第58卷 第5期
2022年9月

南京大学学报(自然科学)
JOURNAL OF NANJING UNIVERSITY
(NATURAL SCIENCE)

Vol. 58, No. 5
Sept., 2022

DOI:10.13232/j.cnki.jnju.2022.05.012

基于随机跳跃蝠鲼算法优化的电影信息数据聚类

黄 鹤[1,2],李潇磊[1,2],王 珺[1*],王会峰[1],茹 锋[1,2]

(1.西安市智慧高速公路信息融合与控制重点实验室,长安大学,西安,710064;
2.长安大学电子与控制工程学院,西安,710064)

摘 要:针对传统K均值聚类(K-Means Clustering,KMC)算法在对电影信息数据聚类的过程中,初始聚类中心选取随机性较大、聚类结果不稳定且算法容易陷入局部最优、影响迭代精度等不足,提出一种基于随机跳跃式翻滚觅食蝠鲼优化的K均值联合迭代聚类算法(MRRJRFO-KMC),实现对电影信息数据的聚类.首先,提出一种均值最大最小距离积法来初始化聚类中心,改善聚类中心选取的随机性,避免随机初始化对聚类结果造成的不稳定性.其次,在迭代的过程中加入蝠鲼觅食优化算法,并对蝠鲼觅食优化算法中螺旋觅食和翻滚觅食进行改进,提出一种随机跳跃式翻滚觅食蝠鲼优化的策略,解决了蝠鲼觅食优化算法易陷入局部最优的问题.将随机跳跃式翻滚觅食蝠鲼优化算法加入KMC算法,对KMC算法迭代过程中的聚类中心进行优化,提高了聚类精度.在Iris,Aggregation,Ecoli和Seeds国际标准数据集上对MRRJRFO-KMC算法、MRFO-KMC算法、KMC算法、K-Means++算法、模糊C均值(Fuzzy C-Means,FCM)聚类算法进行比较测试,实验结果表明,MRRJRFO-KMC算法和其他算法相比,准确性和收敛速度都有所提升.在电影信息数据处理过程中,该算法能够根据所给的信息进行有效的聚类,应用价值明显.

关键词:蝠鲼觅食优化算法,K均值聚类,均值最大最小距离积法,随机跳跃式翻滚,电影信息数据

中图分类号:TP301.6 **文献标志码**:A

Movie information data clustering optimized based on random jumping manta ray algorithm

Huang He[1,2], Li Xiaolei[1,2], Wang Jun[1*], Wang Huifeng[1], Ru feng[1,2]

(1. Xi′an Key Laboratory of Intelligent Expressway Information Fusion and Control, Chang′an University, Xi′an, 710064, China; 2. School of Electronic and Control Engineering, Chang′an University, Xi′an, 710064, China)

Abstract: For the traditional K-Means Clustering (KMC) algorithm in the process of clustering movie information data, the selection of initial clustering center is relatively random, the clustering result is unstable and the algorithm is easy to fall into local optimum, affecting the iterative accuracy and other shortcomings. This paper proposes a K-Means joint iterative clustering algorithm based on Manta Ray with Random Jumping Roll Foraging Optimization algorithm (MRRJRFO-KMC) to realize the clustering of movie information data. Firstly, a mean max-min distance product method is proposed to initialize the cluster centers, which improves the randomness of cluster center selection and avoids the instability of clustering results caused by random initialization. Secondly, the Manta Ray Foraging Optimization algorithm is added in the iterative process, and spiral foraging and roll foraging in the Manta Ray Foraging Optimization algorithm are improved. A Manta Ray with Random Jumping Roll Foraging Optimization algorithm is proposed, which solves the problem that the Manta Ray Foraging

基金项目:国家重点研发计划(2021YF2501200),国家自然科学基金(52172379,52172324),陕西省重点研发计划(2021SF-483),陕西省自然科学基础研究计划(2021JM-184),西安市智慧高速公路信息融合与控制重点实验室(长安大学)开放基金(300102321502),中央高校基本科研业务费(300102240203)

收稿日期:2022-05-18

*通讯联系人,E-mail:jwang@nwu.edu.cn

Optimization algorithm is easy to fall into local optimum. The Manta Ray with Random Jumping Roll Foraging Optimization algorithm is added to KMC algorithm, and the clustering center in the KMC algorithm is optimized, which improves the clustering precision. MRRJRFO-KMC algorithm, MRFO-KMC algorithm, KMC algorithm, K-Means++ algorithm and Fuzzy C-Means (FCM) algorithm are calculated on Iris, Aggregation, Ecoli and Seeds international standard datasets. Experimental results show that the accuracy and convergence speed of MRRJRFO-KMC algorithm are improved compared with other algorithms. In the process of movie information data processing, the algorithm can effectively cluster according to the given information, and its application value is obvious.

Key words: MRFO, K-Means clustering, Mean Maximum Minimum Distance Product Method, random jump roll, movie information data

随着信息技术的不断发展,世界进入大数据时代,网上出现海量的电影资源,其信息数据也呈现爆炸式增长[1]. 如何在大量电影信息数据中通过电影和用户的相关数据分析出用户喜欢的电影,将爱好一致的用户聚集在一起,需要设计相应的聚类算法. 聚类作为统计学习的重要组成部分,是一种无监督的学习算法,主要思想是通过比较不同样本之间的特点,将相似的数据样本归类同一簇中[2]. K均值聚类(K-Means Clustering, KMC)算法是一种常见的聚类分析算法,具有收敛速度快且容易实现等特点,得到了广泛应用[3],但KMC选取初始聚类中心时随机性比较强,导致结果不稳定,使电影信息用户的聚类结果误差较大. 对此,学者们提出了一些改进思路. Capó et al[4]针对KMC算法初始化问题,提出一种高效的划分-融合思路,可在达到固定点后重新启动K-means算法,减小初始化过程中的误差. Vouros et al[5]通过对随机式算法的理论和实验比较,证明对于不太复杂的随机方法,多次执行可以得到较好的初始聚类中心. 周本金等[6]提出一种最小化误差平方和的方法,每次选择聚类中心时计算当前能够减少的误差平方和,选取最大程度减少误差的数据点,初始化聚类中心,但该方法复杂程度较高,不适合处理较大的数据集. 李丽亚和闫宏印[7]设计了加权自适应的KMC方法,通过加入权重系数,在处理大数据时能提高聚类效果,但在对权重进行分配时随机性较大,算法稳定性不够. Shen et al[8]通过剔除远离样本点的一些数据再选取初始聚类中心来实现对KMC聚类中心的优化,但在处理高维数据集时迭代速度较慢. 近年来,利用群体智能算法优化KMC的方法成为研究热点. 刘佳鸣等[9]结合灰狼算法(Grey Wolf Optimization, GWO)和KMC,提高了全局搜索能力,但聚类精度有限. 尤烽骅等[10]通过改进的人工蜂群与KMC进行交叉,优化了KMC的初始聚类中心,但蜂群的更新机制会增加聚类耗时. 蝠鲼觅食优化算法(Manta Ray Foraging Optimization algorithm, MRFO)是2020年Zhao et al[11]提出的一种新型群体智能优化算法,模拟了蝠鲼觅食的过程,收敛速度快,寻优能力强,适用于聚类算法的优化. 在此基础上,本文提出一种随机跳跃式翻滚觅食的蝠鲼优化算法(Manta Ray with Random Jumping Roll Foraging Optimization algorithm, MRRJRFO),解决了MRFO在迭代过程中易陷入局部最优的问题. 同时,利用MRRJRFO优化KMC聚类方法,在最大最小距离积法(Maximum Minimum Distance Product, MMDP)[10]的基础上提出一种均值最大最小距离积法(Mean MMDP, MMMDP)来初始化聚类中心,并利用MRRJRFO联合迭代寻找新的聚类中心,提高了聚类精度与聚类速度. 利用优化KMC聚类电影信息数据,可以提高速度,得到更好的聚类效果.

1　相关理论

1.1　KMC的不足　KMC将给定数据集中相同类别的数据划分为一类,具体步骤为:

(1)从n个数据中随机选取k个对象作为初始聚类中心;

(2)计算其余每个数据到选定的初始聚类中心的欧式距离,根据距离最小原则归类;

(3)计算各个聚类簇中数据到其对应的聚类中心的平均值,作为新的聚类中心;

(4)重复步骤(2)和步骤(3),直到聚类中心不再发生改变.

聚类中心 C_j 的计算公式为:

$$C_j=\frac{1}{C_j}\sum_{i=1}^{n}s_i,j=1,2,\cdots,p \tag{1}$$

其中,n 为聚类样本个数,p 为聚类中心个数. 聚类中心 C_j 与样本中的数据 s_i 的欧式距离计算公式为:

$$d(s_i,C_j)=\sqrt{\sum_{p=1}^{d}(s_{ip}-C_{jp})^2} \tag{2}$$

其中,d 为数据的维度.

从上述过程可以看出,KMC 的优点是简单易实现,收敛速度较快[12-13],在初始聚类中心选取得当的情况下聚类效果良好. 但 KMC 也有三方面的不足,需要改进:

(1)初始化具有很大随机性,难以保证初始聚类中心的正确选取;

(2)对离群点、噪声点敏感;

(3)更新过程易陷入局部最优.

1.2 MMMDP 方法 传统的 MMDP 可以克服初始聚类中心在选取中存在随机性的问题,避免初始点选取不当造成的聚类误差,但其在选取第一个初始聚类中心时仍然存在随机性的问题,且有可能选到离群点. 因此,本文提出一种均值最大最小距离积法(MMMDP),在原 MMDP 基础上求出的初始聚类中心平均距离作为最终的初始聚类中心,这样可以进一步减小初始聚类中心选取的随机性,从而选取出合适的初始聚类中心. MMMDP 的步骤如下:

(1)从相应的数据集 D 中随机选取其中一个数据点 Z_1 作为初始聚类中心,将 Z_1 放入新的集合 Z,同时在原来的数据集 D 中将 Z_1 删除;

(2)计算更新之后的数据集 D 中每个元素到 Z_1 的距离,选出其中最大的距离对应的元素 Z_2;

(3)将所选的元素 Z_2 放入集合 Z,并在原来的集合 D 中删除 Z_2;

(4)计算更新之后 D 中各元素到 Z 中各元素的距离,将计算后的结果存入集合 T;

(5)找出 D 中各元素对应的 T 集合中最大距离和最小距离,求其乘积,将最大乘积对应的元素 Z_3 从 D 中删除,并存入集合 Z;

(6)求集合 Z 中当前所有元素的平均距离 V_1 作为初始聚类中心,并将 V_1 放入集合 V;

(7)以此类推,若 V 中的元素小于 P 则返回步骤(3),若 V 中元素大于 P 则初始点的选取结束,输出包含 P 个初始聚类中心点的集合 V.

从上述步骤可以看出,在 KMC 算法中加入 MMMDP 后,在选取初始聚类中心时不再是任意选取,而是从待求的数据集中,通过计算每个数据点间的距离积,选取最大乘积所对应的元素,这样就克服了之前的随机性. 多次迭代后选取多个元素,求取其平均距离,作为初始聚类中心,可以进一步提升初始聚类中心选取的准确性,使得随机性降低,鲁棒性加强.

2 MRRJRFO 算法

2.1 MRFO 算法 MRFO[14]模拟蝠鲼种群不同的觅食方式进行数学建模,描述位置的更新方式,在求解空间中搜索最优解. 蝠鲼种群的位置更新方式比较独特,算法的鲁棒性与求解精度比传统群体智能算法有明显提升. 蝠鲼种群的觅食方式有三种:链式觅食、螺旋觅食以及翻滚觅食.

2.1.1 链式觅食 在链式觅食过程中,种群排列构成一条捕食链,蝠鲼个体下一个位置不仅由当前最优位置决定,还与前一个体的位置相关. 蝠鲼在链式觅食中更新方式的数学模型如下:

$$x_i^d(t+1)=\begin{cases}x_i^d(t)+r\left(x_{\text{best}}^d(t)-x_i^d(t)\right)+\\ \qquad\alpha\left(x_{\text{best}}^d-x_i^d(t)\right), & i=1\\ x_i^d(t)+r\left(x_{i-1}^d(t)-x_i^d(t)\right)+\\ \qquad\alpha\left(x_{\text{best}}^d-x_i^d(t)\right), & i=2,3,\cdots\end{cases} \tag{3}$$

$$\alpha=2r\sqrt{|\lg(r)|} \tag{4}$$

其中,$x_i^d(t)$ 表示第 t 代第 i 个蝠鲼个体在 d 维上的位置,r 为 $[0,1]$ 上的随机数,$x_{\text{best}}^d(t)$ 为第 t 代最优的蝠鲼个体在 d 维上的位置.

2.1.2 螺旋觅食 蝠鲼发现猎物后,一般都会采用一种类似螺旋的方式靠近猎物[15],因为有链式觅食方式的存在,蝠鲼个体在螺旋觅食移动的过程中还会受到前一个蝠鲼个体的影响. 蝠鲼在螺旋觅食中位置更新方式的数学模型如下:

(1)当 $t/T>rand$ 时,蝠鲼进行螺旋觅食的方程为:

$$x_i^d(t+1)=\begin{cases}x_{\text{best}}^d(t)+r\left(x_{\text{best}}^d(t)-x_i^d(t)\right)+\\ \quad\beta\left(x_{\text{best}}^d-x_i^d(t)\right), & i=1\\ x_{\text{best}}^d(t)+r\left(x_{i-1}^d(t)-x_i^d(t)\right)+\\ \quad\beta\left(x_{\text{best}}^d-x_i^d(t)\right), & i=2,3,\cdots\end{cases}\tag{5}$$

$$\beta=2\mathrm{e}^{r_1\frac{T-i+1}{T}}\sin 2\pi r_1\tag{6}$$

其中,T表示迭代总次数,t表示当前迭代次数,$rand$,r和r_1都表示$[0,1]$上的随机数.

(2)当$t/T\leqslant rand$时,蝠鲼进行螺旋觅食的方程为:

$$x_i^d(t+1)=\begin{cases}x_{\text{rand}}^d(t)+r\left(x_{\text{rand}}^d(t)-x_i^d(t)\right)+\\ \quad\beta\left(x_{\text{rand}}^d-x_i^d(t)\right), & i=1\\ x_{\text{rand}}^d(t)+r\left(x_{i-1}^d(t)-x_i^d(t)\right)+\\ \quad\beta\left(x_{\text{rand}}^d-x_i^d(t)\right), & i=2,3,\cdots\end{cases}\tag{7}$$

$$x_{\text{rand}}^d=Lb^d+r\left(Ub^d-Lb^d\right)\tag{8}$$

其中,$x_{\text{rand}}^d(t)$为第t代d维的随机位置,Ub^d和Lb^d为变量的上下界.

2.1.3　翻滚觅食　蝠鲼种群在进行翻滚觅食的过程中,蝠鲼的每个个体会把当前最优解的位置当作支点,翻滚至与其当前位置成镜像关系的另一侧.其数学表达如下:

$$x_i^d(t+1)=x_i^d(t)+S\left(r_2x_{\text{best}}^d-r_3x_i^d(t)\right)\quad i=1,2,\cdots,N\tag{9}$$

$$S=2\tag{10}$$

其中,r_2和r_3是$[0,1]$上的随机数,S为翻滚因子,N为个体数量.

2.1.4　算法流程　MRFO算法流程如下:

步骤1. 设定相关参数,初始化种群;

步骤2. 计算种群适应度值;

步骤3. 判断是否$rand<0.5$,如果成立则蝠鲼进行螺旋觅食,如果不成立则进行链式觅食;

步骤4. 计算种群适应度值,更新最优位置;

步骤5. 进行翻滚觅食,更新蝠鲼的位置;

步骤6. 计算种群适应度值,更新最优位置;

步骤7. 判断是否满足终止条件,满足则输出最优值,不满足则返回执行步骤2至步骤6.

2.2　MRRJRFO算法　MRFO算法在迭代的过程中易陷入局部最优,搜索精度有限,因此,为了进一步提高搜索精度,跳出局部最优,本文提出MRRJRFO算法,更容易寻找全局最优解.

2.2.1　随机跳跃式翻滚　MRFO算法中,翻滚觅食是一个比较频繁的动作,可以提高蝠鲼的觅食效率.其数学模型中存在一个翻滚因子S决定翻滚的距离,S一般取经验值2,但容易使算法在迭代搜索过程中陷入局部最优.因此,本文提出一种新的随机跳跃式翻滚方式,根据每次迭代后种群的适应度不同,在翻滚因子中增加一个变化的翻滚系数ρ,随着适应度变化而变化,可以理解为蝠鲼在觅食的过程中随着巨大的胸鳍扇动进行跳跃式翻滚.加入ρ后蝠鲼种群的变化方式变得更灵活,有利于跳出局部最优和寻求全局最优.翻滚系数的更新方式如式(11)所示:

$$\rho=\frac{5}{0.25+\mathrm{e}^{\left(\frac{fitness(i)+fitness(1)}{2fitness(i)}\right)}}\tag{11}$$

其中,$fitness(i)$为第i次迭代时蝠鲼的适应度,在S中加入ρ后的变化如图1所示.

由图可见,翻滚因子S随着迭代不断变化.每次迭代中蝠鲼的适应度不同,翻滚系数ρ随着迭代而随机变化,所以加入翻滚系数ρ后S也在不断变化,这样可以使蝠鲼在下一次迭代时位置的随机性增强,搜索范围扩大.随着迭代的进行,种群逐渐找到最优解,适应度变化幅度减小,S也趋于稳定.加入ρ后,翻滚觅食的更新方式变为:

$$x_i^d(t+1)=x_i^d(t)+\rho S\left(r_2x_{\text{best}}^d-r_3x_i^d(t)\right)\quad i=1,2,\cdots,N\tag{12}$$

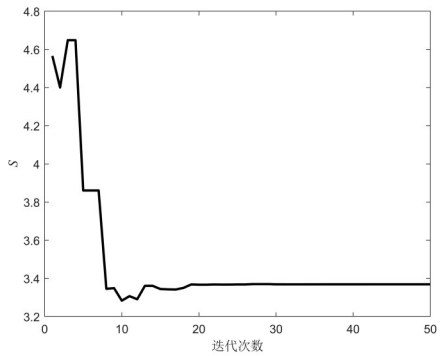

图1　翻滚因子S的变化曲线

Fig. 1　The change curve of rolling factor S

2.2.2　融合莱维飞行-高斯变异策略　在蝠鲼的螺旋觅食中引入莱维飞行策略,可以增加种群位置的多样性,扩大搜索范围.随机游走按功能设计了两种方式:长距离游走可以扩大搜索范围,跳

出局部最优;短距离搜索可以加快收敛速度. 翻滚觅食后,对当前蝠鲼的最优位置进行高斯变异,可以进一步提升蝠鲼的全局搜索能力. 莱维飞行的具体描述如下:

$$L(\lambda)=\frac{\mu}{|\nu|^{\frac{1}{\delta}}} \tag{13}$$

其中,$\delta=1.5$,μ和ν服从正态分布:

$$\mu\sim N\left(0,\sigma_\mu^2\right) \tag{14}$$

$$\nu\sim N\left(0,\sigma_\nu^2\right) \tag{15}$$

$$\sigma_\mu=\left\{\frac{\Gamma(1+\delta)\sin\frac{\pi\delta}{2}}{\Gamma\left(\frac{1+\delta}{2}\right)\delta\times 2^{\frac{\delta-1}{2}}}\right\} \tag{16}$$

其中,$\Gamma(*)$为伽马函数,$\sigma_\nu=1$. 增加莱维飞行后,式(5)和式(7)的更新方式分别变为:

$$x_i^d(t+1)=\begin{cases}x_{\text{best}}^d(t)+r\left(x_{\text{best}}^d(t)-x_i^d(t)\right)+\\ \quad\beta\left(x_{\text{best}}^d-x_i^d(t)\right)\times L(\lambda), & i=1\\ x_{\text{best}}^d(t)+r\left(x_{i-1}^d(t)-x_i^d(t)\right)+\\ \quad+\beta\left(x_{\text{best}}^d-x_i^d(t)\right)\times L(\lambda), & i=2,3,\cdots\end{cases} \tag{17}$$

$$x_i^d(t+1)=\begin{cases}x_{\text{rand}}^d(t)+r\left(x_{\text{rand}}^d(t)-x_i^d(t)\right)+\\ \quad\beta\left(x_{\text{rand}}^d-x_i^d(t)\right)\times L(\lambda), & i=1\\ x_{\text{rand}}^d(t)+r\left(x_{i-1}^d(t)-x_i^d(t)\right)+\\ \quad\beta\left(x_{\text{rand}}^d-x_i^d(t)\right)\times L(\lambda), & i=2,3,\cdots\end{cases} \tag{18}$$

高斯分布函数的表达式如式(19)所示,高斯变异可以提高算法的寻优能力和所优化算法的精度. 对蝠鲼位置进行高斯变异的方式如式(20)所示:

$$f(x_0)=\frac{1}{\sqrt{2\pi}}\mathrm{e}^{\left(-\frac{x_0^2}{2}\right)} \tag{19}$$

$$x_{\text{newbest}}=x_{\text{best}}+Gaussian(x_0)\times x_{\text{best}} \tag{20}$$

其中,x_{newbest}为更新后蝠鲼的位置,x_{best}为更新前蝠鲼的位置,高斯函数中x_0取0到1的随机数.

2.2.3 MRRJRFO算法流程 MRRJRFO的算法流程如图2所示.

具体流程如下:

步骤1. 随机生成蝠鲼种群位置,计算种群适应度;

步骤2. 判断$rand<0.5$,成立则执行螺旋觅食,同时在螺旋觅食过程中加入莱维飞行机制,不成立则执行链式觅食;

步骤3. 计算适应度值,更新最优位置;

步骤4. 根据每次适应度的不同加入翻滚系数ρ,进行随机跳跃式翻滚觅食;

步骤5. 对当前位置进行高斯变异;

步骤6. 计算适应度值,更新最优位置;

步骤7. 判断是否满足结束条件,满足则输出最优值,否则重复执行步骤2至步骤7.

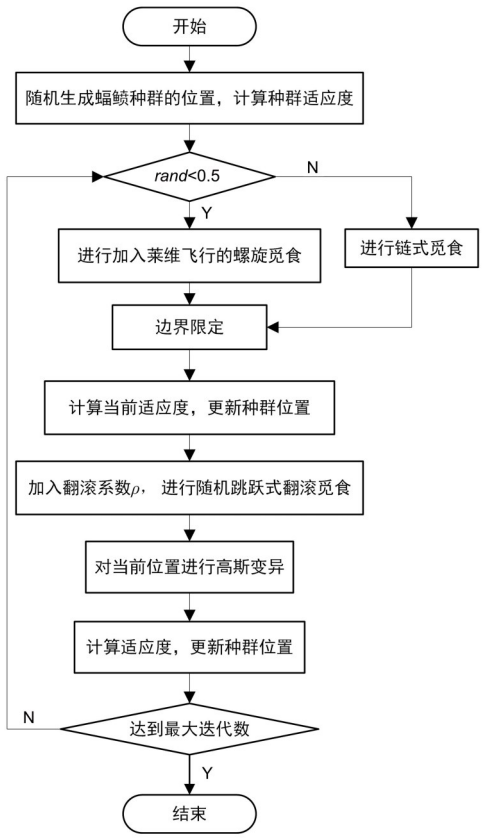

图2 MRRJRFO算法的流程图

Fig. 2 The flowchart of the MRRJRFO algorithm

2.2.4 适应度函数的选择 为了更好地引导MRRJRFO种群的进化方向,更好地优化聚类中心,提升聚类效果,需要引入适应度函数. 本文采用如下所示的适应度函数:

$$F_i=\frac{L_i}{N_i},i=1,2,\cdots,k \tag{21}$$

$$L_i=\sum_{s_i\in c_i}d(s_i,c_i) \tag{22}$$

其中,L_i表示第i类种群中的对象到该类聚类中心的距离之和,N_i表示第i类种群的数量. 从式(21)

可以看出,当前聚类结果的适应度不仅与各类中的点到该类聚类中心的距离之和有关,还和该类中点总数有关.所用的适应度函数可以反映该类中所有样本点到聚类中心的平均欧式距离,能较好地评价聚类效果,引导MRRJRFO种群的进化方向,进一步优化结果.

3　联合迭代机制

KMC通过每次迭代求各个簇均值来更新聚类中心,容易导致陷入局部最优,优化精度低.因此,本文结合MRRJRFO算法和KMC的特点,提出一种随机跳跃式翻滚觅食蝠鲼优化算法和K均值聚类算法联合迭代的聚类算法(MRRJRFO-KMC).首先,将数据的分类信息引入MRRJRFO算法中更新聚类中心,并通过适应度的计算来判断聚类中心选取是否合适,因此需要通过式(21)计算适应度.该适应度函数包括各类中的对象到该类聚类中心的距离之和以及各类包含的数量,可以有效地评价数据聚类的效果.确定每个样本的类别后,计算新聚类中心的适应度,若适应度优于上次聚类中心的适应度,则用新的聚类中心取代历史聚类中心,直到达到最大迭代次数.这样在KMC算法中融入MRRJRFO算法,可以实现联合迭代,寻求最优解.MRRJRFO-KMC的算法流程如图3所示,算法的具体步骤如下:

(1)输入标准数据集D,根据类别个数确定该数据集中的类的个数k.

(2)利用MMMDP初始化每一类的聚类中心.

(3)计算数据集D中每个点到聚类中心的欧式距离,将最小欧式距离对应的点归类到相应的聚类中心.

(4)在当前归并的每个类别中通过MRRJRFO算法进行寻优操作,以得到新的聚类中心.

(5)计算新的聚类中心的适应度,若适应度优于上次聚类中心的适应度,则用新的聚类中心取代历史聚类中心.

(6)判断迭代数是否达到了规定的次数,若达到则结束循环,若未达到则返回执行步骤(3),直至达到最大迭代次数.

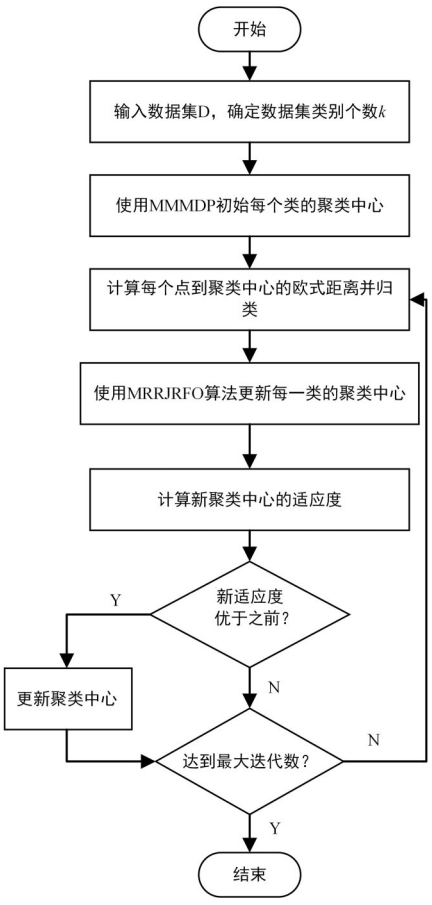

图3　MRRJRFO-KMC算法流程图

Fig. 3　The flowchart of the MRRJRFO-KMC algorithm

4　实验结果分析与应用

硬件实验平台采用Inter Core i5-7300HQ处理器、内存8 G的计算机,操作系统为Windows 10,编译软件为Matlab R2018b.

4.1　MRRJRFO算法性能测试　为了验证MRRJRFO的优化性能,与MRFO[16],GWO[17]和MFO[18]算法在不同的单峰值和多峰值测试函数上进行测试比较.单峰值函数测试算法的收敛精度,多峰值函数测试算法寻找全局最优的能力,测试迭代800次.由于篇幅限制,本文分别采用三个单峰值函数和三个多峰值函数进行测试:

(1)Sphere函数

$$f(x)=\sum_{i=1}^{n}x_i^2 \tag{23}$$

Sphere函数范围为$[-100,100]$,维度为30.

(2)Schwefel 1.2函数

$$f(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i}x_j\right)^2 \tag{24)A}$$

Schwefel 1.2 函数范围为$[-100,100]$,维度为 30.

(3)Schwefel 2.21 函数

$$f(x)=\max_i\{|x_i|,1\leqslant i\leqslant n\} \tag{25}$$

Schwefel 2.21 函数范围为$[-100,100]$,维度为 30.

(4)Rastrigin 函数

$$f(x)=\sum_{i=1}^{n}\left(x_i^2-10\cos(2\pi x_i)+10\right)^2 \tag{26}$$

Rastrigin 函数范围为$[-5.12,5.12]$,维度为 30.

(5)Ackley 函数

$$f(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right)+20+\mathrm{e} \tag{27}$$

Ackley 函数范围为$[-32,32]$,维度为 30.

(6)Griewank 函数

$$f(x)=\frac{1}{4000}\sum_{i=1}^{n}(x_i-100)^2-\prod_{i=1}^{n}\cos\left(\frac{x_i-100}{\sqrt{i}}\right)+1 \tag{28}$$

Griewank 函数范围为$[-600,600]$,维度为 30.

图 4 为各算法在不同测试函数上的适应度曲线,表 1 为经过 30 次测试后的适应度最优值、均值和标准差,表中黑体字表示最优的结果.

由图 4 可以看出,本文 MRRJRFO 算法和 MRFO 相比,在单峰值测试函数和多峰值测试函数上的收敛精度都有提高,收敛速度也更快;和 GWO 和 MFO 相比,MRRJRFO 在 Schwefel 1.2 和 Schwefel 2.21 函数上的收敛精度明显提高;在其他的几种测试函数上,MRRJRFO 的收敛速度同样比 GWO 和 MFO 更快.

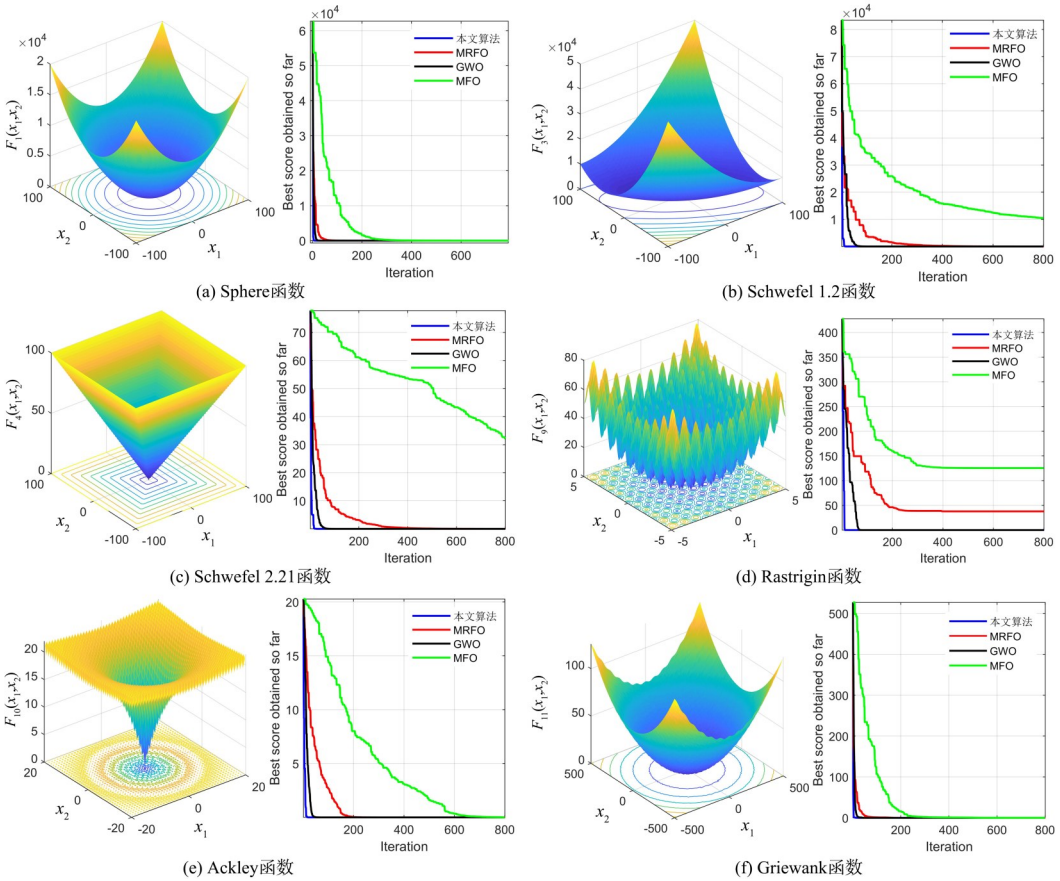

图 4 测试函数适应度曲线

Fig. 4 The fitness curves of test functions

表1　不同算法在六种测试函数上的实验结果

Table 1　Experimental results of different algorithms on six test functions

测试函数	算法	最优适应度	均值	标准差
Sphere函数	MRFO	6.44×10^{-25}	4.14×10^{-28}	5.22×10^{-28}
	GWO	1.01×10^{-69}	2.68×10^{-67}	4.18×10^{-67}
	MFO	1.57×10^{-3}	1.52×10^{-3}	1.84×10^{-3}
	本文算法	$\mathbf{9.03\times10^{-216}}$	$\mathbf{8.86\times10^{-186}}$	0
Schwefel 1.2函数	MRFO	0.162	0.74	0.43
	GWO	8.51×10^{-26}	4.89×10^{-22}	4.18×10^{-22}
	MFO	9.64×10^{2}	1.06×10^{4}	1.04×10^{4}
	本文算法	$\mathbf{1.73\times10^{-176}}$	$\mathbf{3.60\times10^{-173}}$	0
Schwefel 2.21函数	MRFO	3.0×10^{-3}	5.63×10^{-3}	2.44×10^{-3}
	GWO	5.56×10^{-18}	1.59×10^{-17}	7.66×10^{-18}
	MFO	19.48	29.91	10.46
	本文算法	$\mathbf{1.10\times10^{-108}}$	$\mathbf{4.49\times10^{-93}}$	$\mathbf{6.35\times10^{-93}}$
Rastrigin函数	MRFO	24.87	29.52	3.66
	GWO	0	0	0
	MFO	1.76×10^{2}	1.23×10^{2}	37.98
	本文算法	0	0	0
Ackley函数	MRFO	1.51×10^{-14}	0.52	0.53
	GWO	1.15×10^{-14}	1.33×10^{-14}	1.78×10^{-15}
	MFO	1.35×10^{-2}	8.46	8.60
	本文算法	$\mathbf{8.88\times10^{-16}}$	$\mathbf{8.88\times10^{-16}}$	0
Griewank函数	MRFO	0	1.89×10^{-2}	1.51×10^{-2}
	GWO	0	0	0
	MFO	1.53×10^{-3}	7.87×10^{-3}	4.87×10^{-3}
	本文算法	0	0	0

由表可见,MRRJRFO在单峰值测试函数上的最优适应度比MRFO,GWO和MFO更好,均值也优于其他算法,说明MRRJRFO的收敛精度最高.同时,标准差也最小,说明本文算法更稳定.在多峰值测试函数Rastrigin上,MRRJRFO的最优适应度优于MRFO和MFO算法,和GWO算法的最优适应度相同,都为0.而在Ackley函数上,MRRJRFO的最优适应度优于其他算法,同时均值和标准差也最低.在Griewank函数上,MRRJRFO的最优适应度与MRFO和GWO算法相当,优于MFO算法,同时比MRFO和MFO的均值和标准差更小,说明MRRJRFO更稳定.在以上多峰值函数上MRRJRFO都具有最优的适应度和标准差,说明寻优能力较强,有利于跳出局部最优,也更稳定.

由于MRRJRFO算法在多峰值测试函数中表现较好,这样通过与KMC算法进行联合迭代可以更好地寻找全局最优解.同时,MRRJRFO算法在单峰值测试函数中的表现也较好,所以在聚类过程中寻找全局最优解的同时,能进一步提升寻优精度,达到更好的聚类效果.

4.2　MRRJRFO-KMC算法性能评价　为了评价MRRJRFO-KMC算法的聚类优化效果,与MRFO-KMC,KMC[19],K-Means++[20]及模糊C均值(Fuzzy C-Means,FCM)聚类算法[21]在标准的国际数据集Iris,Aggregation,Ecoli和Seeds上进行测试.每种聚类算法均迭代50次,同时,MRRJRFO-KMC算法中MRRJRFO算法也迭代50次.设置FCM算法的权重指数m为2.1.每种算法进行20次实验.各标准数据集的特征如表2所示.

表 2 不同标准数据集的特征

Table 2 Features of different standard datasets

数据集	样本数目	属性维度	类别个数	各类样本数
Iris	150	4	3	50,50,50
Aggregation	788	2	7	170,34,273,102,130,45,34
Ecoli	366	8	8	143,77,2,2,259,20,5,52
Seeds	210	7	3	70,70,70

采用准确率(Accruacy,ACC)、调整兰德系数(Adjusted Rand Index,ARI)、标准化互信息(Normalized Mutual Information,NMI)三种指标来对不同聚类函数的测试结果进行评价.

(1)ACC可以体现聚类结果的准确程度,计算公式为:

$$ACC=\frac{TP+TN}{TP+TN+FP+FN} \tag{29}$$

其中,TP表示被正确地划分为正类的个数,TN表示被正确地划分为负类的个数,FP表示被错误地划分为正类的个数,FN表示被错误地划分为负类的个数.ACC越大,聚类划分的准确程度越高.

(2)ARI体现两个数据分布的吻合程度,其计算公式为:

$$ARI=\frac{RI+E(RI)}{\max(RI)-E(RI)} \tag{30}$$

其中,RI表示兰德系数.ARI越大,吻合程度越好.

(3)NMI表示两组数据的关联程度,计算公式为:

$$NMI=\frac{I(Q;W)}{\max\left(H(Q),H(W)\right)} \tag{31}$$

其中,Q表示聚类结果标签,W表示真实标签.$I(Q;W)$表示Q和W之间的互信息.$H(Q)$和$H(W)$分别表示Q和W的信息熵.NMI越大,两组数据关联程度越高.

图5为各算法通过本文提出的适应度函数在Iris,Aggregation,Ecoli和Seeds数据集上得到的收敛曲线.

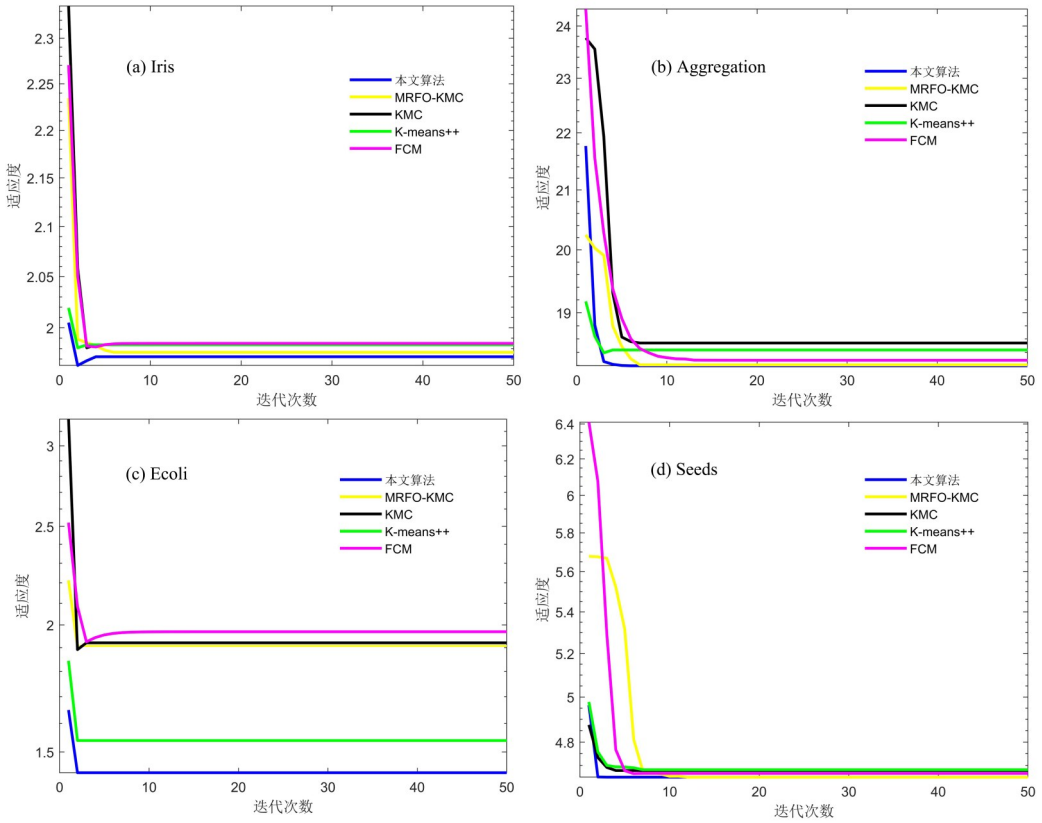

图5 各算法在不同数据集上的适应度曲线

Fig.5 Fitness curves of each algorithm on different datasets

由图可见,MRRJRFO-KMC在不同数据集上都能获得较好的收敛精度,收敛速度也比其他算法更快.其中,在Iris和Ecoli数据集上,本文算法的收敛精度更高,适应度更好;在Aggregation和Seeds数据集上,本文算法和MRFO-KMC算法的收敛精度最高,优于其他聚类算法,而且本文算法的收敛速度明显优于MRFO-KMC.

表3为各算法在不同数据集上的聚类评价指标,表中数据为20次实验结果的均值.

表3　各算法在不同数据集上的聚类评价指标

Table 3　Clustering evaluation index of each algorithm on different datasets

数据集	算法	*ACC*	*ARI*	*NMI*
Iris	MRFO-KMC	0.8900	0.7527	0.7620
	KMC	0.8453	0.6659	0.7203
	K-Means++	0.8833	0.6598	0.7170
	FCM	0.8852	0.6725	0.7137
	本文算法	0.9267	0.8091	0.8300
Aggregation	MRFO-KMC	0.6853	0.5486	0.3743
	KMC	0.6713	0.5243	0.3443
	K-Means++	0.6777	0.5334	0.3559
	FCM	0.6853	0.5517	0.3732
	本文算法	0.7931	0.7286	0.5741
Ecoli	MRFO-KMC	0.9470	0.9225	0.7846
	KMC	0.7619	0.6780	0.6468
	K-Means++	0.8619	0.6980	0.6468
	FCM	0.9273	0.8238	0.6415
	本文算法	0.9935	0.9929	0.8397
Seeds	MRFO-KMC	0.9095	0.7671	0.6796
	KMC	0.8023	0.7488	0.6345
	K-Means++	0.8295	0.7576	0.6723
	FCM	0.9033	0.7635	0.6739
	本文算法	0.9286	0.8487	0.7234

由表可见,本文算法在各数据集上的*ACC*,*ARI*和*NMI*都优于其他算法,其中*ACC*指标在Aggregation数据集上明显优于其他算法,*ARI*和*NMI*指标在每个数据集上的提升都比较明显.说明本文算法的聚类准确度高,优化效果最好.

通过比较图5和表3可以看出,KMC的稳定性最差;K-Means++的精度和稳定性比KMC算法有所提升;FCM算法中引入了模糊控制的概念,使迭代曲线变得相对稳定,但该算法对初始点的选取不够稳定,不能保证全局最优.本文算法可以很好地解决聚类中心选取随机性的问题,同时也有利于跳出局部最优,提升了聚类精度、稳定性和迭代速度.

4.3　MRRJRFO-KMC在电影信息数据集中的聚类分析　选取MovieLens数据集进行聚类分析研究.MovieLens数据集由GroupLens小组在网站上搜集并提供给学者研究,共有四个版本,分别为100 kB,1 MB,10 MB,20 MB.各版本的具体内容如表4所示.

表4　MovieLens数据集及其不同版本的属性

Table 4　Properties of different versions of MovieLens dataset

数据集	100 kB	1 MB	10 MB	20 MB
用户数	943	6040	71567	138439
影片数	1682	3900	10681	27287
评分数	10万	100万	1000万	2000万

本文选择20 MB的数据集进行研究,包括用户信息$U=\{u_1,u_2,\cdots,u_n\}$、电影信息$E=\{e_1,e_2,\cdots,e_n\}$和用户评分$g\in(0,1,2,3,4,5)$.用户信息提供每个用户的个人信息及观影信息;电影信息包括爱情、动作、科幻等19种类型的27287部电影;用户评分指用户对电影的打分信息,分0～5六个级别,0分表示未对电影做评价,1分代表评价最低,5分代表评价最高,1～5评价逐次提升.因为20 MB数据比较集中,没有更新收录最新的电影,所以在使用MovieLens数据集的同时,在豆瓣网上对近年的电影进行爬虫抓取,爬取用户信息数据、电影信息数据和用户评分数据,并将其中的非数值型数据转化为数值型,最终爬取得到200部最新的电影.将20 MB数据集中的电影与爬取的电影相结合,构成最终的聚类数据集.将数据写成(u,e,g)的形式,表示用户u对电影e的评价为g.

选取10%的数据集进行实验,对用户聚类,实验目的是将电影兴趣一致的用户聚为一类,所以要对用户进行特征向量提取,即提取每个用户对每部电影的评价.用MRRJRFO-KMC聚类,将每个用户聚在不同类型的电影上的评分相加,再除以总评次数,得到最终结果,如表5所示.

由表可见,聚类结果分为四类.分析可知,第

表 5 不同用户对不同类型电影的评分聚类

Table 5 Clustering of ratings for different types of movies by different users

Genres	Label1	label2	label3	label4
动作	4.56	1.35	1.73	3.67
冒险	3.23	0.38	1.28	4.38
科幻	1.23	1.78	2.35	4.74
爱情	1.17	4.37	3.58	2.13
喜剧	0.59	3.96	3.63	1.32
战争	4.26	1.56	0.52	3.23
西部	4.15	0.75	1.37	2.54
动画	0.18	3.78	4.53	0.85
家庭	1.45	4.53	3.98	1.78
纪录	2.12	3.58	3.12	2.54
伦理	1.34	4.28	2.75	2.13
运动	1.65	2.73	3.22	2.65
音乐	0.34	4.33	3.32	0.84
悬疑	3.74	1.85	1.88	4.25
儿童	1.86	3.54	4.38	1.38
犯罪	4.43	1.52	2.17	3.58
惊悚	3.98	0.32	2.11	4.28
奇幻	3.13	1.89	2.33	4.52
传记	2.25	2.32	2.78	3.54

一类人群对动作、战争、西部、犯罪等类型电影的评分比较高,说明他们更偏爱打斗、暴力类的电影;第二类人群对爱情、家庭、伦理、音乐以及纪录类的电影评分比较高,说明他们偏好比较文艺的电影;第三类人群对动画、喜剧、家庭、儿童类电影评分比较高,说明他们可能对比较温馨、温暖的电影较为喜爱;第四类人群和第一类人群对电影的评分比较相似,但对科幻、冒险、奇幻类电影的评分更高,说明他们更偏好科幻类型的影片.

5 结论

本文提出一种基于随机跳跃式翻滚觅食蝠鲼优化的K均值联合迭代聚类算法.首先通过均值最大最小距离积法解决KMC在初始化聚类中心时的随机性问题,提高了聚类精度和稳定性.其次,提出一种随机跳跃式翻滚觅食蝠鲼优化算法,能较好地跳出局部最优,提高全局寻优能力,同时加快寻优速度.通过随机跳跃式翻滚觅食蝠鲼优化算法和KMC算法的联合迭代,提高了聚类算法的聚类精度和收敛速度,增强了聚类算法的稳定性.最后,将本文算法应用到电影信息数据集中,实现了较好的聚类效果,应用价值明显.

参考文献

[1] 李瑞,冯锋.基于聚类和SVD++的电影推荐系统的研究.计算机时代,2020(9):88－90,94. (Li R, Feng F. Research on movie recommendation system using clustering and SVD++. Computer Era, 2020(9): 88－90, 94.)

[2] 王美琪,李建.一种改进K-Means聚类的近邻传播最大最小距离算法.计算机应用与软件,2021,38(7):240－245. (Wang M Q, Li J. An apmmd algorithm of improved K-Means initial cluster center. Computer Applications and Software, 2021, 38(7): 240－245.)

[3] Kim M S, Kim D, Eom H, et al. Segmentation of marine forecast zone in Korea by using wave data: Based on K-Means clustering algorithm. Journal of Coastal Research, 2021, 114(Sp1): 251－255.

[4] Capó M, Pérez A, LozanoJ A. An efficient split-merge Re-start for the K-Means algorithm. IEEE Transactions on Knowledge and Data Engineering, 2022, 34(4): 1618－1627.

[5] Vouros A, Langdell S, Croucher M, et al. An empirical comparison between stochastic and deterministic centroid initialisation for K-means variations. Machine Learning, 2021, 110(8): 1975－2003.

[6] 周本金,陶以政,纪斌,等.最小化误差平方和K-Means初始聚类中心优化方法.计算机工程与应用,2018,54(15):48－52. (Zhou B J, Tao Y Z, Ji B, et al. Optimizing K-Means initial clustering centers by minimizing sum of squared error. Computer Engineering and Applications, 2018, 54(15): 48－52.)

[7] 李丽亚,闫宏印.改进K-Means加权自适应多视图数据聚类算法.计算机仿真,2021,38(8):314－317,429. (Li L Y, Yan H Y. Improved K-Means weighted adaptive multi-view data clustering algorithm. Computer Simulation, 2021, 38(8): 314－317, 429.)

[8] Shen X L, Dou Y, Mills S, et al. Distributed sparse bundle adjustment algorithm based on three-dimensional point partition and asynchronous

communication. Frontiers of Information Technology & Electronic Engineering,2018,19(7):889－904.

[9] 刘佳鸣,况立群,尹洪红,等. 灰狼优化的k均值聚类算法. 中国科技论文,2019,14(7):778－782,807. (Liu J M,Kuang L Q,Yin H H,et al. K-Means clustering algorithm based on grey wolf optimization. China Sciencepaper,2019,14(7):778－782,807.)

[10] 尤烽骅,余玉聪,刘招,等. 基于联合改进人工蜂群及K均值聚类算法的洪水分类研究. 水文,2021,41(4):40－47. (You F H,Yu Y C,Liu Z,et al. Research on flood classification based on joint improved artificial bee colony and K-Means clustering algorithms. Journal of China Hydrology,2021,41(4):40－47.)

[11] Zhao W G,Zhang Z X,Wang L Y. Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. Engineering Applications of Artificial Intelligence,2020(87):103300.

[12] Chowdhury K,Chaudhuri D,Pal A K. An entropy-based initialization method of K-Means clustering on the optimal number of clusters. Neural Computing and Applications,2021,33(12):6965－6982.

[13] Arthur D,Date P. Balanced K-Means clustering on an adiabatic quantum computer. Quantum Information Processing,2021,20(9):294.

[14] 李璟楠,乐美龙. 多种群蝠鲼觅食优化求解多跑道机场航班排序. 航空计算技术,2020,50(6):47－51. (Li J N,Le M L. Multi-group manta ray foraging optimization for multi-runway airport flights sequencing. Aeronautical Computing Technique,2020,50(6):47－51.)

[15] 叶剑华,罗凤章,杨理. 基于改进蝠鲼觅食优化SVM的配电网拓扑辨识. 电力系统及其自动化学报,2021,33(10):43－50. (Ye J H,Luo F Z,Yang L. Distribution network topology identification based on SVM optimized by improved manta ray foraging optimization algorithm. Proceedings of the CSU-EPSA,2021,33(10):43－50.)

[16] 尚秋峰,杨根辈. 基于改进的蝠鲼觅食算法的光纤布拉格光栅解调方法. 光子学报,2021,50(9):63－71. (Shang Q F,Yang G B. Fiber Bragg grating demodulation method based on improved manta ray foraging optimization algorithm. Acta Photonica Sinica,2021,50(9):63－71.)

[17] Tolouei K,Moosavi E,Gholinejad M. An effective MIP model based on grey wolf optimizer for lot-sizing LTPSOP in open-pit mines under uncertainty. Arabian Journal of Geosciences,2021,14(17):1712.

[18] Chen Y B,Nie G H,Zhang H L,et al. Fast motion tracking based on moth-flame optimization and kernel correlation filter. Journal of Intelligent & Fuzzy Systems,2020,39(3):3825－3837.

[19] Mao Y M,Gan D J,Mwakapesa D S,et al. A MapReduce-based K-Means clustering algorithm. The Journal of Supercomputing,2022,78(4):5181－5202.

[20] Kłopotek M A. On the consistency of K-Means++ algorithm. Fundamenta Informaticae,2020,172(4):361－377.

[21] Khan I,Luo Z W,Huang J Z,et al. Variable weighting in fuzzy K-Means clustering to determine the number of clusters. IEEE Transactions on Knowledge and Data Engineering,2020,32(9):1838－1853.

(责任编辑　杨可盛)