

引入Lévy flight和萤火虫行为的鱼群算法

殷红¹, 董康立¹, 彭珍瑞^{1†}, 李少远²

(1. 兰州交通大学 机电工程学院, 甘肃 兰州 730070; 2. 上海交通大学 电子信息与电气工程学院, 上海 200240)

摘要: 针对人工鱼群算法(artificial fish-swarm algorithm, AFSA)和萤火虫算法(firefly algorithm, FA)在多维多极值函数寻优过程中易陷入局部最优和精度有待提高等问题, 提出引入Lévy flight和萤火虫行为的鱼群算法(fish swarm algorithm with Lévy flight and firefly behavior). 该算法将萤火虫算法中萤火虫个体的移动策略引入到鱼群的聚群、觅食两种行为模式中, 进而将Lévy flight引入到鱼群的搜索策略中, 使得鱼群的搜索更加高效. 此外, 采取一种基于动态参数的非线性变视野和变步长的策略来限定鱼群的搜索范围. 仿真分析表明, 新算法较其他测试算法具有更好的全局搜索能力和寻优精度.

关键词: 人工鱼群算法; 萤火虫算法; Lévy flight搜索策略; 行为模式

引用格式: 殷红, 董康立, 彭珍瑞, 等. 引入Lévy flight和萤火虫行为的鱼群算法. 控制理论与应用, 2018, 35(4): 497 – 505

中图分类号: TP18 文献标识码: A

Fish swarm algorithm with Lévy flight and firefly behavior

YIN Hong¹, DONG Kang-li¹, PENG Zhen-rui^{1†}, LI Shao-yuan²

(1. School of Mechatronic Engineering, Lanzhou Jiaotong University, Lanzhou Gansu 730070, China;

2. School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract: Since the artificial fish-swarm algorithm (AFSA) and firefly algorithm (FA) are easily converging to local optimum and have low accuracy in the optimization process for solving multi-dimensional and multi-extreme value functions, an algorithm called Fish swarm algorithm with Lévy flight and firefly behavior (LFFSA) is proposed, which introduces the migration strategy of firefly algorithm into the two behavior patterns of fish swarm as: the swarming and the preying behaviors. Furthermore, the Lévy flight is introduced into the search strategy. Besides, nonlinearity visual and step length based on dynamic parameter are simultaneously considered for limiting the search band. Simulation results demonstrate that the LFFSA has a better performance in convergence speed and optimization accuracy.

Key words: artificial fish-swarm algorithm; firefly algorithm; Lévy flight migration strategy; behavior pattern

Citation: YIN Hong, DONG Kangli, PENG Zhenrui, et al. Fish swarm algorithm with Lévy flight and firefly behavior. *Control Theory & Applications*, 2018, 35(4): 497 – 505

1 引言(Introduction)

优化问题在现代工业和科技领域中发挥着越来越重要的作用. 考虑到现有数值方法的计算成本, 研究人员不得不依靠智能算法来解决一些复杂的优化问题. 智能算法有着不同于传统优化算法的特点: 基于种群的集体学习过程, 自适应性和鲁棒性等等. 近年来, 混合算法由于其较强的解决各种复杂问题的能力得到了广泛的关注^[1].

对于一些简单问题来讲, 基本算法完全能够满足解决问题的需求; 但是对于更为复杂的问题(如NP-hard问题), 基本算法往往不能得到理想的最优解. Wolpert和Macready^[2]在其论文中阐述到, 如果算法A

在某个测试函数中表现优于算法B, 那么一定存在某个测试函数的寻优, 算法B优于算法A. 这便要求进化算法、群体智能算法、机器学习方法以及启发式算法能够取长补短, 以达到获得不同类型问题最优解的目的. 考虑到基本算法无法解决复杂的现实问题, 而混合算法能够改善基本算法的性能(如收敛速度等), 提高迭代产生解的质量, 并且能够将基本算法作为组成部分容纳到一个更大的系统中, 因此混合算法有其存在的必要性和意义.

人工鱼群算法(artificial fish-swarm algorithm, AFSA)是李晓磊等人于2002年提出的一种模拟鱼群特性的群体智能优化算法^[3], 萤火虫算法(firefly algorithm,

收稿日期: 2017-08-03; 录用日期: 2017-11-23.

[†]通信作者. E-mail: pzrui@163.com; Tel.: +86 931-4955726.

本文责任编辑: 丛爽.

国家自然科学基金项目(61463028)资助.

Supported by the National Natural Science Foundation of China (61463028).

FA)是Yang于2009年提出的一种模拟萤火虫群特性的高级启发式算法^[4],现有的仿真实验表明,两种基本算法均具有较高的全局搜索能力.然而,这两种基本算法在多维复杂函数的寻优过程中的效果仍有待提升.目前已有许多针对人工鱼群算法的改进算法,如引入类似粒子群算法中的速度参数^[5]、引入变异算子并与模拟退火相结合^[6]、引入新的行为模式^[7-8]等.对于萤火虫算法,其改进和应用仍然是当前研究热点^[9-15].

人工鱼群算法(AFSA)采用聚群行为和追尾行为并行寻优的方式,具有简单快速的特点.但每条人工鱼确定移动方向后按随机步长移动会造成鱼群无法有效地逼近目标点;同时,当人工鱼不满足上述两种行为的执行条件时会进行觅食行为,即在视野内随机搜索,这种搜索方式低效且易错过最优值.

针对鱼群算法以上两个缺点改进并提出混合算法LFFSA:首先,将每条人工鱼赋予类似萤火虫吸引度的定义,使得人工鱼按照彼此间的吸引度大小进行移动,这样人工鱼能够有效逼近移动目标,提高搜索精度;其次,将进行觅食行为的人工鱼赋予Lévy flight搜索策略,使得人工鱼像果蝇一样进行短距离和偶尔长距离相间的搜索,这样进行觅食行为的一部分人工鱼在当前最优附近搜索,其他少数人工鱼在离当前最优足够远处搜索,有效地避免了陷入局部最优值.

2 人工鱼群算法和萤火虫算法(Artificial fish-swarm algorithm and firefly algorithm)

2.1 人工鱼群算法(Artificial fish-swarm algorithm)

人工鱼群算法是通过模拟人工鱼的觅食、聚群、追尾等行为在搜索域中进行寻优的一种群体智能算法.一条人工鱼个体的当前位置为 X ,其在某时刻视点所在的位置为 X_v ,各位置的食物浓度和适应度函数成正比,鱼群的拥挤度因子为 δ ,人工鱼个体视野范围为Visual,人工鱼个体最大移动步长为Step,鱼群行为模式描述如下^[1]:

觅食行为:人工鱼个体在其视野范围内随机选取一个位置 X_j ,若该位置所具有的食物浓度优于当前所在位置,则人工鱼向该位置移动一步,否则重新选择 X_j 并重新进行判断.如果反复尝试Trynumber次之后仍然不满足移动条件,则该人工鱼个体在视野范围内随机移动一步.

聚群行为:人工鱼个体探索其视野范围内的伙伴数量 nf 及其中心位置 X_c ,若该位置处有较多的食物且不太拥挤,则向该位置随机移动一步,否则转而执行觅食行为.

追尾行为:人工鱼个体探索其视野范围内的伙伴数量 nf 以及拥有最多食物的位置 X_{max} ,若该位置周围不太拥挤,则向该位置随机移动一步,否则转而执

行觅食行为.

人工鱼会选择觅食和聚群中较优者执行,当两者都不满足执行要求时,转而执行觅食行为.每次迭代产生的最优位置及其所对应的食物浓度都会置于“公告板”中,当达到预设的最大迭代次数时,迭代停止,公告板上的最优值便是算法寻优得到的结果.

2.2 萤火虫算法(Firefly algorithm)

萤火虫算法是通过每只萤火虫利用自身发出的荧光信号在特定的范围内吸引其他萤火虫,最终实现大部分萤火虫形成聚集现象从而实现寻优的一种群体智能算法.

萤火虫之间的相互吸引主要依靠荧光亮度和吸引度,其中荧光亮度取决于萤火虫所在位置的适应度值,并且与适应度值成正比;吸引度与萤火虫的荧光亮度成正比,荧光亮度越强,吸引度越高.此外,荧光亮度和吸引度又与萤火虫之间的距离成反比,两只萤火虫之间的距离越远,荧光亮度和吸引度越低.

荧光亮度 $I(r)$ 随距离 r 以指数方式单调变化:

$$I = I_0 e^{-\gamma r}, \quad (1)$$

其中: I_0 是初始荧光亮度; γ 是光照吸收系数.萤火虫的吸引度 β 与相邻萤火虫间的荧光亮度 I 成比例:

$$\beta = \beta_{a0} \cdot e^{-\gamma r^2}, \quad (2)$$

其中 β_{a0} 是 $r = 0$ 时的初始吸引度.

3 Lévy flight

Lévy flight^[16]取名自法国数学家Paul Pierre Lévy,是一种随机搜索策略.其飞行步长满足重尾(heavy-tailed)的Lévy分布,Lévy分布通常以下形式出现:

$$L(s) \sim |s|^{1-\beta}, \quad (3)$$

其中 s 为随机的Lévy步长.在搜索大范围的未知空间的问题上,Lévy随机运动的方差可以比布朗运动的方差增加得更快,因此Lévy flight也比布朗运动更加有效.

自然界动物的觅食行为是一种随机移动.由于下一次的移动往往取决于当前位置以及往下一个位置移动的可能性,因此每次随机移动的有效性就显得更加重要.近期的许多研究表明大多数动物和昆虫的飞行行为呈现出Lévy flight的飞行特点.在许多群智能搜索算法中,Lévy flight已被证明是随机移动模型中最好的几种移动策略之一.

4 引入Lévy flight和萤火虫行为的鱼群算法(Fish swarm algorithm with Lévy flight and firefly behavior)

LFFSA的具体改进描述如下:

以觅食行为为例,基本人工鱼群算法中人工鱼的位置更新公式为

$$X_{\text{next}} = X_i + \text{rand} \cdot \frac{\text{Step} \times (X_j - X_i)}{\text{norm}(X_j - X_i)}, \quad (4)$$

其中: X_{next} 为人工鱼的下一个位置; X_i 为人工鱼当前位置; X_j 为人工鱼在视野中随机选择且目标函数值优于当前位置的位置; rand 为 $[-1, 1]$ 内均匀分布的随机数; Step 为最大移动步长; $\text{norm}(X_j - X_i)$ 为两个位置矢量之间的距离。

1) 首先, 将每一条人工鱼引入类似于萤火虫算法中吸引度的定义:

$$\text{beta}_{ij} = \text{beta}_0 \cdot e^{-\gamma r_{ij}}, \quad (5)$$

其中: $e^{-\gamma r_{ij}}$ 为人工鱼 i 和 j 之间的空间距离; γ 为光强吸收系数, 可设为常数; beta_{ij} 为 $r_{ij} = 0$ 时人工鱼的吸引度, 即最大吸引度。

2) 其次, 代替上述式(4), 人工鱼 X_i 依赖吸引度向 X_j 进行移动的位置更新公式为

$$X_{\text{next}} = X_i + \text{beta}_{ij}(X_j - X_i) + \alpha(\text{rand} - 0.5), \quad (6)$$

其中: X_i 表示萤火虫 i 当前的空间位置; $\text{beta}_{ij}(X_j - X_i)$ 表示人工鱼 i 向人工鱼 j 移动相应的距离; $\alpha(\text{rand} - 0.5)$ 表示位置更新中的扰动机制, 避免人工鱼过早地陷入局部最优, α 为步长因子, 取 $[0, 1]$ 之间的常数, rand 为 $[0, 1]$ 内均匀分布的随机数。

3) 再次, 在式(6)中引入一种线性变化的惯性权重后位置更新公式为

$$X_{\text{next}} = \omega_t X_i + \text{beta}_{ij}(X_j - X_i) + \alpha(\text{rand} - 0.5), \quad (7)$$

$$\omega_t = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \cdot \text{gen} / \text{Maxgen}, \quad (8)$$

其中: ω_t 表示人工鱼对其所处的上一个位置继承的权重的大小; $\omega_{\max}$ 表示最大权重; $\omega_{\min}$ 表示最小权重; gen 表示寻优当前的迭代次数; Maxgen 表示设定的最大迭代次数。

4) 同理, 聚群行为中人工鱼的位置更新公式为

$$X_{\text{next}} = \omega_t X_i + \text{beta}_{ij}(X_c - X_i) + \alpha(\text{rand} - 0.5), \quad (9)$$

其中: X_c 表示人工鱼视野范围内的中心位置; 其他参数含义同式(6)(7)。

5) 将 Lévy flight 作为处于觅食行为的人工鱼的搜索策略。该搜索策略可以用以下公式来表示:

$$X_j = X_i + \alpha \oplus L(\beta), \quad (10)$$

$$L(\beta) = \frac{\phi \times \mu}{|v|^{\frac{1}{\beta}}}(X_i - X_{\text{best}}), \quad (11)$$

其中: X_i 表示人工鱼 i 当前的空间位置; X_j 表示人工鱼搜索到的位置; α 为步长因子; $\oplus$ 表示点对点乘法; $L(\beta)$ 表示由 Lévy flight 生成的随机向量; X_{best} 表示当前公告板上的最优位置; $\mu = t^{-\beta}$, $1 < \beta < 3$; μ 和 v 服从正态分布: $\mu \sim N(0, \phi^2)$, $v \sim N(0, 1)$, 其中

$$\phi = \left\{ \frac{\Gamma(1 + \beta) \sin(\frac{\pi\beta}{2})}{\Gamma[\frac{(1 + \beta)}{2}] \beta \cdot 2^{\frac{(\beta-1)}{2}}} \right\}^{\frac{1}{\beta}}, \quad (12)$$

其中 Γ 是标准的 Gamma 函数。

6) 考虑到太多的行为模式反而会降低算法效率, 为了减小算法的时间复杂度, 本算法放弃了鱼群的追尾行为, 只采取聚群行为和觅食行为进行寻优。

7) 此外, LFFSA 采取一种基于动态参数的非线性变视野和变步长的策略来限定鱼群的搜索范围, 视野和步长更新公式为

$$\text{Visual} = \rho \cdot \text{Visual} + \text{Visual}_{\min}, \quad (13)$$

$$\text{Step} = \rho \cdot \text{Step} + \text{Step}_{\min}, \quad (14)$$

$$\rho = \exp(-30 \times (\frac{\text{gen}}{\text{Maxgen}})^s), \quad (15)$$

其中: $\text{Visual}_{\min}$ 表示最小视野范围, 本文中取为 0.001; $\text{Step}_{\min}$ 表示最小步长, 本文中取为 0.0002; ρ 表示一种非线性变化的权重值; s 为大于 1 的整数, 其通常取值范围为 $[1, 30]$, s 不同的取值会造成权重值呈现出不同的变化趋势, 如图 1 所示。

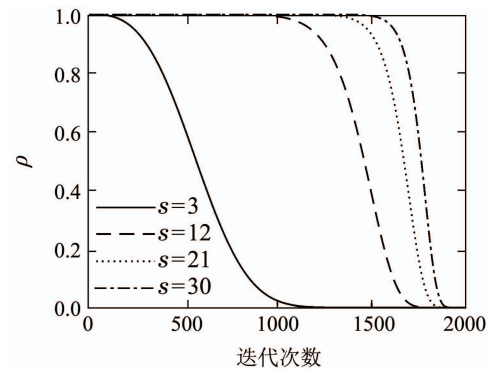


图 1 ρ 值变化曲线

Fig. 1 Value of ρ

ρ 值的大小表示当前视野和步长对于前一时刻的继承比重, ρ 值越快地趋近于 0, 说明当前视野和步长对于前一时刻的继承比重越快地变小。对于单峰的, 即迭代后期需要精细化搜索的测试函数来说, ρ 值需要更快地变小 (即 s 值取较小的值) 来适应更精细的搜索; 对于多峰的, 即迭代前期需要逃脱局部最优的测试函数来说, ρ 值则需要缓慢变小 (即 s 值取较大的值) 使得视野和步长不至于太小以致于陷入局部最优。针对多个不同特点的测试函数, 本文统一取 s 为 3; 其他参数含义同式(8)。

8) 此外, 定义参数

$$\max D = \sqrt{(x_{\max} - x_{\min})^2 \times D} \quad (16)$$

表示当搜索空间为 D 维时两条人工鱼能达到的最大距离, 其中 $x_{\max}$ 和 $x_{\min}$ 表示寻优范围的上下界。初始 Visual 值为 $\max D$ 的值, 初始 Step 为 $\max D/8$, 之后按照

式(13)–(15)进行动态变化.

LFFSA伪代码如下所示.

Algorithm: LFFSA

Begin

初始化 $\{X_1, X_2, \dots, X_n\}$;

计算适应度;

while (gen < Maxgen)或者(满足迭代精度要求)

while $i \leq n$

if X_i 满足聚群行为的条件

根据吸引度进行聚群并得到新的 X_i ;

else 根据Lévy flight进行觅食并得到 X_i ;

end

endwhile

更新“公告板”;

endwhile

得到最优解.

End

5 数值仿真实验(Numerical simulation)

5.1 参数设置(Parameter settings)

为了验证所提出的优化算法的有效性,进行了大量的计算机数值仿真实验,将相同参数设置下5种不同算法的寻优结果进行比较.仿真环境为MATLAB-R2014b,测试PC为Intel Core i5 处理器,运行内存4.0 GB.5种算法的参数设置如表1所示(混合算法与基本算法采用相同的参数):

表 1 参数设置

Table 1 Parameter settings

算法	参数	取值
FA, LFA, FFSA, LFFSA	β_0	1.0
FA, LFA, FFSA, LFFSA	γ	1.0
AFSA, FFSA, LFFSA	δ	0.618
AFSA, FFSA, LFFSA	Trynumber	5
FA, LFA, AFSA, FFSA, LFFSA	Maxgen	2000
FA, LFA, AFSA, FFSA, LFFSA	Population	40

5.2 固定迭代次数时收敛精度比较(Convergence accuracy comparison with the same iteration)

将拥有吸引度定义鱼群算法(FFSA),带有Lévy flight的萤火虫算法(LFA),以及两种基本算法作为对比,选取如表2所示的9个测试函数作为目标函数来对LFFSA进行仿真测试,以此来比较各个算法之间的性能差异.同时,对基本人工鱼群算法进行变视野和步长的改进以观察萤火虫行为和Lévy飞行的改进效果.为了能够更加清楚并且准确地比较各个算法的收敛速度以及精度,将各个算法分别独立运行50次,得出结果的平均值取以10为底的对数作为纵坐标来进行比较.其中,Schwefel函数为拥有最大值418.9829D的多维函数,其余8个测试函数皆为拥有最小值0的多维函数;Sphere, Quartic, Rosenbrock, Quadric为单峰函数, Griewank, Ackley, Rastrigin, Schwefel, Schaffer为多峰函数.各个测试函数的维度D设定为30,算法参数如表1所示.实验结果如图2所示.

表 2 测试函数

Table 2 Test functions

函数名	表达式	最优值	寻优范围
Sphere	$f(x) = \sum_{i=1}^D x_i^2$	0	$(-100, 100)^D$
Quartic	$f(x) = \sum_{i=1}^D ix_i^4$	0	$(-1.28, 1.28)^D$
Ackley	$f(x) = -20 \exp\{-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} - \exp[\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)]\} D + 20 + e$	0	$(-32.768, 32.768)^D$
Rosenbrock	$f(x) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$	0	$(-2.048, 2.048)^D$
Rastrigin	$f(x) = \sum_{i=1}^D \{x_i^2 - 10 \cos(2\pi x_i) + 10\}$	0	$(-5.12, 5.12)^D$
Schwefel	$f(x) = \sum_{i=1}^D \{x_i \cdot \sin \sqrt{ x_i }\}$	418.9829D	$(-500, 500)^D$
Griewank	$f(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	0	$(-600, 600)^D$
Quadric	$f(x) = \sum_{i=1}^D (\sum_{j=1}^i x_j)^2$	0	$(-100, 100)^D$
Schaffer	$f(x) = \sum_{i=1}^{D-1} \{ \frac{\sin^2 \sqrt{x_{i+1}^2 + x_i^2} - 0.5}{(0.001(x_{i+1}^2 + x_i^2) + 1)^2} + 0.5 \}$	0	$(-100, 100)^D$

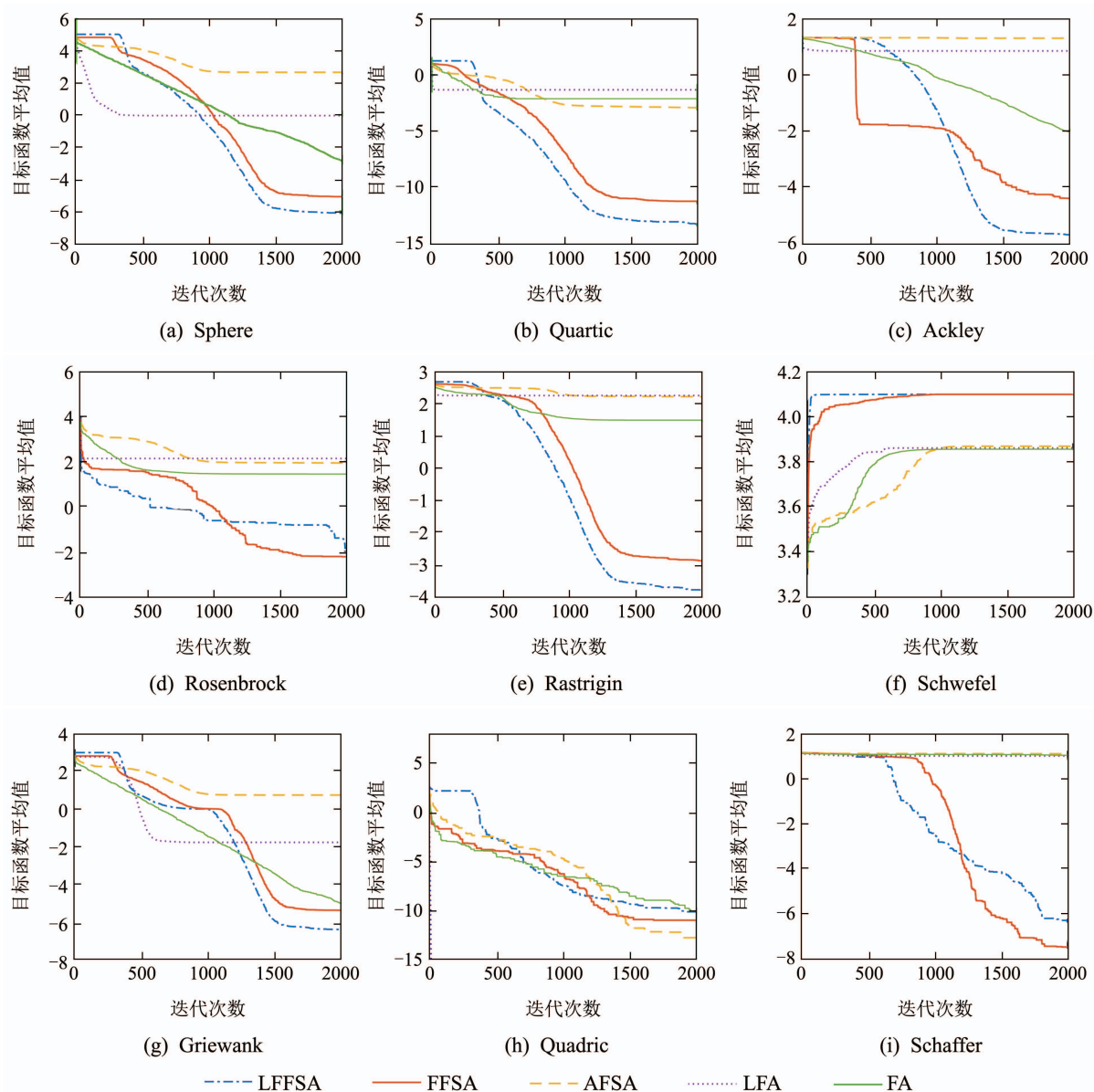


图2 测试函数平均迭代曲线

Fig. 2 Iterative curves of test functions

由图2可以看出,在种群规模,迭代次数以及各个参数都设置相同的情况下,在9个测试函数的寻优过程中,除了Quadric函数在不同算法的寻优过程中都出现了可观的解(LFA甚至在迭代前几步内就得到了最优解),LFFSA和FFSA相比较其他算法在逃脱局部最优以及寻优精度上都展现出了绝对的优越性。在这两种算法的对比中,FFSA对于Schaffer和Rosenbrock的寻优结果优于LFFSA,在其余6种不同测试函数下,LFFSA的寻优显得更为精确,这也体现了Lévy flight对寻优过程的影响。而萤火虫算法(FA)和基本鱼群算法(AFSA)处理多维多极值函数效果较差,基本鱼群算法(AFSA)在对多个测试函数的优化过程中甚至还出现了一直停滞不前,根本无法逃脱局部最优值的现象。

为了更加直观地观察各个算法的寻优能力以及能

够方便与图2进行对照,表3中列出了各个算法在独立运行50次之后所得到的迭代结果的平均值(Average)、最优值(Best)、最差值(Worst)以及50次寻优结果的标准差(std)。

由表3可以看出,对于两种基本算法而言,虽然不能迭代到最优值附近,但FA对多数测试函数的优化结果比AFSA要好,这也证明了萤火虫搜索策略的优势。LFA的寻优表现非常不稳定,不能够适应多种不同类型函数的寻优。LFFSA和FFSA在50次独立寻优的过程中效果显著,都能够适应多种不同类型函数的寻优并得到可接受的解。从结果离散化的角度来看,经过Lévy Flight优化的算法产生的优化结果标准差更大,说明解的波动更大,在优化结果理想的情况下,更有可能在之后的迭代过程中产生更加精确的解。

表3 算法寻优结果比较

Table 3 Optimization results comparison

测试函数	项目	AFSA	std	FA	std	LFA	std	FFSA	std	LFFSA	std
Sphere	Worst	3.0132		-2.8714		1.2394		-5.0348		-6.7579	
	Best	2.3664	0.1800	-3.2583	0.1265	-0.3685	0.1633	-5.2378	0.0706	-8.4927	0.0764
	Average	2.6668		-3.0491		-0.2966		-5.1322		-7.2977	
Quartic	Worst	-2.3216		-1.9329		-2.7284		-11.1708		-14.1973	
	Best	-4.1270	0.1703	-2.81797	0.1432	-3.2539	0.1845	-11.3934	0.0633	-15.5835	0.1319
	Average	-2.8706		-2.2351		-2.9669		-11.2771		-14.7321	
Ackley	Worst	1.3121		-1.7221		1.3116		-3.9602		-6.1479	
	Best	1.3043	0.0017	-2.4114	0.1220	1.3072	0.1412	-4.6671	0.1901	-6.9343	0.1376
	Average	1.3093		-2.0912		1.3096		-4.3223		-6.5776	
Rosenbrock	Worst	2.1070		2.1353		1.7794		-2.0341		-1.0585	
	Best	1.5691	0.0859	1.0028	0.0966	2.0423	0.1447	-2.4049	0.1297	-3.3997	0.3535
	Average	1.9488		1.4338		1.8978		-2.1565		-1.9089	
Rastrigin	Worst	2.2952		1.8674		2.0840		-2.8394		-4.3733	
	Best	2.0838	0.0764	1.2025	0.8414	2.3208	0.1623	-2.9871	0.0470	-5.1245	0.0570
	Average	2.2387		1.4929		2.2115		-2.9393		-4.6463	
Schwefel	Worst	3.8172		3.7886		3.8433		4.0993		4.0993	
	Best	3.9249	0.0167	3.9097	0.0332	3.9048	0.2412	4.0993	$7.47e^{-6}$	4.0993	$4.31e^{-6}$
	Average	3.8699		3.8532		3.8592		4.0993		4.0993	
Quadric	Worst	-11.6658		-9.6986		-372.32		-9.3449		-8.6653	
	Best	-14.0222	1.0569	-10.5425	1.3246	-385.01	1.4133	-12.1996	0.8684	-11.9376	1.3237
	Average	-12.7166		-10.0232		-378.50		-10.9443		-10.0785	
Griewank	Worst	0.9557		0.9557		-1.6162		-5.3033		-6.2707	
	Best	0.6439	0.1324	0.4997	0.2231	-1.9982	0.4066	-5.4446	0.0502	-6.4494	0.0546
	Average	0.7349		0.6972		-1.8347		-5.3571		-6.3718	
Schaffer	Worst	1.1066		1.1121		1.0936		-6.9543		-5.7639	
	Best	1.0769	0.0083	1.0575	0.0146	0.9972	0.0233	-7.7621	0.2934	-7.2139	0.4900
	Average	1.0944		1.0887		1.0053		-7.4185		-6.3114	

表4 LFFSA的时间复杂度估计

Table 4 Complexity estimation of the LFFSA

LFFSA寻优步骤	时间复杂度
1. 初始化 N 条人工鱼	$O(N)$
2. 初始化公告板	$O(N)$
3. 聚群行为	$O(N^2 + 2 * N)$
4. 觅食行为	$O(\text{Trynumber} * N)$
5. 终止条件判断	$O(1)$
6. 公告板信息输出	$O(1)$

5.3 LFFSA时间复杂度分析(Complexity analysis of LFFSA)

5.3.1 理论时间复杂度分析(Computational complexity analysis of LFFSA)

算法的运行效率通常使用时间复杂度来进行估量,一般将算法执行基本操作的次数称为时间复杂度.表4详细列出了LFFSA各个寻优步骤的时间复杂度的值.

以 N 表示种群规模,可以求得AFSA的时间复杂度为

$$O(\text{Maxgen} * (3 * N^2 + \text{Trynumber} * N + 6 * N)),$$

同样地,可求得FA的时间复杂度为

$$O(\text{Maxgen} * (N^2 + N)),$$

LFFSA的时间复杂度为

$$O(\text{Maxgen} * (N^2 + \text{Trynumber} * N + 4 * N)).$$

通过上述分析可以看出,时间复杂度主要与种群规模有关.3种算法的时间复杂度均为 $O(N^2)$ 级别,但FA和LFFSA的时间复杂度要优于AFSA.

5.3.2 试验时间复杂度分析(Experimental complexity analysis of LFFSA)

理论分析只能粗略估算时间复杂度,算法真正运行效率只有试验测试才能知道.由于各个算法能够达到的收敛精度差别太大,对算法进行固定收敛精度的试验比较不可行,因此依旧对各算法进行固

定迭代次数的试验, 对算法运行时间进行统计并分析. 试验参数如第5.2节所述, 各算法对不同测试函数寻优的平均运行时间如表5所示.

表5 算法平均寻优时间

Table 5 Average running time of algorithms

测试函数	寻优时间/s				
	AFSA	FA	LFA	FFSA	LFFSA
Sphere	10.23	7.47	13.1	18.5	7.53
Quartic	12.37	7.73	40.3	21.1	9.60
Ackley	10.87	7.80	15.2	20.7	8.96
Rosenbrock	11.40	7.67	5.0	20.4	10.27
Rastrigin	10.53	7.66	5.5	19.4	8.03
Schwefel	10.53	7.43	20.5	18.7	8.56
Griewank	12.33	8.03	22.3	22.7	9.23
Quadric	30.8	10.2	15.2	38.5	23.0
Schaffer	16.2	8.4	15.5	20.6	13.5

由表5可以看出, 在迭代次数相同的情况下, FA在运行时间上更加具有优势, LFFSA在优化后较AFSA运行速度有了明显的提高. 依然使用聚群行为和追尾行为两种行为模式进行并行搜索的FFSA在运行时间上几乎是LFFSA的两倍, 因此, 减少算法的并行搜索量, 尽可能简化算法的结构是改善算法性能的重要手段. LFA在运行时间上有着不稳定的表现, 并由第5.2节知, 其解的质量也不稳定. 可以看得出Lévy flight并不适合用于所有函数的寻优. 由于不同算法对某些测试函数的寻优无法达到相同

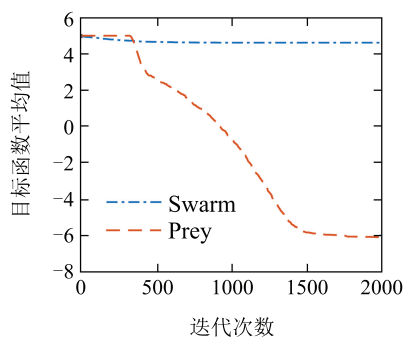
程度的精度, 综合寻优时间和寻优精度进行比较, LFFSA优势明显, 其在9个测试函数的优化过程中得到的解的质量和算法运行速度都是优秀的.

6 算法特性分析 (Characteristic analysis of LFFSA)

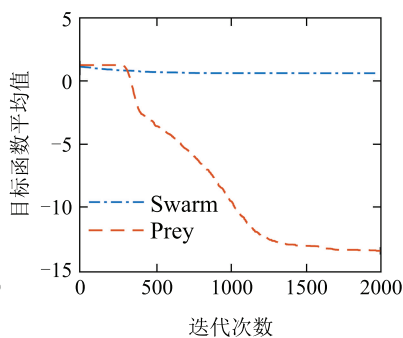
6.1 鱼群行为模式比较分析 (Behavior pattern comparison of LFFSA)

对于LFFSA寻优过程而言, 不同行为模式执行的频率必不相同. 为了比较鱼群不同行为模式在寻优过程中所占的运行比例, 首先将LFFSA中进行聚群行为的人工鱼去掉吸引度的定义, 使其进行盲目移动, 使得只有进行觅食行为的人工鱼按照萤火虫的移动模式进行移动(Prey); 其次, 将进行聚群行为的人工鱼赋予吸引度定义, 而进行觅食行为的人工鱼去掉吸引度的定义, 使得只有进行聚群行为的人工鱼按照萤火虫的移动模式进行移动(Swarm). 算法的参数设置如第5.1节所述, 以比较鱼群觅食行为和聚群行为运行权重的差异. 图3是9个测试函数在鱼群行为模式不同的情况下所得到的实验结果.

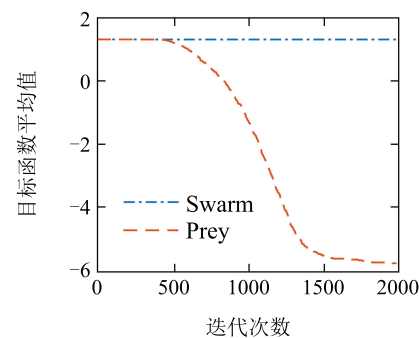
结合图2和图3可以看出, 对于大部分测试函数的优化而言, 只优化了觅食行为的算法(Prey)迭代曲线和LFFSA的优化曲线几乎重合, 说明聚群行为几乎是没有什么做出贡献的; 而对于个别的测试函数如Rosenbrock和Schaffer函数, 只优化了觅食行为的算法(Prey)即使在迭代速度和优化精度上无法达到LFFSA的程度, 但其在各个方面都优于只优化了聚群行为的算法(Swarm).



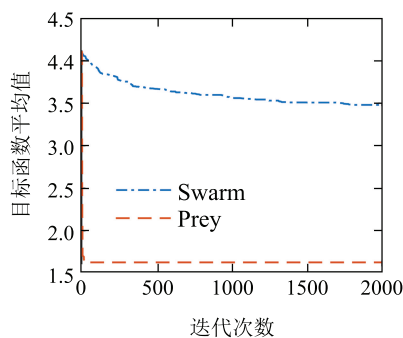
(a) Sphere



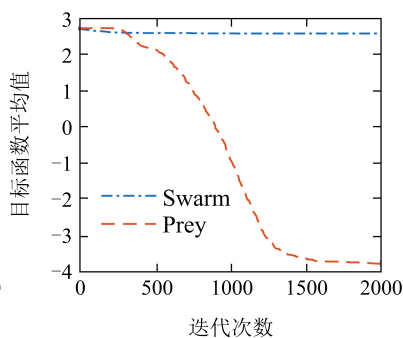
(b) Quartic



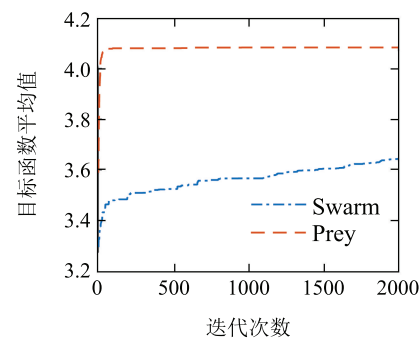
(c) Ackley



(d) Rosenbrock



(e) Rastrigin



(f) Schwefel

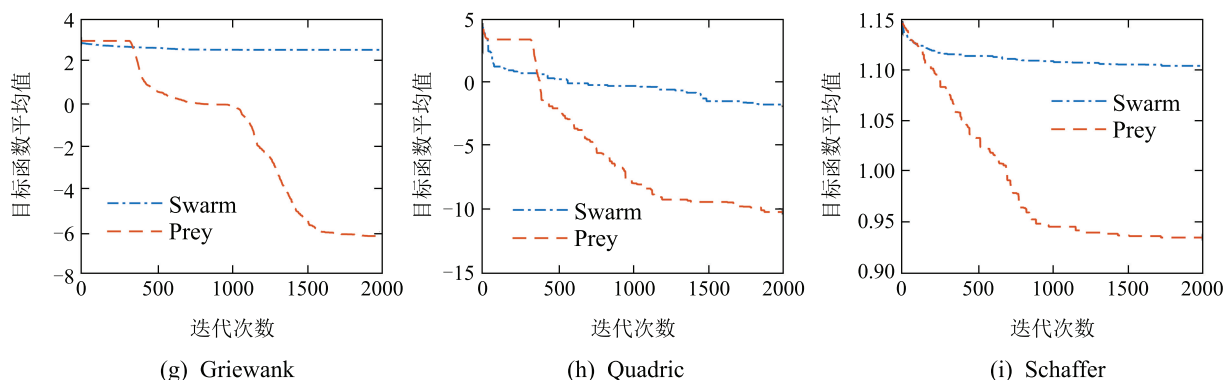


图3 不同行为模式下测试函数平均迭代曲线

Fig. 3 Iterative curves of test functions under different behavior patterns

图3说明了在大多数复杂函数的优化过程中进行觅食行为的概率要高于进行聚群行为的概率. 因此, 对于觅食行为现有的缺陷的改进便显得更为重要. 为了更进一步体现出两种行为模式所占权重的不同, 使用了MATLAB2014b的探查器来对程序的不同部分进行计次比较. 考虑到觅食行为是聚群行为的缺省行为, 因此觅食行为被执行的次数占聚群行为的百分比便可以体现出觅食行为的运行比例. 由种群数40, 迭代次数2000, 重复运行50次可知聚群行为调用的次数为 4×10^6 . 表6是9个测试函数在LFFSA的优化过程中觅食行为所占调用次数的统计情况.

由表6可以看出, 9个测试函数优化过程中觅食行为的调用率均在95%以上; 在其中5个测试函数中的调用率, 接近100%; 测试函数中觅食行为的平均调用率超过96%. 上述数据表明在算法运行过程中, 搜索大概率下更易满足觅食行为的要求, 只能转而执行觅食行为, 也说明了觅食行为的改进对于鱼群算法优化的重要性.

表6 觅食行为调用次数统计情况

Table 6 Calling percent of preying behavior in LFFSA

测试函数	调用值	调用率	平均值
Sphere	3999860	99.99	
Quartic	3999784	99.99	
Ackley	3999849	99.99	
Rosenbrock	3918695	97.97	
Rastrigin	3999896	99.99	96.49
Schwefel	3810340	95.26	
Griewank	3999655	99.99	
Quadric	3764880	94.12	
Schaffer	3244480	81.11	

6.2 参数改进分析 (Parameter improvement of LFFSA)

通过对LFFSA的4个参数Trynumber, β_0 , γ , δ 取不同的值, 分析其对寻优效果的影响. 以Ackley函数为例, 图4是参数变化的情况下目标函数值的变化情况. 由图4可知, 对于Trynumber而言, 更大的参数值会得到更好的迭代结果, 同样也会增加算法的时间复杂度. 初始吸引度 β_0 越小对寻优越有利. γ 和 δ 对寻优的影响不类似于线性关系, 两者的最优取值分别为2.5和1.2.

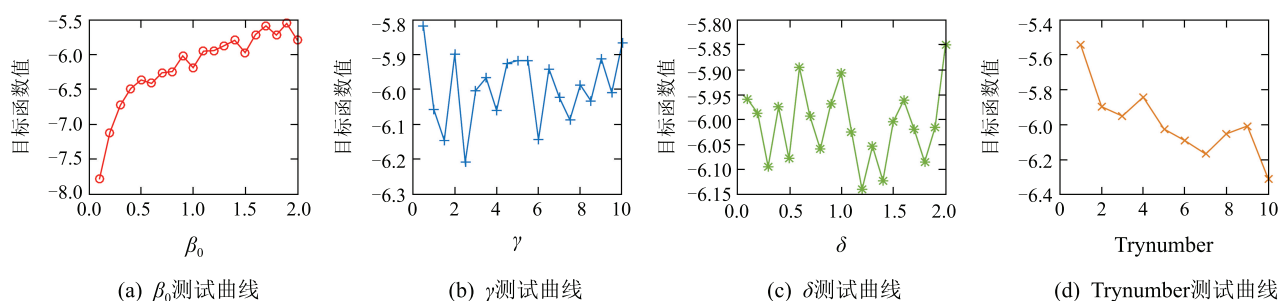


图4 参数测试曲线

Fig. 4 Test curves of parameters

6.3 LFFSA寻优优势和价值分析 (Optimization superiority analysis of LFFSA)

对于基本鱼群算法(AFSA)而言, 两种行为模式

(聚群行为和追尾行为)并列运行, 并将觅食行为嵌套于以上两种行为模式中, 这种寻优方式对于进行全局寻优, 跳出局部最优解和提高种群多样性具有

十分有效的作用, 但是其缺点在于每只人工鱼的移动距离是随机决定的, 无法在选择移动对象之后进行准确而且高效地移动, 且算法复杂度较高, 运行速度慢; 而对于萤火虫算法(FA)而言, 萤火虫所具有的亮度和萤火虫之间的吸引度使得每只萤火虫能够在继承前一只萤火虫的位置的前提下, 根据相互之间吸引度的大小进行更加有效的移动. 将人工鱼赋予吸引度定义的目的就在于将萤火虫所具有的更加优越的移动模式赋予毫无头绪, 盲目移动的人工鱼, 从而提高鱼群寻优的精度, 避免盲目地落入局部最优解当中; 在此基础上Lévy flight策略的加入使得处于觅食行为的人工鱼不再进行盲目地搜索; 减少并行搜索的行为模式的数量以减小算法复杂度, 提高运行速度. 第5.2节所述实验证明了上述改进的有效性.

除了比其他算法更有优势之外, 本方法与很多已有的增加鱼群寻优模式的优化方法截然不同. LFFSA采取了减少鱼群寻优模式的方法来对算法结构进行简化进而减少算法复杂度, 将两种寻优模式并行寻优改为单寻优模式寻优, 通过对人工鱼移动方式的改进来弥补算法的不足, 减少不必要的并行运算, 符合智能算法追求至简化的原则. 在改进AFSA算法性能的同时, LFFSA也结合了萤火虫算法的优势使得能够完成更多不同特点的测试函数的寻优.

7 结论(Conclusions)

本文提出了LFFSA, 并通过9个不同的测试函数对AFSA, FA, LFA, FFSA, LFFSA 5种算法的寻优性能进行了测试与比较. 仿真结果表明, 对于拥有不同特点的测试函数的寻优, 本文所提出的LFFSA在全局搜索能力, 寻优精度和运行速度等方面较其他算法都有更加明显的优势. 其次, 对LFFSA的优越性进行了分析, 并且通过改变不同行为模式下人工鱼的移动策略, 比较了鱼群各个行为模式之间的执行频率的差别, 进一步体现了LFFSA寻优的有效性.

参考文献(References):

- [1] WOLPERT D H, MACREADY W G. No free lunch theorems for optimization [J]. *IEEE Transactions on Evolutionary Computation*, 1997, 1(1): 67 – 82.
- [2] GROSAN C, ABRAHAM A. *Hybrid Evolutionary Algorithms: Methodologies, Architectures, and Reviews* [M] // Hybrid Evolutionary Algorithms. Berlin: Springer, 2007: 1 – 17.
- [3] LI Xiaolei. A new intelligent optimization method—artificial fish school algorithm [D]. Hangzhou: Zhejiang University, 2003. (李晓磊. 一种新型的智能优化方法——人工鱼群算法 [D]. 杭州: 浙江大学, 2003.)
- [4] YANG X S. *Nature-inspired Metaheuristic Algorithms* [M]. London: LuniverPress, 2008: 83 – 96.
- [5] DUAN Qichang, TANG Ruoli, XU Hongying, et al. Simulation analysis of the fish swarm algorithm optimized by PSO [J]. *Control and Decision*, 2013, 28(9): 1436 – 1440. (段其昌, 唐若笠, 徐宏英, 等. 粒子群优化算法仿真分析 [J]. 控制与决策, 2013, 28(9): 1436 – 1440.)
- [6] ZHANG Meifeng, SHAO Cheng, GAN Yong, et al. Hybrid artificial swarm optimization algorithm based on mutation operator and simulated annealing [J]. *Acta Electronica Sinica*, 2006, 34(8): 1381 – 1385. (张梅凤, 邵诚, 甘勇, 等. 基于变异算子与模拟退火混合的人工鱼群优化算法 [J]. 电子学报, 2006, 34(8): 1381 – 1385.)
- [7] WANG C R, ZHOU C L, MA J W. An improved artificial fish-swarm algorithm and its application in feed-forward neural networks [C] // *International Conference on Machine Learning and Cybernetics*. Guangzhou: IEEE, 2005: 2890 – 2894.
- [8] TSAI H C, LIN Y H. Modification of the fish swarm algorithm with particle swarm optimization formulation and communication behavior [J]. *Applied Soft Computing*, 2011, 11(8): 5367 – 5374.
- [9] JAGATHEESAN K, ANAND B, SAMANTA S, et al. Design of a proportional-integral-derivative controller for an automatic generation control of multi-area power thermal systems using firefly algorithm [J]. *IEEE/CAA Journal of Automatica Sinica*, 2017, PP(99): 1 – 14.
- [10] SU H J, CAI Y, DU Q. Firefly-algorithm-inspired framework with band selection and extreme learning machine for hyperspectral image classification [J]. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2017, 10(1): 309 – 320.
- [11] SELVARASU R, MUNAGALA S K, CHRISTOBER ARC. Self-adaptive firefly algorithm based multi-objectives for multi-type FACTS placement [J]. *IET Generation, Transmission & Distribution*, 2016, 10(11): 2576 – 2584.
- [12] DAWIT T, LEE C H, LIN Y W, et al. A modified firefly algorithm for photovoltaic maximum power point tracking control under partial shading [J]. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 2016, PP(99): 1 – 1.
- [13] MICHAEL A, PIERGIORGIO A, CHRISTIAN M, et al. Firefly algorithm for finding optimal shapes of electromagnetic devices [J]. *IEEE Transactions on Magnetics*, 2016, 52(3): 1 – 4.
- [14] AKANKSHA M, VENKATA N K G. Line utilisation factor-based optimal allocation of IPFC and sizing using firefly algorithm for congestion management [J]. *IET Generation, Transmission & Distribution*, 2016, 10(1): 115 – 122.
- [15] SU H J, YONG B, DU Q. Hyperspectral band selection using improved firefly algorithm [J]. *IEEE Geoscience and Remote Sensing Letters*, 2016, 13(1): 68 – 72.
- [16] YANG X S, DEB S. Cuckoo search via Lévy flights [J]. *World Congress on Nature & Biologically Inspired Computing*, 2010, 71(1): 210 – 214.

作者简介:

殷红 (1978–), 女, 副教授, 主要研究方向为智能优化和模式分析, E-mail: yinhong@mail.lzjtu.cn;

董康立 (1994–), 男, 硕士研究生, 主要研究方向为智能优化和模式分析, E-mail: 1083797729@qq.com;

彭珍瑞 (1972–), 男, 教授, 博士生导师, 主要研究方向为智能优化、测控技术, E-mail: pzrui@163.com;

李少远 (1965–), 男, 教授, 从事预测控制、自适应智能控制、模糊智能控制的研究, E-mail: syli@sjtu.edu.cn.