

控制与决策

Control and Decision

面向多峰优化问题的自主学习萤火虫算法

赵嘉, 陈文平, 肖人彬, 王晖

引用本文:

赵嘉, 陈文平, 肖人彬, 王晖. 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.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2020.1812>

您可能感兴趣的其他文章

Articles you may be interested in

[基于改进量子粒子群的K-means聚类算法及其应用](#)

K-means clustering algorithm based on improved quantum particle swarm optimization and its application

控制与决策. 2022, 37(4): 839–850 <https://doi.org/10.13195/j.kzyjc.2020.1302>

[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R2指标和目标空间分解的高维多目标粒子群优化算法](#)

R2 indicator and objective space partition based many-objective particle swarm optimizer

控制与决策. 2021, 36(9): 2085–2094 <https://doi.org/10.13195/j.kzyjc.2020.0113>

[基于局部搜索的反向学习竞争粒子群优化算法](#)

Opposition-based learning competitive particle swarm optimizer with local search

控制与决策. 2021, 36(4): 779–789 <https://doi.org/10.13195/j.kzyjc.2019.1150>

[基于仿生算法改进粒子滤波的SLAM算法精度预测](#)

Accuracy prediction of SLAM algorithm based on bionic algorithm to improve particle filter

控制与决策. 2021, 36(1): 166–172 <https://doi.org/10.13195/j.kzyjc.2019.0555>

面向多峰优化问题的自主学习萤火虫算法

赵嘉^{1,2}, 陈文平¹, 肖人彬^{3†}, 王晖^{1,2}

(1. 南昌工程学院 信息工程学院, 南昌 330099; 2. 南昌工程学院 江西省水信息协同感知与智能处理重点实验室, 南昌 330099; 3. 华中科技大学 人工智能与自动化学院, 武汉 430074)

摘要: 萤火虫算法在处理多峰优化问题时易陷入局部最优, 针对该问题提出一种自主学习萤火虫算法. 该算法将粒子按适应度划分为自主学习粒子和普通粒子, 自主学习粒子从种群中随机选择一个粒子并随机选择一个维度, 使用 3 种学习策略产生 3 个候选解, 在自身以及候选解中选择最好的解; 普通粒子同时选择两个优于自身的粒子进行学习. 自主学习粒子能够维持算法对多个极值空间的探索并提高算法优化精度; 普通粒子以两个粒子的混合信息为指引, 使算法跳出局部最优. 此外, 使用淘汰机制, 让算法舍弃对劣质极值空间的维护, 进而提高对优质极值空间的开发, 实验结果表明, 所提出算法在处理多峰优化问题时具有高效的性能.

关键词: 萤火虫算法; 多峰优化问题; 自主学习; 双样本学习; 淘汰机制; 局部最优

中图分类号: TP18

文献标志码: A

DOI: 10.13195/j.kzyjc.2020.1812

开放科学(资源服务)标识码(OSID):



引用格式: 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.

Firefly algorithm based on self-learning for multi-peak optimization problem

ZHAO Jia^{1,2}, CHEN Wen-ping¹, XIAO Ren-bin^{3†}, WANG Hui^{1,2}

(1. School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China; 2. Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing, Nanchang Institute of Technology, Nanchang 330099, China; 3. School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: The firefly algorithm is prone to fall into local optimality when dealing with multi-peak optimization problems. To solve this problem, this paper proposes a firefly algorithm based on self-learning. According to the fitness, the particles are divided into self-learning particles and ordinary particles. The self-learning particles randomly select a particle from the population and randomly select a dimension to generate three candidate solutions by using three learning strategies, and the best solution among itself and the candidate solutions is obtained. Ordinary particles simultaneously choose two particles superior to themselves to learn. Self-learning particles can not only maintain the algorithm's exploration of multiple extremum spaces, but also improve the algorithm's optimization accuracy. Guided by the mixed information of two particles, ordinary particles make the algorithm jump out of the local optimal. In addition, the elimination mechanism is used to make the algorithm abandon the maintenance of low-quality extremum space, so as to improve the development of high-quality extremum space. Experimental results show that the proposed algorithm has high performance in multi-peak optimization problems.

Keywords: firefly algorithm; multi-peak optimization problems; self-learning; two-sample learning; elimination mechanism; local optimal

0 引言

在科学和工程领域中存在许多多峰优化问题, 该类问题定义域内有多个极值点, 传统的如基

于导数和基于梯度的优化方法大多根据目标函数的局部可展性确定每一步的搜索方向^[1], 仅能保证算法的局部收敛, 易陷入局部最优, 难以得到函数的

收稿日期: 2020-12-25; 录用日期: 2021-04-21.

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 (52069014, 51669014); 江西省杰出青年基金项目 (2018ACB21029).

责任编辑: 陈家伟.

† 通讯作者. E-mail: rbxiao@hust.edu.cn.

全局最优点. 多峰优化问题存在多极值的复杂特性, 对优化算法的性能提出了更高的要求. 群智能算法 (swarm intelligence algorithm, SIA)^[2] 具有自组织、自适应以及自学习等特性, 在求解多峰优化问题时弱化了问题的具体场景, 具有很强的鲁棒性. 因此, 利用 SIA 求解多峰优化问题成为优化领域的研究热点. 常见的 SIA 有: 粒子群优化算法 (particle swarm optimization, PSO)^[3]、人工蜂群算法 (artificial bee colony algorithm, ABC)^[4]、蝙蝠算法 (bat algorithm, BA)^[5] 等.

萤火虫算法 (firefly algorithm, FA)^[6-8] 是 SIA 家族中的一员, 该算法由 Yang^[7] 在 2008 年提出, 其灵感源于自然界中萤火虫发光吸引配偶的行为. 算法将随机生成的解视为萤火虫, 每个解根据它们在目标函数中的表现分配一个亮度, 每只萤火虫会被比自身更亮的萤火虫吸引, 算法通过萤火虫之间不断地吸引运动达到寻优的目的.

虽然 FA 思想简单、易于实现且优化性能高, 但其学习方式会导致种群中最优萤火虫的领导力过于强大, 所有萤火虫会迅速聚集在最优萤火虫附近. 一旦最优粒子陷入局部最优, 所有粒子会快速聚集在一起, 形成“群集现象”, 导致种群陷入局部最优且难以逃逸. 为了提高 FA 的优化性能, 学者们对 FA 的学习策略进行了改进. Zhou 等^[9] 将原有的向所有优于自身的萤火虫学习改为向部分优于自身的萤火虫学习, 削弱了最优萤火虫的领导力, 同时降低了算法的时间复杂度; 周凌云等^[10] 使用正交实验设计方法以萤火虫的位置信息构造指导向量, 指导最优萤火虫飞向最优区域, 从而引导整个种群快速准确地向全局最优解收敛; 赵嘉等^[11] 在萤火虫使用 FA 的学习机制后, 构造广义中心粒子并令其进行深度学习, 然后让广义中心粒子引导种群进化; Wang 等^[12] 受中国古代“阴阳学说”思想的启发, 选择最优萤火虫作为“阳萤火虫”, 通过“阳萤火虫”产生“阴萤火虫”, 然后“阴阳萤火虫”通过调节达到阴阳平衡, 避免种群进化陷入困境.

上述改进算法虽然能够使萤火虫种群以更高效的方式向更优异的空间进行搜索, 有效提高算法性能, 但是在面对多峰优化问题时, 算法使用的学习策略仍然没有摆脱“群集现象”, 萤火虫种群在某一萤火虫的引导下聚集在一起, 这不利于算法对多极值空间的搜索. 面对多峰优化问题, 算法既要保持群体内个体之间的相对独立性, 以维持对多个极值空间的广泛搜索, 避免种群陷入局部最优; 又要保证个体自身

的学习能力, 进而保证对优质极值空间的深度开发, 提高算法优化精度.

基于上述分析, 本文为了求解多峰优化问题, 提出一种自主学习萤火虫算法 (firefly algorithm based on self-learning, SLFA). 算法将萤火虫按适应度排序并划分为自主学习粒子和普通粒子. 自主学习粒子从种群中随机选择一个粒子并随机选择一个维度, 使用 3 种学习策略产生 3 个候选解; 然后从自身以及候选解中自主选择适应度最好的解, 普通粒子同时选择两个比自身亮的粒子进行学习. 在种群中, 自主学习粒子具有自主能力, 其地位是相同的, 不存在领导和被领导的关系, 所以可以看作是一个个相对独立的个体, 这种独立性有利于种群对多极值空间的探索. 此外, 自主学习粒子的适应度好, 更具学习潜力, 使用自主学习策略能够提高粒子自身的学习能力, 加强对极值点的开发, 保证算法的优化精度. 普通粒子以两个优秀粒子的混合信息为指引, 有更大的概率探索到当前种群未探索到的区域, 使算法跳出局部最优, 进一步提高种群对多极值空间的搜索能力. 最后, 使用淘汰机制, 当普通粒子的适应度优于自主学习粒子, 则代表普通粒子找到了更优质的极值空间, 令普通粒子取代适应度最差的自主学习粒子, 使算法舍弃劣质极值空间, 转而开发更优质的极值空间. 为了验证所提出算法的优化性能, 使用两组测试函数将所提出算法与一些经典以及最新的算法进行比较. 实验结果表明, SLFA 能够高效处理多峰优化问题, 且对单峰问题也具有较好的优化性能.

1 萤火虫算法

设 D 为搜索空间的维度, 每只萤火虫在搜索空间中的位置 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ 代表一个可行解, 其对应优化问题的适应值为 $f(X_i)$. 对于最小化问题, 适应值越小则适应度 fit_i 越大, 此时适应度与适应值成反比 $fit_i \propto 1/f(X_i)$; 对于最大化问题, 适应值越大则适应度越大, 此时适应度与适应值成正比 $fit_i \propto f(X_i)$. 以适应度衡量萤火虫 i 的亮度 (解的质量), 适应度越大代表解的质量越好. 萤火虫向比自身更亮的萤火虫移动, 定义萤火虫 i 被更亮的萤火虫 j 吸引的位置更新公式为

$$x_{id}(t+1) = x_{id}(t) + \beta(x_{jd}(t) - x_{id}(t)) + \alpha\varepsilon. \quad (1)$$

其中: x_{id} 和 x_{jd} 分别为萤火虫 i 、 j 的 d 维位置; β 为吸引力; α 为步长因子, 取值为 0.2; t 为当前迭代次数; ε 为 $[-0.5, 0.5]$ 上服从均匀分布的随机数.

萤火虫的亮度和吸引力通过以下方式计算:

$$I = I_0 e^{-\gamma r_{ij}^2}, \quad (2)$$

$$\beta = \beta_0 e^{-\gamma r_{ij}^2}. \quad (3)$$

其中: β_0 为萤火虫的吸引力系数, 通常取值为 1; γ 为光吸收系数, 一般设为常数 1; r_{ij} 为萤火虫个体 i 与 j 之间的欧氏距离, 由下式决定:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{d=1}^D (x_{id} - x_{jd})^2}. \quad (4)$$

2 自主学习萤火虫算法

FA 中, 每个粒子要向所有优于自身的粒子学习, 假设粒子数量为 N , 优化问题的维度为 D , 对于给定的问题 f , 求其函数值的时间复杂度为 $O(f)$, 则种群进化一代的时间复杂度为 $O(N^2(D+f))$. 可以看出, FA 的时间复杂度较高, 并且每进化一代每个粒子要运动多次, 过多的运动会产生震荡, 浪费计算资源. 更严重的是, 这将导致最优粒子具有强大的吸引力, 整个种群受到最优粒子的领导迅速聚集到其附近. 这种“群集现象”不利于算法对多极值空间的搜索, 一旦最优粒子陷入局部最优, 整个种群也将陷入局部最优且难以逃逸.

SLFA 中, 将粒子划分为自主学习粒子和普通粒子, 每个自主学习粒子需要以不同的策略学习 3 次, 普通粒子只需要学习 1 次. 假设自主学习粒子的个数为 K , 则种群进化一代的时间复杂度为 $O((N+2K)(D+f))$, 其值远小于 FA, 并且每个粒子只需要运动一次, 杜绝了震荡现象的发生. 此外, 针对多峰优化问题, 算法需要兼顾探索与开发的平衡, 既要保证探索尽可能多的极值点, 避免算法陷入局部最优, 又要保证对极值点的开发, 提高算法优化精度. 这就要求粒子相对独立自主, 且具有较强的学习能力. 本文提出的 SLFA 在处理多峰优化问题时具有如下优势:

1) 自主学习粒子拥有自主学习能力, 它们之间相对独立, 不会被单一粒子吸引而聚集在一起. 因此, 算法能同时对多个极值空间进行开发.

2) 自主学习粒子学习能力强, 能够对极值空间进行深层次开发, 保证算法的优化精度.

3) 普通粒子以两个粒子的混合信息为指引, 有更大的概率探索到当前种群未探索到的区域, 使算法跳出局部最优, 进一步提高种群对多极值空间的搜索能力.

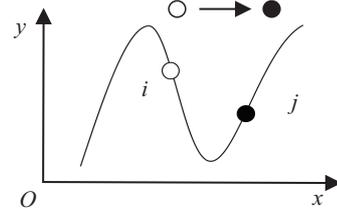
4) 使用淘汰机制, 当普通粒子的适应度优于自主学习粒子时, 令普通粒子取代适应度最差的自主学习粒子. 此时有两种情况: 若普通粒子陷入已找到的优质极值点, 则舍弃劣质极值点, 加强对优质极值点的

开发, 提高算法优化精度; 若普通粒子探索到未知的优质极值点, 则舍弃劣质极值点, 维护新的优质极值点.

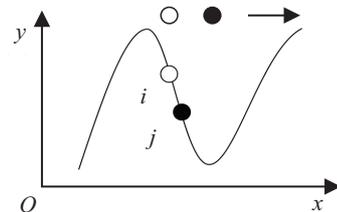
2.1 自主学习

萤火虫算法使用整体维度更新策略, 粒子进行位置更新时, 每一个维度的值都需要更新, 这种更新方式将导致粒子在学习过程中虽然提高了适应度, 但是某些维度的重要位置信息会缺失. 在自主学习粒子的学习过程中, 本文摒弃传统的整体维度更新形式, 采用随机选择一个维度的方式进行更新. 基于粒子在空间中的分布情况提出了相应的更新策略.

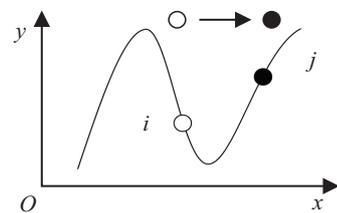
若学习粒子 i 与其学习对象 j 在同一极值区域, 则 i 与 j 以及它们所处区域极值点的分布情况如图 1 所示的 4 种分布: 1) 学习粒子 i 与优质学习对象异侧, 萤火虫 i 向适应度更好的 j 学习, 并且两只萤火虫处在极值点的两侧, 此时 i 应该向 j 靠近, 如图 1(a) 所示; 2) 学习粒子 i 与优质对象同侧, 虽然 i 向 j 学习能改善自身位置, 但是无法改善种群目前寻到的最优位



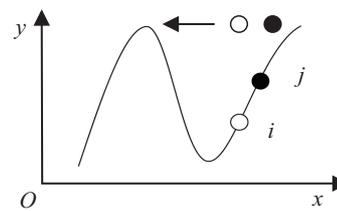
(a) 与优质对象异侧



(b) 与优质对象同侧



(c) 与劣质对象异侧



(d) 与劣质对象同侧

图 1 粒子的分布情况

置,此时更高效的学习方式应该是*i*在*j*的位置基础上沿着*i*指向*j*的方向继续探索,如图1(b)所示;3)学习粒子*i*与劣质对象异侧,虽然*j*的适应值劣于*i*,但是*i*向*j*学习也能寻到更好的位置,此时*i*应该向*j*靠近,如图1(c)所示;4)学习粒子*i*与劣质对象同侧,此时*i*应该远离*j*,如图1(d)所示。

若学习粒子*i*与其学习对象*j*之间存在多个峰值,则其分布也可以抽象为图1所示的4种情况。此时图中的极值点可能是*i*所在区域的极值点、*j*所在区域的极值点或*i*与*j*之外未知但更优质的极值点。因此,使用图中所示的运动方式可以加强算法对*i*或*j*所处极值区域的开发,或者发掘更优质的极值区域。

通过上述分析可知,虽然粒子的位置关系有4种,但其运动方式只有3类。针对图1(a)和图1(c)所示的粒子分布,不论粒子*i*的适应度是否劣于粒子*j*,*i*都应该向*j*靠近,其学习策略为正向靠近;针对图1(b)所示的粒子分布,粒子*i*应跃过其学习对象*j*,方向为由*i*指向*j*,其学习策略为正向跃迁;针对图1(d)所示的粒子分布,粒子*i*应远离其学习对象*j*,学习策略为反向逃逸。3类运动方式对应的学习策略定义如下。

定义1 正向靠近为

$$x_{opt1} = x_{id} + r_1(x_{jd} - x_{id}). \quad (5)$$

定义2 正向跃迁为

$$x_{opt2} = x_{jd} + r_2(x_{jd} - x_{id}). \quad (6)$$

定义3 反向逃逸为

$$x_{opt3} = x_{id} - r_3(x_{jd} - x_{id}). \quad (7)$$

定义中,*j*为从种群中随机选取的一个粒子,*d*为随机选择的一个维度, r_1 、 r_2 和 r_3 为(0,1)内服从均匀分布的随机数。

粒子学习完毕后,从3种学习策略产生的候选解以及自身解中自主选择适应度最好的解,以最小化优化问题为例,其自主选择公式如下:

$$f_i^{t+1} = \min\{f_i^t, f_{opt1}, f_{opt2}, f_{opt3}\}. \quad (8)$$

其中: f_{opt1} 、 f_{opt2} 、 f_{opt3} 分别为萤火虫*i*使用正向靠近、正向跃迁以及反向逃逸产生候选解的适应值, f_i^t 为第*t*代时萤火虫*i*的适应值。

2.2 双样本学习

在标准萤火虫算法中,萤火虫每次只选择一个样本进行学习,使得种群很难逃脱已经陷入的局部最优。如图2所示,搜索空间中有3个局部极值区域,其中阴影处为全局最优,萤火虫*j*和*k*已经陷入局部极值所处的区域,那么萤火虫*i*无论是向*j*学习还是向*k*学习都会陷入种群已有的困境中。为了使*i*能够探索

到全局最优区域,令*i*同时向*j*和*k*学习,这样便能探测到优化空间内种群未探测到的区域。

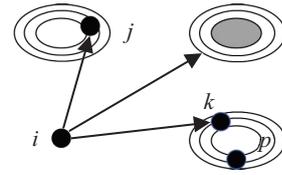


图2 双样本学习

萤火虫选择双样本学习如下所示:

$$x_{id}(t+1) = x_{id}(t) + r_1\beta_1(x_{jd}(t) - x_{id}(t)) + r_2\beta_2(x_{kd}(t) - x_{id}(t)) + \alpha\varepsilon. \quad (9)$$

其中:*j*和*k*为从种群中随机选择的适应值优于*i*的粒子, r_1 和 r_2 为(0,1)内服从均匀分布的随机数,为了避免图2萤火虫*k*和*p*处于同一个极值点附近时*i*出现过度学习的情况,令 $r_1 + r_2 = 1$ 。

2.3 算法流程

SLFA 首先按适应度排序将萤火虫种群划分为自主学习粒子和普通粒子;然后,两种粒子使用各自的进化策略进行进化;最后将进化后的萤火虫种群进行排序并重新划分两种粒子,令优质的普通粒子取代劣质的自主学习粒子。SLFA 具体算法流程如下所示。

The proposed SLFA

- 1: initialize a population of fireflies randomly
- 2: calculate the fitness values of each firefly
- 3: FEs = $N(N$ is firefly population)
- 4: while (FEs < MAX_FEs)
- 5: sort the particles and divide them into two parts
- 6: for $i = 1$ to N
- 7: if i is a self-learning particle
- 8: move firefly i according to 式(8)
- 9: FEs = FEs + 1
- 10: else
- 11: move firefly i according to 式(9)
- 12: FEs = FEs + 1
- 13: end if
- 14: end for
- 15: end while

3 算法策略分析

3.1 测试函数

采用优化领域内常用的一组测试函数^[11,13-14]对SLFA 策略进行分析,其由12个函数组成, $f_1 \sim f_7$ 为单峰函数,只有一个极值点,用于测试算法的局部开

发精度; $f_8 \sim f_{12}$ 为多峰函数, 在定义域内有多个极值点, 用于测试算法的全局探索能力。

3.2 萤火虫划分策略选择

SLFA 首先需要将萤火虫按适应度排序并划分为自主学习粒子和普通粒子, 然后根据划分结果对粒子使用相应的学习策略。为验证划分策略对算法优化性能的影响, 提出了3种划分策略并进行实验:

1) 固定比例划分。假设萤火虫种群规模为 N , P 为自主学习粒子在种群中的占比。进行排序后, 种群中前 NP 个粒子进行自主学习, 剩下的 $N(1 - P)$ 个粒子选择双样本学习。对于固定分配的策略, 令自主学习粒子占比 $P = 10\%, 20\%, \dots, 90\%$ 。

2) 自适应划分。以所有萤火虫的平均适应值作为界限自适应划分两种粒子, 适应值小于平均适应值的粒子划分为自主学习粒子; 适应值不小于平均适应值的粒子划分为普通粒子。

3) 动态划分。一般而言, 在算法运行前期算法的搜索策略应该更偏重于全局搜索, 在后期更偏向于局部搜索。在 SLFA 算法中, 自主学习粒子局部搜索能力强, 普通粒子全局搜索能力强。因此, 假设 K 为自主学习粒子个数, 则 K 应该随算法评估次数的增加而增加, 并且由于普通粒子要从优于自身的粒子中随机选择两个粒子进行学习, 至少应该存在两个自主学习粒子, K 的最小值为2, 其可以定义为

$$k = 2 + \lfloor (N - 2)(\text{FEs}/\text{MAX_FEs}) \rfloor. \quad (10)$$

其中: FEs 和 MAX_FEs 分别为算法当前评估次数和最大评估次数, K 向下取整。

通过实验从3种划分策略中选择使算法优化性能最大化的策略, 为公平比较, 设置相同的参数,

种群规模 $N = 20$, 维度 $D = 30$, 最大评估次数 $\text{MAX_FEs} = 5 \times 10^5$, 参数 $\beta_0 = 1, \gamma = 1/\Gamma$ (Γ 为优化函数的域长度)。此外, 步长因子 α 按文献[14]提出的动态方法设置, 令

$$\alpha_0 = 0.2, \alpha(t + 1) = \alpha(t)\Gamma e^{-\text{FEs}/(\text{MAX_FEs}70)}.$$

算法运行30次, 均值结果如表1所示, 最好的优化结果用粗体表示, 表中最后一行为对各算法进行 Friedman 检验的结果, 值越小表明算法的综合优化越好。由表1可见, 在函数 f_6 和 f_{12} 上, 所有划分策略都得到相同的优化结果, 在函数 f_5, f_7, f_{10} 和 f_{11} 上, 所有划分策略都得到相近的优化结果, 说明在这6个函数上, 算法对划分策略的改变不敏感; 在函数 f_1, f_2 和 f_8 上, 固定比例较高时 ($P \geq 60\%$), 算法使用固定比例划分策略的性能优于自适应划分策略和动态划分策略; 函数 f_3 和 f_4 上, 固定比例较低时 (分别为 $P \leq 30\%$ 和 $P \leq 60\%$), 使用固定比例划分策略的性能优于自适应划分策略和动态划分策略; 在函数 f_9 上, 使用 $P \geq 30\%$ 的固定比例划分策略和自适应划分策略都能使算法寻到全局最优值。从 Friedman 检验结果可以看出, 当 $P \geq 50\%$ 时, 使用固定比例划分策略的性能优于自适应划分策略, 当 $P \geq 10\%$ 时, 使用固定比例划分策略的性能优于动态划分策略。使用 $P = 60\%$ 时的固定划分策略优化性能最佳, 因此在后续实验中, 萤火虫划分策略选择 $P = 60\%$ 时的固定划分策略。

3.3 算法运动过程比较

为验证 SLFA 能有效处理多峰优化问题, 选取多峰函数 f_9 , 在维度 $D = 2$ 的情况下, 比较 SLFA 和 FA 的种群运动过程, 如图3所示。

表1 使用不同划分策略时算法优化的均值结果

	固定比例划分 (P)									自适应划分	动态划分
	10%	20%	30%	40%	50%	60%	70%	80%	90%		
f_1	1.91e-126	1.01e-108	1.05e-97	8.02e-115	3.03e-138	1.02e-155	1.44e-168	1.89e-182	1.49e-188	2.21e-154	3.16e-106
f_2	4.18e-56	1.76e-55	3.67e-50	1.18e-60	9.26e-71	5.37e-78	3.81e-86	5.44e-92	2.69e-97	2.26e-76	6.97e-56
f_3	4.56e-24	8.63e-15	1.40e-09	9.20e-06	3.90e-03	7.11e-02	1.05e+00	6.52e+00	3.22e+01	6.46e-01	2.97e-08
f_4	2.40e-22	1.98e-32	1.80e-41	1.08e-37	6.05e-30	2.01e-20	7.36e-13	5.50e-08	1.02e-03	1.30e-15	4.74e-19
f_5	1.99e+01	3.92e+00	4.21e+00	1.45e+00	6.15e-01	1.29e+00	9.96e-01	8.09e-01	8.77e-01	7.24e-01	8.33e+00
f_6	0.00e+00	0.00e+00	0.00e+00								
f_7	4.23e-03	2.20e-03	1.62e-03	1.11e-03	1.03e-03	9.98e-04	1.01e-03	1.02e-03	1.01e-03	9.73e-04	2.81e-03
f_8	-11510.1	-12182.6	-12490.5	-12553.7	-12565.5	-12569.5	-12569.5	-12569.5	-12569.5	-12565.5	-11492.4
f_9	1.33e-01	3.32e-02	0.00e+00	0.00e+00	1.33e-01						
f_{10}	1.04e-14	6.51e-15	4.50e-15	4.50e-15	4.26e-15	4.14e-15	4.26e-15	4.73e-15	6.39e-15	1.03e-14	4.38e-15
f_{11}	1.33e-02	7.20e-03	6.31e-03	3.32e-03	1.72e-03	2.21e-03	3.29e-04	2.47e-04	4.11e-04	1.15e-03	4.92e-03
f_{12}	1.57e-32	1.57e-32	1.57e-32								
rank	8.17(11)	7.50(9)	6.83(8)	6.00(7)	4.88(3)	4.58(1)	4.67(2)	4.92(4)	5.13(5)	5.42(6)	7.92(10)

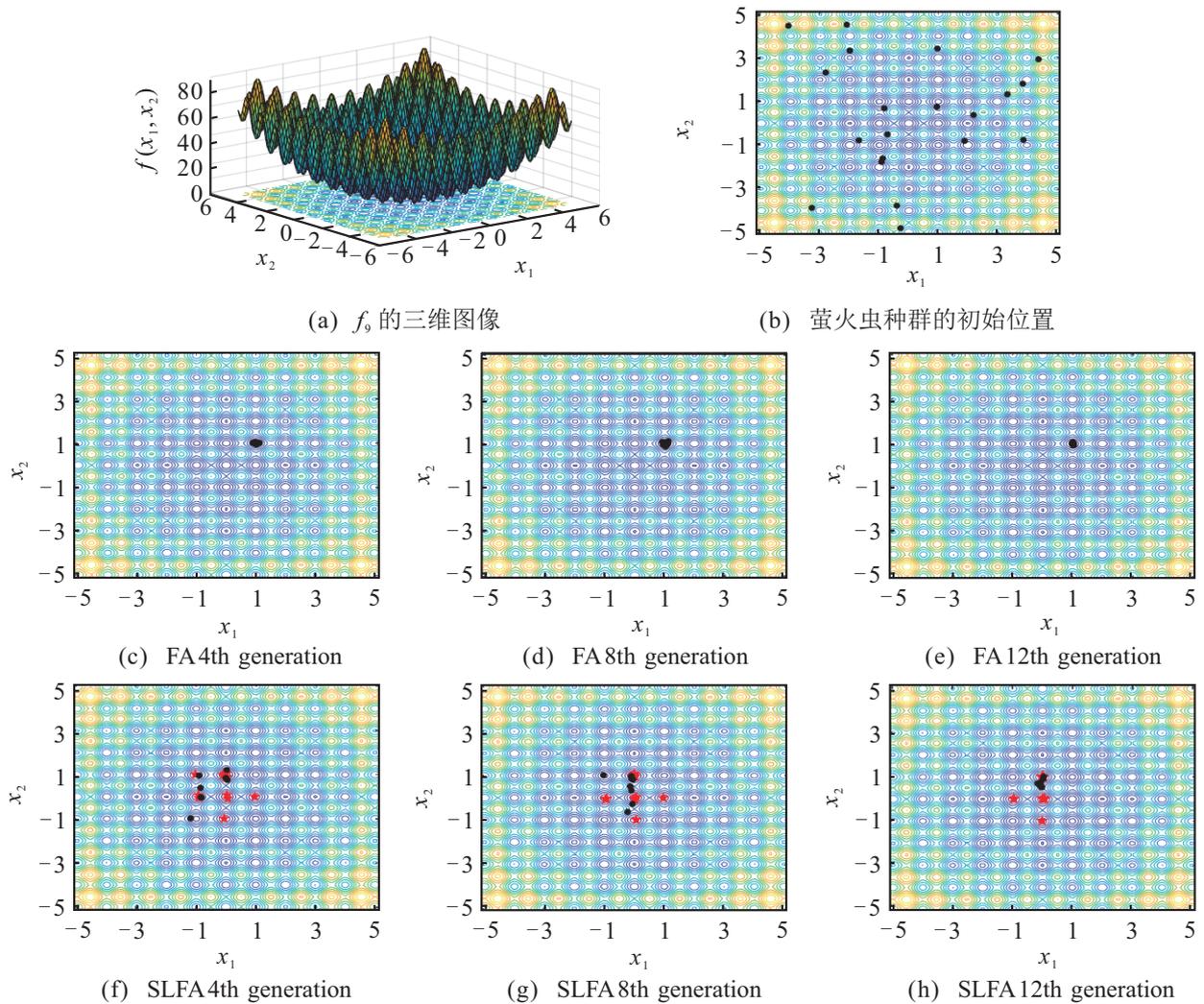


图3 两种算法运动过程比较

图3(a)为 f_9 的三维图像,可以看出,该函数是个复杂的多峰函数,优化空间内有非常多的极值点.图3(b)的黑点为萤火虫种群的初始位置,全局最优点在 $(0,0)$ 处.

图3(c)~图3(e)分别为使用FA迭代4、8、12次时的种群位置,可以看出,种群迭代4次时便陷入了局部最优,且随着迭代的进行,种群的“群集现象”越来越明显,算法失去跳出局部最优的能力.算法迭代50次的优化结果为1.990.

图3(f)~图3(h)分别为使用SLFA迭代4、8、12次时的种群位置,其中五角星为自主学习粒子,圆点为普通粒子.可以看出,种群迭代4次时,自主学习粒子保存有6个极值区域,其中包括全局最优所在区域.迭代8次时,最劣质极值区域的自主学习粒子消失,而全局最优点所在区域的自主学习粒子增多,算法舍弃劣质极值区域而加强优质极值区域的开发,验证了淘汰机制的有效性.迭代12次时,算法又进一步舍弃了一个劣质极值区域,全局最优点所在区域的自主学习粒子进一步增多.算法迭代50次的优化结果

为 9.415×10^{-13} .

由上述分析可知,SLFA在处理多峰优化问题时的性能达到了算法所用策略的预期效果.而在优化效率上,同样是迭代50次,FA的评估次数为 $50 \times 20 \times 20 = 20\,000$ 次. SLFA的评估次数为 $50 \times (12 \times 3 + 8) = 2\,200$ 次,远低于FA,但其优化结果远优于FA,验证了自主学习策略的高效性.

4 算法比较

4.1 与FA及其改进算法比较

使用上一节的12个测试函数,在维度 $D = 30$ 的条件下,将SLFA与FA^[2]、WSSFA^[15-16]、MFA^[17]、RaFA^[18]、ApFA^[19]、DLFA^[11]和LVFA^[14]进行比较.

SLFA的参数与上一节一致,各比较算法的其他参数与相应参考文献一致.为公平比较,参照RaFA、ApFA、DLFA、LVFA的实验设置,设置算法终止条件为最大评估次数MAX_FEs= 5×10^5 ,萤火虫种群数量 $N = 20$,各算法运行30次,寻优结果的平均值与标准差如表2所示.

表2 9种算法在12个测试函数上的优化均值结果

		FA	WSSFA	VSSFA	MFA	RaFA	ApFA	DLFA	LVFA	SLFA
f_1	mean	6.67e+04	6.34e+04	5.84e+04	1.56e-05	5.36e-184	2.02e-44	0.00e+00	7.65e-204	1.02e-155
	std	1.83e+04	4.91e+04	1.17e+04	2.31e-05	6.82e-184	2.85e-44	0.00e+00	0.00e+00	1.38e-154
f_2	mean	5.19e+02	1.35e+02	1.13e+02	1.85e-03	8.76e-05	1.83e-12	0.00e+00	1.24e-102	5.37e-78
	std	1.42e+02	5.66e+02	3.93e+01	3.57e-03	7.58e-05	4.01e-12	0.00e+00	6.24e-103	1.50e-76
f_3	mean	2.43e+05	1.10e+05	1.16e+05	5.89e-05	4.91e+02	1.01e+01	1.58e-09	1.53e-21	7.11e-02
	std	4.85e+04	4.60e+05	3.64e+04	4.52e-05	1.06e+02	5.92e+00	1.10e-08	4.51e-20	5.12e-01
f_4	mean	8.35e+01	7.59e+01	8.18e+01	1.73e-03	2.43e+00	1.30e-07	0.00e+00	1.00e-96	2.01e-20
	std	3.16e+01	1.88e+01	2.32e+01	3.86e-03	1.87e+00	8.85e-08	0.00e+00	2.70e-95	4.20e-19
f_5	mean	2.69e+08	2.49e+08	2.16e+08	2.29e+01	2.92e+01	2.81e+01	1.99e+00	3.82e+01	1.29e+00
	std	6.21e+07	2.76e+08	5.79e+07	3.25e+00	4.19e+00	3.98e-01	1.62e+01	3.61e+02	1.01e+01
f_6	mean	7.69e+04	6.18e+04	5.48e+04	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
	std	3.38e+03	5.12e+04	2.16e+03	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
f_7	mean	5.16e+01	3.24e-01	4.43e+01	1.30e-01	5.47e-02	2.76e-03	1.26e-02	1.96e-03	9.98e-04
	std	2.46e+01	4.80e+01	1.72e+01	2.36e-01	4.65e-02	1.10e-02	1.86e-02	4.54e-03	2.57e-03
f_8	mean	-1.56e+03	-2.01e+03	-1.85e+03	-7.63e+03	-41.21e+04	-6.25e+03	-8.86e+03	-7.38e+03	-1.26e+04
	std	3.77e+03	3.23e+03	8.56e+02	8.72e+02	3.61e+02	1.26e+02	2.04e+04	4.48e+03	9.96e-12
f_9	mean	3.33e+02	3.61e+02	3.12e+02	6.47e+02	2.69e+01	1.21e+01	0.00e+00	3.19e+01	0.00e+00
	std	6.28e+01	8.56e+01	4.18e+01	2.53e+01	1.52e+01	2.77e+00	0.00e+00	5.07e+01	0.00e+00
f_{10}	mean	2.03e+01	2.05e+01	2.03e+01	4.23e-04	3.61e-14	2.55e-14	5.68e-15	5.09e-15	4.14e-15
	std	2.23e-01	5.56e-01	2.46e-01	3.35e-04	5.98e-14	6.48e-15	9.64e-15	8.61e-15	3.49e-15
f_{11}	mean	6.54e+02	6.09e+02	5.47e+02	9.86e-03	0.00e+00	3.33e-16	2.99e-02	2.38e-03	2.21e-03
	std	1.69e+02	4.19e+02	1.29e+02	6.81e-03	0.00e+00	2.22e-16	1.52e-01	2.45e-02	3.50e-02
f_{12}	mean	7.16e+08	6.18e+08	3.99e+08	5.04e-08	4.50e-05	1.23e-16	1.57e-32	1.57e-32	1.57e-32
	std	1.82e+08	8.38e+08	1.05e+08	3.27e-08	6.28e-04	1.59e-16	1.50e-47	1.49e-47	1.50e-47
$w/l/t$		12/0/0	12/0/0	12/0/0	10/1/1	9/1/2	10/1/1	5/3/4	6/2/4	

表2最后一行 $w/l/t$ 表示与该算法相比, SLFA 在 w 个函数上性能更优, 在 l 个函数上性能相当, 在 t 个函数上性能更差. 由表2可见: SLFA 在所有测试函数上的优化结果均优于 FA、WSSFA、VSSFA; 与 MFA 相比, SLFA 在 10 个函数上性能更优, 在 f_6 上寻优结果相同, 在 f_3 上较差; 与 RaFA 相比, SLFA 在 9 个函数上性能更优, 在 f_6 上性能相当, 在 f_1 和 f_{11} 上性能更差; SLFA 在 10 个函数上优于 ApFA, 在 f_6 上相当, 均寻到了全局最优, 在 f_{11} 上处于劣势; 与 DLFA 和 LVFA 相比, SLFA 的性能在 $f_1 \sim f_4$ 这 4 个单峰函数上处于劣势, 在单峰函数 f_5 和 f_7 以及多峰函数 f_8 、 f_{10} 和 f_{11} 上胜出, 在 f_6 和 f_{12} 上相当. 综合比较, 与其他算法相比, SLFA 在多峰函数上的优势比在单峰函数上的优势更明显, 表明 SLFA 更善于处理多峰优化问题. 由表3结果可以看出, SLFA 算法在单峰函数上性能较差, 在 9 种算法中排第3, 排名低于 DLFA 和

LVFA, 在多峰函数上性能最优. 在所有测试函数的综合性能上, SLFA 的秩均值最小, 表明 SLFA 的综合优化性能最好.

表3 9种算法在12个测试函数上的Friedman检验结果

algorithm	rank		
	single-peak	multi-peak	all
	$f_1 \sim f_7$	$f_8 \sim f_{12}$	$f_1 \sim f_{12}$
FA	9.00(9)	8.30(9)	8.71(9)
WSSFA	7.57(8)	8.00(8)	7.75(8)
VSSFA	7.43(7)	7.10(7)	7.29(7)
MFA	4.64(5)	6.00(6)	5.21(6)
RaFA	4.79(6)	3.60(4)	4.29(5)
ApFA	4.07(4)	3.80(5)	3.96(4)
DLFA	2.07(1)	2.90(2)	2.42(2)
LVFA	2.64(2)	3.60(3)	3.04(3)
SLFA	2.79(3)	1.70(1)	2.33(1)

图4给出了9种算法在单峰函数 f_3 、 f_7 和多峰函数 f_8 、 f_{12} 上的收敛过程曲线. 由图4可见, SLFA 在函数 f_3 上收敛速度较慢, 这是由于在单峰优化问题的简单环境中, 自主学习粒子的自主性优势没有得到发挥, 各自主学习粒子各自为政, 其收敛速度慢于围绕最优粒子进行探索的其他算法. 同样是单峰优化问题, 在函数 f_7 上, 虽然 SLFA 前期收敛速度较慢, 但当其他算法停止进化时, SLFA 依然保持较快的收敛速度, 在后期超越其他算法, 达到更高的优化精度, 进一步表明了算法策略的有效性. 而对于多峰优化问题,

在函数 f_8 、 f_{12} 上, SLFA 无论是在收敛精度和收敛速度上都拥有较明显的优势, 表明了本文算法在多峰优化问题上的有效性.

此外, 使用自主学习策略的粒子从多个候选解中选择最优解, 虽然看似浪费了较多的计算资源, 但是这种做法能够极大地提高粒子的学习能力. 从收敛曲线上看, 本文算法在收敛速度上也具有优势, 进一步验证了自主学习的高效性.

4.2 与相关算法比较

利用优化难度较大的 CEC2013 测试函数^[20], 将 SLFA 与一些经典和最新的算法进行比较. CEC2013 测试函数由 28 个函数组成, 其中 $f_1 \sim f_5$ 为单峰函数, $f_6 \sim f_{20}$ 为多峰函数, $f_{21} \sim f_{28}$ 为复合函数. 由于本文算法是基于多峰优化问题提出的, 选取多峰函数 $f_6 \sim f_{20}$ 进行测试, 进一步验证 SLFA 在多峰优化问题上的优化性能.

比较算法包括: 改进人工蜂群算法 AABC^[21]、OCABC^[22]、qABC^[23] 和 ABCG^[24], 改进粒子群优化算法 PSODDS^[25]、SRPSO^[26]、CCPSO-ISM^[27] 和 MSMP SO^[28]. 实验设置参照该测试集提出时所列出的相关设置, 设置 $D = 30$, $MAX_FEs = 10000 \times D$, 各算法的具体参数参照相应的参考文献. 算法在每个函数上运行 30 次, 优化结果如表 4 所示.

由表 4 可见, 与改进人工蜂群算法 AABC、OCABC、qABC 和 ABCG 相比, SLFA 的优化性能分别在 14、11、11 和 12 个函数上胜出, 在 1、4、4 和 1 个函数上处于劣势; 与改进粒子群优化算法 PSODDS、SRPSO、CCPSO-ISM 和 MSMP SO 相比, SLFA 分别在 13、12、11 和 8 个函数上胜出, 在 1、3、3 和 6 个函数上处于劣势. 从这些数据可以看出, 与比较算法相比, SLFA 性能占优的函数个数远多于处于劣势的函数个数, 并且 SLFA 在函数 f_{11} 上寻到了全局最优, 而其他算法在所有函数上都没有寻到全局最优. 由此可知, 在优化难度更大的 CEC2013 测试函数上, SLFA 依然保持高效的性能. 由 Friedman 检验可以看出, SLFA 在多峰函数上的综合优化性能明显优于相关比较算法.

5 结论

本文针对多峰优化问题, 分析了萤火虫算法存在“群集现象”, 使得当最优萤火虫陷入局部最优时种群也会陷入局部最优且无法逃逸, 不利于算法对多极值空间的探索. 在充分考虑多峰优化问题具有多极值优化空间特性基础上, 提出一种自主学习萤火虫算法. 算法将萤火虫划分为自主学习粒子和普通粒子,

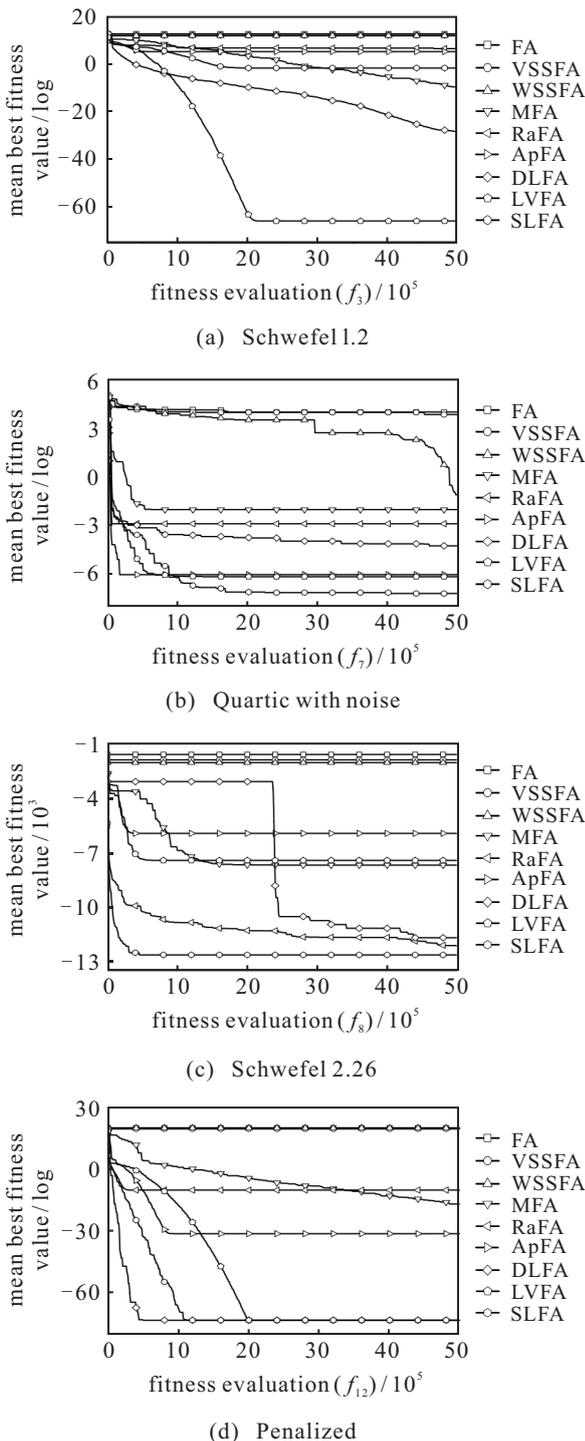


图4 算法收敛过程曲线

表4 9种算法在CEC2013测试函数上的优化均值结果

		AABC	OCABC	qABC	ABCG	PSODDS	SRPSO	CCPSO-ISM	MSMPSO	SLFA
f_6	mean	6.09e+02	1.79e+01	1.53e+01	4.16e+01	6.05e+01	6.95e+01	6.37e+01	7.91e+01	1.72e+01
	std	3.55e+02	4.92e+00	2.04e+00	1.32e+01	2.95e+00	1.90e+01	2.32e+01	2.85e+01	5.55e+00
f_7	mean	5.13e+01	1.14e+02	1.22e+02	9.94e+01	1.05e+02	4.44e+01	6.40e+00	1.99e+01	4.94e+01
	std	1.86e+01	1.71e+01	1.25e+01	1.36e+01	1.98e+01	1.96e+01	3.30e+00	1.13e+01	1.37e+01
f_8	mean	2.10e+01	2.10e+01	2.10e+01	2.09e+01	2.09e+01	2.10e+01	2.09e+01	2.09e+01	2.09e+01
	std	5.48e-02	3.26e-02	5.86e-02	4.44e-02	4.61e-02	2.91e-02	4.06e-02	5.45e-02	5.18e-02
f_9	mean	2.49e+01	2.93e+01	2.97e+01	2.91e+01	2.61e+01	1.79e+01	1.71e+01	1.45e+01	1.51e+01
	std	3.89e+00	1.40e+00	1.60e+00	1.08e+00	2.26e+00	2.96e+00	2.98e+00	2.68e+00	5.26e+00
f_{10}	mean	5.68e+02	1.51e+00	4.12e-01	2.64e-01	7.63e+01	9.07e+01	1.18e-01	8.67e-02	5.77e-02
	std	2.63e+02	2.38e-01	7.69e-02	1.03e-01	4.89e+01	6.78e+01	4.60e-02	3.79e-02	3.48e-02
f_{11}	mean	1.65e+02	5.68e-14	1.42e-13	1.49e-13	7.98e+01	3.86e+01	2.39e+01	2.59e+01	0.00e+00
	std	4.91e+01	6.75e-30	4.30e-14	2.94e-14	2.17e+01	8.66e+00	6.32e+00	6.38e+00	0.00e+00
f_{12}	mean	1.80e+02	1.52e+02	2.59e+02	1.31e+02	1.62e+02	5.60e+01	3.91e+01	5.16e+01	7.94e+01
	std	3.35e+01	3.28e+01	1.94e+01	2.21e+01	3.56e+01	1.46e+01	1.34e+01	1.67e+01	1.42e+01
f_{13}	mean	2.13e+02	2.29e+02	3.41e+02	2.07e+02	2.71e+02	1.19e+02	9.44e+01	1.04e+02	1.02e+02
	std	5.64e+01	1.99e+01	3.59e+01	2.08e+01	4.49e+01	2.15e+01	2.42e+01	2.70e+01	2.26e+01
f_{14}	mean	3.84e+03	3.69e-01	1.83e-01	2.50e-01	2.00e+03	1.37e+03	1.40e+03	1.44e+03	4.71e-01
	std	5.20e+02	4.08e-01	6.45e-02	5.04e-02	4.55e+02	2.20e+02	3.17e+02	3.86e+02	4.37e-01
f_{15}	mean	3.81e+03	3.55e+03	3.64e+03	3.82e+03	4.09e+03	3.02e+03	3.67e+03	3.21e+03	3.61e+03
	std	9.53e+02	4.04e+02	3.40e+02	3.23e+02	3.99e+02	6.53e+02	1.28e+03	4.22e+04	4.21e+02
f_{16}	mean	2.52e+00	1.30e+00	1.02e+00	1.65e+00	8.72e-01	2.50e+00	2.39e+00	2.29e+00	9.35e-01
	std	1.93e-01	2.55e-01	1.66e-01	2.21e-01	4.15e-01	2.32e-01	2.33e-01	3.62e-01	1.72e-01
f_{17}	mean	2.26e+02	2.71e+01	2.79e+01	3.04e+01	1.08e+02	6.16e+01	5.85e+01	5.60e+01	3.04e+01
	std	4.71e+01	9.58e+00	7.12e+00	2.97e-03	1.48e+01	7.09e+00	6.96e+00	5.86e+00	5.53e-03
f_{18}	mean	2.36e+02	1.97e+02	2.69e+02	2.10e+02	1.71e+02	1.99e+02	1.86e+02	6.95e+01	8.73e+01
	std	7.53e+01	2.42e+01	3.84e+01	1.45e+01	3.183+01	2.02e+01	1.55e+01	1.44e+01	1.62e+01
f_{19}	mean	1.55e+03	5.35e-01	2.27e-01	6.91e-01	1.02e+01	4.05e+00	2.67e+00	3.27e+00	6.06e-01
	std	3.42e+03	2.60e-01	4.30e-02	1.84e-01	5.16e+00	1.81e+00	4.17e-01	6.16e-01	1.70e-01
f_{20}	mean	1.07e+01	1.41e+01	1.45e+01	1.44e+01	1.42e+01	1.39e+01	1.23e+01	9.92e+00	1.14e+01
	std	7.21e-01	5.31e-01	2.91e-01	4.12e-01	8.48e-01	1.29e+00	1.64e+00	6.68e-01	1.53e+00
rank		7.43 (9)	4.77 (4)	5.50 (7)	5.23 (5)	6.33 (8)	5.43 (6)	4.00 (3)	3.60 (2)	2.70 (1)
$w/l/t$		14/0/1	11/0/4	11/0/4	12/2/1	13/1/1	12/0/3	11/1/3	8/1/6	

自主学习粒子具有自主学习的能力,不仅能够维持算法对多个极值空间的开发,而且其强大的学习能力能够保证算法的优化精度;普通粒子有更大概率探索到当前种群未探索到的区域,使算法跳出局部最优,进一步加强算法对多极值空间的探索能力.此外,使用淘汰机制使算法舍弃对劣质极值空间的维护,进而提高对优质极值空间的开发,进一步提高算法的优化精度.实验结果表明,自主学习萤火虫算法能够有效处理多峰优化问题.本文的自主学习策略使用随机选择一维的进化方式,该方式虽然获得了非常好的优化性能,但维度个数确定和哪些维度被选择对萤火虫算法的性能影响较大.下一步将通过强化学习等理论分析粒子各维度对算法优化性能的影响,研究多维度的自主学习策略,以获得更高效的优化算法.

参考文献(References)

[1] 孙小军, 赵天绪, 焦建民. 多峰极值优化问题的算法性能分析和比较[J]. 昆明理工大学学报: 理工版, 2009, 34(5): 104-107.
(Sun X J, Zhao T X, Jiao J M. Performance analysis and comparison of algorithms for optimizing multiple hump function[J]. Journal of Kunming University of Science and Technology: Science and Technology, 2009, 34(5): 104-107.)

[2] Kennedy J, Eberhart R C, Shi Y. Swarm intelligence[M]. San Francisco: Morgan Kaufmann Publishers, 2001: 287-325.

[3] 邓志诚, 孙辉, 赵嘉, 等. 方波触发勘探与开发的粒子群优化算法[J/OL]. 自动化学报, DOI: 10.16383/j.aas.c190842.
(Deng Z C, Sun H, Zhao J, et al. Particle swarm optimization with square wave triggered exploration

- and exploitation[J/OL]. *Acta Automatica Sinica*, DOI: 10.16383/j.aas.c190842.)
- [4] Wang Y C, Liu J H, Xiao R B. Artificial bee colony algorithm based on stimulus-response labor division[J]. *Control and Decision*, 2022, 37(4): 881-891.
- [5] 崔志华, 张春妹, 时振涛, 等. 基于蝙蝠算法的观测矩阵优化算法[J]. *控制与决策*, 2018, 33(7): 1341-1344. (Cui Z H, Zhang C M, Shi Z T, et al. Measurement matrix optimization algorithm with bat algorithm[J]. *Control and Decision*, 2018, 33(7): 1341-1344.)
- [6] Zhao J, Tang J J, Shi A Y, et al. Improved Density Peaks clustering based on Firefly Algorithm[J]. *International Journal of Bio-Inspired Computation*, 2020, 15(1): 24-42.
- [7] Yang X S. *Nature-inspired metaheuristic algorithms*[M]. London: Luniver Press, 2008: 81-89.
- [8] 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.
- [9] Zhou L Y, Ding L X, Ma M D, et al. An accurate partially attracted firefly algorithm[J]. *Computing*, 2019, 101(5): 477-493.
- [10] 周凌云, 丁立新, 何进荣. 精英正交学习萤火虫算法[J]. *计算机科学*, 2015, 42(10): 211-216. (Zhou L Y, Ding L X, He J R. Elite orthogonal learning firefly algorithm[J]. *Computer Science*, 2015, 42(10): 211-216.)
- [11] 赵嘉, 谢智峰, 吕莉, 等. 深度学习萤火虫算法[J]. *电子学报*, 2018, 46(11): 2633-2641. (Zhao J, Xie Z F, Lv L, et al. Firefly algorithm with deep learning[J]. *Acta Electronica Sinica*, 2018, 46(11): 2633-2641.)
- [12] Wang W C, Xu L, Chau K W, et al. Yin-Yang firefly algorithm based on dimensionally Cauchy mutation[J]. *Expert Systems with Applications*, 2020, 150: 113216.
- [13] 谢智峰, 吴润秀, 吕莉. 多策略融合学习萤火虫算法在年径流预测中的应用[J]. *南昌工程学院学报*, 2021, 40(1): 20-27. (Xie Z F, Wu R X, Lv L. Application of multi-strategy fusion learning firefly algorithm in annual runoff forecast[J]. *Journal of Nanchang Institute of Technology*, 2021, 40(1): 20-27.)
- [14] Zhao J, Chen W P, Ye J, et al. Firefly algorithm based on level-based attracting and variable step size[J]. *IEEE Access*, 2020, 8: 58700-58716.
- [15] Yu S H, Su S B, Lu Q P, et al. A novel wise step strategy for firefly algorithm[J]. *International Journal of Computer Mathematics*, 2014, 91(12): 2507-2513.
- [16] Yu S H, Zhu S L, Ma Y, et al. A variable step size firefly algorithm for numerical optimization[J]. *Applied Mathematics and Computation*, 2015, 263: 214-220.
- [17] Fister Jr I, Yang X S, Fister I, et al. Memetic Firefly algorithm for combinatorial optimization[C]. *Proceedings of International Conference on Bioinspired Optimization Methods and their Applications*. Piscataway: IEEE, 2012: 1-14.
- [18] Wang H, Wang W J, Sun H, et al. Firefly algorithm with random attraction[J]. *International Journal of Bio-Inspired Computation*, 2016, 8(1): 33-41.
- [19] Wang H, Zhou X Y, Sun H, et al. Firefly algorithm with adaptive control parameters[J]. *Soft Computing*, 2017, 21(17): 5091-5102.
- [20] Liang J J, Qu B Y, Suganthan P N, et al. Problem definitions and evaluation criteria for the CEC2013 special session on real-parameter optimization[R]. Zhengzhou: Zhengzhou University, 2013.
- [21] Liao X, Zhou J Z, Zhang R, et al. An adaptive artificial bee colony algorithm for long-term economic dispatch in cascaded hydropower systems[J]. *International Journal of Electrical Power & Energy Systems*, 2012, 43(1): 1340-1345.
- [22] Gao W F, Liu S Y, Huang L L. A novel artificial bee colony algorithm based on modified search equation and orthogonal learning[J]. *IEEE Transactions on Cybernetics*, 2013, 43(3): 1011-1024.
- [23] Karaboga D, Gorkemli B. A quick artificial bee colony (qABC) algorithm and its performance on optimization problems[J]. *Applied Soft Computing*, 2014, 23: 227-238.
- [24] Xiang W L, Meng X L, Li Y Z, et al. An improved artificial bee colony algorithm based on the gravity model[J]. *Information Sciences*, 2018, 429: 49-71.
- [25] Jin X, Liang Y Q, Tian D P, et al. Particle swarm optimization using dimension selection methods[J]. *Applied Mathematics and Computation*, 2013, 219(10): 5185-5197.
- [26] Tanweer M R, Suresh S, Sundararajan N. Self regulating particle swarm optimization algorithm[J]. *Information Sciences*, 2015, 294: 182-202.
- [27] Li Y H, Zhan Z H, Lin S J, et al. Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems[J]. *Information Sciences*, 2015, 293: 370-382.
- [28] 邓先礼, 魏波, 曾辉, 等. 基于多种群的自适应迁移 PSO 算法[J]. *电子学报*, 2018, 46(8): 1858-1865. (Deng X L, Wei B, Zeng H, et al. A multi-population based self-adaptive migration PSO[J]. *Acta Electronica Sinica*, 2018, 46(8): 1858-1865.)

作者简介

赵嘉(1981—), 男, 教授, 博士, 从事机器学习与数据挖掘、计算智能等研究, E-mail: zhaojia925@163.com;

陈文平(1995—), 男, 硕士生, 从事智能计算的研究, E-mail: chen_9731@163.com;

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

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

(责任编辑: 郑晓蕾)