

# 改进萤火虫算法及其收敛性分析

张大力<sup>1</sup>, 夏红伟<sup>1</sup>, 张朝兴<sup>2</sup>, 马广程<sup>1</sup>, 王常虹<sup>1,\*</sup>

(1. 哈尔滨工业大学航天学院, 黑龙江 哈尔滨 150001; 2. 上海航天控制技术研究所, 上海 201109)

**摘要:** 萤火虫算法因具有结构简单、控制参数少、易于实现的特点而得到广泛的关注和应用,但其易陷入局部最优导致过早收敛,从而影响寻优精度。针对这一问题,本文在位置更新规则中加入随机扰动因子,并剔除了冗余的随机项,以提高算法搜索能力;引入位置置换变异和差分进化算法中的最优变异策略,在保持种群多样性的同时,增强算法跳出局部最优的能力。采用马尔可夫过程证明了算法以概率1收敛到全局最优。利用基准函数和装箱问题对算法进行仿真测试,结果表明,改进后的算法能够有效跳出局部最优,对给出的所有问题均能找到理论最优解,寻优精度和成功率有明显提升。

**关键词:** 萤火虫算法; 随机扰动; 变异策略; 马尔可夫过程; 函数优化; 装箱问题

**中图分类号:** N 945.15

**文献标志码:** A

**DOI:**10.12305/j.issn.1001-506X.2022.04.27

## Improved firefly algorithm and its convergence analysis

ZHANG Dali<sup>1</sup>, XIA Hongwei<sup>1</sup>, ZHANG Chaoxing<sup>2</sup>, MA Guangcheng<sup>1</sup>, WANG Changhong<sup>1,\*</sup>

(1. School of Astronautics, Harbin Institute of Technology, Harbin 150001, China;

2. Shanghai Aerospace Control Technology Institute, Shanghai 201109, China)

**Abstract:** The firefly algorithm has been widely concerned and applied because of its characteristics of simple structure, few control parameters and easy implementation, but it is easy to fall into local optimum, which leads to premature convergence and affects the optimization accuracy. To solve this problem, this paper adds a random factor into the individual location update rule to improve search capabilities, and the redundant random items are eliminated. To maintain the diversity of the population and enhance the capabilities of the algorithm to jump out of the local optimum, a position substitution mutation strategy and an optimal mutation strategy which come from the differential evolution algorithm are introduced. The Markov process is used to theoretically analyze the improved algorithm, and it is proved that the algorithm converges to the global optimum with probability of 1. The improved algorithm is simulated and tested using classic benchmark functions and the bin packing problem. Simulation results show that the improved algorithm can effectively jump out of the local optimum and find the theoretical optimal solution for all the given problems with better optimization accuracy and success rate.

**Keywords:** firefly algorithm; stochastic disturbance; mutation strategy; Markov process; function optimization; bin packing problem

## 0 引言

社会性生物的某些集体行为往往呈现出寻优的本质,通过对这类行为的公式化,得到了一类被称为群智能算法的优化算法<sup>[1]</sup>,蚁群优化(ant colony optimization, ACO)算法<sup>[2]</sup>、

粒子群优化(particle swarm optimization, PSO)算法<sup>[3]</sup>、人工蜂群(artificial bee colony, ABC)算法<sup>[4]</sup>等均属此列。由于算法简单灵活、鲁棒性好,可用于求解许多传统方法无法解决的 NP-hard 问题,如旅行商问题、装箱问题等,已经广泛应用于人工智能、通信网络和工业生产等领域<sup>[5]</sup>。

收稿日期:2021-12-07; 修回日期:2022-01-25; 网络优先出版日期:2022-03-01。

网络优先出版地址:https://kns.cnki.net/kcms/detail/11.2422.TN.20220301.1416.010.html

基金项目:国家自然科学基金(61304108);国家重点研发计划(2020YFC2200600)资助课题

\* 通讯作者。

引用格式:张大力,夏红伟,张朝兴,等.改进萤火虫算法及其收敛性分析[J].系统工程与电子技术,2022,44(4):1291-1300.

**Reference format:** ZHANG D L, XIA H W, ZHANG C X, et al. Improved firefly algorithm and its convergence analysis[J]. Systems Engineering and Electronics, 2022, 44(4): 1291-1300.

萤火虫算法 (firefly algorithm, FA) 是英国学者 Yang 于 2008 年提出的一种元启发式算法<sup>[6]</sup>。该算法通过模拟自然界中萤火虫个体之间相互吸引的理想化行为达到寻优目的, 一经提出便受到国内外研究人员的关注, 广泛应用于计算机科学<sup>[7]</sup>、复杂方程求解<sup>[8]</sup>、结构和工程优化领域<sup>[9-10]</sup>。但许多构造的和现实的优化问题具有大量的局部最优解, 容易使算法因种群多样性的快速衰减而陷入早熟停滞, 大幅降低优化效果。同时, 大量局部最优对种群的吸引可能导致算法不断震荡而无法收敛。有学者通过引入自适应或随机机制针对上述问题进行改进。Gandomi 等<sup>[11]</sup>采用 12 种不同的混沌映射对参数进行自适应调节, 其中高斯映射调节吸引度参数的改进算法取得了一定效果, 但性能提升有限, 寻优成功率不高。Yu 等<sup>[12]</sup>设计了一种自适应步长策略来提高算法的全局探索能力, 每个个体的随机步长根据当前和历史信息分别计算, 然而实际寻优效果与标准算法区别很小。Altay 等<sup>[13]</sup>将该改进算法用于调节数字水印技术中的比例因子, 由于缺乏与标准算法的横向对比, 不足以说明改进算法的优越性。Verma 等<sup>[14]</sup>在算法初始化阶段引入反向学习策略提高算法收敛速度, 在迭代过程中采用基于维度的位置更新方法跳出局部最优, 但与其他优化算法相比没有体现出明显的竞争力。Manju 等<sup>[15]</sup>参考量子 PSO 提出了量子 FA, 并利用基于二次插值的重组算子改进搜索过程, 提高了算法的收敛速度和寻优能力, 但无法处理复杂病态优化问题。柳长源等<sup>[16]</sup>引入了一种驱散机制, 若种群最优值在连续多次迭代后未发生变化, 则对最优个体采用柯西分布进行变异, 聚集在最优个体附近的个体采用均匀分布进行驱散, 从函数优化结果来看改进效果并不显著。刘景森等<sup>[17]</sup>提出了一种具有振荡、约束和自然选择机制的改进策略, 提高了算法对高维问题的求解能力, 但是增加了额外的调节参数, 不利于推广应用, 其收敛性证明仿照 PSO 算法证明的思路, 简化到二维空间进行分析, 无法向高维情况推广。部分学者选择从个体信息交互的角度对算法进行改进。Wang 等<sup>[18]</sup>提出了一种随机吸引模型, 通过将个体与其邻域内有限个其他个体进行通信, 降低了算法计算复杂度, 但寻优性能没有明显提升。在此基础上, Dhal 等<sup>[19]</sup>进一步考虑粗糙集和局部搜索机制, 提出了一种随机吸引粗糙 FA, 并成功应用于图像分割过程, 但与标准算法及其他传统优化算法相比没有明显的竞争力。Shan 等<sup>[20]</sup>通过分布式并行技术将种群划分为多个子群, 子群之间基于 4 种不同的通信策略进行信息交换, 但每个子群的寻优能力偏弱, 不能保证整体的寻优效果, 且增加了额外的配置参数, 适用性不强。Elakkiya 等<sup>[21]</sup>提出了一种基于社区启发的改进策略, 个体只被由  $c$  个较优个体组成的所谓社区中的个体吸引, 并给出了社区规模的自适应更新机制, 降低了计算复杂度, 但无法保证寻优精度。还有一种改进策略是在 FA 中引入其他算法的思想。Zhao 等<sup>[22]</sup>借鉴 ABC 算法的思想将萤火虫种群分为领导者、开发者和跟随者 3 种角色, 并为每个角色分配了一种学习策略, 用来平衡算法的探索和挖掘能力, 但没有取得显著的性能改进。Xing 等<sup>[23]</sup>为了使算法跳出局部最优, 引入了遗传

算法中的变异算子, 同时仿照 PSO 算法将全局最优纳入到位置更新公式中, 改进算法用于解决协同干扰资源分配问题, 寻优结果与标准算法相比有少许提升。薛晗<sup>[24]</sup>等结合文化算法和 FA 的优点, 将迭代过程中获取的经验和知识组成信仰空间, 并设计了一套规则来影响种群更新, 但算法没有足够的跳出局部最优的能力。Gupta 等<sup>[25]</sup>将引力搜索和 FA 通过串行运行的方式结合起来, 提出了一种混合引力搜索 FA, 计算复杂度较高, 寻优能力没有明显提升。此外, Cheng 等<sup>[26]</sup>和 Xue 等<sup>[27]</sup>采用混合编码的形式对个体进行编码, 利用组合变异、协同进化等方式加快算法收敛速度, 但该方法只适用于解决离散工程优化问题, 缺乏适用性。

基于以上分析可以看出, 目前 FA 仍然具有较大的改进潜力, 并且理论分析比较薄弱, 也在一定程度上制约了 FA 的进一步改进和发展。因此, 本文提出一种改进 FA (improved FA, IFA), 以提升算法性能。首先, 介绍 FA 的基本原理。其次, 给出新的个体位置更新机制。围绕种群更新过程, 引入位置置换变异策略和差分进化 (differential evolution, DE) 算法中的最优变异策略 (DE/best/1), 并给出算法流程。然后, 给出改进算法收敛性的理论证明。最后, 采用一组典型基准函数和装箱问题对改进算法有效性进行验证。

## 1 FA 基本原理

FA 受萤火虫群体的集聚性启发, 其仿生原理如下: 搜索空间中的单个元素代表萤火虫个体; 拟优化的目标抽象为个体所在位置的亮度, 衡量个体位置的优劣并决定其移动方向; 元素搜索和更新的寻优过程模拟个体间的吸引和移动行为。

算法数学描述如下:

(1) 个体  $i$  的位置记为  $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ , 其亮度即为该位置对应的目标函数值  $f(x_i)$ 。

(2) 个体  $i$  与  $j$  间的吸引度更新公式定义为

$$\beta_{ij} = \beta_0 e^{-\gamma r_{ij}^\gamma} \quad (1)$$

式中:  $\beta_0$  为初始吸引度, 一般取值为 1;  $\gamma$  为光强吸收系数, 一般取值为 1;  $r_{ij}$  为个体间的笛卡尔距离, 有

$$r_{ij} = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2} \quad (2)$$

(3) 个体  $i$  被  $j$  吸引并向其移动的位置更新公式为

$$x_i^{k+1} = x_i^k + \beta_{ij} \cdot (x_j^k - x_i^k) + \alpha \epsilon_i \quad (3)$$

式中:  $k$  为算法迭代次数;  $\alpha$  为步长因子;  $\epsilon_i$  为随机数, 服从均匀分布或高斯分布。

## 2 IFA

威廉·奥卡姆曾言, “如无必要, 勿增实体”, 这就是著名的“奥卡姆剃刀定律”。从优化算法设计的角度来诠释, 可理解为在效果相同或相近的前提下, 尽量选择简单框架和规则。在这种思想的指导下, 本文提出以下改进策略。

### 2.1 随机扰动策略

原位置更新式 (3) 中第 2 项是一个由个体  $i^k$  指向个体  $j^k$  的一类直线运动, 如图 1 所示。以二维空间为例, 假设无

穷远处为参考点,  $\mathbf{A}$  和  $\mathbf{B}$  分别表示个体  $i^k$  和吸引力项的位置矢量, 其线性组合构成了个体  $i^k$  的新位置  $i^{k+1}$  的矢量  $\mathbf{C}$ 。吸引力项能够使种群迅速向亮度更高的个体聚集, 加速算法收敛, 但由于确定性信息交互的局限性, 需要一定的随机机制来提高算法的搜索范围。同时, 自然界萤火虫个体的运动是包含随机属性的, 位置更新后可能出现在图 1“ $\odot$ ”所示的任意位置。

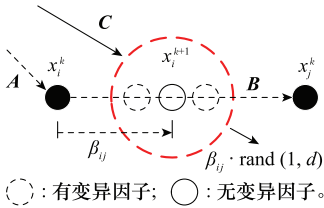


图 1 吸引力项运动特征

Fig. 1 Motion characteristic of attraction term

由于式(3)第 3 项随机项  $\alpha \epsilon_i$  的变化是独立的, 没有利用个体和种群信息, 因此其变化很难与个体的进化相匹配。一方面, 如果  $\alpha$  取值过小, 那么较小的搜索半径无法提供跳出局部最优的能力, 如果  $\alpha$  取值较大, 则会因为较大的搜索半径导致较慢的收敛速度和较低的收敛精度。另一方面, 一旦在随机项中引入个体信息, 那么可与改进后的第 2 项合并同类项, 在功能上被其覆盖。

基于以上考虑, 向原位置更新式(3)中引入一个随机扰动因子  $S_i$ , 使得吸引力项拥有一个随机权重, 并去掉式(3)第 3 项, 有

$$x_i^{k+1} = x_i^k + \beta_{ij} \cdot S_i \cdot (x_j^k - x_i^k) \quad (4)$$

为保证式(4)在初始阶段的搜索能力, 将吸引度更新为

$$\beta_{ij} = \beta_0 + \beta_{re} e^{-r_{ij}^2} \quad (5)$$

式中:  $\beta_{re}$  为吸引度调节尺度, 根据调试经验建议取值范围为 0.7~0.9。

随机扰动因子的引入提高了个体的搜索范围。一方面, 在算法初始阶段, 个体间距离较大, 式(5)表示的随机权重简化为  $\beta_0 \cdot S_i$ , 此时搜索步长较长, 算法能够在全局范围内进行寻优, 一定程度上提高了算法的全局探索能力; 另一方面, 在算法迭代后期, 个体间距离大幅减小, 随机权重简化为  $(\beta_0 + \beta_{re}) \cdot S_i$ , 而此时搜索步长较小, 算法能够在当前位置的邻域深度挖掘。个体搜索范围的提升使得种群在运动过程中能够覆盖到更多搜索空间中的区域, 如图 2 所示。

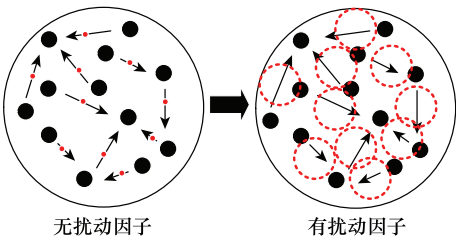


图 2 种群运动趋势和个体分布情况

Fig. 2 Population movement trend and individual distribution

下面对随机扰动因子  $S_i$  的选择进行分析。典型的随机分布有均匀分布、正态分布、柯西分布、列维分布等。为了更直观地区分不同分布的特征, 给出上述分布 100 000 个样本的分布直方图, 如图 3 所示。

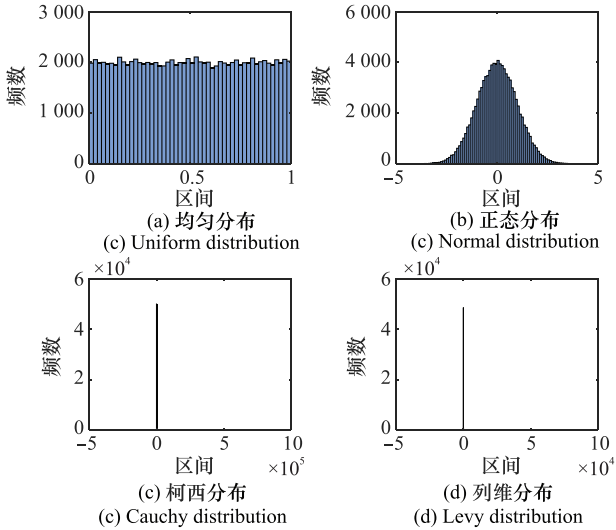


图 3 4 种不同概率分布的 100 000 样本分布

Fig. 3 Four different probability distributions of 100 000 random samples

由图 3 可以看出, 均匀分布产生的样本分布均匀且取值区间较小, 选择均匀分布作为随机扰动因子可获得在各个方向上更加稳定的搜索能力。正态分布产生的样本呈现出左右对称、中间高两侧低的钟型分布特征, 在零点附近取值的概率较高, 离原点越远取值概率越低, 选择正态分布作为随机扰动因子, 存在初始阶段收敛慢的情况, 不能保证算法收敛速度。柯西分布和列维分布产生的样本呈现出明显的厚尾特征, 在原点附近取值的概率较高, 偶尔会出现非常大的样本, 选择柯西或列维分布作为随机扰动因子, 算法搜索能力不稳定, 且产生的大样本很有可能使个体更新超出搜索空间, 精度和收敛速度均无法保证。基于以上分析, 随机扰动因子首选均匀分布, 在第 4.1 节中给出仿真, 对以上分析做进一步说明。

## 2.2 双重变异策略

为进一步提升算法寻优性能, 本文围绕个体和种群的更新过程增加两个变异机制。

### 2.2.1 位置置换变异策略

本文提出一种位置置换变异策略, 具体描述如下: 萤火虫个体  $i$  在第  $k$  步迭代完成一轮位置更新后的位置记为  $x_i^k$ , 随机抽取  $d$  维位置数据中的两个互换位置, 重复  $n_{\text{Mutation}}$  次后得到  $x_{m_i}^k$ 。计算目标函数在  $x_{m_i}^k$  的适应度  $f(x_{m_i}^k)$ , 若  $f(x_{m_i}^k) < f(x_i^k)$ , 则令  $x_i^k = x_{m_i}^k$ , 进一步, 若  $f(x_i^k) < f(x_{\text{best}}^k)$ , 则令  $x_{\text{best}}^k = x_i^k$ 。经变异处理后, 种群  $X^k$  更新为  $X_{M_i}^k$ , 该部分流程如图 4 所示。

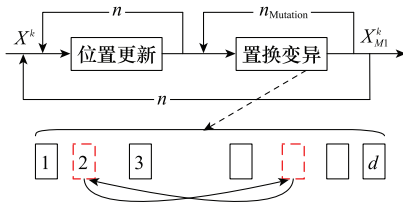


图 4 位置置换变异原理和种群更新流程

Fig. 4 Principle of position substitution mutation and population renewal process

参数  $n_{\text{Mutation}}$  的取值是位置置换变异策略的关键。一方面,由于该操作在个体位置更新后进行,且嵌套于种群更新过程中,因此  $n_{\text{Mutation}}$  不能过大,否则会提高算法计算复杂度,降低收敛速度。另一方面,若次数太少,个体位置变化较小,无法达到跳出局部最优的目的,降低寻优精度。因此,置换变异至少 2 次,至多建议不超过 5 次,本文取  $n_{\text{Mutation}} = 3$ 。

### 2.2.2 DE/best/1 变异策略

为充分利用种群最优信息提高算法性能,引入差分进化算法中的 DE/best/1 变异策略。该策略基于历史最优信息进行操作,只包含一个差分向量,搜索过程不考虑问题的特性,是有很强竞争力的搜索策略。具体操作是,从种群  $X_{M1}^k$  中随机抽取两个个体  $r_1$  和  $r_2$ ,有

$$x_{m2i}^k = x_{\text{best}}^k + F \cdot (x_{m1r_1}^k - x_{m1r_2}^k) \quad (6)$$

其中,摄动因子  $F \sim U(0.4, 0.9, 1, d)$ 。经变异处理后,种群  $X_{M1}^k$  更新为  $X_{M2}^k$ ,如图 5 所示。计算  $f(x_{m2i}^k)$ ,若  $f(x_{m2i}^k) < f(x_{\text{best}}^k)$ ,则  $x_{\text{best}}^k = x_{m2i}^k$ 。

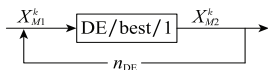


图 5 DE/best/1 变异种群更新流程

Fig. 5 DE/best/1 cross mutation population renewal process

DE/best/1 变异操作的执行次数为  $n_{\text{DE}}$ 。由于该部分没有循环嵌套,  $n_{\text{DE}}$  可以取大一些的数,从而充分利用历史最优信息指导搜索过程,提高算法收敛精度。参考差分进化算法的取值,  $n_{\text{DE}}$  可取种群规模  $n$ 。

两种变异策略得到的子代种群采取与父代种群竞争的模式选取进入下一轮循环的种群,在每轮迭代中保留较优个体,提升整体种群质量。

### 2.3 算法平衡性分析

全局探索与局部挖掘能力是群智能优化算法解决问题的核心,二者之间的平衡对算法性能至关重要。研究普遍认为,在算法开始阶段应以全局探索为主,在后期应以局部挖掘为主。改进算法能够很好地满足此规律。在算法迭代初始阶段,个体间相距较远,从位置更新式(4)和 DE/best/1 变异策略式(6)可直观地看出,此时算法有较大的搜索步长,能够在解空间内进行全局探索。算法迭代后期,种群不断聚集,搜索步长随之减小,此时算法的局部挖掘能力得到不断强化。第 2.2.1 节所述的位置置换变异策略是对个体

自身信息的突变,该操作不受迭代过程的影响,无论在迭代前期还是迭代后期均能使个体有效地跳出当前位置,从而避免算法陷入局部最优。需要注意的是,在实际运行中,全局探索和局部挖掘能力是算法各个策略共同触发的,没有明确的界限。

### 2.4 算法流程

下面给出改进算法的流程。

**步骤 1** 初始化控制参数  $\beta_0, \beta_e, \gamma$ , 位置变异次数为  $n_{\text{Mutation}}$ , DE/best/1 变异执行  $n_{\text{DE}}$  次,种群规模为  $n$ ,搜索空间下界为  $x_{\min}$ ,上界为  $x_{\max}$ ,最大迭代次数为 MaxGeneration。

**步骤 2** 构建服从均匀分布的初始种群  $X_0$ , 计算适应度和当前全局最优解  $g_0$ 。

**步骤 3** 取种群中的一个个体,判断其与其他个体的吸引关系,根据式(2)、式(4)、式(5)更新该个体的位置并进行限位,同时更新全局最优值  $g_0$ ,该步骤执行  $n$  次。

**步骤 4** 根据第 2.2.1 节给出的策略对步骤 3 中选定的个体进行位置置换变异,该步骤执行  $n_{\text{Mutation}}$  次。

**步骤 5** 步骤 3~步骤 4 连续执行,重复次数为  $n$  次,得到子种群  $X_{M1}$ 。

**步骤 6** 根据第 2.2.2 节给出的策略对种群  $X_{M1}$  进行变异操作,同时更新全局最优值  $g_0$ 。该步骤执行  $n_{\text{DE}}$  次,得到子种群  $X_{M2}$ 。

**步骤 7** 父代种群和子代种群拼接并排序后截断保留维数据,得到下一轮迭代的父代种群  $X_K$ 。

**步骤 8** 本文设置停止准则为迭代次数  $k$  达到 Max-Generation,若满足条件,则终止判断,输出优化结果并可视化;若不满足条件,则重复步骤 3~步骤 7。

### 2.5 算法计算复杂度分析

结合算法流程进行分析,为了简便这里记  $N$  为迭代次数,  $N_f$  为适应度函数的计算量,那么算法语句总的执行次数可以表示为  $N \cdot [(5 + N_f) \cdot n^2 + n_{\text{Mutation}} \cdot (N_f + 4) \cdot n + (N_f + 7) \cdot n]$ ,此时改进算法的时间复杂度为  $O(N \cdot N_f \cdot n^2)$ 。显然,算法的计算复杂度与问题本身的复杂度直接相关,主要由适应度函数计算量和算法计算量决定。

## 3 收敛性证明

### 3.1 随机算法的收敛准则

Wets<sup>[28]</sup> 给出了一般随机优化算法收敛性判别准则,下面重述其主要结论。

假设有随机优化问题  $[f, S_D]$ , 其迭代方式为  $D$ , 第  $k$  次迭代结果  $x_k$ , 则  $x_{k+1} = D(x_k, \zeta)$ 。其中,  $f$  为适应度函数,  $S_D$  为解空间,  $\zeta$  是  $k$  次迭代产生的所有解。

在 Lebesgue 测度空间定义搜索的下确界:

$$\varphi = \inf\{t: v[x \in S_D | f(x) < t] > 0\} \quad (7)$$

式中:  $v[X]$  表示集合  $X$  上的 Lebesgue 测度。根据式(7)定义最优解区域为

$$R_{\epsilon, M} = \begin{cases} \{x \in S_D \mid f(x) < \varphi + \epsilon\}, & \varphi \text{ 有限} \\ \{x \in S_D \mid f(x) < -C\}, & \varphi = -\infty \end{cases} \quad (8)$$

式中:  $\epsilon > 0$  且足够小,  $C > 0$  且足够大。若算法找到了  $R_{\epsilon, M}$  中的一个点, 则算法收敛。

下面给出算法收敛的充分条件:

**条件 1** 假设有算法  $D$ , 使得  $\{f(x_k)\}_{k=0}^{\infty}$  非增, 即  $f(D(z, \zeta)) \leq f(z)$ , 若  $\zeta \in S_D$ , 则有  $f(D(z, \zeta)) \leq f(\zeta)$ 。

**条件 2** 对  $S_D$  的任意 Borel 子集  $A$ , 若  $A$  的 Lebesgue 测度  $v(B) > 0$ , 则  $\prod_{k=0}^{\infty} (1 - \mu_k(A)) = 0$  成立。其中  $\mu_k(A)$  是由测度  $\mu_k$  所得到的  $A$  的概率。

**定理 1** 全局搜索收敛

假设  $f$  可测,  $S_D$  是  $R^n$  上的可测子集, 算法  $D$  满足上述两个条件,  $\{x_k\}_{k=0}^{\infty}$  是  $D$  产生的解, 那么,  $\lim_{k \rightarrow \infty} P(x_k \in R_{\epsilon, M}) = 1$ , 即算法  $D$  全局收敛<sup>[29]</sup>。其中,  $P(x_k \in R_{\epsilon, M})$  是  $x_k$  在  $R_{\epsilon, M}$  中的概率测度。

**证明** 根据条件 1, 若  $x_k \notin R_{\epsilon, M}$ ,  $\forall \ell < k$ , 有  $x_\ell \notin R_{\epsilon, M}$ , 由条件 2, 可得

$$P(x_k \in S \setminus R_{\epsilon, M}) \leq \prod_{\ell=0}^{k-1} (1 - \mu_\ell(R_{\epsilon, M})) \quad (9)$$

$$P(x_k \in R_{\epsilon, M}) = 1 - P(x_k \in S \setminus R_{\epsilon, M}) \geq 1 - \prod_{\ell=0}^{k-1} (1 - \mu_\ell(R_{\epsilon, M})) \quad (10)$$

$$1 \geq \lim_{k \rightarrow \infty} P(x_k \in R_{\epsilon, M}) \geq 1 - \lim_{k \rightarrow \infty} \prod_{\ell=0}^{k-1} (1 - \mu_\ell(R_{\epsilon, M})) = 1 \quad (11)$$

证毕

### 3.2 改进算法基本概念的数学描述

**定义 1** 萤火虫位置状态和位置状态空间

萤火虫位置的状态  $y = (x, g_b)$ , 由萤火虫的位置  $x$  和全局最优值  $g_b$  构成, 所有可能的个体位置状态的集合称为萤火虫的位置状态空间, 记为  $Y = \{y = (x, g_b) \mid x, g_b \in S, f(g_b) < f(x)\}$ 。其中, 萤火虫个体在一次迭代过程中产生的两组子种群的个体位置记为  $x_{m1}$  和  $x_{m2}$ , 其位置状态分别记为  $y_{m1} = (x_{m1}, g_b)$  和  $y_{m2} = (x_{m2}, g_b)$ ,  $Y$  为包含两子种群个体的位置状态空间。

**定义 2** 群体状态和群体状态空间

种群位置状态集合称为群体状态, 记为  $q = \{y_1, y_2, \dots, y_n\}$ , 所有群体状态构成的群体状态空间记为  $Q = \{q = (y_1, y_2, \dots, y_n), y_i \in Y, 1 \leq i \leq n\}$ , 其中子群位置状态记为  $q_{m1} = (y_{m11}, y_{m12}, \dots, y_{m1n})$  和  $q_{m2} = (y_{m21}, y_{m22}, \dots, y_{m2n})$ , 且  $Q$  包含两子群状态空间。

### 3.3 IFA 的马尔可夫模型建立

**定义 3** 对  $\forall y_i = (x_i, g_{b_i}) \in Y, \forall y_{m1i} = (x_{m1i}, g_{b_i}) \in Y$  位置状态由  $y_i$  转移到  $y_{m1i}$  记为  $T_{ym1}(y_i) = y_{m1i}$ 。

**定理 2** 萤火虫的位置状态由  $y_i$  转移到  $y_{m1i}$  的转移概率为

$$p(T_{ym1}(y_i) = y_{m1i}) = p(x_i \rightarrow x_{m1i}) \cdot p(g_{b_i} \rightarrow g_{b_i}') \cdot p(x_{m1i}' \rightarrow x_{m1i}) \cdot p(g_{b_i}' \rightarrow g_{b_i}'') \quad (12)$$

式中:  $p(x_i \rightarrow x_{m1i}')$  为式(4)对应的转移概率;  $p(g_{b_i} \rightarrow g_{b_i}')$  为该步骤全局最优位置的转移概率;  $p(x_{m1i}' \rightarrow x_{m1i})$  为步骤 4 对应的转移概率;  $p(g_{b_i}' \rightarrow g_{b_i}'')$  为该步骤更新全局最优的转移概率。

**证明** 萤火虫位置状态由  $y_i$  转移到  $y_{m1i}$ , 意味着  $x_i \rightarrow x_{m1i}', g_{b_i} \rightarrow g_{b_i}', x_{m1i}' \rightarrow x_{m1i}, g_{b_i}' \rightarrow g_{b_i}$  同时成立。作为子种群, 有

$$p(x_i \rightarrow x_{m1i}') = 1 \quad (13)$$

全局最优位置  $g_{b_i}$  的转移概率为

$$p(g_{b_i} \rightarrow g_{b_i}') = \begin{cases} 1, & f(x_{m1i}') \leq f(g_{b_i}) \\ 0, & f(x_{m1i}') > f(g_{b_i}) \end{cases} \quad (14)$$

位置状态由  $x_{m1i}' \rightarrow x_{m1i}$  的概率为

$$p(x_{m1i}' \rightarrow x_{m1i}) = \begin{cases} 1, & f(x_{m1i}') > f(x_{m1i}) \\ 0, & f(x_{m1i}') \leq f(x_{m1i}) \end{cases} \quad (15)$$

全局最优位置的再次转移概率为

$$p(g_{b_i}' \rightarrow g_{b_i}'') = \begin{cases} 1, & f(x_{m1i}) \leq f(g_{b_i}') \\ 0, & f(x_{m1i}) > f(g_{b_i}') \end{cases} \quad (16)$$

证毕

**定义 4** 对  $\forall q_i = (y_{i1}, y_{i2}, \dots, y_{in}) \in Q, \forall q_{m1i} = (y_{m1i1}, y_{m1i2}, \dots, y_{m1in}) \in Q$ , 群体状态由  $q_i$  转移到  $q_{m1i}$ , 记作  $T_{qm1}(q_i) = q_{m1i}$ 。

**定理 3** 萤火虫位置群体状态由  $q_i$  转移到  $q_{m1i}$  的转移概率为

$$P(T_{qm1}(q_i) = q_{m1i}) = \prod_{k=1}^n P(T_{ym1}(y_{ik}) = y_{m1ik}) \quad (17)$$

**证明** 若萤火虫位置群体状态由  $q_i$  转移到  $q_{m1i}$ , 那么表示  $q_i$  中所有个体的位置状态同时转移到  $q_{m1i}$  中所有个体的位置状态, 即  $T_{ym1}(y_{i1}) = y_{m1i1}, T_{ym1}(y_{i2}) = y_{m1i2}, \dots, T_{ym1}(y_{in}) = y_{m1in}$  同时成立。

证毕

**定义 5** 在 IFA 算法迭代中, 对  $\forall q_{m1i} = (y_{m1i1}, y_{m1i2}, \dots, y_{m1in}) \in Q, \forall q_{m2i} = (y_{m2i1}, y_{m2i2}, \dots, y_{m2in}) \in Q$ , 记作  $T_{qm2}(q_{m1i}) = q_{m2i}$ 。

**定理 4** 萤火虫位置群体状态由  $q_{m1i}$  转移到  $q_{m2i}$  的转移概率为

$$P(T_{qm2}(q_{m1i}) = q_{m2i}) = \prod_{k=1}^n P(T_{ym2}(q_{m1i}) = y_{m2ik}) \quad (18)$$

**证明** 若萤火虫位置群体状态由  $q_{m1i}$  转移到  $q_{m2i}$ , 那么有  $q_{m1i}$  中所有个体位置状态同时转移到  $q_{m2i}$  中所有个体位置状态, 即  $T_{ym2}(q_{m1i}) = y_{m2i1}, T_{ym2}(q_{m1i}) = y_{m2i2}, \dots, T_{ym2}(q_{m1i}) = y_{m2in}$  同时成立。在迭代过程中, 第二次变异操作个体的状态转移与种群状态转移有关, 根据式(7)有

$$\begin{cases} P(T_{ym2}(q_{m1i}) = y_{m2i}) = \delta \cdot P(g_{b_i}' \rightarrow g_{b_i}''), \\ 1/(n-1)(n-2) \leq \delta \leq 1 \\ P(g_{b_i}'' \rightarrow g_{b_i}'') = \begin{cases} 1, & f(x_{m1i}) \leq f(g_{b_i}'') \\ 0, & f(x_{m1i}) > f(g_{b_i}'') \end{cases} \end{cases} \quad (19)$$

证毕

**定义 6** 对  $\forall q_i = (y_{i1}, y_{i2}, \dots, y_{in}) \in Q, q_j = (y_{j1}, y_{j2}, \dots, y_{jn}) \in Q$ , 将萤火虫位置的群体状态  $q_i$  转移到  $q_j$  记作  $T_q(q_i) = q_j$ 。

**定理 5** 萤火虫位置群体状态由  $q_i$  转移到  $q_j$  的转移概率为

$$P(T_q(q_i) = q_j) = \prod_{k=1}^n P(T_{ym1}(y_{ik}) = y_{m1k}) \cdot \prod_{k=1}^n P(T_{ym2}(q_{m1k}) = y_{m2k}) \cdot P_{\text{sort\&-cut}} \cdot P(\text{gb}''_i \rightarrow \text{gb}_j) \quad (20)$$

式中:  $P_{\text{sort\&-cut}}$  为对父代种群和两子种群进行排序和截断操作,  $P(\text{gb}''_i \rightarrow \text{gb}_j)$  为寻找种群中的最优个体的转移概率。

**证明** 若萤火虫位置的群体状态由  $q_i$  一步转移到  $q_j$ , 就意味着位置的群体状态发生了集体转移, 由定理 1~定理 3 可推出最终的转移概率公式。证毕

**定理 6** 萤火虫的位置群体状态序列  $\{q(t); t \geq 0\}$  是离散时空上的有限齐次马尔可夫链。

**证明** 首先, 对于任何一个优化算法来讲, 其搜索空间都是有限的, 任意个体状态  $y = (x, g_b)$  中的  $x, g_b$  也是有限的, 所以位置的状态空间是有限空间。又由于  $q = (y_1, y_2, \dots, y_n)$  由  $n$  个个体组成,  $n$  为大小有限的正整数, 故萤火虫位置的群体状态  $q$  也是有限的。

其次,  $\forall q(t-1) \in Q, \forall q(t) \in Q$ , 其状态转移概率  $P(T_q(q(t-1)) = q(t))$  是由群体中所有个体的转移概率和两子群的转移概率决定。由定理 1~定理 3, 可得

$$P(T_{ym1}(y(t-1)) = y_{m1}(t-1)) = P(x(t-1) \rightarrow x_{m1'}(t-1)) \cdot P(g_b(t-1) \rightarrow g_b'(t-1)) \cdot P(x_{m1'}(t-1) \rightarrow x_{m1}(t-1)) \cdot P(g_b'(t-1) \rightarrow g_b(t-1)) \quad (21)$$

$$P(T_{qm1}(q(t-1)) = q_{m1}(t-1)) = \prod_{k=1}^n P(T_{ym1}(y_k(t-1)) = y_{m1k}(t-1)) \quad (22)$$

$$P(T_{qm2}(q_{m1}(t-1)) = q_{m2}(t-1)) = \prod_{k=1}^n P(T_{ym2}(q_{m1k}(t-1)) = y_{m2k}(t-1)) \quad (23)$$

则

$$P(T_q(q(t-1)) = q(t)) = \prod_{k=1}^N P(T_{ym1}(y_k(t-1)) = y_{m1k}(t-1)) \cdot \prod_{k=1}^N P(T_{ym2}(q_{m1k}(t-1)) = y_{m2k}(t-1)) \cdot P_{\text{sort\&-cut}} \quad (24)$$

通过以上分析,  $P(T_q(q(t-1)) = q(t))$  仅与  $t-1$  时刻的位置状态有关, 故萤火虫位置的群体状态  $\{q(t); t \geq 0\}$  具有马尔可夫性, 且算法迭代与时间无关, 所以该过程是齐次的。证毕

### 3.4 IFA 的收敛性分析

**定义 7** 定义最小化问题  $f: \mathbf{R}^n \rightarrow \mathbf{R}_f, S_f \subseteq \mathbf{R}^n$

$$\min f(x)$$

$$\text{s. t. } x \in S_f$$

式中:  $x = \{x_1, x_2, \dots, x_n\}$  为  $n$  维决策变量, 所有满足约束的

值构成的集合称为可行域  $S_f$ 。位置最优位置状态集为  $R$ , 其群体状态集为  $H = \{q = (y_1, y_2, \dots, y_n) \mid \exists y_i \in R, 1 \leq i \leq n\}$ 。

**定义 8** 对闭集  $W, \forall i \in C$ , 有  $\sum_{j \in C} P_{ij} = 1$ , 意味着从状态空间  $W$  内的任何一点出发, 始终无法达到  $W$  以外的任意状态。

**定理 7** 对于个体位置状态集  $\{y(t); t \geq 0\}$ , 集合  $R$  是空间  $Y$  上的一个闭集; 对位置群体状态集  $\{q(t); t \geq 0\}$ , 集合  $H$  是空间  $Q$  上的一个闭集。

**证明** 假如  $\forall y_i \in R, \exists y_j \notin R$ , 使得  $y_i \rightarrow y_j$  成立, 由式(14)、式(16)、式(19)、式(20), 至少保证  $g_{b_i} \rightarrow g_{b_j}, g_{b'_i} \rightarrow g_{b'_j}, g_{b''_i} \rightarrow g_{b''_j}, g_{b'''_i} \rightarrow g_{b'''_j}$  同时成立, 即转移概率均为 1, 则必有  $f(g_{b_j}) < f(g_{b_i})$ , 那么  $y_j \in R$ , 与假设条件矛盾。故假设条件不成立, 由闭集的定义知,  $R$  是  $Y$  上的一个闭集。

同理, 假如  $\forall q_i \in H, \exists q_j \notin H, q_i \rightarrow q_j$  成立, 那么至少存在一个个体的位置由  $R$  的内部转移至外部, 又已证明  $R$  是  $Y$  上的一个闭集, 根据式(20), 有  $P(T_q(q_i) = q_j) = 0$ , 由闭集定义知, 集合  $H$  是空间  $Q$  上的一个闭集。证毕

**定理 8** 在萤火虫位置群体状态空间  $Q$  中, 无法找到  $H$  外的非空闭集  $B$ , 使其满足  $B \cap H = \emptyset$ 。

**证明** 假设在位置群体空间  $Q$  中能够找到一个闭集  $B$ , 满足  $B \cap H = \emptyset$ , 那么  $\exists q_i \in H, \forall q_j \in B$ , 有  $f(g_{b_j}) > f(g_{b_i})$ 。根据式(20), 其满足各个组件不为 0 的条件, 那么存在转移概率  $P(T_q(q_i) = q_j) \neq 0$ , 即  $B$  不是闭集。定理得证。

证毕

**引理 1** 假设  $\langle u_n \rangle$  是一组服从  $\text{rand}(0, 1)$  的随机变量, 且满足独立同分布, 那么

$$\lim_{n \rightarrow +\infty} \frac{1}{n} \sum_{i=1}^n u_i = \int_0^1 s ds$$

**引理 2** 假设马尔可夫链存在一个非空闭集  $C$ , 且无法找到满足  $C \cap D = \emptyset$  的另一个非空闭集  $D$ , 那么在  $j \in C$  时,  $\lim_{n \rightarrow +\infty} P(x_n = j) = \pi_j$ , 且在  $j \notin C$  时,  $\lim_{n \rightarrow +\infty} P(x_n = j) = 0$ 。

**定理 9** 随着萤火虫位置群体状态不断迭代更新, 当次数趋于无穷时, 萤火虫位置群体状态序列必然进入最优集  $H$ 。

**证明** 由定理 6 和定理 7 以及引理 1 和引理 2 得证。

**定理 10** 在迭代次数趋于无穷时, IFA 能够以概率 1 收敛于全局最优。

**证明** 一方面, 在每次迭代过程中, IFA 都保留了全局最优, 从而保证了适应度的非增性, 满足收敛条件一; 另一方面, 根据定理 9, 当迭代次数趋于无穷时, 萤火虫位置的群体序列必然进入最优集, 即找不到全局最优的概率为 0, 满足收敛条件二。故 IFA 以概率 1 收敛于全局最优。证毕

## 4 仿真实验

由于实际应用中算法运行时间有限, 故需通过固定时间的仿真实验对算法有效性进行验证。算法运行环境为 Intel Core i5-3470 处理器, CPU 3.20 GHz, 3.48 GB 内存, Windows 10, 64 位操作系统。



4.1 随机扰动因子性能对比

为验证第 2.1 节中的结论,对 4 种随机扰动因子下的改进算法 RIFA, GIFA, CIFA, LIFA 进行仿真试验和性能对比,R(平均)、G(正态)、C(Cauchy)、L(Levy)分别对应改进算法采用的随机分布。选用 4 个典型基准函数,如表 1 所示。

算法参数设置情况为  $\beta_0=1, \beta_{re}=0.8, \gamma=1, n_{Mutation}=3, n=40, n_{DE}$ 取种群规模  $n, d=30, F \sim U(0.4, 0.9, 1, d)$ ,最大迭代次数为 2 000。初始种群由问题域内进行随机初始化产生。为提高结果的可信度,保证对比实验客观公正,对每一个测试,所有算法均独立执行 50 次,取平均最优值、时间和成功率(success rate, SR)的统计结果进行比较,最优结果由粗体标出,仿真结果如表 2 所示。

表 2 4 种随机扰动因子仿真结果  
Table 2 Four random disturbance factors simulation results

扰动因子	$f_1$			$f_2$			$f_3$			$f_4$		
	最优值	时间/s	SR/%	最优值	时间/s	SR/%	最优值	时间/s	SR/%	最优值	时间/s	SR/%
GIFA	2.301 3e-154	95.899 8	100	4.146 1e-09	82.201	0	0	15.544 3	10	0	24.261 5	10
CIFA	3.051 1e-109	41.303 4	100	0.001 065 2	46.112 3	0	60.357 1	17.461 3	0	0	24.782 4	50
LIFA	3.877 7e-105	534.637 7	100	1.060 1e-04	145.988 4	0	95.542 1	15.591 9	0	0	38.973 5	10
RIFA	<b>0</b>	<b>25.457 8</b>	<b>100</b>	<b>1.384 9e-27</b>	<b>21.950 9</b>	<b>70</b>	<b>0</b>	<b>11.521 1</b>	<b>100</b>	<b>0</b>	<b>13.181 7</b>	<b>100</b>

表 2 的统计结果显示,随机扰动因子选择均匀分布时,在寻优精度、收敛时间和 SR 上的表现最好,对所有基准函数均能找到最优解;选择正态分布时,算法收敛时间大幅提高,寻优精度和 SR 也出现了下降,特别是函数  $f_2 \sim f_4$  的表现很差;柯西扰动因子的表现与正态分布类似,收敛时间比前者稍短,但寻优精度更加无法保证;列维扰动因子表现最差,运行效率最低,优化结果几乎不可用。因此,改进算法的随机扰动因子优先选择均匀分布。

表 3 固定迭代次数仿真结果  
Table 3 Simulation results of fixed iterations

算法	$f_1$			$f_2$			$f_3$			$f_4$		
	最优值	时间/s	SR/%	最优值	时间/s	SR/%	最优值	时间/s	SR/%	最优值	时间/s	SR/%
PSO	1.455 7e-57	2.909 6	100	19.565 6	2.373 9	0	26.863 9	2.149 3	0	3.330 7e-16	2.647 6	23.3
DE	5.758 9e-29	2.696 9	100	24.598 4	2.799 9	0	23.737 9	2.290 8	0	<b>0</b>	3.038 3	70
FA	8.378 4e-88	14.653 8	100	0.007 324 5	8.805 5	0	21.838 7	1.185	0	<b>0</b>	3.024 9	90
IFA	<b>0</b>	25.457 8	<b>100</b>	<b>2.258 8e-26</b>	21.950 9	<b>70</b>	<b>0</b>	11.521 1	<b>100</b>	<b>0</b>	13.181 7	<b>100</b>

表 3 的统计结果显示,对于函数  $f_1 \sim f_4$ ,改进算法均能够找到最优解、精度和 SR 均好于对比方法,在  $f_2$  和  $f_3$  尤为明显。但在平均仿真时间上,改进算法比其他算法长。这是由于 PSO、DE 和 FA 过早陷入局部最优,根据优化算法本身运行机制,如果没有足够的跳出局部最优的能力,那么会不断加速收敛,因此其运行时间短于改进算法。

函数  $f_2$  是典型的复杂病态二次函数,其全局最优位于狭窄的、抛物形的山谷中,缺乏有价值的信息,优化难度较大,比较考验算法全局探索和局部挖掘能力。各方法对该函数进行优化的寻优曲线如图 6 所示。

表 1 基准函数

Table1 Benchmarks functions

函数名	数学表达式	规模	最优值
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	$\pm 100$	0
Rosenbrock	$f_2(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$\pm 30$	0
Rastrigin	$f_3(x) = \sum_{i=1}^D [x_i^2 - 10\cos(2\pi x_i + 10)]$	$\pm 5.12$	0
Griewank	$f_4(x) = \frac{1}{4\,000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	$\pm 600$	0

4.2 与其他算法的性能对比

选取已在实践中检验过的优秀算法 PSO、DE 以及 FA 进行对比试验。各个算法的参数设置情况如下:

- (1) PSO:惯性权重  $\omega=1$ , 阻尼比  $\omega_{damp}=0.99$ , 加速因子  $c_1=1.5, c_2=2.0$  [30]。
  - (2) DE(DE/rand/1):缩放因子 F 为范围在  $0.2 \sim 0.8$  的均匀随机数,变异概率  $p_{cr}=0.2$  [31]。
  - (3) FA: $\alpha=0.2, \alpha_{damp}=0.98, \beta_0=2, \gamma=1$ 。
  - (4) IFA 参数同第 4.1 节。
- 其他仿真条件同第 4.1 节,仿真结果见表 3。

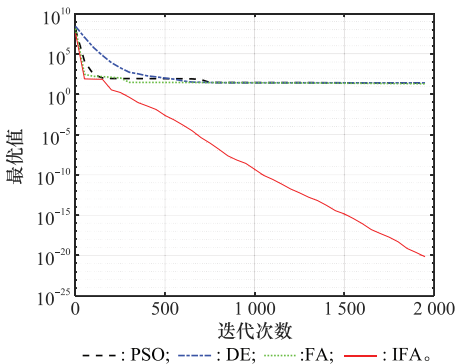


Fig. 6 Performance comparison of different algorithms

由图 6 可以看出,PSO、DE 和 FA 均出现了早熟收敛现象,而 IFA 在短暂收敛后,在迭代到 151 步时成功跳出局部最优并最终达到极高的优化精度。现给出算法迭代 150 步到 151 步时的个体分布情况。由于个体维数较高,因此通过评估每个个体与平均个体在搜索空间中的相对距离得到个体分布散点图,如图 7 所示。

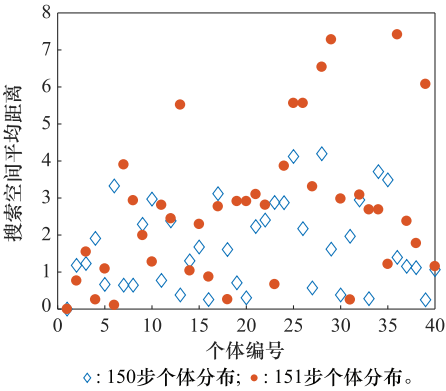


图 7 典型迭代时刻个体分布

Fig. 7 Individual distribution of typical iteration time

可以看出,种群在 151 步时部分个体突变至距离平均个体较远的位置,获得了更优解的同时,带动种群整体移动,整体的多样性比 150 步时有了明显的提升。通过以上仿真和分析,证明了所提改进策略在基准函数优化方面的有效性。

4.3 装箱问题实例

本文将 IFA 用于求解装箱问题。装箱问题是一个经典 NP-hard 组合优化问题<sup>[32]</sup>,在运筹学、离散数学、计算机科学以及工业生产等方面有着广泛的应用。

设有足够数量的结构相同容量一致的箱子  $B_1, B_2, \dots$ , 每个箱子的最大容量为  $C_{\max}$ 。现有一组  $n$  个物品  $I_1, I_2, \dots, I_n$ , 体积分别为  $v_j (0 < v_j < C_{\max}, j = 1, 2, \dots, n)$ , 需装入箱内,要求在满足约束的情况下打包所有物品,要求所需箱子数量尽可能的少。两个约束分别为:① 所有物体均不可分割,需保持完整并恰好放入一个箱子中;② 每个箱子所放物品体积总和均不能超过容量上限。假设物品形状对装箱没有影响,则该问题可描述为

$$\min z = \sum_{i=1}^n y_i$$

$$\text{s. t. } \begin{cases} \sum_{j=1}^n v_j x_{ij} \leq C y_i, i = 1, 2, \dots, n \\ \sum_{i=1}^n x_{ij} = 1, j = 1, 2, \dots, n \end{cases} \quad (25)$$

其中,

$$y_i = \begin{cases} 1, \text{箱子 } B_i \text{ 被使用} \\ 0, \text{否则} \end{cases}$$

$$x_{ij} = \begin{cases} 1, \text{物品 } J_j \text{ 放入箱子 } B_i \text{ 中} \\ 0, \text{否则} \end{cases}$$

假设 30 个物品体积分别为  $[6, 3, 4, 6, 8, 7, 4, 7, 7, 5, 5, 6, 7, 7, 6, 4, 8, 7, 8, 8, 2, 3, 4, 5, 6, 5, 5, 7, 7, 12]$ , 箱子容量  $C=30$ 。固定迭代 1 000 步,种群规模为 20,其余参数不变。优化结果与 PSO、DE 和 FA 比较,算法收敛情况如图 8 所示。

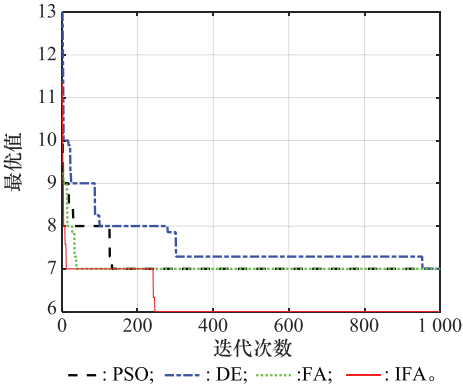


图 8 装箱问题的性能比较

Fig. 8 Comparison on bin packing problem

由图 8 可以看出,IFA 仅需 12 步即可达到其他 3 种算法的优化效果,并且经进一步迭代后能够跳出局部最优,从而得到更优的结果。表 4 给出了几种优化算法的总体运行时间(total time, TT)、收敛时间(convergence time, CT)和最优结果(Best)。比较可知,IFA 在优化结果和收敛速度方面都有着明显的优势。

表 4 装箱问题优化结果

Table 4 Optimization results of bin packing problem

参数	算法			
	PSO	DE	FA	IFA
TT/s	43.070 4	13.112 2	21.155 9	22.640 1
CT/s	5.728 4	12.495 9	1.404 3	5.569 5
Best	7	7	7	6

5 结 论

本文为克服 FA 易陷入局部最优的缺陷,提出了 IFA。首先分析了个体位置更新机制的特点,提出了一种随机扰动策略,阐述了改进策略的有效性,并分析了不同扰动因子选取的可行性。通过引入元素置换变异策略和最优差分变异策略,提高了算法跳出局部最优的能力。基于随机算法收敛准则和马尔可夫过程的理论分析指出,改进算法在足够大的迭代次数下能够以概率 1 收敛到全局最优。对 4 种基准函数的仿真结果表明,随机扰动因子选择均匀分布时有最佳表现。在相同条件下与几种经典算法相比,改进算法的收敛精度和 SR 更高,验证了改进算法的优越性。将该方法对经典装箱问题进行求解,得到了较好的优化结果,验证了改进方法的有效性和实用价值。



## 参考文献

- [1] GHAREHCHOPOGH F S, GHOLIZADEH H. A comprehensive survey: whale optimization algorithm and its applications[J]. *Swarm and Evolutionary Computation*, 2019, 48: 1–24.
- [2] COLORNI A, DORIGO M, MANIEZZO V. Distributed optimization by ant colonies[C]//*Proc. of the First European Conference on Artificial Life*, 1991: 134–142.
- [3] KENNEDY J, EBERHART R. Particle swarm optimization[C]//*Proc. of the IEEE International Conference on Neural Networks*, 1995: 1942–1948.
- [4] KARABOGA D, BASTURK B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm[J]. *Journal of Global Optimization*, 2007, 39(3): 459–471.
- [5] SAYADI M K, HAFEZALKOTOB A, NAINI S G J. Firefly-inspired algorithm for discrete optimization problems: an application to manufacturing cell formation[J]. *Journal of Manufacturing Systems*, 2013, 32(1): 78–84.
- [6] YANG X S. *Nature-inspired metaheuristic algorithms*[M]. Frome: Luniver Press, 2008.
- [7] DARWISH S M. Combining firefly algorithm and Bayesian classifier: new direction for automatic multilabel image annotation[J]. *IET Image Processing*, 2016, 10(10): 763–772.
- [8] 张笑笑, 丁洁玉. 萤火虫算法求解多体系统动力学微分-代数方程[J]. *动力学与控制学报*, 2021, 19(2): 85–90.  
ZHANG X X, DING J Y. Firefly algorithm for multi-body system dynamics differential-algebraic equations[J]. *Journal of Dynamics and Control*, 2021, 19(2): 85–90.
- [9] TRIPATHI S. A brief review of firefly algorithm: application in structural optimization problem[J]. *Journal of the Institute of Engineering*, 2020, 15(2): 183–191.
- [10] BABBAR A, PRAKASH C, GUPTA M K, et al. Application of hybrid nature-inspired algorithm: single and bi-objective constrained optimization of magnetic abrasive finishing process parameters[J]. *Journal of Materials Research and Technology*, 2020, 9(4): 7961–7974.
- [11] GANDOMI A H, YANG X S, TALATAHARI S, et al. Firefly algorithm with chaos[J]. *Communications in Nonlinear Science and Numerical Simulation*, 2013, 18(1): 89–98.
- [12] YU S H, YANG S L, SU S B. Self-adaptive step firefly algorithm[J]. *Journal of Applied Mathematics*, 2013: 832718.
- [13] ALTAY E Y, ULUTA G. Self-adaptive step firefly algorithm based robust watermarking method in DWT-SVD domain[J]. *Multimedia Tools and Applications*, 2021, 80: 23457–23484.
- [14] VERMA O P, AGGARWAL D, PATODI T. Opposition and dimensional based modified firefly algorithm[J]. *Expert Systems with Applications*, 2016, 44: 168–176.
- [15] MANJU A, NIGAM M J. An improved quantum inspired firefly algorithm with interpolation operator[J]. *Advances in Intelligent Systems and Computing*, 2014, 258: 69–84.
- [16] 柳长源, 任宇艳, 毕晓君. 基于改进萤火虫算法的区域交通信号配时优化[J]. *控制与决策*, 2020, 35(12): 2829–2834.  
LIU C Y, REN Y Y, BI X J. Timing optimization of regional traffic signals based on improved firefly algorithm[J]. *Control and Decision*, 2020, 35(12): 2829–2834.
- [17] 刘景森, 毛艺楠, 李煜. 具有振荡约束的自然选择萤火虫优化算法[J]. *控制与决策*, 2020, 35(10): 2363–2371.  
LIU J S, MAO Y N, LI Y. Natural selection firefly optimization algorithm with oscillation and constraint[J]. *Control and Decision*, 2020, 35(10): 2363–2371.
- [18] WANG H, WANG W J, ZHOU X Y, et al. Firefly algorithm with neighborhood attraction[J]. *Information Sciences*, 2017, 382: 374–387.
- [19] DHAL K G, DAS A, RAY S, et al. Randomly attracted rough firefly algorithm for histogram based fuzzy image clustering[J]. *Knowledge-Based Systems*, 2021, 216(1): 106814.
- [20] SHAN J, PAN J S, CHANG C K, et al. A distributed parallel firefly algorithm with communication strategies and its application for the control of variable pitch wind turbine[J]. *ISA Transactions*, 2021, 115: 79–94.
- [21] ELAKKIYA E, SELVAKUMAR S, VELUSAMY R L. CIFAS: community inspired firefly algorithm with fuzzy cross-entropy for feature selection in Twitter Spam detection[C]//*Proc. of the 11th International Conference on Computing, Communication and Networking Technologies*, 2020.
- [22] ZHAO J, CHEN W P, XIAO R B, et al. Firefly algorithm with division of roles for complex optimal scheduling[J]. *Frontiers of Information Technology and Electronic Engineering*, 2021, 22(10): 1311–1333.
- [23] XING H X, WU H, CHEN Y, et al. A cooperative interference resource allocation method based on improved firefly algorithm[J]. *Defence Technology*, 2021, 17(4): 1352–1360.
- [24] 薛晗, 邵哲平, 潘家财, 等. 基于文化萤火虫算法-广义回归神经网络的船舶交通流量预测[J]. *上海交通大学学报*, 2020, 54(4): 421–429.  
XUE H, SHAO Z P, PAN J C, et al. Vessel traffic flow prediction based on CFA-GRNN algorithm[J]. *Journal of Shanghai Jiaotong University*, 2020, 54(4): 421–429.
- [25] GUPTA D K, SONI A K, JHA A V, et al. Hybrid gravitational-firefly algorithm-based load frequency control for hydro-thermal two-area system[J]. *Mathematics*, 2021, 9(7): 712.
- [26] CHENG Z W, SONG H H, CHANG T Z, et al. An improved mixed-coded hybrid firefly algorithm for the mixed-discrete SSCGR problem[J]. *Expert Systems with Applications*, 2022, 188: 116050.
- [27] XUE X S, CHEN J F. Optimizing sensor ontology alignment through compact co-firefly algorithm[J]. *Sensors*, 2020, 20(7): 2056.
- [28] WETS J B. Minimization by random search techniques[J]. *Mathematics of Operations Research*, 1981, 6(1): 19–30.
- [29] 刘晓东, 孙丽君, 陈天飞. 布谷鸟算法的收敛性分析及性能比较[J]. *计算机科学与探索*, 2020, 14(10): 1644–1655.  
LIU X D, SUN L J, CHEN T F. Convergence analysis and

performance comparison of cuckoo search algorithm[J]. Journal of Frontiers of Computer Science and Technology, 2020, 14(10): 1644–1655.

[30] JIAO B, LIAN Z G, GU X S. A dynamic inertia weight particle swarm optimization algorithm[J]. Chaos, Solitons & Fractals, 2008, 37(3): 698–705.

[31] DAS S, KONAR A, CHAKRABORTY U K. Improved differential evolution algorithms for handling noisy optimization problems[C]//Proc. of the IEEE Congress on Evolutionary Computation, 2005: 1691–1698.

[32] JOHNSON D S. Near-optimal bin packing algorithms[D]. Massachusetts: Massachusetts Institute of Technology, 1973.

## 作者简介

**张大力**(1991—),男,博士研究生,主要研究方向为智能优化、轨迹规划、轨道动力学。

**夏红伟**(1979—),男,教授,博士,主要研究方向为控制理论与控制方法、导航制导与控制、航天器地面仿真。

**张朝兴**(1982—),男,高级工程师,硕士,主要研究方向为航天器仪器仪表、电信技术、航天器地面仿真。

**马广程**(1972—),男,教授,博士,主要研究方向为航天器地面仿真、运动控制。

**王常虹**(1961—),男,教授,博士,主要研究方向为控制理论与控制方法、导航制导与控制。