Computer Science and Application 计算机科学与应用, 2019, 9(7), 1377-1386
Published Online July 2019 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2019.97155

Hans汉斯

Study on Improvement of Dragonfly Algorithm

Xiaoping Hu, Feiwu Zhou*

Engineering Research Center of Advanced Mining Equipment, Ministry of Education, Hunan University of Science and Technology, Xiangtan Hunan
Email: hxp210@163.com, *1402912258@qq.com

Received: Jul. 4th, 2019; accepted: Jul. 18th, 2019; published: Jul. 25th, 2019

Abstract

An improved dragonfly algorithm (IDA) was proposed to overcome the disadvantages of the standard dragonfly algorithm, such as slow convergence rate and easy to be trapped in local solutions. In order to improve the ability of balancing exploration and exploitation, IDA algorithm proposes two kinds of nonlinear function that can dynamically adjust the convergence factors of the alignment weight and cohesion weight. Grey Wolf mechanism has good performance in exploitation and rate of convergence. In order to improve the convergence accuracy and speed of the dragonfly algorithm, the grey Wolf mechanism was incorporated into the dragonfly algorithm. In the late iteration of the algorithm, the diversity of the population decreases, which makes the algorithm easy to fall into the local solution. The lowliest place elimination series is introduced to improve the diversity of the population and make the algorithm jump out of the local solution. The improved algorithm is simulated with six complex functions and compared with the other three algorithms. The results show that the convergence accuracy, convergence speed and stability of IDA algorithm are better than the other three algorithms.

Keywords

Dragonfly Algorithm, Nonlinear Function, Grey Wolf Mechanism, Lowliest Place Elimination Series

基于蜻蜓算法的改进研究

胡小平，周非无*

湖南科技大学先进矿山装备教育部工程研究中心，湖南 湘潭
Email: hxp210@163.com, *1402912258@qq.com

收稿日期：2019年7月4日；录用日期：2019年7月18日；发布日期：2019年7月25日

*通讯作者。

文章引用：胡小平，周非无．基于蜻蜓算法的改进研究[J]．计算机科学与应用, 2019, 9(7): 1377-1386.
DOI: 10.12677/csa.2019.97155

摘　要

针对标准蜻蜓算法中存在的收敛速度慢，易于局部解的缺点，提出一种改进的蜻蜓算法(IDA)。该算法提出两种非线性函数，分别动态调节列队权重和聚集权重的收敛因子，提高算法平衡全局搜索和局部开发的能力；灰狼机制有较强的局部开发能力和收敛速度，融合灰狼机制以提高蜻蜓算法的收敛精度和速度；算法迭代后期种群多样性下降，引入末位淘汰策略来提高种群的多样性，使算法跳出局部解。通过6个复杂的测试函数对改进算法进行仿真，并和其他三个算法进行对比。结果表明，IDA算法的收敛精度、收敛速度和稳定性都优于其他三个算法。

关键词

蜻蜓算法，非线性函数，灰狼机制，末位淘汰

1. 引言

蜻蜓算法(Dragonfly Algorithm)是由 Seyedali Mirjalili 在 2015 年提出的一种新兴群智能算法[1]。在自然界中，蜻蜓有种独特而罕见的群集行为，蜻蜓聚集只有两个目的：捕猎和迁徙。作者以此为出发点提出两个新的概念：静态和动态。静态(狩猎)，群指蜻蜓群体分成多个小群体分布在不同区域进行来回飞行，类似于全局搜索；动态(迁徙)，是指蜻蜓形成亚群，并向一个方向长距离迁徙，有利于局部开发。DA 算法自提出，因良好的性能引来众多学者的广泛关注。Lang Xu 等[2]提出用差分进化改进蜻蜓算法并且应用于多级彩色图像分割；Mohamed Abdel-Basset 等[3]把二进制版的蜻蜓算法(BDA)来解决 0~1 背包问题，Suresh V 等[4]利用 DA 算法求解太阳能静态经济调度问题的方法。Babayigit 等[5]提出用 DA 算法实现低旁瓣的 CCAAs 设计方法。Jafari，Mohammad 等[6]利用 DA 算法对具有准三角剖分的正交各向异性无限板进行优化。Mohammed Amroune 等[7]用蜻蜓算法对电力系统电压稳定进行评估。Gururaghav Raman 等[8]将蜻蜓算法(DA)应用于光伏系统全局最大功率点(GMPP)跟踪。

虽然该算法有一定的优势，但存在易陷入局部解和收敛速度慢的缺点，针对这些缺点，Sree R. K. S. 等[9]提出融合粒子群优化算法中的记忆功能，加强蜻蜓个体间的信息交流，以提高算法的收敛精度；Hossam M. Zawbaa 等人[10]提出基于极值学习机(ELM)的混合蜻蜓算法；Sayed Gehad Ismail [11]等人提出将混沌映射嵌入到搜索迭代蜻蜓算法中；吴伟民等[12]提出加强个体信息交流的算法，增强个体之间的信息交流。

为改善蜻蜓算法容易陷入局部解，收敛速度慢的缺点，本文提两种非线性函数分别动态调节列队和聚集权重的收敛因子，以提高算法的稳定性。将灰狼机制和末位淘汰策略融入到算法当中，改善易于局部解和收敛速度慢的缺点。提出一种有着较强的全局搜索能力和局部开发能力，且能较好的调节全局搜索和局部开发的改进蜻蜓算法(IDA)。

2. 蜻蜓基本算法

Reynoldz [13]在文章中指出三个关于蜂群体行为准则：分离度、对齐度与聚合度。分离度是指相邻

个体间保持适当距离，以免碰撞；对齐度是指速度和方向与相邻个体对齐；聚合度是指个体飞向相邻区域中心。蜻蜓主要目标都是生存，作者 Seyedali Mirjalili 提出五个因素影响蜻蜓算法的位置更新：分离，列队，聚集，捕食，逃离。数学模型如下：

1) 分离，由公式得：

$$S_i = -\sum_{j=1}^{N} X - X_j \tag{1}$$

X 为蜻蜓当前所在位置，Xj 表示与 X 蜻蜓相邻的第 j 个蜻蜓的位置，N 表示与 X 蜻蜓相邻的个体总数。

2) 列队，由公式得：

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N} \tag{2}$$

Vj 指第 j 个相邻个体的速度

3) 聚集，由公式得：

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X \tag{3}$$

X 是当前个体位置。

4) 捕食，指蜻蜓向猎物靠拢，由公式得：

$$F_i = X^+ - X \tag{4}$$

X^+指食物的位置。

5) 逃离，指蜻蜓逃离天敌，由公式得：

$$E_i = X^- + X \tag{5}$$

X^-指天敌的位置，食物源位置是算法当前最优位置，天敌位置是当前最差的位置。通过以上五种纠正方式的组合形成了蜻蜓的行为，作者通过模仿 PSO 算法提出步长向量、位置向量来描述蜻蜓的位置。

6) 步长向量更新由公式得：

$$\Delta X_{t+1} = \left(sS_i + aA_i + cC_i + fF_i + eE_i\right) + \omega \Delta X_t \tag{6}$$

S_i，A_i，C_i，F_i，E_i 指上文的 5 中纠正方式，ω 为惯性权重，而 s，a，c，f，e 分别指的是分离权重，列队权重，聚集权重，捕食权重，天敌权重，t 表示当前迭代次数。

7) 位置向量更新由公式得：

当 $N > 0$ 时 $X_{t+1} = X_t + \Delta X_{t+1}$ (7)

当 N = 0 时，利用随机游走(Levy flight)在搜索空间中飞行。

$$X_{t+1} = X_t + Le'vy\left(d\right) \times X_t \tag{8}$$

3. 灰狼算法

灰狼优化算法(GWO) [14]是一种群智能算法，通过模拟灰狼群的社会等级制度和捕食方式达到优化的目的。灰狼群体社会的统治阶层比较严格，可以分为四个等级：α 、β 、δ 、ω 。灰狼小组的领导者

被称为α，为最优解；灰狼等级体系中的第二级是β，为次有解；ω可以满足整个群体，为普通狼，保持群体的主导架构，为第四层。第三层是δ，第三优解。以α、β、δ为引导，对目标物进行搜寻。设灰狼种群个数为M，则第i只灰狼在d维度的位置记为$X_i=\left(X_{i1},X_{i2},X_{i3},\cdots,X_{id}\right)$。

在狩猎时，灰狼的位置更新公式如下：

$$X_i\left(t+1\right)=X_p\left(t\right)-A\cdot\left|C\cdot X_p\left(t\right)-X_i\left(t\right)\right| \tag{9}$$

$$A=2a\cdot r_1-a \tag{10}$$

$$C=2r_2 \tag{11}$$

$$a=2-t/t_{\max} \tag{12}$$

式中，t为当前迭代次数，A和C为协同系数向量，X_p为猎物的位置，X为灰狼个体的位置。a为A系数的收敛因子，r_1，r_2表示[0, 1]间的随机数，t_{max}为最大迭代次数

由于猎物的具体位置未知，可根据灰狼算法中的等级制度，以α、β、δ进行引导使狼群接近猎物，灰狼群体可依据α、β、δ进行位置更新，公式如下：

$$X_1=X_\alpha-A\cdot\left|C\cdot X_\alpha-X\right| \tag{13}$$

$$X_2=X_\beta-A\cdot\left|C\cdot X_\beta-X\right| \tag{14}$$

$$X_3=X_\delta-A\cdot\left|C\cdot X_\delta-X\right| \tag{15}$$

$$X\left(t+1\right)=\frac{X_1+X_2+X_3}{3} \tag{16}$$

根据(16)对灰狼个体进行更新。

4. 蜻蜓算法的改进

4.1. 非线性调节收敛因子

在前文提到作者把蜻蜓群体运动分为两种状态：静态和动态。静态类似于全局搜索，动态类似于局部探索。为了保证 DA 算法的收敛性，蜻蜓算法通过调节惯性权重和其他五种权重，使得算法从全局搜索过渡到局部开发。并且，在静态群体中蜻蜓以捕食小型飞行猎物，有较高的聚集权重和较低的列队权重。而在动态群体中，有较低的聚集权重和较高的列队权重[1]。标准算法中聚集、列队权重更新方式的数学表达式如下：

$$a=c=2\cdot r_3\cdot my\_q \tag{17}$$

$$my\_q=0.1-\left(f_{\max}-f_{\min}\right)\cdot\frac{t}{T_{\max}/2} \tag{18}$$

my_q为列队权重和聚集权重的收敛因子，呈线性变化，当$my_q<0$时，$my\_q=0$，r_3为为[0, 1]之间的随机数。为了提高算法平衡全局搜索和局部开发的能力，当处于静态群体时赋予高列队权重和较低的聚集权重；在动态群体时应赋予低列队权重和较高的聚集权重。因此提出两种非线性函数动态控制列队和聚集权重的收敛因子，数学表达式如下：

$$my\_qa=\left(f_{\max}-f_{\min}\right)\cdot b\cdot\left(\cos\left(\frac{t\cdot\pi}{T_{\max}}\right)-q\cdot\left(\frac{t}{T_{\max}}\right)^2+1\right) \tag{19}$$

$$my\_qc = \frac{2}{\pi}\cdot\left(f_{\max} - f_{\min}\right)\cdot\left(\arccos\left(\frac{2\cdot t}{T_{\max}} - 1\right) - \frac{\pi}{2}\right) \tag{20}$$

$T_{\max}$ 为最大迭代次数，常数 b，q 为非线性调节系数，t 为当前迭代，$f_{\max}$，$f_{\min}$ 分别为 a，c 取值的上限和下限。当 $my\_qa < 0$，$my\_qc < 0$ 时，则 $my\_qa = 0$，$my\_qc = 0$。

用 my_qa 调节列队权重的收敛因子和用 my_qc 调节聚集权重的收敛因子。通过两种不同的非线性函数分别调节列队权重和聚集权重的收敛因子，使得算法平衡全局搜索和局部开发的能力得到提高。

4.2. 融合灰狼机制

灰狼算法中的灰狼机制具有结构简单、需要调节的参数少、容易实现的特点，使灰狼机制在局部开发和收敛速度有着良好的性能[15] [16] [17]。为提高 DA 算法的收敛精度和收敛速度，将灰狼机制和蜻蜓算法相结合，即公式(16)。对位置向量更新公式(7)提出改进，公式如下：

$$X_{t+1} = \frac{X_1 + X_2 + X_3}{3} + \Delta X_{t+1} \tag{21}$$

在改进的位置向量更新公式中，包含了个体当前的位置和群体历史最优位置，也包括了蜻蜓算法的步长向量，使得改进后的算法不但融合灰狼机制较强的局部开发能力又能保留了标准 DA 算法较强的全局搜索能力，并且提高算法的收敛速度。

4.3. 末位淘汰策略

在自然界中适者生存不适者淘汰，鲍义东[18]等提出，为了增强狼群的竞争力，引入新的狼，淘汰生存能力较弱的一些狼，以此提高狼群的生存能力。由此受到启发，提出末位淘汰策略来提高算法的多样性。

随着迭代的进行，蜻蜓之间的差异性较小，容易使算法陷入局部解。为了提高算法的多样性，避免陷入局部解，本文引入末位淘汰策略。末位淘汰策略是以蜻蜓个体的适应度值为衡量标准，以适应度值按升序排列，淘汰排在末位的 k 个蜻蜓个体，同时随机产生 k 个新的蜻蜓个体，以保持蜻蜓群体数量的稳定。淘汰蜻蜓的个数影响种群的多样性，k 值较大时，种群的多样性较大，有利于全局搜索，但不利于算法收敛。k 值较小，有利于局部开发。通过对 k 值大小的控制来保证算法的收敛性。k 的值变化如下公式。

$$k = k_{\max} - \left(k_{\max} - k_{\min}\right) * \left(\frac{t}{T_{\max}}\right) \tag{22}$$

k 值为整数，$k_{\max}$ 为淘汰的最大值，$k_{\min}$ 为淘汰的最小值。

4.4. 算法实现

IMDA 具体步骤如下：

步骤 1：初始化参数，包括种群大小 N、维度 d、最大迭代次数 Max，五种行为权重和惯性权重，邻域半径；

步骤 2：随机初始化位置向量 X 和步长向量 ΔX；

步骤 3：$t > 1$，更新 s, a, c, f, e 权重和惯性权重，其中列队权重的收敛因子和聚集权重收敛因子分别由 4.1 章提出的公式(19) (20)进行更新；

步骤 4：计算个体适应度值，更新食物和天敌的位置；

步骤 5：利用公式(1)~(5)更新 S, A, C, F, E；

步骤 6：若存在至少一个相邻个体，则通过(6)公式更新步长向量，用 4.2 章提出的融合灰狼机制策略更新位置向量，即公式(21)。若不存在相邻个体，则以公式(8)更新位置向量；

步骤 7：计算种群个体的适应度值，适应度值以升序排列；

步骤 8：用 4.3 章提出的末位淘汰策略对种群进行淘汰和随机生成。淘汰排在后 k 位的蜻蜓个体，同时随机生成 k 个蜻蜓个体；

步骤 9：$t = t + 1$，如果 $t < \text{Max}$，返回步骤 3。

5. 实验结果及分析

为了进一步验证 IDA 的优化效果，选取两组具有不同特征的基准测试函数，从不同角度对 IDA 算法的性能进行基准测试，测试函数分为两组:单模态函数(f1~f4)、多模态函数(f5~f6)。将 IDA 与标准蜻蜓算法进行对比测试。基准测试函数的具体情况如下表 1。

Table 1. Benchmark function test
表 1. 基准函数测试

函数	维数	取值范围	理论值
$\text{TF1}(x)=\sum_{i=1}^{n}x_i^2$	10	[−100,100]	0
$\text{TF2}(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	10	[−10,10]	0
$\text{TF3}(x)=\sum_{i=1}^{n}\left(\sum_{j-1}^{i}x_j\right)^2$	10	[−100,100]	0
$\text{TF4}(x)=\max_i\{\lvert x_i\rvert,1\le i\le n\}$	10	[−100,100]	0
$\text{TF5}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$	10	[−32,32]	0
$\text{TF6}(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2[1+\sin^2(3\pi x_1+1)]+(x_n-1)^2[1+\sin^2(2\pi x_n)]+\sum_{i=1}^{n}u(x_i,10,100,4)\}$	10	[−50,50]	0

为保证实验的公平性，统一实验环境，实验环境为 Windows 7 操作系统。所用 MATLAB 为 2016a 版本。为了更加合理比较算法的性能，此处选用标准 DA 算法外还选取了一个改进的蜻蜓算法，差分进化的蜻蜓算法(DEDA) [19]，遗传算法(GA)，GDA 是指 4.2 章提出的融合灰狼机制的蜻蜓算法。分别对 6 个基准测试函数进行求解。并从最优解、平均解、标准差三个方面对各算法进行评价。共同参数设置：种群规模 $N = 40$，Max = 500。IDA 中的 $f_{\max} = 0.1$，$f_{\min} = 0$，$k_{\max} = 0.7 * N$，$k_{\min} = 0.25 * N$ 差分进化的蜻蜓算法 $F = 0.5$，CR = 0.1。经典遗传算法 $p_c = 0.8$，$p_m = 0.05$。由于算法具有随机性，本实验对每个测试函数均运行 20 次。各算法对 6 个基准测试函数的计算结果统计如表 2 所示。

Table 2. Test data of each algorithm
表 2. 各算法的测试数据

函数	算法	最优解	平均解	标准差
	GA	0.149	1.092	0.918
	DA	1.790e−08	0.012	0.032
TF1	DEDA	6.212e−24	8.236e−10	1.901e−09
	GDA	4.023e−44	9.213e−39	3.805e−38
	IDA	3.429e−122	2.923e−110	1.526e−109
	GA	0.987	3.343	4.728
	DA	0.022	1.818	1.497
TF2	DEDA	1.051e−08	4.232e−4	1.335e−3
	GDA	9.701e−23	2.758e−21	5.106e−21
	IDA	5.764e−64	7.902e−61	1.659e−60
	GA	1.180e+03	2.806e+03	1.280e+03
	DA	0.796	2.137e+02	4.291e+02
TF3	DEDA	1.080e−12	4.962e−07	1.401e−06
	GDA	7.067e−37	4.962e−32	1.057e−31
	IDA	1.240e−62	7.228e−54	2.729e−53
	GA	2.317	9.613	5.918
	DA	0.327	1.895	1.012
TF4	DEDA	1.037e−05	0.025	0.055
	GDA	1.380e−17	1.374e−15	2.342e−15
	IDA	2.463e−25	6.908e−19	3.072e−18
	GA	1.065	7.351	6.981
	DA	0.127	2.197	0.950
TF5	DEDA	1.546e−10	0.057	0.258
	GDA	4.441e−13	9.948e−10	3.548e−9
	IDA	4.440e−15	7.105e−15	1.509e−14
	GA	2.349	11.584	9.797
	DA	0.012	1.167	1.408
TF6	DEDA	1.091e−6	0.370	1.362
	GDA	1.680e−06	4.954e−03	2.213e−02
	IDA	2.351e−08	4.071e−07	1.879e−07

1) 最优解和平均解反应算法的收敛精度，由表 2 的测试数据可得，GDA 算法在求解 TF1~TF4 单峰基准测试函数时，寻优能力优于标准蜻蜓算法、GA 算法和 DEDA 算法,对于复杂的多峰的基准测试函数优化效果虽不及单峰基准测试函数，但寻优能力依旧优于标准蜻蜓算法、GA 算法和 DEDA 算法,说明引入灰狼机制有利于提高算法的收敛精度。IDA 算法是在 GDA 算法基础上引入末位淘汰策略，由表 2 的测试数据可知，无论是单峰基准测试函数还是多峰基准测试函数，IDA 的寻优能力优于 GDA，说明末位淘汰策略的引入有利于提高算法的多样性，使得算法跳出出局部解。这说明算法经过三方面的改进，使得算法收敛精度和解的整体质量有很大提高，而 DA、GA 和 DEDA 都存在过早收敛。

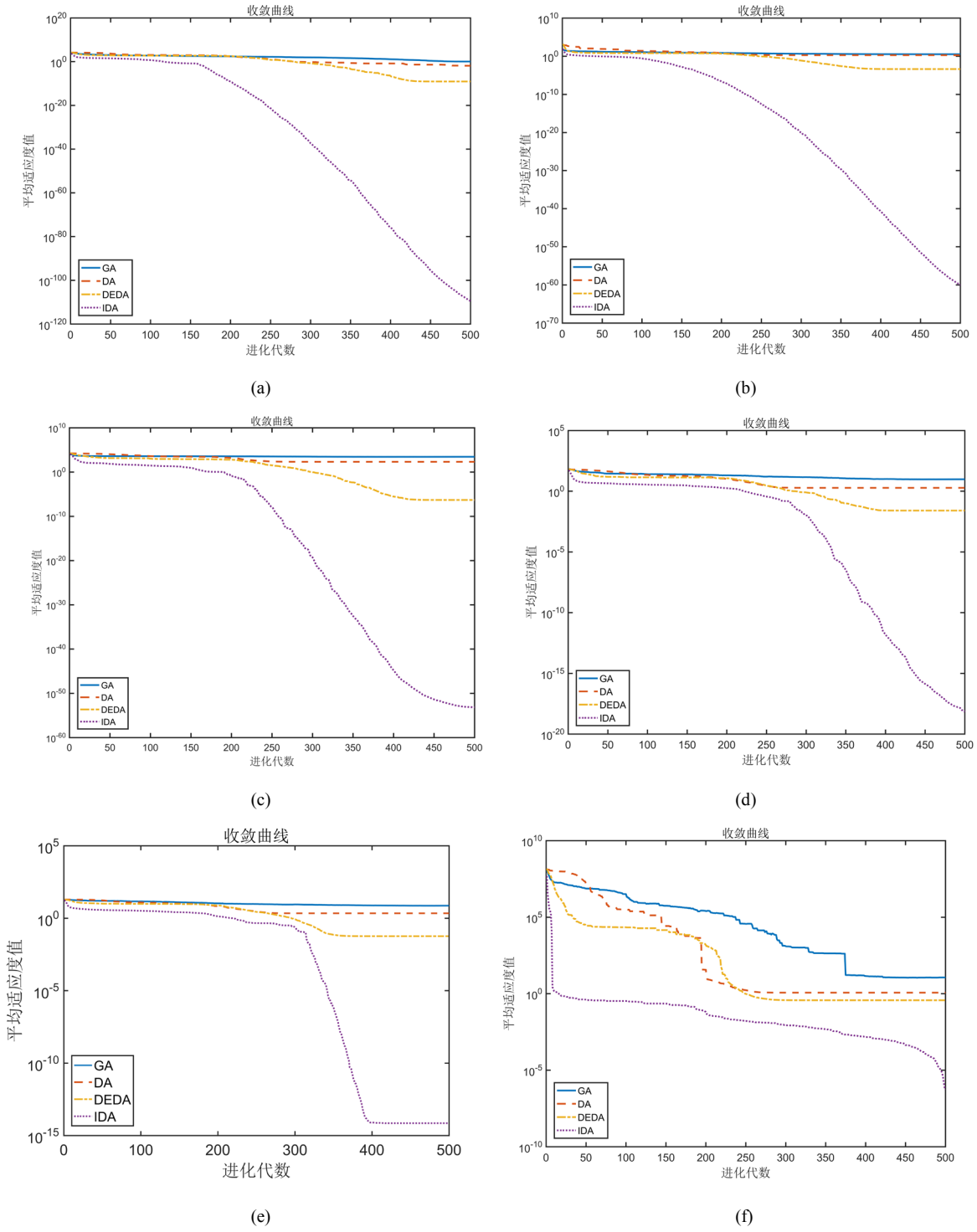

Figure 1. Average convergence curves under different test functions. (a) Average convergence curve of TF1 test function; (b) Average convergence curve of TF2 test function; (c) Average convergence curve of TF3 test function; (d) Average convergence curve of TF4 test function; (e) Average convergence curve of TF5 test function; (f) Average convergence curve of TF6 test function

图 1. 不同测试函数下的平均收敛曲线图。(a) TF1 测试函数的平均收敛曲线图；(b) TF2 测试函数的平均收敛曲线图；(c) TF3 测试函数的平均收敛曲线图；(d) TF4 测试函数的平均收敛曲线图；(e) TF5 测试函数的平均收敛曲线图；(f) TF6 测试函数的平均收敛曲线图

2) 标准差反应算法的稳定性，对于基准测试函数 TF1~TF4 来说，IDA 比标准函数优化精度分别高出 110，61，54，19 个数量级。对于多峰函数 TF5 和 TF6 的优化效果也分别提高了 15 和 7 个数量级，优于其他三个算法，说明改进后的算法在整体上稳定性强，抗“早熟”能力优于其他三个算法。

3) 从图 1(a)~(f)可以看出无论是多峰基准测试函数还是单峰基准测试函数，IDA 算法的收敛速度比其他三个算法都要快，这说明融合灰狼算法有利于提高算法的收敛速度。

总体来说，无论是对于多峰函数还是单峰函数，相同条件下 IDA 所得解的质量皆优于其他三个算法。很大程度上改善了标准算法中收敛精度不高、收敛速度慢和易“早熟”的缺点。

6. 结论

针对标准 DA 算法中的易于陷入局部解、收敛精度低和收敛速度较慢的缺点，本文提出三点改进：基于非线性函数调节列队和聚集权重的收敛因子，提高算法对全局搜索和局部开发的调节能力；融合灰狼机制改进的位置向量，提高算法的收敛精度的同时又能提高算法的收敛速度；末位淘汰策略，淘汰适应度值排在末位的 k 个蜻蜓个体，同时引入 k 个新的个体，以此提高了算法的多样性。通过 6 个基准测试函数仿真结果可知，IDA 算法在整体上，函数优化问题的结果优于 DA 标准算法和其他两个算法。下一步研究目标是应用到机器人路径规划上。

基金项目

国家自然科学基金 No. 61572185。

参考文献

[1] Mirjalili, S. (2016) Dragonfly Algorithm: A New Meta-Heuristic Optimization Technique for Solving Single-Objective, Discrete, and Multi-Objective Problems. *Neural Computing and Applications*, **27**, 1053-1073. https://doi.org/10.1007/s00521-015-1920-1

[2] Xu, L., Jia, H., Lang, C., *et al*. (2019) A Novel Method for Multilevel Color Image Segmentation Based on Dragonfly Algorithm and Differential Evolution. *IEEE Access*. https://doi.org/10.1109/ACCESS.2019.2896673

[3] Abdel-Basset, M., Luo, Q., Miao, F., *et al*. (2017) Solving 0-1 Knapsack Problems by Binary Dragonfly Algorithm. In: *International Conference on Intelligent Computing*, Springer, Cham, 491-502. https://doi.org/10.1007/978-3-319-63315-2_43

[4] Suresh, V. and Sreejith, S. (2017) Generation Dispatch of Combined Solar Thermal Systems Using Dragonfly Algorithm. *Computing*, **99**, 59-80. https://doi.org/10.1007/s00607-016-0514-9

[5] Babayigit, B. (2017) Synthesis of Concentric Circular Antenna Arrays Using Dragonfly Algorithm. *International Journal of Electronics*, **105**, 784-793. https://doi.org/10.1080/00207217.2017.1407964

[6] Jafari, M. and Bayati Chaleshtari, M.H. (2017) Using Dragonfly Algorithm for Optimization of Orthotropic Infinite Plates with a Quasi-Triangular Cut-Out. *European Journal of Mechanics A/Solids*, **66**, 1-14. https://doi.org/10.1016/j.euromechsol.2017.06.003

[7] Amroune, M., Bouktir, T. and Musirin, I. (2018) Power System Voltage Stability Assessment Using a Hybrid Approach Combining Dragonfly Optimization Algorithm and Support Vector Regression. *Arabian Journal for Science & Engineering*, **43**, 3023-3036. https://doi.org/10.1007/s13369-017-3046-5

[8] Raman, G., Manickam, C., *et al*. (2016) Dragonfly Algorithm Based Global Maximum Power Point Tracker for Photovoltaic Systems. Advances in Swarm Intelligence, Springer International Publishing, Berlin. https://doi.org/10.1007/978-3-319-41000-5_21

[9] Sree, R.K.S. and Murugan, S. (2017) Memory Based Hybrid Dragonfly for Numerical Optimization Problems. *Expert Systems with Application*, **83**, 63-78. https://doi.org/10.1016/j.eswa.2017.04.033

[10] Zawbaa, H.M., Emary, E., Salam, M.A., *et al*. (2016) A Hybrid Dragonfly Algorithm with Extreme Learning Machine for Prediction. *International Symposium on Innovations in Intelligent Systems and Applications*, Sinaia, 2-5 August 2016.

[11] Ismail, S.G., Alaa, T. and Ella, H.A. (2018) Chaotic Dragonfly Algorithm: An Improved Metaheuristic Algorithm for

Feature Selection. *Applied Intelligence*, **49**, 188-205.

[12] 吴伟民, 吴汪洋, 林志毅, 李泽熊, 方典禹. 基于增强个体信息交流的蜻蜓算法[J]. 计算机工程与应用, 2017, 53(4): 10-14.

[13] Reynolds, C.W. (1987) Flocks, Herds, and Schools: A Distributed Behavioral Model. ACM, New York. https://doi.org/10.1145/37401.37406

[14] Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014) Grey Wolf Optimizer. *Advances in Engineering Software*, **69**, 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[15] Malik, M.R.S., Mohideen, E.R. and Ali, L. (2016) Weighted Distance Grey Wolf Optimizer for Global Optimization Problems. *IEEE International Conference on Computational Intelligence & Computing Research*, Madurai, 10-12 December 2015. https://doi.org/10.1109/ICCIC.2015.7435714

[16] Long, W., Liang, X., Cai, S., *et al*. (2016) A Modified Augmented Lagrangian with Improved Grey Wolf Optimization to Constrained Optimization Problems. *Neural Computing and Applications*, **28**, 1-18. https://doi.org/10.1007/s00521-016-2357-x

[17] Zhang, S., Luo, Q. and Zhou, Y. (2017) Hybrid Grey Wolf Optimizer Using Elite Opposition-Based Learning Strategy and Simplex Method. *International Journal of Computational Intelligence and Applications*, **16**, Article ID: 1750012. https://doi.org/10.1142/S1469026817500122

[18] 鲍义东, 夏栋梁, 赵伟艇. 基于凸策略优胜劣汰蚁群算法的机器人路径规划[J]. 计算机系统应用, 2015, 24(8): 122-127.

[19] 赵齐辉, 杜兆宏, 刘升, 陈思静. 差分进化的蜻蜓算法[J]. 微电子学与计算机, 2018, 35(7): 101-105.