

控制与决策

Control and Decision

一种改进 ε 约束飞蛾火焰优化算法及其在约束优化问题中的应用

叶文静, 曹萃文, 顾幸生

引用本文:

叶文静, 曹萃文, 顾幸生. 一种改进 ε 约束飞蛾火焰优化算法及其在约束优化问题中的应用[J]. 控制与决策, 2023, 38(10): 2841–2849.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2021.2067>

您可能感兴趣的其他文章

Articles you may be interested in

[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>

[一种求解约束多目标问题的协作进化算法](#)

A collaborative evolutionary algorithm for solving constrained multi-objective problems
控制与决策. 2021, 36(11): 2656–2664 <https://doi.org/10.13195/j.kzyjc.2020.0791>

[具有重组学习和混合变异的动态多种群粒子群优化算法](#)

Dynamic multi-population particle swarm optimization algorithm with recombined learning and hybrid mutation
控制与决策. 2021, 36(12): 2871–2880 <https://doi.org/10.13195/j.kzyjc.2020.0898>

[基于双种群模糊引力搜索算法的舰载机甲板作业调度](#)

Flight deck operations scheduling based on dual population fuzzy gravitational search algorithm
控制与决策. 2021, 36(11): 2751–2759 <https://doi.org/10.13195/j.kzyjc.2020.0523>

[带峰值能耗约束流水线调度的协同群智能优化](#)

Cooperative memetic optimization for flowshop scheduling with peak power consumption constraint
控制与决策. 2021, 36(10): 2350–2358 <https://doi.org/10.13195/j.kzyjc.2020.0429>

一种改进 ε 约束飞蛾火焰优化算法及其在 约束优化问题中的应用

叶文静, 曹萃文, 顾幸生[†]

(华东理工大学 能源化工过程智能制造教育部重点实验室, 上海 200237)

摘要: 提出一种改进 ε 约束法的飞蛾火焰优化算法(ε improved moth-flame optimization algorithm, ε IMFO)求解约束优化问题. 该算法采用 ε 约束法对约束进行处理, 考虑种群整体的约束违反度变化, 提出一种基于火焰种群约束违反度的阈值 ε 计算公式; 改进火焰种群的更新方法, 首先, 根据种群中个体的约束违反度与 ε 的关系将其分为两类: 一类是约束违反度小于等于 ε 的个体, 按照目标函数值排序, 另一类是约束违反度大于 ε 的个体, 按约束违反度排序; 然后, 先选择第1类中的个体, 若数量没有达到种群数量的要求, 继续从第2类中选取个体形成新一代火焰种群; 最后, 提出一种改进的飞蛾变异策略, 在原始飞蛾变异策略的基础上引入2个随机火焰个体影响飞蛾变异, 并增加优秀火焰个体对飞蛾变异的指导作用. 通过25个测试函数以及2个实际的工程优化问题分别与其他13种算法进行的算法性能测试对比表明, ε IMFO算法在求解精度和稳定性等方面具有优势.

关键词: 约束优化问题; 飞蛾火焰优化算法; 改进 ε 约束法; 火焰更新; 飞蛾变异策略

中图分类号: TP301.6

文献标志码: A

DOI: 10.13195/j.kzyjc.2021.2067

引用格式: 叶文静, 曹萃文, 顾幸生. 一种改进 ε 约束飞蛾火焰优化算法及其在约束优化问题中的应用[J]. 控制与决策, 2023, 38(10): 2841-2849.

An ε improved moth-flame optimization algorithm for solving constrained optimization problems and engineering applications

YE Wen-jing, CAO Cui-wen, GU Xing-sheng[†]

(Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China)

Abstract: An ε improved moth-flame optimization algorithm (ε IMFO) is proposed to solve constrained optimization problems in this paper. Considering the change of the constraint violation degree of the population, this algorithm uses improved ε constraint method to deal with the constraints, and develops a threshold ε formula based on the constraint violation degree of the flame population. The update method of the flame population is improved. According to the relationship between the constraint violation degree of individuals in the population and ε , they are divided into two categories: one is the individuals whose constraint violation degree is less than or equal to ε , sorted according to the objective function value; the other is the individuals whose constraint violation degree is bigger than ε , sorted by constraint violation degree. Then the individuals in the first category are firstly selected. If the number does not meet the population size, individuals are selected from the second category to build a new flame population. Finally, an improved moth mutation strategy is proposed. Based on the original moth mutation strategy, two random flame individuals are introduced to affect moth mutation, and the guiding effect of the excellent flame individuals on moth mutation is increased. The comparison of algorithm performance tests with other 13 algorithms among 25 benchmark test functions and 2 actual engineering optimization problems shows that the ε IMFO algorithm is superior in terms of the accuracy and stability.

Keywords: constrained optimization problem; moth-flame optimization algorithm; ε improved constraint method; flame update; moth mutation strategy

收稿日期: 2021-11-26; 录用日期: 2022-05-31.

基金项目: 国家自然科学基金项目(61673175, 61973120).

责任编辑: 张国山.

[†]通讯作者. E-mail: xsgu@ecust.edu.cn.

*本文附带电子附录文件, 可登录本刊官网该文“资源附件”区自行下载阅览.

0 引言

现实世界中的许多优化问题均会受到约束条件的限制,这类问题被称作约束优化问题,以最小化问题为例,约束优化问题可描述为

$$\begin{aligned} \min & f(X). \\ \text{s.t.} & g_i(X) \leq 0, \quad i = 1, 2, \dots, \text{ieq}; \\ & h_j(X) = 0, \quad j = 1, 2, \dots, \text{eq}; \\ & l_k \leq x_k \leq u_k, \quad k = 1, 2, \dots, d. \end{aligned} \quad (1)$$

其中: $f(X)$ 为目标函数, $g_i(X)$ 为不等式约束条件, $h_j(X)$ 为等式约束条件, X 为决策向量, d 为决策变量的个数, ieq 为不等式约束的个数, eq 为等式约束的个数, l_k 和 u_k 分别为第 k 个决策变量 x_k 的下界和上界. 当问题是最大化目标的形式时,可将其转换为式(1)所示的形式后进行求解.

约束优化问题在科学研究和工程运用中具有重要研究价值,但是由于约束优化问题存在可行域难以确定、约束条件复杂以及搜索方向不确定等问题,其求解是优化领域的一大难题. 目前求解约束优化问题主要有两类方法,一类是包括内点罚函数法、单纯形法和 Lagrange 法等的确性算法,另一类是包括遗传算法、粒子群优化算法、差分进化算法等的智能进化算法^[1]. 近年来,针对智能进化算法中约束条件的处理,人们提出了各种处理机制, Cantú 等^[2]归纳了比较有效的约束处理机制,包括罚函数法^[3]、随机排序法^[4]、可行性规则^[5]、 ε 约束法^[6]和基于梯度的修复方法^[7]等. Huang 等^[8]提出了一种基于协同进化机制的差分进化算法,该算法将协同进化模型融入差分进化算法,采用双种群并行策略:一个种群对惩罚因子进行自适应调整,另一个种群根据适应度值进行迭代操作,从而获得满意的可行解. Wang 等^[9]提出了自适应权衡模型进化策略求解约束优化问题. 根据种群中可行解的存在情况,该算法采用不同的方案对种群进行进化并得到问题的解. Takahama 等^[10]提出了 ε DE 算法,采用 ε 约束法对约束条件进行处理,通过引入基于梯度的变异策略引导不可行解转化为可行解,并采用精英保留策略保留种群中的优秀个体. Gao 等^[11]提出了 DPDE (dual-population differential evolution, DPDE) 算法,在 DPDE 算法中,约束优化问题被视为具有目标函数值和约束违反度 2 个目标的双目标问题. 在迭代过程中,DPDE 首先判断种群中可行解的数量,当可行解数量在特定区间时,再根据个体是否为可行解将种群分为两类子种群,并分别优化目标函数值和约束违反度;2 个子种

群间使用信息共享策略交换搜索信息,从而求解约束优化问题.

飞蛾火焰优化算法 (moth-flame optimization algorithm, MFO) 是 Mirjalili^[12] 于 2015 年受飞蛾夜间飞行行为启发所提出的一种算法. MFO 算法具有参数少,适用范围广等优点. 为进一步提高 MFO 算法性能,大量学者对其进行了研究和改进. 其中, Emary 等^[13] 通过引入混沌理论,利用混沌函数对 MFO 算法的距离系数进行调整,从而更好地平衡算法的全局与局部搜索. Wang 等^[14] 提出了 CMFO (chaotic moth-flame optimization, CMFO) 算法,该算法利用 Logistic 混沌映射产生的随机序列作为初始种群,在进化过程中对最优个体使用混沌扰动进而避免算法陷入局部最优解. Pelusi 等^[15] 提出一种具有混合搜索阶段的 MFO 算法,将算法的搜索过程分为探索、混合和开发 3 个阶段,在所有阶段,均采用动态交叉机制改进火焰. 在探索阶段,飞蛾采用原始变异方式进行变异;在混合阶段,引入最佳飞蛾影响飞蛾变异,在开发阶段,设计一个随机的精英火焰集更新飞蛾位置. 该算法改善了原始算法全局搜索能力弱,收敛速度慢等不足.

本文基于前人在对智能算法中的约束条件进行处理的有效成果和 MFO 算法的基础上,提出一种改进 ε 约束飞蛾火焰优化算法 (ε IMFO) 求解约束优化问题. ε IMFO 算法在约束条件的处理上采用 ε 约束法,考虑种群的约束违反度变化情况,提出一种基于火焰种群约束违反度的 ε 确定方法;结合 ε 约束法的规则,改进火焰更新方法. ε IMFO 算法还进一步改进飞蛾的变异方式,通过在原始的飞蛾变异策略中引入 2 个随机火焰个体影响飞蛾变异并增加优秀火焰个体对飞蛾变异的指导作用. 最后采用 25 个基准测试函数和 2 个实际工程优化问题对 ε IMFO 算法与其他 13 种算法进行性能测试对比.

1 飞蛾火焰优化算法

Mirjalili^[12] 受飞蛾夜间飞行行为的启发,将飞蛾围绕火焰螺旋飞行的行为作为算法的进化机制,提出了飞蛾火焰优化算法. 在 MFO 算法中,飞蛾代表在搜索空间中进行探索的个体,火焰种群用于记录飞蛾种群已探索到的最优位置. 在探索过程中,每只飞蛾均有一个对应的火焰个体作为搜索指引,并不断调整自己的位置向火焰靠近. 火焰数量在探索过程中自适应减少,直至仅剩唯一的火焰个体,即为问题的最优解.

在 MFO 算法中飞蛾种群以 M 表示,飞蛾种群的

适应度值以 OM 表示. 火焰种群以 F 表示, 火焰种群的适应度值以 OF 表示.

MFO 算法主要包括初始化、飞蛾变异和火焰更新 3 个步骤.

在初始化阶段中, 初始飞蛾种群在搜索空间中随机产生, 随后根据种群中个体适应度值对飞蛾种群进行排序并赋值给火焰种群, 从而得到初始火焰种群.

在飞蛾变异阶段中, 每只飞蛾根据一个对应的火焰个体按照螺旋飞行轨迹进行变异, 飞蛾的变异方式如下式所示:

$$M_i = F_j + e^{bt} \cdot \cos(2\pi t) \cdot D_i, \quad (2)$$

$$j = \begin{cases} i, & i \leq \text{flame no;} \\ \text{flame no}, & \text{otherwise.} \end{cases} \quad (3)$$

其中: $D_i = |F_j - M_i|$ 为第 j 个火焰与第 i 飞蛾间的距离; b 为对数螺旋函数中的常数; t 为距离系数, 其更新方法如下式所示; **flame no** 为迭代过程中火焰种群个体数量, 按下式进行变化:

$$t = \text{rand} \left(-\frac{k}{T_{\max}} - 1, 1 \right). \quad (4)$$

其中: k 为当前的迭代次数, T_{\max} 为算法最大迭代次数.

$$\text{flame no} = \text{round} \left(\text{NP} - k \cdot \frac{\text{NP} - 1}{T_{\max}} \right). \quad (5)$$

在火焰更新阶段中, 通过对飞蛾种群和火焰种群中的个体按照适应度值进行升序排序并选取前 **flame no** 个个体作为新一代的火焰种群.

MFO 算法的具体步骤如下.

step 1: 初始化参数: 最大迭代次数 T_{\max} , 飞蛾种群个体数量 NP, 火焰种群初始个体数量 NP, 决策变量个数 d ;

step 2: 对飞蛾种群 M 和火焰种群 F 进行初始化操作;

step 3: 根据式(5)计算火焰个体的数量 **flame no**;

step 4: 根据式(4)更新位置参数 t ;

step 5: 根据式(2)和(3)更新飞蛾位置;

step 6: 计算飞蛾种群的适应度值;

step 7: 火焰种群排序更新;

step 8: 若算法达到最大迭代次数, 则寻优过程结束并输出寻优结果, 否则返回至 step 3 继续执行下一代迭代操作.

2 改进飞蛾火焰优化算法

为使得 MFO 算法可用于求解约束优化问题, 本文对 MFO 算法进行改进并提出 ε IMFO 算法. 该算法改进了 ε 约束法、火焰更新方法并采用一种改进的飞

蛾变异策略.

2.1 基于火焰种群整体约束违反度的 ε 约束法

在约束处理阶段, ε 约束法^[6] 被广泛使用. 当约束优化问题如式(1)所示时, ε 约束选择规则如下.

1) 若 2 个个体的约束违反度小于阈值 ε 时, 选择目标函数小的个体;

2) 若 2 个个体的约束违反度相等, 则选择目标函数小的个体;

3) 否则, 选择约束违反度更小的个体.

约束违反度按照下式计算:

$$v(X) = \sum_{i=1}^{\text{ieq}} \max\{g_i(X), 0\}^p + \sum_{j=1}^{\text{eq}} |h_j(X)|^p. \quad (6)$$

其中: $v(X)$ 为约束违反度; p 为系数, 在本文中 p 取 1.

ε 的确定规则^[10] 如下式所示:

$$\varepsilon(k) = \begin{cases} v(X_\theta), & k = 0; \\ \varepsilon(0) \left(1 - \frac{k}{T_c}\right)^{\text{cp}}, & 0 < k < T_c; \\ 0, & k \geq T_c. \end{cases} \quad (7)$$

其中: $\varepsilon(k)$ 为第 k 次迭代时的 ε 值; $v(X_\theta)$ 为初始种群按照个体约束违反度进行升序排序后第 θ ($\theta = 0.05 \times \text{NP}$) 个个体的约束违反度; $T_c \in [0.1 \times T_{\max}, 0.8 \times T_{\max}]$, $\text{cp} \in [2, 10]$.

上述 ε 约束法在进化过程中没有考虑种群整体约束违反度的变化, 因此本文提出一种基于火焰种群整体约束违反度的 ε 约束法, ε 的进化方式如下式所示:

$$\varepsilon(k) = \begin{cases} \frac{\sum_{i=1}^{\text{NP}} v(M_i)}{\text{NP} + 1}, & k = 0; \\ \varepsilon_1(k), & 0 < k < T_{\max}/3; \\ \varepsilon_2(k), & T_{\max}/3 \leq k \leq 2 \times T_{\max}/3; \\ 0, & k > 2 \times T_{\max}/3. \end{cases} \quad (8)$$

$$\varepsilon_1(k) = \begin{cases} +\infty, & \text{num}_{\text{obj}} \geq \text{best}_{\text{obj}} > \text{NP} \times \alpha; \\ \varepsilon_2(k), & \text{otherwise.} \end{cases} \quad (9)$$

$$\varepsilon_2(k) = \begin{cases} \frac{\sum_{j=1}^{\text{NP}} v(F_j)}{\text{NP} + 1}, & \frac{\sum_{j=1}^{\text{NP}} v(F_j)}{\text{NP} + 1} < \varepsilon(k-1); \\ \varepsilon(k-1), & \text{otherwise.} \end{cases} \quad (10)$$

其中: $v(M_i)$ 为第 i 个飞蛾的约束违反度; $v(F_j)$ 为第 j 个火焰的约束违反度; best_{obj} 为算法当前最优解的目标函数值; $\text{num}_{\text{obj}} \geq \text{best}_{\text{obj}}$ 为火焰种群中目标函数值大于 best_{obj} 的个体数量, $\alpha \in [0.1, 1]$. 在前期迭代进化

过程中,当 $\text{num}_{\text{obj}} \geq \text{best}_{\text{obj}} > \text{NP} \times \alpha$ 时,放宽约束条件的限制,火焰种群可吸收一些目标函数值更加优异的个体.

2.2 改进火焰更新方法

在原始MFO算法中,火焰更新方法中仅需考虑适应度值.但是在使用 ε 约束规则处理约束优化问题时,需同时考虑目标函数和约束违反度,因此原始火焰更新方法不再适用.对此,本文结合 ε 约束法的规则特性,提出一种改进火焰更新方法.改进火焰更新方法如下:将待排序种群(在第1次迭代时,待排序种群为飞蛾种群,否则,待排序种群为飞蛾种群和火焰种群)的个体按照约束违反与阈值 ε 的关系将其分为子种群1和子种群2,子种群1中的个体约束违反度小于等于阈值 ε ,按照目标函数值升序排序;子种群2中的个体约束违反度大于阈值 ε ,按照约束违反度升序排序.优先选择子种群1中的个体,若数量小于火焰种群要求的个体数量NP,则继续从子种群2中选取相应数量的个体,从而形成新一代的火焰种群.

2.3 改进飞蛾变异策略

原始飞蛾变异策略仅考虑单一火焰个体对飞蛾变异的影响,且优秀火焰对飞蛾的变异指导作用少,对此,本文在原始飞蛾变异策略的基础上,筛选出前 $\beta \times \text{NP}$ 个最优火焰对飞蛾的变异进行指导,并加入2个随机火焰个体影响飞蛾变异,提出一种改进飞蛾变异策略,如下式所示:

$$M_i = F_h + e^{bt_1} \cdot \cos(2\pi t_1) \cdot (F_i - M_i) + e^{bt_2} \cdot \cos(2\pi t_2) \cdot (F_{r_1} - F_{r_2}). \quad (11)$$

其中: t_1 和 t_2 计算公式与式(4)一致, r_1 和 r_2 为 $0 \sim \text{NP}$ 间的随机数, h 的确定方式如下式所示:

$$h = \begin{cases} i, & i < \text{round}(\beta \times \text{NP}); \\ i \% \text{round}(\beta \times \text{NP}), & \text{otherwise.} \end{cases} \quad (12)$$

在该变异策略中,火焰个体数量不再自适应减小,而是保持不变.飞蛾围绕前 $\beta \times \text{NP}$ 个的优秀火焰个体进行探索,增加优秀火焰个体在飞蛾变异过程中的指导作用,提高算法的求解精度和收敛速度;通过加入2个随机火焰个体影响飞蛾变异,增加飞蛾种群的多样性.

ε IMFO算法的具体步骤如下.

step 1: 初始化参数:最大迭代次数 T_{max} ,飞蛾种群个体数量和火焰种群个体数量NP,决策变量个数 d ;

step 2: 对飞蛾种群 M 进行初始化操作;

step 3: 计算飞蛾种群 M 中个体的目标函数值和约束违反度;

step 4: 根据式(8)~(10)计算阈值 ε ;

step 5: 火焰种群 F 按照改进后的火焰更新方式进行更新;

step 6: 飞蛾种群 M 根据式(11)和(12)进行变异操作;

step 7: 若算法达到最大迭代次数,则寻优过程结束并输出寻优结果,否则返回至step 3继续执行下一代迭代操作.

3 实验结果和分析

为检验 ε IMFO的性能,本文通过CEC 2006中的17个测试函数^[16]以及文献[17]中的8个测试函数对算法进行测试. ε IMFO算法在Visual Studio 2017软件上进行编程,计算机配置为Intel(R) Core(TM) i7-8550U CPU@1.80 GHz 1.99 GHz,16 GB 内存,Windows 10操作系统.

3.1 算法参数分析

为了测试 α 和 β 参数对 ε IMFO性能的影响,将 α 参数分别设置为0.1、0.2、0.3、0.4、0.5、0.6、0.7、0.8、0.9以及1.0; β 参数分别设置为0.1、0.15、0.2、0.25、0.3、0.35、0.4、0.45以及0.5.对25个测试函数在不同参数设置下独立运行25次,各算例的 α 和 β 的优秀取值区间如表1所示.因此,根据表1结果综合考虑,本文 α 取0.5, β 取0.15进行 ε IMFO算法性能测试.

表1 各测试函数值的优秀参数取值区间

problem	α	β	problem	α	β
g_1	0.4~0.8	0.15~0.50	g_{14}	0.1~1.0	0.05~0.40
g_2	0.3~0.5	0.15~0.50	g_{15}	0.1~1.0	0.05~0.50
g_3	0.2~0.6	0.10~0.40	g_{17}	0.1~1.0	0.05~0.35
g_4	0.1~1.0	0.05~0.50	g_{18}	0.1~1.0	0.05~0.35
g_5	0.1~0.9	0.05~0.45	R_1	0.1~1.0	0.05~0.35
g_6	0.4~0.6	0.10~0.30	R_2	0.1~1.0	0.05~0.45
g_7	0.2~1.0	0.10~0.40	R_3	0.1~1.0	0.05~0.50
g_8	0.1~1.0	0.05~0.50	R_4	0.1~0.6	0.10~0.35
g_9	0.1~1.0	0.05~0.50	R_5	0.1~0.5	0.10~0.15
g_{10}	0.3~0.6	0.10~0.30	R_8	0.1~1.0	0.05~0.50
g_{11}	0.2~0.8	0.10~0.40	R_9	0.1~1.0	0.05~0.50
g_{12}	0.2~1.0	0.10~0.50	R_{10}	0.1~0.6	0.05~0.40
g_{13}	0.1~0.8	0.05~0.20			

3.2 算法性能对比分析

为了检验 ε IMFO求解约束优化问题的性能,实验测试CEC 2006中17个测试函数时, ε IMFO最大迭代次数 T_{max} 为2000,飞蛾和火焰个体数量NP为100,目标函数的最大估计次数为200000, α 为0.5, β 为

0.15,本文对各测试函数独立运行25次并将结果与ODPSO^[18]、ATMES^[9]、ICTLBO^[19]、ETLBO^[20]、AIS-ZYH^[21]、SAMODE^[22]和DECV^[23]算法的结果进行比较,其中ODPSO结果摘自文献[18].ATMES、

ICTLBO、ETLBO、AIS-ZYH、SAMODE和DECV结果摘自文献[19]. ε IMFO与其余7种算法计算所得的结果如表2所示,其中mean和std分别为25次独立运行结果的平均解和标准差.加粗显示的为最优结果.

表2 ε IMFO与不同算法在CEC 2006中17个测试函数中的结果对比

problem	standard	ε IMFO	ODPSO	ATMES	ICTLBO	ETLBO	AIS-ZYH	SAMODE	DECV
g_1	mean	-15	-15	-15	-15	-15	-15	-15	-14.855
	std	0.00e+00	3.63e-16	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	4.59e-01
g_2	mean	-0.746939	-0.740480694	-0.787637	-0.799622	-0.803619	-0.802193	-0.798735	-0.569458
	std	2.76e-02	4.34e-02	1.18e-02	5.17e-03	0.00e+00	5.19e-10	8.80e-03	9.51e-02
g_3	mean	-1.0005001	-0.997140653	-0.9999	-1.0005	-1.0003	-1.0005	-1.0005	-0.134
	std	0.00e+00	1.29e-02	1.02e-04	1.97e-13	1.40e-04	1.77e-11	0.00e+00	1.17e-01
g_4	mean	-30665.53867	-30665.53867	-30665.539	-30665.539	-30665.539	-30665.539	-30665.5386	-30665.539
	std	0.00e+00	9.89e-06	7.43e-12	7.43e-12	0.00e+00	3.69e-13	0.00e+00	1.56e-06
g_5	mean	5126.496714	5181.393285	5127.7321	5126.4967	5168.7194	5126.4981	5126.497	5126.497
	std	0.00e+00	7.47e+01	2.15e+00	1.86e-12	5.41e+01	1.70e-02	0.00e+00	0.00e+00
g_6	mean	-6961.813876	-6961.813876	-6961.8139	-6961.8139	-6961.814	-6961.8139	-6961.814	-6961.814
	std	0.00e+00	0.00e+00	3.71e-12	3.71e-12	0.00e+00	1.90e-12	0.00e+00	0.00e+00
g_7	mean	24.306209	24.33487941	24.31456	24.3062	24.31	24.35572	24.3096	24.794
	std	1.92e-08	5.86e-02	1.42e-02	5.40e-14	7.11e-03	8.20e-03	1.59e-03	1.37e+00
g_8	mean	-0.09582504	-0.095825041	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825
	std	0.00e+00	5.67e-18	6.12e-17	0.00e+00	0.00e+00	0.00e+00	0.00e+00	4.23e-17
g_9	mean	680.6300574	680.6300574	680.6	680.63	680.63	680.650308	680.63	680.63
	std	0.00e+00	9.27e-13	1.26e-02	4.64e-13	0.00e+00	1.20e-08	1.16e-05	3.45e-07
g_{10}	mean	7049.248021	7053.512497	7277.47	7049.31286	7143.45	7049.57032	7059.81345	7103.548
	std	0.00e+00	1.19e+01	1.97e+02	8.39e-02	1.13e+02	4.50e-04	7.86e+00	1.48e+02
g_{11}	mean	0.7499	0.7499	0.75	0.7499	0.74998	0.749999	0.7499	0.7499
	std	0.00e+00	1.13e-16	2.82e-04	1.13e-16	7.06e-05	1.40e-08	0.00e+00	1.12e-16
g_{12}	mean	-1	-1	-1	-1	-1	-1	-1	-1
	std	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
g_{13}	mean	0.05394151	0.340434355	0.053959	0.20788595	0.83851	0.05394	0.053942	0.382401
	std	0.00e+00	2.36e-01	1.06e-05	1.92e-01	2.26e-01	7.80e-10	1.75e-08	2.68e-01
g_{14}	mean	-47.76488846	-47.7578018	-47.7278	-47.7649	-43.805	-47.7649	-47.68115	-47.72254
	std	0.00e+00	1.25e-02	5.05e-02	2.10e-08	2.32e+00	1.00e-12	4.04e-02	1.62e-01
g_{15}	mean	961.7150223	961.7464745	961.7153	961.71502	962.044	961.71502	961.71502	961.71502
	std	0.00e+00	6.41e-02	2.69e-04	4.64e-13	4.39e-01	0.00e+00	0.00e+00	2.31e-13
g_{17}	mean	8853.533875	8923.934753	8896.4008	8880.59525	8895.7544	8853.5397	8853.5397	8919.93636
	std	0.00e+00	7.26e+01	3.27e+01	3.69e+01	5.14e+01	1.90e-09	1.15e-05	2.59e+01
g_{18}	mean	-0.8660254	-0.827792699	-0.843026	-0.8660254	-0.865755	-0.866025	-0.866024	-0.859657
	std	0.00e+00	7.80e-02	6.35e-02	1.48e-13	5.09e-04	1.30e-15	7.04e-07	3.48e-02

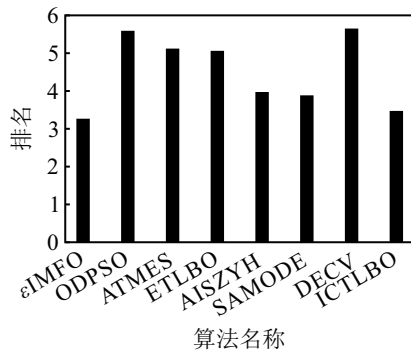


图1 对比算法在17个测试函数上的Friedman检验排名

由表2可见: ϵ IMFO在16个函数(g_1 、 g_3 、 g_4 、 g_5 、 g_6 、 g_7 、 g_8 、 g_9 、 g_{10} 、 g_{11} 、 g_{12} 、 g_{13} 、 g_{14} 、 g_{15} 、 g_{17} 、 g_{18})中获得了最优解,在15个函数(g_1 、 g_3 、 g_4 、 g_5 、 g_6 、 g_7 、 g_9 、 g_{10} 、 g_{11} 、 g_{12} 、 g_{13} 、 g_{14} 、 g_{15} 、 g_{17} 、 g_{18})中标准差为0.表明该算法具有良好的求解精度和稳定性.根据表2中各算法的平均值,采用Friedman检验统计方法对各不同算法的性能进行排序,结果如图1所示.由图1可见, ϵ IMFO在8种算法中排名第1,ICTLBO排名第2.与ODPSO和DECV相比, ϵ IMFO有非常明显的性能改进.

表3 ϵ IMFO与不同算法在8个测试函数中的结果对比

problem	algorithm	best	median	mean	worst	std	FR	SR
RC ₁	ϵ IMFO	1.89e+02	1.89e+02	1.89e+02	1.89e+02	2.90e-14	100	100
	IUDE	1.89e+02	2.60e+02	2.29e+02	1.85e+02	8.06e+01	24	4
	ϵ MAGES	1.89e+02	4.92e+02	4.55e+02	4.37e+02	2.23e+02	84	20
	iLSHADE ϵ	1.90e+02	1.94e+02	2.06e+02	2.29e+02	1.93e+01	28	4
RC ₂	ϵ IMFO	7.05e+03	7.05e+03	7.05e+03	7.05e+03	3.71e-12	100	100
	IUDE	7.05e+03	7.05e+03	7.15e+03	5.94e+03	7.54e+02	92	92
	ϵ MAGES	7.05e+03	7.80e+03	7.74e+03	7.48e+03	7.50e+02	96	96
	iLSHADE ϵ	7.05e+03	7.05e+03	7.05e+03	7.05e+03	2.13e-11	100	100
RC ₃	ϵ IMFO	-4.53e+03	-4.53e+03	-4.53e+03	-4.53e+03	0.00e+00	100	100
	IUDE	-4.53e+03	-1.43e+02	-6.25e+03	-1.83e+04	6.75e+03	64	12
	ϵ MAGES	-1.43e+02	7.63e+01	-1.60e+02	4.95e+02	8.88e+02	92	0
	iLSHADE ϵ	-4.53e+03	-1.43e+02	-8.18e+02	5.34e+02	1.91e+03	100	20
RC ₄	ϵ IMFO	-3.88e-01	-3.88e-01	-3.88e-01	-3.88e-01	0.00e+00	100	100
	IUDE	-2.86e-01	-5.92e-01	-4.96e-01	-1.00e+00	1.82e-01	0	0
	ϵ MAGES	-3.88e-01	-3.75e-01	-5.50e-01	-1.00e+00	2.86e-01	72	24
	iLSHADE ϵ	-3.75e-01	-3.75e-01	-3.75e-01	-3.73e-01	4.61e-04	100	0
RC ₅	ϵ IMFO	-4.00e+02	-3.65e+02	-2.83e+02	-6.54e-02	1.47e+02	100	12
	IUDE	-4.00e+02	-4.00e+02	-3.51e+02	-8.30e-03	1.32e+02	100	80
	ϵ MAGES	-4.00e+02	-3.98e+02	-3.63e+02	-1.00e+02	7.55e+01	100	36
	iLSHADE ϵ	-4.00e+02	-8.06e-03	-1.17e+02	1.57e+01	1.79e+02	100	44
RC ₈	ϵ IMFO	2.00e+00	2.00e+00	2.00e+00	2.00e+00	0.00e+00	100	100
	IUDE	2.00e+00	2.00e+00	2.00e+00	2.00e+00	6.41e-17	100	100
	ϵ MAGES	2.00e+00	2.00e+00	1.99e+00	1.29e+00	1.52e-01	96	64
	iLSHADE ϵ	2.00e+00	2.00e+00	2.00e+00	2.00e+00	0.00e+00	100	100
RC ₉	ϵ IMFO	2.56e+00	2.56e+00	2.56e+00	2.56e+00	9.06e-16	100	100
	IUDE	2.56e+00	2.56e+00	2.56e+00	2.56e+00	1.36e-15	100	100
	ϵ MAGES	2.56e+00	2.56e+00	2.56e+00	1.93e+00	2.70e-01	92	92
	iLSHADE ϵ	2.56e+00	2.56e+00	2.56e+00	2.56e+00	1.46e-07	100	100
RC ₁₀	ϵ IMFO	1.08e+00	1.08e+00	1.08e+00	1.08e+00	4.53e-16	100	100
	IUDE	1.08e+00	1.08e+00	1.11e+00	1.25e+00	7.08e-02	100	88
	ϵ MAGES	1.08e+00	1.08e+00	1.08e+00	1.25e+00	3.47e-02	100	92
	iLSHADE ϵ	1.08e+00	1.25e+00	1.22e+00	1.25e+00	6.48e-02	100	88

对于文献[17]中 8 个测试函数, ϵ IMFO 的参数设置如下: α 为 0.5, β 为 0.15, 当 $d \leq 10$ 时, 最大目标函数估计次数为 100 000, 最大迭代次数 T_{max} 为 1 000, 飞蛾和火焰数量 NP 为 100; 当 $d > 10$ 时, 最大目标函数估计次数为 200 000, 最大迭代次数 T_{max} 为 2 000, 飞蛾和火焰数量 NP 为 100. 将 ϵ IMFO 算法独立运行 25 次后的计算结果与 IUDE^[24]、 ϵ MAgES^[25] 和 iLSHADE ϵ ^[26] 算法进行比较, IUDE、 ϵ MAgES 和 iLSHADE ϵ 算法的结果摘自文献[17]. ϵ IMFO 算法与其他算法的比较结果如表 3 所示, 其中 best、median、mean、worst 和 std 分别为 25 次独立运行获得解中的最优值、中位数、平均值、最差值以及标准差, FR 和 SR 分别为 25 次运行中最优解为可行解和算法最优解达到已知最优解的比例, 算法最优解应满足 $f(x) - f(x^*) \leq 10^{-8}$.

由表 3 可见, ϵ IMFO 对于 8 个算例的可行率为 100%, 在除 RC₅ 外其余 7 个算例上的成功率为 100%. 与其他算法比较发现, 对于 RC₁、RC₂、RC₃、RC₄ 和 RC₁₀, ϵ IMFO 是唯一一个可行率和成功率皆为 100% 的算法, 优于其他对比算法; 对于 RC₅, ϵ IMFO 算法在可行率上不输于其余对比算法; 在 RC₈ 和 RC₉ 中, ϵ IMFO 的可行率和成功率比 ϵ MAgES 好且不输于 IUDE 与 iLSHADE ϵ . 在这 4 种算法中, ϵ IMFO 表现最佳.

在以上 25 个测试函数上的出色表现验证了 ϵ IMFO 算法在求解约束优化问题中具有更好的求解效果和稳定性.

3.3 算法复杂度分析

算法的计算复杂度是评估其运行时间的关键指标, 可根据算法的结构和实现定义. ϵ IMFO 算法的计算复杂度取决于飞蛾个体数量、最大迭代次数以及每次迭代中火焰的更新操作. 因此, ϵ IMFO 算法的时间复杂度为

$$\begin{aligned}
 O(\epsilon\text{IMFO}) &= \\
 &O(T_{\max} \times (O(\text{Bubble Sort}) + \\
 &O(M \text{ update}) + O(\epsilon \text{ update}))) = \\
 &O(T_{\max} \times (NP^2 + NP \times d + NP)) = \\
 &O(T_{\max} \times NP^2 + T_{\max} \times (d + 1)) \approx \\
 &O(T_{\max} \times NP^2). \tag{13}
 \end{aligned}$$

4 ϵ IMFO 算法在工程优化问题中的应用

problem 1 焊接梁设计^[27]. 该问题是在约束条件下设计焊接梁, 目标为最小化成本, 包含 7 个约束

条件和 4 个变量.

problem 2 拉伸/压缩弹簧设计^[28]. 该问题的目标是优化拉伸或压缩弹簧的重量, 包含 4 个约束条件和 3 个变量.

上述 2 个工程问题, 采用与第 3.1 节中相同的参数确定方法, 得到优秀求解结果情况下的 α 和 β , 优秀取值区间如表 4 所示. 因此, ϵ IMFO 算法参数设置如下: α 为 0.5, β 为 0.15, 最大函数评估次数为 100 000, 最大迭代次数 T_{max} 为 1 000, 飞蛾个体和火焰个体的数量 NP 为 100. ϵ IMFO 算法独立运行 25 次, 并将其结果与 CDE^[29]、IFA^[30] 和 UABC^[31] 的结果进行对比. CDE、IFA、UABC 算法结果摘自文献[32], 其结果如表 5 所示 (NA 为原文没有提供具体数据).

表 4 工程优化问题的优秀参数取值区间

problem	α	β
problem 1	0.1 ~ 1.0	0.05 ~ 0.30
problem 2	0.4 ~ 0.6	0.05 ~ 0.40

表 5 ϵ IMFO 与不同算法在 2 个工程算例的测试结果对比

problem	standard	CDE	IFA	UABC	ϵ IMFO
problem 1	best	1.733 461	1.724 894	1.724 852	1.724 852
	mean	1.768 158	1.724 978	1.724 853	1.724 852
	worst	1.824 105	1.725 321	NA	1.724 852
	std	2.20e-02	1.33e-05	1.70e-06	0.00e+00
problem 2	best	0.012 670 2	0.012 665 8	0.012 665	0.012 665
	mean	0.012 670 3	0.012 706 0	0.012 683	0.012 665
	worst	0.012 679 0	0.012 812 0	NA	0.012 665
	std	2.70e-05	4.37e-05	3.31e-05	0.00e+00

各算法的测试收敛如图 2 和图 3 所示. 图中 proposed algorithm 即为 ϵ IMFO 算法. 由图 2 和图 3 可见, ϵ IMFO 算法具有较快的收敛速度. 观察分析表 5 发现, 在 problem 1 和 problem 2 中, ϵ IMFO 算法在求解精度和稳定性上优于其余 3 种对比算法, 其求解精度最高, 稳定性最好.

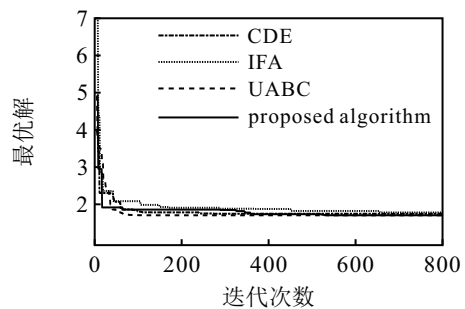


图 2 problem 1 算法收敛对比

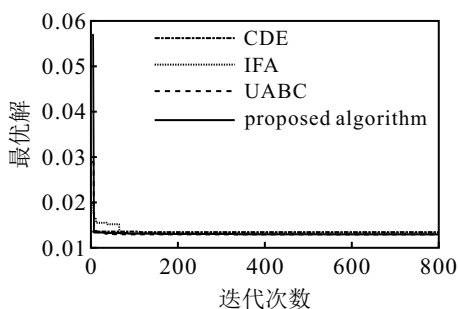


图3 problem 2算法收敛对比

5 结论

本文提出 ε IMFO算法求解约束优化问题. 该算法对 ε 约束法中 ε 的计算公式进行改进, 在火焰更新阶段采用改进的更新方法, 在飞蛾变异阶段增加优秀火焰个体对飞蛾变异的指导作用并引入2个随机火焰个体影响飞蛾变异. 利用25个基准测试函数验证所提出算法的可行性和有效性, 实验结果表明, 所提出 ε IMFO算法在求解效果和稳定性上表现优异, 适用于求解约束优化问题. 本文还将算法用于解决2个工程优化问题, 取得了良好的优化结果. 通过分析测试, 本文给出了 α 和 β 的合理取值, 在其他约束优化问题上 α 和 β 的值可采用相似的方式确定. 在后续的研究中, 可考虑在该算法的基础上进行改进后用于求解多目标约束优化问题.

参考文献(References)

- [1] Ab Wahab M N, Nefti-Meziani S, Atyabi A. A comprehensive review of swarm optimization algorithms[J]. PLoS One, 2015, 10(5): e0122827.
- [2] Cantú V H, Azzaro-Pantel C, Ponsich A. Constraint-handling techniques within differential evolution for solving process engineering problems[J]. Applied Soft Computing, 2021, 108: 107442.
- [3] Gen M, Cheng R. A survey of penalty techniques in genetic algorithms[C]. Proceedings of IEEE International Conference on Evolutionary Computation. Nagoya, 1996: 804-809.
- [4] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization[J]. IEEE Transactions on Evolutionary Computation, 2000, 4(3): 284-294.
- [5] Deb K. An efficient constraint handling method for genetic algorithms[J]. Computer Methods in Applied Mechanics and Engineering, 2000, 186(2/3/4): 311-338.
- [6] Takahama T, Sakai S. Constrained optimization by ε constrained particle swarm optimizer with ε -level control[M]. Advances in Soft Computing. Heidelberg: Springer, 2005: 1019-1029.
- [7] Chootinan P, Chen A. Constraint handling in genetic algorithms using a gradient-based repair method[J]. Computers & Operations Research, 2006, 33(8): 2263-2281.
- [8] Huang F Z, Wang L, He Q. An effective co-evolutionary differential evolution for constrained optimization[J]. Applied Mathematics and Computation, 2007, 186(1): 340-356.
- [9] Wang Y, Cai Z X, Zhou Y R, et al. An adaptive tradeoff model for constrained evolutionary optimization[J]. IEEE Transactions on Evolutionary Computation, 2008, 12(1): 80-92.
- [10] Takahama T, Sakai S. Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites[C]. IEEE International Conference on Evolutionary Computation. Vancouver, 2006: 1-8.
- [11] Gao W F, Yen G G, Liu S Y. A dual-population differential evolution with coevolution for constrained optimization[J]. IEEE Transactions on Cybernetics, 2015, 45(5): 1094-1107.
- [12] Mirjalili S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm[J]. Knowledge-Based Systems, 2015, 89: 228-249.
- [13] Emary E, Zawbaa H M. Impact of chaos functions on modern swarm optimizers[J]. PLoS One, 2016, 11(7): e0158738.
- [14] Wang M J, Chen H L, Yang B, et al. Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses[J]. Neurocomputing, 2017, 267: 69-84.
- [15] Pelusi D, Mascella R, Tallini L, et al. An improved moth-flame optimization algorithm with hybrid search phase[J]. Knowledge-Based Systems, 2020, 191: 105277.
- [16] Liang J J, Runarsson T P, Mezura-Montes E. Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization[R]. Singapore: Nanyang Technological University, 2006.
- [17] Kumar A, Wu G H, Ali M Z, et al. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results[J]. Swarm and Evolutionary Computation, 2020, 56: 100693.
- [18] 梁静, 葛士磊, 瞿博阳, 等. 求解电力系统经济调度问题的改进粒子群优化算法[J]. 控制与决策, 2020, 35(8): 1813-1822.
(Liang J, Ge S L, Qu B Y, et al. Improved particle swarm optimization algorithm for solving power system economic dispatch problem[J]. Control and Decision, 2020, 35(8): 1813-1822.)

- [19] Yu K J, Wang X, Wang Z L. Constrained optimization based on improved teaching-learning-based optimization algorithm[J]. *Information Sciences*, 2016, (352/353): 61-78.
- [20] Rao R V, Patel V. An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems[J]. *International Journal of Industrial Engineering Computations*, 2012, 3(4): 535-560.
- [21] Zhang W W, Yen G G, He Z S. Constrained optimization via artificial immune system[J]. *IEEE Transactions on Cybernetics*, 2014, 44(2): 185-198.
- [22] Elsayed S M, Sarker R A, Essam D L. Multi-operator based evolutionary algorithms for solving constrained optimization problems[J]. *Computers & Operations Research*, 2011, 38(12): 1877-1896.
- [23] Mezura-Montes E, Miranda-Varela M E, del Carmen Gómez-Ramón R. Differential evolution in constrained numerical optimization: An empirical study[J]. *Information Sciences*, 2010, 180(22): 4223-4262.
- [24] Trivedi A, Srinivasan D, Biswas N. An improved unified differential evolution algorithm for constrained optimization problems[C]. *IEEE Congress on Evolutionary Computation*. Rio de Janeiro, 2018: 1-10.
- [25] Hellwig M, Beyer H-G. A matrix adaptation evolution strategy for constrained real-parameter optimization[C]. *IEEE Congress on Evolutionary Computation*. Rio de Janeiro, 2018: 749-756.
- [26] Fan Z, Fang Y, Li W J, et al. LSHADE44 with an improved ϵ constraint-handling method for solving constrained single-objective optimization problems[C]. *IEEE Congress on Evolutionary Computation*. Rio de Janeiro, 2018: 431-438.
- [27] Ragsdell K M, Phillips D T. Optimal design of a class of welded structures using geometric programming[J]. *Journal of Engineering for Industry*, 1976, 98(3): 1021-1025.
- [28] Belegundu A D, Arora J S. A study of mathematical programming methods for structural optimization. Part I: Theory[J]. *International Journal for Numerical Methods in Engineering*, 1985, 21(9): 1583-1599.
- [29] Huang F Z, Wang L, He Q. An effective co-evolutionary differential evolution for constrained optimization[J]. *Applied Mathematics and Computation*, 2007, 186(1): 340-356.
- [30] 龙文, 蔡绍洪, 焦建军, 等. 求解约束优化问题的萤火虫算法及其工程应用[J]. *中南大学学报: 自然科学版*, 2015, 46(4): 1260-1267.
(Long W, Cai S H, Jiao J J, et al. Firefly algorithm for solving constrained optimization problems and engineering applications[J]. *Journal of Central South University: Science and Technology*, 2015, 46(4): 1260-1267.)
- [31] Brajevic I, Tuba M L. An upgraded artificial bee colony algorithm for constrained optimization problems[J]. *Journal of Intelligent Manufacturing*, 2013, 24(4): 729-740.
- [32] 石建平, 李培生, 刘国平, 等. 求解约束优化问题的改进果蝇优化算法及其工程应用[J]. *控制与决策*, 2021, 36(2): 314-324.
(Shi J P, Li P S, Liu G P, et al. Improved fruit fly optimization algorithm for solving constrained optimization problems and engineering applications[J]. *Control and Decision*, 2021, 36(2): 314-324.)

作者简介

叶文静(1997—),女,硕士生,从事约束优化算法的研究, E-mail: ywj18321676036@163.com;

曹萃文(1967—),女,副教授,博士,从事流程工业生产计划与生产调度技术、供应链管理与优化等研究, E-mail: caocuiwen@ecust.edu.cn;

顾幸生(1960—),男,教授,博士生导师,从事工业过程故障检测与故障诊断、工业生产计划与生产调度等研究, E-mail: xsgu@ecust.edu.cn.