

控制与决策

Control and Decision

双重驱动的果蝇优化算法及其在PID控制器中的应用

凤丽洲, 王友卫, 韩琳琳, 杨贵军

引用本文:

凤丽洲, 王友卫, 韩琳琳, 等. 双重驱动的果蝇优化算法及其在PID控制器中的应用[J]. *控制与决策*, 2021, 36(9): 2225–2233.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2020.0046>

您可能感兴趣的其他文章

Articles you may be interested in

基于PID参数整定的线性自抗扰控制参数整定

Tuning of linear active disturbance rejection controllers based on PID tuning rules
控制与决策. 2021, 36(7): 1592–1600 <https://doi.org/10.13195/j.kzyjc.2019.1408>

基于复杂昂贵仿真的体系效能多目标优化

Complex and expensive simulation based multi-objective optimization to system-of-system effectiveness
控制与决策. 2021, 36(3): 589–598 <https://doi.org/10.13195/j.kzyjc.2019.0844>

求解约束优化问题的改进果蝇优化算法及其工程应用

Improved fruit fly optimization algorithm for solving constrained optimization problems and engineering applications
控制与决策. 2021, 36(2): 314–324 <https://doi.org/10.13195/j.kzyjc.2019.0557>

四旋翼无人机抗干扰轨迹跟踪控制

Anti-interference trajectory tracking control of quadrotor UAV
控制与决策. 2021, 36(2): 379–386 <https://doi.org/10.13195/j.kzyjc.2019.0875>

基于知识粒度特征的多目标粗糙集属性约简算法

Multi objective rough set attribute reduction algorithm based on characteristics of knowledge granularity
控制与决策. 2021, 36(1): 196–205 <https://doi.org/10.13195/j.kzyjc.2019.0490>

双重驱动的果蝇优化算法及其在PID控制器中的应用

凤丽洲¹, 王友卫^{2†}, 韩琳琳¹, 杨贵军¹

(1. 天津财经大学 统计学院, 天津 300222; 2. 中央财经大学 信息学院, 北京 100081)

摘要: 针对传统群智能优化算法面临计算复杂性高、参数依赖性强、全局优化能力弱的问题, 在传统果蝇优化算法中引入细菌趋化理论, 提出一种基于双重驱动的果蝇优化算法. 综合考虑优势果蝇群体和劣势果蝇群体的分布特点, 提出多驱逐剂与多引诱剂的概念, 并在二者的双重驱动下更新果蝇位置, 避免传统果蝇方法在位置更新过程中单纯依靠局部最优(差)位置带来的无效搜索. 利用果蝇适应值信息, 提出多驱逐剂和多引诱剂的带权质心向量计算方法, 自适应地确定果蝇搜索半径, 避免传统方法面临的参数依赖性强问题. 在典型测试函数上的实验结果表明, 所提出算法较现有典型算法参数依赖性小、收敛精度高、收敛速度快, 且其优化后的PID控制器响应速度快, 稳定性高, 验证了其在PID参数优化领域的有效性.

关键词: 群智能优化; 果蝇优化算法; 细菌趋化; 多驱逐剂; 多引诱剂; PID控制器

中图分类号: TP273

文献标志码: A

DOI: 10.13195/j.kzyjc.2020.0046

开放科学(资源服务)标识码(OSID):



引用格式: 凤丽洲, 王友卫, 韩琳琳, 等. 双重驱动的果蝇优化算法及其在PID控制器中的应用[J]. 控制与决策, 2021, 36(9): 2225-2233.

Double drive fruit fly optimization algorithm and its application in PID controller

FENG Li-zhou¹, WANG You-wei^{2†}, HAN Lin-lin¹, YANG Gui-jun¹

(1. School of Statistics, Tianjin University of Finance and Economics, Tianjin 300222, China; 2. School of Information, Central University of Finance and Economics, Beijing 100081, China)

Abstract: In order to solve the problems of high computational complexity, strong parameter dependence and weak global optimization ability of traditional swarm intelligence optimization algorithm, a fruit fly optimization algorithm based on double drive with the theory of bacterial chemotaxis is proposed. Considering the distribution of the superior and the inferior fruit fly groups, the concepts of multiple repellents and multiple attractants are proposed, and the location of fruit fly is updated under the double drive, so as to avoid the invalid search of the traditional methods which only depend on the local best (worst) position during the updating process. Then, based on the fitness value information of the fruit flies, a weighted centroid vector calculation method of multiple repellents and multiple attractants is proposed to determine the searching radiuses of fruit flies adaptively and avoid the problem of strong parameter dependence faced by traditional methods. The experimental results on standard functions show that, the proposed method has lower parameter dependence, higher convergence accuracy and convergence speed than existing typical algorithms. Moreover, the PID controller optimized by the proposed method has high response speed and stability, showing the ability of the proposed algorithm on PID parameter optimization.

Keywords: swarm intelligence optimization; fruit fly optimization algorithm; bacterial chemotaxis; multiple repellents; multiple attractants; PID controller

0 引言

在工业过程控制领域中, PID控制技术具有重要地位^[1]. 传统PID参数优化方法多采用试验加试凑的人工方法, 精确度低, 耗时久, 很难适应自动化的需求,

因此, 如何有效实现参数优化成为提高PID控制技术的关键^[2-3].

近年来, 受生物学启发, 学界将群智能优化算法引入科学^[4]、电力^[5]、医学^[6]、服务^[7]等领域的参数优

收稿日期: 2020-01-10; 修回日期: 2020-04-21.

基金项目: 国家社会科学基金项目(18CTJ008); 天津市自然科学基金项目(18JJCQNJC69600); 国家自然科学基金项目(61906220); 教育部人文社科基金项目(19YJCZH178).

责任编辑: 林崇.

†通讯作者. E-mail: wyw4966198@126.com.

化问题中,亦有学者提出利用群智能优化算法选择PID最优参数^[8]。然而,传统群智能优化算法,如:蚁群算法^[3]、遗传算法^[8]、粒子群算法 particle swarm optimization, PSO^[9]、差分进化算法 (differential evolution algorithm, DE)^[10]等,存在计算复杂性高、参数依赖性强、全局优化能力有限等问题。文献[11-12]提出了果蝇优化算法 (fruit fly optimization algorithm, FOA),通过模拟果蝇的觅食行为获得全局最优参数。相较于传统群智能优化算法,FOA凭借其计算复杂性低、参数依赖性弱的优势,广泛应用于PID参数优化领域。

为避免FOA算法陷入局部最优,韩俊英等^[13]通过引入混沌算法,提出了具有较强全局搜索能力的自适应混沌FOA(ACFOA)。为避免无效的全局搜索,韩俊英等^[14]基于最优和最差个体提出了改进FOA算法(BWFOA),通过增加最差个体的学习过程改进果蝇进化策略,以提高其跳出局部最优的能力。此外,韩俊英等^[15]还在细菌趋化性的基础上提出了果蝇优化算法BCFOA,引入吸引和排斥机制,通过计算种群的适应度方差决定是否执行。该算法解决了传统FOA算法只关注当前全局最优解导致易陷入局部最优的问题,但忽略了多个最差个体的整体排斥效应,容易在种群迭代后期出现全局搜索失效的现象。上述方法均直接利用FOA算法中的气味浓度判断值选择全局最优果蝇,因此无法实现针对负参数的优化。为此,Pan等^[16]去掉气味浓度判定值,提出了一种改进的FOA算法(IFFO),直接使用果蝇的位置向量计算适应度,动态调整搜索半径以提高全局搜索能力。此外,为扩大果蝇搜索范围,张清勇等^[17]提出一种新型二级协作果蝇优化算法(TCFOA),利用二级结构扩大搜索范围,有效提高了获得高质量解的可能性。韩俊英等^[18]通过在优化方程中添加“历史认知”调整迭代寻优过程中历史位置对本次学习的影响,减少寻优轨迹曲折迂回而错过全局最优解的现象。

本文在BWFOA和BCFOA算法的基础上,综合考虑优势群体和劣势群体对个体位置更新过程的影响,提出基于多驱逐剂反向驱动和多引诱剂正向牵引的果蝇优化算法。算法的主要特点包括:1)为避免传统FOA算法单纯依靠全局最优位置更新果蝇位置可能造成的局部收敛问题,提出多引诱剂的概念,在加快收敛速度的同时解决局部最优问题;2)引入细菌反向趋化理论,提出多驱逐剂概念,避免BWFOA及BCFOA算法单纯依靠局部最优及局部最差个体更新果蝇位置造成的偏差;3)在多驱逐剂与多引诱剂

双重驱动下,提出针对多驱逐剂和多引诱剂的带权质心向量计算方法,实现果蝇种群搜索半径自动确定。实验结果表明,所提出算法能有效解决传统果蝇算法面临的参数依赖性强、收敛速度慢、易陷入局部极值等问题,并且能使得PID控制器获得较高的准确性和稳定性。

1 PID控制理论

PID控制器(比例-积分-微分控制器)是生产控制领域中常用的反馈回路元件^[9]。给定控制对象 M ,PID控制器的执行原理如图1所示。

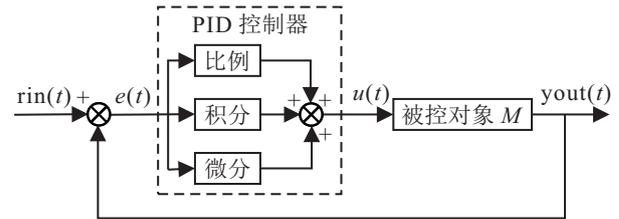


图1 PID控制器原理

图1中: $rin(t)$ 、 $yout(t)$ 和 $e(t)$ 分别为系统的输入信号、输出信号和系统误差。由图1可见,PID控制器输入 $e(t)$ 通过比例单元P、积分单元I和微分单元D的线性组合输出 $u(t)$ 。PID控制器的传递函数^[9]如下所示:

$$u(t) = K_p \times e(t) + K_i \times \int_0^1 e(t)dt + K_d \times \frac{d(e(t))}{dt} \quad (1)$$

其中: K_p 为比例因子, K_i 为积分系数, K_d 为导数系数。研究表明, K_p 、 K_i 和 K_d 对PID控制器的性能有很大影响,适当选择这些参数可以减小系统误差,否则将直接影响控制系统稳定性,给控制系统的安全带来隐患。

2 传统FOA算法

受到果蝇在觅食过程中凭借敏锐的嗅觉和视觉发现食物最优位置的启发,潘文超^[11-12]提出了果蝇优化算法FOA。在算法每次迭代过程中,每个果蝇个体根据共享的当前最佳果蝇位置确定自己下一次迭代应该选择的新位置,执行步骤如下。

算法1^[11-12] 果蝇优化算法FOA。

输入:由 N 只果蝇构成的种群 X ,最近的食物位置 X_{min} ,最远的食物位置 X_{max} ,最大迭代次数 T ;

输出:食物最优位置 $X_B = (x_{bx}, x_{by})$,最优气味浓度 $smell_{gb}$ 。

step 1: 初始化如下参数:果蝇数量 N 、最大迭代次数 T 、全局最优食物位置 $X_B = (x_{bx}, x_{by})$ 、全局最佳气味浓度 $smell_{gb}$ 。

step 2: 由式(2)获得每个果蝇的位置 $x_i = (x_{ix}, x_{iy})$,

x_{iy}), 有

$$x_{ix} = x_{bx} + \text{rand}(), \quad x_{iy} = x_{by} + \text{rand}(). \quad (2)$$

step 3: 计算 x_i 对应的气味浓度判定值 s_i , 有

$$d_i = \sqrt{x_{ix}^2 + x_{iy}^2}, \quad s_i = 1/d_i. \quad (3)$$

step 4: 根据适应度函数 fit 求得果蝇 x_i 的气味浓度 smell_i , 有

$$\text{smell}_i = \text{fit}(s_i). \quad (4)$$

step 5: 找到种群中气味浓度最大的果蝇, 并将其位置和气味浓度分别表示为 (x'_{bx}, x'_{by}) 和 smell_b .

step 6: 如果 $\text{smell}_b > \text{smell}_{gb}$, 则设位置 (x'_{bx}, x'_{by}) 为果蝇最优位置 x_b .

step 7: 重复 step 3 ~ step 6, 直到达到最大迭代次数 T 为止.

3 本文算法

趋化性是细菌觅食过程中的一个重要现象, 细菌会感知环境并作出调整: 若接近引诱剂可以增加细菌个体的生存几率, 则逃离驱逐剂有助于细菌在别处找到更好的食物^[15]. 为了解决局部最优问题, BCFOA 和 BWFOA 在 FOA 中引入细菌吸引和排斥机制, 并通过计算种群的适应度方差确定是否执行, 其工作原理及存在的不足如图2所示.

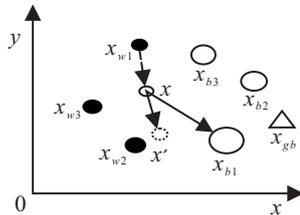


图2 BCFOA 和 BWFOA 的原理及不足

图2中: x_{w1} 、 x_{b1} 分别为当前最差果蝇及当前最优果蝇, 果蝇个体 x 将在最差个体 x_{w1} 和最优个体 x_{b1} 的驱动下前进, 更新后位置为 x' . 然而, 假设在该次迭代过程中, 图2中的 x_{w1} 、 x_{w2} 和 x_{w3} 是表现最差的3个果蝇个体(其形状大小表示对应的适应值大小), x_{b1} 、 x_{b2} 和 x_{b3} 是表现最好的3个果蝇个体, x_{gb} 为全局最优位置. 显然, 更新后位置 x' 仍然在表现较差的果蝇个体 x_{w2} 附近, 较严重地偏离了全局最优位置 x_{gb} .

上述问题存在的原因是: BCFOA 方法和 BWFOA 方法仅使用单个驱逐剂或单个引诱剂确定果蝇个体的新位置, 忽略了适应值在种群中的分布情况, 因此所得结果仅受全局最优位置和全局最差位置的影响, 导致与实际情况产生较大偏差. 为此, 本文提出多驱逐剂和多引诱剂的概念, 并在此基础上提出了一种基于多引诱剂正向引导和多驱逐剂反向驱动的

果蝇优化算法. 相关概念定义如下.

定义1(多引诱剂) 给定果蝇种群 $X = \{x_i\}$ ($1 \leq i \leq N$), 将第 t 次迭代情况下适应值最大的前 n 只果蝇集合称为该迭代次数下的多引诱剂, 记为

$$X_{bt} = \{x_b^1, x_b^2, \dots, x_b^i, \dots, x_b^n\}, \quad 1 \leq j \leq n, \quad x_b^j \in X. \quad (5)$$

定义2(多驱逐剂) 给定果蝇种群, 将第 t 次迭代情况下适应值最小的前 n 只果蝇集合称为该迭代次数下的多驱逐剂, 记为

$$X_{wt} = \{x_w^1, x_w^2, \dots, x_w^i, \dots, x_w^n\}, \quad 1 \leq j \leq n, \quad x_w^j \in X. \quad (6)$$

定义3(双重驱动) 给定果蝇种群 $X = \{x_i\}$ ($1 \leq i \leq N$), 第 t 次迭代中的果蝇个体 x_i 将受到多引诱剂和多驱逐剂的共同作用, 即以一定概率在多驱逐剂和多引诱剂双重作用下进行位置更新.

在此基础上, 提出一种基于多引诱剂和多驱逐剂双重驱动下的果蝇位置更新过程, 如图3所示. 图3中, 实线圆表示果蝇局部化搜索区域, 为动态确定搜索区域半径 r 的大小, 给出一种基于带权质心向量的搜索半径计算方法. 该方法首先计算多引诱剂和多驱逐剂的带权质心向量, 然后根据质心向量之间的距离动态确定 r 值, 以此避免传统算法对搜索半径取值依赖性强的问题.

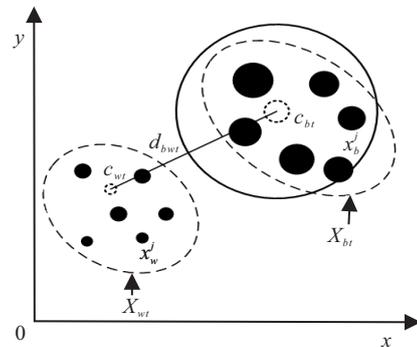


图3 基于多引诱剂和多驱逐剂的果蝇位置更新

综上, 本文果蝇位置更新步骤可描述如下.

step 1: 结合不同种群中个体的适应值大小, 计算多引诱剂和多驱逐剂的带权质心向量, 分别记为 c_{bt} 和 c_{wt} , 有

$$c_{bt} = \{x_b^i\} = \left\{ \frac{1}{n} \sum_{k=1}^n \left(\frac{\text{fit}_b^k}{\text{fit}_b} \times x_{b,i}^k \right) \right\}, \quad (7)$$

$$c_{wt} = \{x_w^i\} = \left\{ \frac{1}{n} \sum_{k=1}^n \left(\frac{\text{fit}_w^k}{\text{fit}_w} \times x_{w,i}^k \right) \right\}. \quad (8)$$

其中: $x_{b,i}^k$ 为多引诱剂 X_{bt} 中的第 k 个个体的第 i 维分量值, $x_{w,i}^k$ 为多驱逐剂 X_{wt} 中的第 k 个个体的第 i 维分量值, fit_b^k 为 X_{bt} 中第 k 个个体的适应值, fit_w^k 为 X_{wt}

中第 k 个个体的适应值, fit_b 、 fit_w 为归一化系数, 定义为

$$fit_b = \sum_{l=1}^n fit_b^l, \quad (9)$$

$$fit_w = \sum_{l=1}^n fit_w^l.$$

在此基础上, 计算位置 c_{bt} 与 c_{wt} 之间的距离 d_{bwt} 及果蝇搜索半径 r 如下:

$$\begin{cases} d_{bwt} = \sqrt{\sum_{j=1}^D (x_{bt}^j - x_{wt}^j)^2}, \\ r = \frac{d_{bwt} \times \delta}{2}, \\ \delta = 1 - t/T. \end{cases} \quad (10)$$

其中: δ 为搜索半径调节因子, t 为当前迭代次数, T 为最大迭代次数.

step 2: 针对第 t 次迭代中的果蝇个体, 利用轮盘赌策略协调 x_i 的全局搜索能力和局部搜索能力. 即: 首先随机选择果蝇个体 x_i 的第 j 个维度, 然后利用下面的方式更新果蝇 x_i 位置:

1) 如果 $rand() < p$, 则 x_i 在整个搜索区间内执行全局化随机搜索, 有

$$x_{it}^j = x_{min} + (x_{max} - x_{min}) \times rand(), \quad (11)$$

其中 x_{max} 、 x_{min} 分别为搜索区间上下边界.

2) 否则, 在多驱逐剂和多引诱剂双重作用下, 以 c_{bt} 为圆心、 r 为半径进行局部化搜索, 有

$$x_{it}^j = x_{bt}^j + (-1)^{\text{floor}(\text{rand}() \times 10^6)} \times r. \quad (12)$$

其中: $rand()$ 随机产生一个 $(0, 1)$ 区间内的随机数; p 为全局-局部协调因子, p 值越大, 算法的全局化搜索能力越强, 收敛得越慢; 反之, 算法的局部化搜索能力越强, 收敛得越快. 为提高算法收敛速度, 设置 $p = t/T$. 可见, 在搜索前期, 果蝇在多引诱剂和多重驱逐剂双重驱动下在 c_{bt} 位置附近搜索; 随着迭代次数增加, 果蝇将逐渐增加全局搜索概率, 以避免收敛过早可能带来的局部极值问题.

依据上述概念及果蝇位置更新方法, 基于双重驱动的果蝇优化算法流程如图4所示. 由图4可见, 本文在更新果蝇位置时考虑到适应值分布的影响, 以多引诱剂和多重驱逐剂的双重驱动为依据, 利用轮盘赌策略协调果蝇的全局搜索能力和局部搜索能力, 可以避免 FOA、BWFOA 及 BCFOA 单纯依靠全局最优位置更新果蝇位置所造成的局部收敛问题和果蝇位置更新偏差. 相应地, 当处理最小值优化问题时, 本文算法的执行步骤如下.

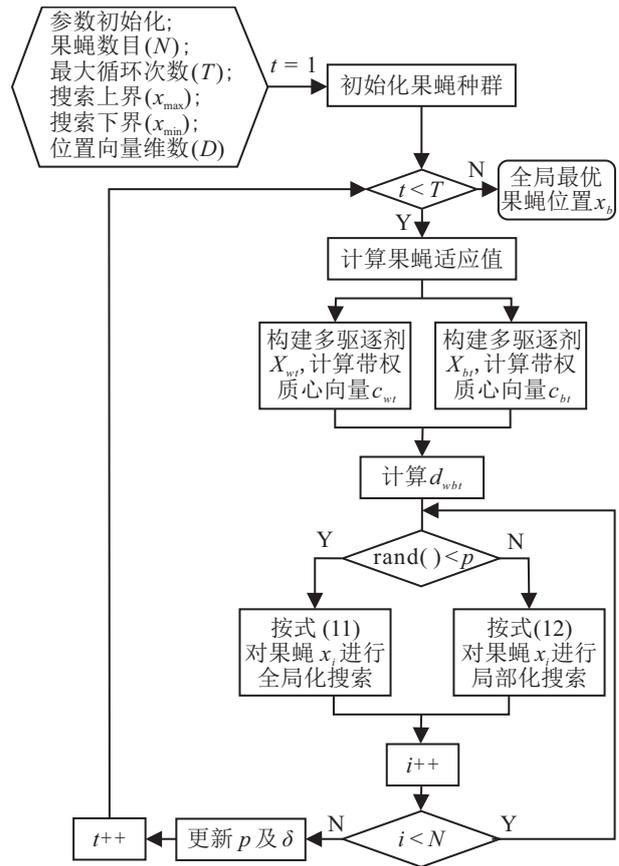


图4 基于双重驱动的果蝇优化算法流程

算法2 基于双重驱动的果蝇优化算法流程.

输入: 果蝇种群 X (个体数量为 N), 搜索下界 x_{min} , 搜索上界 x_{max} , 最大迭代次数 T , 搜索向量维数 D , 多引诱剂及多重驱逐剂中个体数量 n ;

输出: 食物最优位置 x_b .

1: for $i \leftarrow 1 : 1 : N$

2: 由式(13)得到果蝇 x_i 的随机位置

3: for $j \leftarrow 1 : 1 : D$

4: 有

$$x_{ij} \leftarrow x_{min} + (x_{max} - x_{min}) \times rand(),$$

$$0 \leq j \leq D \quad (13)$$

5: 式(13)中 x_{ij} 为 x_i 对应的第 j 维度上的分量

6: end for

7: end for

8: $t \leftarrow 0$

9: while ($t < T$)

10: for $i \leftarrow 1 : 1 : N$

11: 计算 x_i 对应的位置向量的适应值 fit_i

12: end for

13: 分别找到 n 个适应值最高的果蝇和 n 个适应值最低的果蝇构建多重引诱剂 X_{bt} 和多重驱逐剂 X_{wt}

14: 由式(7)~(10)计算多重引诱剂和多重驱逐剂的

带权质心向量 c_{bt} 、 c_{wt} 以及搜索半径 r

```

15: for  $i \leftarrow 1 : 1 : N$ 
16: 由式(11)和(12)更新果蝇  $x_i$  的位置
17: end for
18: 更新  $p$  以及式(10)中的  $\delta$ 
19:  $t^{++}$ 
20: end while
    
```

可见: 1~8为果蝇位置初始化过程, 对应的时间复杂度为 $O(ND)$. 9~20为果蝇位置更新过程. 其中: 10~12为果蝇适应值计算, 时间复杂度为 $O(N)$; 13、14为多驱逐剂、多引诱剂构建及质心距离计算过程, 时间复杂度为 $2 \times [O(N \log N) + O(nD)] + O(D)$; 15~17为果蝇位置更新过程, 对应时间复杂度为 $O(N)$. 综上, 本文算法整体时间复杂度为

$$TC = O(ND) + T \times (O(N) + 2 \times [O(N \log N) + O(nD)] + O(D) + O(N)). \quad (14)$$

化简并去掉其中的常数项, 得

$$TC = O(ND + TN \log N + TD). \quad (15)$$

表1给出了几种常见优化算法的时间复杂度. 可见, IFFO算法时间复杂度最低, BCFOA算法时间复杂度最高. 将本文算法与FOA、ACFOA、PSO等典型算法相比, 有

$$\frac{ND + TN \log N + TD}{TND} = \frac{1}{T} + \frac{\log N}{D} + \frac{1}{N}. \quad (16)$$

由于 $1 \ll T$ 且 $1 \ll N$, 当 $D > \log N$ 时, 本文算

法时间复杂度将低于FOA、ACFOA、PSO等典型算法.

表1 不同算法的时间复杂度对比

算法	时间复杂度
FOA/ACFOA/BWFOA/PSO/DE	$O(TND)$
BCFOA	$O(TN^2D)$
IFFO	$O(N(D+T))$

4 实验结果与分析

为了比较不同算法的优化效果, 将本文算法与PSO^[9]、DE^[10]、FOA^[11,12]、ACFOA^[13]、BWFOA^[14]、BCFOA^[15]和IFFO^[16]7种算法进行对比实验. 为公平起见, 设置所有算法中果蝇数量 $N = 50$.

4.1 参数依赖性比较

为了比较不同算法对于初始参数的依赖性, 表2给出几种典型优化算法所需参数的统计结果. 由表2可知, FOA与BCFOA算法对应参数数目最少(4个), 仅需群体规模、最大迭代次数及搜索范围这几个基本参数. ACFOA算法所需参数数目最多(8个), 原因在于该算法使用Logistic混沌映射增加果蝇产生位置的随机性, 而Logistic算法本身会受到混沌迭代次数、控制参数等因子的影响. 本文算法所需参数数目为5个, 少于BWFOA、IFFO、ACFOA、PSO、DE等典型算法(6个), 表明本文算法相对于后者而言对参数的依赖性较小, 算法易获得更为稳定的寻优结果.

表2 不同算法所需参数统计结果

所需参数	FOA	ACFOA	BWFOA	BCFOA	IFFO	PSO	DE	本文算法
群体规模 N	✓	✓	✓	✓	✓	✓	✓	✓
最大迭代次数 T	✓	✓	✓	✓	✓	✓	✓	✓
特征维数 D	✓	✓	✓	✓	✓	✓	✓	✓
搜索范围最大(小)值 $x_M(x_m)$	✓	✓	✓	✓	✓	✓	✓	✓
混沌映射迭代次数 t_1		✓						
混沌控制参数 u		✓						
方差阈值 δ		✓						
搜索半径最大(小)值 $r_M(r_m)$					✓			
多引诱剂/多驱逐剂个体数量 m								✓
学习系数最大(小)值 $w_{\max}(w_{\min})$			✓					
速度最大(小)值 $v_{\max}(v_{\min})$						✓		
速度调节系数 c_1 、 c_2						✓		
变异概率 va							✓	
交叉概率 cr							✓	

4.2 典型测试函数实验

如表3所示, 使用4个典型的单峰函数 ($F_1 \sim F_4$)^[13,20] 和4个典型的多峰函数 ($F_5 \sim F_8$)^[16,21] 检验

不同算法的性能. 为确保算法的收敛性, 设置最大迭代次数 $T = 2000$.

表3 不同算法的性能对比

函数	公式表示	目标最优(极小)值	目标位置分量值
F_1	$f(x) = \sum_{i=1}^D x_i + 1 $	0	$(-1, -1, \dots, -1)$
F_2	$f(x) = \sum_{i=1}^D (100(x_i - x_i^2)^2 + (x_i - 1)^2)$	0	$(1, 1, \dots, 1)$
F_3	$f(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	0	$(0, 0, \dots, 0)$
F_4	$f(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	0	$(0, 0, \dots, 0)$
F_5	$f(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	0	$(0, 0, \dots, 0)$
F_6	$f(x) = \sum_{i=1}^{D-1} \frac{\sin^2 \sqrt{x_i^2 + x_{i+1}^2} - 0.5}{(1 + 0.001(x_i^2 + x_{i+1}^2))^2} + 0.5$	0	$(0, 0, \dots, 0)$
F_7	$f(x) = \sum_{i=1}^D x_i \sin(x_i) + 0.1x_i $	0	$(0, 0, \dots, 0)$
F_8	$f(x) = \frac{\pi}{d} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)^2 \right\} +$ $\sum_{i=1}^D \mu(x_i, 10, 100, 4),$ $y_i = 1 + \frac{1}{4}(x_i + 1),$ $\mu(x_i, a, k, m) = \begin{cases} k(x_i - a)m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)m, & x_i < -a \end{cases}$	0	$(-1, -1, \dots, -1)$

4.2.1 收敛精度的比较

将每个测试函数的维度 D 分别设置为 2、4、6、8 和 10. 分别针对每一个 D 值进行 100 次实验, 计算每

种算法的平均全局最优值 (v_{ba}) 和达到最优值的成功率均值 (p_h), 结果如表 4 所示. 为便于观察, 各函数对应的最优 v_{ba} 值以加粗形式表示.

表4 不同算法的收敛精度比较

对比算法	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	$p_h/\%$
FOA	1.559	3.695	6.17e-2	2.38e-4	2.56e-2	2.26e-2	2.26e-2	3.226	46.225
IFFO	3.25e-1	9.92e-1	3.01e-5	4.36e-6	8.36e-2	4.39e-2	6.07e-4	5.21e-4	68.636
BWFOA	1.038	18.365	3.58e-2	1.07e-4	5.32e-2	1.97e-1	1.69e-1	5.317	53.183
BCFOA	0.967	16.332	4.06e-3	2.81e-3	1.42e-2	6.22e-1	5.38e-2	6.236	57.562
ACFOA	2.352	1.007	2.97e-2	1.06e-4	3.22e-3	4.69e-3	9.21e-4	1.962	56.362
PSO	6.16e-3	3.46e-2	5.12e-4	1.26e-4	3.67e-3	6.71e-3	9.24e-3	4.25e-4	76.362
DE	2.58e-4	6.85e-4	8.11e-9	5.66e-4	3.38e-3	1.36e-3	6.38e-4	3.88e-4	82.332
本文算法	1.02e-3	3.17e-4	2.35e-4	9.13e-6	6.16e-4	7.14e-4	5.14e-5	2.35e-4	85.361

由表 4 可知, ACFOA 全局优化性能优于 FOA, 表明利用混沌算法可以一定程度跳出局部最优. 除 F_1 、 F_3 外, 对应的结果均优于 DE, 验证了基于多引诱剂和多驱逐剂的双向驱动策略在提升收敛精度方面的有效性. 总体而言, 本文算法对应的 p_h 值最大 (85.361%) 且 4 次获得最优 v_{ba} 值, 表明基于多引诱剂和多驱逐剂双重驱动的果蝇位置更新策略可以有效提高算法全局搜索能力, 提高算法收敛精度.

4.2.2 收敛速度比较

令每个测试函数中的维度 D 分别为 2、4、6、8 和 10. 分别针对每一个 D 值进行 100 次实验, 计算各

算法对应的平均收敛迭代次数 (记为 n_{ba}), 结果如表 5 所示.

表5 不同算法的收敛速度比较

对比算法	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
FOA	506	743	302	624	716	297	896	627
IFFO	622	839	320	668	862	407	1035	882
BWFOA	192	336	169	277	246	251	477	569
BCFOA	662	927	454	774	974	638	1242	1214
ACFOA	679	923	532	817	1023	545	1289	1031
PSO	125	139	128	219	159	197	325	286
DE	468	752	657	752	963	536	856	958
本文算法	625	851	331	656	869	402	1132	851

由表5可知,多峰函数对应的 n_{ba} 值一般都比单峰函数高,表明多峰函数相对单峰函数更易陷入局部极值.进一步发现,PSO算法收敛速度最快,在400次迭代后均能达到收敛状态.由于FOA、IFFO及BWFOA算法在种群收敛时不执行全局搜索,其收敛速度较BCFOA、DE等算法稍快.虽然本文算法对应的 n_{ba} 值普遍高于FOA、BWFOA和IFFO,但在大多数情况下仍明显低于ACFOA、BCFOA及DE算法,表明本文算法能够较好地兼顾收敛精度和收敛速度.

4.3 PID仿真实验

4.3.1 常见PID控制系统参数优化

常规PID控制系统原理如图1所示,本文使用ITAE定义果蝇算法的适应度函数^[20],即

$$F(x) = \int_0^t t \times |e(t)|. \quad (17)$$

其中: t 为系统时间, $e(t)$ 为时间 t 时刻的系统误差.进一步地,给出受控对象的数学模型

$$G(s) = \frac{523\ 407}{s^3 + 86.65s^2 + 10\ 465s}. \quad (18)$$

其中:采样时间为1ms,输入信号为阶跃信号.

在搜索过程中,每只果蝇皆表示为常规PID系统中的3维向量 $x = (K_p, K_i, K_d)$.同时,设定搜索边界为 $x_{\min} = 0, x_{\max} = 5$,最大迭代次数 $T = 500$.针对每种算法进行100次实验,求得各算法对应的平均适应值 F_a ,结果如图5所示.

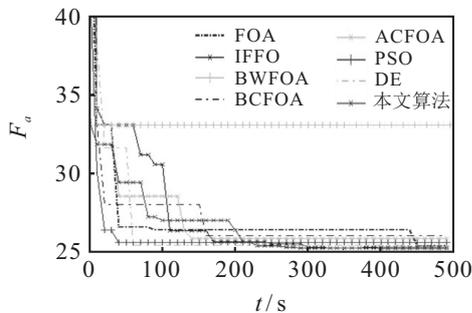


图5 不同算法的收敛过程

由图5可知,BWFOA的性能最差,该算法在15次迭代后即出现了过早收敛的情况.FOA、BCFOA、ACFOA及PSO在优化精度方面较BWFOA表现得好,对应的 F_a 值偏低,但是均出现了局部极值的情况.IFFO、DE和本文算法收敛精度近似,虽然DE收敛速度较快,但对应的 F_a 值(25.35)仍稍高于本文算法(25.16).

表6给出了不同算法对应的最优适应值(v_{aof})和最优位置向量(x_b).可见,本文算法获得的最优位置为 $x_b = (0.863\ 5, 0.000\ 2, 0.009\ 2)$,且 v_{aof} 值普遍低于其他算法,表明本文算法可以有效避免局部最优问题.此外,BWFOA对应的 v_{aof} 值为32.99,明显高于本

文算法.原因在于BWFOA算法在更新果蝇位置时仅利用当前最优位置和最差位置信息,忽略了最优群体和最差群体的整体分布影响.本文提出的多驱逐剂和多引诱剂的双重驱动策略有效综合了两类种群对果蝇位置的排斥力和牵引力,并通过两类种群的带权质心向量之间距离动态调整果蝇的搜索范围,在保证收敛速度的同时,有效降低算法陷入局部最优的可能性.

表6 不同算法平均最优适应值 v_{aof} 和最优位置向量 x_b

对比算法	v_{aof}	x_b
FOA	25.34	(0.855 3, 0, 0.004 2)
IFFO	25.23	(0.868 1, 0, 0.009 1)
BWFOA	32.99	(0.565 4, 0, 0)
BCFOA	25.97	(0.858 3, 0, 0.005 6)
ACFOA	25.73	(0.845 8, 0, 0.008 1)
PSO	25.42	(0.895 6, 0.003 6, 0.009 2)
DE	25.35	(0.867 8, 0, 0.008 6)
本文算法	25.16	(0.863 5, 0.000 2, 0.009 2)

系统上升时间和超调量是自动控制系统响应分析中的两个重要指标.上升时间是指输出信号从最终值的10%上升到90%所需的时间.当系统输出超过稳态值时,超调量是最大输出值超出稳态值的差值与稳态值的百分比.可见,上升时间越短,系统响应越快;超调量越小,系统越稳定.假设 $\text{rin}(t)$ 是单位阶跃信号, $\text{yout}(t)$ 是单位阶跃响应信号,图6给出了不同优化算法响应信号 $\text{yout}(t)$ 值的变化.表7给出了相应的系统上升时间(v_{srt} ,单位s)和超调量(v_{os}).

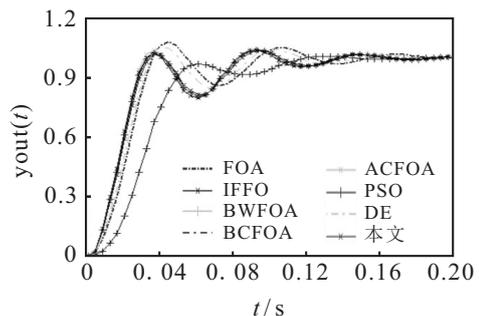


图6 不同算法下响应信号的变化

表7 不同算法的上升时间值 v_{srt} 和超调值 v_{os}

对比算法	v_{srt}	$v_{os}/\%$
FOA	0.015	7.76
IFFO	0.011	4.44
BWFOA	0.042	0.57
BCFOA	0.014	5.33
ACFOA	0.011	3.47
PSO	0.011	3.79
DE	0.009	3.76
本文算法	0.009	3.72

由表7可知: BWFOA取得的 v_{srt} 值最大; 本文算法和DE算法取得最小 v_{srt} 值0.009, 表明这两种算法的优化结果使得被控系统在响应速度上较其他算法明显偏快. 此外, 联合图6和表7可知, PSO、DE及本文算法对应的 v_{os} 值相近, 均低于BWFOA和ACFOA以外的其他方法, 表明这3种算法虽然在前期优化中出现一定的震荡, 但相对于FOA、IFFO以及BCFOA算法而言仍具有较高的稳定性.

4.3.2 串级PID控制系统参数优化

典型的串级PID控制器将内外两环常规控制器并联调节^[22]. 假定受控对象数学模型仍如式(18)所示, 根据串级PID原理给出外环算式如下:

$$u_1(t) = K_{p1} \times e_1(t) + K_{i1} \times \int_0^t e_1(t)dt + K_{d1} \times \frac{d(e_1(t))}{dt}. \quad (19)$$

其中: u_1 为外环调节器输出, $e_1(t)$ 为当前误差, K_{p1} 为外环比例系数, K_{i1} 为外环积分系数, K_{d1} 为外环微分系数.

在此基础上, 给出内环算式如下:

$$u_2(t) = K_{p2} \times e_2(t) + K_{i2} \times \int_0^t e_2(t)dt + K_{d2} \times \frac{d(e_2(t))}{dt}. \quad (20)$$

其中: u_2 为内环调节器输出, $e_2(t)$ 为内环误差, K_{p2} 为内环比例系数, K_{i2} 为内环积分系数, K_{d2} 为内环微分系数. 为验证本文算法的参数优化效果, 使用不同优化算法对上述串级PID系统中6个参数 K_{p1} 、 K_{i1} 、 K_{d1} 、 K_{p2} 、 K_{i2} 及 K_{d2} 进行优先, 对比实验所得结果如表8和表9所示.

表8 串级PID下不同算法平均最优适应值 v_{aof} 和最优位置向量 x_b

对比算法	v_{aof}	x_b
FOA	15.68	(1.526, 1.123, 0.029, 1.538, -0.106, 0.022)
IFFO	14.32	(1.722, -0.052, 0.024, 1.569, -0.206, 0.041)
BWFOA	19.15	(1.925, 1.038, 0.026, 1.368, -0.185, 0.052)
BCFOA	18.46	(1.916, 1.026, 0.031, 1.526, -0.195, 0.067)
ACFOA	18.26	(1.568, 1.322, 0.008, 1.522, -0.196, 0.033)
PSO	14.35	(1.766, 1.057, 0.014, 1.734, -0.178, 0.028)
DE	14.11	(1.865, -0.057, 0.012, 1.734, -0.195, 0.023)
本文算法	12.59	(2.236, -0.025, 0.020, 1.676, -0.211, 0.018)

由表6和表8可知, 串级PID对应的 v_{aof} 值较常规PID明显偏低, 表明串级PID比常规PID能获得较强的准确性和稳定性. 进一步发现, 本文算法对应的 v_{aof} 值(12.59)最低, 验证了该算法在降低串级PID输

出误差方面的有效性. 对比表7和表9可知, 虽然串级PID系统对应的 v_{os} 值与常规PID相近, 但所得 v_{srt} 值明显偏低, 表明前者能以较快速度达到稳定状态. 此外, 本文算法在上升时间值和超调值方面的表现与DE算法近似, 明显优于FOA、PSO等典型算法, 表明使用本文算法进行参数优化后的串级PID仿真系统响应速度快、稳定性好, 验证了其在串级PID参数优化领域中的有效性.

表9 串级PID下不同算法上升时间值 v_{srt} 和超调值 v_{os}

对比算法	v_{srt}	$v_{os}/\%$
FOA	0.012	6.52
IFFO	0.008	4.21
BWFOA	0.011	1.38
BCFOA	0.009	4.62
ACFOA	0.008	4.11
PSO	0.008	3.78
DE	0.005	3.41
本文算法	0.005	3.38

5 结论

本文提出了一种基于多驱避剂与多引诱剂双重驱动的果蝇优化算法, 主要贡献如下: 1) 综合考虑优势果蝇群体和劣势果蝇群体的分布特点, 提出了多驱避剂与多引诱剂的概念, 避免传统方法在果蝇位置更新过程中单纯依靠单个全局最优(差)个体造成的位置偏差; 2) 综合果蝇适应值信息计算多驱避剂和多引诱剂的带权质心向量, 自适应地确定果蝇种群的搜索半径, 避免传统方法的参数依赖性强问题. 针对8个典型测试函数的优化结果表明, 所提出算法具有良好的全局搜索能力、较高的收敛精度和收敛速度. 针对常规PID和典型串级PID控制系统的仿真结果表明, 与其他算法相比, 使用所提出算法优化后的PID控制器具有误差小、响应速度快、稳定性高等优点, 充分显示了所提出算法在PID参数优化中的有效性.

参考文献(References)

- [1] Kaliannan J, Baskaran A, Dey N. Automatic generation control of thermal-thermal-hydro power systems with PID controller using ant colony optimization[J]. Korean Journal of Internal Medicine, 2015, 6(2): 18-34.
- [2] Garpinger O, Hägglund T, Aström K J. Performance and robustness trade-offs in PID control[J]. Journal of Process Control, 2014, 24(5): 568-577.
- [3] 段海滨, 王道波, 黄向华, 等. 基于蚁群算法的PID参数优化[J]. 武汉大学学报, 工学版, 2004, 37(5): 97-100. (Duan H B, Wang D B, Huang X H, et al. Research and realization on parameters optimization of PID controller based on ant colony algorithm[J]. Journal of Wuhan

- University: Engineering Edition, 2004, 37(5): 97-100.)
- [4] Seyedali M, Pradeep J, Shahrzad S. Multi-objective ant lion optimizer: A multi-objective optimization algorithm for solving engineering problems[J]. Applied Intelligence, 2017, 46(1): 79-95.
- [5] Hu R, Wen S P, Zeng Z G, et al. A short-term power load forecasting model based on the generalized regression neural network with decreasing step fruit fly optimization algorithm[J]. Neurocomputing, 2017, 221: 24-31.
- [6] Shen L M, Chen H L, Yu Z, et al. Evolving support vector machines using fruit fly optimization for medical data classification[J]. Knowledge-Based Systems, 2016, 96: 61-75.
- [7] 徐晓飞, 刘志中, 王忠杰, 等. S-ABC-面向服务领域的人工蜂群算法范型[J]. 计算机学报, 2015, 38(11): 2301-2317.
(Xu X F, Liu Z Z, Wang Z J, et al. S-ABC-service domain-oriented artificial bee colony algorithm paradigm[J]. Chinese Journal of Computers, 2015, 38(11): 2301-2317.)
- [8] 郑坤明, 张秋菊. 基于弹性动力学模型与遗传算法的Delta机器人模糊PID控制[J]. 计算机集成制造系统, 2016, 22(7): 1707-1716.
(Zheng K M, Zhang Q J. Fuzzy PID control of delta robot based on elastic dynamic model and genetic algorithm[J]. Computer Integrated Manufacturing Systems, 2016, 22(7): 1707-1716.)
- [9] Li Y, Hou J L, Yuan J. Experiment and vibration suppression algorithm for high-branch pruning manipulator based on fuzzy PID with improved PSO[J]. Transactions of the Chinese Society of Agricultural Engineering, 2017, 33(10): 49-58.
- [10] Storn R. System design by constraint adaptation and differential evolution[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(1): 22-34.
- [11] 潘文超. 应用果蝇优化算法优化广义回归神经网络进行企业经营绩效评估[J]. 太原理工大学学报: 社会科学版, 2011, 29(4): 1-5.
(Pan W C. Using fruit fly optimization algorithm optimized general regression neural network to construct the operating performance of enterprises model[J]. Journal of Taiyuan University of Technology: Social Sciences Edition, 2011, 29(4): 1-5.)
- [12] Pan W T. A new fruit fly optimization algorithm: Taking the financial distress model as an example[J]. Knowledge-Based Systems, 2012, 26: 69-74.
- [13] 韩俊英, 刘成忠. 自适应混沌果蝇优化算法[J]. 计算机应用, 2013, 33(5): 1313-1316.
(Han J Y, Liu C Z. Adaptive chaos fruit fly optimization algorithm[J]. Journal of Computer Applications, 2013, 33(5): 1313-1316.)
- [14] 韩俊英, 刘成忠. 反向认知的高效果蝇优化算法[J]. 计算机工程, 2013, 39(11): 223-225.
(Han J Y, Liu C Z. Efficient fruit fly optimization algorithm with reverse cognition[J]. Computer Engineering, 2013, 39(11): 223-225.)
- [15] 韩俊英, 刘成忠. 基于细菌趋化的果蝇优化算法[J]. 计算机应用, 2013, 33(4): 964-966.
(Han J Y, Liu C Z. Fruit fly optimization algorithm based on bacterial chemotaxis [J]. Journal of Computer Applications, 2013, 33(4): 964-966.)
- [16] Pan Q K, Sang H Y, Duan J H, et al. An improved fruit fly optimization algorithm for continuous function optimization problems[J]. Knowledge-Based Systems, 2014, 62: 69-83.
- [17] 张清勇, 钱浩, 雷德明. 求解多维背包问题的二级协作果蝇优化算法[J]. 控制与决策, 2019, 34(3): 503-510.
(Zhang Q Y, Qian H, Lei D M. Two-level cooperative fruit fly optimization algorithm for multidimensional knapsack problem[J]. Control and Decision, 2019, 34(3): 503-510.)
- [18] 韩俊英, 刘成忠. 基于历史认知的果蝇优化算法[J]. 计算机科学与探索, 2014, 8(3): 368-375.
(Han J Y, Liu C Z. Fruit fly optimization algorithm based on history cognition[J]. Journal of Frontiers of Computer Science and Technology, 2014, 8(3): 368-375.)
- [19] 宋娟. 一种用于PID控制参数优化的混合果蝇算法[J]. 传感器与微系统, 2015, 34(6): 137-140.
(Song J. A hybrid fly fruit algorithm for PID control parameters optimization[J]. Transducer and Microsystem Technologies, 2015, 34(6): 137-140.)
- [20] Niu J, Zhong W, Liang Y, et al. Fruit fly optimization algorithm based on differential evolution and its application on gasification process operation optimization[J]. Knowledge-Based Systems, 2015, 88: 253-263.
- [21] Wu L H, Zuo C L, Zhang H Q. A cloud model based fruit fly optimization algorithm[J]. Knowledge-Based Systems, 2015, 89(C): 603-617.
- [22] 林巍, 王亚刚. 串级控制系统闭环辨识及PID参数整定[J]. 控制工程, 2018, 25(1): 11-18.
(Lin W, Wang Y G. Modeling for cascade control systems based on frequency domain and PID parameter tuning[J]. Control Engineering of China, 2018, 25(1): 11-18.)

作者简介

凤丽洲(1986—), 女, 副教授, 博士, 从事数据挖掘等研究, E-mail: flzvg@126.com;

王友卫(1987—), 男, 副教授, 博士, 从事机器学习、数据挖掘等研究, E-mail: wyw4966198@126.com;

韩琳琳(1997—), 女, 硕士生, 从事数据挖掘的研究, E-mail: 13102175602@163.com;

杨贵军(1970—), 男, 教授, 博士, 从事应用统计、数据挖掘、机器学习等研究, E-mail: yangguijun2008@126.com.

(责任编辑: 郑晓蕾)