

doi:10.16055/j.issn.1672-058X.2021.0005.010

# 基于扇形趋利果蝇优化算法改进的 FS-K 聚类算法\*

曹珍贯, 杨 逊, 吕旻姝, 朱靖雯

(安徽理工大学 电气与信息工程学院, 安徽 淮南 232001)

**摘 要:**针对果蝇算法对高维函数收敛精度低的缺点,提出了一种改进的基于扇形搜索的果蝇算法(Fan search-Fruit Fly Optimization Algorithm, FS-FOA),该算法在原果蝇 FOA 算法的基础上改进了果蝇群体的搜索路径,并赋予果蝇个体趋利性,使更多的果蝇个体朝着味道浓度更大的方向前进,使果蝇群体的搜索方向有更多的选择性,增加果蝇算法在处理高维函数问题上的收敛速度和收敛精度;并将改进的 FS-FOA 算法与 K-means 聚类相结合,提出一种 FS-K 聚类算法,与原 K-means 聚类和原果蝇(FOA)算法进行对比实验,引入 5 个经典的测试函数对原 FOA 算法和 FS-FOA 算法寻优结果进行测试,结果表明采用 FS-FOA 算法具有更高的收敛精度;引入 5 个 UCI 公共数据集对改进 FS-K 聚类算法和原 K-means 算法、SOM 聚类算法、FCM 聚类算法进行测试,结果表明 FS-K 聚类算法具有更好的聚类效果。

**关键词:**果蝇算法;扇区搜索;FS-K 聚类算法

中图分类号:TP181

文献标志码:A

文章编号:1672-058X(2021)05-0061-07

## 0 引 言

果蝇算法(FOA)是由潘文超<sup>[1]</sup>在 2012 年提出的,是基于仿生果蝇群体觅食行为的寻优算法;文献[2]对果蝇算法和其他算法进行了对比,结果表明果蝇算法简单、参数少、运行效率高,并且在进化次数低时<sup>[2]</sup>,果蝇算法比混合跳蛙算法<sup>[3]</sup>、和声算法、人工蜂群算法的收敛精度和收敛速度都高,适合于实际应用或与其他算法结合进行分阶段优化。但是,FOA 和其他全局优化算法一样,标准果蝇优化算法也容易陷入局部最优,特别对于高维多极值复杂优化问题,在搜索计算后期收敛速度变慢,收敛精度降低。FOA 算法目前的优化途径大致分为两种:对算法自身的改进;将 FOA 算法与其他算法结合,

产生新算法。其中文献[4-5]将果蝇算法的固定步长改进为自适应步长;文献[6]将果蝇群分为搜索果蝇和跟随果蝇,分别进行全局和局部搜索;文献[7]将 FOA 算法与差分算法相结合,提高果蝇种群多样性;文献[8-9]也都是通过与其他算法相结合,弥补果蝇算法缺陷。智能群算法的改进应该遵循“内部完善-外部提升”<sup>[10]</sup>,但是目前少有学者对果蝇算法自身的搜索特性进行改进。

针对上述问题,提出了一种基于扇区搜索机制的果蝇优化算法,并将该方法应用于 K-means 算法优化。本文首先介绍了果蝇算法工作原理和扇区搜索机制搜索原理,然后将改进的果蝇算法用于 K-means 聚类算法的优化,最后通过对比实验验证改进的 FS-K 算法具有更好的聚类效果。

收稿日期:2020-09-27;修回日期:2020-11-20.

\* 基金项目:安徽省科技重大专项项目(201903a07020013)

作者简介:曹珍贯(1979—),男,浙江泰兴人,博士,副教授,从事智能控制研究.

## 1 果蝇算法

果蝇优化算法是一种仿生果蝇觅食行为的算法,模拟了果蝇群的两种觅食搜索方式:嗅觉搜索(果蝇群捕捉空气中的气味逼近气味源)和视觉搜索(果蝇个体在靠近气味源的同时,使用视觉发现食物与种群位置,并靠近)<sup>[1]</sup>。

果蝇算法模拟果蝇觅食行为可以总结为以下 7 步:

**Step 1** 初始化参数:群体规模  $S$ , 最大迭代数  $M\_G$ , 随机初始化群体坐标  $X_{\_axis}$ ,  $Y_{\_axis}$ 。

**Step 2** 在群体坐标周围, 随机生成  $S$  个果蝇个体,  $L$  为搜索半径:

$$X_i = X_{\_axis} + L * (2 * rand() - 1)$$

$$Y_i = Y_{\_axis} + L * (2 * rand() - 1)$$

**Step 3** 由于无法得知食物位置, 先计算果蝇个体与原点的距离  $D_i$ , 并将其倒数作为果蝇个体的味道浓度判定值  $S_i$ :

$$D_i = \sqrt{X_i^2 + Y_i^2}$$

$$S_i = 1/D_i$$

**Step 4** 将果蝇个体的味道浓度判定值  $S_i$  带入味道判定函数(即目标函数), 计算果蝇个体味道浓度值  $Sml_i$ :

$$Sml_i = Function(S_i)$$

**Step 5** 找出该果蝇群体中味道浓度最佳的果蝇(最小值或最大值):

$$[b\_Sml\_index] = \min(Sml_i)$$

**Step 6** 将果蝇的最佳味道浓度值  $b\_Sml$  保存下来, 并且记录最优个体的坐标值作为群体坐标:

$$X_{\_axis} = X(b\_index)$$

$$Y_{\_axis} = Y(b\_index)$$

**Step 7** 通过迭代运算, 进行寻优, 重复执行上面的 step 2—step 5, 并判断最佳味道浓度是否优于前一迭代最佳味道浓度, 并且判断当前迭代次数是否小于最大迭代次数  $M\_G$ , 若是, 则执行 Step 6。

## 2 扇形搜索果蝇算法

### 2.1 扇形搜索原理

FOA 算法在进行随机搜索时, 其果蝇群产生机

制是围绕群体坐标进行随机搜索:

$$X_i = X_{\_axis} + R, Y_i = Y_{\_axis} + R$$

其中  $R$  为  $[-L, L]$  的随机数。这种搜索机制在二维空间上表现为中点为种群坐标, 边长为  $2L$  的正方形, 果蝇的搜索范围如图 1 所示。这样的搜索机制使果蝇群在搜索时总是趋向于沿着单一方向移动, 限制了果蝇群搜索方向的多样性。

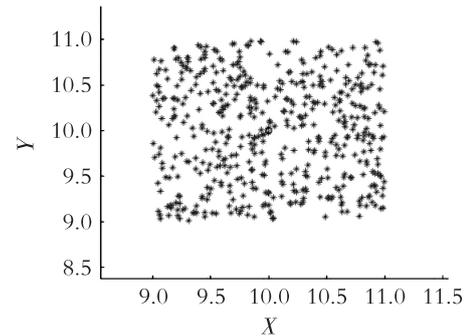


图 1 FOA 果蝇群搜索范围

Fig. 1 FOA drosophila population search area

针对果蝇群搜索范围的不平衡性, 本文提出一种扇形搜索机制, 该机制在果蝇随机搜索时, 以果蝇群坐标为中心, 把搜索区域划分为  $n$  个扇区, 每个扇区均生成果蝇个体。具体实现方法如下:

① 划分扇区, 其中  $t$  为扇区角度;

$$t = 0 : \frac{2\pi}{n} : 2$$

② 在群坐标周围产生果蝇群, 其中  $L$  为果蝇群搜索半径,  $rand()$  为  $[0, 1]$  的随机值;

$$X_i = X_{\_axis} + L * \sin(t) * rand()$$

$$Y_i = Y_{\_axis} + L * \cos(t) * rand()$$

对 FOA 和 FS-FOA 算法进行单独 5 次的运行, 对比了果蝇群的迭代路径, 如图 2。由图 2 可以看出: FS-FOA 果蝇群在搜索食物时, 有 4 次向上方飞行, 有 1 次向右方飞行, 而 FOA 果蝇群的迭代方向较为单一, 总是趋向于沿着  $Y=X$  的方向飞行。说明基于扇形搜索机制的果蝇群在寻找食物源时, 在种群前进方向上的选择性增加了, 相较于 FOA 算法, FS-FOA 算法的果蝇群总是朝着味道浓度最大的方向前进, 而 FOA 算法在迭代中受到搜索范围的影响, 方向单一, 无法朝着味道值最优的方向前进, 使得 FOA 算法的收敛较慢, 在高维度函数优化问题中, 还会陷入局部最优。

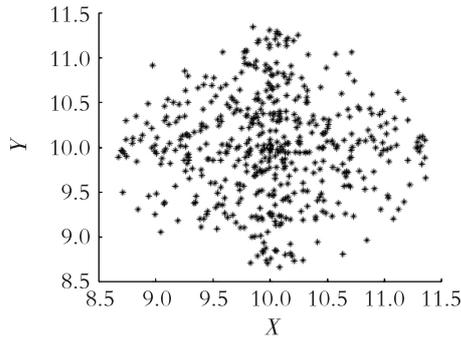


图2 FS-FOA 果蝇群搜索范围

Fig. 2 FS-FOA drosophila population search area

## 2.2 果蝇趋利特性

在自然界,果蝇在觅食过程中,其并不是随机在一个范围搜索,而是多数果蝇朝着食物源的方向飞行,少数果蝇向着其他方向搜索。受此启发,FS-FOA对果蝇个体赋予了不同的性质:趋利性和盲从性。盲从果蝇在果蝇群坐标周围随机搜索,趋利果蝇则更有方向性,总是朝着食物源一侧快速飞行,在食物源单一的情况下,趋利果蝇能够帮助算法减少无效迭代并快速收敛,在相同的迭代次数下,FS-FOA算法总能达到更高的收敛精度。

趋利果蝇的个数由趋利因子 $R$ 决定,趋利果蝇个数 $q=S * R$ ,其中 $R \in [0, 1]$ , $R$ 取值越大,果蝇的趋利性越强,但是 $R$ 的取值不应超过0.5,超过0.5会导致果蝇群忽略对味道源方向的判别。趋利果蝇的方向由趋利果蝇的个数决定,其生成方法如下:

首先,将果蝇的搜索方向人为划分为4个90度扇区,扇区与二维坐标轴重合;再通过此次群坐标与上代坐标对比,判断食物源相对种群的方位属于哪个扇区;趋利果蝇的间隔角度由趋利果蝇个数决定,根据式(1)计算:

$$\theta = 0^\circ : \frac{90^\circ}{q} : 90^\circ \quad (1)$$

趋利果蝇的搜索步长采用自适应步长,其步长值计算如式(2)所示:

$$L_q = L * \ln(gen+3) \quad (2)$$

其中, $L$ 为盲目果蝇的搜索步长, $gen$ 为迭代时所属方向的累计代数, $gen$ 的取值范围为 $gen \in [0, M\_G]$ 。趋利果蝇的自适应步长会使其拥有较高的加速度,能够较快确定食物源确切方向并向目标源飞行,同时在确定了食物源方位后也不会因为有大

大的步长而忽略食物源。趋利果蝇的生成策略如式(3)所示:

$$\begin{aligned} X_q &= X_{axis} + L_q * \sin(\theta) \\ Y_q &= Y_{axis} + L_q * \cos(\theta) \end{aligned} \quad (3)$$

## 3 FS-FOA 算法优化 K-means 聚类算法

聚类算法是数据分析经典算法,属于无监督学习算法,K聚类算法的目的是将数据分为指定类,尽可能使类内样本相似度大,类间相似度小。由于算法较简单且有极强的收敛性,K-means聚类算法在处理大数据的应用背景下有广泛的应用。但是K-means聚类对初始聚类中心的选取依赖性很大,极易陷入局部解。目前比较流行的优化方法是使用群优化算法对K-means算法进行优化,减少聚类算法对初始聚类中心的敏感度<sup>[11]</sup>。

针对K-means对初始聚类中心依赖性大的缺点,本文结合FS-FOA较强的全局搜索能力与K-means算法的快速收敛性,将FS-FOA与K-means相结合产生一种新型聚类算法FS-K。FS-K相较于FOA能够较快收敛,并通过迭代寻找全局最优值,避免陷入局部最优。

算法流程如下:

**Step 1** 对FS-K算法进行初始化:种群个数 $Sizepop$ ;最大迭代次数 $max\_gen$ ;聚类中心数 $K_{num}$ 。

**Step 2** 初始化果蝇群,从样本中随机抽取 $K_{num}$ 个数据作为一个果蝇个体,每个果蝇个体表示一种聚类中心。

**Step 3** 将果蝇个体作为K-means的初始聚类中心,收敛得到的聚类中心作为果蝇个体的初始位置。

**Step 4** 数据点距离所属聚类中心的欧式距离之和作为味道浓度判定函数,计算果蝇个体适应度,选择最优果蝇个体作为种群初始坐标。

**Step 5** 在种群坐标周围随机生成果蝇群,计算果蝇个体的味道浓度,记录最优个体的味道浓度以及位置信息。

**Step 6** 判断果蝇最优个体是否优于当前最优,如果是,将最优个体作为下一代的果蝇群坐标。

**Step 7** 进入迭代寻优并重复执行 **Step 5—Step 6**,直到达到最大迭代次数  $\text{Max gen.}$

## 4 试验与分析

### 4.1 优化函数测试

为了验证 FS-FOA 的聚类效果,采用了 5 种常用的测试函数进行仿真测试。为了更好地对比算法的优化性能,将测试分为固定迭代次数和固定精度两种。其中固定迭代次数试验中,FOA 与 FS-FOA 算法的最大迭代次数均取  $M\_G=100$ ,种群大小  $S=30$ ,

为保持 FOA 与 FS-FOA 的种群大小一样,把 FS-FOA 中盲从果蝇数量设置为 20,趋利果蝇数量设置为 10。固定精度试验是为了测试算法在目标精度下的迭代次数和运行时间,通过对比不同精度的函数优化来对比算法的收敛性。为了保证试验的可靠性,在初始化时,将 FOA 的初始化果蝇坐标  $X_{\text{axis}}$ ,  $Y_{\text{axis}}$  也作为 FS-FOA 的初始化坐标,并将 FOA 与 FS-FOA 算法各独立运行 30 次,取其优化结果的平均值、最优值和方差作对比。固定迭代次数的试验结果如表 2 所示,在目标精度下寻优对比结果如表 3 所示。

表 1 测试函数  
Table 1 Test functions

函数名	表达式	维 度	搜索范围	理论极值	峰 值
Sphere	$F_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]$	0	单峰
Griewank	$F_2(x) = \frac{1}{4000} \sum_{i=1}^n (x_i)^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0	多峰
Rastrigin	$F_3(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	30	$[-5.12, 5.12]$	0	多峰
Schaffer	$F_4(x) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	2	$[-100, 100]$	0	多峰
Sum Squares	$F_5(x) = \sum_{i=1}^n i x_i^2$	30	$[-100, 100]$	0	单峰

表 2 固定代数寻优结果对比

Table 2 Comparison of optimization results of fixed algebra

函 数	维 度	FOA			FS-FOA		
		均 值	最优值	均方差	均 值	最优值	均方差
Sphere	30	2.420e-5	2.294e-5	6.607e-6	1.053e-6	1.053e-6	2.339e-10
Griewank	30	1.632e-6	1.524e-6	4.519e-8	7.039e-8	7.030e-8	3.051e-11
Rastrigin	30	4.803e-3	4.436e-3	1.990e-4	2.089e-4	2.088e-4	3.446e-8
Schaffer	2	1.618e-6	1.539e-6	5.122e-7	7.026e-8	7.020e-8	5.563e-7
Sum Squares	30	3.757e-4	3.575e-4	9.264e-6	1.632e-5	1.631e-5	3.610e-9

由表 2 可以看出:FS-FOA 算法在处理单峰函数  $F_{1(x)}$ ,  $F_{5(x)}$  时,寻优精度会比 FOA 算法高出一个数量级,在处理多峰函数  $F_{2(x)}$ ,  $f_{4(x)}$  时,寻优精度比 FOA 高出两个数量级,说明 FS-FOA 在处理多峰函数时性能提升更明显;此外,FS-FOA 的方差在处理

$F_1, F_2, F_3, F_5$  时,要比 FOA 的方差高出 4 个数量级,在处理二维函数  $F_4$  时,方差与 FOA 算法有相同数量级,说明 FS-FOA 算法在处理高维函数时,其寻优能力更强,稳定性更高。

表 3 目标精度寻优时间对比

Table 3 Comparison of optimization time for target accuracy

函数	维 度	精 度	FOA		FS-FOA		时间占比 $\eta/\%$
			迭代次数	耗时 $t_1/s$	迭代次数	耗时 $t_2/s$	
$F_1$	30	$10^{-6}$	500	0.054	103	0.024	44.44
$F_2$	50	$10^{-8}$	1 374	0.304	235	0.076	24.67
$F_4$	2	$10^{-6}$	127	0.010 1	35	0.009 7	96.4
	30	$10^{-4}$	702	0.085	136	0.029	34.12
$F_3$	30	$10^{-6}$	7 057	0.802	931	0.146	18.2
	50	$10^{-6}$	9 075	1.591	1159	0.275	17.28
$F_5$	30	$10^{-6}$	1 954	0.21	317	0.046	21.9
	30	$10^{-8}$	19 682	1.988	2 264	0.358	18
	50	$10^{-8}$	32 640	4.898	3 520	0.746	15.23

对表 3 进行分析可知: FS-FOA 算法在目标精度下的寻优时间和代数更少,对于 30 维  $F_1, F_2$  函数;FS-FOA 所用时间占比为 44.44% 和 24.67%,其中时间占比  $\eta=t_2/t_1$ ,FS-FOA 在处理  $F_4$  函数时,时间占比为 96.4%,与 FOA 耗时差不多,这是由于处理低维函数时,所用迭代次数太少,在时间上差距不大。但是从  $F_3$  和  $F_5$  函数可以看出,FS-FOA 在处理目标精度更高的寻优时,其优化性能更明显,对  $F_3$  函数 30 维在  $10^{-4}$  和  $10^{-6}$  精度下寻优对比,发现时间占比缩小了 15.9%,对比  $10^{-6}$  精度下 30 维和 50 维,则缩小了 0.9%,说明 FS-FOA 是精度更高的寻

优算法,其寻优性更好。

#### 4.2 聚类试验及对比

实验采用 UCI 标准数据集库中的 Iris, Wine, Glass, 和 CMC 4 个数据集进行性能测试。仿真实验参数设置:初始种群个数  $S$  为 50,最大迭代数为 100,每组数据独立运行 30 次,取数据样本距离所属聚类中心的欧式距离和作为算法的适应度函数,对 K-means, FOA, SOM, FCM 以及 FS-K 的聚类结果进行对比。表 4 为试验得到的 30 次独立试验的最小值、最大值和均值。

表 4 聚类结果对比

Table 4 Comparison of clustering results

数据集	适应度值	K-means	FOA	SOM	FCM	FS-K
Iris	Min	97.22	96.69	103.33	94.58	96.65
	Max	124.18	96.75	129.27	118.67	96.65
	Mean	103.20	96.73	108.85	98.25	96.65
Wine	Min	16 537.47	16 306.06	16 480.44	16 448.37	16 292.23
	Max	18 436.95	16 364.32	17 849.68	17 694.68	16 294.22
	Mean	16 860.39	16 326.15	16 950.57	16 821.59	16 293.04
Glass	Min	213.22	211.62	212.36	208.74	211.79
	Max	250.83	224.03	254.48	237.84	212.96
	Mean	221.59	218.88	230.75	214.45	212.61
CMC	Min	5 542.18	5 534.71	5 541.14	5 526.75	5 532.98
	Max	5 545.33	5 538.26	5 567.65	5 540.62	5 534.24
	Mean	5 543.20	5 535.99	5 550.28	5 537.16	5 533.34

表 4 可以看出:FS-K 的聚类结果在稳定性上要  
比 K-means 和 FOA 高,并且总能跳出局部解,逼近  
最优值,和另外两个经典算法 SOM,FCM 进行对比,  
具有更好的聚类效果。从图 3 可以看出,在收敛性  
上,K-means 的收敛速度最快,能够在 10 代左右收  
敛;FS-K 收敛较慢,在迭代 50 次左右趋于稳定;  
FOA 收敛速度要快于 FS-K,但是容易陷入局部解,  
其跳出局部解的能力没有 FS-K 强。

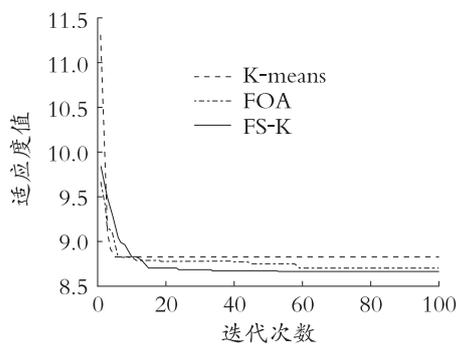


图 3 Iris 数据集聚类曲线对比

Fig. 3 Comparison of Iris data set cluster curves

## 5 结 论

提出了一种扇形搜索的果蝇优化方法,弥补了  
果蝇算法在搜索中受限于搜索范围的缺点,增加了  
果蝇算法跳出局部解的能力和收敛速度。在维度高  
的函数优化问题中,FS-FOA 算法的效果更加显著。  
随后,将 FS-FOA 算法与 K-means 算法相结合,提  
出一种优化聚类方法 FS-K,并使用 UCI 数据集进  
行测试,验证了算法的可行性。但是,采用改进后的  
FS-K 算法的聚类收敛的速度较慢,如何提高 FS-K  
聚类的收敛速度是未来进一步的研究方向。

### 参考文献(References):

- [1] PAN W C. A New Fruit Fly Optimization Algorithm Taking the Financial Distress Model as an Example[J]. Knowledge-Based System, 2012, 26: 69—74
- [2] 刘立群,韩俊英,代永强,等. 果蝇优化算法优化性能对比研究[J]. 计算机技术与发展, 2015, 25(8): 94—98
- [3] LIU L Q, HAN J Y, DAI Y Q, et al. Comparative Study on Optimization Performance of Fruit Fly Optimization Algorithm[J]. Computer Technology and Development, 2015, 25(8): 94—98(in Chinese)
- [4] 高建瓴,潘成成,吴建华,等. 改进混合蛙跳算法的研究[J]. 贵州大学学报(自然科学版), 2019, 36(5): 76—81
- [5] GAO J L, PAN C C, WU J H, et al. Research on Improved Hybrid Frog Jumping Algorithm [J]. Journal of Guizhou University (Natural Science Edition), 2019, 36(5): 76—81(in Chinese)
- [6] SHAN D, CAO G H, DONG H J. LGMS-FOA: An Improved Fruit Fly Optimization Algorithm for Solving Optimization Problems [J]. Mathematical Problems in Engineering, 2013, 2013(7): 1256—1271
- [7] PAN Q K, SANG H Y, DUAN J H, et al. An Improved Fruit Fly Optimization Algorithm for Continuous Function Optimization Problems [J]. Knowledge-Based Systems, 2014, 62(2): 62: 69—83
- [8] 王友卫,凤丽洲. 基于双子群和分区采样的果蝇优化新算法[J]. 浙江大学学报(工学版), 2017, 11(11): 2292—2298
- [9] WANG Y W, FENG L Z. A New Algorithm for Fruit Fly Optimization Based on Gemini Group and Partitioned Sampling [J]. Journal of Zhejiang University (Engineering Science Edition), 2017, 11(11): 2292—2298 (in Chinese)
- [10] NIU J, ZHONG W, LIANG Y, et al. Fruit Fly Optimization Algorithm Based on Differential Evolution and Its Application on Gasification Process Operation Optimization [J]. Knowledge - Based Systems, 2015, 88(11): 253—263
- [11] WU L, ZUO C, ZHANG H. A Cloud Model Based Fruit Fly Optimization Algorithm [J]. Knowledge - Based Systems, 2015, 89: 603—617
- [12] MITIĆ M, VUKOVIĆ N, PETROVIĆ M, et al. Chaotic

- Fruit Fly Optimization Algorithm[J]. Knowledge-Based Systems, 2015,89(8):446—458
- [10] 王林,吕盛祥,曾宇容. 果蝇优化算法研究综述[J]. 控制与决策,2017,32(7):1153—1162
- WANG L, LYU S X, ZENG Y R. Overview of Drosophila Optimization Algorithms [J]. Control and Decision, 2017, 32(7):1153—1162(in Chinese)
- [11] 杨俊闯,赵超. K-Means 聚类算法研究综述[J]. 计算机工程与应用,2019,55(23):7—14,63
- YANG J C, ZHAO C. Overview of K-means Clustering Algorithm Research[J]. Computer Engineering and Applications,2019,55(23):7—14,63(in Chinese)

## FS-K Clustering Algorithm Based on Fan-shaped Profit-seeking Fruit Fly Optimization Algorithm

CAO Zhen-guan, YANG Xun, LYU Min-shu, ZHU Jing-wen

(School of Electrical and Information Engineering, Anhui University of Science and Technology, Anhui Huainan 232001, China)

**Abstract:** According to the shortcoming of low convergence precision of the high dimensional function of Drosophila algorithm, an improved algorithm based on fan-shaped search fruit fly optimization algorithm (FS-FOA) is put forward. This algorithm improves the search path of drosophila group on the basis of the original fruit fly FOA algorithm, gives individual of fruit flies profit-seeking, makes more individuals of the fruit flies move in the direction of bigger taste concentration, makes fruit fly group search direction more selective and increases convergence speed and convergence precision for the fruit fly algorithm to deal with high dimensional function. A FS-K clustering algorithm is forwarded based on the combination of FS-FOA algorithm and k-means clustering. Through the comparison experiments of the original k-means clustering and the original FOA algorithm, five classical test functions are introduced to test the optimizing results of the original FOA algorithm and FS-FOA algorithm, and the results show that FS-FOA algorithm has higher convergence precision. Five UCI public data sets are introduced to test the improved FS-K clustering algorithm, the original k-means algorithm, SOM clustering algorithm and FCM clustering algorithm, and the results show that FS-K clustering algorithm has better clustering effect

**Key words:** fruit fly algorithm; sector search; FS-K clustering algorithm

责任编辑:李翠薇

引用本文/Cite this paper:

曹珍贵,杨逊,吕旻姝,等. 基于扇形趋利果蝇优化算法改进的 FS-K 聚类算法[J]. 重庆工商大学学报(自然科学版), 2021,38(5):61—67

CAO Z G, YANG X, LYU M S, et al. FS-K Clustering Algorithm Based on Fan-shaped Profit-seeking Fruit Fly Optimization Algorithm[J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2021,38(5):61—67