

自适应步长布谷鸟搜索算法

李荣雨 戴睿闻

(南京工业大学计算机科学与技术学院 南京 211800)

摘要 布谷鸟搜索算法(CSA)是一种新颖且简单、高效的生物启发式算法。针对标准算法存在后期收敛速度慢、易陷入局部最优等问题,提出了一种新的自适应步长布谷鸟搜索算法(ASCSA)。通过自适应调整莱维飞行步长使算法在前期拥有较大的寻优空间,提高全局搜索能力;步长随迭代自适应减小,算法的局部开发能力增强。针对偏好随机游动,引入动态惯性权重和记忆策略后,算法能够充分利用历史经验,稳定性得到提高。实验结果表明,改进后的布谷鸟搜索算法的各方面性能较标准算法及相关改进版本都有显著提高。

关键词 布谷鸟搜索算法, 莱维飞行, 自适应步长, 动态惯性权重, 记忆策略

中图分类号 TP301.6 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.05.042

Adaptive Step-size Cuckoo Search Algorithm

LI Rong-yu DAI Rui-wen

(College of Computer Science & Technology, Nanjing Tech University, Nanjing 211800, China)

Abstract Cuckoo search algorithm (CSA) is a novel nature-inspired algorithm which is simple and efficient. To overcome the defections that standard algorithm has slow convergence rate and falls into local optimum easily in the later period, a new adaptive step-size cuckoo search algorithm(ASCSA) was proposed. By adjusting the step-size of Lévy flight adaptively, the algorithm enhances the ability of global search in the earlier period and the local search in the later period. What's more, for the bias random walk, by introducing the dynamic inertial weight and memory strategy, the introduced algorithm can make full use of historical experience. The stability of algorithm has been strengthened. Simulation results show that the performance of ASCSA is obviously improved by compared with the standard CS algorithm and modified ones.

Keywords Cuckoo search algorithm, Lévy flight, Adaptive step-size, Dynamic inertia weight, Memory strategy

1 引言

优化问题在人类社会的发展中发挥着重要作用,现实生活中的实际问题可抽象为不同的数学模型。优化方法是在这些数学模型中求解能够取得最佳方案的可行解,例如在生产调度中寻求最佳方案使效率最大化;遍历城市(节点)寻求最佳路径使总里程最短等问题。传统的优化算法严重依赖于初始点的选择和函数的可微性,自身复杂度较大,随着工程规模和计算难度的不断增大,传统优化算法已难以胜任。

生物启发式算法因其简单、高效已成为目前解决实际问题的一个重要研究方向。生物启发式算法是一种受生物进化机制或社会行为机制启发而直观构造的计算策略。过去几十年,尤其是近20年,多种生物启发式算法如蚁群优化算法^[1](Ant Colony Optimization, ACO)、粒子群优化算法^[2](Particle Swarm Optimization, PSO)、人工蜂群算法^[3](Artificial Bee Colony Algorithm, ABC)、萤火虫算法^[4](Firefly Algorithm, FA)等算法相继诞生。生物启发式算法具有鲁棒性且

通用性较强^[5],为解决复杂优化问题提供了新的思路。生物启发式算法在诸多领域如工程优化^[6]、调度优化^[7]、电力系统优化^[8]已获得应用。布谷鸟搜索算法^[5](CSA)结构简单、全局搜索能力强,设定参数少于PSO和GA等智能算法,在求解上述实际问题时更为高效^[9]。

布谷鸟搜索算法(CSA)作为一种较新的生物启发式算法,由Yang和Deb于2009年提出。该算法模拟了布谷鸟独特的寻窝产卵的行为,并引入了自然界鸟类、果蝇飞行轨迹的莱维飞行机制^[5],能够快速有效地寻找到连续优化问题的最优解。目前,CS算法已经被应用于多种工程优化问题^[9-10],具有潜在的研究价值。虽然布谷鸟搜索算法控制参数少、简单、高效且易于实现,但是该算法同其他生物启发算法一样,也存在后期收敛速度慢、寻优精度低、易陷入局部最优等缺点,故目前改进算法参数、调整算法结构来提高算法性能是一个重要的研究领域。围绕这一领域,国内外众多学者对CSA展开研究,并提出一些改进方法。Walton等人^[11]针对莱维飞行中的随机步长提出一种改进版本以加强局部搜索。Wang

到稿日期:2016-04-30 返修日期:2016-08-12 本文受江苏省高校自然科学基金资助项目(12KJB510007)资助。

李荣雨(1977—),男,博士,副教授,主要研究领域为工业系统的监控与优化控制;戴睿闻(1991—),男,硕士生,主要研究领域为智能优化算法及其应用。

等人^[12]在莱维飞行的更新公式中加入服从 $[0,1]$ 均匀分布的压缩因子,算法效果优于标准CS算法。Tuba等人^[13]提出在偏好随机游动环节加入种群排序的改进版本,改进算法的性能优于标准的CS算法。Jin等人^[14]对发现概率和莱维飞行步长进行动态自适应改进,提高了算法性能。Valian等人^[15]提出步长因子自适应改变的方法,提高了算法的搜索精度。针对步长的改进方法中,文献^[14-15]仅就莱维飞行进行了自适应改进,在处理单峰函数时性能较好,但在处理复杂多峰函数时时间开销大,效率较低。布谷鸟搜索算法的主体由莱维飞行和偏好随机游动两个部分组成,两者分别控制算法的全局搜索和局部搜索。本文分析了CS算法的寻优过程,提出了自适应步长布谷鸟搜索算法(ASCSA)。通过在寻优过程中自适应地改变搜索步长,在偏好随机游动环节中引入动态惯性权重及记忆策略,实现对算法的整体动态调整。标准测试函数的实验结果也验证了所提算法能够较好地解决布谷鸟搜索算法后期收敛速度慢、易陷入局部最优的问题。

2 布谷鸟搜索算法

布谷鸟繁衍后代的方式很奇特且具有攻击性。某些种类的布谷鸟会在宿主的巢穴悄悄产卵并移除巢中的其他鸟蛋,以提高自己后代的生存几率。由于布谷鸟幼雏的孵化时间比宿主的短,先孵化出的布谷鸟幼雏会本能地将巢穴中其他的卵破坏,使自身更可能获得宿主的抚养。在漫长的进化中,为确保自身种群繁衍,宿主鸟不断提高识别外来鸟蛋的能力。当宿主鸟发现其鸟巢中的陌生卵时,它将抛出这枚卵,或者丢弃该鸟巢,在其他地方重新筑窝。为了简化和模拟布谷鸟的繁殖方式,Yang和Deb假定3条理想规则^[16]:

规则1 每只布谷鸟一次只产一个卵,并把该卵放在随机选择的鸟巢中,该鸟窝也被称为寄身巢。

规则2 具有最高质量的卵的寄生巢将被保留到下一代。

规则3 可用鸟巢的数量是固定的,而且鸟巢中寄身卵被发现的概率是 p_a 。当寄身卵被宿主鸟发现后,宿主鸟要么抛出该鸟蛋,要么放弃鸟巢去新位置重新构建新的鸟巢。

布谷鸟搜索算法中,布谷鸟选择产卵的寄生鸟窝表示一个可行解。算法实质是用新的更好的解代替上一代较差的解。基于上述布谷鸟寻窝育雏的习性及规则,下一代鸟窝位置的更新公式可表示为:

$$X_i^{t+1} = X_i^t + \alpha \oplus \text{Lévy}(\beta) \quad (1)$$

其中, X_i^t 表示第 t 代第 i 个解; α 是步长因子,用于控制随机搜索的范围; \oplus 为点对点乘法。该式描述的莱维飞行是一个马尔可夫链^[17],即下一代的位置仅取决于当前位置。 $\alpha \oplus \text{Lévy}(\beta)$ 是莱维飞行步长,表示从第 i 代的所有鸟窝(可行解)分别以莱维随机分布的方式搜寻到第 $i+1$ 代鸟窝(可行解)飞行的距离。

Yang等人通过对莱维分布函数的研究,得到幂次形式的概率密度函数:

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

由式(2)可知,布谷鸟连续的位置变化形成了一个带有重尾^[18]的概率分布,使布谷鸟算法从本质上体现了布谷鸟飞行

的轨迹,即寻优路径由频繁的短跳跃与偶然出现的长跳跃组成,这种寻优方式可以使CS算法拥有更大的搜索空间,更容易跳出局部最优。

文献^[9]采用式(3)计算莱维随机数:

$$\text{Lévy}(\beta) \sim \frac{Q * \mu}{|v|^{1/\beta}} \quad (3)$$

其中, μ 和 v 服从标准正态分布^[19], $\beta=1.5$ 。

$$Q = \frac{(\Gamma(1+\beta) * \sin(\pi * \beta / 2))^{1/\beta}}{\Gamma((\frac{1+\beta}{2}) * \beta * 2^{(\beta-1)/2})} \quad (4)$$

为了便于对莱维飞行进行计算,采用文献^[9]的步长因子:

$$\alpha = \alpha_0 * (X_i^t - X_b) \quad (5)$$

其中, $\alpha_0=0.01$, X_b 表示当前最优解。

结合式(1)~式(5),可用式(6)更新通过莱维飞行生成的解。

$$X_i^{t+1} = X_i^t + 0.01 * \frac{Q * \mu}{|v|^{1/\beta}} (X_i^t - X_b) \quad (6)$$

发现概率 p_a 表示宿主鸟识别巢中外来鸟蛋的概率。在算法过程中,莱维飞行更新鸟窝位置后,产生一个随机数与发现概率 p_a 进行比较。当随机数大于 p_a 时,即表示布谷鸟鸟蛋被发现,算法进入偏好随机游动环节,此时将淘汰该鸟蛋(可行解),并重新在附近生成一个新解:

$$X_i^{t+1} = X_i^t + v * (X_j^t - X_k^t) \quad (7)$$

其中, X_j^t 和 X_k^t 表示第 t 代的两个随机解; v 是服从 $[0,1]$ 上均匀分布的压缩因子,参与生成新的解来替换淘汰的解。

3 自适应步长布谷鸟搜索算法

布谷鸟搜索算法的高效性源于特殊的莱维飞行机制。在寻优过程中,若步长因子 α 一直较大,则算法探索能力强,但是无法获得高精度全局最优解;若步长因子 α 一直较小,则算法在达到目标精度时会付出更多的迭代次数^[12]。标准布谷鸟算法步长因子固定,无法做到寻优过程中步长的自适应调整,算法收敛速度慢。

对于偏好随机游动,由式(6)可知,莱维飞行机制已将搜索精度缩小一个或者多个数量级,若在随机偏好游动中再次缩小数量级则会导致数量级跳跃过快,错过全局最优解。此外,生物启发式算法源于对自然界生物行为习性的刻画,具有较大的随机性。

3.1 自适应莱维飞行

针对标准布谷鸟搜索算法收敛速度慢、寻优精度低的缺点,本文通过自适应莱维飞行机制进行全局搜索:莱维飞行步长随迭代的进行不断减小。改进的算法在寻优初期拥有较大的步长因子,从而扩大算法前期的搜索空间,提高全局搜索能力;在寻优过程中,步长减小,提高算法局部搜索性能。

基于以上分析,将式(5)中的 α_0 由标准算法的0.01改为式(8):

$$\alpha_0 = 0.001 * t_{\max} * \exp(- (t_i / t_{\max})) \quad (8)$$

改进的莱维飞行更新位置公式为:

$$X_{g+1,i} = X_{g,i} + 0.001 * t_{\max} * \exp(- (\frac{t_i}{t_{\max}})) * \frac{Q * \mu}{|v|^{1/\beta}} * (X_{g,i} - X_{best}) \quad (9)$$

其中, t_i 表示当前迭代次数, t_{\max} 表示迭代总次数。

基于以下两点原因采用式(9)的自适应莱维飞行更新鸟窝位置:

1) 迭代前期较大的步长有利于提高全局搜索能力,避免早熟收敛;迭代后期减小步长,从而提高搜索精度。

2) 步长因子按照非线性方式减小,前期衰减速度较快,后期衰减放缓,从而保证算法前期收敛速度快,算法后期以较小及变化幅度很小的步长搜索,以提高搜索精度。

3.2 动态惯性权重的偏好随机游动

布谷鸟搜索算法通过莱维飞行使可行解接近全局最优位置。为避免陷入局部最优,算法采用偏好随机游动淘汰部分可行解,并在其附近产生相同数量的新解。就解的精度数量级而言,当式(7)中 v 的取值较小时,被淘汰解 (X_i^t) 与其偏移量 ($v * (X_j^t - X_k^t)$) 将相差多个数量级,局部搜索效果不显著。就算法的执行过程而言,布谷鸟算法对被淘汰鸟窝的更新公式可表示为:

$$X_i^{t+1} = X_i^t + 0.01 * \frac{Q * \mu}{|v|^{1/\beta}} (X_i^t - X_b) + v * (X_j^t - X_k^t) \quad (10)$$

式(10)包括全局搜索(莱维飞行)和局部搜索(偏好随机游动)。

动态惯性改进策略^[20]是一种控制种群探索能力和开发能力的机制,能够有效提高算法的搜索能力,平衡局部搜索和全局搜索之间的关系。本文在偏好随机游动环节引入动态惯性权重,以使其具备自适应性。动态惯性权重 w 为:

$$w = 1 - e^{(-t_{max} * 2/t_i)} \quad (11)$$

其中, t_i 表示当前迭代次数, t_{max} 表示迭代的总次数。

这种改进通过迭代前期设定较大的惯性权重扩大寻优空间,增强算法的全局探索能力;同时为确保后期收敛到全局最优值,算法的局部搜索能力应不断增强,此时惯性权重适当降低。改进的偏好随机游动可表示为:

$$X_i^{t+1} = X_i^t + w * (X_j^t - X_k^t) \quad (12)$$

此外,局部搜索的更新方案使用了双随机解,无法保留上一代的有利信息,不利于精确搜索。本文将记忆策略引入偏好随机游动,即用上一代的鸟窝位置替换一个随机鸟窝,达到充分使用历史经验的目的。故本文将随机偏好游动的更新公式修改为:

$$X_i^{t+1} = X_i^t + w * (X_j^t - X_k^t) \quad (13)$$

3.3 改进算法的流程

自适应步长布谷鸟搜索算法(ASCSA)着眼于对算法的整体优化,兼顾了对莱维飞行和偏好随机游动的改进,使得算法在寻优过程中始终能够进行自适应调整,提高了收敛速度和精度。同时,在局部搜索环节引入记忆策略后,算法能够充分利用当前的有利信息,使后期收敛精度得到提高,在一定程度上克服了生物启发式算法普遍存在的随机性强的缺点。改进的布谷鸟搜索算法的步骤如下。

Step1 设置算法参数:解的维数 n , 鸟窝个数 m , 发现概率 p_a , 最大迭代次数 N , 搜索上下界。初始化鸟巢(解), 计算所有鸟巢(解)的适应度。

Step2 对每个鸟巢采用改进的莱维飞行(式(9))产生新解。

Step3 计算更新解的适应度,若新的解适应度更高,则替换旧的解。

Step4 按发现概率 p_a 随机淘汰部分解,并采用改进的偏好随机游动(式(13))产生与淘汰解数量相同的新解。

Step5 计算各鸟巢的适应度,输出当前最优解。

Step6 判断算法的终止条件,若满足则获得结果;否则重复步骤 Step2—Step6。

执行算法时,鸟窝数一般设置为 15~50,发现概率 $p_a = 0.25, \alpha = 1$ 。

4 仿真与分析

为证明 ASCSA 具有更快的收敛速度和更高的计算精度,本文选取 6 个标准测试函数进行测试,同时将文献[14-15]两改进步长的布谷鸟算法加入对比,以验证所提算法的性能优势。

4.1 测试函数

本文选取的测试函数涵盖单峰函数和复杂多峰函数,包括: Sphere 函数、Schaffer 函数、Rastrigin 函数、Griewank 函数等。各测试函数的参数设定如表 1 所列,算法参数设置为:鸟窝规模 $n = 30, p_a = 0.25$ ^[16]。

表 1 测试函数及参数设定

| | 测试函数 | 搜索空间 | 理论最优 | 维度 | 迭代次数 |
|----------------|---|-------------------|------|-------|------|
| Sphere | $f_1(x) = \sum_{i=1}^N x_i^2$ | $[-100, 100]$ | 0 | 10/30 | 1000 |
| Schwefel P2.22 | $f_2(x) = \sum_{i=1}^N x_i + \prod_{i=1}^N x_i $ | $[-50, 50]$ | 0 | 10/30 | 1000 |
| Schaffer | $f_3(x) = \frac{\sin^2 \sqrt{\frac{1}{4} \sum_{i=1}^d x_i^2} - 0.5}{[1 + 0.001(\sum_{i=1}^d x_i^2)]^2} + 0.5$ | $[-10, 10]$ | 0 | 10/30 | 1000 |
| Rosenbrock | $f_4(x) = \sum_{i=1}^N [100X_{i+1} - x_i^2]^2 + (x_i - 1)^2$ | $[-2.048, 2.048]$ | 0 | 10/30 | 1000 |
| Rastrigin | $f_5(x) = \sum_{i=1}^N [x_i^2 - 10 \cos(2\pi x_i) + 10]$ | $[-5.12, 5.12]$ | 0 | 10/30 | 1000 |
| Griewank | $f_6(x) = \frac{1}{4000} \sum_{i=1}^N x_i^2 - \prod_{i=1}^N \cos(\frac{x_i}{\sqrt{i}}) + 1$ | $[-600, 600]$ | 0 | 10/30 | 1000 |

4.2 实验评价标准

本文将 ASCSA 与两种改进步长的布谷鸟搜索算法即 NICS^[14]算法、ICS^[15]算法在不同维度的情况下进行对比实

验,以证明所提算法具备的竞争力与优势。NICS 与 ICS 算法皆是针对莱维飞行步长进行的自适应改进算法,具有代表性。实验中各算法独立运行 30 次,设置终止条件为迭代次数达到

1000次。为评价算法的优化效果,本文给出如下几个判定标准:1)全局最优解(GlobalBest),算法运行30次得到的全局最优解;2)平均最优值(MeanBest),算法运行30次后得到的最优值的期望,用来衡量算法寻优的平均质量;3)标准差(Std),算法运行30次得到的最优值与平均最优值之间的标准差,评价算法寻优的稳定性。

4.3 实验结果与分析

10维情况下的实验结果如表2及图1—图6所示。为清晰体现 ASCSA 在收敛速度和寻优精度上的优势,仿真图像中横坐标为迭代次数,纵坐标为目标函数适应度的数量级。从表2及仿真图像可知,3种改进算法较标准算法在性能方面都有提升。本文提出的 ASCSA 较其他两种改进算法即 ICS 和 NICS 在寻优精度和收敛速度上都有明显提高。

表2 10维情况下测试指标的对比

| 函数名 | 算法 | GlobalBest | MeanBest | Std |
|----------------|-------|------------|----------|----------|
| Sphere | CSA | 3.08e-15 | 4.19e-14 | 5.78e-14 |
| | ICS | 3.26e-33 | 5.833-31 | 1.06e-30 |
| | NICS | 4.07e-28 | 1.21e-25 | 1.43e-25 |
| | ASCSA | 1.28e-89 | 2.97e-87 | 7.41e-87 |
| Schwefel P2.22 | CSA | 3.24e-8 | 6.02e-7 | 6.28e-7 |
| | ICS | 5.82e-13 | 7.23e-12 | 2.81e-12 |
| | NICS | 4.85e-21 | 2.04e-20 | 1.63e-20 |
| | ASCSA | 2.03e-60 | 8.64e-58 | 8.83e-59 |
| Schaffer | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 |
| Rosenbrock | CSA | 1.98e-1 | 4.45e-1 | 6.84e-2 |
| | ICS | 2.72e-1 | 5.08e-1 | 1.89e-2 |
| | NICS | 1.90e-1 | 2.76e-1 | 3.61e-2 |
| | ASCSA | 2.15e-3 | 4.87e-2 | 4.01e-3 |
| Rastrigin | CSA | 7.32e+0 | 1.75e+1 | 3.01e+0 |
| | ICS | 1.93e-2 | 5.80e-2 | 1.76e-1 |
| | NICS | 1.13e+0 | 6.61e+0 | 6.76e-1 |
| | ASCSA | 4.09-9 | 2.54e-3 | 2.88e-4 |
| Griewank | CSA | 1.09e-2 | 3.88e-1 | 3.29e-1 |
| | ICS | 5.33e-4 | 1.54e-2 | 2.24e-3 |
| | NICS | 3.51e-2 | 2.05e-1 | 3.07e-2 |
| | ASCSA | 0 | 2.57e-3 | 5.86e-4 |

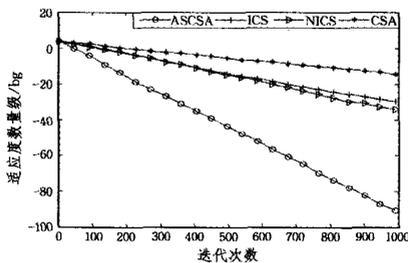


图1 Sphere收敛曲线

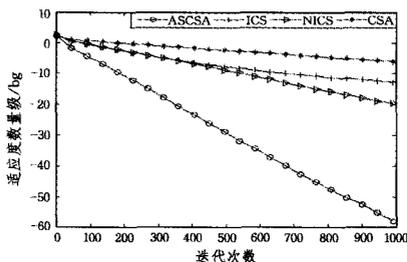


图2 Schwefel P2.22收敛曲线

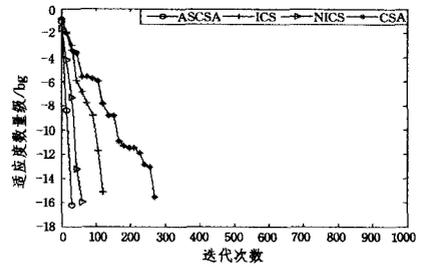


图3 Schaffer收敛曲线

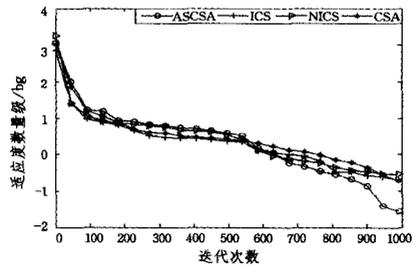


图4 Rosenbrock收敛曲线

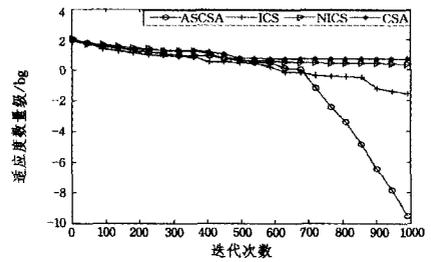


图5 Rastrigin收敛曲线

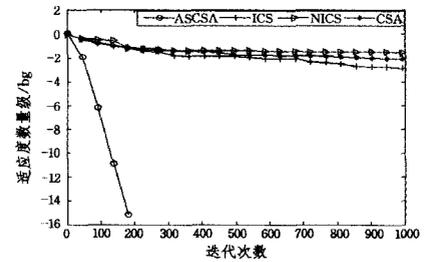


图6 Griewank收敛曲线

Sphere及Schwefel P2.22为简单的单峰函数,全局最优解易于求解,常用来衡量算法的寻优速度。由实验结果可知:1000次迭代后,ASCSA的最优解较两种改进算法ICS及NICS分别提高了56和61、47和39个数量级。就稳定性而言,本文算法的稳定性也高于其他算法。显然本文提出的改进算法的性能更强。

Schaffer函数为复杂单峰函数,在这个函数测试中,4种算法皆避免了早熟收敛,达到理论最优位置,但由图3可知,ASCSA的收敛速度最快,若运用在实时系统中以速度作为性能指标,本文算法将更具优势。Rosenbrock也是复杂单峰函数,该函数的全局最优值位于一个平滑、狭长的抛物线形山谷内,且提供的信息有限,优化算法难以找到搜索方向。表2及图4亦反映了该测试函数的特点,3种改进算法由于并未从搜索方向这个角度进行优化,改进效果不显著,但引入记忆性的ASCSA能够利用历史经验,迭代过程更加平滑,全局最优

解及平均最优解表现最佳。

图 5、图 6 反映的是多峰函数 Rastrigin 及 Griewank 函数的寻优情况。这两个函数都是典型的多模态非线性全局优化函数,峰形呈现高低起伏的不定跳跃性,具有震荡性,搜索空间存在很多局部最小值,适合衡量算法的全局搜索能力。由表 2 及图 5、图 6 可知,在 Rastrigin 及 Griewank 函数中,ICS 和 NICS 算法均陷入局部最优;本文所提的 ASCSA 因其在寻优过程中整体采用自适应步长进行搜索,且在偏好随机游动中引入了记忆策略,保留了上一代的有利信息,提高了局部搜索的精度,在多峰函数测试中均成功跳出了局部最优,性能最强。

对于处理不同维度的连续函数优化问题,本文提出的 ASCSA 同样具备优势。仿真实验结果如表 3 及图 7—图 12 所示。

表 3 30 维情况下测试指标的对比

| 函数名 | 算法 | GlobalBst | MeanBst | Std |
|-----------------|-------|-----------|----------|----------|
| Sphere | CSA | 2.25e-3 | 4.60e-2 | 7.42e-3 |
| | ICS | 2.94e-8 | 6.65e-7 | 3.09e-7 |
| | NICS | 4.11e-8 | 5.89e-7 | 2.02e-7 |
| | ASCSA | 3.54e-18 | 4.43e-16 | 2.89e-16 |
| Schwefel P2. 22 | CSA | 1.22e-2 | 3.92e-2 | 5.04e-2 |
| | ICS | 2.86e-5 | 1.85e-4 | 2.77e-4 |
| | NICS | 3.53e-4 | 8.08e-4 | 5.74e-4 |
| | ASCSA | 5.49e-14 | 2.33e-13 | 2.89e-13 |
| Schaffer | CSA | 0 | 0 | 0 |
| | ICS | 0 | 0 | 0 |
| | NICS | 0 | 0 | 0 |
| | ASCSA | 0 | 0 | 0 |
| Rosenbrock | CSA | 2.61e+1 | 3.97e+1 | 3.20e+1 |
| | ICS | 2.77e+1 | 4.83e+1 | 4.34e+1 |
| | NICS | 2.49e+1 | 3.52e+1 | 3.28e+1 |
| | ASCSA | 2.51e+1 | 3.14e+1 | 3.43e+1 |
| Rastrigin | CSA | 8.02e+1 | 1.16e+2 | 4.55e+1 |
| | ICS | 1.08e+2 | 1.51e+2 | 1.32e+2 |
| | NICS | 5.22e+1 | 9.26e+1 | 5.94e+1 |
| | ASCSA | 1.41e+1 | 2.94e+1 | 2.21e+1 |
| Griewank | CSA | 1.09e-2 | 6.85e-2 | 4.49e-2 |
| | ICS | 1.02e-4 | 1.05e-3 | 1.28e-3 |
| | NICS | 4.36e-6 | 1.55e-4 | 3.02e-4 |
| | ASCSA | 0 | 2.91e-4 | 4.39e-5 |

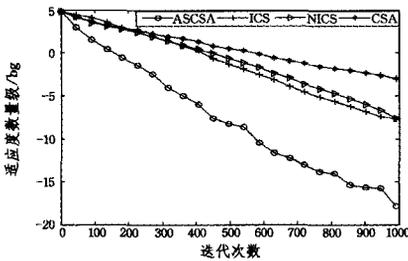


图 7 Sphere 收敛曲线

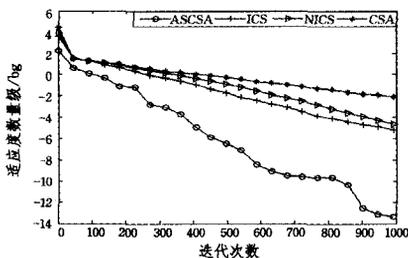


图 8 Schwefel P2.22 收敛曲线

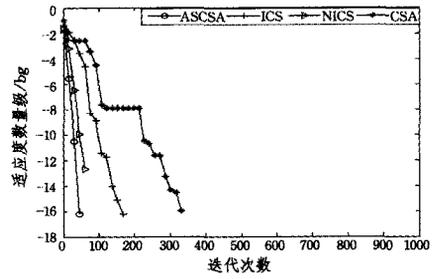


图 9 Schaffer 收敛曲线

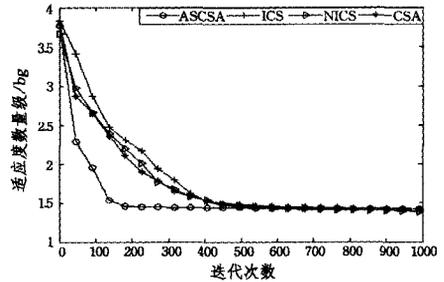


图 10 Rosenbrock 收敛曲线

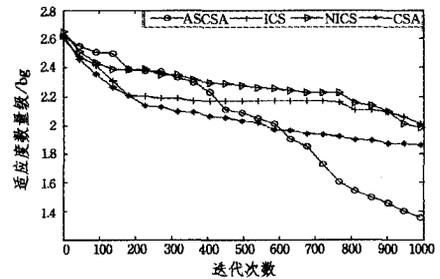


图 11 Rastrigin 收敛曲线

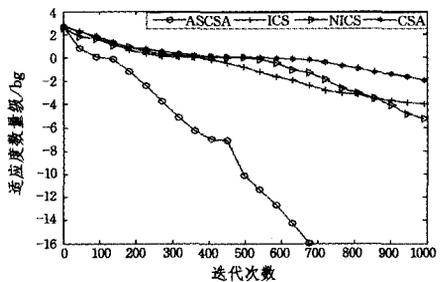


图 12 Griewank 收敛曲线

表 3 列出了在 30 维情况下,4 种算法寻优指标的统计。对于简单单峰函数 Sphere 和 Schwefel P2. 22, ASCSA 较其他两个自适应改进算法 ICS 和 NICS 分别高出了 10 个和 9 个数量级,9 个和 10 个数量级,寻优精度的优势较为明显。4 个算法在处理 Schaffer 函数时都能跳出局部最优,但本文所提算法所需的迭代次数最少。在 Rosenbrock 函数测试中,由于计算复杂度随问题维度增加而增大,4 种算法无法找到有效的搜索方向,皆陷入局部最优位置,改进效果不明显,但本文所提算法的前期收敛速度好于前三者。

在多峰复杂函数测试中,ASCSA 表现出更好的性能。对于 Rastrigin 函数,函数本身具有强烈震荡性,不易求得全局最优解。由实验结果可见,ICS 和 NICS 算法的寻优曲线均差于标准 CS 算法,ASCSA 的寻优精度虽然不高,但仍高于其

他3种算法,效果最好。Griewank函数的特点是维数越高,其性质越趋于单峰。实验结果表明:ICS和NICS算法较标准CS算法在收敛精度上的提高有限,陷入局部最优。虽然在平均适应度方面,本文提出的算法与NICS接近,但ASCSA在30次迭代中有一半的成功率跳出局部最优并收敛到理论最优位置。

不同维度下的函数测试结果表明:ASCSA在大多数函数优化问题上能展现出良好的性能优势,且能够较好地解决标准CS算法存在的后期收敛速度慢、易陷入局部最优等问题。

结束语 布谷鸟搜索算法(CSA)是一种较新的生物启发式算法。该算法简单、高效,能够解决连续函数优化问题,并被成功地应用于解决大量实际问题。由于标准布谷鸟搜索算法在莱维飞行和偏好随机游动两方面不具备自适应性且随机性较大,算法存在后期收敛速度慢、精度不高、易陷入局部最优等缺陷。本文着眼于对算法的整体改进,通过对布谷鸟搜索算法寻优过程和特性的分析,提出了自适应步长布谷鸟搜索算法(ASCSA)。算法兼顾了全局搜索和局部搜索的自适应改进,在偏好随机游动环节引入记忆策略,加快了算法的搜索速度,提高了计算精度,不易陷入局部最优。仿真实验进一步表明:新的自适应步长布谷鸟搜索算法收敛速度快、求解精度高,改善了标准算法在后期寻优精度低、收敛速度慢、易陷入局部最优等问题,在解决连续函数优化问题上具备优势。

参考文献

- [1] DORIGO M, MANIEZZO V, COLORNI A. The ant system: Optimization by a colony of cooperating agents [J]. IEEE Trans. on Systems, Man, and Cybernetics, Part B, 1996, 26(1): 29-41.
- [2] KENNEDY J, EBERHART R C. Particle swarm optimization [C]// IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE Press, 1995, 1942-1948.
- [3] BAHRIYE A, KARABOGA D A. Modified artificial bee colony algorithm for real-parameter optimization[J]. Information Sciences(S0020-0255), 2012, 192(1): 120-142.
- [4] YANG X S. Nature-inspired metaheuristic algorithms[M]. Frome, UK: Luniver Press, 2008: 81-96.
- [5] YANG X S, DEB S. Cuckoo search via Levy flight [C]// Proceedings of World Congress on Nature & Biologically Inspired Computing. India, Washington: IEEE Publications, 2009: 210-214.
- [6] AKAY, KARABOGA. Artificial bee colony algorithm for large-scale problems and engineering optimization[J]. Journal of Intelligent Manufacturing, 2010, 23(4): 1001-1014.
- [7] LIU Z G, LI Y, et al. Multi-resource Constrained Job-shop Optimization Scheduling Based on Ant Colony Algorithm[J]. Journal of System Simulation, 2007, 19(1): 216-220. (in Chinese)
刘志刚, 李言, 等. 基于蚁群算法的Job-Shop的多资源约束车间作业调度[J]. 系统仿真学报, 2007, 19(1): 216-220.
- [8] DEL VALLE Y, VENAYAGAMOORTHY G, MOHAGHEGHI S, et al. Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems[J]. IEEE Transactions on Evolutionary Computation, 2008, 12(2): 171-195.
- [9] YANG X S, DEB S. Engineering optimization by cuckoo search [J]. Int'l Journal of Mathematical Modeling and Numerical Optimization, 2010, 1(4): 330-343.
- [10] GADOMI A, YANG X S, ALAVI A. Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems [J]. Engineering with Computers, 2013, 29(29): 17-25.
- [11] 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.
- [12] WANG L J, YIN Y L, ZHONG Y W. Cuckoo search with varied scaling factor[J]. Frontiers of Computer Science, 2015, 9(4): 623-635.
- [13] TUBA M, SUBOTIC M, STANAREVIC N. Modified cuckoo search algorithm for unconstrained optimization problems[C]// Leandre R, Demiralp M, Tuba M, et al., eds. Proc. of the European Computing Conf. (ECC 2011). Athens: WSEAS Press, 2011: 263-268.
- [14] JIN Q B, QI L F. Novel improved cuckoo search for PID controller design[J]. Transactions of the Institute of Measurement & Control, 2014, 37(6): 1-11.
- [15] VALIA E, TAVAKOLI S, MOHANNA S. Improved cuckoo search for reliability optimization problems [J]. Computers & Industrial Engineering, 2013, 64(1): 459-468.
- [16] YANG X S, DEB S. Cuckoo search: recent advances and applications[J]. Neural Computing and Application, 2014, 24(1): 169-174.
- [17] HE X S, LI N, YANG X S, et al. Multi-objective Cuckoo Search Algorithm[J]. Journal of System Simulation, 2015, 27(4): 731-737.
- [17] HE X S, LI N, YANG X S, et al. Multi-objective Cuckoo Search Algorithm[J]. Journal of System Simulation, 2015, 27(4): 731-737. (in Chinese)
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.
- [18] PAVLYUKOVICH I. Lévy flights, non-local search and simulated annealing[J]. Journal of Computational Physics, 2007, 226(2): 1830-1844.
- [19] ZHENG H Q, ZHOU Y Q. A Novel Cuckoo Search Optimization Algorithm Base on Gauss Distribution [J]. Journal of Computational Information Systems, 2012, 8(10): 4193-4200.
- [20] SHI, EBERHART R. A modified particle swarm optimizer [C]// The 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence, Anchorage, USA, 1998: 69-73.