

文章编号:1671-6833(2017)06-0033-06

精英反向学习的单纯形交叉布谷鸟搜索算法

林要华, 王李进

(福建农林大学 计算机与信息学院,福建 福州 350002)

摘 要: 提出一种加强搜索能力的改进布谷鸟搜索算法,该算法采用精英反向学习策略促使 Lévy Flights 随机走动中的部分精英个体进行反向搜索,以避免搜索新个体的趋同性;并采用单纯形交叉操作在 Biased 随机走动中随机选择一个个体进行精细搜索,以降低搜索的盲目性以及低效性. 另外,提出的算法采用混沌映射模型实现发现概率参数的自适应控制. 仿真实验结果表明,该算法能够总体上有效改善算法的搜索能力和收敛速度.

关键词: 布谷鸟搜索算法;单纯形交叉;反向学习;混沌映射

中图分类号: TP301 文献标志码: A doi:10.13705/j.issn.1671-6833.2017.06.033

0 引言

近年来,涌现诸如人工蜂群算法^[1]、萤火虫算法^[2]、群搜索算法^[3]、布谷鸟搜索算法^[4-5]等一些新颖算法. 布谷鸟搜索算法(cuckoo search, CS)在求解函数优化问题上能获得较好的性能,同时具有运算简单和参数较少的特点,这些使得该算法已成为研究的热点. 一方面,一些学者针对算法中的随机走动进行了研究与改进,文献[6-8]开展了 Lévy Flights 随机走动及其步长的相关研究工作;文献[9-10]则对加强 Biased 随机走动的搜索方式进行了研究. 另一方面,一些学者通过混合其他优化算法以提高算法的搜索能力^[11-13].

上述研究容易导致种群的同质性问题,而且种群中的当前最优个体无法从自身获得任何步长信息,这也限制了该个体的搜索能力. 另外, Biased 随机走动中的随机交叉搜索方式可能会导致盲目或者低效搜索问题. 因此,笔者针对上述两个问题,引进解对称和单纯形交叉策略,以改善 CS 算法的收敛速度以及求精能力,并提出了精英反向学习的单纯形交叉布谷鸟搜索算法(elite opposition-based learning based simplex crossover cuckoo

search algorithm, ESCS).

1 布谷鸟搜索算法

布谷鸟搜索算法是 Yang 和 Deb^[5-6] 基于布谷鸟的繁殖策略而构建的算法. 该算法主要包含 3 种部件: Lévy Flights 随机走动、Biased 随机走动以及贪婪选择. 对于求解连续函数优化问题, CS 算法首先在问题的解空间 $[x_{j,min}, x_{j,max}]$ ($j = 1, 2, \dots, D$)内随机初始化种群.

$x_{i,j,0} = x_{i,j,min} + r(x_{i,j,max} - x_{i,j,min})$, $i = 1, 2, \dots, N$, (1)
式中: $r \in [0, 1]$ 是缩放因子; N 是种群大小.

在初始化之后, CS 算法进入进化阶段. 在进化过程中, Lévy flights 随机走动和 Biased 随机走动被分别用于搜索新的个体. 在每个随机走动之后, CS 算法比较每个当前个体和新搜索个体的适应值,贪婪选择适应值较好的个体进入种群.

在 Lévy flights 随机走动中,种群以当前获得的最优个体为导向进行进化,其形式可表达为

$$X_{i,G+1} = X_{i,G} + \alpha_0 \frac{\varphi \cdot u}{|v|^{1/\beta}} (X_{i,G} - X_{best}). \quad (2)$$

式中: $X_{i,G}$ 、 $X_{i,G+1}$ 和 X_{best} 分别表示第 G 代的第 i 个当前个体、新个体以及当前最优个体; α_0 为缩放因子(取 0.01); μ, ν 服从正态分布; $\beta = 1.5$; 以及

收稿日期:2017-02-22;修订日期:2017-07-29
基金项目:福建省自然科学基金资助项目(2016J01280);福建省教育厅资助项目(JB09114)
作者简介:林要华(1979—),男,福建宁德人,福建农林大学讲师,主要从事智能算法及其应用研究.
通信作者:王李进(1977—),男,福建泉州人,福建农林大学副教授,博士,主要从事智能算法、智能信息处理研究,
E-mail:lijinwang@fafu.edu.cn.

$$\varphi = \left(\frac{\Gamma(1 + \beta) \cdot \sin\left(\frac{\pi \times \beta}{2}\right)}{\Gamma\left(\frac{1 + \beta}{2}\right) \cdot \beta \cdot 2^{\frac{\beta-1}{2}}} \right)^{1/\beta}, \quad (3)$$

其中, Γ 表示伽马(Gama)方程.

在 Biased 随机走动中, 为了搜索新的个体, 当前个体随机选择两个个体, 各维按一定的概率接受它们的步长信息, 具体如下:

$$x_{i,j,G+1} = \begin{cases} x_{i,j,G} + r(x_{m,j,G} - x_{n,j,G}), & \text{if } \text{rand}(0,1) > p_a; \\ x_{i,j,G}, & \text{otherwise,} \end{cases} \quad (4)$$

式中, $x_{m,j,G}$ 、 $x_{n,j,G}$ 表示第 G 代的第 m 和第 n 个解; r 是缩放因子; $p_a \in [0,1]$, 是发现概率.

2 提出的改进算法

针对 CS 算法的搜索新个体的趋同性以及搜索低效率问题, 提出一种基于精英反向学习的单纯形交叉布谷鸟搜索算法, 这种算法采用精英反向学习、单纯形交叉以及发现概率 p_a 的自适应控制策略.

2.1 精英反向学习

基于群体搜索的智能算法的迭代过程是一个搜索区域变换到另一个搜索区域的过程^[14]. 反向学习 (opposition-based learning, OBL)^[15] 是实现区域变换搜索的一种模式. 文献^[14]概括了 4 种常见反向学习模型.

$$x^* = k(a + b) - x, \quad (5)$$

式中: a 、 b 为搜索空间的上下界; k 为实数.

当 $k = 0$ 时, 得解对称的一般反向学习模型, 即

$$x^* = -x. \quad (6)$$

当 $k = 1/2$ 时, 得区间对称的一般反向学习模型, 即

$$x^* = \left(\frac{a + b}{2} \right) - x. \quad (7)$$

当 $k = 1$ 时, 得反向学习模型, 即

$$x^* = (a + b) - x. \quad (8)$$

当 $k = r$ 时, r 为 $[0,1]$ 之间的随机数时, 得到随机的一般反向学习模型, 即

$$x^* = r(a + b) - x. \quad (9)$$

2.2 单纯形交叉

单纯形交叉 (Simplex Crossover, SPX)^[16] 是一种基于交叉操作的局部搜索技术, 常用于加强个体的求精, 并呈现出潜在的性能^[17].

算法 1 SPX 的流程如下.

步骤 1: 计算 n 个父个体 $x_i (i = 1, \dots, n)$ 的中心位置 O .

$$O = \frac{1}{n} \sum_{i=1}^n x_i.$$

步骤 2: 生成 $n - 1$ 个随机数 $r_i, r_i = u^{\frac{1}{i+1}}, i = 1, \dots, n - 1; u$ 为 $[0,1]$ 之间的随机数.

步骤 3: 计算 $y_i = O + \varepsilon(x_i - O), i = 1, \dots, n$, ε 为扩张因子, $\varepsilon = 1.0$.

$$C_i = \begin{cases} 0, & i = 1; \\ r_{i-1}(y_{i-1} - y_i + C_{i-1}), & i = 2, \dots, n \end{cases}$$

步骤 4: 生成子个体 $C = y_n + C_n$.

2.3 发现概率 p_a 的自适应控制

式(4)中的 p_a 控制着个体获取外部信息的概率. 越小的发现概率使得个体接受外部信息的机会越大, 这有利于搜索的全局性; 反之, 有利于个体的求精能力. 显然, 动态控制 p_a 有利于改善搜索部件的效率. 因此, 与文献^[14]类似, ESCS 采用高斯混沌映射模型控制 p_a 参数, 见式(10), 其中初始 $p_a(0) = 0.7$.

$$p_a(i + 1) = \begin{cases} 1, & p_a(i) = 0; \\ 1/\text{mod}(p_a(i), 1), & p_a(i) \neq 0. \end{cases} \quad (10)$$

2.4 ESCS 算法框架

算法 2 给出了 ESCS 算法的流程. 需要指出的是在步骤 2.1 中, 除了采用式(8)外, 也可采用式(6)、式(7)和式(9)的反向学习模型搜索新的个体, 并引入参数 M 控制精英个体规模. 本文简单采用种群中的前 M 个最优个体作为精英个体. 另外, ESCS 选择待交叉个体以及当前种群中的 $K - 1$ 个随机个体, 组合成 K 个个体作为单纯形交叉的父个体.

算法 2 ESCS 的流程如下.

步骤 1: 采用公式(1)初始化种群 $X_{ij} (i = 1, \dots, N; j = 1, \dots, D)$;

步骤 2: 当迭代终止条件未满足;

步骤 2.1: 的伪代码如下.

① For $i = 1$ to N

② if X_i 是精英个体而且精英个数 $\leq M$

③ 采用式(8)反向学习搜索个体 V_i

④ Else

⑤ 采用式(2)搜索个体 V_i

⑥ Endif

⑦ 如果 V_i 的适应值优于 X_i , 则用 V_i 替换 X_i

⑧ Endfor

步骤 2.2: 伪代码如下.

① 从 $[1, N]$ 中随机产生待用单纯形交叉个体 indSPX

② 选择 n 个体成为父个体

③ 根据式(10)获得 p_a

④For $i = 1$ to N

⑤if $i = \text{indSPX}$

⑥采用算法 1 生成候选子个体 V_i

⑦else

⑧采用公式(4)生成候选子个体 V_i

⑨Endif

⑩如果 V_i 的适应值优于 X_i ,则用 V_i 替换 X_i

⑪Endfor

步骤 2.3:更新当前最优个体;

步骤 3:输出最优个体.

3 仿真与结果

为了分析 ESCS 算法的性能,选择文献[18]中的 10 个旋转变换函数作为算法的测试函数.其中, $F_1 \sim F_5$ 为单峰函数; $F_6 \sim F_{10}$ 为多峰函数.更详细的测试函数内容见文献[18].同时,为了公平评价算法的性能,笔者采用平均适应值误差及其收敛曲线图评价准则.其中,适应值误差 E 计算见式(11).

$$E = f(X) - f_{\min}, \tag{11}$$

式中: X 为算法得到的解,是已知的最优适应值.从公式可以看出,适应值误差越小则性能越优.

3.1 参数分析

ESCS 算法引入参数 M 和 K 分别用于控制参与精英反向学习个体的规模和参与单纯形交叉的父个体规模.由于 M 和 K 存在不同的取值范围,这就形成规模较大的组合,因此,本实验中假设其中一个参数取常量情况下分析另一个参数.

首先,为了分析参数 M 对算法性能的影响,图 1 给出了 ESCS 算法在 K 为常量以及不同 M 时求解 10 个测试函数($D = 30$)Friedman 的检验平均排序.其中,每个 M 取值所对应的算法在每个测试函数上独立运行 25 次,种群大小 $N = 30$,最大函数评价次数为 $10\,000 \times D$.

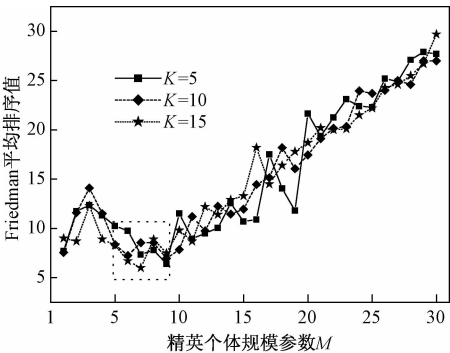


图 1 不同 M 时的平均排序值

Fig.1 Mean rank under different M

从图 1 可以看出, M 取较小时,算法总体上获得较好性能.这是因为大部分的个体仍采用式(2)搜索新个体时,种群的小部分精英个体参与反向学习确保了新个体的趋同性,从而加强其搜索能力.另外,从图 1 可知,当 $M \in [6, 9]$ 时,算法的性能总体最优;综合不同 K 值情况下的性能可知, $M = 9$ 较为理想.

图 2 给出当 $M \in [6, 9]$ 时,取不同 K 值的算法求解函数的 Friedman 检验的平均排序.从图 2 中可以看出, M 不同时,算法获得最优的平均排序的 K 值是不同的.但是,从图 2 中的水平线(平均排序值 = 13.3)可知,只有 $K = 15$ 所对应的 4 个 M 的平均排序都处于水平线下,而 K 取其他值所对应 M 的平均排序少数低于水平线下,这说明了 $K = 15$ 是较为理想的.

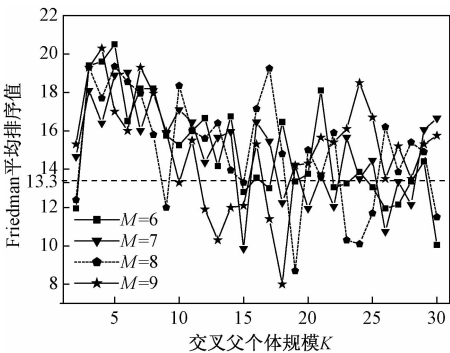


图 2 不同 K 时的平均排序值

Fig.2 Mean rank under different K

3.2 性能分析

表 1 给出了 ESCS 算法和 CS 算法在 $D = 30$ 维上的“ $E_{\text{avg}} \pm E_{\text{sd}}$ ”(E_{avg} 是平均适应值误差, E_{sd} 是标准方差).其中,每个算法在每个测试函数上独立运行 25 次,两种算法的种群大小 $N = 30$,而 ESCS 算法的 $M = 9, K = 15$,最大函数评价次数为 $10\,000 \times D$.另外,表 1 也给出了两种算法在 0.05 显著水平下的 Wilcoxon 符号秩检验,“+”表示 ESCS 算法显

表 1 CS 和 ESCS 求解 $D = 30$ 维的性能

Tab.1 Performance of CS and ESCS for solving $D = 30$

函数	CS		差异 性	ESCS	
	$E_{\text{avg}} \pm E_{\text{sd}}$			$E_{\text{avg}} \pm E_{\text{sd}}$	
F_1	2.52E-30	$\pm 7.51\text{E}-30$	=	2.02E-30	$\pm 1.01\text{E}-29$
F_2	7.38E-03	$\pm 8.82\text{E}-03$	+	2.08E-09	$\pm 4.88\text{E}-09$
F_3	2.22E+06	$\pm 6.19\text{E}+05$	+	1.27E+05	$\pm 1.06\text{E}+05$
F_4	1.38E+03	$\pm 7.33\text{E}+02$	+	1.88E+01	$\pm 1.91\text{E}+01$
F_5	3.09E+03	$\pm 6.75\text{E}+02$	+	5.83E+02	$\pm 4.32\text{E}+02$
F_6	2.22E+01	$\pm 2.40\text{E}+01$	+	4.78E-01	$\pm 1.32\text{E}+00$
F_7	7.09E-04	$\pm 1.62\text{E}-03$	-	1.66E-02	$\pm 1.32\text{E}-02$
F_8	2.09E+01	$\pm 4.34\text{E}-02$	=	2.09E+01	$\pm 1.04\text{E}-01$
F_9	2.74E+01	$\pm 5.44\text{E}+00$	+	2.05E+01	$\pm 5.97\text{E}+00$
F_{10}	1.62E+02	$\pm 2.59\text{E}+01$	+	5.81E+01	$\pm 1.22\text{E}+01$

著优于 CS 算法,“-”表示 CS 算法显著优于 ESCS 算法,而“=”表示两算法不存在显著差异。

从表 1 可以看出,ESCS 算法总体上可以改善求精能力。由 Wilcoxon 统计检验的结果分析可知,虽然在 F_7 测试函数上,ESCS 算法的求解能力未显著优于 CS 算法,但在其他测试函数上,ESCS 算法性能显著优于 CS 算法。

为了进一步比较算法的收敛性能,图 3 形化展

示 CS 和 ESCS 算法的收敛过程,其中,收敛曲线图中的平均适应值误差均是变换后的平均适应值误差,变换函数是 \lg 。从图 3 可以看出,对于 ESCS 算法获得的适应值优于 CS 算法的函数,ESCS 算法同时具有较快的收敛速度,在迭代后期具有较好的求精能力,可见图 3 中的(a)~(f)、(h)、(i)和(j)。对于 F_7 函数的收敛过程见图 3(g),ESCS 算法在前期具有较快的收敛,但后期陷入局部最优。

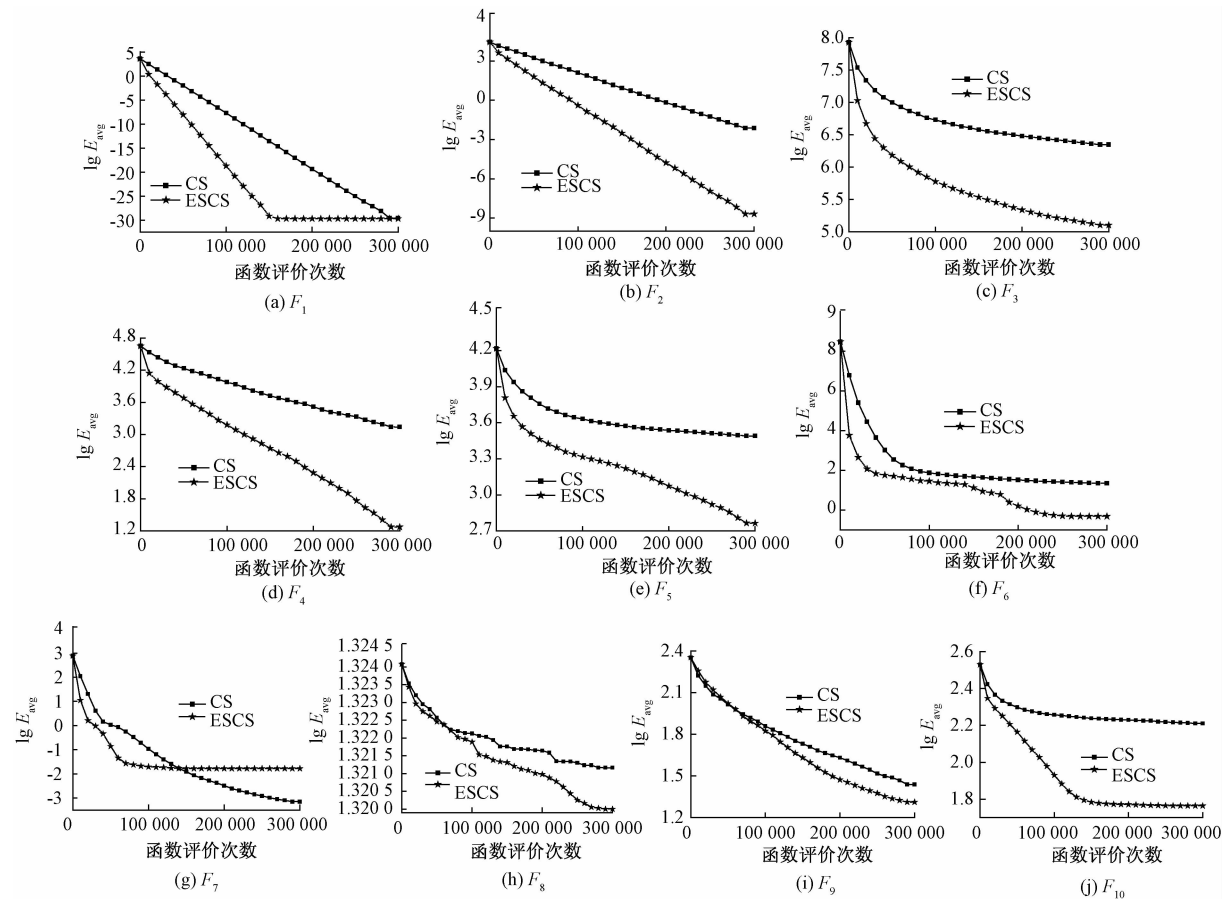


图 3 CS 和 ESCS 的收敛曲线
Fig. 3 Convergence curves of ESCS and CS

综合上述的实验结果,相对于 CS 算法,ESCS 算法对于大多数函数具有较强的求精能力以及较快的收敛速度。

3.3 ESCS 与其他改进的 CS 算法比较

为了进一步观察 ESCS 与其他改进的 CS 算法的竞争优势,表 2 列出 ESCS 算法与 CSPSO^[11]、OLCS^[12]以及 CCS^[8]在 $D = 30$ 维空间上的求精能力比较结果,其中用于对比算法的实验数据来源于文献[8]。表中加粗数据表示相应算法在该函数获得最优性能,可以得出,不同算法在求解给定的测试函数上具有不同的优势;ESCS 算法在 F_4 、 F_5 、 F_6 和 F_{10} 函数上获得最好的解质量。另外,CSPSO、OLCS、CCS 和 ESCS 4 种算法的 Friedman 检验结果

分别为 3.2、3.3、1.8、1.7。从检验结果可知,ESCS 具有最小的平均排序值,这说明 ESCS 算法总体性能最优,其次依次为 CCS、CSPSO 和 OLCS。

3.4 不同反向学习模型对算法性能影响分析

为了分析不同反向学习模型对算法性能的影响,表 3 给出了 ESCS 在不同反向学习模型下求解测试函数所获得的平均误差,而表 4 列出相关算法的 Friedman 检验结果。其中,ESCS₆、ESCS₇ 和 ESCS₉ 表示 ESCS 算法分别采用式(6)、式(7)以及式(9)的反向学习模型。

从表 3 可知,不同反向学习模型使算法求解部分函数时获得较好性能。例如,ESCS₆ 算法在 F_1 和 F_3 上获得最佳性能;ESCS₇ 算法在在 F_1 上

表 2 ESCS 与 CSPSO、OLCS 以及 CCS 的性能比较

Tab.2 Performance comparison of ESCS with CSPSO, OLCS and CCS

函数	CSPSO	OLCS	CCS	ESCS
	$E_{avg} \pm E_{sd}$	$E_{avg} \pm E_{sd}$	$E_{avg} \pm E_{sd}$	$E_{avg} \pm E_{sd}$
F_1	2. 52E - 28 ± 2. 31E - 28	1. 31E - 26 ± 1. 05E - 26	0. 00E + 00 ± 0. 00E + 00	2. 02E - 30 ± 1. 01E - 29
F_2	3. 28E - 11 ± 9. 03E - 11	5. 80E - 02 ± 5. 16E - 02	5. 34E - 09 ± 7. 74E - 09	2. 08E - 09 ± 4. 88E - 09
F_3	7. 44E + 05 ± 6. 43E + 05	2. 83E + 06 ± 7. 55E + 05	8. 89E + 04 ± 5. 33E + 04	1. 27E + 05 ± 1. 06E + 05
F_4	8. 41E + 01 ± 1. 05E + 02	2. 01E + 03 ± 8. 15E + 02	3. 20E + 01 ± 9. 74E + 01	1. 88E + 01 ± 1. 91E + 01
F_5	2. 89E + 03 ± 7. 82E + 02	2. 54E + 03 ± 5. 86E + 02	6. 84E + 02 ± 6. 65E + 02	5. 83E + 02 ± 4. 32E + 02
F_6	2. 31E + 00 ± 4. 79E + 00	2. 52E + 01 ± 2. 04E + 01	9. 31E - 01 ± 2. 58E + 00	4. 78E - 01 ± 1. 32E + 00
F_7	7. 40E - 03 ± 2. 20E - 15	3. 43E - 04 ± 4. 42E - 04	5. 71E - 03 ± 6. 58E - 03	1. 66E - 02 ± 1. 32E - 02
F_8	2. 09E + 01 ± 5. 49E - 02	2. 09E + 01 ± 5. 75E - 02	2. 09E + 01 ± 7. 18E - 02	2. 09E + 01 ± 1. 04E - 01
F_9	1. 55E + 02 ± 2. 47E + 01	3. 56E + 01 ± 6. 56E + 00	1. 62E + 01 ± 5. 19E + 00	2. 05E + 01 ± 5. 97E + 00
F_{10}	2. 57E + 02 ± 6. 81E + 01	1. 52E + 02 ± 3. 48E + 01	6. 14E + 01 ± 1. 48E + 01	5. 81E + 01 ± 1. 22E + 01

表 3 不同反向学习模型的 ESCS 性能

Tab.3 Performance of ESCS by different learning models

函数	ESCS ₆	ESCS ₇	ESCS ₉	ESCS
	$E_{avg} \pm E_{sd}$	$E_{avg} \pm E_{sd}$	$E_{avg} \pm E_{sd}$	$E_{avg} \pm E_{sd}$
F_1	0. 00E + 00 ± 0. 00E + 00	0. 00E + 00 ± 0. 00E + 00	2. 02E - 30 ± 1. 01E - 29	2. 02E - 30 ± 1. 01E - 29
F_2	4. 28E - 07 ± 6. 36E - 07	2. 94E - 07 ± 4. 37E - 07	3. 81E - 07 ± 4. 88E - 09	2. 08E - 09 ± 4. 88E - 09
F_3	1. 12E + 05 ± 6. 80E + 04	1. 55E + 05 ± 8. 42E + 04	1. 43E + 05 ± 1. 06E + 05	1. 27E + 05 ± 1. 06E + 05
F_4	8. 66E + 01 ± 7. 75E + 01	8. 31E + 01 ± 8. 24E + 01	1. 29E + 02 ± 1. 91E + 01	1. 88E + 01 ± 1. 91E + 01
F_5	1. 03E + 03 ± 5. 44E + 02	9. 91E + 02 ± 4. 97E + 02	8. 74E + 02 ± 4. 32E + 02	5. 83E + 02 ± 4. 32E + 02
F_6	3. 51E + 00 ± 2. 49E + 00	4. 48E + 00 ± 3. 91E + 00	4. 89E + 00 ± 1. 32E + 00	4. 78E - 01 ± 1. 32E + 00
F_7	1. 38E - 02 ± 1. 69E - 02	1. 31E - 02 ± 1. 24E - 02	1. 18E - 02 ± 1. 32E - 02	1. 66E - 02 ± 1. 32E - 02
F_8	2. 09E + 01 ± 7. 14E - 02	2. 09E + 01 ± 1. 06E - 01	2. 09E + 01 ± 1. 04E - 01	2. 09E + 01 ± 1. 04E - 01
F_9	2. 31E + 01 ± 6. 74E + 00	2. 39E + 01 ± 7. 10E + 00	2. 13E + 01 ± 5. 97E + 00	2. 05E + 01 ± 5. 97E + 00
F_{10}	6. 24E + 01 ± 1. 41E + 01	6. 78E + 01 ± 1. 33E + 01	6. 13E + 01 ± 1. 22E + 01	5. 81E + 01 ± 1. 22E + 01

获得最优值;ESCS₉ 算法求解 F_7 获得较好适应值;ESCS 算法在较多的函数上获得较好解. 同时,根据表 4 中各算法的平均排序可知,基于式(8)的 ESCS 算法具有较小的平均排序,这说明采用反向学习模型总体最优.

表 4 4 个不同反向学习模型的 Friedman 检验结果

Tab.4 Friedman results of the four algorithms by different learning models

函数	ESCS ₆	ESCS ₇	ESCS ₉	ESCS
排序值	2. 75	2. 75	2. 85	1. 65

4 结论

笔者提出一种基于精英反向学习和单纯形交叉以及发现概率自动控制相结合的布谷鸟搜索算法. 仿真实验验证了提出的策略总体上能够改善算法的求精能力和收敛速度,从而也验证提出的策略是有效和可行的. 下一步工作将提出的算法应用于实际工程优化问题.

参考文献:

[1] KARABODA D. An idea based on honey bee swarm for numerical optimization [R]. Kayseri, Turkey: Computer Engineering Department, Erciyes University, 2005.

[2] YANG X. Nature-inspired metaheuristic algorithms [M]. Frome, UK: Luniver Press, 2008.

[3] HE S, WU Q, SAUNDERS J. A novel group search optimizer inspired by animal behavioral ecology [C]// Proceedings of 2006 IEEE Congress on Evolutionary Computation. Vancouver, BC: IEEE, 2006:1272 - 1278.

[4] YANG X, DEB S. Cuckoo search via Levy flight [C]//Proceedings of World Congress on Nature & Biologically Inspired Computing. Piscataway, NJ: IEEE, 2009: 210 - 214.

[5] YANG X, DEB S. Engineering optimization by cuckoo search [J]. Int J Math Modelling Num Opt, 2010, 1 (4) :330 - 343.

[6] VALIAN E, MOHANNA S, TAVAKOLI S. Improved cuckoo search algorithm for global optimization [J].

- Int J Communications and Information Technology, 2011, 1(1):31–44.
- [7] WANG L, YIN Y, ZHONG Y. Cuckoo search with varied scaling factor [J]. Frontiers of computer science, 2015, 9(4):623–635.
- [8] WANG L, ZHONG Y. Cuckoo search algorithm with chaotic maps [J]. Mathematical problems in engineering, 2015(1):1–14.
- [9] 王李进, 尹义龙, 钟一文. 逐维改进的布谷鸟搜索算法 [J]. 软件学报, 2013, 24(11):2687–2698.
- [10] 王李进, 钟一文, 尹义龙. 带外部存档的正交交叉布谷鸟搜索算法 [J]. 计算机研究与发展, 2015, 52(11):2496–2507.
- [11] WANG F, HE X, LUO L, et al. Hybrid optimization algorithm of PSO and cuckoo search [C]//Proc of the 2nd Int Conf on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC). Piscataway, NJ: IEEE, 2011:1172–1175.
- [12] LI X, WANG J, YIN M. Enhancing the performance of cuckoo search algorithm using orthogonal learning method [J]. Neural computing & applications, 2014, 24(6):1233–1247.
- [13] WANG L, ZHONG Y. One position inheritance based cuckoo search algorithm [J]. Int J Computing Science and Mathematics, 2015, 6(6):546–554.
- [14] 王晖. 基于一般反向学习的群体随机搜索算法框架 [J]. 南昌工程学院学报, 2012, 31(3):1–6.
- [15] TIZHOOSH H. Opposition-based learning: a new scheme for machine intelligence [C]//Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce. Piscataway, NJ: IEEE, 2005, 1:695–701.
- [16] TSUTSUI S, YAMAMURA M, HIGUCHI T. Multi-parent recombination with simplex crossover in real coded genetic algorithms [C]//Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation (GECCO'99). Orlando, Florida, USA: CECCO, 1999:657–664.
- [17] NOMAN N, IBA H. Accelerating differential evolution using an adaptive local search [J]. IEEE transaction on evolutionary computation, 2008, 12(1):107–125.
- [18] SUGANTHAN P, HANSEN N, LIANG J, et al. Problem definitions and evaluation criteria for the CEC2005 special session on real-parameter optimization [R]. Singapore, singapore: Nanyang Technological University, Singapore, 2005.

Elite Opposition-based Learning Based Simplex Crossover Cuckoo Search Algorithm

LIN Yaohua, WANG Lijin

(College of Computer And Information Science, Fujian Agriculture And Forestry University, Fuzhou 350002, China)

Abstract: Cuckoo search algorithm iteratively uses Lévy Flights random walk and Biased random walk to search for new individuals. In this paper, an enhanced cuckoo search was proposed, which employed elite opposition-based learning, simplex crossover and parameter control for the fraction probability. The elite opposition-based learning strategy was used to avoid the new individuals being homogeneous in the Lévy Flights random walk. The simplex crossover strategy was utilized to reduce the inefficiency of Biased random walk. The chaotic map was used to adaptively adjust the parameter p_a to balance the exploration and the exploitation. The results of experiment showed the proposed strategies were overall effective, and make a great improvement on the performance of solution and convergence.

Key words: cuckoo search algorithm; simplex crossover; opposite learning; chaotic maps