第 29 卷 第 5 期
Vol. 29 No. 5

控 制 与 决 策
Control and Decision

2014 年 5 月
May 2014

文章编号: 1001-0920 (2014) 05-0838-05 DOI: 10.13195/j.kzyjc.2013.0101

基于变异精密搜索的蜂群聚类算法

罗 可, 李 莲, 周博翔
(长沙理工大学 计算机与通信工程学院，长沙 410114)

摘 要: 针对 K-means 聚类算法过度依赖初始聚类中心、局部收敛、稳定性差等问题, 提出一种基于变异精密搜索的蜂群聚类算法. 该算法利用密度和距离初始化蜂群, 并根据引领蜂的适应度和密度求解跟随蜂的选择概率 P; 然后通过变异精密搜索法产生的新解来更新侦查蜂, 以避免陷入局部最优; 最后结合蜂群与粗糙集来优化 K-means. 实验结果表明, 该算法不仅能有效抑制局部收敛、减少对初始聚类中心的依赖, 而且准确率和稳定性均有较大的提高.

关键词: 聚类；粗糙集；人工蜂群；K-means；变异算子
中图分类号: TP18 **文献标志码:** A

Artificial bee colony rough clustering algorithm based on mutative precision search

LUO Ke, *LI Lian*, *ZHOU Bo-xiang*
(Institute of Computer and Communication Engineering，Changsha University of Sciences and Technology，Changsha 410014，China．Correspondent：LI Lian，E-mail：lilianhappy2012@163.com)

Abstract: For the problems of the traditional K-means clustering algorithm such as depending overly on initial clustering centers, the poor global search ability and stability, an artificial bee colony rough clustering algorithm based on mutative precision search is proposed, which generates initial swarm by density and distance, and gets the selection probability of onlooker bees according to the fitness and density of lead bees, then updates scout bees through the method of mutative precision search, in order to avoid falling into local optimum. Finally, the rough set is combined to optimize K-means. The experiment results show that this algorithm not only can suppress the local convergence effectively and reduce the dependence on the initial cluster center, but also has higher accuracy and stronger stability than others.

Key words: clustering；rough set；artificial bee colony；K-means；mutation operator

0 引 言

聚类就是将数据对象分成多个簇, 同一个簇中对象之间具有较高的相似度, 而不同簇中对象差别较大. 聚类分析现已成为数据挖掘研究领域中一个非常活跃的研究课题[1].

K-means 算法因简单、局部收敛性好等特点而被广泛应用, 但该算法依然存在对初始中心敏感, 不能处理边界对象等缺陷. 鉴于此, 不少学者通过结合遗传算法[2-3]、粗糙集[4]、差分演化[5]、POS[6]、重力搜索[7]、ACO[8-9]、人工蜂群[10-11]等算法, 对 K-means 的不足加以改进, 取得了一定的效果, 但对于复杂问题还存在精度不高、效率低等问题.

人工蜂群算法 (ABC) 通过模拟蜂群采蜜来实现对复杂问题的求解. ABC 算法具有全局搜索能力强、鲁棒性好等优点, 因而得到了广泛研究. 文献 [12] 提出了正弦函数的初始化策略和基于最优解的搜索策略的 ABC 算法. 文献 [13] 在文献 [12] 的基础上提出了改进的搜索策略. 文献 [14] 通过定义个体的调整系数和个体与群体间的差异系数实现 ABC 算法全局探索和局部开发能力的平衡. 此外, ABC 算法也得到了广泛的应用. 文献 [10] 引用蜂群交配优化算法, 增强了 K-mean 算法的聚类效果及效率. 文献 [11] 结合人工蜂群算法优化 K-mean 聚类算法. 文献 [15] 提出了一种多父系人工蜂群聚类算法, 算法计算被分配到多个处理单元, 显著减少了聚类算法的处理时间. 这些算法各有特色, 但无法保证全局探索和局部开发能

收稿日期: 2013-01-22；**修回日期:** 2013-04-10.
基金项目: 国家自然科学基金项目(11171095, 71371065)；湖南省科技计划项目(2013SK3146)；湖南省自然科学衡阳联合基金项目(10JJ8008).
作者简介: 罗可(1961−), 男, 教授, 博士, 从事数据挖掘、计算机应用等研究；李莲(1987−), 女, 硕士生, 从事数据库技术、数据挖掘的研究.

力的平衡, 易陷入局部极值和早熟收敛. 因此, 本文提出了一种变异精密搜索的 ABC 算法. 该算法从蜂群初始化、跟随蜂的选择概率 P、跟随蜂搜索方法 3 个方面改进了传统 ABC 算法的不足, 平衡了全局探索和局部开发能力, 增强了算法的全局搜索能力. 本文融合改进的 ABC 算法、粗糙集以及 K-means 各自的优点, 提出了一种新的聚类算法, 并通过仿真实验验证了其有效性.

1　相关知识介绍

1.1　K-means 聚类算法简介

K-means 算法从给定的样本集 $X = \{x_1, x_2, \cdots, x_n\}$ 中找到 k 个聚类中心 $\{a_1, a_2, \cdots, a_k\}$, 按最小距离原则将所有样本分配到对应的类 C_i 中, 从而将样本集划分为 k 个簇 $C_1, C_2, \cdots, C_k$; 按 $c_i = \dfrac{1}{|C_i|}\sum\limits_{x\in C_i} x$ 更新聚类中心, 其中 $|C_i|$ 为第 i 个簇的样本数; 再按最小距离原则更新样本的所属类, 根据使函数 $E = \sum\limits_{i=1}^{k}\sum\limits_{x\in C_i}|x-a_i|^2$ 最小准则, 进行迭代, 直到簇中心不变[16].

1.2　粗糙集理论

粗糙集理论[17]主要研究不精确和模糊的知识, 在数据挖掘领域得到了成功应用. 下面给出粗糙集中与本文相关的一些定义.

定义 1 (上近似、下近似及边界集)　给定知识库 $K = (U, R)$, 对于 $X \neq \varphi$ 且 $X \subseteq U$, 存在一个等价关系 $R \in \mathrm{ind}(K)$, 称 $\underline{R}X = \bigcup\{Y \in U/R\,|Y \subseteq X\}$ 为 X 关于 R 的下近似, $\overline{R}X = \bigcup\{Y \in U/R|Y \bigcap X \neq \varphi\}$ 为 X 关于 R 的上近似, $\mathrm{BNR}(X) = \overline{R}X - \underline{R}X$ 则称为 X 的 R 边界区域.

定义 2 (粗糙集)　若 $\overline{R}X \neq \underline{R}X$, 则 X 为 R 的粗糙集, 否则称 X 为 R 精确集.

2　ABC 算法及其改进

2.1　ABC 算法

首先随机生成含有 SN 个解的初始蜂群 $\{Z_1, Z_2, \cdots, Z_{\mathrm{SN}}\}$, 每个解 Z_i 是一个 D 维的向量; 然后, 引领蜂根据下式:

$$v_{ij} = z_{ij} + \varphi_{ij}(z_{ij} - z_{kj}) \tag{1}$$

进行邻域搜索, 比较记忆最优解和邻域搜索解, 当搜索解优于记忆最优解时, 替换记忆解; 反之, 保持不变. 其中: $j \in \{1, 2, \cdots, D\}$, $k \in \{1, 2, \cdots, \mathrm{SN}\}$, k 随机选取且 $k \neq i$; φ_{ij} 为 $[-1, 1]$ 之间的随机数; v_{ij} 代表新位置.

跟随蜂根据轮盘赌原则以概率

$$p_i = \frac{\mathrm{fit}_i}{\sum\limits_{n=1}^{\mathrm{SN}}\mathrm{fit}_n} \tag{2}$$

选择引领蜂, 并在引领蜂附近进行邻域搜索产生新解.

如果引领蜂 Z_i 连续 Limit 次迭代没有改变, 则将引领蜂变成侦察蜂, 通过下式随机搜索一个新解来替代 Z_i:

$$z_i^j = z_{\min}^j + \mathrm{rand}(0,1)(z_{\max}^j - z_{\min}^j). \tag{3}$$

其中: $j \in \{1, 2, \cdots, D\}$, $z_{\min}^j$ 和 $z_{\max}^j$ 表示所有蜜蜂中第 j 维的最小值和最大值.

2.2　变异精密搜索的 ABC 算法

2.2.1　蜂群初始化的改进

传统的 ABC 算法随机选取初始蜂群, 很难保证优良蜜蜂的存在, 且蜂群在备选解空间中分配不均, 也影响了算法的整体性能. 鉴于此, 本文提出了一种基于密度和最大最小距离的蜂群初始化方法, 具体步骤如下.

Step 1: 计算任意两个样本间的距离 $\mathrm{dist}(x_i, x_j)$, 并将其记录在矩阵 D 中, 则样本间的平均距离为 $\overline{d} = \sum \mathrm{dist}(x_i, x_j)/n^2, i, j = 1, 2, \cdots, n.$

Step 2: 样本 x_i 的密度定义为 $\mathrm{Density}(x_i) = \{p \in C\,|\mathrm{dist}(x_i, p) \leqslant r\}$, 表示以 x_i 为中心点, r 为半径组成的球体中所包含的样本数. 其中 $r = \alpha^*\overline{d}$ 为半径, α 为常数, C 为样本集. 样本集平均密度为

$$\mathrm{MDensity}(x_i) = \frac{\sum\limits_{i=1}^{n}\mathrm{Density}(x_i)}{n}.$$

根据 $\mathrm{Density}(x_i) \leqslant M\mathrm{Density}(x_i)/4$ 将孤立样本从 C 中排除.

Step 3: 采用最大最小距离法得到初始聚类中心. 密度最大的样本为第 1 个聚类中心 v_1, 与其距离最远的为第 2 个聚类中心 v_2, 对于 C 中剩余样本, 根据矩阵 D, 分别求出其中心到 $v_1, v_2, \cdots, v_m$ 的距离为 $d_{i1}, d_{i2}, \cdots, d_{im}$, 取 $d_i = \min(d_{i1}, d_{i2}, \cdots, d_{im})$, $d = \max(d_i)$ 对应的粒子中心为 v_i, 以此类推计算 v_k. 将得到的 k 个聚类中心作为初始化一个蜜蜂的位置编码, 并计算蜜蜂的适应度值, 得到第 1 个蜜蜂. 反复执 SN 次, 生成含 SN 个蜜蜂的初始蜂群 $\{Z_1, Z_2, \cdots, Z_{\mathrm{SN}}\}$.

实验表明, 采用这种方法生成的初始蜂群性质优良, 可加快算法的收敛速度.

2.2.2　概 率 p_i

本文提出了由引领蜂的适应度 fit_i 和密度 ρ_i 共同决定跟随蜂的选择概率, 平衡了传统的 ABC 算法全局探索和局部开发能力, 改善了只由适应度计算概

率而使算法陷入局部最优的缺点, 即

$$p_i = \frac{\rho_i \mathrm{fit}_i}{\sum_{n=1}^{\mathrm{SN}} \rho_n \mathrm{fit}_n}, \tag{4}$$

其中 ρ_i 表示第 i 只蜜蜂所表示的 k 个聚类中心点的平均密度. 聚类中心点的密度越大, 说明其越接近最优聚类中心点, 其所代表的引领蜂被选择的概率越大.

2.2.3 侦查蜂根据变异精密搜索法搜索新解

若连续迭代 Limit 次, Z_i 引领蜂的适应度不变, 且不是当前全局最优, 则将该引领蜂变成侦察蜂, 通过变异精密搜索法产生 s 个新解, 选择适应度最大的新解替代 Z_i. 具体的变异精密搜索法如下.

蜜蜂 $Z_i(z_{i1}, z_{i2}, \cdots, z_{ik})$ 根据下式:

$$z'_{ij} = z_{ij} + \mathrm{rand}(0,1)\,|z_{\mathrm{best}} - z_{ij}| \tag{5}$$

变异 k 维中的一维, 产生 C_k^1 个新解; 变异两维产生 C_k^2 个新解. 以此类推, 变异 k 维产生 C_k^k 个新解, 总共产生 $s = C_k^1 + C_k^2 + \cdots + C_k^k$ 个新解. 其中 z_{ij} 为蜜蜂当前解, z_{best} 为全局最优解. 根据式 (5) 变异后的 s 个新个体为

C_k^1个新解 : $(z'_{i1}, z_{i2}, \cdots, z_{ik}), \cdots, (z_{i1}, z_{i2}, \cdots, z'_{ik})$;

C_k^2个新解 : $(z'_{i1}, z'_{i2}, \cdots, z_{ik}), \cdots, (z_{i1}, z_{i2}, \cdots, z'_{ij}, z'_{ik})$;

$\vdots$

C_k^k个新解 : $(z'_{i1}, z'_{i2}, \cdots, z'_{ij}, z'_{ik})$.

传统的随机搜索法搜索到最优解的可能性很小, 而变异精密搜索算法可以在局部最优解与全局最优解之间产生许多邻域点, 选择最好的解来替代原来的解, 以此来完成侦查蜂的搜索, 帮助蜂群跳出局部最优而精密快速搜寻到全局最优解.

3 改进的人工蜂群粗糙聚类算法

3.1 适应度函数

类内距离是用来评价聚类的内聚程度, 即

$$J_w = \sum_{i=1}^{k} \Big(\omega_l \sum_{x_j \in \underline{c}_{il}} \|x_j - v_i\|^2 + \omega_{bnr} \sum_{x_j \in c_{ibnr}} \|x_j - v_i\|^2\Big). \tag{6}$$

其中: v_i、$\underline{c}_{il}$、c_{jbnr} 分别为第 i 类的聚类中心、下近似集和边界集, ω_l、ω_{bnr} 分别为第 i 类的下近似集和边界集的权重.

适应度函数表示为

$$\mathrm{fit}_i = \frac{1}{J_m}. \tag{7}$$

3.2 新的聚类中心

新的聚类中心

$$v_j = \begin{cases} \dfrac{\omega_l}{|\underline{c}_{jl}|} \displaystyle\sum_{x_i \in \underline{c}_{jl}} x_i + \dfrac{\omega_{bnr}}{|c_{jbnr}|} \displaystyle\sum_{x_i \in c_{jbnr}} x_i,\ c_{jbnr} \neq \varphi; \\ \dfrac{\omega_l}{|\underline{c}_{jl}|} \displaystyle\sum_{x_i \in \underline{c}_{jl}} x_i,\ \mathrm{else}. \end{cases} \tag{8}$$

其中: $j = 1, 2, \cdots, k$, $|\underline{c}_{jl}|$ 和 $|c_{jbnr}|$ 分别表示下近似集和边界集的样本个数.

3.3 粗糙集聚类算法

粗糙 K-means 聚类算法是将样本划分到类的上近似集和下近似集中, 求出边界集. 通过边界集和下近似来计算新的聚类中心, 然后重新划分样本集, 此过程迭代进行, 直到形成稳定的聚类结果. 算法步骤如下.

Step 1: 确定初始聚类中心 $v_1, v_2, \cdots, v_k$, 其中 k 为聚类数目.

Step 2: 根据近邻原则, 将每个样本 x 分配给最近的类的上下近似集, 即对 $\forall x_m \in U$, 找出与其距离最小的中心点 v_i, 有 $d(x_m, v_i) = \min\{d(x_m, v_j), j = 1, 2, \cdots, k\}$, 则 $x_m \in \overline{v}_i$; 如果 $\exists c_j$, 使得 $d(x_m, v_i) - d(x_m, v_j) < \gamma^* d(v_i, v_j)$, 则令 $x_m \in \overline{v}_j$, 否则 $x_m \in \underline{v}_i$.

Step 3: 根据式 (7) 计算适应度, 如果 $\|F(t) - F(t-1)\| \leqslant \varepsilon$ 或者 $t \geqslant t_{\max}$, 则结束; 否则, 根据式 (8) 更新聚类中心, 且 $t = t + 1$, 转 Step 2.

3.4 算法设计

1) 蜜蜂编码.

为了减少蜜蜂编码长度, 本文采用实数编码方式. 蜜蜂编码由蜜蜂的位置和适应度值组成, 蜜蜂的位置由 k 个聚类中心组成. 蜂蜜编码如下:

$$Z_i = (z_{i1}, z_{i2}, \cdots, z_{ik}, \mathrm{fit}_i).$$

2) 算法步骤.

Step 1: 生成初始蜂群 $\{Z_1, Z_2, \cdots, Z_{\mathrm{SN}}\}$.

Step 2: 根据蜜蜂的适应度值反序排序, 前 50% 为引领蜂, 后 50% 为跟随蜂. 每只引领蜂根据式 (1) 做邻域搜索产生新解. 当所有引领蜂领域搜索完成后, 根据式 (4) 计算概率 p_i.

Step 3: 每只跟随蜂按轮盘赌原则, 依概率 p_i 选择引领蜂后, 根据式 (1) 做邻域搜索产生新解.

Step 4: 侦察蜂通过变异精密搜索法搜索新解.

Step 5: 对每只蜜蜂进行一次粗糙聚类, 用得到的新的聚类中心更新蜂群.

Step 6: 若当前迭代次数大于最大次数 MCN, 则停止迭代; 否则, 转到 Step 2, 且 $t = t + 1$.

3.5 算法时间复杂度分析

在 Step 1 中, 蜂群初始化的时间复杂度为 $O(n{*}n + k{*}\mathrm{SN})$; Step 2 中引领蜂邻域搜索以及计

算概率的时间复杂度为 $O(\mathrm{SN}*\log\mathrm{SN}+\mathrm{SN}/2+k*n*\mathrm{SN})$; Step 3 中跟随蜂邻域搜索的时间复杂度为 $O(k*\mathrm{SN}/2)$; Step 4 中侦查蜂搜索的时间复杂度为 $O(s*M)$, 其中 s 为搜索解个数, M 为侦查蜂个数; Step 5 中粗糙集聚类的时间复杂度为 $O(k*n*\mathrm{SN})$. 因为 $k<M<s\ll\mathrm{SN}\ll n$, 且算法总共迭代 MCN, 所以算法总的时间复杂度为 $O(n*n*\mathrm{MCN})$.

4　实验结果与分析

实验中的操作系统为 Windows XP, 集成开发环境为 Microsoft Visual C++ 6.0; Matlab 7.0. 硬件条件为: Intel(R) Core(TM) i3-2100 CPU @3.10 GHz, 4 GB 的内存.

为验证本文算法的有效性和可行性, 将本文算法在 Iris 和 Wine 标准数据集上的测试结果与 K-means、GA、K-NM-PSO[6]、ACO[8]、PSO-ACO-K[9]、HBMO[10]、ABC1[11]、(ABC2)[15]进行比较. 各数据集的特征如表 1 所示. 经过 10 次实验, 本文算法效果最好时各数据集中各参数的设置如下:

蜂群个数 SN= 100, 最大循环次数 MCN = 1 000, Limit = 15. Iris 中参数选择 $\omega_l=0.7$, $\omega_{bnr}=0.3$, $\alpha=0.96$, $\gamma=0.2$. Wine 中参数选择 $\omega_l=0.72$, $\omega_{bnr}=0.28$, $\alpha=0.145$, $\gamma=0.3$.

表 1　实验中涉及的数据集

数据集名称	样本数目	属性维数	类别数
Iris	150	4	3
Wine	178	13	3

在 Iris、Wine 数据集上, 本文算法的适应度随迭代次数增加的变化情况分别如图 1 和图 2 所示.

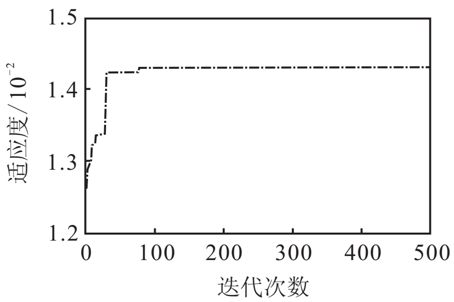

图 1　Iris 数据集上适应度变化情况

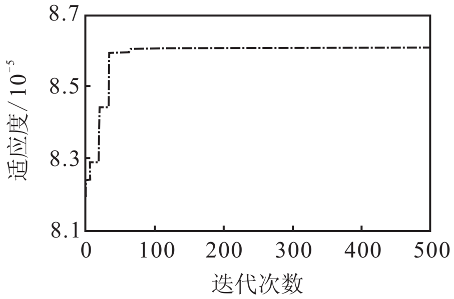

图 2　Wine 数据集上适应度变化情况

本文中最大迭代次数 MCN 为 1 000, 由于 500 到 1 000 次迭代的适应度值均没有变化, 只画了前 500 次迭代的适应度图, 以便更清楚观察适应度的变化情况. 由图 1 和图 2 可知: 本文算法在各数据集中适应度值变化范围很小, 而且很快便达到局部极值. 随着迭代次数的增加, 侦查蜂通过精密搜索法, 多次跳出局部最优, 最终达到全局最优. 而其他几种算法, 由于陷入局部最优, 或者没能多次跳出局部最优, 最终达不到全局最优解, 至使算法的准确率 (表 2、表 3) 远低于本文算法. 由此可见, 本文算法能有效地避免陷入局部极值, 具有很强的全局搜索能力和稳定性.

采用本文算法进行了 10 次独立实验, 每次运行都会产生不同的随机种子来得到相应的参数值, 对 10 次实验结果求平均值, 可得到平均准确率. 各类算法在 Iris、Wine 数据集上的类内距离、平均准确率及执行时间比较分别如表 2 和表 3 所示.

表 2　各算法在 Iris 数据集中的聚类结果比较

算法	类内距离			执行时间	平均准确率
	最小	平均	最大		
K-means	97.333	106.050	120.450	**0.4**	78.20
GA	113.987	125.197	139.778	105.53	77.80
ACO	97.101	97.172	97.808	33.72	77.90
K-NM-PSO	96.660	96.670	97.010	48.13	89.93
PSO-ACO-K	96.650	96.650	96.650	16.00	78.80
HBMO	96.752	96.953	97.758	35.25	78.10
ABC1	78.940	78.940	78.940	29.68	—
ABC2	96.650	96.650	96.650	8.54	89.80
本文算法	**69.865**	**70.124**	**71.032**	17.23	**94.44**

表 3　各算法在 Wine 数据集中的聚类结果比较

算法	类内距离			执行时间	平均准确率
	最小	平均	最大		
K-means	16 555.68	18 061.00	18 563.12	**0.7**	52.10
GA	16 530.534	16 530.534	16 530.534	226.68	51.50
ACO	16 530.534	16 530.534	16 530.534	68.29	51.90
K-NM-PSO	16 292.000	16 293.000	16 279.460	589.40	71.91
PSO-ACO-K	26 295.310	26 295.310	26 295.310	30.00	52.10
HBMO	16 357.284	16 357.284	16 357.284	55.18	51.80
ABC1	16 257.280	16 260.520	16 279.460	48.85	—
ABC2	16 292.180	16 292.870	16 294.170	90.02	72.40
本文算法	**11 626.200**	**11 631.600**	**11 652.400**	67.39	**84.27**

通过比较表 2 和表 3 各项实验结果可知: 本文算法的最大、最小、平均类内距离均远远小于其他几种算法, 在 Wine 数据集上表现尤为明显, 比 ABC2 算法减小了 28.47%. 说明本文算法稳定性得到了很大提高, 聚类效果优于其他算法. 聚类准确率方面, 本文算法也优于其他算法, 在 Iris 数据集中, 比最优的 K- NM -PSO 算法高出 5.02%; 在 Wine 数据集中, 比最优的 ABC1 算法高出 16.39%. 在执行时间方面, K-means 的执行时间最短, 其他几种算法因为加入智能优化

算法, 导致时间远大于 K-means. 而在几种智能优化算法中, ABC1 在 Iris 数据集中的执行时间最短, 但在 Wine 数据集中却相对较高, 说明 ABC1 对高维数据集的适应性较差. 本文算法在 K-means 的基础上加上了 ABC 以及粗糙集, 所以执行时间增长了, 但相对于 GA、ACO、K-NM-PSO 还是有所提高. 由此可知, 本文算法对高维数据集具有较好的适应性, 具有较强的鲁棒性.

5 结 论

本文提出了一种变异精密搜索的 ABC 算法, 并结合粗糙集来优化 K-means 算法. 蜂群初始化采用密度和最大值小距离法, 使初始蜂群在备选解空间中均匀分布, 且性质优良, 加快了算法的收敛速度; 根据引领蜂的适应度和密度共同决定它被跟随蜂选择的概率, 平衡了全局探索和局部开发能力; 根据变异精密搜索法为侦察蜂产生新解, 能有效地跳出局部最优而找到全局最优解, 增强了算法的全局搜索能力. 虽然本文算法在执行时间方面改进较小, 但在正确率和稳定性方面远高于其他几种算法, 而且本文算法对高维数据集具有较好的适应性. 当然, 本文算法也存在着一些不足: 蜂群在寻优过程中搜索行为的自适应调整, 参数 k 值能否自适应的给出, 将成为下一步的研究方向.

参考文献(References)

[1] Han J W, Kamber M. Data mining: Concepts and techniques[M]. New York: Morgan Kaufmann Publishers, 2001: 335-388.

[2] Chang D X, Zhang X D, Zheng C W. A genetic algorithm with gene rearrangement for K-means clustering[J]. Pattern Recognition, 2009, 42(7): 1210-1222.

[3] Xiao J, Yan Y P, Zhang J, et al. A quantum inspired genetic algorithm for k-means clustering[J]. Expert Systems with Applications, 2010, 37(7): 4966-4973.

[4] Chen J Y, Zhang C S. Efficient clustering method based on rough set and genetic algorithm[J]. Procedia Engineering, 2011, 15: 1498-1503.

[5] Wojciech Kwedlo. A clustering method combining differential evolution with the K-means algorithm[J]. Pattern Recognition Letters, 2011, 32(12): 1613-1621.

[6] Kao Y T, Zahara E. A hybridized approach to data clustering[J]. Expert Systems with Applications, 2008, 34(3): 1754-1762.

[7] Abdolreza H, Salwani A. A combined approach for clustering based on K-means and gravitational search algorithms[J]. Swarm and Evolutionary Computation, 2012, 6: 47-52.

[8] Shelokar P S, Jayaraman V K. An ant colony approach for clustering[J]. Analytica Chimica Acta, 2004, 509(2): 187-195.

[9] Taher N, Babak A. An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis[J]. Applied Soft Computing, 2010, 10(1): 183-197.

[10] Mohammad F, Babak A. Application of honey-bee mating optimization algorithmon clustering[J]. Applied Mathematics and Computation, 2007, 190(2): 1502-1513.

[11] Zhang C S, Ouyang D T, Ning J X. An artificial bee colony approach for clustering[J]. Expert Systems with Applications, 2010, 37(7): 4761-4767.

[12] Gao W F, Liu S Y. A modified artificial bee colony algorithm[J]. Computers & Operations Research, 2012, 39(3): 687-697.

[13] Gao W F, Liu S Y, Huang L L. A global best artificial bee colony algorithm for global optimization[J]. J of Computational and Applied Mathematics, 2012, 236(11): 2741-2753.

[14] 刘勇, 马良. 函数优化的蜂群算法[J]. 控制与决策, 2012, 27(6): 886-890.
(Liu Y, Ma L. Bees algorithm for function optimization[J]. Control and Decision, 2012, 27(6): 886-890.)

[15] Anan B, Booncharoen S, Tiranee A. The best-so-far ABC with multiple patrilines for clustering problems[J]. Neurocomputing, 2013, 116: 355-366.

[16] Macqueen J. Some methods for classification and analysis of multivariate observations[C]. Proc of the 5th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley: University of California Press, 1967: 281-297.

[17] 王彪, 段禅伦, 吴昊, 等. 粗糙集与模糊集的研究及应用[M]. 北京: 电子工业出版社, 2008: 1-4.
(Wang B, Duan C L, Wu H. The research and applications of rough set and fuzzy set[M]. Beijing: Publishing House of Electronics Industry, 2008: 1-4.)

(责任编辑: 孙艺红)