

均衡单进化布谷鸟算法

傅文渊

(1. 华侨大学信息科学与工程学院, 福建厦门 361002; 2. 中山大学电子与信息工程学院, 广东广州 510006; 3. 厦门市专用电路系统重点实验室, 福建厦门 361008; 4. 福建省电机控制与系统优化调度工程技术研究中心, 福建厦门 361002)

摘 要: 针对布谷鸟算法采用整体评价策略处理多维度自变量相关优化问题时, 维度耦合现象会恶化算法的搜索速度和收敛精度, 提出均衡单进化的布谷鸟算法(ESCES). 该算法给出一种新型的均衡单进化函数评价策略, 即每一代进化只随机更新目标函数的单个维度, 并且随机更新的维度服从均匀分布, 避免多维度之间互相干扰. 同时, 提出两种新型随机游动步长更新学习律, 提高了优化算法的全局搜索速度和收敛精度. 实验测试结果和显著性统计结果表明, ESCES算法与5个改进CS算法及7个其它最新智能优化算法相比, 在全局寻优性能、搜索速度和收敛精度上均获得较大的改进.

关键词: 进化; 评价策略; 布谷鸟算法; 发现概率

中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2019)02-0282-07

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2019.02.004

Equilibrium Single Evolution Based Cuckoo Search Algorithm

FU Wen-yuan

(1. College of Information Science and Engineering, Huaqiao University, Xiamen, Fujian 361002, China;
2. School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, Guangdong 510006, China;
3. Xiamen Key Laboratory of ASIC System, Xiamen, Fujian 361008, China;
4. Fujian Engineering Research Center of Motor Control and System Optimal Schedule, Xiamen, Fujian 361002, China)

Abstract: For the whole evaluation strategy in cuckoo search algorithm in the face of multi-dimension function optimization problems, the coupling phenomena among dimensions will deteriorate the search speed and convergence accuracy. Therefore, a new cuckoo search algorithm based on the equilibrium single evolution mechanism is proposed. Then, a new equilibrium single evolution evaluation strategy is also used to update randomly the single dimension of the objective function on each iteration. Note that the randomly updated dimensions obey the uniform distribution to avoid mutual interference between dimensions. Furthermore, two new random walking update laws are proposed to improve the global search speed and convergence accuracy. The results of the 10 benchmark functions and statistical significance demonstrate that ESCES algorithm has a great improvement in global optimization performance, search speed and convergence accuracy compared with the five modified CS algorithms and seven other state-of-the-art algorithms.

Key words: evolution; evaluation strategy; cuckoo search algorithm; discovery probability

1 引言

随着电子与计算机等领域的迅速发展, 传统的优化方法已经很难满足具有 NP 难度的设计需求. 由于新兴的智能启发式优化算法可以较好的解决这类问题, 因此得到了国内外学者的普遍关注和重点研究. 布谷鸟算法是众多智能启发式优化算法中的一种新型算

法, 由剑桥大学 Yang X. S. 教授等人于 2009 年提出^[1,2]. 该算法由于结构简单, 调节参数较少, 执行速度较快, 因此广受国内外学者的关注, 已经成功应用于集成电路设计^[3]、神经网络训练^[4-6]、工程结构优化^[7-9]、多目标优化^[10-12]和电子与电力系统控制^[13,14]等领域.

国内外学者对布谷鸟算法进行了较为深入的研究, 并取得了较为丰富的研究成果. 文献[15]利用梯度

下降算法更新 Levy 随机游动步长提高了布谷鸟算法的局部搜索性能. 文献[16]将布谷鸟算法中的 Levy 随机游动步长和发现概率设计为种群变化的函数, 较好的平衡了算法的全局寻优能力和局部搜索能力. 文献[17]在文献[16]基础上, 设计了一种发现概率随种群变化而自适应改变的策略, 有效提高了算法的收敛速度. 文献[18]将发现概率设置为动态随机变化, 增强了算法的收敛速度, 但是由于改进算法的搜索步长的容许范围过小, 因此算法全局收敛性能较差. 文献[19]提出一种基于梯度信息的布谷鸟优化算法, 克服了 CS 算法中局部收敛性能较差的缺陷. 文献[20]将牛顿快速下降法与布谷鸟算法结合, 有效的解决了 CS 算法在进化后期收敛速度较慢的问题. 但是该算法由于只注重局部收敛速度而忽视算法全局开采性能, 导致在处理高维度多模函数优化问题时容易陷入局部最优. 文献[21]提出基于量子效应的异向搜索布谷鸟算法, 利用量子隧穿效应跳跃机理逃离局部最优, 极大地提高了算法的搜索速度和收敛精度.

迄今为止, 绝大部分文献在考察算法进化的有效性均采用目标函数适应度值整体评价的策略. 这种评价策略对于单维度优化问题效果较为有效, 但对于多维度自变量相关问题, 由于存在维度间相互耦合现象, 因此会制约算法的搜索速度和收敛精度. 文献[22]和[23]通过重新构建评价因子对更新后的维度进行评定, 以降低对目标函数适应度整体评价的依赖. 然而, 对于复杂的多维度自变量强耦合目标函数或者更复杂的非线性优化问题, 重新构建新的评价因子是困难的, 甚至是不可实现的, 这将极大提高算法运行复杂度及降低算法的实用性. 文献[24]在文献[22, 23]的基础上提出了一种逐维更新的函数评价策略, 将目标函数各维度逐一更新. 该算法具有一定的竞争力, 但逐维更新的评价策略对于多维度自变量强耦合优化问题, 易陷入局部最优. 同时, 该算法每次进化过程必须按照维度逐一更新, 导致函数评价次数急剧增加, 执行效率较低.

基于此, 本文提出一种均衡单进化的布谷鸟算法 (Equilibrium Single Evolution based Cuckoo Search, ESECS). 该算法给出一种新型的均衡单进化函数评价策略. 同时, 针对传统 CS 算法全局搜索效率较低和算法执行后期出现迟滞现象, 提出两种新型随机游动步长更新学习律, 以便充分利用单次进化的信息加快算法搜索速度, 避免优化算法执行末期出现的迟滞现象, 从而提高算法的全局搜索速度和收敛精度.

2 布谷鸟算法 (CS)

布谷鸟是一种特殊的鸟类, 它采用奇特的寄生宿主巢穴的方式来孕育繁殖后代. 布谷鸟在繁殖期间寻

找与自己孵化期与雏鸟期相似的宿主, 宿主孵卵之前将自己孵育的蛋置入寄生宿主的巢穴, 让寄生的宿主鸟孵化布谷鸟蛋. 剑桥大学的 Yang X. S. 和 Deb 等人基于布谷鸟孵育寄生机理, 提出布谷鸟算法^[1]. 该算法的 1 个宿主巢穴代表 1 个候选解. 采用 Levy 飞行随机游动搜索寄生宿主巢穴位置, 也即是候选解 $z_{i+1,j}$ 的数学表达式,

$$z_{i+1,j} = z_{i,j} + (\eta \oplus L(\lambda)) \cdot (z_{i,j} - z_{i,g}) \quad (1)$$

其中 $z_{i,j}$ 代表算法进化至第 i 代时的第 j 个候选解; $z_{i,g}$ 为第 i 代搜索到的全局最优解; $L(\lambda)$ 表示 Levy 飞行随机游动路径; η 表示算法初始搜索步长, 与优化问题的维度及复杂性有关; \oplus 表示点对点乘法. $L(\lambda)$ 表示 Levy 随机游动搜索路径, 它服从 Levy 概率分布:

$$L(\lambda) \sim \frac{u}{|v|^{1/\lambda}} \cdot \left(\frac{\Gamma(1+\lambda) \sin(\pi \cdot \lambda/2)}{\Gamma\left(\frac{1+\lambda}{2}\right) \cdot \lambda \cdot 2^{(\lambda-1)/2}} \right)^{1/\lambda}, \quad 0 < \lambda \leq 2 \quad (2)$$

其中 $\lambda = 1.5$, 且 u, v 服从标准高斯分布, 即 $u, v \sim N(0, 1)$. 对于进化过程中较差的候选解, 算法采用偏好随机游动方式更新. 寄生宿主按一定概率 P 发现外来的布谷鸟幼雏并舍弃该巢穴后生成相同数量的新巢穴

$$z_{i+1,j} = z_{i,j} + m \cdot (z_{i,k} - z_{i,s}) \quad (3)$$

其中, m 为算法的控制缩放系数, 且 $m \sim U(0, 1)$. $z_{i,k}$ 和 $z_{i,s}$ 分别为算法进化到第 i 代的两个随机候选解.

3 均衡单进化布谷鸟算法 (ESECS)

3.1 均衡单进化评价策略

为便于理解本文提出的均衡单进化评价策略, 假设优化问题的目标函数为 $f = x_1^2 x_3^2 + (x_2 - x_3)^2$, 其全局最优解为 $(0, 0, 0)$, 算法进化的初始数值为 $(3, 2, 2)$. 据此求得初始化目标函数适应度值为 $f(z_{i,j}) = f(3, 2, 2) = 36$. 由 CS 算法可知, 将宿主巢穴位置更新为 $z_{i+1,j} = (1.5, 1.2, -3.0)$, 由于 $f(z_{i+1,j}) = 37.89 > f(z_{i,j}) = 36$, 因此判断该次进化无效, 算法重新执行搜索. 但是, 分析前后两代进化的数据和理论全局最优解 $(0, 0, 0)$ 可知, 第 $i+1$ 代的候选解 $z_{i+1,j}$ 中第 1 维由数值 3 进化为 1.5, 第 2 维由数值 2 进化为 1.2, 这两者均沿全局最优的方向进化, 而仅仅第 3 维度偏离全局最优方向. 由于第 3 维度的退化导致目标函数适应度值变差, 而忽略了第 1 和第 2 维度的进化, 使算法所需的进化代数增加, 寻优速度降低.

均衡单进化评价策略是每一代进化只随机更新目标函数的单个维度, 并且随机更新的维度服从整数均匀分布, 与其它维度组合构成一个新的候选解, 然后评价该候选解, 若优于上一代函数适应度值, 则保留更新的候选解并继续进化, 直至满足算法停止条件. 由于采

用贪婪法则,因此算法只接受能改善当前候选解的更新值,确保了算法在优化过程中搜索方向的针对性调整,并且不会影响算法的效率.为便于理解该评价策略,仍然以上述例子阐述,其全局最优解为(0,0,0),算法进化的初始数值为(3,2,2).首先产生整数均匀分布的随机维度为3,其他维度(本例中分别为1和2)保持不变.将宿主巢穴位置及适应度值更新为 $f(z_{i+1,j}^{(3)}) = f(3, 2, 1.2) = 13.6$.在一个种群内继续随机更新维度,再次产生整数均匀分布的随机维度为1,其他维度(本例中分别为2和3)保持不变.若维度1对应的更新数值为3.5,那么目标函数适应度值更新为 $f(z_{i+2,j}^{(1)}) = f(3.5, 2, 1.2) = 18.28$.此时对种群内所有的候选解进行评价,结果表明: $f(z_{i+2,j}^{(1)}) > f(z_{i+1,j}^{(3)})$,适应度的更新值未能改善当前候选解 $z_{i+1,j}^{(3)} = (3, 2, 1.2)$,因此算法将舍弃当前选择候选解 $z_{i+1,j}^{(3)}$,并进行其他维度的更新.采用贪婪法则的均衡单进化评价策略不会因为某些维度的退化而舍弃全局最优的进化方向,充分利用目标函数的单维度信息引导算法进行精细化搜索,获得更高质量的求解结果.

3.2 ESECS 算法实现

ESECS 算法通过在 Levy 飞行随机游动方式和偏好随机游动方式引入均衡单进化评价策略,避免多维度之间互相干扰.传统的 CS 算法中宿主巢穴只是单纯的按照 Levy 飞行随机游动和偏好随机游动产生候选解,并没有充分利用当前候选解本身的信息寻优.基于此,本文提出了两种新型随机游动步长更新学习律(4)和(5),如下所示:

$$z_{i+1,j}^{(l)} = z_{i,j}^{(l)} + \eta \cdot (z_{i,j}^{(l)} - z_{i,g}^{(l)}) \oplus L^{(l)}(\lambda) \quad (4)$$

$$z_{i+1,j}^{(l)} = z_{i,j}^{(l)} + w \cdot (z_{i,k}^{(l)} - z_{i,s}^{(l)}) \quad (5)$$

其中, $z_{i,g}^{(l)}$ 为第*i*代搜索的最优解 $z_{i,g}$ 对应的维度*l*数值; $z_{i,j}^{(l)}$ 和 $z_{i+1,j}^{(l)}$ 分别代表算法进化至第*i*代和第*i+1*代时的第*j*个候选解 $z_{i,j}$ 和 $z_{i+1,j}$ 对应的维度*l*数值; $z_{i,k}^{(l)}$ 和 $z_{i,s}^{(l)}$ 分别为算法进化到第*i*代的两个随机候选解 $z_{i,k}$ 和 $z_{i,s}$ 对应的维度*l*数值; w 为算法的控制缩放系数,且 $0 < w < 3$.ESECS 算法首先在当前最优解 $z_{i,g}$ 的基础上利用(4)对当前搜索的最优解的第*l*维度 $z_{i,g}^{(l)}$ 以 Levy 飞行随机游动生成新的候选解进行更新,提高算法全局寻优速度,同时利用(5)对当前搜索的最优解的第*l*维度 $z_{i,g}^{(l)}$ 进行更新,将传统 CS 算法的控制缩放系数*m*由随机数修改为(5)中大于0的确定数 w ($0 < w < 3$),以加强局部搜索能力.

ESECS 算法的执行步骤如下所示:

步骤 1 算法初始化,种群为*N*个宿主巢穴,优化问题的目标函数维度为*D*,最大进化代数*M*,发现概率为*P*,算法控制缩放系数为*w*.

步骤 2 计算随机化宿主巢穴位置 $z_{i,j}$,($j = 1, 2, \dots, N$)对应的函数适应度值 $f(z_{i,j})$.

步骤 3 产生均匀随机整数*l*,($1 \leq l \leq D$),利用式(4)的 Levy 随机游动方式进行巢穴位置更新,并按照发现概率*P*舍弃候选解 $z_{i+1,j}$.

步骤 4 采用式(5)的偏好随机游动方式产生新的宿主巢穴,同时比较算法控制缩放系数*w*与发现概率*P*的大小,若 $w > P$,则利用偏好随机游动方式产生一组巢穴位置.反之,与步骤 2 中上一代宿主巢穴位置 $z_{i,g}$ 比较,并替换步骤 3 中被舍弃的候选解 $z_{i+1,j}$.

步骤 5 计算步骤 4 中种群产生的候选解对应的适应度函数值 $f(z_{i+1,j})$,同时更新对应的最优解 $z_{i+1,g}$.

步骤 6 若满足算法终止条件,则输出当前进化的最优值及最优解,并停止算法,否则转向步骤 2 继续执行算法.

4 数值仿真及分析

4.1 测试函数及实验标准

为了验证 ESECS 算法的有效性,本文选择 CEC2005 中 10 个不同类型的基准函数,按其特点分为 2 类,第 1 类是旋转单模函数($f1 \sim f6$),第 2 类为旋转多模函数($f8 \sim f10, f13$).为保证函数标号连续性,将 $f1 \sim f6$ 命名为 F1 ~ F6, $f8 \sim f10$ 命名为 F7 ~ F9, $f13$ 命名为 F10 同时选取 12 个改进 CS 算法及最新的其他优化算法实验,分别为 ICS[25],DDICS[24],PSCS[26],ORCSA[27],mCS[4],HABC[28],ILABC[29],IBA[30],SCQQSO[31],AABCLS[32],ImHS[33],MFA[34].每种算法独立运行次数为 30,种群数目为 25,迭代次数为 10000 次.

4.2 算法收敛精度比较

为全面测试 ESCES 算法的性能,取其平均优化均值作为评价指标.表 1 和表 2 分别为各种改进 CS 算法和其他优秀算法*D*=30 的优化结果.同时,对表 1~2 的实验结果进行统计双侧*t*检验,考察算法的置信度.设置统计检验的置信水平 $\alpha = 0.05$.当 $h = '+'$ 时,表示 ESCES 算法相对于比较算法,具有显著优越性.当 $h = '-'$ 时,表示 ESCES 算法显著差于比较算法.当 $h = '='$ 时,表示 ESCES 算法与比较算法无显著统计学特性.表格中加粗的数值表示算法在相应的测试函数中取得的最好结果.

从表 1~2 的统计结果得出 ESCES 算法在大部分测试函数上都取得了较好的结果.在这些测试函数中,F1~F6 是具有旋转偏移的单模函数,全局最优解有且仅有 1 个.F6 是病态的非凸单模函数,函数变化呈现“马鞍”曲线,并且函数各维度之间互相耦合,具有极强的隐蔽性和欺诈性,传统优化算法很难搜索到全局最

优解,然而 ESCES 算法依然能够搜索到较好的优化解. F4 涵盖单模、偏移、缩放、维度相关和噪声等特点,函数变化曲线随机,并且算法寻优过程无明显规律,其优化难度极大,ESCES 算法是 13 种优化算中唯一搜索到近似全局最优解的算法.

多模函数通常在函数定义域内存在大量局部极值点,当优化算法的种群多样性恶化时,极易出现迟滞现象,使算法陷入局部最优解. 在多模函数优化中,除了 F8 和 F9 外,ESCES 算法的寻优性能均表现优异. F10 是一个复杂的多模问题,由于目标函数只提供了少量信息,使算法很难辨别搜索方向,因此搜索到全局最优解的机会微乎其微,主要用于检验算法的群体多样性. 从多模函数优化结果得出 ESECS 算法具有较好的全局寻优能力.

象,使算法陷入局部最优解. 在多模函数优化中,除了 F8 和 F9 外,ESCES 算法的寻优性能均表现优异. F10 是一个复杂的多模问题,由于目标函数只提供了少量信息,使算法很难辨别搜索方向,因此搜索到全局最优解的机会微乎其微,主要用于检验算法的群体多样性. 从多模函数优化结果得出 ESECS 算法具有较好的全局寻优能力.

表 1 $D=30$ 时各种改进 CS 算法的优化结果

Function	ICS	h	DDICS	h	PSCS	h	ORCSA	h	mCS	h	ESECS
F1	-450.0000	+	-450.0000	+	114.9254	+	-450.0000	+	-450.0000	+	-450
F2	-449.9853	+	311.7213	+	1.39E+04	+	-449.9880	+	-450.0000	+	-450.0000
F3	4.24E+07	+	1.60E+07	+	5.43E+08	+	2.81E+07	+	2.51E+07	+	1.47E+06
F4	1.44E+03	+	1.73E+04	+	3.47E+03	+	578.1917	+	1.95E+04	+	-226.53
F5	2.54E+03	+	4.17E+03	+	3.38E+03	+	3.84E+03	+	2.16E+03	+	2.05E+03
F6	390.0000	+	540.2867	+	7.34E+09	+	390.0000	+	390.0000	-	390.0000
F7	-119.0358	+	-118.9810	+	-119.1269	+	-119.0723	+	-118.8715	+	-119.61
F8	-326.5247	+	-223.8538	+	-322.5124	+	-328.0727	+	-329.1873	-	-316.08
F9	-252.2426	-	-159.0464	+	-42.5505	+	-172.9369	+	-160.2355	+	-196.40
F10	-127.3961	+	-120.2487	+	-128.5544	+	-124.0654	+	-128.0978	+	-128.89

表 2 $D=30$ 时其他优秀算法的优化结果

Function	HABC	h	ILABC	h	AABCLC	h	SCQQSO	h	IBA	h	ImHS	h	MFA	h	ESECS
F1	-450.00	+	3.67E+05	+	-450.00	+	-450.00	+	-450.00	+	-450.00	+	-450.00	+	-450
F2	4.78E+05	+	6.12E+05	+	4.72E+05	+	-450.00	+	-450.00	+	-90.00	+	-450.00	-	-450.00
F3	2.42E+09	+	7.52E+09	+	6.92E+09	+	6.97E+06	+	3.02E+07	+	4.60E+07	+	1.25E+08	+	1.47E+06
F4	1.50E+04	+	9.47E+04	+	6.76E+04	+	3.75E+02	+	3.45E+02	+	8.86E+02	+	4.52E+03	+	-2.27E+02
F5	2.73E+05	+	1.75E+05	+	6.19E+05	+	1.47E+03	-	5.56E+03	+	3.52E+03	+	3.96E+04	+	2.052E+03
F6	4.26E+04	+	1.45E+04	+	2.92E+04	+	390.00	+	390.00	+	390.00	+	390.00	-	390.00
F7	-118.83	+	-119.01	+	-118.59	+	-119.05	+	-119.02	+	-118.98	+	-119.00	+	-119.61
F8	-317.53	+	2.48E+06	+	-305.01	+	-330.00	+	-330.00	-	-298.18	+	-303.28	+	-316.08
F9	1.58E+03	+	1.61E+03	+	1.03E+03	+	-94.08	+	-260.27	-	-149.74	+	-78.39	+	-196.40
F10	-117.17	+	5.64E+07	+	-126.89	+	-127.68	+	-125.91	+	-128.52	+	-123.88	+	-128.89

由表 1 可知,除了 F8 和 F9 外,ESCES 算法均搜索到更好的全局最优解. 对比算法 mCS 在单维度和多维度函数优化中表现较好,仅次于 ESCES 算法. 由表 2 可知,与其他目前最新的优化算法相比,ESCES 算法的综合表现最好. 除了 F5, F8 和 F9 外,ESCES 算法搜索到的平均适应度值最优. ESCES 算法优化 F8 和 F9 时没能搜索到最优解的主要原因是 F8 和 F9 的全局最优解处于算法初始化随机维度对应的区间之外,导致均衡单进化函数评价策略不能进行有效的局部搜索,影响寻优解的质量. IBA 算法在单维度和多维度函数优化中性能较好,仅次于 ESCES 算法. SCQQSO 算法在单维度函数优化中表现最佳. 通过 13 种算法的显著性统计结果得出 ESCES 算法显著占优的比率最大,其具有更优秀的搜索性能.

4.3 算法收敛速度比较

限于文章篇幅,选取 F1、F2、F4 和 F8 测试 13 种优化算法的收敛速度,实验结果如图 1 所示. 对于 F2, DDICS 算法和 ESCES 算法的搜索速度和收敛精度优于其它改进 CS 算法. ESCES 算法初始收敛速度小于 DDICS 算法,但是在迭代中后期的收敛速度大于 DDICS 算法,这也体现了 ESCES 算法的优越性. 对于 F4 和 F8, ESCES 算法的搜索速度显著优于其它优化算法. 从整体上分析,ESCES 算法提出的均衡单进化评价策略不仅注重整体进化结果,而且关注进化过程中各个维度的进化方向,强化全局勘探能力的同时提升局部寻优能力,求解质量因而得到显著提高.

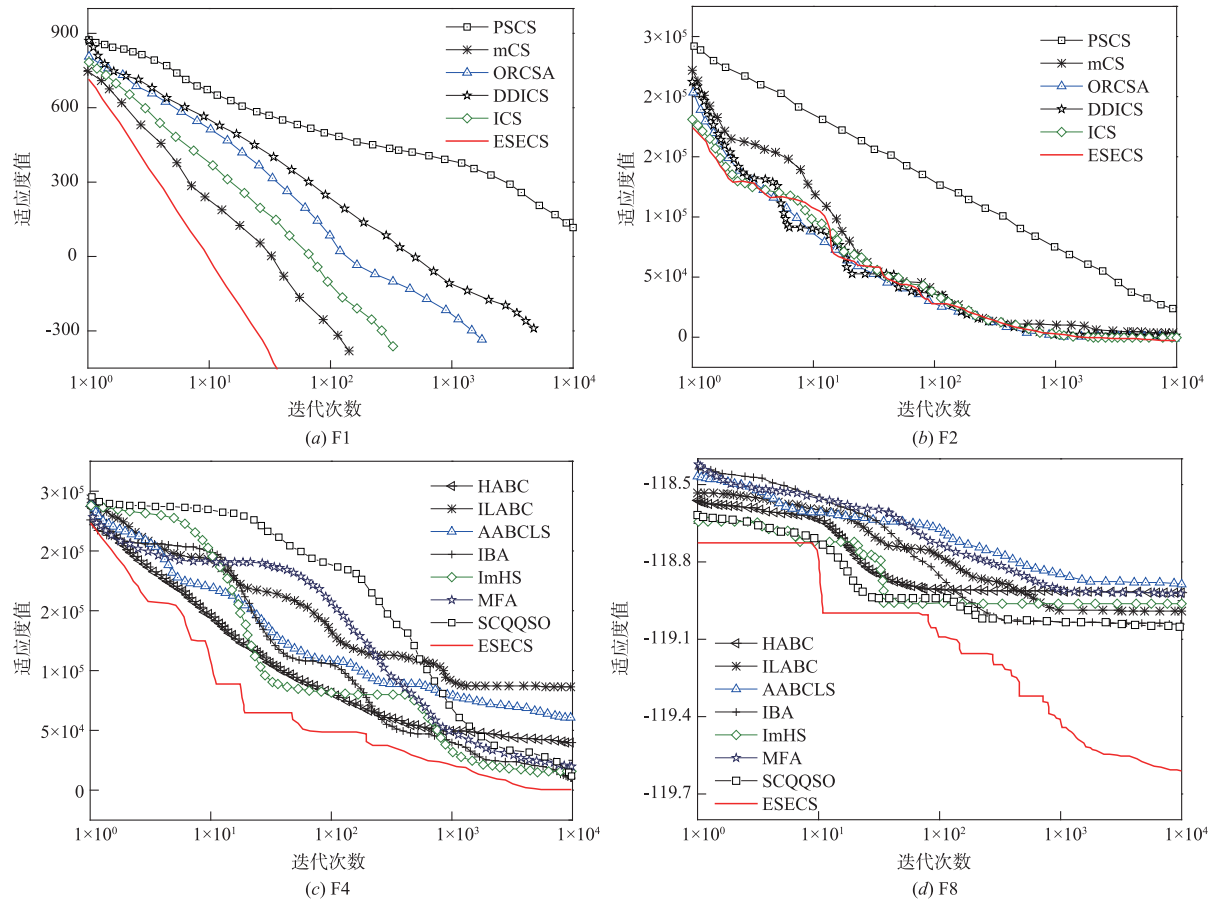


图1 优化算法在部分测试函数的适应度值收敛特性

4.4 ESECS 算法复杂度分析

优化算法的计算量主要集中在目标函数的评价阶段,其算法复杂度主要体现为目标函数的评价次数. 设置算法最大进化代数 M ,目标函数的维度为 D . 基本CS算法的计算复杂度为 $T(CS) = O(M \times (f(1:D) + N))$. 由ESECS算法的执行步骤可知,相比于基本CS算法,步骤3中计算随机整数 l 是增加的步骤,其复杂度为 $T(l) = O(M \times N)$. 因此,ESECS算法的计算复杂度为 $T(ESECS) = T(CS) + T(l) = O(M \times (f(1:D) + N))$. 由此可知,ESECS算法与基本CS算法的复杂度一致,同属于一个数量级,并未增加算法复杂度.

5 结论

布谷鸟算法是一种新型的智能优化算法,绝大部分文献在衡量算法进化的有效性均采用目标函数整体评价的策略. 这种评价策略对于单维度优化问题效果较为有效,但对于多维度自变量相关问题,由于存在维度间相互耦合现象,因此会制约算法的搜索速度和收敛精度. 本文引入一种新型的均衡单进化函数评价策略,避免多维度之间互相干扰. 同时,提出了两种新型随机游动步长更新学习律,提高了优化算法的全局搜

索速度和收敛精度. 为了考察算法的综合性能,本文对10个国际通用的基准测试函数进行了性能仿真,结果表明,提出的ESECS算法比改进CS算法及其他最新智能优化算法具有更优秀的全局寻优性能.

参考文献

- [1] Yang X S, Deb S. Engineering optimization by Cuckoo search[J]. International Journal of Mathematical Modelling & Numerical Optimisation, 2010, 1(4): 330-343.
- [2] Basu M, Chowdhury A. Cuckoo search algorithm for economic dispatch[J]. Energy, 2013, 60(7): 99-108.
- [3] El-Maleh A H, Sait S M, Bala A. State assignment for area minimization of sequential circuits based on cuckoo search optimization[J]. Computers & Electrical Engineering, 2015, 44(14): 13-23.
- [4] Zhu X, Wang N. Cuckoo search algorithm with membrane communication mechanism for modeling overhead crane systems using RBF neural networks[J]. Applied Soft Computing, 2017, 56: 458-471.
- [5] Xiao L, Shao W, Yu M, et al. Research and application of a hybrid wavelet neural network model with the improved cuckoo search algorithm for electrical power system fore-

- casting[J]. *Applied Energy*, 2017, 198: 203 – 222.
- [6] Sun W, Sun J. Daily PM_{2.5} concentration prediction based on principal component analysis and LSSVM optimized by cuckoo search algorithm[J]. *Journal of Environmental Management*, 2017, 188(1): 144 – 152.
- [7] Aziz M A E. Source localization using TDOA and FDOA measurements based on modified cuckoo search algorithm [J]. *Wireless Networks*, 2017, 23(2): 487 – 495.
- [8] Radovan R. Bulatovic, Stevan R. Dordevic, Vladimir S. Dordevic. Cuckoo search algorithm: A metaheuristic approach to solving the problem of optimum synthesis of a six-bar double dwell linkage[J]. *Mechanism and Machine Theory*, 2013, 61(1): 1 – 13.
- [9] Han W, Xu J, Zhou M, et al. Cuckoo search and particle filter-based inversing approach to estimating defects via magnetic flux leakage signals[J]. *IEEE Transactions on Magnetics*, 2016, 52(4): 1 – 11.
- [10] Srivastav A, Agrawal S. Multi-objective optimization of slow moving inventory system using cuckoo search[J]. *Intelligent Automation & Soft Computing*, 2017, 3(6): 1 – 7.
- [11] Yamany W, El-Bendary N, Hassanien A E, et al. Multi-objective cuckoo search optimization for dimensionality reduction[J]. *Procedia Computer Science*, 2016, 96(8): 207 – 215.
- [12] Wang Z, Li Y. Irreversibility analysis for optimization design of plate fin heat exchangers using a multi-objective cuckoo search algorithm[J]. *Energy Conversion & Management*, 2015, 10(1): 126 – 135.
- [13] Piechocki J, Ambroziak D, Palkowski A, et al. Use of modified cuckoo search algorithm in the design process of integrated power systems for modern and energy self-sufficient farms[J]. *Applied Energy*, 2014, 114(114): 901 – 908.
- [14] Nadjemi O, Nacer T, Hamidat A, et al. Optimal hybrid PV/wind energy system sizing: Application of cuckoo search algorithm for Algerian dairy farms[J]. *Renewable & Sustainable Energy Reviews*, 2017, 70: 1352 – 1365.
- [15] Walton S, Hassan O, Morgan K, et al. Modified cuckoo search: A new gradient free optimization algorithm[J]. *Chaos Solitons & Fractals*, 2011, 44(9): 710 – 718.
- [16] Valian E, Tavakoli S, Mohanna S, et al. Improved cuckoo search for reliability optimization problems[J]. *Computers & Industrial Engineering*, 2013, 64(1): 459 – 468.
- [17] Li X T, Yin M H. Modified cuckoo search algorithm with self adaptive parameter method[J]. *Information sciences*, 2015, 298(12): 80 – 97.
- [18] Walton S, Hassan O, Morgan K, et al. Modified cuckoo search: A new gradient free optimisation algorithm[J]. *Chaos Solitons & Fractals*, 2011, 44(9): 710 – 718.
- [19] Fateen S E K, Bonilla-Petriciolet A. A note on effective phase stability calculations using a Gradient-Based Cuckoo Search algorithm [J]. *Fluid Phase Equilibria*, 2014, 375(6): 360 – 366.
- [20] Xiao L, Shao W, Yu M, et al. Research and application of a hybrid wavelet neural network model with the improved cuckoo search algorithm for electrical power system forecasting[J]. *Applied Energy*, 2017, 198: 203 – 222.
- [21] Cheung N J, Ding X M, Shen H B. A nonhomogeneous cuckoo search algorithm based on quantum mechanism for real parameter optimization[J]. *IEEE Transactions on Cybernetics*, 2017, 47(2): 391 – 402.
- [22] Zhong Y, Liu X, Wang L, et al. Particle swarm optimisation algorithm with iterative improvement strategy for multi-dimensional function optimisation problems[J]. *International Journal of Innovative Computing & Applications*, 2012, 4(3): 223 – 232.
- [23] Ren W J, Pan Q K, Liang J J. An improved harmony search algorithm for multi-dimensional function optimization problem[A]. *IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications [C]*. IEEE, 2010. 391 – 395.
- [24] 王季进, 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.
- Wang L J, Yi Y L, Zhong Y W. Cuckoo search algorithm with dimension by dimension improvement[J]. *Journal of Software*, 2013, 24(11): 2687 – 2698. (in Chinese)
- [25] Valian E, Tavakoli S, Mohanna S, et al. Improved cuckoo search for reliability optimization problems[J]. *Computers & Industrial Engineering*, 2013, 64(1): 459 – 468.
- [26] Li X, Yin M. A particle swarm inspired cuckoo search algorithm for real parameter optimization[J]. *Soft Computing*, 2016, 20(4): 1389 – 1413.
- [27] Nguyen T T, Vo D N. The application of one rank cuckoo search algorithm for solving economic load dispatch problems [J]. *Applied Soft Computing*, 2015, 37(C): 763 – 773.
- [28] Ma L B, Zhu Y L, Zhang D Y, et al. A hybrid approach to artificial bee colony algorithm[J]. *Neural Computing & Applications*, 2016, 27(2): 387 – 409.
- [29] Gao W F, Huang L L, Liu S Y, et al. Artificial bee colony algorithm based on information learning[J]. *IEEE Transactions on Cybernetics*, 2015, 45(12): 2827 – 2939.
- [30] Wu Z, Yu D. Application of improved bat algorithm for solar PV maximum power point tracking under partially shaded condition[J]. *Applied Soft Computing*, 2018, 62(C): 101 – 109.
- [31] Li Y, Bai X, Jiao L, et al. Partitioned-cooperative quan-

tum-behaved particle swarm optimization based on multi-level thresholding applied to medical image segmentation [J]. *Applied Soft Computing*, 2017, 56(C): 345 – 356.

- [32] Singh S, Jagdish J, Bansal C, et al. Accelerating artificial bee colony algorithm with adaptive local search [J], *Memetic Computing*, 2015, 7(3): 215 – 230.
- [33] Portilla-Flores E A, Sánchez-Márquez A, Flores-Pulido L, et al. Enhancing the harmony search algorithm performance on constrained numerical optimization [J]. *IEEE Access*, 2017, (99): 1 – 21.
- [34] He L, Huang S. Modified firefly algorithm based multilevel thresholding for color image segmentation [J]. *Neurocomputing*, 2017, 240: 152 – 174.

作者简介



傅文渊 男, 1982 年 9 月出生, 福建邵武人, 讲师, 主要从事智能信号优化与智能学习控制、电路与系统设计及嵌入式系统设计等方面的研究工作。

E-mail: fwy@hqu.edu.cn