算法的收敛速率和精度^[17]。如何权衡算法的“探索与开发”能力决定了算法的优化性能,文献[18]通过概率选择两种搜索方程来达到一定程度上平衡ABC算法开发与探索能力的目的。SOS算法的搜索方程因其缩放因子的随机性,因此具有较强的探索能力;但其搜索方程固定,始终以当前个体为基准,使得开发能力差,表现为早熟、收敛精度不够高、收敛速度慢。

由达尔文的生物进化论可知,生物的进化具有多样性与渐进性,并不能一开始就朝着某一方向跳跃式的进化。受此启发,针对SOS算法的缺陷,提出一种多策略自适应算法MSASOS。

2.1 多策略“互利”搜索过程

不同策略有适合于各自求解的优化问题领域,某一策略并不能适应于求解所有优化问题。本文根据适应度将种群中的个体分为高等、低等和一般3类,由个体适应度与最优值间的距离 d 决定。规定一般生物的适应度与最优值的最大间距为 d_{max} ,最小间距为 d_{min} ,因此,当 $d < d_{\text{min}}$ 时,个体为高等生物;当 $d > d_{\text{max}}$ 时,个体为低等生物;其它的为一般生物。不同群体采用不同的搜索策略,由劣到优循序渐进。3种策略具体如下所示:

1)低等生物。初始搜索阶段,为保证种群多样性,生物间的“相互作用”不能在立即就朝着当前最优解进行,而应具有随机性。因此该群体的搜索策略是以自身为基准个体,加上一个随机扰动。该策略能够较好的游历整个种群,搜索无偏性,具有较强的全局搜索能力。具体搜索算子如下所示:

$$\begin{aligned} \mathbf{X}_{i\text{new}} &= \mathbf{X}_i + \text{rand}(-1, 1)(\mathbf{X}_j - \mathbf{M}_v B_1) \\ \mathbf{X}_{j\text{new}} &= \mathbf{X}_j + \text{rand}(-1, 1)(\mathbf{X}_j - \mathbf{M}_v B_2) \end{aligned} \quad (4)$$

式中: $i, j \in \{1, 2, \dots, N\}$, $i \neq j$ 。

2)一般生物。该群体起到一个过渡作用,促使算法由全局搜索向局部搜索转化。考虑到 best (最

好个体)引导策略可能引起算法早熟,遂采用 p best(适应度排序靠前的第 P 个个体)引导策略以期加速收敛的同时维护种群多样性。该策略以适应度排序靠前的第 p 个个体 $\mathbf{X}_{p\text{ best}}$ 为基准个体,将问题解朝全局最优方向引导,进入局部搜索阶段。搜索算子为:

$$\begin{aligned} \mathbf{X}_{i\text{ new}} &= \mathbf{X}_{p\text{ best}} + \text{rand}(-1,1)(\mathbf{X}_{p\text{ best}} - \mathbf{M}_V B_1) \\ \mathbf{X}_{j\text{ new}} &= \mathbf{X}_{p\text{ best}} + \text{rand}(-1,1)(\mathbf{X}_{p\text{ best}} - \mathbf{M}_V B_2) \end{aligned} \quad (5)$$

3)高等生物。该群体的搜索策略以当前种群中最佳个体为基准个体,在其领域内进行搜索,具有很好的局部搜索能力,提高算法的收敛速度和精度。搜索算子为:

$$\begin{aligned} \mathbf{X}_{i\text{ new}} &= \mathbf{X}_{\text{ best}} + \text{rand}(-1,1)(\mathbf{X}_{\text{ best}} - \mathbf{M}_V B_1) \\ \mathbf{X}_{j\text{ new}} &= \mathbf{X}_{\text{ best}} + \text{rand}(-1,1)(\mathbf{X}_{\text{ best}} - \mathbf{M}_V B_2) \end{aligned} \quad (6)$$

式中: $\text{best} \in \{1, 2, \dots, N\}$ 为当前种群中最好个体的整数下标。

2.2 自适应“共栖”搜索过程

SOS算法始终采用的都是“current-to-best”的搜索策略,其搜索方程因缩放因子的随机性,限制了全局最优值的引导作用,导致收敛精度不够高和收敛时间偏大。为此,本文提出了一种具有全局最优引导的自适应混合搜索策略,具体形式如下:

$$\begin{aligned} \mathbf{X}_{i\text{ new}} &= \mathbf{X}_i + (1-\omega)(\mathbf{X}_i - \mathbf{X}_j) + \omega(\mathbf{X}_{\text{ best}} - \mathbf{X}_j) \quad (7) \\ \omega &= \omega_{\text{ min}} + (\omega_{\text{ max}} - \omega_{\text{ min}}) \frac{G}{G_{\text{ max}}} \quad (8) \end{aligned}$$

式中: $\omega \in [0, 1]$, 为缩放因子, $\omega_{\text{ max}}$ 、 $\omega_{\text{ min}}$ 分别为缩放因子的最大值和最小值; $G_{\text{ max}}$ 为最大循环次数, G 为当前循环次数。

该策略通过增加一项差分扰动因子来保持个体间的差异性,同时引入自适应缩放因子,循环初期 ω 较小,算法更加注重全局搜索,减少向搜索空间单一点靠拢的趋势,避免算法陷入局部极小;随着算法的进行, ω 逐渐增大,当前最优个体的引导作用加强,引领算法前进,搜索过程不再只是一种完全的随机搜索,而是更有目的性和方向性。

2.3 改进的超边界处理方式

在搜索过程中,难免会出现超边界的情况。SOS算法直接取临近的边界值,这样一定程度上破坏了种群多样性。本文算法对超边界个体进行一次变异,这样避免了个体超出边界,也保证了种群多样性。具体过程如下:

$$\begin{aligned} \mathbf{X}_i &= \mathbf{X}_{\text{ max}} + \text{rand}(0,1)(\mathbf{X}_{\text{ max}} - \mathbf{X}_i), \mathbf{X}_i > \mathbf{X}_{\text{ max}} \\ \mathbf{X}_i &= \mathbf{X}_{\text{ min}} + \text{rand}(0,1)(\mathbf{X}_{\text{ min}} - \mathbf{X}_i), \mathbf{X}_i < \mathbf{X}_{\text{ min}} \end{aligned} \quad (9)$$

式中: $\mathbf{X}_{\text{ max}}$ 为解空间最大边界值; $\mathbf{X}_{\text{ min}}$ 为解空间最小边界值。

2.4 MSASOS 算法实现步骤

Step 1 设置种群规模参数 N 、问题维数 D 、最大循环次数 $G_{\text{ max}}$ 、较优个体 $\mathbf{X}_{p\text{ best}}$ 的选择参数 p 和缩放因子参数 $\omega_{\text{ max}}$ 、 $\omega_{\text{ min}}$, 针对不同问题设置评价距离 $d_{\text{ max}}$ 、 $d_{\text{ min}}$ 和终止条件;按式(1)生成初始种群;

Step 2 计算种群中个体的适应度,根据适应度确定当前最优解 $\mathbf{X}_{\text{ best}}$ 和排序第 p 的较优解;

Step 3 设置 $i = 1$;

Step 4 随机个体 $\mathbf{X}_j (j \neq i)$ 与 \mathbf{X}_i 按照适应度值选择“互利”搜索策略,按式(4~6)进行更新操作,生成新个体,选择较优个体进入下一步;

Step 5 按式(7)进行“共栖”操作, ω 由式(8)确定;

Step 6 将 \mathbf{X}_i 的变异得到的“寄生向量” \mathbf{X}_{pv} 与随机个体 $\mathbf{X}_j (j \neq i)$ 实施评价选择,更新 \mathbf{X}_j ;

Step 7 $i = i + 1$;如果所有的目标个体都已完成更新操作,即当 $i = N$,则进行下一步,否则返回 Step2;

Step 8 当达到终止条件时,算法停止,否则返回 Step2,开始下一次迭代。

3 仿真实验

3.1 实验设置

为了验证本文提出的 MSASOS 算法的性能,选取 14 个常用基准函数作为测试集,并与 SOS、PSO、GWO 及 ABC 算法进行比较。表 1 给出了 14 个基准函数的维数、搜索范围以及最优值。其中 $f_1 \sim f_6$ 、 f_{10} 、 f_{14} 是单峰函数, $f_7 \sim f_9$ 、 f_{11} 、 f_{12} 、 f_{13} 是多峰函数。仿真软件为 Matlab R2013a,在仿真中,设置 MSASOS、SOS 算法种群个数为 30,由于 MSASOS 和 SOS 算法循环一次每个个体的评价次数为其它 3 种算法的 4 倍,因此,为保证评价次数一样,将 PSO、GWO 及 ABC 算法的种群个数设置为 120。具体参数设置如下:

MSASOS: $N = 30$, $\omega_{\text{ max}} = 0.9$, $\omega_{\text{ min}} = 0.1$, 评价距离 $d_{\text{ max}} = 1e - 1$, $d_{\text{ min}} = 1e - 6$, 取 $p = 12$ 。

SOS: $N = 30$, 其余参数见文献[9]。

PSO: $N = 120$, 权重因子 $\omega = 0.6$, $c_1 = c_2 = 2$, 其余参数见文献[7]。

ABC: $N = 120$, $\text{limit} = 100$, 其余参数见文献[14]。

GWO: $N = 120$, 其余参数见文献[15]。

设置最大循环次数为 2 500,实验次数为 30 次,算法达到最大循环次数或得到最优值时停止。

表 1 基准函数

Tab.1 Benchmark functions

序号	Function	D	Range	optimal
f_1	Zakharov	50	$[-5, 10]$	0
f_2	Sphere	50	$[-10, 10]$	0
f_3	SumSquares	50	$[-100, 100]$	0
f_4	Quartic	50	$[-1.28, 1.28]$	0
f_5	Schwefel 2.22	50	$[-10, 10]$	0
f_6	Schwefel 1.2	50	$[-100, 100]$	0
f_7	Griewank	50	$[-600, 600]$	0
f_8	Rastrigin	50	$[-5.12, 5.12]$	0
f_9	Rosenbrock	30	$[-30, 30]$	0
f_{10}	Colville	4	$[-10, 10]$	0
f_{11}	Booth	2	$[-10, 10]$	0
f_{12}	Schaffer	2	$[-100, 100]$	0
f_{13}	Boachevsky3	2	$[-100, 100]$	0
f_{14}	Easom	2	$[-100, 100]$	-1

3.2 实验结果与分析

图 1 为 5 种算法对 Sphere、Rosenbrock、Rastrigin 和 Griewank 4 个具有代表性的基准函数的收敛过程对比图。表 2 为测试结果,表中 Best、Worst、Mean、Std 和 Time 分别为算法独立实验 30 次的最优值、最差值、平均值、标准差和平均时间, Best、Worst 反映了解的质量, Std 反映了算法的稳定性, Mean 反映了在给定测试次数下算法所能达到的精度, Time 反映了算法的收敛速度。

结果表明 MSASOS 算法无论在收敛速度、搜索精度还是稳定性方面均优于 SOS 和其它 3 种算法。其中,对于函数 $f_1 \sim f_8, f_{11} \sim f_{14}$, MSASOS 算法在都能在短时间内收敛到最优值;对于函数 f_9, f_{10} , 虽都未在指定条件下收敛到最优值, MSASOS 算法具有明显较高的收敛精度。对于函数 $f_2 \sim f_8, f_{11} \sim f_{14}$, 虽然 SOS、PSO、ABC 和 GWO 算法也能够将其中某些函数收敛至全局最优值,但 MSASOS 算法的收敛速度明显更快;从图 1 中可以看出, MSASOS 算法具有明显更高的收敛速度和搜索精度,在较短时间内便能达到最优值。

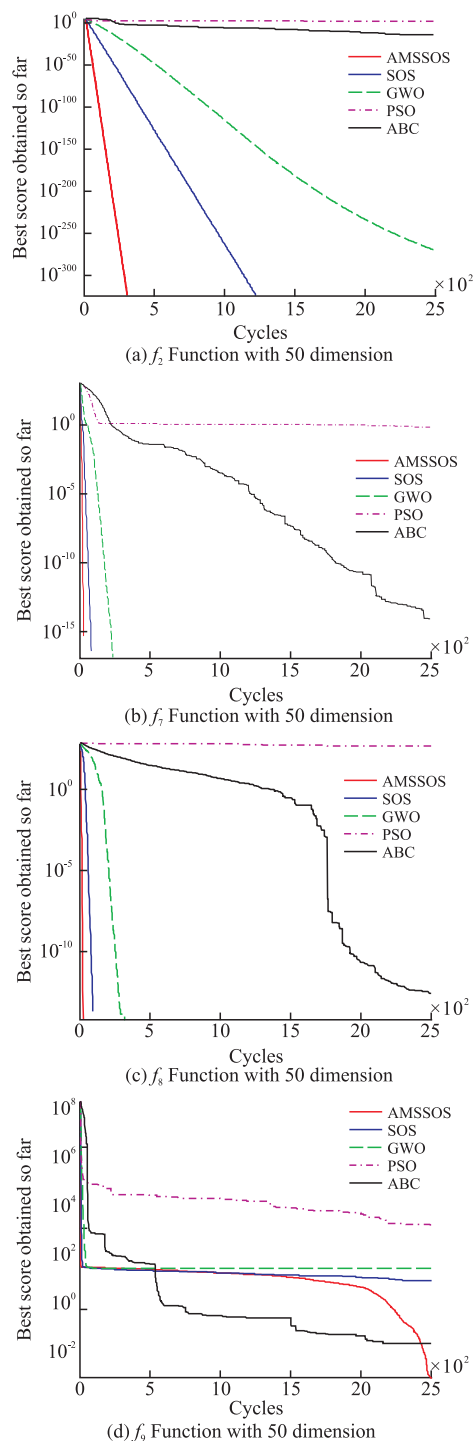


图 1 5 种算法对 4 个基准函数的收敛过程
Fig.1 Evolution process of five algorithm for four benchmark functions

表 2 与 SOS、PSO、GWO 和 ABC 算法的性能对比 ($f_1 - f_7$)
 Tab.2 Performance comparison with SOS, PSO, GWO and ABC ($f_1 - f_7$)

f	Method	Best	Worst	Mean	Std	Time/s
f_1	MSASOS	0	0	0	0	1.769 9
	SOS	9.933 8e-88	8.115 9e-87	4.554 7e-87	5.036 4e-87	5.945 6
	GWO	3.519 0e-101	9.107 6e-101	6.313 3e-101	10.370 4	10.370 4
	PSO	184.633 1	243.489 6	214.061 3	41.617 8	11.909 2
	ABC	516.305 5	537.001 2	526.653 4	14.634 1	16.230 6
f_2	MSASOS	0	0	0	0	0.772 3
	SOS	0	0	0	0	3.092 5
	GWO	5.230 3e-273	3.788 0e-269	4.938 5e-270	0	7.578 2
	PSO	23.620 0	49.417 6	31.624 7	7.013 2	9.433 9
	ABC	1.416 3e-15	2.094 9e-15	1.796 4e-15	2.525 6e-16	14.454 1
f_3	MSASOS	0	0	0	0	0.730 2
	SOS	0	0	0	0	2.922 1
	GWO	3.775 2e-274	7.864 7e-270	1.216 8e-270	0	7.199 6
	PSO	465.817 9	1.281 3e+03	705.539 8	232.254 6	8.735 0
	ABC	9.341 9e-16	1.615 2e-15	1.249 2e-15	1.809 3e-16	13.436 7
f_4	MSASOS	0	0	0	0	0.557 0
	SOS	0	0	0	0	2.260 8
	GWO	0	0	0	0	7.216 3
	PSO	148.699 8	290.913 1	208.883 6	38.308 6	11.525 3
	ABC	2.644 7e-16	4.206 1e-16	3.056 9e-16	4.396 1e-17	19.504 0
f_5	MSASOS	0	0	0	0	1.736 9
	SOS	0	0	0	0	8.450 0
	GWO	1.241 2e-159	3.220 0e-157	5.448 5e-158	9.616 4e-158	14.647 7
	PSO	29.254 5	29.254 5	24.865 2	3.055 1	17.188 9
	ABC	2.033 5e-10	3.937 6e-10	2.947 1e-10	5.577 3e-11	23.319 1
f_6	MSASOS	0	0	0	0	1.673 4
	SOS	0	0	0	0	6.737 1
	GWO	1.087 8e-270	1.017 0e-268	5.139 2e-269	0	16.379 0
	PSO	498.314 0	934.712 2	716.513 1	308.580 1	23.141 4
	ABC	7.595 4e-15	1.076 4e-14	9.179 9e-15	2.240 8e-15	33.379 1
f_7	MSASOS	0	0	0	0	0.086 3
	SOS	0	0	0	0	0.243 6
	GWO	0	0	0	0	1.008 9
	PSO	0.705 6	0.796 0	0.750 8	0.063 9	4.001 0
	ABC	6.661 3e-16	1.443 3e-15	1.054 7e-15	5.495 3e-16	9.582 5
f_8	MSASOS	0	0	0	0	0.067 7
	SOS	0	0	0	0	0.215 4
	GWO	0	0	0	0	0.658 5
	PSO	380.512 0	453.186 4	416.849 2	51.388 6	2.622 3
	ABC	1.136 9e-13	1.705 3e-13	1.421 1e-13	4.019 4e-14	7.167 3
f_9	MSASOS	0.001 4	0.006 3	0.003 9	0.003 5	5.838 3
	SOS	8.568 9	8.703 8	8.636 4	0.095 4	5.732 1
	GWO	25.227 8	27.110 0	26.168 9	1.330 9	9.094 8
	PSO	1.151 5e+03	1.878 8e+03	1.515 2e+03	514.284 7	10.209 1
	ABC	0.048 4	0.072 7	0.060 6	0.017 2	14.515 0
f_9	MSASOS	0	1.774 9e-31	8.874 7e-32	1.255 1e-31	3.254 6
	SOS	0	2.374 0e-30	1.187 0e-30	1.678 7e-30	3.762 0
	GWO	0.030 6	1.314 6	2.501 3e-04	0.908 0	7.976 7
	PSO	1.595 3e-04	3.407 2e-04	2.501 3e-04	1.281 2e-04	9.326 9
	ABC	0.220 4	0.336 9	0.278 7	0.082 3	13.365 5
f_{11}	MSASOS	0	0	0	0	0.357 2
	SOS	0	0	0	0	0.467 0
	GWO	2.607 9e-05	0.507 3	0.456 5	0.160 4	1.917 9
	PSO	0	3.155 4e-28	3.226 4e-29	9.954 3e-29	2.250 2
	ABC	8.127 7e-18	6.176 0e-07	6.176 0e-08	1.953 0e-07	6.277 7
f_{12}	MSASOS	0	0	0	0	0.651 8
	SOS	0	0	0	0	0.325 9
	GWO	0	0	0	0	0.363 9
	PSO	0	0	0	0	1.010 8
	ABC	0	0	0	0	1.613 4
f_{13}	MSASOS	0	0	0	0	0.045 7
	SOS	0	0	0	0	0.100 3
	GWO	0	0	0	0	0.128 2
	PSO	0	0	0	0	0.388 3
	ABC	0	0	0	0	2.680 7
f_{14}	MSASOS	-1	-1	-1	-1	0.311 9
	SOS	-1	-1	-1	-1	0.156 0
	GWO	-1	-1	-1	-1	1.638 4
	PSO	-1	-1	-1	-1	1.912 1
	ABC	-1	-1	-1	-1	3.034 6

4 结语

针对 SOS 算法的收敛速度慢、后期搜索迟滞且易早熟等问题,本文提出了具有多策略的 MSASOS

改进算法。该算法首先利用个体适应度将种群分为多个群体,每个群体使用不同的策略以实现不同功能,然后根据目标函数值实时反馈的信息,引入一种混合搜索策略,在提高了算法局部开发能力的同时,有效地保持了算法良好的全局探索能力。通过对

14个基准函数的仿真测试以及与标准SOS算法、GWO算法、PSO算法和ABC算法相比较的实验结果可以看出,本文算法在收敛速度、搜索精度、鲁棒性以及快速跳出局部最优等方面具有更好的性能。下一步将研究如何使本文算法与实际应用相结合。

参考文献(References):

- [1] TANG K S, MAN K F, KWONG S, et al. Genetic Algorithms and Their Applications[J]. IEEE Signal Processing Magazine, 1996, 13: 22-37.
- [2] WILLIAM F F. Genetic Algorithms[J]. Journal of the American Statistical Association, 2002, 97(457): 366-375.
- [3] SAAD G Y, NADA M. Ant Colony Optimization[J]. International Journal of Computer Science and Network Security, 2008, 8(6): 351-357.
- [4] LESLIE P C, MANUEL L I, THOMAS S. Ant Colony Optimization on a Limited Budget of Evaluations[J]. Swarm Intelligence, 2015, 9(2-3): 103-124.
- [5] RICCARDO P, JAMES K, TIM B. Particle Swarm Optimization[J]. Swarm Intelligence, 2007, 1(5): 33-57.
- [6] KENNEDY J, EBERHART R. Particle Swarm Optimization[C]//IEEE International Conference on Neural Networks, 1995: 1942-1949.
- [7] LI X, ZHANG J, YIN M. Animal Migration Optimization: an Optimization Algorithm inspired by Animal Migration Behavior[J]. Neural Computing and Applications, 2014, 24(7-8): 1867-1877.
- [8] 李牧东,赵辉,翁兴伟.具有广泛学习策略的回溯搜索优化算法[J].系统工程与电子技术,2015,37(4): 958-963.
LI Mudong, HAO Hui, WENG Xingwei. Backtracking Search Optimization Algorithm with Comprehensive Learning Strategy[J]. Systems Engineering and Electronics, 2015, 37(4): 958-963. (in Chinese)
- [9] CHENG M Y, PRAYOGO D. Symbiotic Organisms Search: A New Metaheuristic Optimization Algorithm[J]. Computers and Structures, 2014, (139): 98-112.
- [10] WANG Y, CAI Z X, ZHANG Q F. Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters[J]. IEEE Transaction on Evolutionary Computation, 2011, 15(1): 55-66.
- [11] LIANG J J, QIN A K, SUGANTHAN PN, et al. Comprehensive Learning Particle Swarm Optimizer Ffor Global Optimization of Multimodal Functions [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281 - 295.
- [12] ZHANG J Q, SANDERSON A C. SADE: Adaptive Differential Evolution with Optional External Archive [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 945-958.
- [13] YANG X S, DEB S. Cuckoo Search via Levy Flights [J]. Mathematics, 2010: 210-214.
- [14] KARABOGA D, BASTURK B. A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm [J]. Journal of Global Optimization, 2007, 39(3): 459-471.
- [15] SEYEDALI M, SEYED M M, ANDREW L. Grey Wolf Optimizer[J]. Advances in Engineering Software, 2014, (69): 46-61.
- [16] RODRIGUEZ F J, GARCIA-MARTINEZ C, LOZANO M. Hybrid Metaheuristics Based on Evolutionary Algorithms and Simulated Annealing: Taxonomy Comparison, and Synergy Test[J]. IEEE Transactions on Evolutionary Computation, 2012, 16(6): 787-800.
- [17] 赵辉,李牧东,翁兴伟.具有自适应全局最优引导快速搜索策略的人工蜂群算法[J].控制与决策,2014,29(11): 2041-2047.
ZHAO Hui, LI Mudong, WENG Xingwei. Improved Artificial Bee Colony Algorithm with Self-Adaptive Global Bestguided Quick Searching Strategy[J]. Control and Decision, 2014, 29(11): 2041-2047. (in Chinese)
- [18] 高卫峰,刘三阳,黄玲玲.受启发的人工蜂群算法在全局优化问题中的应用[J].电子学报,2012,40(12): 2396-2403.
GAO Weifeng, LIU Sanyang, HUANG Lingling. Inspired Artificial Bee Colony Algorithm for Global Optimization Problems [J]. Acta Electronica Sinica, 2012, 40(12): 2396-2403. (in Chinese)

(编辑:徐敏)