文章编号：1674-9057(2022)01-0223-06　　doi:10.3969/j.issn.1674-9057.2022.01.025

基于仿生优化算法的聚类改进算法

覃承友，谢晓兰，王悦悦，郭　杨

（桂林理工大学 a. 广西嵌入式技术与智能系统重点实验室；b. 信息科学与工程学院，广西 桂林　541006）

摘　要：为提升聚类算法的聚类效果，采用仿生优化算法与 k 均值聚类算法相结合(BFOA-K)实现数据聚类。在聚类过程中，为解决 k 均值对于初始质心敏感以及容易陷入局部最优的问题，使用果蝇优化算法确定 k 均值聚类算法的质心，再使用 k 均值聚类算法进行数据聚类。针对果蝇优化算法对于飞行步长的影响，采用 F 分布动态改变步长，提升算法全局搜索能力。同时采用精英保留策略，提升果蝇种群的多样性，扩大了搜索范围和提升了搜索效率。利用 4 个 UCI 标准数据集对算法进行仿真实验，结果表明，本文提出的 BFOA-K 算法在各项聚类评估指标中都优于其余对比算法，提升了算法的收敛性，证明了算法的有效性和可行性。

关键词：仿生优化算法；k 均值聚类；果蝇优化算法；F 分布；精英保留策略

中图分类号：TP301.6　　**文献标志码**：A

聚类，是数据挖掘中一个重要的组成部分，其目的为按照设定的特定标准把物理或抽象的对象集合分割成不同类或簇的小集合，这些小集合中的对象彼此相似，并与其他类或簇的对象有所差异，即聚类产生的同一类数据对象尽可能聚合在一起，不同的数据对象尽可能分离[1]。k 均值聚类算法(k-means clustering algorithm，简称 k-means)在聚类算法中有着举足轻重的地位，虽然其逻辑理论简单、可高效实现，应用十分广泛，在故障检测[2]、无人机应急[3]、隐私保护[4]等方面都有着广泛的应用，但是 k-means 算法本身也存在着一定缺陷，如对初始聚类中心非常敏感，且在迭代过程中容易陷入局部最优等问题。针对这些问题，对于 k-means 算法的改进方法有很多，其中仿生优化算法作为当今研究的热门领域，和 k-means 进行结合使用是一个可行的方法，其模拟了昆虫、兽群、鸟群和鱼群等群体的行为，群体中每个成员通过学习自身和其他成员的经验不断优化搜索，可以有效地找出全局最优解。如在文献[5]中提出了基于改进人工蜂群算法的 k 均值聚类算法，使用最大最小距离初始化蜂群，构造出一种基于全局引导的位置更新策略，提高了迭代寻优的效率；文献[6]提出了基于改进粒子群算法的 k 均值聚类算法，使用加入混沌搜索过程的粒子群算法增加粒子的多样性，为 k-means 有效地找到了较好的初始中心；文献[7]提出了基于云计算平台的仿生优化聚类数据挖掘算法，通过利用相似聚类和狼群优化算法，以头狼的位置确定聚类中心，在不断的狼群优化以及相似聚类中实现了在云计算平台对数据样本大且高维的数据聚类，并提升了收敛速度。上述算法在改进 k-means 中侧重点各有不同，对于 k-means 对初始聚类中心的敏感和易陷入局部最优的问题并没有得到很好的解决或达到很好的聚类效果。

本文在研究 k-means 聚类算法及仿生优化算法上，将属于仿生优化算法中的果蝇优化算法(fruit fly optimization algorithm，FOA)与 k-means 进行结合以达到更好的聚类效果。用改进的果蝇优化算法改良 k-means 的聚类中心，解决 k-means 易陷入局部最优及其初始聚类中心敏感的问题。考虑到不同算法模型在不同数据集上的效果有所差异，为了实验的严谨以及准确性，本方法选择不同 UCI 的标准数据集进行实验对比分析。

收稿日期：2021-06-04
基金项目：国家自然科学基金项目(61762031)；广西科技重大专项(桂科 AA19046004)；广西重点研发计划项目(桂科 AB18126006)
作者简介：覃承友（1996—），男，硕士研究生，研究方向：云计算、智能计算，1379390014@qq.com。
通讯作者：谢晓兰，博士，教授，237290696@qq.com。
引文格式：覃承友，谢晓兰，王悦悦，等. 基于仿生优化算法的聚类改进算法[J]. 桂林理工大学学报，2022，42(1)：223-228.

1 相关算法

1.1 *k*-means 聚类算法

对于给定数据集 $X=(X_1, X_2, \cdots, X_n)$ 其中包含 n 个成员对象，k-means 的任务就是将这 n 个对象根据某种距离或某种相似性划分到 k 个簇中 $(C_1, C_2, \cdots, C_k, 2 \leqslant k \leqslant n)$，簇的聚类中心称为质心，并使得这 k 个簇的聚类均值达到总体误差最小，即损失函数达到最小，k-means 的损失函数为

$$E=\sum_{i=1}^{k}\sum_{x\in C_i}\| x-\varphi_i \|^2, \tag{1}$$

其中：x 表示数据样本点；C_i 表示第 i 个簇；k 表示划分的簇数；φ_i 为簇 C_i 的质心，表示为

$$\varphi_i=\frac{1}{|C_i|}\sum_{x\in C_i}x。 \tag{2}$$

算法的具体步骤为：

步骤 1：给定需要聚类产生的簇数 k，在数据集中随机抽取 k 个点作为初始质心，给定迭代次数和聚类效果阈值。

步骤 2：按照式(3) 计算给定数据集中所有对象与各个质心的欧氏距离

$$dist(x, y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}, \tag{3}$$

其中：$dist(x, y)$ 表示每个对象的欧氏距离；x_i、y_i 表示对象的坐标；n 表示每个簇中的对象总数。

步骤 3：根据簇中所有的对象，根据式(2) 重新计算质心。

步骤 4：判断是否达到迭代次数或满足聚类效果阈值，若是，停止算法，保存质心与各个聚类的对象；若否，重复步骤 2、3 和 4。

1.2 果蝇优化算法

果蝇优化算法[8]的核心在于果蝇个体利用自身的嗅觉获取食物源的浓度值，并向周围的其他果蝇个体发送气味信息，再利用视觉飞向浓度值最高的果蝇位置，飞行的半径取决于食物源的浓度值大小。标准的果蝇优化算法流程如下[9-10]。

步骤 1：初始种群规模 *sizepop*，最大迭代次数 *maxgen*，维度 *dim* 和果蝇个体的飞行半径 L，并初始化果蝇种群的位置，其对应的初始位置信息定义为

$$\begin{cases}X_{\text{axis}}=10\ \text{rand}(1, dim);\\ Y_{\text{axis}}=10\ \text{rand}(1, dim)。\end{cases} \tag{4}$$

步骤 2：每一个果蝇个体根据赋予的飞行半径和方向，开始更新自身的位置，更新公式如下

$$\begin{cases}X_i=X_{\text{axis}}+\text{rand}(L);\\ Y_i=Y_{\text{axis}}+\text{rand}(L)。\end{cases} \tag{5}$$

步骤 3：由于初始食物源的方向和距离是未知的，因此每一个果蝇个体根据式(6) 计算距离原点的距离 *dist*，并根据式(7) 作为其对于食物源的浓度值判定 S_i，同时利用式(8) 计算每一个果蝇个体的味道浓度值 $Smell_i$

$$dist_i=\sqrt{X_i^2+Y_i^2}; \tag{6}$$

$$S_i=1/dist_i; \tag{7}$$

$$Smell_i=\text{fitness}(S_i)。 \tag{8}$$

式中：X_{axis}、Y_{axis} 表示初始种群坐标；rand( ) 表示随机分布数；X_i、Y_i 为更新坐标；i 表示第 i 个果蝇个体；fitness( ) 表示味道浓度值的判断函数，即适应度函数。

步骤 4：记录当前群体中最优味道浓度值的果蝇位置信息，并将其具体信息存入最优群体 $Smell_{best}$。

步骤 5：根据最优味道浓度值和记录的位置信息 *bestSmell*，果蝇群体的所有个体都往这个方向飞去。

步骤 6：判断是否达到迭代次数 *maxgen*，否，则重复步骤 2 ~ 6；是，则输出最优值，结束循环。

通过以上迭代步骤可知，FOA 的搜索策略是基于全局搜索的，并根据每次迭代的最优解指导下一次迭代的方向与位置，能够使种群以当前最优解继续开展随机搜索，从而朝着更优的方向搜索前进。

2 基于果蝇优化算法的 *k*-means 算法

2.1 基于 *F* 分布的果蝇飞行半径更新策略

F 分布式是一种非对称分布[11]，其有两个自由度，且位置不可置换，设 X 服从自由度为 n_1 的卡方分布，Y 服从自由度为 n_2 的卡方分布，且 X、Y 独立，则称随机变量 $F=(X/n_1)/(Y/n_2)$ 服从自由度为 (n_1, n_2) 的 F 分布记为 $F \sim F(n_1, n_2)$，其概率密度表达式为

$$f(x; n_1, n_2)=\begin{cases}\dfrac{1}{B(n_1/2, n_2/2)}n_1^{n_1/2}n_2^{n_2/2}x^{(n_1/2)-1}(n_2+n_1x)^{-(n_1+n_2)/2}, x>0;\\ 0, x\leqslant 0。\end{cases} \tag{9}$$

$$B(a, b)=\int_0^1 x^{a-1}(1-x)^{b-1}\mathrm{d}x。 \tag{10}$$

其中：n_1 为第一自由度；n_2 为第二自由度。对于自由度不同的 F 分布如图 1 所示。

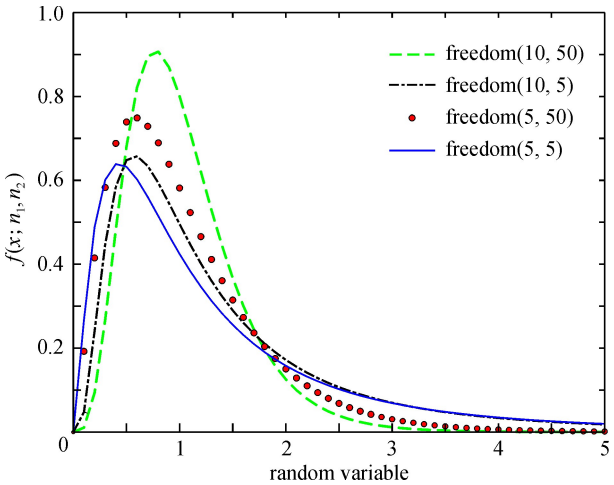


图 1　不同自由度的 F 分布曲线对比

Fig. 1　Comparison of F distribution curves with different degrees of freedom

使用基于 F 分布的果蝇飞行半径更新策略能有效优化果蝇优化算法中的飞行机制。在原始的果蝇优化算法中，每次的飞行半径是固定的，这就影响了算法在迭代中的寻优能力：如设置过大的飞行半径，虽然在算法迭代前期能很好地扩大搜索范围，并有效避免陷入局部最优，但是在算法迭代后期却容易在最优解附近产生震荡，难以有效接近最优解；若设置过小的飞行半径，则容易在算法迭代前期过早的陷入局部最优，难以搜索最优解。因此，选择一个合适的飞行半径对于果蝇优化算法十分重要，使用本文基于 F 分布的飞行半径，能有效规避以上两个问题。由图 1 可知，在 F 分布的曲线中，概率密度的变化曲线是非对称的，算法设置每迭代一次，F 分布的自变量 x 增加 0.05，其对应的概率密度即为算法变化步长。以基于 F 分布的变化飞行半径，算法能在开始迭代的时候能很好地扩大搜索范围，迭代到最后能避免震荡并找到全局最优解，有效提升果蝇优化算法的寻优效率。

2.2　基于精英保留机制的果蝇种群优化

精英保留机制，是一种高效的全局搜索机制，近年来在仿生优化算法的改进中运用十分广泛，其主要思想是针对一个问题求出所有可行解，通过可行解求出反向解，并对原解和反向解进行评估比较，从而选择较优的解作为精英解保留为下一代。

反向解的求解取决于反向点，设有 n 个二维空间点的集合 $S=(S_1, S_2, \cdots, S_n)$，$S_a$ 和 S_b 为集合的最大值和最小值，则对于点 S_i 的反向点定义为 S_i'，其表达式为

$$S_i' = S_a + S_b - S_i。\tag{11}$$

通过比较原始解和反向解，选择其中的最优解作为精英解。设点 S_i 为果蝇的原始解，则 S_i' 为反向解，设 $f(S)$ 为目标函数，当 $f(S_i) \geqslant f(S_i')$ 时，称为 S_i 精英解；当 $f(S_i) < f(S_i')$ 时，则称 S_i' 为精英解。

由于精英解是比较原始解和反向解得来的，因此能及时避免对劣质果蝇个体的继续搜索，让种群向着更优的区域搜索。使用精英解对果蝇种群的改进，能有效扩大种群的搜索范围，避免陷入局部最优。

2.3　改进的 k-means 算法

本文基于以上两点改进策略提出了基于改进果蝇优化算法的 k-means 聚类算法 BFOA-K。该算法的基本思想是：通过改进的果蝇优化算法进行一次迭代搜索最优值，将得到的最优值的位置信息作为 k-means 聚类的初始质心并开始一次聚类分析，再利用聚类得到新的质心，将新的质心的位置信息返回更新果蝇种群，反复交替迭代执行 FOA 和 k-means 直到满足最大迭代次数结束算法。改进的 BFOA-K 算法的流程如下：

步骤 1：初始果蝇种群规模 *sizepop*，维度 *dim* 最大迭代次数 *maxgen*，利用 F 分布定义果蝇飞行半径 L 和初始化果蝇种群的位置，并给定聚类的簇数 k。

步骤 2：利用式(11)重新定义果蝇个体的位置信息，比较它们的最优值来定义精英群体，同时每一个果蝇个体根据赋予的飞行半径和方向，开始更新自身的位置。

步骤 3：由于初始食物源的方向和距离是未知的，因此先计算每一个果蝇个体距离原点的距离 *dist*，并根据式(7)作为其对于食物源的浓度值判定 S_i，根据式(8)计算每一个果蝇个体的味道浓度值 $Smell_i$。

步骤 4：记录当前群体中最优味道浓度值的果蝇位置信息，并将其具体信息存入最优群体 $Smell_{best}$。

步骤 5：根据记录的最优个体的位置信息 *bestSmell*，将其作为聚类的初始质心，并执行如下操作：

a. 根据欧氏距离利用式(3)计算数据集中每个对象所隶属的质心；

b. 根据分配的簇中的对象，根据式(2)重新计算每个簇的质心；

记录每个质心的位置信息和聚类结果，并将质心位置重新作为果蝇种群的个体，根据最优味道浓度值和记录的位置信息，果蝇群体的每一个个体都往这个方向飞去。

步骤 6：判断是否达到迭代次数 *maxgen*。否，重复步骤 2~6；是，则输出最优值，结束循环。

本文的算法流程如图 2 所示。

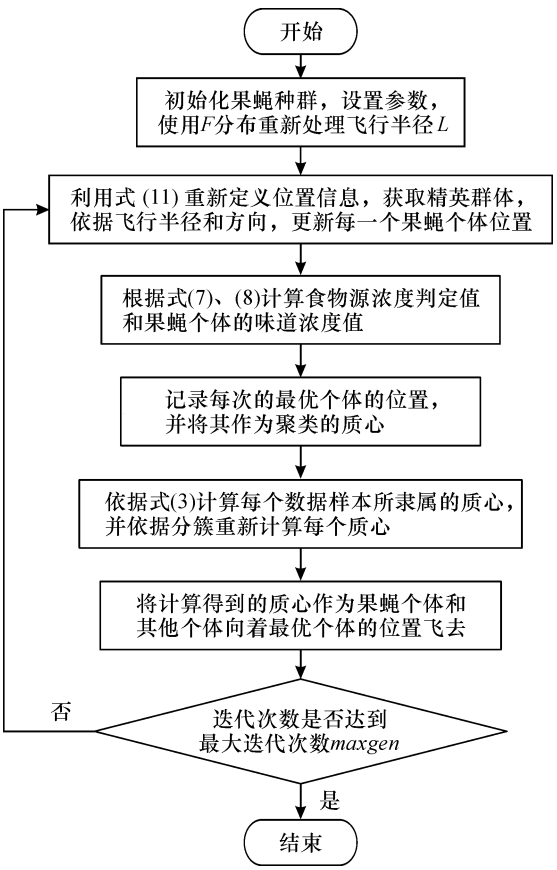


图 2 本文算法流程图

Fig. 2 Algorithm flow chart in this study

3 仿真结果及分析

3.1 实验分析

为了验证本文算法的有效性，使用 4 组算法实验作为对比，分别为传统 *k*-means 聚类算法[12]、IABC 算法[5]、基于自适应参数改进的密度聚类算法 *K*ANN-DBSCAN[13]、无改进果蝇优化 *k*-means 聚类算法 FOA-*K* 和本文提出的基于改进果蝇优化算法的 *k*-means 聚类算法 BFOA-*K*。为了验证算法的多样性，将采用多组数据集进行实验，数据集采用 UCI 上的 Iris、Wine、WDBC 和 Yeast 4 个标准数据集。本文实验环境为 Windows 10 操作系统，CPU 参数为 Xeon(R) E-2124G@ 3.40 GHz，16 GB 内存，Matlab R2018b，Python 3.7。各个数据集的详细信息见表 1。

表 1 各数据集详细信息

Table 1 Details of each dataset

数据集	所含类别	属性个数	样本个数
Iris	3	4	150
Wine	3	13	178
WDBC	2	30	569
Yeast	14	103	2 417

3.2 结果与分析

聚类的目的是将特征属性相近的划分成同一个簇，在聚类划分簇中有很多的评价指标，本文评价指标采用调整兰德指数(*ARI*)衡量数据分布的吻合程度

$$ARI = \frac{RI - \mathrm{E}(RI)}{\max(RI) - \mathrm{E}(RI)}, \tag{12}$$

其中：*RI* 表示兰德指数；$\mathrm{E}(RI)$ 表示兰德指数的期望；$\max(RI)$ 表示最大兰德指数。

采用聚类精确度(*ACC*)比较获得标签和数据提供的真实标签

$$ACC = N_{\mathrm{AC}}/N, \tag{13}$$

其中：N 表示样本总数；N_{AC} 表示聚类正确的数量。

采用标准化互信息(*NMI*)度量聚类结果的相近程度

$$NMI(\boldsymbol{U}, \boldsymbol{V}) = 2\frac{MI(\boldsymbol{U}, \boldsymbol{V})}{H(\boldsymbol{U}) + H(\boldsymbol{V})}。 \tag{14}$$

其中：$\boldsymbol{U}$ 表示真实标签向量；$\boldsymbol{V}$ 表示预测标签向量；$MI(\boldsymbol{U}, \boldsymbol{V})$表示互信息；$H(\boldsymbol{U})$和$H(\boldsymbol{V})$表示信息熵。

由于 4 个数据集的属性各不相同，因此在选择 *F* 分布的自由度时也不同，Iris、Wine、WDBC 和 Yeast 选择的 *F* 分布的自由度依次是(5, 5)、(10, 5)、(5, 50)和(10, 50)，算法对比将在每个数据集上重复运行 100 次，计算每个评价指标的平均值。上述评价指标设置为值越大表示效果越好。实验结果如表 2~表 5 所示。

表 2 5 种算法在 Iris 数据集上的比较

Table 2 Comparison of five algorithms on Iris dataset %

指标	*k*-means	IABC	*K*ANN-DBSCAN	FOA-*K*	BFOA-*K*
ARI	85.37	96.11	87.83	83.94	96.66
ACC	94.26	96.18	95.12	95.81	97.34
NMI	89.56	90.85	89.59	90.67	92.49

表 3　5 种算法在 Wine 数据集上的比较

Table 3　Comparison of five algorithms on Wine dataset　%

指标	*k*-means	IABC	*K*ANN-DBSCAN	FOA-*K*	BFOA-*K*
ARI	84.50	90.57	85.56	85.32	92.47
ACC	67.87	77.32	66.35	71.85	79.38
NMI	71.40	89.68	69.37	77.18	82.61

表 4　5 种算法在 WDBC 数据集上的比较

Table 4　Comparison of five algorithms on WDBC dataset　%

指标	*k*-means	IABC	*K*ANN-DBSCAN	FOA-*K*	BFOA-*K*
ARI	49.34	67.36	45.08	69.19	65.72
ACC	53.37	64.72	51.62	61.77	67.61
NMI	60.60	70.26	59.76	73.74	77.28

表 5　5 种算法在 Yeast 数据集上的比较

Table 5　Comparison of five algorithms on Yeast dataset　%

指标	*k*-means	IABC	*K*ANN-DBSCAN	FOA-*K*	BFOA-*K*
ARI	31.93	37.52	28.14	37.44	40.67
ACC	28.68	40.05	26.77	39.42	42.25
NMI	27.37	46.47	25.38	46.40	47.22

在 4 个数据集的测试中，加入改进果蝇优化算法的 *k*-means 聚类算法在各方面都能达到更好的聚类效果。BFOA-*K* 相较于传统 *k*-means 算法，*ARI* 平均提升了 20.81%，*ACC* 平均提升了 23.56%，*NMI* 平均提升了 29.75%，较于其他对比算法平均也有 12.67%的提升。在样本数量及属性数不大的情况下，5 种算法在各种评价指标下都有很好的聚类效果，但总体还是本文提出的算法要优于其他对比算法；而当样本数量及属性数增加时，5 种算法的性能都有所下降，但本文算法还是稍微优于其他对比算法，其主要原因是 BFOA-*K* 算法不仅在果蝇飞行半径上使用了可变的 *F* 分布，动态调整果蝇搜索半径，而且在果蝇的种群优化上使用了精英策略，使得果蝇扩大了搜索范围，增加了果蝇的多样性。但是在高维度和高样本量的数据集中，尤其是在 Yeast 数据集下每个算法表现都不好，这主要是因为涉及高维高样本的数据集中需要进行一定的降维或特征提取，所以会丢失一定的信息量，导致聚类效果不明显，改进数据集降维方法将是以后的研究重点。

为了验证本文算法的收敛速度及收敛程度，选取了样本属性都有一定差异的 Iris 和 Yeast 两个数据集进行比较，两个数据集随迭代次数的目标函数适应值变化曲线如图 3、图 4 所示。

可以看出，本文提出的 BFOA-*K* 算法在两个数据集的测试中，收敛速度和程度都优于传统 *k*-means 聚类算法；尽管 IABC 算法在 Iris 数据集的迭代过程

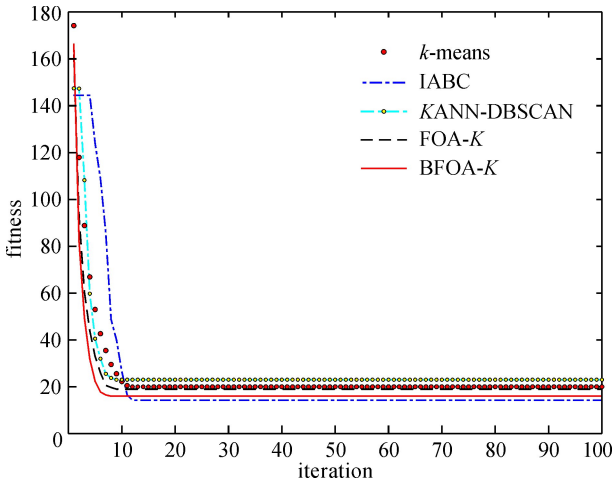


图 3　5 种算法在 Iris 数据集迭代图

Fig. 3　Iterative graphs of five algorithms in Iris dataset

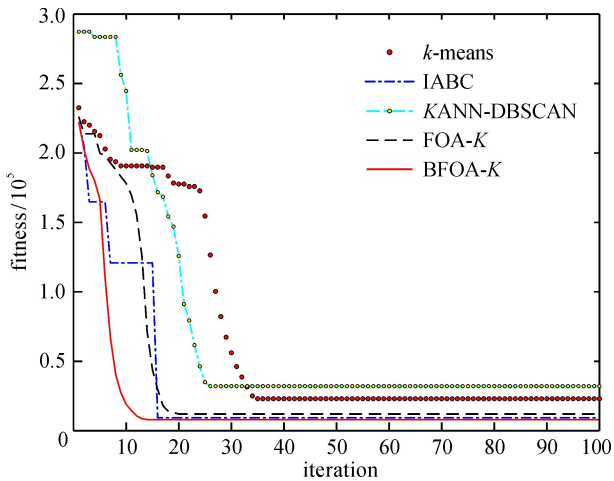


图 4　5 种算法在 Yeast 数据集迭代图

Fig. 4　Iterative graphs of five algorithms in Yeast dataset

有更好的适应度，因 BFOA-*K* 在聚类过程中注重的是聚类效果和收敛速度的平衡，所以 BFOA-*K* 的聚类效果与 IABC 差距并不大，且 BFOA-*K* 的收敛速度明显快于 IABC。在样本数量和属性数均不多的 Iris 数据集中，5 个算法的性能都极为相近，但是加入 *F* 分布变化步长的 BFOA-*K* 在改变飞行半径的情况下，收敛速度和收敛程度都略优于其他算法。而在样本数量和属性数与 Iris 数据集差异明显的 Yeast 数据集的测试中，无论是收敛速度还是收敛程度，BFOA-*K* 的目标函数适应值都要优于其他算法，这主要得益于精英策略对果蝇种群搜索范围的优化和 *F* 分布的飞行半径的双重影响。在样本数量不高和维度较低的数据集中，*F* 分布能发挥大部分的作用，在高维高数据样本的数据集中，两个改进策略则能共同发挥作用，改善聚类效果。通过 Iris 和 Yeast 两个数据集的对比实验，证明了本文

算法能在不同数据集上在更短的时间内拥有更好的聚类效果，证明了算法性能的有效性。

4 结束语

本文针对 k-means 聚类算法对于初始质心的敏感和容易陷入局部最优的问题，提出了基于改进果蝇优化算法的 k-means 算法(BFOA-K)。通过分析 k-means 算法和 FOA 算法的特点，在融合算法的同时改进了 FOA，通过加入 F 分布，利用其特性作为动态调整果蝇的飞行半径；利用精英果蝇，拓展了果蝇的搜索范围；利用种群的多样性提升了算法的稳定性。采用 4 个 UCI 的标准数据集进行仿真实验，将本文算法与多种算法进行比较，结果显示，在各个评价指标中本文提出的算法都有良好的聚类效果，证明了算法的有效性，并用两个对比数据集进行测试，结果表明无论在收敛速度还是收敛程度本文的 BFOA-K 算法都有着优异的表现，有效证明了算法的稳定性。

本文算法在针对高维和高数据样本的数据集中聚类效果没有体现出明显的优异性，今后需要结合其他改进策略，对算法进行更进一步的研究，得到更加完善的结果。

参考文献：

[1] 牛秦洲，陈艳. 基于 MCL 与 KNN 的混合聚类算法 [J]. 桂林理工大学学报，2015，35 (1)：181-186.

[2] Li H F，Fan R，Shi Q S，et al. Class imbalanced fault diagnosis via combining K-means clustering algorithm with generative adversarial networks[J]. Journal of Advanced Computational Intelligence and Intelligent Informatics，2021，25(3)：346-355.

[3] Zhang S，Shi S，Feng T M，et al. Trajectory planning in UAV emergency networks with potential underlaying D2D communication based on K-means[J]. EURASIP Journal on Wireless Communications and Networking，2021(2021)：107. https://doi.org/10.1186/s13638-021-01987-3.

[4] Fan Y K，Bai J R，Lei X，et al. PPMCK：privacy-preserving multi-party computing for K-means clustering [J]. Journal of Parallel and Distributed Computing，2021，154：54-63.

[5] 喻金平，郑杰，梅宏标. 基于改进人工蜂群算法的 K 均值聚类算法 [J]. 计算机应用，2014，34 (4)：1065-1069.

[6] 汤深伟，贾瑞玉. 基于改进粒子群算法的 k 均值聚类算法[J]. 计算机工程与应用，2019，55 (18)：140-145.

[7] 申燕萍，顾苏杭，郑丽霞. 基于云计算平台的仿生优化聚类数据挖掘算法[J]. 计算机科学，2019，46(11)：247-250.

[8] Pan W T. A new Fruit Fly Optimization Algorithm：taking the financial distress model as an example [J]. Knowledge-Based Systems，2012，26：69-74.

[9] 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.

[10] 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，2014，62：69-83.

[11] 宗序平，赵俊，陶伟. 统计学上三大分布推导方法 [J]. 数学的实践与认识，2009，39 (7)：215-220.

[12] 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.

[13] 李文杰，闫世强，蒋莹，等. 自适应确定 DBSCAN 算法参数的算法研究 [J]. 计算机工程与应用，2019，55 (5)：1-7.

Improved clustering algorithm based on bionic optimization algorithm

QIN Cheng-you，XIE Xiao-lan，WANG Yue-yue，GUO Yang

(a. Guangxi Key Laboratory of Embedded Technology and Intelligent System；b. College of Information Science and Engineering，Guilin University of Technology，Guilin 541006，China)

Abstract：To improve the clustering of the clustering algorithm，the bionic optimization algorithm and the k-means clustering algorithm are combined to realize data clustering(BFOA-K). In the clustering process，the fruit fly optimization algorithm is used to determine the centroid of the k-means clustering algorithm，and the k-means clustering algorithm is used for data clustering. To solve the problem that k-means is sensitive to the initial centroid and easy to fall into the local optimum，and the influence of the fruit fly optimization algorithm on the flight step length，the F distribution is used to dynamically change the step length which improves the global search ability of the algorithm. The elite retention strategy is adopted to increase the diversity of the fruit fly population，expand the scope of the search and improve the search efficiency. Four UCI standard datasets are used to test the algorithm. The experiment shows that the algorithm is better than other comparison algorithms in all clustering evaluation indicators. The BFOA-K improves the convergence of the algorithm with effectiveness and feasibility.

Key words：bionic optimization algorithm；k-means clustering；fruit fly optimization algorithm；F distribution；elite retention strategy

(编辑 王蓉嵘)