◆生物信息◆

基于模糊逻辑 COOT 优化 K 调和均值的数据聚类算法*

戴峦岳[1,2],梁宵月[1],王　帅[1],王震坡[2**]
(1. 北京牡丹电子集团有限公司博士后科研工作站,北京　100089;2. 北京理工大学电动车辆国家工程实验室,北京　100081)

摘要:针对 K 调和均值(K-Harmonic Means,KHM)聚类算法易陷入局部最优的不足,本文结合 KHM 聚类算法的快速局部开发和白骨顶鸡优化算法(Coot optimization algorithm,COOT)的全局勘探能力,提出一种模糊逻辑 COOT 优化 KHM 的数据聚类算法(Fuzzy COOT K-Harmonic Means,FCOOTKHM)。将 KHM 聚类算法生成的初始聚类解输入白骨顶鸡初始种群结构再进行迭代寻优。同时,为了进一步提升 COOT 的搜索精度,设计模糊逻辑对 COOT 的收敛因子和领导者种群占比进行自适应调整,均衡算法的搜索与开发能力。使用聚类调和平均值评估种群个体的适应度,结合智能算法启发式搜索对聚类结果迭代寻优。利用加州大学欧文分校(University of California Irvine,UCI)数据库中的 7 个数据集对 FCOOTKHM 的聚类性能进行验证分析。结果表明,FCOOTKHM 在准确率、精确度、召回率、F 度量、Kappa 系数和收敛效率等指标上均表现更好,该算法能够实现更精确的数据聚类。

关键词:模糊逻辑;模糊系统;白骨顶鸡优化算法;K 调和均值;聚类;收敛性

中图分类号:TP393　　**文献标识码**:A　　**文章编号**:1005-9164(2024)05-0900-12
DOI:10.13656/j.cnki.gxkx.20240526.001

数据聚类是模式识别、数据挖掘和机器学习领域的重要手段[1],尤其对车联网大数据的高效存储极为关键。作为一种无监督学习方法,数据聚类分析的目标是将数据集内的数据对象进行聚类划分,将数据对象划分为不相交子集,使得相同聚类内数据具有更高相似度,而不同聚类内的数据对象相似性较低[2]。K 均值(K-Means,KM)聚类算法是一种最简单且高效的聚类算法,但具有明显的初始聚类质心敏感性高且易于过早收敛、陷入局部最优的不足[3]。为了克服 KM 聚类算法的不足,K 调和均值(K-Harmonic Means,KHM)聚类算法[4]利用平均调和均值获取新的聚类中心,在一定程度上克服了初始质心敏感性问题,但模型对于噪声仍有敏感性,且依然存在局部最优问题。近年来,融入智能优化算法的数据聚类分析手段得到了广泛研究,借助智能算法具有更加强大的全局搜索性能优势,使聚类划分具有更高的相似度,

收稿日期:2023-11-24　　修回日期:2023-12-14
*北京市博士后工作经费资助项目(202304013)资助。
【第一作者简介】
戴峦岳(1986—),男,博士,主要从事大数据分析与车联网研究。
【**通信作者简介】
王震坡(1976—),男,博士,教授,博士生导师,主要从事新能源汽车大数据分析研究,E-mail:wangzhenpo@bit.edu.cn。
【引用本文】
戴峦岳,梁宵月,王帅,等. 基于模糊逻辑 COOT 优化 K 调和均值的数据聚类算法[J]. 广西科学,2024,31(5):900-911.
DAI L Y,LIANG X Y,WANG S,et al. A Data Clustering Algorithm Based on Fuzzy COOT K-Harmonic Means [J]. Guangxi Sciences,2024,31(5):900-911.

从而找到全局最优解[5]。如王永刚等[6]提出一种基于和声搜索(Harmony Search,HS)的文本数据聚类算法,但和声搜索算法本身种群多样性不足,且解的质量与记忆考虑、微调扰动因子密切相关,无法保证最优。张雪峰等[7]、娄奥等[8]引入引力搜索机制并结合 KM 聚类算法实现数据聚类,但由于缺少记性属性,算法容易出现早熟收敛。Babu 等[9]提出一种细菌优化算法的聚类方法,但其收敛性能具有不确定性。Li 等[10]提出基于粒子群优化的聚类方法,但算法开采能力不足,容易陷入局部最优。叶廷宇等[11]、王海玲等[12]利用人工蜂群算法优化 KM 聚类的初始质心,提高了聚类准确性,但其问题在于维度增加后算法收敛效率下降较多。Jin 等[13]则将混沌机制引入蜂群算法,在一定程度上提升了种群多样性和算法收敛速度。除此之外,混合智能优化算法[14]、变异萤火虫算法[15]和烟花算法[16]也都在数据聚类优化问题中得到了广泛应用。

白骨顶鸡优化算法(Coot optimization algorithm,COOT)[17]是一种模拟白骨顶鸡在水面上以不同运动模式捕食的新型启发式搜索算法。该算法关键参数少、模型简单且易于实现,在基准函数寻优问题上,也体现出优于粒子群算法、引力搜索算法和灰狼算法的搜索精度和收敛速度。COOT 也在机器学习模型优化[18]和无线传感器网络(Wireless Sensor Network,WSN)节点覆盖优化[19]问题上得到了有效验证。然而,处理数据集聚类这类高维优化问题时,COOT 的搜索精度与收敛因子调整、种群搜索的自适应性高度相关,造成算法依然存在搜索精度差、收敛慢的不足。为了解决 KHM 聚类算法求解聚类易陷入局部最优的不足,本文提出一种模糊逻辑 COOT 优化 KHM 的数据聚类算法(Fuzzy COOT K-Harmonic Means,FCOOTKHM),利用模糊逻辑机制提升 COOT 的搜索自适应性以及全局搜索能力,改进 COOT 和 KHM 聚类算法求解数据聚类问题,并提高 COOT 的全局搜索能力和 KHM 聚类算法的局部寻优能力,准确高效地搜索聚类中心最优解。同时,通过加州大学欧文分校(University of California Irvine,UCI)数据库,验证改进算法在求解精度和收敛速度上的性能。

1 模型构建

1.1 标准 COOT

白骨顶鸡种群有 3 种行为模式:链式运动、领导者运动和追随者运动。整个种群会以适应度高低划分为领导者和追随者。领导者具有更强的逼近食物源的能力。追随者的位置更新具有两种模式:主动更新和被动更新;主动更新时,追随者可以随机运动和链式运动模式更新位置,丰富搜索的多样性,此时不依赖于领导者;被动更新时,追随者受到领导者的牵引作用而向其逼近。

白骨顶鸡初始种群位置生成方式为

$$CootPos(i)=r(1,d)\times(ub-lb)+lb, \tag{1}$$

式中,$r(\cdot)$为[0,1]内的随机值,d 为搜索维度,[lb,ub]为个体的搜索范围,$CootPos(i)$为个体 i 的位置。

①追随者位置更新

追随者可按概率进行位置主动更新和位置被动更新。主动更新又可分为随机运动和链式运动两种模式。在随机运动中,首先以公式(2)生成个体的一个随机位置:

$$Q=r(1,d)\times(ub-lb)+lb, \tag{2}$$

式中,Q 为随机生成的个体位置。

受到随机位置的牵引,白骨顶鸡个体会向随机位置的方向移动,此时位置更新方式定义为

$$FollowPos(i)=FollowPos(i)+A\times R_1\times(Q-FollowPos(i)), \tag{3}$$

式中,$FollowPos(i)$为追随者 i 的位置,$R_1\in[0,1]$为随机值,收敛因子 A 的公式定义为

$$A=1-\frac{t}{T_{\max}}, \tag{4}$$

式中,t 为当前迭代次数,$T_{\max}$ 为最大迭代次数。

在链式运动中,白骨顶鸡个体会在两个相邻的个体间进行链式运动,此时位置更新方式定义为

$$FollowPos(i)=0.5\times(FollowPos(i-1)+FollowPos(i)), \tag{5}$$

式中,$FollowPos(i-1)$为相邻前一个追随者个体的位置。

被动更新时,追随者受领导者牵引。选择的领导者为

$$k=1+(i \bmod N_L), \tag{6}$$

式中,k 为领导者个体索引,i 为追随者索引,N_L 为领导者个体的数量。此时追随者被动位置更新方式为

$$FollowPos(i)=LeadPos(k)+2\times R_2\times\cos(2R\pi)\times(LeadPos(k)-FollowPos(i)), \tag{7}$$

式中,$LeadPos(k)$为领导者 k 的位置,R、$R_2\in[0,1]$为随机值。

②领导者位置更新

领导者通常具有更强的逼近食物源的能力,会不断向最优解的区域靠近,因此主要受到全局最优解的引领。领导者位置更新方式为

$$LeadPos(i)=\begin{cases}B\times R_3\times\cos(2R\pi)\times(\mathrm{gBest}-LeadPos(i))+\mathrm{gBest}, R_4<0.5\\ B\times R_3\times\cos(2R\pi)\times(\mathrm{gBest}-LeadPos(i))-\mathrm{gBest}, R_4\geqslant 0.5\end{cases},\tag{8}$$

式中,$LeadPos(i)$为领导者 i 的位置,gBest 为当前种群的最优解,R_3、$R_4\in[0,1]$为随机值,收敛因子 B 的公式定义为

$$B=2-\frac{2t}{T_{\max}}。\tag{9}$$

对于 COOT,领导者种群的位置更新受到全局最优解的牵引,充当全局搜索的角色。但在算法迭代过程中,领导者种群的占比是固定不变的。这就决定了整个算法寻优过程中,全局搜索能力保持不变。然而,对于群体智能算法而言,迭代前期应以大规模全局搜索为主,即以较大领导者占比对解空间广泛搜索,加快算法收敛;迭代后期应加大局部开发比重,即以较小领导者占比(追随者占比随之相应增加)对局部区域提高开采精度,整体提升算法收敛精度。

此外,在追随者随机运动和领导者位置更新过程中,收敛因子 A 和 B 的作用是均衡两个种群类别对搜索与开发的比例,决定白骨顶鸡是扩大搜索范围还是收缩包围圈捕食。但从 COOT 的模型可知,两个参数仅是根据迭代次数进行简单的线性递减,这显然不符合种群的捕食规律。对于收敛因子 A 而言,值越大,表明随机个体的影响力越大,前次迭代的自身影响力越小,此时利于搜索。对于收敛因子 B 而言,值越大,前次迭代的自身影响力越大,全局最优解的影响力越小,此时利于开发。

1.2 模糊逻辑 COOT(FCOOT)

1.2.1 模糊逻辑领导者种群占比调整机制

为了按照算法迭代的进程动态设置白骨顶鸡领导者的种群占比,本文引入一种基于模糊逻辑的领导者种群占比调整机制,建立一种模糊推理系统对领导者在整个种群中的占比进行非线性自适应更新。

模糊推理系统是以模糊逻辑为计算工具[20],能够高效实现多输入变量与单输出变量间的复杂非线性映射关系,主要由模糊化、模糊规则库、模糊推理和去模糊化 4 个模块构成,其结构如图 1 所示。

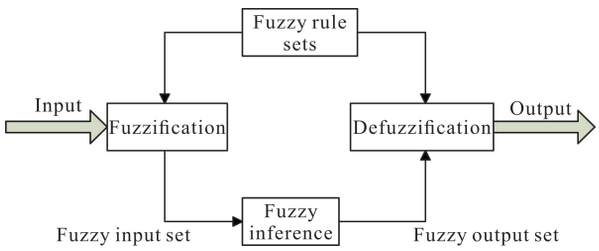


图 1 模糊推理系统

Fig. 1 Fuzzy inference system

设置模糊推理系统的输入变量为迭代进程和种群多样性。迭代进程变量 $iter$ 定义为

$$iter=\frac{t}{T_{\max}},\tag{10}$$

即迭代进程变量为算法当前迭代次数和最大迭代次数的比值。

种群多样性变量 div 定义为

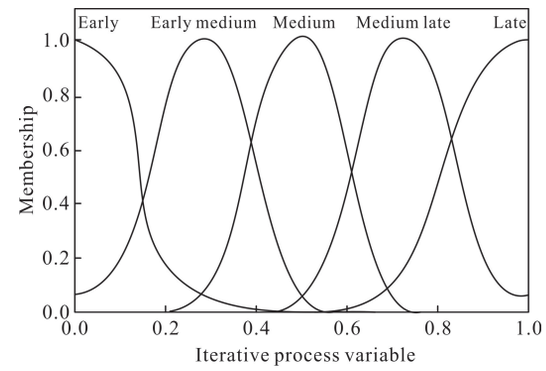
$$div=\frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{j=1}^{D}x_{i,j}(t)x_{\mathrm{best},j}(t)}{\sqrt{\sum_{j=1}^{D}x_{i,j}^2(t)}\sqrt{\sum_{j=1}^{D}x_{\mathrm{best},j}^2(t)}},\tag{11}$$

式中,N 为种群个体总数,D 为位置维度,$x_{i,j}(t)$为 t 迭代时个体 i 维度 j 的位置,$x_{\mathrm{best},j}(t)$为此时最优解的 j 维度位置。种群多样性变量可视为当前白骨顶鸡个体与最优个体间的夹角余弦值,可反映个体间的相似性。种群多样性变量越大,反映相似性越强,表明种群多样性较低;反之亦然。

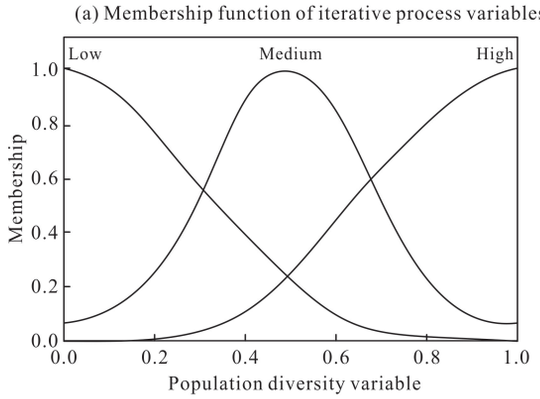
将迭代进程变量和种群多样性变量作为模糊推理系统的两个输入,并将种群多样性变量通过归一化处理将其值落入[0,1]之间,设计如图 2 所示的两个隶属度函数。迭代进程变量由 5 个隶属度函数划分为早期(Early)、早中期(Early medium)、中期(Medium)、中晚期(Medium late)和晚期(Late)。而种群多样性变量由 3 个隶属度函数划分为低多样性(Low)、中多样性(Medium)和高多样性(High)。

模糊推理系统的输出变量领导者种群占比由隶属度函数划分为较小(Very small)、小(Small)、中等(Medium)、大(Big)和较大(Very big),5 个隶属度函数分别用于表示输出量领导者种群占比的大小,如图 3 所示。

结合两个系统输入变量对领导者种群占比进行选择,还需设计相应的 IF-THEN 模糊规则。表 1 为该模糊推理系统的模糊规则,以第 2 行第 2 列所示规则为例,具体模糊规则可解释为“IF *iter* is Early medium and *div* is Medium , THEN *leader* is Medi -



(a) Membership function of iterative process variables



(b) Membership function of the population diversity variables

图 2　输入量隶属度函数

Fig. 2　Membership functions of input variables

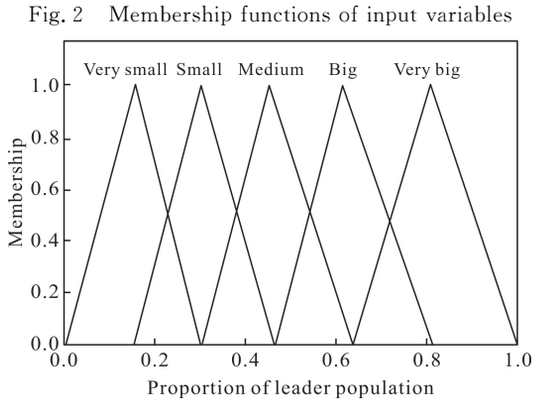


图 3　输出量隶属度函数

Fig. 3　Membership function of output variable

um”。在设计模糊规则过程中,在迭代早期,应尽可能扩大搜索范围,赋予较大占比的领导者种群规模,增强领导者的引领作用,以实现全局搜索。随着迭代的进行,算法应逐步转入小范围局部开发,此时应减小领导者规模,以较大追随者占比提高局部开发的精度。同时,迭代过程中若种群多样性逐步缺失,种群趋于早熟收敛,也应在局部开发过程中提高领导者占比,以丰富种群多样性;而在种群多样性较好时,则可以相应减小领导者的种群占比。

表 1　IF-THEN 模糊规则

Table 1　IF-THEN fuzzy rules

迭代进程 Iterative process	种群多样性 Population diversity		
	低 Low	中 Medium	高 High
Early	Very big	Very big	Big
Early medium	Very big	Medium	Small
Medium	Big	Small	Big
Medium late	Medium	Small	Very small
Late	Medium	Small	Very small

结合 IF-THEN 模糊规则,经过去模糊化即可得到模糊推理系统的输出领导者占比,如图 4 所示。在两个输入变量迭代进程和种群多样性的作用下,领导者的种群占比可动态调整,使得算法在迭代前后期以更强的搜索和开发性能搜索最优解。

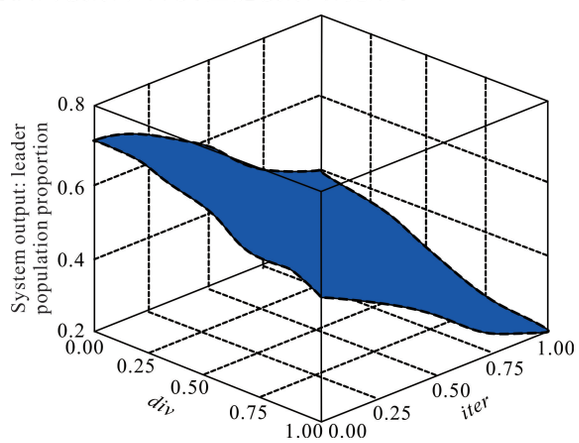


图 4　输出变量领导者占比的变化趋势

Fig. 4　Change trend of leader population proportion in output variable

1.2.2　模糊逻辑收敛因子调整机制

为了在不同的迭代阶段对两个收敛因子进行非线性调节,引入模糊逻辑使收敛因子能够根据算法的迭代进程和个体适应度误差情况进行自适应调整。

为了计算收敛因子的取值,设置模糊推理系统的输入变量为迭代进程和适应度误差。引入适应度误差作为模糊系统输入因子之一的原因在于,迭代过程中,若个体与最优解适应度相差较大,则应相应优化控制种群全局搜索与局部开发的比例,以此控制个体向最优解的递进程度。迭代进程计算方式同公式(10)。适应度误差变量 err 定义为

$$err=\frac{1}{N}\sum_{i=1}^{N}(fit(X_i)-fit_{\min}),\tag{12}$$

式中,N 为种群个体总数,$fit(X_i)$为个体 i 的适应度,$fit_{\min}$ 为当前最优适应度。可见,适应度误差变

量可用于度量每个个体适应度与最优个体间的差异性。

迭代进程的隶属度函数同图 2(a)。适应度误差的隶属度函数如图 5 所示,适应度误差由 3 个隶属度函数划分为低误差(Low)、中等误差(Medium)和高误差(High)。

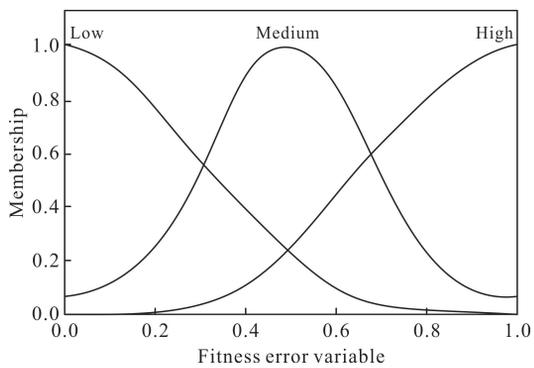


图 5 适应度误差隶属度函数

Fig. 5 Membership function of fitness error variable

由于收敛因子 A、B 的原始取值范围分别为[0,1]和[0,2],故作为系统输出量时可以由隶属度函数划分为较小(Very small)、小(Small)、中等(Medium)、大(Big)和较大(Very big)5 个阶段,如图 6 所示。

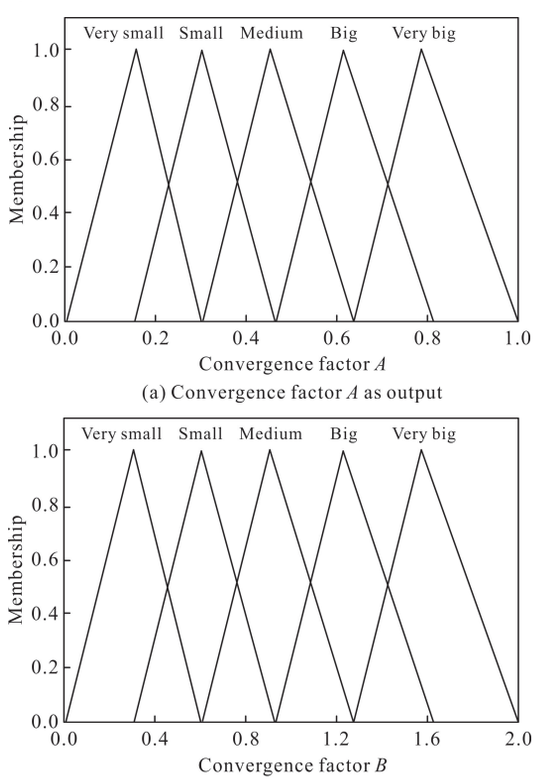


图 6 收敛因子的隶属度函数

Fig. 6 Membership functions of convergence factors

结合两个系统输入变量对两个收敛因子 A 和 B 进行选择,设计如表 2 所示的 IF-THEN 模糊规则。

表 2 收敛因子 IF-THEN 模糊规则

Table 2 IF-THEN fuzzy rules of convergence factor

迭代进程 Iterative process	适应度误差 Fitness error		
	低 Low	中 Medium	高 High
Early	Very big	Very big	Big
Early medium	Big	Medium	Medium
Medium	Medium	Medium	Small
Medium late	Medium	Small	Very small
Late	Very small	Small	Very small

结合表 2 所示的模糊规则,经过去模糊化即可得到模糊推理系统的输出收敛因子,如图 7 所示。在两个输入变量迭代进程和适应度误差的作用下,收敛因子可动态调整,实现收敛性的自适应平衡。

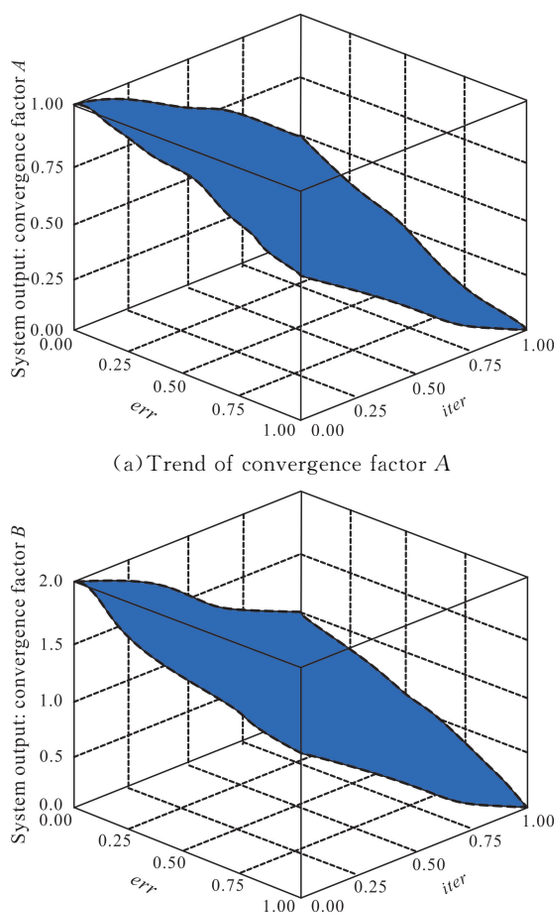


图 7 输出变量收敛因子的变化趋势

Fig. 7 Change trends of convergence factors in output variable

1.2.3　FCOOT 的有效性验证

为了验证模型逻辑对 COOT 改进的有效性，利用一个单峰函数和两个高维多峰基准函数作为测试函数对 FCOOT 进行性能验证。单峰函数名称为 Rosenbrock，表达式为

$$f_1(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i)^2]+(x_i-1)^2, \tag{13}$$

式中，x_i、x_{i+1} 为两个相邻的变量，n 为搜索空间的维度，搜索范围为[−30,30]，理论最优解为 0。

一个高维多峰基准函数名称为 Schwefel 2.26，表达式为

$$f_2(x)=418.983n-\sum_{i=1}^{n}x_i\sin(\sqrt{|x_i|}), \tag{14}$$

式中，n 为搜索空间的维度。该函数搜索范围为[−500,500]，理论最优解为 0。

另一个高维多峰基准函数名称为 Schwefel，表达式为

$$f_3(x)=-\sum_{i=1}^{n}(x_i\sin\sqrt{|x_i|}), \tag{15}$$

式中，n 为搜索空间的维度。该函数搜索范围为[−500,500]，理论最优解为 0。

图 8 是 Schwefel 2.26 函数的三维空间图像。可见，该函数具有较多的局部极值点，极其考验算法的搜索精度以及能否有效跳离局部极值点的能力。

设置种群规模 $N=30$，最大迭代次数 $T_{max}=400$，领导者种群的初始占比设置为 72%，两个收敛因子的初始值分别设置为 $A=1$、$B=2$。利用 FCOOT 搜索测试函数 $f(x)$ 的最优解，并与 COOT 进行比较，验证模糊逻辑机制改进 COOT 性能的可行性。同时，将仅融合模糊逻辑领导者种群占比调整机制的 COOT 命名为 FCOOT1，仅融合模糊逻辑收敛因子调整机制的 COOT 算法命名为 FCOOT2，引入这两个子算法进行消融实验对比，验证单策略改进的性能。表 3 是 COOT、FCOOT1、FCOOT2 和 FCOOT 在 Rosenbrock 函数、Schwefel 2.26 函数和 Schwefel 函数上的最优解、平均值以及标准差上的对比结果，其中平均值为算法独立运行 10 次后的均值结果。这 4 种算法在 Schwefel 2.26 函数上的收敛性对比结果如图 9 所示。可见，FCOOT1 和 FCOOT 均能找到函数最优解，FCOOT2 虽然没有找到最优解，但其搜索精度比 COOT 高出若干数量级。相比而言，模糊逻辑领导者种群占比调整机制比模糊逻辑收敛因子调整机制更能提高算法的搜索精度，前者比后者在目标函数平均值上可以提高 15 个数量级。从收敛性看，FCOOT 能够最快提高搜索精度，接近最优解，COOT 则在迭代过程中处于比较平缓的状态，说明其种群个体进化比较慢，且最后收敛于局部最优。

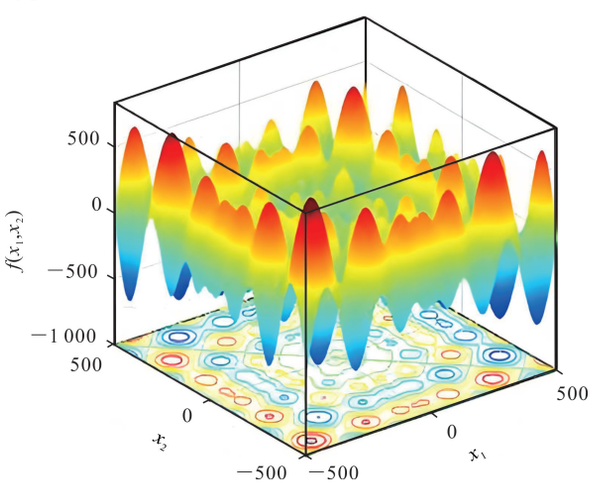


图 8　Schwefel 2.26 函数的三维空间图像

Fig. 8　Function 3D image of Schwefel 2.26

表 3　不同算法的函数测试结果

Table 3　Function test results of different test algorithms

测试函数 Test function	测试算法 Test algorithm	最优解 Best	平均值 Mean	标准差 Standard deviation
Rosenbrock	COOT	4.12E−01	4.08E−01	3.19E−01
	FCOOT1	3.96E−03	4.39E−03	6.63E−04
	FCOOT2	1.36E−03	1.95E−03	3.81E−03
	FCOOT	2.36E−09	2.70E−09	5.39E−08
Schwefel 2.26	COOT	7.23E−55	1.84E−32	6.01E−29
	FCOOT1	0.00E+00	5.38E−100	5.32E−76
	FCOOT2	5.38E−103	8.49E−85	4.57E−34
	FCOOT	0.00E+00	3.43E−111	1.01E−98

续表

Continued table

测试函数 Test function	测试算法 Test algorithm	最优解 Best	平均值 Mean	标准差 Standard deviation
Schwefel	COOT	2.48E−12	4.77E−12	3.05E−12
	FCOOT1	4.03E−19	5.57E−19	3.28E−20
	FCOOT2	3.92E−23	4.69E−23	8.01E−22
	FCOOT	0.00E+00	0.00E+00	1.03E−36

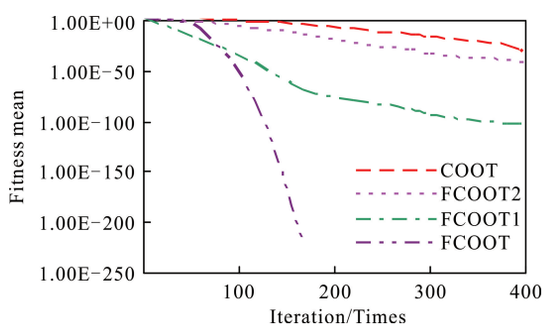


图 9　算法收敛曲线

Fig. 9　Algorithm convergence curve

2　FCOOTKHM

2.1　KHM 聚类算法

不同于 KM 聚类算法,KHM 聚类算法以调和平均值作为距离度量方式进行簇群划分,具有比 KM 聚类算法更快的收敛速度。同时,KHM 聚类算法不存在对初始聚类质心敏感的不足,但依然存在易陷入局部最优的缺陷。KHM 聚类目标函数可定义为

$$\mathrm{KHM}(Z,C)=\sum_{i=1}^{n}\frac{k}{\sum_{j=1}^{k}\frac{1}{\|z_i-c_j\|^{p}}},\quad(16)$$

式中,n 为数据对象总数,k 为质心数量,p 为指数,z_i 为数据集 $Z=(z_1,z_2,\cdots,z_n)$ 内的一个数据矢量 i,$C=(c_1,c_2,\cdots,c_k)$ 为聚类质心集,c_j 为聚类质心 j。KHM 聚类过程为

步骤 1:在 D 维搜索空间内随机选择 k 个聚类质心。

步骤 2:根据公式(16)计算目标函数值。

步骤 3:对于每个数据点对象 x_i,计算 x_i 在每个聚类质心内的隶属度 $m(c_j\backslash x_i)$,计算公式为

$$m(c_j\backslash x_i)=\frac{\|x_i-c_j\|^{(-p-2)}}{\sum_{j=1}^{k}\|x_i-c_j\|^{(-p-2)}},\quad(17)$$

同时,计算数据对象 x_i 在聚类质心内的权重值 $w(x_i)$

$$w(x_i)=\frac{\sum_{j=1}^{k}\|x_i-c_j\|^{(-p-2)}}{(\sum_{j=1}^{k}\|x_i-c_j\|^{-p})^2}。\quad(18)$$

步骤 4:对于每个聚类质心,根据其隶属度和权重函数重新计算数据点对象 x_i 的聚类质心为

$$c_j=\frac{\sum_{i=1}^{n}m(x_j/x_i)w(x_i)x_i}{\sum_{i=1}^{n}m(x_j/x_i)w(x_i)}。\quad(19)$$

步骤 5:重复步骤 2 至步骤 4 直到聚类质心无变化或到达最大迭代次数为止。

步骤 6:将数据矢量 x_i 重新分配至具有最大隶属度的聚类质心 j。

2.2　FCOOTKHM 设计

本文设计一种 FCOOT 结合 KHM 聚类算法的聚类算法 FCOOTKHM。该算法融合了 KHM 聚类算法的局部开发能力和 FCOOT 强大的全局搜索能力,将 KHM 聚类算法的初始聚类结果输入 FCOOT,作为算法的初始种群结构,并通过白骨顶鸡觅食的迭代机制,结合模糊逻辑实现全局寻优,有效避免局部最优,防止聚类解出现早熟收敛状态。构建算法的具体步骤如下。

步骤 1:初始化算法参数,包括种群规模 N、KHM 聚类最大迭代次数 $KT_{\max}$、FCOOT 最大迭代次数 $T_{\max}$。

步骤 2:随机选择 K 个聚类质心,执行 KHM 聚类迭代过程,生成初始聚类解。

步骤 3:将步骤 2 生成的初始聚类解选为聚类质心作为 FCOOT 的初始种群结构,即将聚类质心编码为白骨顶鸡个体位置信息。

步骤 4:利用模糊逻辑机制决定领导者种群占比,并随机选择领导者个体,利用 KHM 聚类目标函数 KHM(X,C)计算个体适应度,确定全局最优解 gBest。

步骤 5:确定种群个体角色,对于领导者个体,利用模糊逻辑更新收敛因子 B,根据公式(8)更新领导

者位置。

步骤 6:对于跟随者个体,利用模糊逻辑更新收敛因子 A,若跟随者进行主动更新,生成一个(0,1)间随机量 $rand$,若 $rand<0.5$,则根据公式(3)更新位置;反之,则根据公式(5)更新位置。被动更新时则根据公式(6)确定其跟随的领导者个体,再根据公式(7)更新位置。

步骤 7:计算种群个体适应度,并择优保留。具体地,对于适应度更优的跟随者,则互换其领导者与跟随者角色。

步骤 8:判断是否达到 FCOOT 最大迭代次数,如未达到,则返回步骤 4 执行;否则,输出此时全局最优解,算法终止。

3 实验与结果分析

3.1 实验数据

利用 UCI 数据库中 7 个典型数据集进行算法实验测试,表 4 给出了每个数据集的规模、类别及其特征属性。7 个数据集同时涵盖有中小维度、中小样本量和较大维度及大容量样本集,利于全面测试算法处理不同规模不同特征维度数据集的性能。以中小维度小样本量的 Iris(鸢尾花)数据集为例,该数据集包含 3 个鸢尾花品种:Setosa、Versicolour 和 Virginica,每个种类鸢尾花有 50 个样本数据,总共有 150 个对象实例,数据集中有花萼长、花萼宽、花瓣长和花瓣宽 4 个特征。实验运行环境为 Windows 10,软件为 Pycharm 2022。算法参数中,种群规模 $N=30$,KHM 聚类最大迭代次数 $KT_{\max}=100$,FCOOT 最大迭代次数 $T_{\max}=400$,领导者种群初始占比设置为 72%,两个收敛因子的初始值分别设置为 $A=1$、$B=2$。

表 4 测试数据集

Table 4 Test datasets

数据集 Dataset	分类数量 Number of classes	特征数量 Number of characteristics	数据量 Data volume	各分类样本尺寸 Sample size per class
Iris	3	4	150	50,50,50
Glass	6	9	214	70,17,76,13,9,29
Cancer	2	9	683	444,239
CMC	3	9	1 473	629,334,510
Wine	3	13	178	59,71,48
Synthetic1	3	2	300	100,100,100
Synthetic2	3	3	300	100,100,100

3.2 评估指标

选择 KHM 聚类算法(以下简称为“KHM”)[4]、引力搜索 K 均值(Gravitational Search Algorithm K-Means,GSAKM)算法[7]、粒子群优化 K 调和均值(Particle Swarm Optimization K-Harmonic Means,PSOKHM)算法[10]以及 COOTKHM[17]与 FCOOTKHM 进行实验的纵横向对比。选择 KHM 的目标函数 KHM(X,C)作为聚类指标用于度量聚类质量,该函数相当于数据对象与质心的距离函数,两者距离越小,表明聚类质量越高。引入准确率(Accuracy)、精确度(Precision)、召回率(Recall)、F 度量(F-measure)以及 Kappa 系数进行具体量化比较。同时可通过计算迭代过程中聚类解的适度值描述算法的收敛情况,来评估聚类算法的收敛速度及收敛精度。为了消除偶然性因素影响,将算法执行 20 次并取其均值结果进行比较。

准确率指标定义为

$$\text{Accuracy}=\sum_{k=1}^{K}\frac{n_k}{n}, \tag{20}$$

式中,K 为聚类数量,n_k 为正确划入聚类 k 的样本数,n 为样本总数。

精确度指标定义为

$$\text{Precision}=\frac{Q_{\text{TP}}}{Q_{\text{TP}}+Q_{\text{FP}}}, \tag{21}$$

式中,Q_{TP} 为真正类,Q_{FP} 为假正类,该指标即为实际划入聚类的样本占比。

召回率指标定义为

$$\text{Recall}=\frac{Q_{\text{TP}}}{Q_{\text{TP}}+Q_{\text{FN}}}, \tag{22}$$

式中,Q_{FN} 为假负类。

F 度量指标定义为

$$\text{F-measure}=\frac{2\text{Recall}\times\text{Precision}}{\text{Recall}+\text{Precision}}。 \tag{23}$$

Kappa 系数指标定义为

$$\text{Kappa}=\frac{(p_0-p_e)}{(1-p_e)}, \tag{24}$$

式中,p_0 为观测一致率,p_e 为期望一致率。

3.3 实验结果

表 5 是算法在所有指标上的结果,表 5 中的加粗部分表示该指标在算法中的最优解。总体看,FCOOTKHM 在多数数据集上能够得到算法的最优解,尤其对于分类比较明确的数据集,如 Wine 数据集和 Cancer 数据集,聚类的准确率、精确度和 Kappa 系数都具有较大的优势。在 Wine 数据集上,与 KHM 和 COOTKHM 相比,FCOOTKHM 的准确

率、精确度和 Kappa 系数分别提升了 13.88%、12.12%、6.49%和 8.20%、1.57%、1.98%。即使在特征维度有所增加以及样本量较大的 CMC 数据集中,FCOOTKHM 依然表现出与对比模型更强的聚类性能,这表明模糊逻辑对于改进算法的搜索能力也是有效可行的。F 度量可以用于均衡精确度和召回率两个指标,生成综合评价。FCOOTKHM 的 F 度量指标在多数情况下也是最优的,表明该算法能够在模糊逻辑机制下生成更符合数据集实际分布特征的聚类结果。

表 5　不同算法的聚类指标结果

Table 5　Cluster index results of different algorithms

数据集 Dataset	算法 Algorithm	聚类指标 Cluster index				
		准确率 Accuracy	精确度 Precision	召回率 Recall	F 度量 F-measure	Kappa 系数 Kappa coefficient
Iris	KHM	0.852 5	0.858 5	0.880 2	0.865 4	0.698 4
	GSAKM	0.860 9	0.882 6	0.685 7	0.871 1	0.703 7
	PSOKHM	0.881 2	0.881 5	0.894 6	0.883 7	0.715 3
	COOTKHM	0.918 4	0.918 2	0.901 1	0.890 8	0.793 7
	FCOOTKHM	**0.946 2**	**0.945 8**	**0.906 9**	**0.908 3**	**0.874 6**
Glass	KHM	0.689 2	0.713 5	0.792 4	0.883 6	0.342 1
	GSAKM	0.696 5	0.720 2	0.805 6	0.895 3	0.359 8
	PSOKHM	**0.796 3**	0.727 5	0.806 7	0.863 5	0.362 7
	COOTKHM	0.723 1	0.735 8	0.830 1	0.895 6	0.458 3
	FCOOTKHM	0.794 5	**0.793 7**	**0.857 6**	**0.908 7**	**0.473 6**
Cancer	KHM	0.540 9	0.557 8	0.543 8	0.576 8	0.228 4
	GSAKM	0.556 7	0.522 4	0.516 7	0.523 3	0.238 5
	PSOKHM	0.599 6	0.575 8	0.528 7	0.550 4	0.284 7
	COOTKHM	0.586 7	0.584 9	0.587 2	0.584 6	0.302 8
	FCOOTKHM	**0.697 3**	**0.608 3**	**0.608 7**	**0.593 8**	**0.324 7**
CMC	KHM	0.477 2	0.487 3	0.490 4	0.478 2	0.511 2
	GSAKM	0.494 6	0.515 4	**0.690 8**	0.490 4	0.583 2
	PSOKHM	0.508 7	0.538 5	0.509 8	0.534 2	0.598 3
	COOTKHM	0.519 2	0.575 4	0.535 7	0.564 7	0.694 8
	FCOOTKHM	**0.583 7**	**0.673 4**	0.679 2	**0.746 4**	**0.782 7**
Wine	KHM	0.573 6	0.526 5	0.472 4	0.503 9	0.632 2
	GSAKM	0.580 9	0.534 8	0.479 8	0.527 8	0.647 0
	PSOKHM	0.581 6	0.562 5	0.498 3	0.537 4	0.652 8
	COOTKHM	0.603 7	0.581 2	0.500 3	0.536 7	0.660 1
	FCOOTKHM	**0.653 2**	**0.590 3**	**0.527 1**	**0.573 6**	**0.673 2**
Synthetic1	KHM	0.285 7	0.284 7	0.290 8	0.308 2	0.573 6
	GSAKM	0.307 5	0.311 6	0.340 2	0.349 2	0.609 8
	PSOKHM	**0.415 2**	0.373 4	0.387 3	0.395 8	0.676 3
	COOTKHM	0.393 0	0.383 7	0.3847	0.438 2	0.702 9
	FCOOTKHM	0.402 3	**0.457 2**	**0.394 5**	**0.583 7**	**0.782 7**
Synthetic2	KHM	0.453 7	0.468 7	0.431 6	0.420 9	0.783 7

续表

Continued table

数据集 Dataset	算法 Algorithm	聚类指标 Cluster index				
		准确率 Accuracy	精确度 Precision	召回率 Recall	F 度量 F-measure	Kappa 系数 Kappa coefficient
	GSAKM	0.495 0	0.503 4	0.470 5	0.451 1	0.793 2
	PSOKHM	0.501 9	0.513 4	0.490 5	0.503 8	0.805 8
	COOTKHM	0.574 1	0.594 5	0.474 9	0.592 1	0.872 2
	FCOOTKHM	**0.604 8**	**0.608 2**	**0.573 8**	**0.676 3**	**0.892 7**

Note:the bolded data in the table indicate the optimal solution of the algorithm.

选取 Iris、Glass 和 CMC 这 3 个数据集,直观比较算法在搜索目标函数 KHM(X,C)最优解时的收敛性情况,结果如图 10 所示。首先,从最终收敛处得到的目标适应度看,FCOOTKHM 是所有算法中最小的,说明其生成的聚类能够使得每个数据对象均加入到最优的质心之中,从而最大化相同分类中数据的相似性。从搜索到最优解的收敛迭代次数看,FCOOTKHM 的收敛迭代次数也是最少的,说明模糊逻辑能够加快模型的搜索速度,快速地提高模型的聚类精度。利用模糊逻辑对种群领导者占比和收敛因子进行自适应调整,能够提高算法跳离局部最优的概率,加快种群个体搜索的收敛速度,有效预防算法在局部极值处的长期震荡。由此可见,本文在 KHM 中融入 FCOOT 的策略在提升算法收敛精度和收敛速度上均是有效可行的。

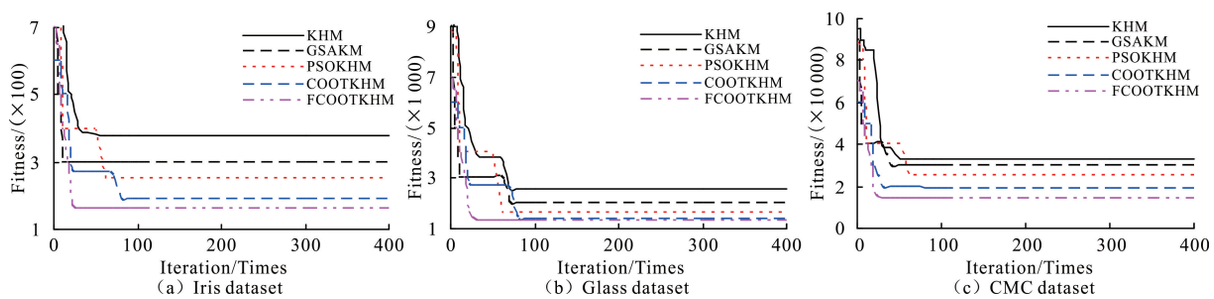


图 10 算法收敛性能

Fig. 10 Algorithm convergence performance

KHM(X,C)侧重于类内数据的聚合效果,在此进一步引入粗糙集聚类指标距离比值 h 比较算法性能。该比值为各类内对象到整个数据中心的分布距离与类内对象距离之比,定义为

$$h=\frac{\sum_{i=1}^{K}\sum_{j=1}^{|C_i|}|x_{ij}-m|^2}{\sum_{i=1}^{K}\sum_{j=1}^{|C_i|}|x_{ij}-m_i|^2}, \tag{25}$$

式中,K 为聚类数,$|C_i|$为聚类 i 对象数量,m_i 为聚类 C_i 的质心,x_{ij} 为聚类 i 的对象 j,m 为整个数据对象中心,定义为

$$m=\sum_{i=1}^{K}\sum_{j=1}^{H}x_{ij}/H, \tag{26}$$

式中,H 为数据对象总数。根据 h 的定义可知,相同聚类内数据对象分布越紧凑,距离越小;而不同聚类内数据对象越离散,则其到整个数据中心的距离越大。这表明距离比值 h 越大,算法的聚类性能越好。

图 11 是 5 种对比算法得到的距离比值结果。可见,FCOOTKHM 能够在各个数据集上得到更好的聚类效果,也更加符合数据的分布特征。

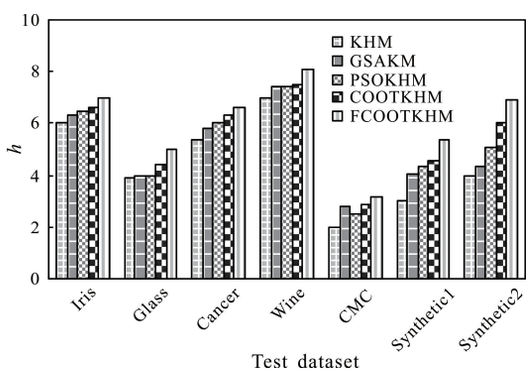


图 11 距离比值指标

Fig. 11 Distance ratio index

综合以上结论可知,在不同特征维度和样本模型数据集测试下,FCOOTKHM 无论是聚类指标还是

迭代效率方面均具有更好的性能优势,且聚类准确率更高,实现了聚类内数据高度紧凑相似,聚类间数据充分离散分布。这也说明将模糊逻辑融入 COOT 并以其优化 KHM 的思路是切实可行的,算法通过模糊推理系统对 COOT 的收敛因子和领导者种群占比进行自适应调整,以此提升算法的全局寻优能力,并结合 KHM 聚类算法搜索最优的聚类质心。

4 结论

本文提出了一种结合模糊逻辑 COOT 优化 KHM 的数据聚类算法(FCOOTKHM)。在 COOT 中引入模糊逻辑机制对算法收敛因子和领导者种群占比进行自适应调整,从而提升算法搜索和开发能力。同时,将 KHM 聚类初始解作为改进算法的初始种群组成,结合 KHM 的快速局部开发能力和 COOT 的全局勘探能力对聚类解迭代寻优,解决 KHM 易陷入局部最优的不足。数据集测试结果表明,改进算法聚类性能无论是稳定性还是全局收敛性都得到了有效提升。未来研究可集中在算法搜索精度的持续改进以及结合车联网大数据特征选择问题进一步优化聚类模型。

参考文献

[1] GUO C,TANG H,NIU B. Evolutionary state-based novel multi-objective periodic bacterial foraging optimization algorithm for data clustering [J]. Expert Systems, 2022,39(1):e12812.

[2] HOU Q,WANG G J,WANG X Z,et al. Research and application on spark clustering algorithm in campus big data analysis [J]. Journal of Computer Science Research,2020,2(1):16-20.

[3] HARTIGAN J A,WONG M A. Algorithm AS 136:a K-means clustering algorithm [J]. Journal of the Royal Statistical Society Series C (Applied Statistics),1979, 28(1):100-108.

[4] 张文宇,张茜,杨媛,等. 基于改进 GWO-CV 优化的 K-调和均值聚类算法[J]. 统计与决策,2020,36(16):9-13.

[5] IKOTUN A M,ALMUTARI M S,EZUGWU A E. K-means-based nature-inspired metaheuristic algorithms for automatic data clustering problems:recent advances and future directions [J]. Applied Sciences, 2021, 11(23):11246.

[6] 王永刚,李靖,王文慧,等. 基于和声搜索机制的特征选择与文本聚类分析[J]. 计算机工程与设计,2022, 43(2):472-478.

[7] 张雪峰,杜孝平,王晓健,等. 基于引力搜索机制的数据聚类及特征选择算法[J]. 计算机工程与设计,2021, 42(9):2536-2544.

[8] 娄奥,姚敏立,袁丁. 基于 GSA 算法改进的 K 均值聚类[J]. 计算机工程与设计,2020,41(4):1001-1005.

[9] BABU S S,JAYASUDHA K. A simplex method-based bacterial colony optimization algorithm for data clustering analysis [J]. International Journal of Pattern Recognition and Artificial Intelligence, 2022, 36 (12): 2259027.

[10] LI Y,QI J F,CHU X Q,et al. Customer segmentation using K-means clustering and the hybrid particle swarm optimization algorithm [J]. The Computer Journal,2023,66(4):941-962.

[11] 叶廷宇,叶军,王晖,等. 结合人工蜂群优化的粗糙 K-means 聚类算法[J]. 计算机科学与探索,2022,16(8): 1923-1932.

[12] 王海玲,杨俊杰. 基于改进蜂群算法及 K 均值聚类的 WSN 分簇路由算法[J]. 计算机应用与软件,2022, 39(9):178-182,232.

[13] JIN Q B,LIN N,ZHANG Y M. K-means clustering algorithm based on chaotic adaptive artificial bee colony [J]. Algorithms,2021,14(2):53.

[14] CHEN X J,ZHAO J,JIA X Z,et al. Multi-step wind speed forecast based on sample clustering and an optimized hybrid system [J]. Renewable Energy, 2021, 165:595-611.

[15] 李兆彬,叶军,周浩岩,等. 变异萤火虫优化的粗糙 K-均值聚类算法[J]. 山东大学学报(工学版),2023, 53(4):74-82.

[16] 巨金香,张福泉,黄锐. 基于烟花算法优化 k 均值聚类的教学质量评估模型[J]. 济南大学学报(自然科学版),2022,36(6):755-760.

[17] NARUEI I,KEYNIA F. A new optimization method based on COOT bird natural life model [J]. Expert Systems with Applications,2021,183:115352.

[18] MEMARZADEH G,KEYNIA F. A new optimal energy storage system model for wind power producers based on long short term memory and coot bird search algorithm [J]. Journal of Energy Storage, 2021, 44: 103401.

[19] KURAN E C,KURAN U,ER M B. Sub-image histogram equalization using coot optimization algorithm for segmentation and parameter selection [C]// Academy and Industry Research Collaboration Center (AIRCC). Computer Science & Information Technology (CS & IT). Vancouver:[s. n.],2022:33-46.

[20] VALDEZ F,CASTILLO O,PERAZA C. Fuzzy logic in dynamic parameter adaptation of harmony search optimization for benchmark functions and fuzzy controllers [J]. International Journal of Fuzzy Systems, 2020, 22(4):1198-1211.

A Data Clustering Algorithm Based on Fuzzy COOT K-Harmonic Means

DAI Luanyue[1,2], LIANG Xiaoyue[1], WANG Shuai[1], WANG Zhenpo[2**]
(1. Postdoctoral Research Station of Beijing Peony Electronic Group Co., Ltd., Beijing, 100089, China; 2. National Engineering Laboratory for Electric Vehicles, Beijing Institute of Technology, Beijing, 100081, China)

Abstract: To solve the problem that the clustering algorithm K-Harmonic Means (KHM) is easy to fall into a local optimum, a clustering algorithm Fuzzy COOT K-Harmonic Means (FCOOTKHM) combining the rapid local development capability of KHM and the global exploration capability of Coot optimization algorithm (COOT) is proposed. The initial clustering solution generated by KHM is input as the initial population structure of COOT, and then iterative optimization is carried out. To further improve the search accuracy of COOT, a fuzzy logic is designed to adaptively adjust the convergence factor and leader population proportion of COOT, which can balance the search and development capabilities of the algorithm. The harmonic mean of clustering is used to evaluate the fitness of individual populations and iteratively search for clustering results. Seven datasets of University of California Irvine (UCI) were used to validate the clustering performance of FCOOTKHM. The results showed that the improved algorithm performed better in terms of accuracy, precision, recall, F-measure, Kappa coefficient and convergence speed, which can enable more accurate data clustering.

Key words: fuzzy logic; fuzzy system; Coot optimization algorithm (COOT); K-Harmonic Means (KHM); clustering; convergence

责任编辑:梁　晓,于子涵