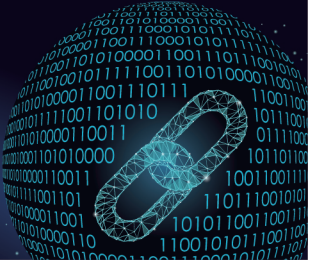




中国科技期刊卓越行动计划项目入选期刊

控制与决策

CONTROL AND DECISION



求解带约束优化问题的混合式多策略萤火虫算法

吕莉, 潘宁康, 肖人彬, 王晖, 谭德坤

引用本文:

吕莉, 潘宁康, 肖人彬, 王晖, 谭德坤. 求解带约束优化问题的混合式多策略萤火虫算法[J]. 控制与决策, 2024, 39(8): 2551–2559.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2023.0051>

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>

一种基于MOEA/D的组合权重方法

A combination weight method based on MOEA/D

控制与决策. 2021, 36(12): 3056–3062 <https://doi.org/10.13195/j.kzyjc.2020.0592>

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>

基于正态云模型的状态转移算法求解多目标柔性作业车间调度问题

State transition algorithm based on normal cloud model for solving multi-objective flexible job shop scheduling problem

控制与决策. 2021, 36(5): 1181–1190 <https://doi.org/10.13195/j.kzyjc.2019.1233>

基于改进萤火虫算法的区域交通信号配时优化

Timing optimization of regional traffic signals based on improved firefly algorithm

控制与决策. 2020, 35(12): 2829–2834 <https://doi.org/10.13195/j.kzyjc.2019.1835>

求解带约束优化问题的混合式多策略萤火虫算法

吕莉^{1†}, 潘宁康¹, 肖人彬², 王晖¹, 谭德坤¹

(1. 南昌工程学院 信息工程学院, 南昌 330099; 2. 华中科技大学 人工智能与自动化学院, 武汉 430074)

摘要: 目前多目标优化算法主要针对如何处理多个目标之间的冲突, 对于如何处理约束考虑较少, 鉴于此, 提出一种求解带约束优化问题的混合式多策略萤火虫算法(HMSFA-PC). 首先, 提出一种改进的动态罚函数策略对约束优化问题进行预处理, 将其转换为非约束优化问题; 其次, 对萤火虫算法本身进行改进, 采用 Lévy flights 搜索机制有效地增大搜索范围; 接着, 引入随机扩张因子改进算法吸引模型, 使种群突破束缚, 有效避免早熟收敛, 提出自适应维度重组机制, 根据不同迭代时期选择差异性较大的个体进行信息交互、相互学习. 为检验算法处理无约束优化问题的性能, 将其在基准测试函数上与部分典型算法进行比较; 为检验算法处理约束优化问题的性能, 将其在实际约束测试问题中与一些顶尖约束求解算法进行比较. 结果表明, HMSFA-PC 在处理无约束优化问题时具有收敛速度快、收敛精度高等优势, 并且在动态罚函数的协作下求解实际约束优化问题时仍具有良好的优化性能.

关键词: 萤火虫算法; 约束多目标优化; 动态罚函数法; Lévy flights; 随机扩张因子; 自适应维度重组

中图分类号: TP301.6

文献标志码: A

DOI: 10.13195/j.kzyjc.2023.0051

引用格式: 吕莉, 潘宁康, 肖人彬, 等. 求解带约束优化问题的混合式多策略萤火虫算法[J]. 控制与决策, 2024, 39(8): 2551-2559.

Hybrid multi-strategy firefly algorithm for solving optimization problems with constraints

LV Li^{1†}, PAN Ning-kang¹, XIAO Ren-bin², WANG Hui¹, TAN De-kun¹

(1. School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China; 2. School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: We propose a hybrid multi-strategy firefly algorithm (HMSFA-PC) for solving constrained optimization problems. Firstly, an improved dynamic penalty function strategy is proposed to preprocess the constrained optimization problem so as to convert it into an unconstrained optimization problem. Secondly, the firefly algorithm itself is improved: the Lévy flights search mechanism is adopted to effectively increase the search range; a random expansion factor is introduced to improve the attraction model of the algorithm so that the population breaks through the constraint, effectively avoiding premature convergence and maintaining the population convergence; an adaptive dimensional reorganization mechanism is proposed to maintain the population convergence. The adaptive dimensional reorganization mechanism is proposed to select individuals with greater variability according to different iteration periods to interact with information and learn from each other, effectively improving the diversity of the population. To test the performance of the algorithm in dealing with unconstrained optimization problems, it is compared with some typical algorithms on the benchmark test function; to test the performance of the algorithm to deal with constrained optimization problems, it is compared with some top constraint solving algorithms on actual constraint test problems. The results show that the HMSFA-PC has the advantages of fast convergence and high convergence accuracy when dealing with unconstrained optimization problems, and still has good optimization performance when solving real constrained optimization problems with the collaboration of dynamic penalty functions.

Keywords: firefly algorithm; constrained multi-objective optimization; dynamic penalty function method; Lévy flights; stochastic expansion factor; adaptive dimensional reorganization

收稿日期: 2023-01-13; 录用日期: 2023-06-04.

基金项目: 科技创新 2030-“新一代人工智能”重大项目(2018AAA0101200); 国家自然科学基金项目(62066030).

责任编委: 陈家伟.

[†]通讯作者. E-mail: lvli623@163.com.

0 引言

约束优化问题(constrained optimization problem, COPs)作为优化领域中的重要问题,广泛存在于工程实际中且求解困难,因此,研究此类问题具有重要实际意义. 寻找COPs的最优解是一项极其严峻的挑战,需要根据问题的类型以及算法具有的特征正确选择和使用算法.

相较于大多数基于梯度下降的传统优化算法,群智能算法具有实现简单、搜索能力强和扩展性强的特点,已成功应用于无约束多目标优化领域. 然而,几乎所有的群智能算法在算法原理和算法设计上都是针对无约束优化问题的,没有考虑如何解决含约束的问题. 因此,将群智能算法用于求解COPs很有意义. 群智能算法在求解COPs时除了要解决无约束多目标优化过程中面临的提高算法搜索能力、避免陷入局部最优以及合理设置参数等多项要求外,还必须针对问题的强约束性提高目标函数和约束条件的合理平衡能力,从而保证算法获得可靠的求解性能和全局优化效果^[1].

2008年, Yang^[2]通过对萤火虫群体行为的模仿,提出萤火虫算法(firefly algorithm, FA). 与其他群智能算法相比,FA具有参数较少和容易实现等优点,一经提出便被用于求解不同领域的优化问题^[3]. 鉴于求解单目标优化问题的有效性,FA被扩展为多目标萤火虫算法(multi-objective firefly algorithm, MOFA)^[4]以求解多目标优化问题. 正如“无免费午餐定理”所指出的,没有单一算法可以完美解决所有问题,每个算法都有各自的优缺点,MOFA也是如此,在寻找问题最优解的过程中MOFA存在寻优精度低、易陷入早熟收敛、算法多样性不佳的问题.

为了解决MOFA存在的不足,学者们提出了各自的改进措施^[5-8],这些改进虽然在一定程度上改善了MOFA的优化性能,但存在如下问题:均未提及对于约束的处理;没有凸显出萤火虫个体之间的性能差异,普遍存在“集群学习”现象;算法的分布性和收敛性存在进一步提升的空间. 鉴于此,本文提出一种求解带约束优化问题的混合式多策略萤火虫算法(hybrid multi-strategy firefly algorithm for solving optimization problems with constraints, HMSFA-PC),通过Lévy flights、随机扩张因子以及自适应维度重组机制3种策略对算法本身进行改进,在处理COPs时通过改进动态罚函数策略将实际约束问题进行预处理,从而将其转换为非约束优化问题进行求解.

1 相关基础知识

1.1 约束优化问题

求解COPs时算法的性能主要取决于算法本身和约束处理方法,其本质上是寻求一个决策变量 x 的过程,使 x 最小化目标函数的同时满足对搜索空间线性和非线性约束^[9]. 不失一般性,其模型可表示为

$$\begin{cases} \min F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T, \\ x = (x_1, x_2, \dots, x_n) \in \Omega \rightarrow R^n, \\ F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \in \Omega \rightarrow R^m. \end{cases}$$

$$\text{s.t. } h_i(X) \leq 0, i = 1, 2, \dots, s;$$

$$g_j(X) = 0, j = s + 1, s + 2, \dots, q. \quad (1)$$

其中: x 为 n 维决策空间中的决策向量, $F(x)$ 为 m 维目标空间中的目标向量; $h_i(X)$ 和 $g_j(X)$ 分别为第 i 个不等式约束和第 j 个等式约束.

1.2 多目标萤火虫算法

在MOFA中,亮度和吸引力是两个关键要素. 萤火虫个体所处位置的优劣及其移动的方向由亮度决定,移动的距离由吸引力决定,通过亮度和吸引力的不断更新实现目标优化. 萤火虫之间的吸引力与其亮度成正比,而与距离成反比.

萤火虫 j 对萤火虫 i 的吸引力定义如下:

$$\beta_{ij}(r_{ij}) = \beta_0 e^{-\gamma(r_{ij})^2}. \quad (2)$$

其中: β_0 为最大吸引力,等于 $r = 0$ 光源处的吸引力,通常取 $\beta_0 = 1$; γ 为光吸收系数,一般取 $\gamma \in [0.01, 100]$; r_{ij} 为萤火虫 i 到萤火虫 j 的欧氏距离.

解决多目标优化问题,常常利用Pareto支配的概念确定萤火虫个体之间的吸引关系,以萤火虫 i 作为研究对象,如果萤火虫 $j \prec i$,表明 j 为更亮的萤火虫,此时萤火虫 i 被更亮的萤火虫 j 吸引.

MOFA的萤火虫位置更新分为两种情况:

一是如果萤火虫 i 寻找到支配解萤火虫 j ,即 $j \prec i$,则萤火虫 i 的位置更新为

$$x_i(t+1) = x_i(t) + \beta_{ij}(r_{ij}) \cdot (x_j(t) - x_i(t)) + \alpha_t \varepsilon_i^t. \quad (3)$$

其中: t 为当前迭代次数; x_i 、 x_j 分别为萤火虫 i 和 j 处的空间位置; $\beta_{ij}(r_{ij})$ 为萤火虫 i 与 j 之间的吸引力; α_t 为常数,一般取 $\alpha \in [0, 1]$; ε_i^t 为由高斯分布、均匀分布或其他分布得到的随机数向量.

二是如果萤火虫 i 未找到支配解萤火虫 j ,则萤火虫 i 的位置更新公式为

$$x_i(t+1) = g_*^t + \alpha_t \varepsilon_i^t, \quad (4)$$

其中 g_* 为由多个目标函数以随机加权的方式转换

为单目标函数取得的最优值。

1.3 惩罚函数法

约束处理技术主要分为5类: 罚函数法^[10]、可行性准则法^[11]、约束法^[12]、多目标优化法^[13]和混合法^[14]。罚函数法通过给目标函数增加一个惩罚项, 将COPs转化为无约束优化问题进行求解。相较于其他4种约束处理技术罚函数法具有执行简单普适性较高的特点, 因此应用广泛, 尤其是应用到群智能算法里处理COPs^[15], 故本文将改进的萤火虫算法与罚函数法相结合进行研究。

罚函数法主要通过通过对目标函数施加惩罚措施构造出新的目标函数, 从而达到将COPs转化为无约束优化问题的目的。作为惩罚函数法的核心问题, 惩罚因子的选取一直是一个难题, 若惩罚因子取值过小则对不可行解的惩罚不足, 导致算法对可行解的搜索受到局限, 反之则对算法收敛到全局最优解产生不利因素, 因此惩罚因子值的合理选取对求解此类COPs至关重要。

Saha等^[16]提出了一种模糊惩罚法, 主要通过两个惩罚项对约束条件进行惩罚: 第1个惩罚项通过Mamdani模糊推理系统将可行解的比例等信息进行输入从而得到所需要的惩罚值; 第2项则通过可行解的比例动态调约束条件和目标函数。Tessema等^[17]提出一种自适应惩罚函数法, 依据种群中实时可行解的比例对惩罚因子的大小进行自适应调节, 通过找出种群中较优的不可行解从而引导种群向可行域移动。Kusakci等^[18]通过将惩罚函数与改进的协方差矩阵适应进化策略相结合, 从而得到具有较高准确性的解。

2 混合式多策略萤火虫算法

针对MOFA在处理实际约束问题时存在进化后期收敛速度慢、早熟收敛、难以收敛到最优解、种群多样性不佳的问题, 首先, 采用lévy flights搜索机制有效地增大搜索范围, 增强种群逃逸出局部最优的能力, 有效维护种群的收敛性; 其次, 为跳出局部最优避免早熟收敛, 在种群迭代过程中加入随机扩张因子, 使种群不断向着最优解逼近; 最后, 加入提出的自适应维度重组机制, 在算法原有的基础上使种群中差异性较大的个体进行维度重组, 极大地改善了算法的多样性, 有效避免了在迭代过程中由于MOFA采用随机吸引模型导致种群多样性不佳的问题。

2.1 Lévy flights

Lévy flights的移动步长具有交错移动的特点, 属于非高斯随机过程。在算法中采用Lévy flights搜索

机制可以有效增强种群寻优的探寻范围, 使种群更加逼近真实Pareto前沿^[19]。

Lévy flights扰动项的计算公式如下:

$$s_i = \frac{\lambda}{|v|^{1/\varphi}}. \quad (5)$$

其中 λ 和 v 均服从正态分布

$$\begin{cases} \lambda \sim N(0, \eta_\lambda^2), \\ v \sim N(0, \eta_v^2). \end{cases} \quad (6)$$

η_λ 和 η_v 定义如下:

$$\begin{cases} \eta_\lambda = \left\{ \frac{\Gamma(1+\varphi) \sin(\pi\varphi/2)}{\Gamma[(1+\varphi)/2] 2^{(\varphi-1)/2} \varphi} \right\}^{1/\varphi}, \\ \eta_v = 1, \end{cases} \quad (7)$$

其中 φ 的取值范围为 $[1, 2]$, 本文取 $\varphi = 1.5$ 。经过Lévy flights改进后的萤火虫算法位置更新公式(3)更新为

$$\begin{aligned} x_i(t+1) &= x_i(t) + \beta_{ij}(r_{ij}) \cdot (x_j(t)x_i(t)) + \\ &\alpha_t \cdot \text{Lévy} \sim \lambda / |v|^{1/\varphi}, \end{aligned} \quad (8)$$

式(4)更新为

$$x_i(t+1) = g_*^t + \alpha_t \cdot \text{Lévy} \sim \lambda / |v|^{1/\varphi}, \quad (9)$$

其中 $\text{Lévy} \sim \lambda / |v|^{1/\varphi}$ 为Lévy flights随机扰动项。

2.2 随机扩张因子

由于MOFA位置更新公式固有的缺陷和Pareto最优解的局限性, 算法极易陷入局部最优。第2.1节经Lévy flights对位置更新公式进行改进, 算法寻优速度和收敛性虽然有了极大的改善, 但是种群集群现象严重, 具体表现为较集中地分布在Pareto前沿某一段区域, 而不是均匀分布于整个前沿上, 由此提出一种随机扩张因子策略以增加萤火虫的移动步长, 从而突破种群束缚, 改善算法的分布性。

由于在MOFA中萤火虫位置更新过程中只受到支配其的萤火虫个体和随机扰动项的影响, 不受任何支配的萤火虫个体仅受到当前最优解和随机扰动项的影响, 使种群很难逃逸出局部最优, 集群效应严重。本文在Lévy flights改进的基础上加入随机扩张因子 c , 并使 c 在迭代过程中依据迭代次数的不同改变相应值的大小, c 应保证种群在迭代前期具有较强的全局探索能力, 并且在后期具有较强的局部开发能力, 计算公式如下:

$$c = \begin{cases} 6 \cdot e^{(t/\text{Maxit})}, & t \leq \text{Maxit}/2; \\ e^{(t/\text{Maxit})}, & t > \text{Maxit}/2. \end{cases} \quad (10)$$

经改善后, 式(8)更新为

$$x_i(t+1) = x_i(t) + c \cdot \beta_{ij}(r_{ij}) \cdot (x_j(t)x_i(t)) +$$

$$\alpha_t \cdot \text{Lévy} \sim \lambda/|v|^{1/\varphi}, \quad (11)$$

式(9)更新为

$$x_i(t+1) = g_*^t + c \cdot \alpha_t \cdot \text{Lévy} \sim \lambda/|v|^{1/\varphi}. \quad (12)$$

2.3 自适应维度重组

经前两部分改进,算法的收敛性和分布性都有了不同程度的提高,但是仍存在局部最优和多样性较差的问题.为了充分利用萤火虫个体之间的优良信息,本文提出自适应维度重组机制,根据个体间的相似性大小决定是否进行重组操作.每次萤火虫吸引力、位置、亮度更新后,在萤火虫种群中采用随机选择的方法从个体中选择比例为 p_c 的个体进行维度重组产生新个体,选取比例 p_c 的计算公式为

$$c = \begin{cases} \frac{0.25(N - \text{nan})}{N} + 0.25, & t \in [0, \alpha \text{Maxit}]; \\ \frac{0.5(N - \text{nan})}{N} + 0.5, & t \in [\alpha \text{Maxit}, \text{Maxit}]. \end{cases} \quad (13)$$

其中: N 为种群数量; nan 为非支配解的个数,以保证选取比例 p_c 具有较合适的范围; α 为随机数,一般取0.5; Maxit 为最大迭代次数,表示迭代次数在 $[0, \text{Maxit}/2]$ 和 $[\text{Maxit}/2, \text{Maxit}]$ 时分别采用不同的比例计算公式,确保种群个体可以以优异的比例进行维度重组.

确定种群中需要进行维度重组的个体比例后,需要进行参与重组个体的选择,选择个体时应保证两个个体存在较大的差异性.本文将差异性定义如下:假设萤火虫个体 P 和 Q 在对应维度上的差值为 B ,维度总长度为 L ,个体在此维度上的取值区间为 H ,如果 $B > H/5$ 且连续存在 $\geq (1/3)L$ 个维度,则认为个体 P 与 Q 之间存在较大的差异性,可进行维度重组操作.选择一段差异性较大的区间进行重组,计算交叉后子代个体的适应值,如果适应值优于父代个体,则萤火虫的位置进行更新,否则萤火虫位置保持不变.

为了更便于展示自适应维度重组机制对种群的改善情况,以二目标问题为例,绘制自适应维度重组前后个体分布图如图1所示.

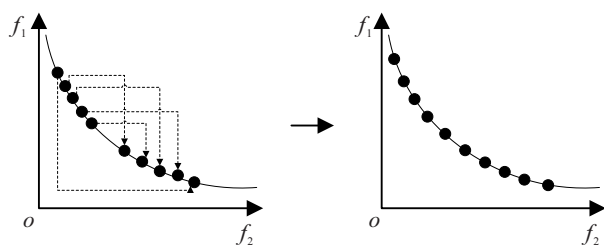


图1 自适应维度重组前后MOFA搜索机制简化模型

图1中黑色圆圈表示萤火虫个体,黑色曲线表示

Pareto前沿.可以看出在维度重组前虽然经第2.1节和第2.2节改进后的萤火虫个体收敛性和分布性均有了不同程度的提升,但是多样性仍有一定的提升空间.图1中经曲线连接的两个个体为满足文中设置的差异性个体,即进行维度重组的两个个体.右图可以清晰地看出经维度重组后个体的多样性有了进一步的提高.

2.4 HMSFA-PC算法流程

输入: 决策变量维数 D , 区间范围 $[a, b]$, 种群规模 N , 最大迭代次数 Maxit , 光吸收系数 γ , 最大吸引力 β_0 , 初始步长 α ;

输出: Pareto最优解集.

step 1: 初始化, 设置决策变量维度 D 、区间范围 $[a, b]$, 种群规模 N , 最大迭代次数 Maxit , 光吸收系数 γ , 最大吸引力 β_0 , 以及初始步长 α .

step 2: 种群初始化, 计算每一个目标函数上的适应值.

step 3: 当 $t \leq \text{Maxit}$ 时, 对萤火虫进行遍历.

step 4: 采用 Lévy flights 生成扰动项 $\text{Lévy} \sim \lambda/|v|^{1/\varphi}$, 按照式(10)生成扩张因子.

step 5: 萤火虫 i 与萤火虫 j 进行比较, 若萤火虫个体 $j < i$, 则按照式(11)进行位置更新; 如果萤火虫 i 不受任何个体支配, 则按照式(12)进行位置更新.

step 6: 自适应维度重组. 对位置更新后的萤火虫个体进行维度重组操作, 比较维度重组前后的个体, 如果重组后的个体适应值优于变异前个体适应值, 则更新萤火虫位置, 否则保留萤火虫个体原来的位置.

step 7: 确保萤火虫个体位置在界限范围内, 如果不在则强制等于上界或等于下界.

step 8: 更新个体的适应值.

step 9: $t = t + 1$.

step 10: 若 $t > \text{Maxit}$, 则算法运行结束, 输出 Pareto最优解集.

2.5 算法时间复杂度分析

设种群具有 N 个个体, M 个目标, 变量维数为 D . MOFA采用全吸引模型(即两层循环)遍历所有个体, 其时间复杂度为 $O(N)^2$. HMSFA-PC在MOFA的基础上添加 Lévy flights、随机扩张因子和自适应维度重组. 设对个体进行 Lévy flights 所需要的时间为 $h(D)$, 计算随机扩张因子需要的时间为 $g(D)$, 由于求解的个体维数相比种群中的个体数要小得多, $h(D)$ 和 $g(D)$ 远小于 N ; 维度重组包括两层循环, 第1层循环为遍历种群中 Pareto 等级为1的个体, 设该个体为 K , 第2层循环为遍历个体的所有维度. 综

上, HMSFA-PC 的时间复杂度为 $(h(D) + g(D)) \times O(N)^2 + O(K \times D)$, 由于 $K < N, D < N$, 比较可得 HMSFA-PC 与 MOFA 的时间复杂度在同一数量级上. HMSFA-PC 虽然增加了部分计算量, 但 HMSFA-PC 能以较少的迭代次数收敛到 Pareto 前沿, 相较 MOFA 更具有实际意义.

3 约束处理

处理 COPs 时存在非线性约束、约束区域不连续等挑战, 求解 COPs 的关键在于如何处理可行解与不可行解的关系. 由第 1.3 节所述, 选取罚函数法作为约束的处理方法, 即对于约束优化问题, 通过罚函数进行预处理, 将其转化为非约束优化问题从而进行求解, 但惩罚因子的取值依赖于所优化的问题, 难以选取. 为此, 本文在原有的罚函数法基础上提出一种新的动态罚函数策略, 惩罚因子根据种群中的可行解比例和进化代数动态变化^[20]. 为方便设计, 文中等式约束和不等式约束均用相同的惩罚因子, 则优化问题为

$$\min P(X) = \min F(X) + \mu H(X). \quad (14)$$

其中: $\min F(X) = (f_1(x), f_2(x), \dots, f_m(x))^T$ 为各个目标函数;

$$H(x) = \sum_j^s V_j(g_j(x)) + \sum_{j=s+1}^q V_j(h_j(x)) |h_j(x)|,$$

$V_j(g_j(x))$ 和 $V_j(h_j(x))$ 均为指示函数, 且有

$$V_j(g_j(x)) = \begin{cases} 1, & g_j(x) > 0; \\ 0, & g_j(x) \leq 0. \end{cases} \quad (15)$$

$$V_j(h_j(x)) = \begin{cases} 1, & |h_j(x)| > \varepsilon; \\ 0, & |h_j(x)| \leq \varepsilon. \end{cases} \quad (16)$$

这里: $g_j(x)$ 和 $h_j(x)$ 分别为等式约束条件和不等式约束条件, ε 为精度要求, 当 $V(x) < \varepsilon$ 时, 无约束优化问题和 COPs 的解是等价的. 惩罚因子按如下方式进行选取:

$$\mu(t) = 10^{\frac{(\delta_2 - \delta_1) \times (\text{npop} - \text{nan})}{\text{Maxit} - t}} + \delta_1. \quad (17)$$

其中: t 为种群当前迭代次数; Maxit 为种群最大迭代次数; δ_1 、 δ_2 为本文自定义的常数, 在此选 $\delta_1 = 1$ 、 $\delta_2 = 3$. 符合期望, 即在进化初期采用较小的惩罚系数有利于种群勘探; 在进化后期或当种群的可行解比例大于设定的阈值时采用较大的惩罚系数, 有利于种群的开发, 以此平衡种群的勘探能力和开发能力.

4 实验与结果分析

4.1 无约束优化问题

为了检验处理无约束优化问题时算法的优化性能, 将 HMSFA-PC 与 5 种典型的多目标优化算法在

ZDT、Deb、Viennet 和 DTLZ 系列函数上进行实验验证, 包括基于 Pareto 优势的算法 MOPSO^[21]、PESA-II^[22]、MOFA^[4] 以及基于分解的算法 NSGA-III^[23]、MOEA/D^[24].

为综合评估所有算法的收敛性和多样性, 选用 IGD 指标在 ZDT、Deb、Viennet 和 DTLZ 系列多目标测试函数集上实现 6 种对比算法的结果测试论证, 并采用 Friedman 检验对各算法进行排名分析. 为确保算法对比的公平性, 各算法种群规模设置为 100, 每个测试函数评估 10 000 次, 为平衡随机性, 各算法在每个函数上独立运行 30 次. 算法参数设置与相应文献保持一致.

表 1 给出了 HMSFA-PC 与 5 种典型的多目标优化算法 (multi-objective evolutionary algorithm, MOEA) 在 18 个测试问题上的 IGD 性能评价指标的平均值、标准差和程序运行的运行时间 (单位 s), 同时列出了不同算法在测试函数集上获得的最优值总数. 此外, 表 1 倒数第 2 行运用 Friedman 检验给出各算法优化结果的秩平均值, 针对 IGD 指标, 秩平均值越小表示算法的性能越优. 最后一行为各算法优化性能的平均排名, 其中加粗数据表示最优结果.

根据表 1, 从算法的综合性能角度进行分析, HMSFA-PC 在 18 个测试问题共有 15 次取得了最好的优化效果, PESA-II 取得 2 次, NSGA-III 和 MOEA/D 均取得 1 次. 可以看出, 在 ZDT、Deb、Viennet 测试问题中, HMSFA-PC 均取得了最好的优化结果. 值得注意的是, 在求解 DTLZ1 时, 此测试函数具有多模态的特点导致从决策空间到目标空间的映射并不总是一一对应, 所以 HMSFA-PC 在求解此问题时有较大的困难. 6 种对比算法在 18 个测试问题上取得的最优值个数表明, HMSFA-PC 在综合性能上优于其余算法, 结果充分展示出改进策略对算法寻优过程的改进作用非常明显. 根据“世上没有免费的午餐”定律, 很多改进算法在追求算法精度的同时, 势必会牺牲算法的时间或空间. 由表 1 可见, 在算法运行时间方面, MOPSO 略优于其他对比算法, HMSFA-PC、MOFA 和 PESA-II 不分上下, MOEA/D 和 NSGA-III 相较于其他算法运行时间虽然保持同一数量级, 但是明显有了不同程度的增加. 值得一提的是, MOPSO 虽然在算法运行时间方面略优于 HMSFA-PC, 但是精度方面要比 HMSFA-PC 差得多.

从 Friedman 检验结果可见, HMSFA-PC 获得的秩平均值最小, 排名第 1, 且远小于其他 5 种对比算法, 这表明在 6 种算法中, HMSFA-PC 的综合性能最好.

表1 HMSFA-PC 与 5 种经典 MOEA 在 IGD 上的实验结果与运行时间 单位: s

测试函数	结果	MOPSO	NSGA-III	MOEA/D	PESA-II	MOFA	HMSFA-PC
ZDT1	均值	1.01e+00	1.96e-02	1.48e-01	3.55e-02	6.43e-02	3.83e-03
	标准差	2.36e-01	3.56e-03	6.73e-02	3.08e-02	1.17e-02	1.21e-05
	运行时间	3.80e+00	9.73e+00	3.26e+01	1.42e+01	1.82e+01	1.83e+01
ZDT2	均值	1.79e+00	8.18e-02	5.50e-01	1.26e-01	1.98e-01	3.76e-03
	标准差	2.24e-01	1.06e-01	9.79e-02	7.99e-02	6.13e-02	2.22e-05
	运行时间	4.30e+00	4.55e+01	2.17e+01	1.32e+01	1.95e+01	1.75e+01
ZDT3	均值	4.83e-01	1.71e-01	2.34e-01	1.34e-01	4.87e-02	4.63e-03
	标准差	2.57e-01	2.33e-02	7.49e-02	2.63e-02	9.44e-03	1.03e-04
	运行时间	2.03e+00	3.88e+01	3.24e+01	1.13e+01	1.88e+01	2.30e+01
ZDT4	均值	1.72e+01	4.59e-01	6.58e-01	2.21e-01	4.08e-01	5.85e-03
	标准差	3.77e+00	2.90e-01	1.84e-01	6.59e-02	9.48e-02	1.97e-04
	运行时间	3.15e+00	4.13e+01	2.08e+01	1.17e+01	1.83e+01	2.06e+01
ZDT6	均值	4.76e-01	2.20e-01	8.49e-02	2.63e-02	4.06e-01	3.10e-03
	标准差	7.01e-01	6.89e-02	2.66e-02	1.54e-02	1.34e-01	1.90e-05
	运行时间	2.59e+00	4.49e+01	2.17e+01	1.36e+01	2.02e+01	2.29e+01
Deb1	均值	8.47e-03	1.47e-02	1.10e-02	8.50e+00	1.15e-02	3.77e-03
	标准差	8.21e-04	7.74e-03	1.80e-03	8.37e-04	1.40e-03	5.24e-05
	运行时间	1.07e+01	4.17e+01	3.53e+01	2.85e+01	1.74e+01	1.74e+01
Deb2	均值	7.75e-03	2.45e-02	3.42e-02	1.02e-02	1.83e-02	4.11e-03
	标准差	9.65e-04	1.15e-02	1.51e-02	1.57e-03	3.46e-03	4.49e-05
	运行时间	6.63e+00	4.11e+01	3.32e+01	1.63e+01	1.82e+01	1.61e+01
Deb3	均值	7.52e-03	1.31e-02	1.52e-02	8.09e-03	9.32e-03	3.86e-03
	标准差	1.09e-03	4.79e-03	6.50e-03	1.04e-03	1.61e-03	1.86e-05
	运行时间	8.62e+00	3.37e+01	3.51e+01	3.52e+01	1.86e+01	1.70e+01
Viennet1	均值	1.34e-01	1.71e-01	2.04e-01	1.39e-01	1.59e-01	1.16e-01
	标准差	3.89e-03	1.17e-02	5.12e-03	7.36e-03	5.77e-03	1.23e-03
	运行时间	1.03e+01	3.52e+01	3.62e+01	3.87e+01	1.72e+01	1.78e+01
Viennet2	均值	1.80e-02	3.78e-02	1.10e-01	1.88e-02	2.92e-02	1.17e-02
	标准差	1.04e-03	1.24e-02	9.26e-04	7.89e-04	3.44e-03	3.27e-04
	运行时间	5.14e+00	3.73e+01	3.28e+01	2.18e+01	1.81e+01	1.73e+01
Viennet3	均值	4.66e-02	1.65e+00	1.99e+00	5.42e-02	1.53e+00	3.69e-02
	标准差	4.62e-03	4.22e-01	6.02e-02	5.30e-03	8.33e-01	1.16e-03
	运行时间	5.95e+00	3.83e+01	3.57e+01	1.99e+01	1.79e+01	1.60e+01
Viennet4	均值	2.16e-01	2.33e-01	7.29e-01	1.89e-01	2.39e-01	1.57e-01
	标准差	1.42e-02	4.93e-02	1.04e-01	1.72e-02	2.88e-02	3.16e-03
	运行时间	1.25e+01	3.74e+01	2.79e+01	2.67e+01	1.78e+01	1.80e+01
DTLZ2	均值	9.94e-02	5.49e-02	5.49e-02	6.69e-02	7.02e-02	5.86e-02
	标准差	5.42e-03	2.07e-04	2.46e-04	3.86e-03	1.75e-03	1.06e-03
	运行时间	9.73e+00	3.64e+01	3.39e+01	2.50e+01	1.77e+01	1.84e+01
DTLZ3	均值	1.29e+02	8.71e+00	1.89e+01	8.88e+00	3.82e+01	4.13e-01
	标准差	7.95e+01	3.19e+00	1.03e+01	3.79e+00	1.15e+01	7.20e-02
	运行时间	3.85e+00	4.45e+01	3.37e+01	1.31e+01	1.83e+01	1.75e+01
DTLZ4	均值	4.67e-01	2.49e-01	4.38e-01	6.44e-02	6.98e-01	5.74e-02
	标准差	2.76e-01	2.51e-01	3.46e-01	1.68e-03	1.56e-01	7.08e-04
	运行时间	4.10e+00	3.75e+01	3.35e+01	2.45e+01	1.78e+01	1.73e+01
DTLZ5	均值	1.61e-02	1.38e-02	3.27e-02	1.25e-02	5.35e-01	5.14e-01
	标准差	7.75e-03	1.36e-03	4.15e-04	2.62e-03	6.52e-02	4.35e-02
	运行时间	1.06e+01	3.85e+01	3.47e+01	2.56e+01	1.73e+01	1.67e+01
DTLZ6	均值	2.69e+00	1.81e-02	1.08e-01	1.45e-02	7.90e+00	5.37e-01
	标准差	9.01e-01	1.81e-03	2.40e-01	2.47e-03	4.84e-01	1.33e-01
	运行时间	1.03e+01	3.78e+01	3.53e+01	2.41e+01	1.86e+01	1.84e+01
DTLZ7	均值	9.71e-01	2.16e-01	2.09e-01	1.07e-01	1.05e-01	6.43e-02
	标准差	1.65e-01	7.82e-02	2.73e-02	1.49e-02	7.76e-03	3.53e-02
	运行时间	8.14e+00	3.73e+01	2.57e+01	2.19e+01	1.81e+01	1.67e+01
总计		0	1	1	2	0	15
秩		4.22	3.64	4.58	2.78	4.28	1.50
最终排名		4	3	6	2	5	1

为了更直观地展现所提出算法与各经典对比算法的综合性能,限于篇幅,从18种测试函数中选取2个典型的测试函数绘出IGD指标箱线图如图2所示.

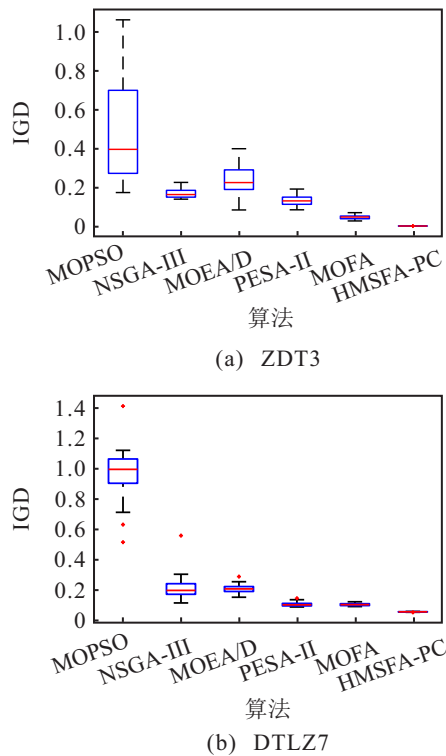


图2 6种MOEA在2个典型MOP上的IGD指标箱线图

由图2可以清晰地看出:HMSFA-PC无论是在2目标测试问题ZDT3上还是在3目标测试问题DTLZ7上,收敛稳定性和收敛精度明显比其他5种典型的MOEA优秀,表明在种群寻优的过程中可以以更高的速度和精度逼近Pareto前沿;在DTLZ7中,PESA-II和MOFA的收敛稳定性与收敛精度相差不大,值得一提的是无论是在ZDT3还是在DTLZ7中,MOPSO相较于其他几种对比算法在收敛精度和收敛稳定性方面均有明显的差距.

4.2 实际约束问题

为了检验在处理实际约束优化问题时算法的优化性能,本节将HMSFA-PC在IEEE CEC2021的29个实际约束优化问题上进行实验验证.选用文献[25]求解实际约束问题的7种算法,包括ToP^[26]、TiGe2^[27]、cNSGAIII^[28]、cMOEA/D^[28]、CCMO^[29]、cARMOEA^[30]以及AnD^[31],在CEC2021测试函数集

的机械设计问题(21个)、化学工程问题(3个)以及工艺、设计和合成问题(5个)中进行对比分析.为公平起见,本节所有算法种群大小、评估次数和运行次数均按文献[25]设置.测试函数的定义、表达式、性质等详见文献[25].HMSFA-PC的实验结果如表2所示,其余7种对比算法的结果取自文献[25].对实际约束优化问题进行实验时采用文献[25]所设计的排名方案-绩效得分(performance score)对各算法进行分析排名,结果如表3所示.

表2 HMSFA-PC在CEC2021测试问题中RCM01~RCM29上的实验结果

测试函数	最优值	均值	标准差
RCM01	6.06e-01	6.04e-01	2.50e-03
RCM02	3.93e-01	3.92e-01	1.29e-05
RCM03	9.05e-01	9.00e-01	2.23e-04
RCM04	8.63e-01	8.63e-01	2.43e-04
RCM05	4.35e-01	4.36e-01	2.15e-04
RCM06	2.75e-01	2.74e-01	2.08e-02
RCM07	3.48e-01	2.48e-01	5.92e-04
RCM08	2.87e-02	2.86e-02	7.61e-05
RCM09	4.10e-01	4.09e-01	1.97e-04
RCM10	8.49e-01	8.48e-01	3.87e-05
RCM11	9.88e-02	9.80e-02	4.98e-04
RCM12	5.72e-01	5.71e-01	2.11e-04
RCM13	9.05e-02	8.68e-02	6.98e-03
RCM14	6.18e-01	6.20e-01	5.08e-04
RCM15	0	0	0
RCM16	7.64e-01	7.64e-01	9.12e-05
RCM17	3.61e-01	2.89e-01	4.01e-02
RCM18	4.05e-01	3.98e-01	4.12e-05
RCM19	0	0	0
RCM20	1.77e-01	1.73e-01	1.82e-08
RCM21	3.17e-02	3.17e-02	7.34e-07
RCM22	5.98e+00	2.18e+00	1.13e-02
RCM23	9.99e-01	9.99e-01	4.41e-17
RCM24	1.24e+00	9.82e-01	2.32e-02
RCM25	2.42e-01	2.42e-01	4.02e-06
RCM26	1.49e-01	1.48e-01	2.54e-04
RCM27	3.07e+00	3.06e+00	6.52e-16
RCM28	1.00e+00	1.00e+00	0
RCM29	7.88e-01	7.87e-01	9.32e-04

表3 各算法在CEC2021测试问题中RCM01~RCM29上的对比结果

结果	ToP	TiGe2	cNSGAIII	cMOEA/D	CCMO	cARMOEA	AnD	HMSFA-PC
总计	4	1	3	1	1	5	2	18
绩效得分	0.2845	0.6207	0.2672	0.6336	0.3190	0.2716	0.4052	0.1940
最终排名	4	7	2	8	5	3	6	1

由表3可见,在29个测试函数上,HMSFA-PC获得了次数最多的最优值,为18次,cARMOEA获得了5次,ToP获得了4次,cNSGAIII获得了3次,AnD获得了2次,其余算法均获得了1次.值得注意的是,在问题RCM15(spring design)和RCM19(multi-product batch plant)上,所提出算法并没有给出有效解,原因可能是此问题的约束边界取值波动较大,动态惩罚系数对于求解此问题并没有给出较好的取值范围,但是在RCM22(Haverly's pooling problem)和RCM28(two reactor problem)测试问题上,众多对比算法在没有解的情况下HMSFA-PC给出了最优解.根据performance score得分情况可以看出,HMSFA-PC的平均值最小,排名第1,其次为cNSGAIII,最差为cMOEA/D.综合来看,在求解实际约束问题时HMSFA-PC表现出更好的优化性能,具有较优的实际意义.

5 结论

针对实际约束优化问题,本文提出了一种求解带约束优化问题的混合式多策略萤火虫算法.通过引入Lévy flights搜索机制增大种群的寻优空间,提升算法全局遍历能力;在Lévy flights基础上加入随机扩张因子缓解种群在寻优过程中的集群效应,有效避免种群陷入局部最优;加入提出的自适应维度重组机制,在算法原有的基础上使种群中差异性较大的个体进行维度重组,从而充分利用种群原有信息,极大地改善了算法的多样性;最后结合改进后的动态罚函数法高效处理实际约束问题.利用18个基准测试函数和29个工程实际约束优化问题验证HMSFA-PC的性能,综合实验结果表明,HMSFA-PC具有较好的全局寻优和局部勘探能力,种群的多样性也得到了极大提高,无论是在18个基本的测试函数上还是在29个工程实际约束优化问题上,HMSFA-PC相较于对比算法表现出更好的寻优能力和综合性能.然而,在求解问题的过程中发现,HMSFA-PC在处理多模态优化以及约束区间波动较大的问题上存在一定的缺陷,需进一步提升算法的寻优能力,并且时间复杂度相较于原算法有了一定程度的增加.未来将进一步优化算法并结合不同的策略处理高维实际约束优化问题.

参考文献(References)

- [1] 王蕊,顾清华.一种求解约束多目标问题的协作进化算法[J].控制与决策,2021,36(11):2656-2664.
(Wang R, Gu Q H. A collaborative evolutionary algorithm for solving constrained multi-objective problems[J]. Control and Decision, 2021, 36(11): 2656-2664.)
- [2] Yang X S. Nature-inspired metaheuristic algorithms[M]. Frome: Luniver Press, 2008.
- [3] 赵嘉,陈文平,肖人彬,等.面向多峰优化问题的自主学习萤火虫算法[J].控制与决策,2022,37(8):1971-1980.
(Zhao J, Chen W P, Xiao R B, et al. Firefly algorithm based on self-learning for multi-peak optimization problem[J]. Control and Decision, 2022, 37(8): 1971-1980.)
- [4] Yang X S. Multiobjective firefly algorithm for continuous optimization[J]. Engineering with Computers, 2013, 29(2): 175-184.
- [5] Tsai C W, Huang Y T, Chiang M C. A non-dominated sorting firefly algorithm for multi-objective optimization[C]. The 14th International Conference on Intelligent Systems Design and Applications. Okinawa, 2015: 62-67.
- [6] 谢承旺,张飞龙,陆建波,等.一种多策略协同的多目标萤火虫算法[J].电子学报,2019,47(11):2359-2367.
(Xie C W, Zhang F L, Lu J B, et al. Multi-objective firefly algorithm based on multiply cooperative strategies[J]. Acta Electronica Sinica, 2019, 47(11): 2359-2367.)
- [7] Lv L, Zhao J, Wang J Y, et al. Multi-objective firefly algorithm based on compensation factor and elite learning[J]. Future Generation Computer Systems, 2019, 91: 37-47.
- [8] 赵嘉,陈丹丹,肖人彬,等.一种基于最大最小策略和非均匀变异的萤火虫算法[J].智能系统学报,2022,17(1):116-130.
(Zhao J, Chen D D, Xiao R B, et al. A heterogeneous variation firefly algorithm with maximin strategy[J]. CAAI Transactions on Intelligent Systems, 2022, 17(1): 116-130.)
- [9] Karaboga D, Akay B. A modified artificial bee colony (ABC) algorithm for constrained optimization problems[J]. Applied Soft Computing, 2011, 11(3): 3021-3031.
- [10] Mirjalili S, Lewis A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.
- [11] Mirjalili S. SCA: A sine cosine algorithm for solving optimization problems[J]. Knowledge-Based Systems, 2016, 96: 120-133.
- [12] Hashim F A, Hussain K, Houssein E H, et al. Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems[J]. Applied Intelligence, 2021, 51(3): 1531-1551.
- [13] Yadav A. AEFA: Artificial electric field algorithm

- for global optimization[J]. *Swarm and Evolutionary Computation*, 2019, 48: 93-108.
- [14] Wolpert D H, Macready W G. No free lunch theorems for optimization[J]. *IEEE Transactions on Evolutionary Computation*, 1997, 1(1): 67-82.
- [15] 李智勇, 黄滔, 陈少淼, 等. 约束优化进化算法综述[J]. *软件学报*, 2017, 28(6): 1529-1546.
(Li Z Y, Huang T, Chen S M, et al. Overview of constrained optimization evolutionary algorithms[J]. *Journal of Software*, 2017, 28(6): 1529-1546.)
- [16] Saha C, Das S, Pal K, et al. A fuzzy rule-based penalty function approach for constrained evolutionary optimization[J]. *IEEE Transactions on Cybernetics*, 2016, 46(12): 2953-2965.
- [17] Tessema B, Yen G G. An adaptive penalty formulation for constrained evolutionary optimization[J]. *IEEE Transactions on Systems, Man, and Cybernetics*, 2009, 39(3): 565-578.
- [18] Kusakci A O, Can M. An adaptive penalty based covariance matrix adaptation-evolution strategy[J]. *Computers & Operations Research*, 2013, 40(10): 2398-2417.
- [19] Yang X S, Deb S. Cuckoo search via lévy flights[C]. *World Congress on Nature & Biologically Inspired Computing*. Coimbatore, 2010: 210-214.
- [20] 刘佛祥. 基于动态罚函数和多目标的人工蜂群算法求解约束优化问题的研究[D]. 南京: 南京师范大学, 2017.
(Liu F X. Research on artificial bee colony algorithm based on dynamic penalty function and multi-objective to solve constrained optimization problems[D]. Nanjing: Nanjing Normal University, 2017.)
- [21] Coello C A C, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2004, 8(3): 256-279.
- [22] Gadhvi B, Savsani V, Patel V. Multi-objective optimization of vehicle passive suspension system using NSGA-II, SPEA2 and PESA-II[J]. *Procedia Technology*, 2016, 23: 361-368.
- [23] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach—Part I: Solving problems with box constraints[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4): 577-601.
- [24] Qi Y T, Ma X L, Liu F, et al. MOEA/D with adaptive weight adjustment[J]. *Evolutionary Computation*, 2014, 22(2): 231-264.
- [25] Kumar A, Wu G H, Ali M Z, et al. A benchmark-suite of real-world constrained multi-objective optimization problems and some baseline results[J]. *Swarm and Evolutionary Computation*, 2021, 67: 100961.
- [26] Liu Z Z, Wang Y. Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces[J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(5): 870-884.
- [27] Zhou Y L, Zhu M, Wang J H, et al. Tri-goal evolution framework for constrained many-objective optimization[J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020, 50(8): 3086-3099.
- [28] Jain H, Deb K. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach—Part II: Handling constraints and extending to an adaptive approach[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4): 602-622.
- [29] Tian Y, Zhang T, Xiao J H, et al. A coevolutionary framework for constrained multi-objective optimization problems[J]. *IEEE Transactions on Evolutionary Computation*, 2021, 25(1): 102-116.
- [30] Tian Y, Cheng R, Zhang X Y, et al. An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility[J]. *IEEE Transactions on Evolutionary Computation*, 2018, 22(4): 609-622.
- [31] Liu Z Z, Wang Y, Huang P Q. And: A many-objective evolutionary algorithm with angle-based selection and shift-based density estimation[J]. *Information Sciences*, 2020, 509: 400-419.

作者简介

吕莉(1982—), 女, 教授, 硕士生导师, 从事智能计算、目标跟踪与检测、大数据挖掘等研究, E-mail: lvli623@163.com;

潘宁康(1998—), 男, 硕士生, 从事智能计算的研究, E-mail: pannk730@163.com;

肖人彬(1965—), 男, 教授, 博士生导师, 从事群集智能、复杂系统建模与分析等研究, E-mail: rbxiao@hust.edu.cn;

王晖(1982—), 男, 教授, 硕士生导师, 从事群智能算法、进化计算等研究, E-mail: huiwang@whu.edu.cn;

谭德坤(1973—), 男, 副教授, 硕士生导师, 从事群智能算法、机器学习等研究, E-mail: 57326905@qq.com.