

2-19-2020

Cuckoo Search Algorithm with Dynamic Step and Discovery Probability

Jingsen Liu

1. Institute of Intelligent Network System, Henan University, Kaifeng 475004, China;;2. College of Software, Henan University, Kaifeng 475004, China;;

Xiaozhen Liu

2. College of Software, Henan University, Kaifeng 475004, China;;

Li Yu

3. Institute of Management Science and Engineering, Henan University, Kaifeng 475004, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Cuckoo Search Algorithm with Dynamic Step and Discovery Probability

Abstract

Abstract: In order to further improve the low accuracy and slow convergence speed of algorithm search, a cuckoo search algorithm with dynamic step size and probability of discovery is proposed. *The algorithm dynamically constrains the Levy's moving step of each generation by introducing the step adjustment factor, which makes the Levy's flight mechanism adaptive. In the probability of finding, the random inertia weight with uniform distribution and F distribution is used to change the fixed value of the probability of discovery, to strengthen the diversity of the population and to keep the balance between global search and local exploration.* The experiment result proves that the proposed algorithm has a good feasibility, and the optimization results and the convergence speed of the algorithm increase.

Keywords

cuckoo search algorithm, step adjustment factor, levy flight, adaptive, random inertia weight

Recommended Citation

Liu Jingsen, Liu Xiaozhen, Li Yu. Cuckoo Search Algorithm with Dynamic Step and Discovery Probability[J]. Journal of System Simulation, 2020, 32(2): 289-298.

具有动态步长和发现概率的布谷鸟搜索算法

刘景森^{1,2}, 刘晓珍², 李煜^{3*}(1. 河南大学智能网络系统研究所, 河南 开封 475004; 2. 河南大学软件学院, 河南 开封 475004;
3. 河南大学管理科学与工程研究所, 河南 开封 475004)

摘要: 为了进一步改善算法搜索过程中存在的求解精度偏低、收敛速度缓慢等现象, 提出具有动态步长和发现概率的布谷鸟搜索算法。该算法通过引入步长调整因子动态约束每一代种群的莱维移动步长, 使算法的莱维飞行机制具有自适应性。在发现概率上, 使用具有均匀分布和 F 分布特性的随机惯性权重, 改变发现概率的固定取值, 加强种群的多样性, 保持算法全局搜索、局部探索之间的平衡状态。通过实验证明, 所提算法具有良好的可行性, 其寻优结果、收敛速度均有提高。

关键词: 布谷鸟搜索算法; 步长调整因子; 莱维飞行; 自适应; 随机惯性权重

中图分类号: TP301.6 文献标识码: A 文章编号: 1004-731X (2020) 02-0289-10

DOI: 10.16182/j.issn1004731x.joss.17-9093

Cuckoo Search Algorithm with Dynamic Step and Discovery Probability

Liu Jingsen^{1,2}, Liu Xiaozhen², Li Yu^{3*}(1. Institute of Intelligent Network System, Henan University, Kaifeng 475004, China; 2. College of Software, Henan University, Kaifeng 475004, China;
3. Institute of Management Science and Engineering, Henan University, Kaifeng 475004, China)

Abstract: In order to further improve the low accuracy and slow convergence speed of algorithm search, a cuckoo search algorithm with dynamic step size and probability of discovery is proposed. *The algorithm dynamically constrains the Levy's moving step of each generation by introducing the step adjustment factor, which makes the Levy's flight mechanism adaptive. In the probability of finding, the random inertia weight with uniform distribution and F distribution is used to change the fixed value of the probability of discovery, to strengthen the diversity of the population and to keep the balance between global search and local exploration.* The experiment result proves that the proposed algorithm has a good feasibility, and the optimization results and the convergence speed of the algorithm increase.

Keywords: cuckoo search algorithm; step adjustment factor; levy flight; adaptive; random inertia weight

引言

由于优化问题的复杂性导致传统数值计算方法自身缺陷暴露, 难以满足人们的需求。研究者们受到生物群体自然现象规律的启发, 提出了诸多基

于仿生学的智能优化算法, 如: 由自然界生物进化过程为启发的遗传算法^[1], 由生物群体内个体间的合作与竞争为启发的差分进化算法^[2], 由自然界中蚂蚁集体寻径为启发的蚁群算法^[3], 由鸟群觅食中的迁徙过程为启发的粒子群算法^[4-5], 由果蝇觅食行为推演出的果蝇算法^[6], 由萤火虫发光的生物特性而提出的萤火虫算法^[7-8]等。这些仿生算法不仅成功地应用于复杂优化问题方面, 还为此问题的解决提供了新思路。



收稿日期: 2017-11-20 修回日期: 2018-01-16;
基金项目: 河南省科技攻关重点项目(162102110109);
作者简介: 刘景森(1968-), 男, 河南开封, 博士, 教授, 研究方向为智能算法、网络信息安全等; 刘晓珍(1992-), 女, 河南濮阳, 硕士, 研究方向为智能算法; 李煜(通讯作者 1969-), 女, 河南开封, 博士, 教授, 研究方向为智能算法、电子商务等。

<http://www.china-simulation.com>

2009 年, Yang 和 Deb 提出一种模拟自然界布谷鸟寄生习性的新型启发式算法——布谷鸟搜索算法^[9-10]。作为一种新兴计算技术, 算法模型中通过莱维飞行、随机游走两大机制来有效地进行全局及局部搜索。目前, 该算法的应用领域极为广泛, 如: 函数优化^[11]、图像处理^[12]、电力系统优化^[13]、人脸识别^[14]、混沌系统参数估计^[15]等。

布谷鸟搜索算法一方面因其操作模型简单、参数设置少、通用性强等独特优势, 受到广大研究者的青睐; 另一方面因算法搜索过程中存在的收敛速度慢, 易陷入局部解等情况, 引起国内外学者的广泛关注, 并且人们对原始算法做出了相应改进。如: Valian 等^[16]通过构建衰减指数函数与线性递减函数分别动态调整步长因子与发现概率, 提升了算法的整体寻优性能。马卫等^[17]将模式搜索、自适应竞争排名构建、合作分享 3 大机制引入布谷鸟搜索算法中, 优化了算法的求解性能, 并在解决多峰及高维函数优化问题中取得良好成效。Layeb 等^[18]将一些量子计算概念引入算法中, 增强了算法种群的多样性, 并在求解集装箱问题上得到较好地应用。贾云璐等^[19]根据反馈控制原理, 将年龄结构、变异成功率等种群特征作为反馈信息引入到 CS 算法中, 增强算法的寻优能力。贺兴时等^[20]在布谷鸟搜索算法中引入小生境技术的逐步档案缩减法, 提升了寻找 Pareto 解集的收敛速度, 并较好地应用于求解多目标问题。上述文献的改进使算法在各自应用领域的求解性能均有提升, 但布谷鸟搜索算法仍有进一步改进的空间。本文提出一种具有动态步长和发现概率的布谷鸟搜索算法(Cuckoo search algorithm with dynamic step and discovery probability, DCS)。改进后的算法, 首先在莱维飞行中引入调整因子, 动态调整搜索步长, 加快算法的收敛速度, 提高求解精度; 然后对发现概率采用动态惯性权重策略, 增强算法种群的多样性, 减少陷入局部极值的次数, 有效平衡算法全局与局部搜索能力; 最后为了全面测试改进后算法的有效性, 文章选取多个经典目标函数及算法进行仿真实验设计。

1 基本布谷鸟搜索算法

布谷鸟具有一种特殊的孵育行为, 它们从不构建鸟窝去孵化幼卵, 而是寻找与其生活习性相似的鸟类, 让它们代替自己孵化幼卵, 但是这些鸟蛋一旦被宿主发现, 宿主便抛弃这些鸟蛋或新筑鸟巢。为了简化自然界中布谷鸟的繁衍习性, Yang 等将布谷鸟的产卵行为假设为 3 个理想形态^[9]:

- (1) 将每个鸟所产的唯一鸟蛋随机放入一个宿主巢穴中。
- (2) 将具有优质鸟蛋的鸟窝保留到下一代。
- (3) 可供选择的鸟窝数目是固定的, 并且存放的鸟蛋一旦被主人发现将被淘汰。此时, 外来鸟蛋被发现的概率为 P_α 。

基于以上假设, 可将布谷鸟探索优质鸟巢的路径与位置更新公式设为:

$$x_i^{t+1} = x_i^t + \delta \oplus Levy(\lambda) \quad (1)$$

式中: x_i^t 为第 $i(i=1, 2, \dots, n)$ 个鸟巢在第 t 代的位置; δ 为步长因子, 用来控制步长大小, $\delta = \delta_0(x_i^t - x_{best})$, δ_0 为常数($\delta_0=0.01$); x_{best} 为当前最优解; \oplus 为点对点乘法; $levy(\lambda)$ 为服从莱维分布的行走步长:

$$Levy(\lambda) \sim \mu = t^{-\beta}, 1 < \beta \leq 3 \quad (2)$$

算法中产生莱维随机数^[10]的公式为:

$$Levy(\lambda) = \frac{\phi \times \mu}{|v|^{1/\beta}} \quad (3)$$

式中: μ, v 服从标准正态分布, $\beta=1.5$ 。

$$\phi = \left(\frac{\Gamma(1+\beta) \times \sin(\pi \times \frac{\beta}{2})}{\Gamma\left(\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}\right)} \right)^{\frac{1}{\beta}} \quad (4)$$

综上所述, 布谷鸟位置更新公式可表示为:

$$x_i^{t+1} = x_i^t + \delta_0 \frac{\phi \times \mu}{|v|^{1/\beta}} (x_i^t - x_{best}) \quad (5)$$

若宿主发现外来鸟蛋后, 依据概率 P_α 丢弃部分鸟蛋, 然后利用偏好随机游走策略生成同等数量的新解:

$$x_i^{t+1} = x_i^t + \gamma(x_g^t - x_k^t) \quad (6)$$

式中: $\gamma \in [0,1]$ 且服从均匀分布; x'_g, x'_k 为在第 t 代种群中随机挑选的两个位置。

2 改进后的布谷鸟搜索算法

2.1 具有调整因子的自适应步长策略

基本布谷鸟搜索算法的莱维飞行机制是一种服从莱维分布的随机搜索方式,即寻优路径中短步长与长步长交替出现。算法前期,偶然出现的大步长可以很好地促使算法跳出局部极值,快速锁定最优值所在范围,增强算法的收敛速度与寻优能力;但是,在算法后期,搜索逐渐趋向于最优解,若此时步长随机性太强,易使算法跳离全局最优解,降低寻优精度质量。基本布谷鸟搜索算法中,采用步长因子 δ 去控制步长移动大小,若 δ 取值偏大,则算法全局搜索能力增强,但寻优精度降低;若 δ 取值偏小,则局部搜索精细,但总迭代次数增加,收敛性变差。因此,式(1)中 δ 的固定取值在寻优过程中无法动态调整步长大小。基于上述分析,为提高算法收敛速度、寻优精度及有效平衡全局与局部搜索之间的关系,本文提出一种拥有非线性递减思想的步长调整函数,改变式(5)中 δ_0 的固定取值,其表达式为:

$$\tau = \tau_{\min} + (\tau_{\max} - \tau_{\min}) \times (1 - \frac{t}{T})^\rho \quad (7)$$

改进后的位置更新公式如下:

$$x_i^{t+1} = x_i^t + \tau \times \frac{\phi \times \mu}{|v|^\beta} (x_i^t - x_{\text{best}}) \quad (8)$$

式中: τ_{\min} 和 τ_{\max} 分别为步长调整因子的最小值和最大值; t 为当前迭代数; T 为算法的最大迭代数; 指数 ρ 为常数,经大量实验测试 $\rho=0.8$ 时,算法效果更佳;式(7)第二项通过 $(1-t/T)^\rho$,使得 τ 在区间 $[\tau_{\min}, \tau_{\max}]$ 上的取值随迭代次数的增加而逐渐减小,从而使莱维飞行策略具有自适应特性。在算法前期,布谷鸟因获得较大步长快速锁定鸟巢最优区域,加速算法收敛;后期因获得较小步长,在最优解附近深入探索,增强寻优精度。

2.2 具有动态惯性权重的发现概率

通过分析原始布谷鸟搜索算法,发现寄身鸟蛋被鸟巢主人所发现的概率(鸟窝被淘汰概率) P_α 通常采用固定取值模式,但在自然界中,由于各种不确定因素的影响鸟窝被发现的概率并不相同。因此,算法中鸟窝被淘汰概率的固定取值会降低种群规模的多样性,从而不利于该算法平衡全局与局部寻优的关系。为了保持全局搜索与局部搜索两者之间的平衡状态,本文将动态惯性权重策略引入发现概率中,改变原有的固定取值模式。其具体表达公式为:

$$\begin{cases} \text{new}P_\alpha = P_\alpha \times \omega \\ \omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times r - \varphi \times \text{frnd}() \end{cases} \quad (9)$$

式中: P_α 为固定取值; $\text{new}P_\alpha$ 为动态发现概率,且 $\text{new}P_\alpha \in [0.2, 0.5]$; ω_{\max} 和 ω_{\min} 为 ω 的最大值和最小值; $r \in (0,1)$ 上的均匀分布; φ 为用来控制惯性权重偏离程度的常数; $\text{frnd}()$ 为服从非对称性 F 分布产生的随机数。

式(9)第 2 项利用服从均匀分布的 r 使 ω 在区间 $[\omega_{\min}, \omega_{\max}]$ 上灵活取值,第 3 项加入偏离因子 φ 利用 F 分布覆盖率的趋向性、不均匀性、非对称性,使惯性权重合理偏离取值更均匀、更灵活、更多样。依据以上分析,式(9)中取值较大的惯性权重与取值偏小的惯性权重随机交替出现,当 ω 取值较小时,种群多样性增强,算法不但可以避免陷入局部解的情况而且精细搜索能力也随之增强;当 ω 偏大时,此时鸟窝更新数量较小,算法收敛加速。总体上讲,具有此惯性权重的发现概率参数设置,不仅可以动态调整鸟巢的发现概率,还保持了种群的活跃性。同时,一定程度上减少了算法后期收敛速度慢、易陷入局部极小及搜索精度偏低的现象。

此外,鸟巢被发现后基本布谷鸟搜索算法采用双随机解的方式产生新解,这样可能会导致局部搜索随机性太强,为了进一步提升算法局部寻优能力,在本文算法中对被发现鸟巢进行位置更新时引入定向变异机制^[21-22],即用当前的解向量作为目标

向量来代替一个随机向量,从而避免较大的随机性所带来的低效搜索。此时文章引用下式在偏好随机游动机制中产生新位置:

$$x_i^{t+1} = x_i^t + \gamma(x_g^t - x_i^t) \quad (10)$$

2.3 算法流程

step 1: 设置算法参数: 种群规模、迭代次数、维度、发现概率、界值大小, 并随机初始化 N 个鸟窝的位置;

step 2: 利用目标函数求出每个个体的适应度, 将具有最好适应度的解选出, 并记录其位置;

step 3: 每个鸟窝应用式(8)进行位置更新;

step 4: 求出新解的适应度值, 若所求结果较好, 代替上代的旧解;

step 5: 使用随机数 r 与鸟窝被淘汰概率式(9)进行比较, 如果 $r > \text{new}P_\alpha$, 则用式(10)产生同等数量的新解替代被淘汰的解;

step 6: 再次计算全部个体的适应度, 选出并记录全局最优解;

step 7: 判断是否满足算法设置的迭代次数: 若满足, 结束搜索过程, 输出最优结果; 否则, 继续 step 3~step 6 的迭代优化。

2.4 DCS 算法的收敛性分析

收敛性分析为验证算法有效性提供了理论依据, 因此许多学者在研究过程中也对一些算法做了相应分析。如: 任子晖等^[23]、王凡等^[24]通过分析马尔科夫链的一系列性质分别对粒子群算法、布谷鸟搜索算法的全局收敛性进行有效分析; 文献[25-26]则采用二阶常系数非齐次差分方程分别对蝙蝠算法、粒子群算法的收敛性进行分析。文章为了进一步说明改进后算法的可行性, 则采用差分方程法分析 DCS 算法的全局收敛性。

改进后布谷鸟搜索算法采用式(8)进行位置更新, 为简化计算: 设公式(3)为 r_1 , x_{best} 记为 b , 将第 $t+1$ 代的最优位置记为 b_1 , 式(8)可简化为:

$$x(t+1) = x(t) + \tau \times r_1(x(t) - b) \quad (11)$$

$$x(t+2) = x(t+1) + \tau \times r_1(x(t+1) - b_1) \quad (12)$$

由式(11)和式(12)可得:

$$x(t+2) - \tau r_1 x(t+1) - (1 + \tau r_1)x(t) = -\tau r_1(b + b_1) \quad (13)$$

其式(13)的特征方程为:

$$\lambda^2 - \tau r_1 \lambda - (1 + \tau r_1) = 0$$

则 $\Delta = (-\tau r_1)^2 + 4(1 + \tau r_1) = (\tau r_1 + 2)^2 \geq 0$, 所以只需考虑以下 2 种情况:

1) 当 $\Delta=0$ 时, $\lambda = \lambda_1 = \lambda_2 = \tau r_1/2$, 此时 $x(t) = A_0 + A_1 t$, A_0 、 A_1 为待定系数, 计算可得:

$$\begin{cases} A_0 = x(0) \\ A_1 = \frac{(2 + \tau r_1)x(0) - 2\tau r_1 b}{\tau r_1} \end{cases}$$

2) 当 $\Delta > 0$ 时, $\lambda_{1,2} = (\tau r_1 \pm \sqrt{\Delta})/2$, 此时 $x(t) = A_0 + A_1 \lambda_1^t + A_2 \lambda_2^t$, A_0 、 A_1 、 A_2 为待定系数, 计算可得:

$$\begin{cases} A_0 = x(0) \\ A_1 = \frac{\lambda_2 x(0) - (1 + \lambda_2)x(1) + x(2)}{(\lambda_2 - \lambda_1)(1 - \lambda_1)} \\ A_2 = \frac{\lambda_1 x(0) - (1 + \lambda_1)x(1) + x(2)}{(\lambda_1 - \lambda_2)(1 - \lambda_2)} \end{cases}$$

其中

$$x(1) = (1 + \tau r_1)x(0) - \tau r_1 b$$

$$x(2) = (1 + \tau r_1)^2 x(0) - [(1 + \tau r_1)b + b_1] \tau r_1$$

根据上述分析, 若要证明算法迭代收敛, 则需满足 $t \rightarrow \infty$ 时 $x(t)$ 可趋近于某一特定值。因此, 若想满足以上两种情况的 $x(t)$ 均收敛, 则必须使 $\|\lambda_1\| < 1$ 、 $\|\lambda_2\| < 1$ ^[25]。经计算可得以下结论:

当 $\Delta=0$ 时, 收敛区域需满足: $\tau r_1 + 2 = 0$;

当 $\Delta > 0$ 时, 收敛区域需满足: $\tau r_1 + 2 > 0$ 且 $|\lambda_{1,2}| < 1$, 则 $-2 < \tau r_1 < 2$ 。

综合上述 2 种情况, 收敛区域为 $-2 \leq \tau r_1 < 2$ 。

3 仿真实验

为了充分证明改进后算法的有效性与可行性, 选取 8 个经典测试函数通过原始布谷鸟搜索算法(Cuckoo search algorithm, CS)^[9]、Improved cuckoo search for reliability optimization problems(ICS)^[16]、粒子群算法(Particle Swarm Optimization, PSO)^[4]

及文章提出的 DCS 算法进行对比实验。具体测试函数如下:

(1) Sphere 函数:

$$f_1(x) = \sum_{i=1}^n x_i^2$$

该函数在(0,...,0)处取得最小值 0。

(2) Schwefel's problem 2.22 函数:

$$f_2(x) = \sum_{i=1}^n |x_i| + \prod_{i=1}^n |x_i|$$

该函数在(0,...,0)处取得最优值 0。

(3) Griewank 函数:

$$f_3(x) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

该函数在(0,...,0)处取得最小值 0。

(4) Schwefel's problem 2.21 函数:

$$f_4(x) = \max_i \{|x_i|, 1 \leq i \leq n\}$$

该函数在(0,...,0)处取得最小值 0。

(5) Zakharov 函数:

$$f_5(x) = \sum_{i=1}^n x_i^2 + \left(\frac{1}{2} \sum_{i=1}^n ix_i\right)^2 + \left(\frac{1}{2} \sum_{i=1}^n ix_i\right)^4$$

该函数在(0,...,0)处取得最小值 0。

(6) Branins'srcos 函数:

$$f_6(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right) \cos(x_1) + 10$$

该函数在 $(-\pi, 12.275)$, $(\pi, 2.275)$, $(9.424, 78, 2.475)$ 处取得最小值 0.397 887。

(7) Six-Hump Camel-Back 函数:

$$f_7(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$

该函数在 $(-0.089, 0.712)$, $(0.089, -0.712)$ 处取得小值-1.031 628 5。

(8) Hansen 函数:

$$f_8(x) = \sum_{i=1}^n i \times \cos((i-1)x_1 + i) \sum_{j=1}^n j \times \cos((j+1)x_2 + j)$$

该函数在 $(-7.589, 593, -7.708, 314)$, $(-7.589, 893, -1.425, 128)$, $(-7.589, 593, 4.858, 057)$, $(-1.306, 708,$

$-7.708, 314)$, $(-1.306, 708, -1.425, 128)$, $(-1.306, 708, 4.858, 057)$, $(4.976, 478, -7.708, 314)$, $(4.976, 478, 4.858, 057)$, $(4.976, 478, -1.425, 128)$ 处取得最小值-176.541 793。

上述 8 个测试函数中, $f_1(x)$, $f_2(x)$, $f_4(x)$ 为单峰函数, 常用于验证算法的收敛速度与寻优精度。 $f_3(x)$, $f_5(x)$ ~ $f_8(x)$ 为复杂多峰函数, 存在较多的局部极小值, 可有效地测试算法跳离局部极值的情况及算法的全局收敛性能。其中, $f_3(x)$, $f_5(x)$ 函数局部极值众多, 易早熟收敛, 可评价算法的全局探索能力。 $f_6(x)$ ~ $f_8(x)$ 函数均有多个全局最优点和若干局部极小, 对于测试算法收敛能力较强。因此, 以上选取的函数具有各自的独特性, 适合对算法性能的全面测试。

仿真实验是充分验证算法是否可行的依据, 因此大量的实验是测试的根本。由于篇幅有限, 文章给出了上述 8 个不同复杂程度的测试函数, 来验证本文 DCS 算法的可行性。但为了更全面的阐述改进后算法的有效性, 文章在 2.4 小节中也对改进算法的收敛性做了理论分析, 证明了本文 DCS 算法的全局收敛性。同时, 文章的 8 个测试函数的特性及测试性能各不相同, 可全面测试本文算法性能。

3.1 实验环境及参数设置

为了进一步验证本文算法的优越性, 在保证客观、公平的前提下, CS、ICS、PSO、DCS 算法均采用 Windows7 运行环境、Matlab R2014a 编程环境。实验过程中, 各算法参数设置详情为:

CS 算法, 种群规模 $N=25$; 总迭代次数 $T=1000$; 鸟巢被发现概率取值 $P_a=0.25$; 算法维数 $D=2/10/30$ 。

ICS 算法是布谷鸟搜索算法中的经典改进算法, 且此算法也对莱维飞行及发现概率进行了改进。实验中算法的种群规模、最大迭代次数、维度均与 CS 算法保持一致; 鸟巢被发现概率取

值^[16] $P_{\alpha,\min}=0.005$ ， $P_{\alpha,\max}=1$ ；步长控制因子 $\alpha_{\min}=0.05$ ， $\alpha_{\max}=0.5$ 。

PSO 算法是智能优化算法中最具有代表性的算法，因此文章在实验对比中选用了此算法。其中，算法种群规模、最大迭代次数、维度与 CS 算法相同；学习因子的取值范围为 1.5~2；惯性权重 $\omega \in [0.4, 1.1]$ 。

DCS 算法，参数设置基本与 CS 保持一致，其中，步长调整因子 $\tau \in [0.8, 1.6]$ ；发现概率 $\text{new}P_{\alpha} \in [0.2, 0.5]$ 。

基于上述参数的设置，实验分别运用上述算法对 8 个不同难度的测试函数进行仿真实验，从而全面测试本文算法在寻优过程中的求解性能。下面详细分析算法的寻优精度与收敛速度。

3.2 寻优精度分析

CS、ICS、PSO、DCS 四种算法针对每个测试函数在相同环境下分别独立运行 30 次。其中，函数 1~5 为高维函数，维数 $D=10/30$ ；函数 6~8 为低维函数，维数 $D=2$ 。以下数据为函数 1~5 在不同维数下 4 种算法的求解精度，其中加粗了精度较好的解。

低维情况下，函数 6~8 在 4 种算法中均可探索出全局最优解，所以文章未用表格赘述。但依据表 1~5 的数据可以看出，本文提出的 DCS 算法在同一函数相同维数下的求解精度均高于其他 3 种算法。在不同维数下，对于前 5 个测试函数，CS、ICS、PSO 和 DCS 的寻优精度随着维数的增加而有所降低，尤其是 $f_3(x)$ 函数最为明显，此函数在 10 维下可以找到函数的全局最优解，在 30 维中寻优精度虽然有所降低但测试数据依然表明 DCS 算法的求解精度均高于 CS、ICS、PSO 算法。仿真实验结果表明，在不同维度的测试函数上，本文 DCS 算法的寻优精度比其他 3 种算法的寻优精度有一定程度地提高。

表 1 $f_1(x)$: Sphere 函数的仿真结果

Tab. 1 Simulation results of Sphere function

维数	算法	最优解	最差解	平均值
10	CS	2.210e-18	9.647e-17	2.318e-17
	ICS	1.288e-33	9.293e-32	1.880e-32
	PSO	4.855e-08	3.675e-04	4.867e-05
	DCS	2.417e-63	8.323e-59	8.446e-60
30	CS	1.922e-06	1.186e-05	6.548e-06
	ICS	3.860e-11	5.693e-10	1.409e-10
	PSO	6.620e-02	3.018e-01	1.676e-01
	DCS	1.657e-19	3.230e-17	8.505e-18

表 2 $f_2(x)$: Schwefel's problem 2.22 函数的仿真结果

Tab. 2 Simulation results of Schwefel's problem 2.22 function

维数	算法	最优解	最差解	平均值
10	CS	1.623e-08	2.984e-07	9.458e-08
	ICS	2.781e-18	3.392e-17	1.432e-17
	PSO	3.214e-03	7.217e-02	2.577e-02
	DCS	1.618e-43	1.393e-39	1.987e-40
30	CS	1.223e-02	4.510e-02	2.355e-02
	ICS	4.407e-05	9.045e-04	2.934e-04
	PSO	8.814e-02	8.520e-01	4.672e-01
	DCS	1.231e-13	8.008e-12	2.114e-12

表 3 $f_3(x)$: Griewank 函数的仿真结果

Tab. 3 Simulation results of Griewank function

维数	算法	最优解	最差解	平均值
10	CS	7.735e-10	1.812e-04	1.740e-05
	ICS	1.332e-15	7.485e-07	8.865e-08
	PSO	1.087e-06	2.868e-02	5.764e-03
	DCS	0	0	0
30	CS	9.898e-07	2.382e-04	3.866e-05
	ICS	3.196e-10	4.906e-06	6.601e-07
	PSO	7.818e-05	5.150e-01	4.028e-02
	DCS	0	3.331e-16	3.123e-17

表 4 $f_4(x)$: Schwefel's problem 2.21 函数的仿真结果

Tab. 4 Simulation results of Schwefel's problem 2.21 function

维数	算法	最优解	最差解	平均值
10	CS	3.351e-06	2.374e-05	9.500e-06
	ICS	1.334e-08	1.271e-07	3.957e-08
	PSO	2.800e-03	8.300e-02	3.135e-02
	DCS	2.513e-10	9.242e-08	1.904e-08
30	CS	1.491e-01	7.051e-01	3.750e-01
	ICS	5.409e-02	2.097e-01	9.493e-02
	PSO	7.220e-02	3.990e-01	1.982e-01
	DCS	1.468e-02	1.260e-01	5.324e-02

表 5 $f_3(x)$: Zakharov 函数的仿真结果
Tab. 5 Simulation results of Zakharov function

维数	算法	最优解	最差解	平均值
10	CS	7.604e-13	3.242e-11	9.582e-12
	ICS	5.577e-15	1.760e-13	3.752e-14
	PSO	1.021e-05	4.500e-03	8.147e-04
	DCS	1.806e-17	8.914e-14	1.095e-14
30	CS	6.258e-02	1.962e-01	1.172e-01
	ICS	3.526e-02	1.268e-01	7.340e-02
	PSO	2.614e-01	9.846e-01	6.949e-01
	DCS	3.268e-03	4.247e-02	1.719e-02

3.3 收敛曲线分析

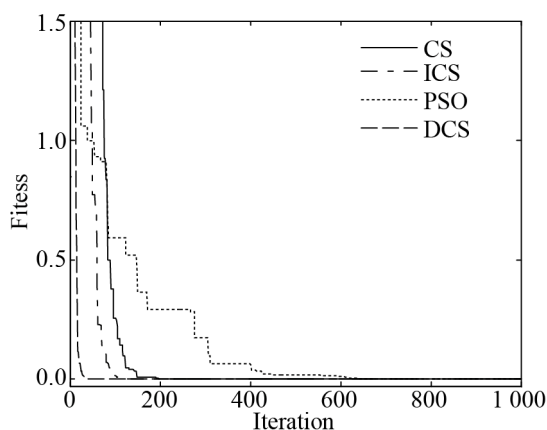
算法收敛速度的快慢、跳出局部极值的情况都可由收敛曲线直观显现。因此, 图 1~4 为 4 种算法在各个函数下的收敛曲线对比图。其中, 图 1~2 中的(a)、(b)与图 3 中的(a)依次为前 5 个测试函数在 10 维下的收敛曲线; 图 3 中的(b)与图 4 的(a)、(b)依次为函数 $f_6(x)$ 、 $f_7(x)$ 和 $f_8(x)$ 在维度 $D=2$ 的收敛曲线。

$f_1(x)$ 、 $f_2(x)$ 、 $f_4(x)$ 为单峰函数, 可以较好地测试算法的收敛性。图 1 中的(a)、(b)和图 2 中(b)的这 3 幅收敛曲线对比图显示, 本文算法在 100 代之内可以最终收敛, 而其他 3 种算法 200~800 代

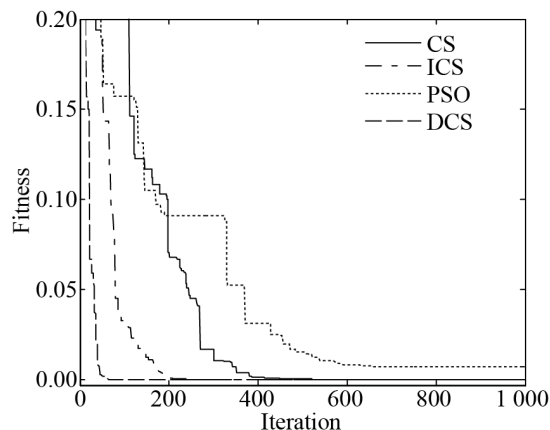
才能最终收敛。其中图 2 中的(b), 本文算法虽然在 200~400 代内完成最终收敛, 但它的收敛速度仍快于其他 3 种算法。 $f_3(x)$ 、 $f_5(x)$ 是复杂多峰函数, 局部解较多且容易陷入局部极值, 可以很好地测试算法的整体寻优性能。

从图 2(a)和图 3(a)的这 2 幅图可以看出, 4 种算法均有陷入局部解的情况, 其中 CS、PSO 算法尤为明显, ICS、DCS 算法的收敛曲线较为光滑, 陷入局部解的次数偏少, 但是 DCS 算法的收敛速度却明显快于其它算法。对于函数 $f_6(x)$ 、 $f_7(x)$ 和 $f_8(x)$, 4 种算法均能找到全局最优解, 但从图 3(b)和图 4(a)、(b)的这 3 幅图中可以看出本文算法在 50 代以内达到最终收敛, 收敛速度相对较快。同时, 观察 3 幅图像发现, PSO 算法陷入局部极值的次数较多, 其他 3 种算法则不明显, 尤其是 DCS 算法的收敛曲线几乎垂直向下。这是因为自适应步长和动态发现概率能够有效地平衡全局搜索与局部搜索, 使算法前期能够快速收敛到最优解区域, 后期可在最优解附近深入搜索。

综上所述, 改进后的算法在收敛速度、寻优精度方面均有较大提升。

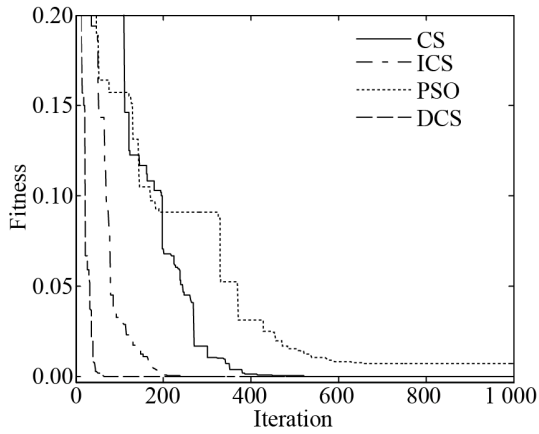


(a) Sphere 函数收敛曲线

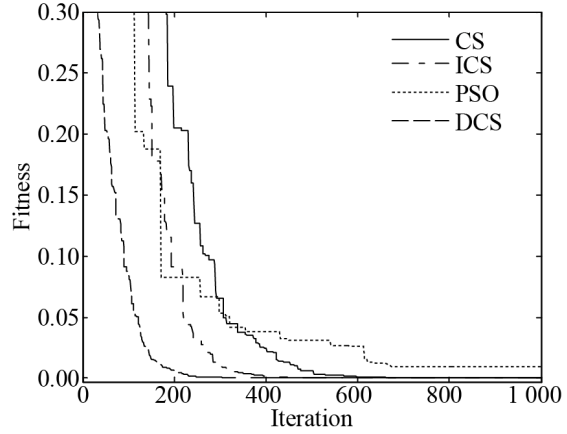


(b) Schwefel's problem 2.22 函数收敛曲线

图 1 $f_1(x)$ 、 $f_2(x)$ 函数的收敛曲线Fig. 1 Convergence curve of $f_1(x)$ 、 $f_2(x)$ function

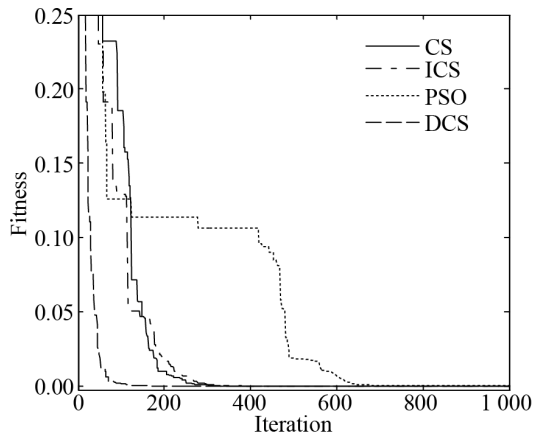


(a) Griewank 函数收敛曲线

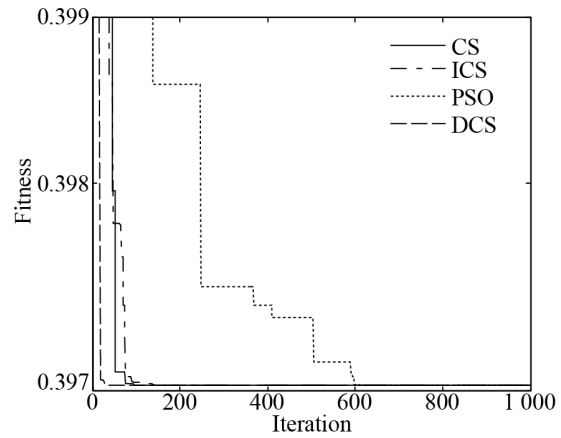


(b) Schwefel's problem 2.22 函数收敛曲线

图 2 $f_3(x)$ 、 $f_4(x)$ 函数收敛曲线
Fig. 2 Convergence curve of $f_3(x)$, $f_4(x)$ function

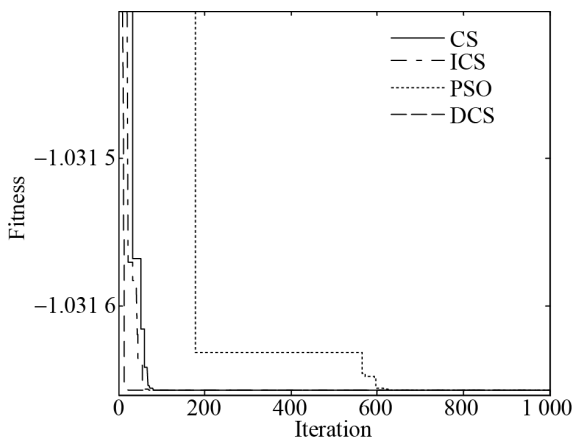


(a) Zakharow 函数收敛曲线

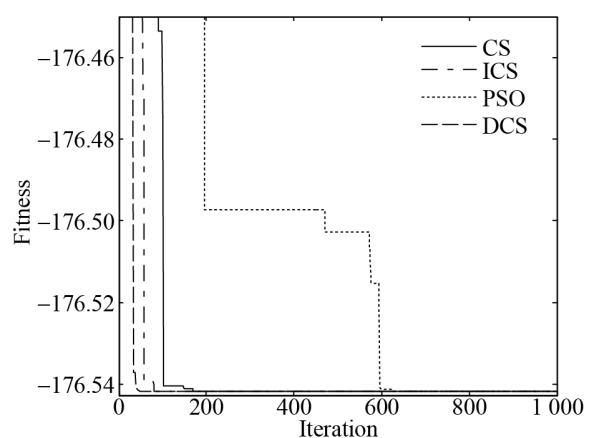


(b) Branin's rcos 函数收敛曲线

图 3 $f_5(x)$ 、 $f_6(x)$ 函数收敛曲线
Fig. 3 Convergence curve of $f_5(x)$, $f_6(x)$ function



(a) Six-Hump Camel-Back 函数收敛曲线



(b) Hansen 函数收敛曲线

图 4 $f_7(x)$ 、 $f_8(x)$ 函数收敛曲线
Fig. 4 Convergence curve of $f_7(x)$, $f_8(x)$ function

4 结论

本文通过分析基本布谷鸟搜索算法中存在的不足之处, 分别对莱维飞行机制、鸟巢被发现概率参数进行改进。文章在鸟巢位置更新公式中, 通过引入步长调整因子来动态约束每一代寻优个体的移动步长, 从而加速算法地收敛速度, 增强求解精度; 同时, 将具有均匀分布与 F 分布特性的动态惯性权重策略嵌入鸟巢被发现概率中, 不仅增强了种群的多样性与活跃性, 还良好地维持了算法全局探索能力、局部开采能力的战略平衡。实验结果表明: 改进后的算法收敛速度较快、求解精度较高, 并且算法陷入局部极值的次数明显减少, 改进后的算法在求解函数优化问题方面有一定的成效。随着布谷鸟算法日益广泛的应用, 将本文算法进一步完善, 更合理地应用到一些工程问题中, 从而获得更好的求解性能是下一步研究的方向和重点。

参考文献:

- [1] Goldberg D E. Genetic algorithm in search, optimization and machine learning[M]. Boston: Addison-Wesley Longman Publishing Co. Inc., 1989.
- [2] Storn R, Price K. Differential Evolution—a Simple and Efficient Heuristic for global Optimization over Continuous Spaces[J]. Journal of Global Optimization (S0925-5001), 1997, 11(4): 341-359.
- [3] Dorigo M, Maniezzo V, Colomi A. Ant system: optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics a Publication of the IEEE Systems Man & Cybernetics Society (S1083-4419), 1996, 26(1): 29-41.
- [4] Shi Y, Eberhart R C. A modified Particle swarm optimizer[C]// IEEE International Conference on Evolutionary Computation. Anchorage: IEEE, 1998: 69-73.
- [5] Kennedy J, Eberhart R. Particle swarm optimization[C]// Proceedings of IEEE International Conference on Neural Networks. Perth, Australia: IEEE ICNN '95', 1995: 1942-1948.
- [6] Pan W T. A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example[J]. Knowledge-Based Systems (S0950-7051), 2012, 26(2): 69-74.
- [7] Krishnanand K N, Ghose D. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics[C]// Proc of IEEE Swarm Intelligence Symposium. Pasadena: IEEE, 2005: 84-91.
- [8] Yang X S. Nature-Inspired Metaheuristic Algorithms[M]. Beckington, UK: Luniver Press, 2008.
- [9] Yang X S, Deb S. Cuckoo search via Levy flight[C]// Proceedings of World Congress on Nature & Biologically Inspired Computing. Coimbatore. India: IEEE, 2009: 210-214.
- [10] Yang X S, Deb S. Engineering optimization by cuckoo search[J]. Int'l Journal of Mathematical Modeling and Numerical Optimization (S2040-3607), 2010, 1(4): 330-343.
- [11] 李煜, 马良. 新型元启发式布谷鸟搜索算法[J]. 系统工程, 2012, 30(8): 64-69.
Li Yu, Ma Liang. A new metaheuristic cuckoo search algorithm [J]. Systems Engineering, 2012, 30(8): 64-69.
- [12] Suresh S, Lal S. An efficient cuckoo search algorithm based multilevel thresholding for segmentation of satellite images using different objective functions [J]. Expert Systems with Applications (S0957-4174), 2016, 58(C): 184-209.
- [13] Abd-Elazim S M, Ali E S. Optimal location of STATCOM in multimachine power system for increasing loadability by Cuckoo Search algorithm[J]. International Journal of Electrical Power & Energy Systems (S0142-0615), 2016, 80: 240-251.
- [14] Naik M K, Panda R. A novel adaptive cuckoo search algorithm for intrinsic discriminant analysis based face recognition[J]. Applied Soft Computing (S1568-4946), 2016, 38(C): 661-675.
- [15] Wang J, Zhou B. A hybrid adaptive cuckoo search optimization algorithm for the problem of chaotic systems parameter estimation[J]. Neural Computing & Applications (S0941-0643), 2016, 27(6): 1511-1517.
- [16] Valia E, Tavakoli S, Mohanna S. Improved cuckoo search for reliability optimization problems[J]. Computers & Industrial Engineering (S0360-8352), 2013, 64(1): 459-468.
- [17] 马卫, 孙正兴. 采用搜索趋化策略的布谷鸟全局优化算法[J]. 电子学报, 2015, 43(12): 2429-2439.
Ma Wei, Sun Zhengxing. A global cuckoo optimization algorithm using coarse-fine search[J]. Acta Electronica Sinica, 2015, 43(12): 2429-2439.

- [18] Layeb A. A novel quantum inspired cuckoo search for knapsack problems[J]. International Journal of Bio-Inspired Computation (S1758-0366), 2011, 3(5): 297-305.
- [19] 贾云璐, 刘胜, 宋颖慧. 基于种群特征反馈的布谷鸟搜索算法[J]. 控制与决策, 2016, 31(6): 969-975.
Jia Yunlu, Liu Sheng, Song Yinghui. Cuckoo search algorithm based on swarm feature feedback[J]. Control and Decision, 2016, 31(6): 969-975.
- [20] 贺兴时, 李娜, 杨新社, 等. 多目标布谷鸟搜索算法[J]. 系统仿真学报, 2015, 27(4): 731-737.
He Xingshi, Li Na, Yang Xinshe. Multi-objective cuckoo search algorithm[J]. Journal of System Simulation, 2015, 27(4): 731-737.
- [21] Zheng W, Fu H, Yang G. Targeted Mutation: A Novel Mutation Strategy for Differential Evolution[C]// IEEE, International Conference on TOOLS with Artificial Intelligence. Italy: IEEE, 2015: 286-293.
- [22] 李荣雨, 戴睿闻. 自适应步长布谷鸟搜索算法[J]. 计算机科学, 2017, 44(5): 235-240.
Li Rongyu, Dai Ruiwen. Adaptive step-size cuckoo search algorithm[J]. Computer Science, 2017, 44(5): 235-240.
- [23] 任子晖, 王坚, 高岳林. 马尔科夫链的粒子群优化算法全局收敛性分析[J]. 控制理论与应用, 2011, 28(4): 462-466.
Ren Zihui, Wang Jian, Gao Yuelin. The global convergence analysis of particle swarm optimization algorithm based on Markov chain[J]. Control Theory & Applications, 2011, 28(4): 462-466.
- [24] 王凡, 贺兴时, 王燕, 等. 基于 CS 算法的 Markov 模型及收敛性分析[J]. 计算机工程, 2012, 38(11): 180-182, 185.
Wang Fan, He Xingshi, Wang Yan, et al. Markov model and convergence analysis based on cuckoo search algorithm[J]. Computer Engineering, 2012, 38(11): 180-182, 185.
- [25] 李枝勇, 马良, 张惠珍. 蝙蝠算法收敛性分析[J]. 数学的实践与认识, 2013, 43(12): 182-190.
Li Zhiyong, Ma Liang, Zhang Huizhen. Convergence analysis of bat algorithm[J]. Mathematics in Practice and Theory, 2013, 43(12): 182-190.
- [26] 刘洪波, 王秀坤, 谭国真. 粒子群优化算法的收敛性分析及其混沌改进算法[J]. 控制与决策, 2006, 21(6): 636-640.
Liu Hongbo, Wang Xiukun, Tan Guozhen. Convergence analysis of particle swarm optimization and its improved algorithm based on chaos[J]. Control and Decision, 2006, 21(6): 636-640.