

一种布谷鸟-交叉熵混合优化算法及其性能仿真

李国成^{1,2}, 肖庆宪¹

(1. 上海理工大学 管理学院, 上海 200093; 2. 皖西学院 金融与数学学院, 六安 237012)

摘要: 为了提高布谷鸟搜索算法在求解复杂优化问题时的收敛速度和搜索精度, 基于交叉熵方法, 构建了一种新的布谷鸟-交叉熵混合优化算法. 该算法将基于模型的交叉熵随机优化算法和基于种群的布谷鸟搜索进行有机融合, 采用协同演化策略, 既提升了混合算法收敛速度, 又改善了其全局优化能力. 对经典测试函数和 PID 控制器整定问题的仿真结果表明, 新算法具有全局搜索能力强、求解精度高和鲁棒性好等特性, 是一种求解复杂优化问题的可行和有效算法.

关键词: 布谷鸟搜索; 交叉熵; 混合优化; 高维函数; 控制器整定

中图分类号: O 229

文献标志码: A

Hybrid Optimization Algorithm Based on Cuckoo Search and Cross Entropy and Its Performance

LI Guocheng^{1,2}, XIAO Qingxian¹

(1. Business School, University of Shanghai for Science and Technology, Shanghai 200093, China;

2. School of Finance & Mathematics, West Anhui University, Lu'an 237012, China)

Abstract: In order to improve the rate of convergence and obtain high optimization precision of cuckoo search, a hybrid optimization algorithm for solving complicated optimization problems was proposed. The proposed algorithm combines model-based cross-entropy method with population-based cuckoo search. The hybrid algorithm not only improves the rate of convergence but also enhances the global search ability by adopting the co-evolution strategy. Simulated experiments were conducted on classical benchmarks and PID controller tuning problem. The results show that the proposed algorithm possesses more powerful global search capacity, higher optimization precision and robustness, and is feasible and effective for solving complicated optimization problems.

Key words: cuckoo search; cross entropy; hybrid optimization; high-dimensional function; controller tuning

收稿日期: 2013-10-04

基金项目: 国家自然科学基金资助项目(11171221); 上海市一流学科建设资助项目(XTKX2012)

第一作者: 李国成(1976-), 男, 博士研究生. 研究方向: 计算智能与金融优化. E-mail: ligc313@163.com

通信作者: 肖庆宪(1956-), 男, 教授. 研究方向: 金融工程. E-mail: qxxiao@163.com

随着科学技术的快速发展,人们在现实世界中面临着更加复杂多变的系统,如电力系统、蛋白质结构、医学图像匹配系统以及金融市场等^[1].这些复杂系统的优化问题常常在高维空间进行,而高维数值优化问题都较为复杂,采用一般的智能算法很难获得全局最优解^[2].为此,许多改进算法纷纷被提出,如动态多群体粒子群优化^[3]、自适应协同演化微分进化算法^[4]、遗传和细菌觅食混合算法^[5].

最近,Yang等^[6]基于布谷鸟孵育寄生的繁殖行为和Lévy飞行特性提出一种新的启发式搜索算法——布谷鸟搜索(cuckoo search,CS).该算法因仿生能力强、控制参数少和极易实现而广泛应用于工程设计、结构参数选择和生产调度等方面^[6-11].与此同时,学者们也从不同角度对CS算法进行了改进,如文献^[12]提出自适应Lévy飞行步长和个体间信息交换以提高CS的收敛速度,文献^[13]采用动态飞行步长和发现概率,文献^[14]提出了二进制CS算法.这些改进虽然取得了很好的应用效果,但还都局限于低维简单优化问题.

布谷鸟搜索是基于种群的元启发式搜索算法,尽管具有很好的仿生性能和寻优效率,但其求解复杂系统优化问题和高维函数优化问题时的全局优化能力和收敛速度仍待改善.而Rubinstein等^[15]提出的交叉熵(cross-entropy,CE)方法是一种基于模型的全局随机优化算法,具有很强的全局优化性能和鲁棒性,在复杂网络可靠性分析、函数优化和工程设计优化方面都取得了很好的应用效果^[16].为此,本文探寻用基于种群的CS算法融合基于模型的交叉熵算法,构建一种新的收敛速度更快、搜索精度更高和全局优化性能更好的布谷鸟-交叉熵混合算法(CSCE),以求解复杂函数优化问题.

1 布谷鸟-交叉熵混合算法

1.1 布谷鸟搜索算法

布谷鸟搜索算法是Yang等^[6]于2009年提出的一种新的启发式搜索算法,该算法通过提炼出3个理想化规则形成优化工具,并应用于工程优化和非线性系统优化,取得了很好的效果^[7-11].其规则如下:

- a. 每只布谷鸟随机选取一个宿主鸟巢并产下一枚卵,该卵代表着优化问题的一个候选解 x ;
- b. 位置较好的宿主鸟巢(最优解 x_{best})将被延续到下一代;

c. 宿主鸟巢的数目 n 是固定的,且宿主以一定概率 P_a 发现并丢弃“外来者”或重新筑巢.

布谷鸟搜索的寻优路径和位置更新是通过实施Lévy飞行来实现的,具体为

$$x_i^{t+1} = x_i^t + a \oplus \text{Lévy}(\lambda) \quad (1)$$

式中, t 为Lévy飞行更新代数; $a > 0$, 为Lévy飞行步长缩放控制参数;算子 $\oplus$ 为点对点乘法; $\text{Lévy}(\lambda)$ 为随机搜索路径; λ 为Lévy分布参数,其随机步长 u 服从分布

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

由此产生的随机游走路径是一条马尔可夫链,它的下一个状态或位置只取决于当前位置和转换概率.该路径长短和方向都是不确定的,其中产生的短步长加速局部搜索,而长步长则产生在距离局部最优值较远的地方,这就确保算法不容易陷入局部最小值,从而使得它在探索解空间时会更加有效.

1.2 交叉熵方法

交叉熵方法^[15]是Rubinstein在研究复杂随机网络中的小概率事件估计问题时提出的一种全局随机优化方法.该方法采用重要度抽样,引入Kullback-Leibler距离来度量两个概率分布的交叉熵并使之最小,用以求解组合优化、稀有事件估计以及机器学习等问题,具有很好的随机性、自适应性和鲁棒性^[16],在复杂网络可靠性分析、组合优化,以及工程设计和控制等方面取得了很好的应用效果^[17-18].对于如下最优化问题

$$\min S(X); X \in \mathbb{R}^n \rightarrow \mathbb{R} \quad (3)$$

交叉熵算法将其关联到相关的概率估计问题,根据一族定义在 X 上的概率密度函数 $\{f(x; v), v \in \nu\}$ 随机化问题(3)(v 为概率分布参数, ν 为概率分布参数集或空间),得到其辅助随机问题为

$$l(\gamma) = P(S(X) \leq \gamma) = E[I_{\{S(X) \leq \gamma\}}] \quad (4)$$

式中, E 是期望算子; I 为示性函数; γ 为自适应更新参数.

为了减小样本数量,CE采用重要度抽样法,将式(4)转化为

$$l(\gamma) = \frac{1}{N} \sum_{i=1}^N I_{\{S(x^i) \leq \gamma\}} \frac{f(x^i; v)}{g(x^i)} \quad (5)$$

式中, N 为样本容量; x^i 为重要度抽样密度 $g(x)$ 生成的样本.

为求得最优的重要度抽样密度,引入Kullback-Leibler距离来度量两个概率分布 $f(x; v)$ 与 $g(x)$ 间距离即交叉熵,并最小化交叉熵为

$$\min_v \frac{1}{N} \sum_{i=1}^N I_{\{S(x^i \leq \gamma)\}} \log f(x^i; v) \quad (6)$$

从而得到最优的 $g^*(x)$, 即为式(3)所描述的问题的最优解 x^* 的概率密度函数.

CE 全局随机优化算法的实施过程为:

步骤 1 预定义概率密度函数的均值 $\hat{\mu}$ 和标准差 $\hat{\sigma}$, 生成候选解集;

步骤 2 用引导搜索向全局最优解逼近的精英解集来按式(7)和式(8)更新概率密度函数的参数, 产生迭代序列, 直至满足终止条件.

$$\hat{\mu}_{k+1} = \hat{\alpha} \hat{\mu} + (1 - \hat{\alpha}) \hat{\mu}_k \quad (7)$$

$$\hat{\sigma}_{k+1} = \hat{\beta}_k \hat{\sigma} + (1 - \hat{\beta}_k) \hat{\mu}_k \quad (8)$$

式中, $\hat{\alpha}, \hat{\beta}_k$ 为平滑因子.

1.3 布谷鸟-交叉熵混合算法

a. 算法原理

本文将交叉熵方法嵌入到布谷鸟搜索中, 通过协同演化共同更新种群, 以有效增强算法搜索过程中种群的多样性, 提高算法的全局优化能力. CSCE 通过 CS 和 CE 两个优化迭代算子实现, 其中 CE 算子利用共同更新后的种群来刷新自己的抽样概率分布的参数, 加快均值和方差的演化速度. 这样利用协同演化的种群来加速 CSCE 算法的演化进程, 大幅减少抽样样本数, 进而降低计算成本, 充分发挥自己的全局优化能力, 为协同演化提供全局更优的种群. 同时, CS 算子通过协同演化获得更好的个体, 极大地丰富了种群的多样性, 从而提高 CSCE 算法的收敛速度, 并增强全局优化能力. 具体算法流程见图 1.

b. 算法步骤

基于布谷鸟搜索方法和交叉熵随机优化算法所融合而成的新型启发式算法 (CSCE) 的具体算法步骤如下:

步骤 1 确定搜索空间, 设置 CS 基本参数: 鸟巢数目 N , 发现概率 P_a , 步长因子 α , 搜索精度 ϵ_{CS} 或最大搜索次数 M_{CS} ; 设置 CE 基本参数: 样本容量 N_c 和有效样本数 N_e , 搜索精度 ϵ_{CE} 或最大迭代次数 M_{CE} , 初始化概率分布参数均值 $\hat{\mu}$ 和标准方差 $\hat{\sigma}$. 随机生成初始种群 X_{CS} , 评估每个个体, 得到最优位置 x_{best} 和最优适应度值 f_{min} .

步骤 2 检测迭代终止条件 1, 若不满足, 则启动 CS 优化迭代算子, 否则迭代结束.

(a) 利用 Lévy 飞行机制, 按式(1)得到一组新

的鸟巢位置, 并与上一代进行比较替换, 得到一组更优的鸟巢位置;

(b) 利用发现遗弃机制, 更新鸟巢位置, 产生随机数 $r \in [0, 1]$, 若 $r > P_a$, 则放弃旧巢另建新巢, 否则维持原状;

(c) 评估鸟巢, 更新最优位置和最优适应度值.

步骤 3 检测迭代终止条件 2, 若不满足, 则启动 CE 优化迭代算子, 否则转到步骤 2.

(a) 按 $Y \sim N(\hat{\mu}, \hat{\sigma}^2)$ 抽取样本 $Y_1, Y_2, \dots, Y_L$, 得到 CE 优化迭代算子的种群 X_{CE} ;

(b) 启动协同演化, 混合 X_{CS} 和 X_{CE} 并进行排序, 更新 X_{CS} 和 X_{CE} , 更新最优位置和最优适应度值;

(c) 选取位置最好的 N_e 个鸟巢来计算参数 $\tilde{\mu}$ 和 $\tilde{\sigma}$, 并用式(7)和式(8)平滑更新 $\hat{\mu}$ 和 $\hat{\sigma}$, 返回到步骤 3.

步骤 4 迭代结束, 输出最优鸟巢位置 (最优解) 和最优适应度值 (最优值).

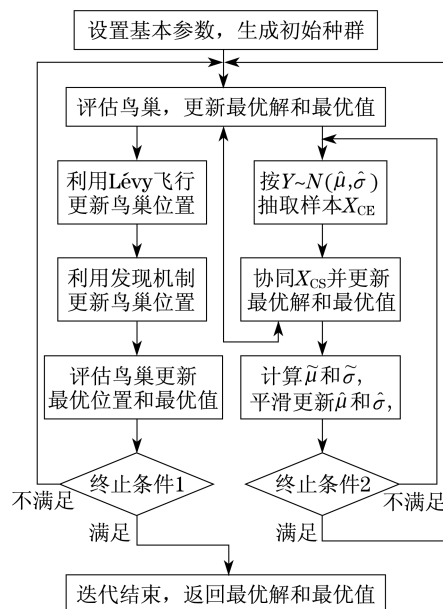


图 1 CSCE 算法流程图

Fig.1 Flow chart for CSCE algorithm

2 CSCE 性能仿真测试

2.1 测试问题与比较对象

为了验证 CSCE 算法求解复杂函数优化问题时的性能, 选取 6 个经典的标准测试函数来进行测试, 解空间维数分别取 50 和 100 两种情形, 并与 CS 算

法、文献[12]提出的布谷鸟改进算法(modified cuckoo search, MCS)和文献[13]提出的改进算法(improved cuckoo search, ICS)进行对比,各测试函

数的具体表达式、定义域及最优值如表1所示.实验硬件环境为 Intel(R) Core(TM) i3 CPU M 2.27 GHz, 2 GB RAM;软件环境为 Windows 7 和 Matlab 2012b.

表1 6个标准测试函数
Tab.1 Six benchmark functions

函数名	表达式	定义域	最优值
Rosenbrock	$f_1(x) = \sum_{i=1}^d [(1-x_i)^2 + 100(x_{i+1} - x_i^2)^2]$	$x_i \in [-100, 100]$	0
Sphere	$f_2(x) = \sum_{i=1}^d x_i^2$	$x_i \in [-5.12, 5.12]$	0
Rastrigin	$f_3(x) = 10d + \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i)]$	$x_i \in [-5.12, 5.12]$	0
Ackley	$f_4(x) = -20\exp\left[-0.2\sqrt{\frac{1}{d}\sum_{i=1}^d x_i^2}\right] - \exp\left[\frac{1}{d}\sum_{i=1}^d \cos(2\pi x_i)\right] + 20 + e$	$x_i \in [-32.768, 32.768]$	0
Griewank	$f_5(x) = \frac{1}{4000}\sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$x_i \in [-600, 600]$	0
Schwefel	$f_6(x) = \sum_{i=1}^d [-x_i \sin(\sqrt{ x_i })]$	$x_i \in [-500, 500]$	-418.982 9d

2.2 参数设置与测试结果

4种算法相关参数设置为:鸟巢数目设定为解空间维数 d , CS, MCS 和 ICS 这3种算法的最大迭代次数均为2 050, 迭代停止标准为达到最大迭代次数, 其它参数分别同文献[6, 12-13]; CSCE 算法中 CS 的最大迭代次数为50, 其它参数设置同文献[6]; CE 样本容量 N_c 和有效样本数 N_e 分别为 d 和30, 最大迭代次数为40. 如此设置参数使得4种算法具有相同的函数评价次数, 以便于对比. 4种算法对每个测试函数独立运行30次, 测试和对比结果见表2和表3(见下页). 表中第一列为6个测试函数, 平均值反映了算法的寻优能力, 标准差反映了算法寻优能力的稳定性, 其最好值均以蓝色字体标识.

从表2和表3可以看出:a. 6个经典测试函数两种维数取值的测试结果中 CSCE 算法在 f_1, f_2, f_4 和 f_5 这4个函数上完全胜出, 优势显著, 在 f_3 上略逊于其它3个函数, 但求解精度仍在一个数量级上, 求解仍然是有效的, 只是函数 f_6 的求解效果要差于其它3种算法, 同时可以看出, 另外两种改进算法的效果并不明显;b. CSCE 算法的搜索精度非常高, 对 f_5 的求解已到达理论最优值, 同时所有求解的标准差都很小, 表明该算法稳定性好、鲁棒性强;c. 随着维数的增加, CSCE 算法的寻优效能并没有发生明显的减弱, 表明该算法更适用于求解高维函数优化问题.

4种算法收敛特征曲线如下页图2和图3所

表2 变量维数 $d=50$ 时4种算法测试结果对比

Tab.2 Comparison of test results of four algorithms with dimension $d=50$

函数	算法	最好值	最差值	平均值	标准差
f_1	CS	5.03E+01	3.05E+02	1.27E+02	5.47E+03
	MCS	2.98E+02	8.11E+02	6.16E+02	2.21E+04
	ICS	7.82E+01	2.03E+02	1.52E+02	1.33E+03
	CSCE	4.70E+01	4.74E+01	4.72E+01	1.11E-02
f_2	CS	1.47E-07	5.43E-07	2.49E-07	1.35E-14
	MCS	3.93E-04	3.70E-03	1.80E-03	6.85E-07
	ICS	1.19E-05	4.63E-05	2.70E-05	8.89E-11
	CSCE	3.81E-61	9.06E-61	5.89E-61	0.00E+00
f_3	CS	1.14E+02	1.49E+02	1.37E+02	1.04E+02
	MCS	1.74E+02	2.35E+02	2.13E+02	3.09E+02
	ICS	1.36E+02	2.07E+02	1.71E+02	6.28E+02
	CSCE	2.21E+02	2.57E+02	2.39E+02	1.86E+02
f_4	CS	6.69E-01	2.58E+00	1.45E+00	7.05E-01
	MCS	1.37E+01	1.89E+01	1.68E+01	3.04E+00
	ICS	4.42E+00	6.77E+00	5.77E+00	5.00E-01
	CSCE	4.44E-15	7.99E-15	6.93E-11	2.95E-30
f_5	CS	1.05E-04	8.90E-03	2.10E-03	7.60E-06
	MCS	3.72E-01	1.02E+00	7.59E-01	3.69E-02
	ICS	1.90E-02	6.37E-02	3.67E-02	2.06E-04
	CSCE	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_6	CS	-1.45E+04	-1.34E+04	-1.40E+04	1.23E+05
	MCS	-1.39E+04	-1.33E+04	-1.37E+04	3.07E+04
	ICS	-1.40E+04	-1.34E+04	-1.37E+04	4.02E+04
	CSCE	-8.18E+03	-7.04E+03	-7.64E+03	1.27E+05

表 3 变量维数 $d = 100$ 时 4 种算法测试结果对比
Tab.3 Comparison of test results of four algorithms with dimension $d = 100$

函数	算法	最好值	最差值	平均值	标准差
f_1	CS	1.57E+03	4.04E+03	2.89E+03	5.53E+05
	MCS	1.73E+03	4.93E+03	3.41E+03	1.10E+06
	ICS	4.28E+03	7.63E+03	5.75E+03	9.21E+05
	CSCE	9.63E+01	9.70E+01	9.67E+01	4.49E-02
f_2	CS	3.80E-03	7.90E-03	5.50E-03	1.65E-06
	MCS	3.40E-03	6.60E-03	5.30E-03	1.62E-06
	ICS	1.21E-02	2.04E-02	1.65E-02	7.87E-06
	CSCE	4.07E-41	7.93E-41	6.15E-41	1.87E-82
f_3	CS	3.82E+02	4.72E+02	4.05E+02	7.22E+02
	MCS	3.62E+02	4.44E+02	4.08E+02	8.92E+02
	ICS	4.65E+02	5.30E+02	4.83E+02	3.19E+02
	CSCE	6.29E+02	6.92E+02	6.59E+02	4.02E+02
f_4	CS	3.44E+00	4.20E+00	3.76E+00	6.80E-02
	MCS	2.99E+00	4.32E+00	3.54E+00	1.64E-01
	ICS	8.89E+00	1.11E+01	9.91E+00	4.22E-01
	CSCE	7.99E-15	1.51E-14	9.77E-15	6.31E-30
f_5	CS	8.41E-01	1.03E+00	9.21E-01	3.20E-03
	MCS	7.91E-01	1.04E+00	9.41E-01	7.80E-03
	ICS	1.02E+00	1.11E+00	1.07E+00	7.06E-04
	CSCE	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_6	CS	-2.44E+04	-2.38E+04	-2.41E+04	3.82E+04
	MCS	-2.50E+04	-2.35E+04	-2.43E+04	2.46E+05
	ICS	-2.51E+04	-2.43E+04	-2.47E+04	6.38E+04
	CSCE	-1.17E+04	-1.04E+04	-1.10E+04	1.96E+05

示,鉴于篇幅所限仅给出 $d = 50$ 时 f_1 和 f_5 的迭代过程.其中,横轴为迭代次数,纵轴为最优函数值,且纵轴采用对数刻度.

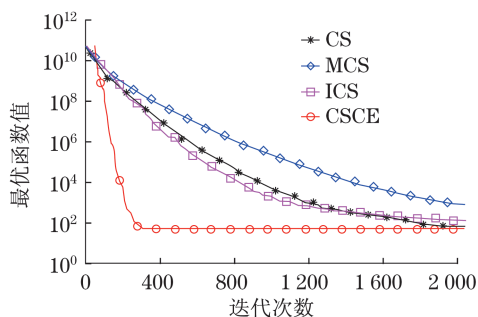


图 2 f_1 的 4 种算法收敛特征曲线对比

Fig.2 Comparison curves of convergence performance of four algorithms for f_1

从图 2 和图 3 可以看出 CSCE 算法的收敛速度明显快于 CS, MCS 和 ICS 3 种算法,同时其求解精度也是最高的.进一步表明 CSCE 算法不仅提高了

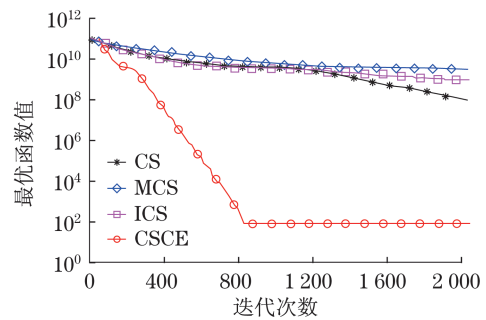


图 3 f_5 的 4 种算法收敛特征曲线对比

Fig.3 Comparison curves of convergence performance of four algorithms for f_5

搜索精度,增强了算法稳定性,同时也改善了算法收敛速度.

3 CSCE 算法在 PID 控制器参数整定中的应用

3.1 PID 控制器参数整定问题

PID 控制器参数整定问题是工业工程中一个难解的优化问题.常用的整定方法有经典的 Ziegler-Nichols 法和智能算法(如 GA, PSO 等),但优化效果都不理想.为此,很多学者提出了一些改进方法,如文献[19]的基于强化缓冲算子的灰色预测 PID 控制以及文献[20]的分数阶 PID 控制,这些方法较传统的 PID 控制都有所改进.本文采用 CSCE 算法来进行 PID 参数整定,并与 CS, MCS 和 ICS 3 种算法进行比较.

假设某二阶延迟系统被控对象传递函数为^[21]

$$G(s) = \frac{2}{s^2 + 1.5s + 2} e^{-0.2s} \quad (9)$$

其 PID 控制器数学模型为

$$G_c(s) = k_p + \frac{k_i}{s} + k_d s \quad (10)$$

式中, s 为传递函数的自变量; $0.01 \leq k_p \leq 20$, $k_i \geq 0.01$, $k_d \leq 2$ 为 3 个待整定参数,通过调整这 3 个参数使得系统满足优化指标要求.

本文参照文献[22]采用如下优化指标来衡量参数整定效果的优劣,即

$$J = \int_0^\infty (w_1 |e(t)| + w_2 u(t)) dt + w_3 t_u \quad (11)$$

式中, $e(t)$ 为系统误差; $u(t)$ 为控制输出; t_u 为上升时间; w_1, w_2, w_3 为权值.为避免超调,当 $e(t) < 0$ 时,上式积分项中加入惩罚项 $w_4 |e(t)|$,且其权值 $w_4 \gg w_1$.本文 4 个权值分别取为 0.999, 0.001, 100 和 $2^{[21]}$,且 J 值越小,整定效果越好.

3.2 PID 控制器参数整定仿真

如上文所描述的 PID 控制器参数整定问题是一种优化问题,其解空间维数为 3,适应度函数如式(11)所定义.4 种算法的参数设置:鸟巢数目为 40;CS,MCS 和 ICS 的最大迭代次数为 100,CSCE 中 CS 和 CE 最大迭代次数均为 10,迭代终止条件为达到最大迭代次数;CE 中的样本数和有效样本数分别为 36 和 15;4 种算法的其它参数如前所设.如此设定以使得它们具有相同的函数评价次数 4 000 次,仿真结果如表 4 所示.

表 4 4 种算法的 PID 控制参数整定结果对比

Tab.4 Comparison of tuning results of four algorithms for PID control parameters

整定算法	k_p	k_i	k_d	J
CS	5.811 3	2	2	107.533 6
MCS	5.820 9	2	2	107.536 4
ICS	5.850 8	2	2	107.537 2
CSCE	5.811 0	2	2	107.533 5

4 种算法对性能指标 J 的优化过程及控制对象的单位阶跃响应曲线如图 4 和图 5 所示(y 为输出信号, t 为时间).

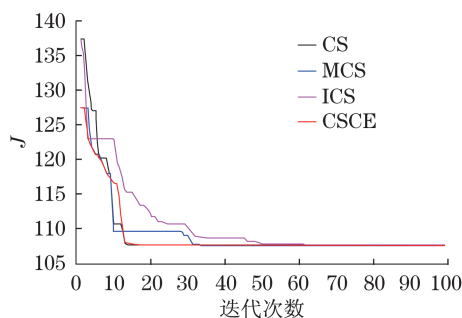


图 4 性能指标 J 的 4 种算法优化过程对比

Fig.4 Comparison of optimization process of four algorithms for performance index J

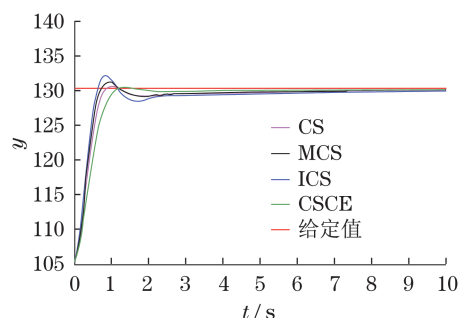


图 5 系统单位阶跃响应曲线

Fig.5 Unit-step response curve of system

由图 4 可见,采用本文所提出的整定方法性能

指标优化速度最快,且精度高于其它 3 种算法.图 5 表明采用本文方法整定参数时系统的超调量很微小且响应衰减迅速,稳定时间短,体现了控制器具有较好的抗干扰能力,同时也表明本文所建算法求解 PID 参数整定问题是可行和有效的.

4 结束语

针对布谷鸟搜索求解复杂函数优化问题的优化性能不强的缺陷,基于交叉熵全局随机优化算法和协同演化的思想,构建了一种新的布谷鸟-交叉熵混合算法.该算法将交叉熵随机优化技术嵌入到布谷鸟搜索过程中,利用交叉熵方法的随机性、自适应性和鲁棒性来改善布谷鸟搜索的全局寻优能力.同时,通过协同演化利用布谷鸟搜索来加快交叉熵算法概率分布参数的收敛速度,从而有力地保证新算法能快速获得全局最优解.标准测试函数和 PID 控制参数整定问题的仿真实验结果表明,CSCE 具有比 CS 算法本身及其一些改进算法更好的寻优性能和稳定性,搜寻精度高,收敛速度快,用其来求解复杂函数优化问题是可行和有效的.

参考文献:

- [1] Schoen F. Global optimization methods for high-dimensional problems [J]. European Journal of Operational Research, 1999, 119 (2): 345 - 352.
- [2] Grosan C, Abraham A, Hassainen A E. A line search approach for high dimensional function optimization [J]. Telecommunication Systems, 2011, 46 (3): 217 - 243.
- [3] Zhao S Z, Liang J J, Suganthan P N, et al. Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization evolutionary computation [C] // Proceedings of the 2008 IEEE Congress on Evolutionary Computation. Hong Kong: IEEE Computer Press, 2008: 3845 - 3852.
- [4] Yang Z Y, Tang K, Yao X. Large scale evolutionary optimization using cooperative coevolution [J]. Information Sciences, 2008, 178 (15): 2985 - 2999.
- [5] Kim D H, Abraham A, Cho J H. A hybrid genetic algorithm and bacterial foraging approach for global optimization [J]. Information Science, 2007, 177 (18): 3918 - 3937.
- [6] Yang X S, Deb S. Cuckoo search via Lévy flights [C] // Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing. Coimbatore: IEEE

- Computer Press, 2009: 210 - 214.
- [7] Moravej Z, Akhlaghi A. A novel approach based on cuckoo search for DG allocation in distribution network[J]. International Journal of Electrical Power & Energy Systems, 2013, 44(1): 672 - 679.
- [8] Gandomi A H, Yang X S, Alavi A H. Cuckoo search algorithm; a metaheuristic approach to solve structural optimization problems [J]. Engineering with Computers, 2013, 29(1): 17 - 35.
- [9] Chandrasekaran K, Simon S P. Multi-objective scheduling problem: hybrid approach using fuzzy assisted cuckoo search algorithm [J]. Swarm and Evolutionary Computation, 2012, 5: 1 - 16.
- [10] 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.
- [11] 刘长平, 叶春明. 求解置换流水车间调度问题的布谷鸟算法[J]. 上海理工大学学报, 2013, 35(1): 17 - 20.
- [12] 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.
- [13] Valian E, Tavakoli S, Mohanna S, et al. Improved cuckoo search for reliability optimization problems [J]. Computers & Industrial Engineering, 2013, 64(1): 459 - 468.
- [14] 冯登科, 阮奇, 杜利敏. 二进制布谷鸟搜索算法[J]. 计算机应用, 2013, 33(6): 1566 - 1570.
- [15] Rubinstein R Y, Kroese D P. The cross-entropy method: a unified approach to combinatorial optimization, Monte Carlo simulation and machine learning[M]. New York: Springer - Verlag, 2004.
- [16] Kroese D P, Portsky S, Rubinstein R Y. The cross-entropy method for continuous multi-extremal optimization [J]. Methodology and Computing in Applied Probability, 2006, 8(3): 383 - 407.
- [17] 李洁, 柴天佑, 宫经宽. 基于交叉熵算法的 PID 控制器设计[J]. 控制与决策, 2011, 26(5): 794 - 796.
- [18] Yildiz T, Yercan F. The cross-entropy method for combinatorial optimization problems of seaport logistics terminal [J]. Transport, 2010, 25(4): 411 - 422.
- [19] 朱坚民, 黄之文, 翟东婷, 等. 基于强化缓冲算子的灰色预测 PID 控制仿真研究[J]. 上海理工大学学报, 2012, 34(4): 327 - 332.
- [20] 于莲芝, 成玲玲. 分数阶 PID 控制运用于励磁控制系统[J]. 上海理工大学学报, 2013, 35(4): 404 - 408.
- [21] 张朝龙, 江巨浪, 江善和, 等. 一种自适应混合粒子群优化算法及其应用[J]. 计算机应用研究, 2011, 28(5): 1696 - 1698.
- [22] 刘金琨. 先进 PID 控制及其 Matlab 仿真 [M]. 3 版. 北京: 电子工业出版社, 2011.

(编辑: 丁红艺)

(上接第 179 页)

参考文献:

- [1] 孙延禄. 从模拟人眼立体视觉功能的差异看 3D 影像技术的类别与未来[J]. 现代电影技术, 2012(1): 13 - 23.
- [2] 尹杰, 靳静娜, 刘志朋, 等. 观看 3D 影视所致不适感的 EEG 初步分析[J]. 北京生物医学工程, 2012, 31(2): 150 - 153.
- [3] 李志斌, 郑刚, 章立新, 等. 基于数字全息术的粒子定位测量[J]. 上海理工大学学报, 2007, 29(3): 237 - 240.
- [4] 汪伟, 李郝林. 基于数码相机图像的三维重构[J]. 上海理工大学学报, 2005, 27(5): 429 - 432.
- [5] 孙伟杰. 3D 影视制作与播放技术探析[J]. 现代电影技术, 2010(9): 18 - 23.
- [6] 孙延禄. 3D 电影摄影基本技术操作参数的选择与计算方法[J]. 现代电影技术, 2011(4): 17 - 22.
- [7] 于路, 王怀军. 3D 拍摄技术与艺术探讨[J]. 广播与电视技术, 2013(7): 68 - 71.
- [8] 赵慧敏, 姜秀华. 基于视频加深度格式的立体视频技术分析[J]. 电视技术, 2014, 38(1): 2 - 5.
- [9] 庞燕. 立体视频的编辑与播放[J]. 科技信息, 2011(7): 57 - 32.
- [10] 赵上元. 3D 影视作品观看舒适度问题的探讨[J]. 现代电影技术, 2011(10): 21 - 28.

(编辑: 丁红艺)