控制与决策
Control and Decision

一种基于改进KH与KHM聚类的混合数据聚类算法

王秋萍， 丁成， 王晓峰

引用本文:
王秋萍, 丁成, 王晓峰. 一种基于改进KH与KHM聚类的混合数据聚类算法[J]. 控制与决策, 2020, 35(10): 2449−2458.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2019.0086

一种基于改进KH与KHM聚类的混合数据聚类算法

王秋萍†, 丁 成, 王晓峰

(西安理工大学 理学院，西安 710054)

摘 要: 为解决K-means聚类对初始聚类中心敏感和易陷入局部最优的问题,提出一种基于改进磷虾群算法与K-harmonic means的混合数据聚类算法. 提出一种具有莱维飞行和交叉算子的磷虾群算法以改进磷虾群算法易陷入局部极值和搜索效率低的不足,即在每次标准磷虾群位置更新后加入新的位置更新方法进一步搜索以提高种群的搜索能力,同时交替使用莱维飞行与交叉算子对当前群体位置进行贪婪搜索以增强算法的全局搜索能力. 20个标准测试函数的实验结果表明,改进算法不易陷入局部最优解,可在较少的迭代次数下有效地搜索到全局最优解的同时保证算法的稳定性. 将改进的磷虾群算法与K调和均值聚类融合,即在每次迭代后用最优个体或经过K调和均值迭代一次后的新个体替换最差个体. 5个UCI真实数据集的测试结果表明:融合后的聚类算法能够克服K-means对初始聚类中心敏感的不足且具有较强的全局收敛性.

关键词: 磷虾群算法；莱维飞行；交叉算子；K调和均值聚类；混合聚类

中图分类号: TP301.6 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2019.0086 **开放科学(资源服务)标识码(OSID):**

引用格式: 王秋萍,丁成,王晓峰. 一种基于改进KH与KHM聚类的混合数据聚类算法[J]. 控制与决策, 2020, 35(10): 2449-2458.

A hybrid data clustering algorithm based on improved krill herd algorithm and KHM clustering

WANG Qiu-ping†, DING Cheng, WANG Xiao-feng

(Faculty of Sciences，Xi'an University of Technology，Xi'an 710054，China)

Abstract: K-means clustering is sensitive to initial clustering centers and prone to fall into local optimum. In order to solve the problem, a hybrid data clustering algorithm based on an improved krill herd algorithm and K-harmonic means clustering is proposed. Firstly, an improved krill herd algorithm with Lévy flight and crossover operator is proposed to improve stagnating local optimum and low search efficiency of the krill herd algorithm. That is, after each standard krill herd location updating, a new location updating method is added to further search to improve the search ability of the population, at the same time, Lévy flight and crossover operators are used alternately to carry out greedy search for the current population position to enhance the global search ability of the algorithm. The experimental results of 20 benchmark test functions show that the improved algorithm is not easy to fall into the local optimum, which can find the global optimal solution via less times of iteration and ensure the stability of the algorithm. Then, the improved krill herd algorithm and the K-harmonic means clustering algorithm are fused to solve the data clustering problem, that is, the worst individual is replaced by the best individual or the new individual by the K-harmonic means processing the worst individual after each iteration. The test results of five real data sets on UCI show that the fused-clustering algorithm overcomes the defect that K-means is sensitive to the initial clustering center and has stronger global convergence.

Keywords: krill herd algorithm；Lévy flight；crossover operator；K-harmonic means clustering；hybrid clustering

0 引 言

近年来,数据挖掘已广泛应用于发现隐藏在数据内部的有用模式和知识. 聚类分析作为一种分析数据的挖掘方法,属于无监督学习技术,其形成方式使每个聚类在同一簇中的对象之间具有较高的内部相似性,而不同簇中的对象之间的相似性较低. 其中,基于划分的聚类是将数据对象集划分为不同的不重叠子集(簇),使每个数据对象恰在一个子集中[1].

K均值(K-means)聚类算法作为使用最广泛的基于划分的聚类算法,因其具有实现简单且聚类速度快的优点,已成功应用于各种重要领域,但同时也存在两个主要问题,即对初始聚类中心的高度敏感

收稿日期: 2019-01-17；**修回日期:** 2019-04-22.
基金项目: 国家自然科学基金项目(61772416).
责任编委: 林崇.
†通讯作者. E-mail: wqp566@sina.com.

性和对局部最优的收敛性[2]. 鉴于此,Zhang等[3]提出K调和均值(K-harmonic means, KHM)聚类算法,其目的是通过使用调和平均的距离作为目标的组成部分解决K均值聚类对初始中心敏感的问题. KHM已在图像处理[4-5]、医学[6]、光学[7]和遥感学[8]等不同领域得到了广泛的应用,但同时KHM也存在对噪声敏感、易陷入局部最优的不足[9]. 因此,近年来一些元启发式算法被融合其中,如模拟退火算法[10]、蚁群算法[11]、粒子群算法[12]、布谷鸟算法[13]等. 此类方法结合智能优化算法全局搜索能力的优势,解决KHM易陷入局部最优的不足. 本文将改进的磷虾群算法(krill herd algorithm, KH)[14]与K调和均值聚类算法相结合,在避免陷入局部最优的同时相对有效地找到全局最优解.

磷虾群算法是Gandomi等[14]从南极磷虾群体的生存环境和生活习性的仿真模拟实验中受到启发,于2012年提出的一种群智能优化算法. 与其他群智能算法一样,磷虾群算法在优化过程中,种群的多样性与算法的收敛速度之间始终存在矛盾. 因此,提出具有莱维飞行和交叉算子的磷虾群算法,即在每次标准磷虾群位置更新后加入新的位置更新方法再次进行搜索以提高种群的搜索能力,同时交替使用莱维飞行与交叉算子对当前群体位置进行贪婪搜索以提高种群多样性. 20个测试函数的实验结果表明,改进的KH算法性能优于标准KH算法及相关对比算法,在一定程度上可防止“早熟”现象的发生,具有较强的全局搜索能力和搜索效率.

为解决KHM易陷入局部最优的不足,本文将改进的KH算法与K调和均值聚类算法相结合用于解决数据聚类问题,提出基于改进磷虾群算法的K调和均值聚类算法. 利用改进后KH算法的全局搜索性与K调和均值高效的局部寻优能力,使得算法能够快速准确地找到最佳聚类中心,同时也解决了K均值聚类过于依赖初始聚类中心的不足. 将新的聚类算法在5个UCI数据集上进行测试,并与K调和均值聚类及2种求解聚类问题的优化算法进行比较,实验结果表明,将改进的KH算法与K调和均值聚类算法结合能够有效解决数据聚类问题,其求解精度和算法稳定性均得到了改善.

1 改进的磷虾群算法

1.1 磷虾群算法

磷虾群算法是一种基于元启发式种群的全局优化算法,是对磷虾群在寻找食物及个体之间相互交流的模拟. 磷虾的移动是一个多目标的过程,包括两个主要目标:提高磷虾密度、到达食物. Gandomi等[14]指出:在自然系统中,每个磷虾个体的适应度应该是磷虾个体与食物的距离和磷虾个体与磷虾群密度最高处的距离的组合,因此,适应度是目标函数值. 每个磷虾的位置X_i代表目标函数的一个可行解,其进化过程受诱导运动、觅食运动和随机扩散三者的协同影响. 磷虾群算法采用拉格朗日模型进行有效搜索,磷虾个体的位置变化速率描述如下:

$$\frac{\mathrm{d}X_i}{\mathrm{d}t}=N_i+F_i+D_i. \tag{1}$$

其中:$X_i=\{X_{i,1},X_{i,2},\dots,X_{i,\mathrm{NV}}\}$,其第$j$维变量$X_{i,j}$的搜索范围为$[\mathrm{LB}_j,\mathrm{UB}_j]$;$N_i$为磷虾$i$受其他磷虾诱导引起的运动;$F_i$为磷虾$i$的觅食运动;$D_i$为磷虾$i$的随机扩散运动.

1) 诱导运动.

磷虾i受其他磷虾诱导引起的运动N_i^{new}定义为

$$N_i^{\mathrm{new}}=N^{\max}\alpha_i+\omega_n N_i^{\mathrm{old}}. \tag{2}$$

其中:$N^{\max}$为最大诱导速度,取$0.01(\mathrm{ms}^{-1})$;N_i^{old}为当前个体i的先前诱导运动;ω_n为诱导运动的惯性权重;α_i为诱导方向,表示为

$$\alpha_i=\alpha_i^{\mathrm{local}}+\alpha_i^{\mathrm{target}}, \tag{3}$$

$\alpha_i^{\mathrm{local}}$为个体$i$受周围“邻居”的诱导方向,$\alpha_i^{\mathrm{target}}$为个体$i$受当前全局最优个体的诱导方向.

2) 觅食运动.

磷虾个体的觅食运动F_i定义为

$$F_i=V_f\beta_i+\omega_f F_i^{\mathrm{old}}. \tag{4}$$

其中:V_f为最大觅食速度,取$0.02(\mathrm{ms}^{-1})$;ω_f为觅食惯性权重;F_i^{old}为先前觅食运动;β_i为觅食方向,表示为

$$\beta_i=\beta_i^{\mathrm{food}}+\beta_i^{\mathrm{best}}, \tag{5}$$

β_i^{food}为个体i受“食物”诱导的方向,β_i^{best}为个体i受自身历史最优个体的诱导方向.

3) 随机扩散.

磷虾个体的随机扩散D_i可定义为

$$D_i=D^{\max}\Big(1-\frac{I}{I_{\max}}\Big)\delta. \tag{6}$$

其中:$D^{\max}$为最大随机扩散速度,取$0.005(\mathrm{ms}^{-1})$;δ为随机扩散方向;I和$I_{\max}$为当前迭代次数和最大迭代次数.

KH算法的粒子更新过程如下:

$$X_i(t+\Delta t)=X_i t+\Delta t\frac{\mathrm{d}X_i}{\mathrm{d}t}, \tag{7}$$

$$\Delta t=\mathrm{Ct}\sum_{j=1}^{\mathrm{NV}}(\mathrm{UB}_j-\mathrm{LB}_j). \tag{8}$$

其中: Δt为具体应用相关的时间间隔;步长因子Ct为常数且$\mathrm{Ct} \in (0,2]$;NV为决策变量的维数; UB_j和LB_j为第j维变量的上界和下界, $j=1,2,\ldots,\mathrm{NV}$.

1.2　新的位置更新方法

为进一步提高磷虾群算法的优化性能,基于文献[15]提出的粒子更新方法,在标准KH位置更新后,采用下式对磷虾群体进行位置更新:

$$X_{i,j}(I+1)=\mu+\sigma\times Z, \tag{9}$$

其中

$$\mu=(X_{i,j}(I)+\mathrm{pbest}_{i,j}(I)+\mathrm{gbest}_j(I))/3, \tag{10}$$

$$\sigma=([(X_{i,j}(I)-\mu)^2+(\mathrm{pbest}_{i,j}(I)-\mu)^2+(\mathrm{gbest}_j(I)-\mu)^2]/3)^{\frac{1}{2}}, \tag{11}$$

$$Z=(-2\ln k_1)^{1/2}\times\cos(2\pi k_2); \tag{12}$$

I为当前迭代次数, $X_{i,j}$为第i个磷虾位置的第j维, $\mathrm{pbest}_{i,j}$为第i个磷虾历史最优位置的第j维, gbest_j为全局最优位置的第j维, k_1和k_2为[0,1]范围内服从均匀分布的随机数.

1.3　莱维飞行

研究表明,许多动物的运动行为表现出莱维飞行的特征[16],作为一种随机的游走方式,其优点在于不间断地短跳跃与偶尔的长跳跃随机交替搜索,扩大了群体的搜索范围,保证算法能够及时跳出局部最优点. 基于莱维飞行[17]的位置更新公式表示如下:

$$X_i^{\mathrm{new}}=X_i+\mathrm{randn}(\mathrm{size}(\mathrm{NV}))\oplus 0.01\frac{\mu}{|\nu|^{\frac{1}{\beta}}}(X_i-X_{\mathrm{worst}}). \tag{13}$$

其中: X_i为当前的磷虾位置; X_i^{new}为莱维飞行后的磷虾位置;NV为维数; X_{worst}为上一代磷虾个体的最劣解,此方法可使磷虾以随机步长飞向原本的小概率探索区域,使搜索区域更加均匀,仿真结果表明其确实有效地增加了种群多样性,避免算法陷入局部最优[18]; μ和ν均服从正态分布$\nu:N(0,1)$, $\mu:N(0,\sigma_\mu^2)$,且有

$$\sigma_\mu=\left\{\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}}\right\}^{1/\beta}, \tag{14}$$

Γ为伽马函数.

在莱维步长中, β的取值范围为(0,2],文献[19]分析, β在低值区域易产生无效的σ,最终导致产生无效的个体. 本文参数β采用更精确的范围[0.1,2], 可有效防止产生无效解[19].

1.4　交叉算子

交叉算子能够有效增强个体间的信息共享,常被引入各种群智能算法中,可从一定程度上防止群体过早收敛. 本文采用算术交叉原则[20-21],对当前磷虾位置进行算术交叉以搜索到更多的邻域可行解,增强磷虾群体的种群多样性,操作如下:

$$X_i^{\mathrm{new}}=rX_i+(1-r)X_{j_i}. \tag{15}$$

其中: X_i^{new}为交叉运算后磷虾i的位置, r为[0,1]中均匀分布的随机数, $X_i(i=1,2,\ldots,N)$为磷虾i的位置. 对$(X_1,X_2,\ldots,X_N)$进行随机排列,有

$$\begin{bmatrix}X_1 & X_2 & \ldots & X_N\\ X_{j_1} & X_{j_2} & \ldots & X_{j_N}\end{bmatrix}. \tag{16}$$

对上层和下层的对应元素利用式(15)进行算术交叉, 显然, 该方法可以产生具有更好多样性的后代,有利于增强算法的全局搜索能力. 同时,对莱维飞行与交叉算子搜索的位置X_i^{new}采用贪心保留策略,即

$$X_i=\begin{cases}X_i^{\mathrm{new}}, & \mathrm{fit}(X_i^{\mathrm{new}})<\mathrm{fit}(X_i);\\ X_i, & \mathrm{fit}(X_i^{\mathrm{new}})\geqslant\mathrm{fit}(X_i).\end{cases} \tag{17}$$

其中$\mathrm{fit}(X_i)$为第i个磷虾的适应度值.

1.5　具有莱维飞行与交叉算子的磷虾群算法

针对KH在优化过程中易出现“早熟”和搜索效率较低的问题,本文在标准磷虾群位置更新后加入新的位置更新方法,进一步改善算法的寻优性能,同时采用莱维飞行与交叉算子交替贪婪搜索的方式,以扩大搜索范围,增强种群的全局搜索能力. 提出一种具有莱维飞行与交叉算子的磷虾群算法(LCKH),算法伪代码描述如下.

算法1　具有莱维飞行与交叉算子的KH算法.

初始化参数: 种群规模N,维数NV,最大迭代次数$I_{\max}$,随机初始化磷虾群位置.

计算磷虾适应度,初始化全局最优个体X_{gb}及适应度K_{gb},初始化个体i的历史最优个体X_{ib_i}及适应度K_{ib_i}.

for $I=1$ to $I_{\max}$ do

　for $i=1$ to N do

　根据式(2)～(6)求得3个运动分量;

　由式(7)更新个体i的位置X_i;

　重新评价适应度,更新X_{ib_i}与K_{ib_i};

　end for

　由式(9)更新群体的位置X;

　if $\mathrm{rem}(I,2)==1$

　由式(13)对磷虾群体位置进行莱维飞行,采用式(17)进行贪心保留;

　else

　由式(15)对磷虾群体位置进行交叉操作,采用式(17)进行贪心保留;

end if

更新整个群体的X_{gb}和K_{gb};

end for

输出最优解.

1.6 实验与分析

为验证LCKH的性能,本文选取20个经典的标准测试函数[14,20]进行仿真实验. 其中:$f_1 \sim f_{10}$为高维单峰函数,$f_{11} \sim f_{20}$为高维多峰函数. 此外,除f_{18}(维数为10)、f_{19}与f_{20}(维数为4)外,其余维数均为30,除f_{18}理论最优值为-9.66外,其余理论最优值均为0,详细见表1.

表 1 标准测试函数

函数名称	搜索空间
f_1(Sphere)	$[-100, 100]$
f_2(Schwefel 1.2)	$[-100, 100]$
f_3(Schwefel 2.21)	$[-100, 100]$
f_4(Schwefel 2.22)	$[-10, 10]$
f_5(Step)	$[-100, 100]$
f_6(Rosenbrock)	$[-30, 30]$
f_7(Dixon)	$[-10, 10]$
f_8(Sum Squares)	$[-10, 10]$
f_9(Brown)	$[-1, 4]$
f_{10}(Powell)	$[-4, 5]$
f_{11}(Griewank)	$[-600, 600]$
f_{12}(Ackley)	$[-32, 32]$
f_{13}(Rastrigin)	$[-5.12, 5.12]$
f_{14}(Alpine)	$[-10, 10]$
f_{15}(Zakharov)	$[-5, 10]$
f_{16}(Levy)	$[-10, 10]$
f_{17}(Wavy1)	$[-\pi, \pi]$
f_{18}(Michalewics)	$[0, \pi]$
f_{19}(Perm)	$[-4, 4]$
f_{20}(Power Sum)	$[0, 4]$

将LCKH与PSO[22]、LPSO[18]、SCA[23]、CS[17]、MFO[24]、KH[14]算法进行比较,其中LPSO为基于莱维飞行的粒子群算法. 数值实验在Matlab 2014环境下实现,各参数设置如下:种群规模为30,最大迭代次数为100次. 为保证比较的公平性,将所有算法的初始化种群设定相同. 在相同条件下,所有算法独立实验50次,取其最优值(best)与平均值(average)以反映解的质量,取标准差(std)以反映算法寻优的稳定性. 同时,为直观反映算法的优劣,根据每个测试函数的平均值进行排名(rank),综合求其总平均(mean rank)得出最终的排名(final rank),其中每个测试函数对应的最优值标记为黑体,见表2.

为了更直观地反映LCKH的优化性能,定性与定量地分析其寻优效果和优势,图1给出了部分测试函数的箱线图,图2给出了7种算法对部分标准测试函数的收敛曲线,表3给出了所有测试函数的收敛曲线积分值. 在相同迭代次数的情况下,积分值越小表明其收敛速度越快.

LCKH算法的优化性能分析如下:

1) 求解精度.

表2中,对比7类算法在20个测试函数上的优化结果,即best、ave与final rank可以直观地看出,无论对于单峰函数还是多峰函数的优化,本文所提出算法LCKH都具有较高的求解精度,尤其对于f_1、f_4、f_9、f_{15}和f_{17},相比较标准KH算法,其解的精度得到了显著提高.

2) 稳定性.

结合表2的Std指标和图1,可定性与定量地分析算法稳定性. 首先,LCKH在20个测试函数上的标准差均为最低,同时,由箱线图可直观看出,LCKH正常值的分布较为集中且异常值较少,其他算法正常值分布较为分散且存在较多异常值,表明LCKH在优化过程中具有良好的稳定性.

3) 收敛速度.

综合表2、表3和图2可定性与定量地分析算法收敛速度. 对于较简单的多维单峰函数(见图2(a)~图2(c)),在迭代前期优势不明显,而在迭代后期LCKH显然具有较快的收敛速度和求解精度;对于较为复杂的多维多峰函数(见图2(d)~图2(f)),其他6类算法出现了搜索停滞或搜索缓慢的现象,即陷入了局部最优而难以跳出,而LCKH并未有早熟现象发生,表明其具备一定的全局搜索能力以防止陷入局部最优. 此外,对于测试函数f_3、f_7、f_{10}、f_{12}、f_{13}、f_{16}和f_{15},虽其求解精度没有大幅度提升,但LCKH的收敛曲线积分值较低,表明在相同的迭代次数下,LCKH具有更快的收敛速度.

综上所述,LCKH在解决单峰或多峰的优化问题时,均可在较短的迭代次数下快速有效地寻找到全局最优解,且最优解具有较高的精度和稳定性.

表 2　7种算法在20个测试函数上的优化结果

算法	f_1				f_2				f_3			
	best	ave	std	rank	best	ave	std	rank	best	ave	std	rank
PSO	1.90e+01	3.64e+01	1.00e+01	2	1.08e+03	3.88e+03	1.11e+04	3	4.10e+00	6.42e+00	1.52e+00	2
LPSO	1.94e+01	4.19e+01	1.20e+01	3	6.90e+02	3.96e+03	1.11e+04	4	4.44e+00	6.44e+00	1.46e+00	3
SCA	2.79e+02	3.56e+03	1.94e+03	5	1.13e+04	3.01e+04	1.16e+04	6	3.88e+01	6.68e+01	1.16e+01	6
CS	2.24e+03	4.70e+03	1.16e+03	6	1.31e+04	2.28e+04	4.93e+03	5	2.71e+01	3.84e+01	4.08e+00	5
MFO	4.52e+03	1.26e+04	5.75e+03	7	1.80e+04	3.67e+04	8.20e+03	7	5.84e+01	7.23e+01	7.56e+00	7
KH	8.45e+00	7.84e+01	6.24e+01	4	7.15e+02	1.78e+03	4.66e+02	2	3.98e+00	8.28e+00	2.36e+00	4
LCKH	**1.84e-03**	**6.41e-03**	**4.07e-03**	**1**	**1.19e+02**	**4.30e+02**	**2.29e+02**	**1**	**5.61e-01**	**2.41e+00**	**1.17e+00**	**1**

算法	f_4				f_5				f_6			
	best	ave	std	rank	best	ave	std	rank	best	ave	std	rank
PSO	2.85e+01	5.30e+01	1.52e+01	4	2.20e+01	4.25e+01	1.65e+01	3	1.16e+04	3.19e+04	1.89e+04	3
LPSO	2.56e+01	4.91e+01	1.43e+01	3	1.80e+01	3.94e+01	1.22e+01	2	1.00e+04	3.66e+04	1.86e+04	4
SCA	9.38e-01	5.81e+00	3.53e+00	2	2.16e+02	4.60e+03	3.44e+03	5	2.20e+04	1.04e+07	1.03e+07	6
CS	4.42e+01	4.64e+02	1.87e+03	6	2.65e+03	4.99e+03	1.19e+03	6	8.50e+05	2.20e+06	9.78e+05	5
MFO	3.52e+01	6.87e+01	1.53e+01	5	5.49e+03	1.19e+04	5.31e+03	7	2.59e+06	1.17e+07	5.88e+06	7
KH	2.41e+02	3.44e+08	9.06e+08	7	8.60e+01	3.86e+02	2.41e+02	4	4.19e+02	2.74e+03	3.90e+03	2
LCKH	**3.00e-02**	**5.29e-02**	**1.81e-02**	**1**	**0.00e+00**	**4.00e-01**	**4.98e-01**	**1**	**1.52e+01**	**7.13e+01**	**6.58e+01**	**1**

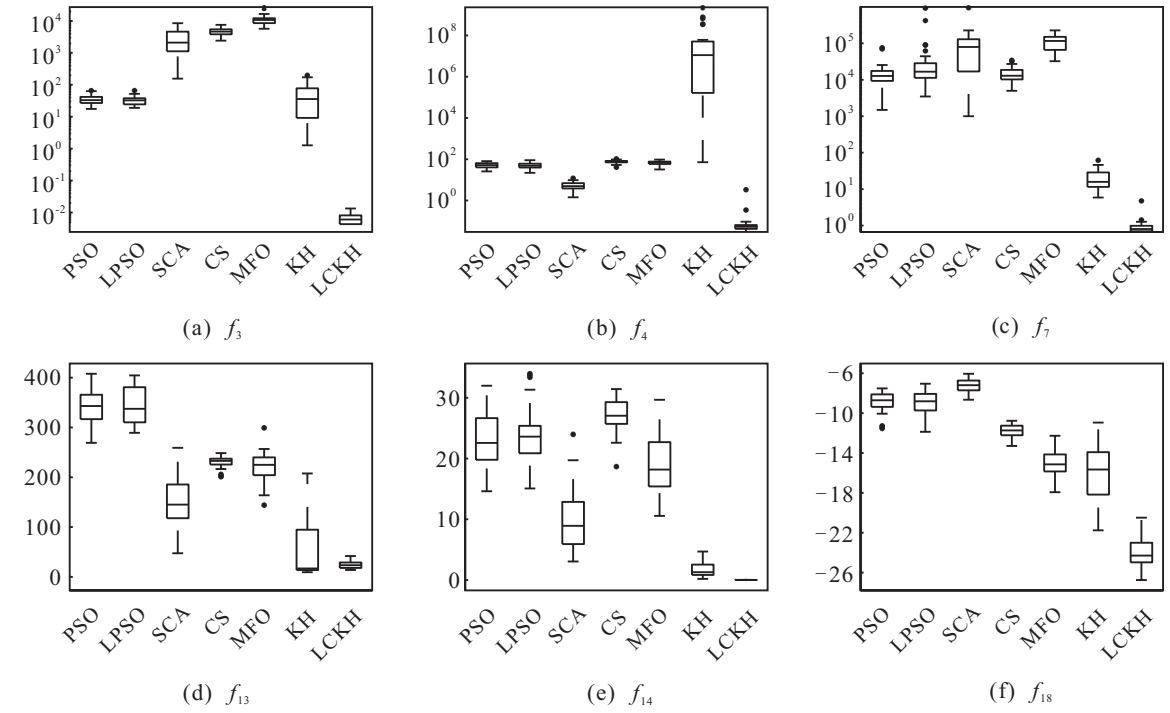
算法	f_7				f_8				f_9			
	best	ave	std	rank	best	ave	std	rank	best	ave	std	rank
PSO	2.01e+03	5.39e+04	1.11e+05	4	4.14e+02	1.24e+03	1.20e+03	6	1.74e+01	4.47e+02	1.87e+03	6
LPSO	2.32e+03	5.56e+04	1.13e+05	5	2.41e+02	8.72e+02	4.93e+02	5	2.26e+01	3.88e+02	1.87e+03	5
SCA	5.91e+03	7.73e+04	5.85e+04	6	2.84e+01	3.64e+02	2.64e+02	3	8.05e-02	1.35e+00	8.91e-01	2
CS	3.58e+03	1.83e+04	8.45e+03	3	3.01e+02	6.00e+02	1.65e+02	4	7.70e+00	1.67e+01	5.25e+00	3
MFO	1.51e+04	1.33e+05	9.93e+04	7	6.65e+02	1.40e+03	4.00e+02	7	9.07e+00	1.87e+01	1.16e+01	4
KH	4.32e+00	2.33e+01	1.64e+01	2	3.21e+00	1.90e+01	1.29e+01	2	1.23e+02	1.21e+03	4.56e+03	7
LCKH	**6.67e-01**	**9.47e-01**	**5.13e-01**	**1**	**8.75e-04**	**8.40e-03**	**1.59e-02**	**1**	**1.59e-05**	**8.98e-02**	**1.54e-01**	**1**

算法	f_{10}				f_{11}				f_{12}			
	best	ave	std	rank	best	ave	std	rank	best	ave	std	rank
PSO	2.85e+02	1.53e+03	1.62e+03	7	1.06e+00	2.40e+00	1.01e+00	3	4.33e+00	5.42e+00	6.04e-01	3
LPSO	2.66e+02	1.41e+03	1.74e+03	6	1.35e+00	3.33e+00	2.08e+00	4	4.62e+00	5.60e+00	5.87e-01	4
SCA	3.97e+01	2.88e+02	2.29e+02	4	2.89e+00	3.84e+01	2.64e+01	5	6.40e+00	1.66e+01	4.61e+00	5
CS	9.14e+01	2.54e+02	7.24e+01	3	3.03e+01	4.60e+01	9.16e+00	6	1.49e+01	1.75e+01	1.33e+00	6
MFO	2.02e+02	6.35e+02	7.26e+02	5	3.78e+01	8.77e+01	3.52e+01	7	1.39e+01	1.82e+01	2.10e+00	7
KH	8.58e-01	5.23e+00	4.43e+00	2	1.04e+00	1.55e+00	5.39e-01	2	3.47e+00	5.15e+00	1.10e+00	2
LCKH	**4.81e-02**	**1.74e-01**	**9.08e-02**	**1**	**4.26e-02**	**9.41e-02**	**2.83e-02**	**1**	**1.27e-02**	**6.48e-01**	**7.59e-01**	**1**

算法	f_{13}				f_{14}				f_{15}			
	best	ave	std	rank	best	ave	std	rank	best	ave	std	rank
PSO	2.57e+02	3.33e+02	3.85e+01	6	1.62e+01	2.40e+01	4.63e+00	5	3.28e+02	4.66e+07	2.55e+08	6
LPSO	2.82e+02	3.40e+02	4.09e+01	7	1.22e+01	2.40e+01	4.85e+00	6	4.55e+02	4.66e+07	2.55e+08	5
SCA	3.34e+01	1.37e+02	4.57e+01	3	1.48e+00	9.99e+00	4.14e+00	3	8.29e+01	1.58e+02	4.51e+01	2
CS	2.05e+02	2.36e+02	1.64e+01	5	2.05e+01	2.65e+01	2.49e+00	7	2.82e+02	4.05e+02	6.45e+01	3
MFO	1.61e+02	2.19e+02	3.15e+01	4	6.57e+00	1.83e+01	4.91e+00	4	3.51e+02	5.46e+02	9.95e+01	4
KH	**7.77e+00**	4.73e+01	5.49e+01	2	8.47e-02	9.93e-01	7.69e-01	2	2.24e+02	6.23e+07	1.49e+08	7
LCKH	1.49e+01	**2.45e+01**	**8.05e+00**	**1**	**4.06e-03**	**2.60e-02**	**2.59e-02**	**1**	**4.95e+01**	**8.96e+01**	**2.44e+01**	1

算法	f_{16}				f_{17}				f_{18}			
	best	ave	std	rank	best	ave	std	rank	best	ave	std	rank
PSO	2.96e+01	7.23e+01	2.26e+01	6	2.46e-01	3.26e-01	7.21e-02	6	−7.9454	−6.0711	7.97e-01	5
LPSO	1.16e+01	6.51e+01	2.35e+01	5	2.31e-01	3.29e-01	7.58e-02	7	−7.8308	−6.0084	7.55e-01	6
SCA	2.72e+01	8.76e+01	3.76e+01	7	3.60e-04	1.30e-02	1.07e-02	3	−4.3834	−3.5457	4.25e-01	7
CS	9.00e+00	4.95e+01	1.97e+01	4	1.35e-01	2.03e-01	3.76e-02	4	−7.5058	−6.5965	4.23e-01	3
MFO	3.36e+00	1.82e+01	8.99e+00	3	1.25e-01	2.12e-01	4.27e-02	5	−9.0851	−7.7262	6.80e-01	2
KH	2.27e-01	2.30e+00	1.75e+00	2	6.65e-04	3.37e-03	2.43e-03	2	−9.0232	−6.5540	1.02e+00	4
LCKH	**3.22e-04**	**2.39e-01**	**5.22e-01**	**1**	**4.28e-08**	**1.92e-07**	**6.97e-08**	**1**	**-9.5207**	**-8.8756**	**4.06e-01**	1

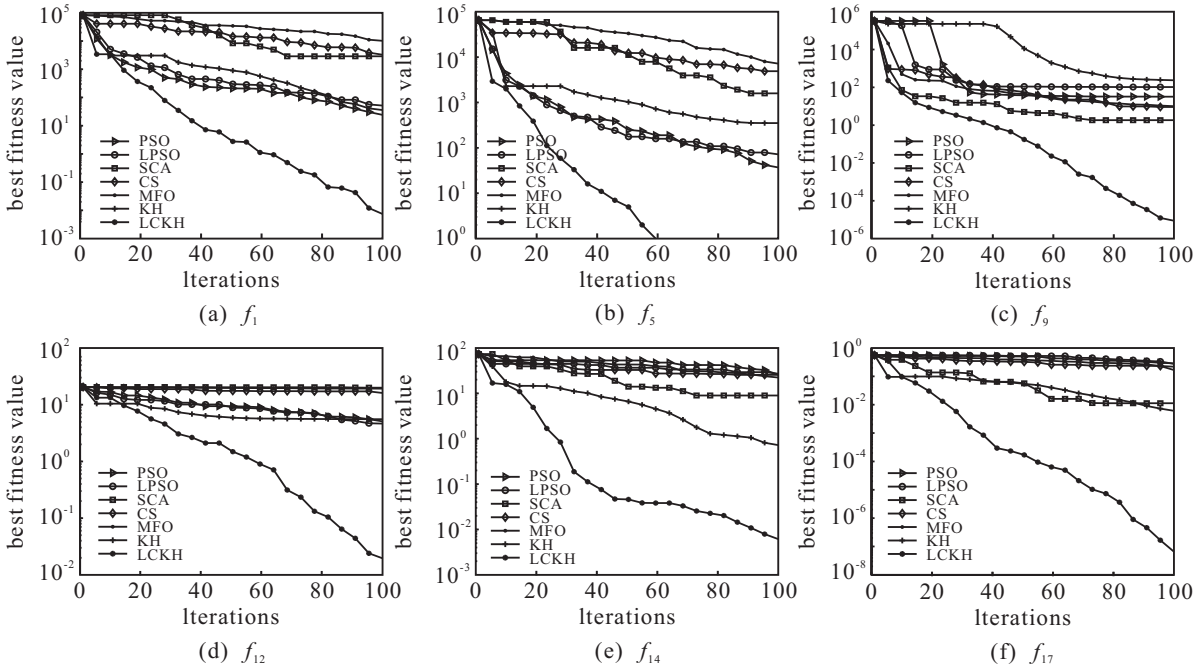
算法	f_{19}				f_{20}				mean rank	final rank
	best	ave	std	rank	best	ave	std	rank		
PSO	4.51e-03	4.80e-01	1.41e+00	2	4.16e-03	2.75e-01	2.86e-01	5	4.35	3
LPSO	2.98e-03	1.21e+00	5.46e+00	5	2.47e-03	2.14e-01	3.22e-01	4	4.55	5
SCA	2.72e-01	4.35e+00	2.92e+00	6	1.06e-01	5.81e+00	8.39e+00	7	4.65	7
CS	4.82e-02	7.45e-01	6.99e-01	4	9.69e-04	4.63e-02	4.51e-02	2	4.50	4
MFO	5.10e-04	5.94e-01	1.82e+00	3	9.26e-05	1.01e-01	1.78e-01	3	5.25	6
KH	1.46e+00	2.39e+02	3.98e+02	7	1.05e-03	4.59e-01	8.08e-01	6	3.6	2
LCKH	**1.23e-03**	**1.25e-01**	**1.44e-01**	**1**	**1.08e-06**	**2.12e-03**	**3.90e-03**	**1**	**1**	**1**



(a) f_3 (b) f_4 (c) f_7

(d) f_{13} (e) f_{14} (f) f_{18}

图 1 部分测试函数的箱线图



(a) f_1 (b) f_5 (c) f_9

(d) f_{12} (e) f_{14} (f) f_{17}

图 2 部分测试函数的收敛曲线

2 基于改进KH与KHM的混合数据聚类算法

2.1 K调和均值聚类

在研究K调和均值聚类前,首先对聚类过程中的符号进行说明:

$X = \{x_1, \ldots, x_n\}$:需要聚类的数据集,对象n个;

$C = \{c_1, \ldots, c_k\}$:簇中心集,聚类中心为k个;

$m(c_j|x_j)$:数据点x_i隶属于c_j的隶属度, 由隶属度函数确定;

$w(x_i)$:在下一次迭代中数据点x_i对计算中心参数的影响程度,由权重函数确定.

KHM聚类算法[5]的基本步骤如下.

step 1: 根据解空间的上下界随机选择初始中心.

step 2: 计算目标函数值

$$\mathrm{KHM}(X,C) = \sum_{i=1}^{n} \frac{k}{\sum\limits_{j=1}^{k} \dfrac{1}{\|x_i - c_j\|^p}}. \tag{18}$$

step 3: 对于每个数据点x_i,计算其在每个中心c_j中的隶属度

表 3　7类算法在20个测试函数上的收敛曲线积分值

函数	PSO	LPSO	SCA	CS	MFO	KH	LCKH
f_1	2.36e+05	2.58e+05	1.67e+06	1.06e+06	1.66e+06	3.61e+05	**1.23e+05**
f_2	6.99e+05	7.09e+05	3.14e+06	2.12e+06	2.65e+06	1.78e+06	**4.65e+05**
f_3	7.94e+02	8.10e+02	2.54e+03	2.05e+03	2.44e+03	6.69e+02	**5.39e+02**
f_4	3.09e+12	3.09e+12	3.48e+12	3.16e+12	1.55e+13	3.63e+13	**3.09e+12**
f_5	2.32e+05	2.47e+05	1.76e+06	1.11e+06	1.69e+06	3.47e+05	**1.19e+05**
f_6	3.75e+08	3.83e+08	6.76e+09	2.89e+09	5.64e+09	9.91e+08	**2.23e+08**
f_7	2.39e+07	2.43e+07	5.23e+07	2.10e+07	4.18e+07	7.98e+06	**1.71e+06**
f_8	1.68e+05	1.73e+05	2.18e+05	1.60e+05	2.29e+05	5.57e+04	**1.76e+04**
f_9	2.93e+09	1.85e+09	1.44e+09	3.87e+08	1.85e+09	9.79e+09	**3.65e+08**
f_{10}	2.24e+05	2.19e+05	2.31e+05	1.32e+05	2.05e+05	8.57e+04	**1.60e+04**
f_{11}	1.06e+04	1.05e+04	1.55e+04	1.01e+04	1.52e+04	3.18e+03	**1.08e+03**
f_{12}	4.29e+02	4.32e+02	5.87e+02	5.77e+02	5.91e+02	3.24e+02	**3.00e+02**
f_{13}	1.25e+04	1.25e+04	1.14e+04	1.05e+04	1.15e+04	7.47e+03	**5.86e+03**
f_{14}	1.47e+03	1.50e+03	1.62e+03	1.42e+03	1.56e+03	6.94e+02	**3.95e+02**
f_{15}	3.67e+09	5.14e+09	5.99e+08	8.78e+08	2.07e+09	2.39e+10	**5.74e+08**
f_{16}	4.10e+03	3.87e+03	4.66e+03	4.05e+03	5.13e+03	1.99e+03	**7.20e+02**
f_{17}	1.66e+01	1.66e+01	7.86e+00	1.37e+01	1.49e+01	4.73e+00	**2.54e+00**
f_{18}	−3.02e+01	−2.98e+01	−2.55e+01	−3.31e+01	−3.06e+01	−3.01e+01	**−4.51e+01**
f_{19}	3.54e+03	3.26e+03	2.54e+03	2.31e+03	4.64e+03	5.96e+03	**1.54e+03**
f_{20}	8.73e+01	9.73e+01	1.63e+02	8.34e+01	2.14e+02	2.73e+02	**6.30e+01**

$$m(c_j|x_i)=\frac{\|x_i-c_j\|^{-p-2}}{\sum\limits_{j=1}^{k}\|x_i-c_j\|^{-p-2}}. \tag{19}$$

step 4: 对于每个数据点x_i,计算其权重

$$w(x_i)=\frac{\sum\limits_{i=1}^{k}\|x_i-c_j\|^{-p-2}}{\Big(\sum\limits_{j=1}^{k}\|x_i-c_j\|^{-p}\Big)^2}. \tag{20}$$

step 5: 对于每个聚类中心c_j,根据其隶属度、权重和所有数据点x_i重新计算聚类中心

$$c_j=\frac{\sum\limits_{i=1}^{n}m(c_j|x_i)w(x_i)x_i}{\sum\limits_{i=1}^{n}m(c_j|x_i)w(x_i)}. \tag{21}$$

step 6: 重复step 2 ∼ step 5, 直到达到预定义的迭代次数或KHM(X,C)没有显著变化为止.

step 7: 将数据点x_i分配到$m(c_j|x_i)$最大的簇j中.

针对K均值对初始聚类中心敏感的问题, 文献[2-3]分别证明了KHM对聚类中心的初始化本质上是不敏感的, 故采用KHM进行聚类, 但是其同时也存在易收敛到局部最优的不足[9].

2.2　基于改进KH与KHM的混合数据聚类算法

由于KH在寻优迭代过程中会反复评估目标函数, 而KHM算法具有更快的收敛速度, 但也往往易陷入局部最优而无法找到最佳的聚类中心. 本文将改进的KH算法(LCKH)与KHM相结合, 提出一种基于改进KH与KHM的混合聚类算法(IKH-KHM). 该算法保持了LCKH和KHM各自的优点, 其思想是利用KHM较快的寻优效率且对初始聚类中心不敏感的优点, 在KH每次种群迭代后, 当全局最差的磷虾个体变为较优的个体时, 其余周围磷虾会迅速向新的最优个体靠拢, 同时LCKH的全局搜索能力可防止其陷入局部最优, 从而达到快速有效地找到最佳聚类中心的目的.

每次迭代后, 排序得出种群最差个体$X_{\rm worst}$, 将其替换为新个体X^*, 此方法的特点在于引入学习机制, 即适应度最差的个体向较好的个体学习, 新个体计算如下:

$$X^*=\begin{cases}\mathrm{KHM}(X_{\rm worst}),\ I<p_rI_{\max};\\ X_{\rm best},\ \text{otherwise.}\end{cases} \tag{22}$$

其中: $X_{\rm best}$为当前全局最优个体, KHM$(X_{\rm worst})$为对

最差个体作一次K调和均值操作(只迭代一次);p_r为[0, 1]之间的常数, 通过实验测试本文取0.5.

IKH-KHM的基本步骤如下.

算法2 基于改进KH与KHM的混合聚类算法.

初始化参数: 种群规模N, 维数NV, 最大迭代次数$I_{\max}$, 在初始解空间内随机初始化磷虾群的位置, 即初始聚类中心.

计算所有磷虾的适应度, 初始化全局最优个体X_{best}和适应度K_{gb}, 初始化个体i的历史最优个体X_{ib_i}和适应度K_{ib_i};

for $I = 1$ to $I_{\max}$ do

for $I = 1$ to N do

求得3个运动分量;

更新个体i的位置X_i;

重新评价适应度, 更新X_{ib_i}与K_{ib_i};

end for

由式(9)更新群体的位置X;

if $\text{rem}(I, 2) == 1$

由式(13)对磷虾群体位置进行莱维飞行, 采用式(17)进行贪心保留;

else

由式(15)对磷虾群体位置进行交叉操作, 采用式(17)进行贪心保留;

end if

更新整个群体的X_{best}与K_{gb};

根据适应度排序找出最差个体X_{worst}与最优个体X_{best};

if $I < p_r I_{\max}$

将KHM(X_{worst})得到的新个体替换最差个体X_{worst};

else

将全局最优个体X_{best}替换最差个体X_{worst};

end if

更新整个群体的X_{best}与K_{gb};

end for

输出最优解.

2.3 实验分析

为验证IKH-KHM的性能, 本文选取UCI数据库中Iris、Wine、Glass、Wisconsin Breast Cancer (Cancer)和Contraceptive Method Choice (CMC)五个常用的数据集进行仿真实验, 表4为这些数据集的属性和特征. 并将其与KHM[5]、PSOKHM[12]和ICMPKHM[13]进行比较. 由于测试数据集与目标函数都相同, 结果直接取自相应参考文献. 具体参数设置: 将IKH-KHM分别执行30次取其平均值, 括号内为标准差以衡量算法稳定性, 磷虾个体数设置为10, 最大迭代次数为100次, 结果KHM(X, C)即为目标函数值, 其值越小表明聚类效果越好, 同时F-Measure为另一重要评价指标, 其值越大表明聚类效果越好, 结果如表5～表7所示.

表4 测试数据集

数据集名称	簇数	特征数	数据个数(每个簇的个数)
Iris	3	4	150 (50, 50, 50)
Wine	3	13	178 (59, 71, 48)
Glass	5	9	214 (70, 17, 76, 13, 9, 29)
Cancer	2	9	683 (444, 239)
CMC	3	10	1473 (629, 333, 511)

表5 $p = 2.5$时, 4种算法在5个数据集上的聚类结果

	指标	KHM	PSOKHM	ICMPKHM	IKH-KHM
Iris	KHM(X, C)	149.333 (0)	149.058 (0.074)	149.045 (0.047 5)	**148.814 7** (6.94e-05)
	F-Measure	0.750 0 (0)	0.753 0 (0.005)	**0.892 2** (0.001 4)	0.885 3 (3.38e-16)
Wine	KHM(X, C)	75 603 652 (16.231)	75 642 795 (123 127)	74 943 916 (121 621)	**75 332 606** (969.52)
	F-Measure	0.421 (0.011)	0.6786 (0.008 7)	0.678 5 (0.009)	**0.706 2** (0.0174)
Glass	KHM(X, C)	1 203 (0.16)	1 196 (0.43)	1 240 (9.65)	**1 188** (22.49)
	F-Measure	0.421 0 (0.011)	0.424 0 (0.003)	0.647 1 (0.019 853)	**0.729 0** (0.060 6)
Cancer	KHM(X, C)	60 189 (0)	59 844 (22)	57 166 (0.625)	**56 817** (0.009 2)
	F-Measure	0.829 (0.000)	0.829 (0.000)	**0.961** (0.000)	0.960 (0.003 1)
CMC	KHM(X, C)	96 520 (0)	96 193 (25)	96 568 (125)	**96 057** (0.305 1)
	F-Measure	0.335 (0.000)	0.333 (0.002)	**0.464 754** (0.003)	0.457 2 (0.009 4)

表 6　$p=3$时, 4种算法在5个数据集上的聚类结果

	指标	KHM	PSOKHM	ICMPKHM	IKH-KHM
Iris	KHM(X,C)	126.517 (0.000)	125.951 (0.052)	126.278 (0.063)	**125.749 0** (6.55e-05)
	F-Measure	0.744 (0.000)	0.744 (0.000)	0.891 1 (0.000 616)	**0.891 8** (5.64e-16)
Wine	KHM(X,C)	1 085 350 475 (10.531)	1 075 350 475 (5 934 548)	1 066 602 517 (3 911 744)	**1 049 058 622** (13 733)
	F-Measure	0.656 0 (0.006 5)	0.647 0 (0.008 4)	0.647 8 (0.010)	**0.651 2** (0.019 3)
Glass	KHM(X,C)	1 535 (0.000)	1 442 (35.87)	1 737 (18.84)	**1 420** (113)
	F-Measure	0.422 (0.000)	0.427 (0.003)	0.672 816 (0.016 377)	**0.725 9** (0.060 3)
Cancer	KHM(X,C)	119 458 (0)	117 418 (237)	113 706 (4.578)	**111 240** (0.954 4)
	F-Measure	0.834 (0.000)	0.834 (0.000)	0.9647 (0.000)	**0.964 7** (0.008 5)
CMC	KHM(X,C)	187 525 (0)	186 722 (111)	187 350 (132.273)	**186 320** (6.796 5)
	F-Measure	0.303 (0.000)	0.303 (0.000)	**0.464** (0.002 978)	0.455 1 (0.010 8)

表 7　$p=3.5$时, 4种算法在5个数据集上的聚类结果

	指标	KHM	PSOKHM	ICMPKHM	IKH-KHM
Iris	KHM(X,C)	113.413 (0.085)	110.004 (0.260)	111.495 (0.472)	**109.127** (4.56e-05)
	F-Measure	0.770 (0.024)	0.762 (0.004)	**0.892 5** (0.000)	0.8918 (5.64e-16)
Wine	KHM(X,C)	16 038 256 460 (0.321)	15 938 236 000 (3.75e+08)	15 668 536 846 (2.64e+08)	**14 191 285 650** (4.92e+04)
	F-Measure	0.635 34 (0.621)	0.631 34 (0.007 6)	0.631 6 (0.006)	**0.649 3** (0.0179)
Glass	KHM(X,C)	1 871.812 (0.000)	1 857.152 (4.937)	2 244.93 (90.13)	**1 820.71** (9.25)
	F-Measure	0.396 (0.000)	0.396 (0.000)	0.671 (0.017)	**0.714 1** (0.054)
Cancer	KHM(X,C)	243 440 (0)	235 441 (696)	232 137 (24.844)	**221 963** (3.034 3)
	F-Measure	0.832 (0.000)	0.835 (0.003)	**0.966** (0.000)	0.964 8 (0.008)
CMC	KHM(X,C)	381 444 (0)	379 678 (247)	383 564 (329)	**378 584** (56.72)
	F-Measure	0.332 (0.000)	0.332 (0.000)	**0.462 1** (0.003)	0.456 3 (0.011)

F-Measure定义如下:

若已知类i中的样本数目n_i、簇j中的样本数目n_j(由算法生成)以及簇j中属于已知类i的样本数目n_{ij},则判准率为$P(i,j)=n_{ij}/n_j$,查全率为$r(i,j)=n_{ij}/n_i$,数据集的总体F-measure为

$$F=\sum_i \frac{n_i}{n}\mathrm{MAX}_j\{F(i,j)\}, \tag{23}$$

其中$F(i,j)=((b^2+1)P(i,j)r(i,j))/(b^2P(i,j)+r(i,j))$,本文$b=1$.

对于数据集Iris, IKH-KHM的目标函数值KHM(X,C)均为最优, $p=3$时的F-measure最优,其余略差于ICMPKHM;对于数据集Wine和Glass, KH-KHM的目标函数值KHM(X,C)与F-measure测度均为最优;对于数据集Cancer和CMC, IKH-KHM的目标函数值KHM(X,C)均为最优,但F-measure略差于ICMPKHM. 综合表5~表7, IKH-KHM的目标函数值在所有数据集中均为最优,表明其对于目标函数值的优化效果显著,同时其F-measure在大部分为最优,部分略差于ICMPKHM.

3　结　论

本文提出了一种基于改进KH与KHM的混合数据聚类算法. 在标准KH中作出以下改进: 1) 加入新的位置更新方法以提高种群的搜索能力,从而提高了算法的搜索效率; 2) 引入莱维飞行与交叉算子交替搜索使算法能够及时跳出局部最优的同时保持了种群的多样性,从而增强了算法的全局搜索能力. 20个测试函数的实验结果表明,改进算法LCKH与其他几个智能算法相比,具有较高的收敛精度和搜索效率,尤其在解决多峰优化问题时体现出其优势. 将LCKH与KHM融合用于求解数据聚类问题, KHM解决了K均值聚类算法对初始聚类中心敏感的问题, IKH-KHM克服了KHM对噪声敏感,易陷入局部最优的不足,从而IKH-KHM在解决数据聚类问题时具有较好全局收敛性和稳定性. 通过对UCI上5个数据集的测试结果验证了所提出算法IKH-KHM的可行性和有效性.

与其他聚类算法相比, IKH-KHM可以更快更准确地进行数据聚类,但与KHM相比运行时间较长.

下一步的研究方向为: 1)通过将KH与其他优化策略或算法相结合,更进一步提高KH的性能,并将算法应用于解决生产调度、路径规划、文本文档聚类和约束优化等实际工程问题; 2)将其他智能算法与KHM结合,以达到更快的收敛速度、精度和运行时间.

参考文献(References)

[1] Tan Pang-ning, Steinbach Michael, Kumar Vipin, et al. Introduction to data mining[M]. Beijing: Posts & Telecom Press, 2011: 306.

[2] Krista Rizman Žalik. An efficient k-means clustering algorithm[J]. Pattern Recognition Letters, 2008, 29(9): 1385-1391.

[3] Zhang Bin, Hsu Meichun, Dayal Umeshwar. K-harmonic means — A data clustering algorithm[R]. Palo Alto: Hewlett-Packard Laboratories, 1999.

[4] Carvalho V O. Combining K-Means and K-Harmonic with fish school search algorithm for data clustering task on graphics processing units[J]. Applied Soft Computing, 2016, 41: 290-304.

[5] Zhou Z, Zhao X, Zhu S. K-harmonic means clustering algorithm using feature weighting for color image segmentation[J]. Multimedia Tools & Applications, 2018, 77: 15139-15160.

[6] Khanmohammadi S, Adibeig N, Shanehbandy S. An improved overlapping k-means clustering method for medical applications[J]. Expert Systems with Applications, 2017, 67: 12-18.

[7] 武斌, 王大智, 武小红,等. 茶叶傅里叶红外光谱的可能模糊K调和均值聚类分析[J]. 光谱学与光谱分析, 2018, 38(3): 745-749.
(Wu B, Wang D Z, Wu X H, et al. Possibilistic fuzzy K-harmonic means clustering of fourier transform infrared spectra of tea[J]. Spectroscopy and Spectral Analysis, 2018, 38(3): 745-749.)

[8] Mahi H, Farhi N, Labed K. Remotely sensed data clustering using K-harmonic means algorithm and cluster validity index[J]. IFIP Advances in Information and Communication Technology, 2018, 456: 105-116.

[9] Yeh W C, Lai C M, Chang K H. A novel hybrid clustering approach based on K-harmonic means using robust design[J]. Neurocomputing, 2016, 173: 1720-1732.

[10] Güngör Z, Ünler A. K-harmonic means data clustering with simulated annealing heuristic[J]. Applied Mathematics & Computation, 2007, 184(2): 199-209.

[11] Jiang H, Yi S, Li J, et al. Ant clustering algorithm with K-harmonic means clustering[J]. Expert Systems with Applications, 2010, 37(12): 8679-8684.

[12] Yang F, Sun T, Zhang C. An efficient hybrid data clustering method based on K-harmonic means and particle swarm optimization[J]. Expert Systems with Applications, 2009, 36(6): 9847-9852.

[13] Bouyer A, Hatamlou A. An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms[J]. Applied Soft Computing, 2018, 67: 172-182.

[14] Gandomi A H, Alavi A H. Krill herd: A new bio-inspired optimization algorithm[J]. Communications in Nonlinear Science & Numerical Simulation, 2012, 17(12): 4831-4845.

[15] Servet M Kiran. Particle swarm optimization with a new update mechanism[J]. Applied Soft Computing, 2017, 60: 670-678.

[16] Chechkin A V, Metzler R, Klafter J, et al. Introduction to the theory of Lévy flights[M]. Anomalous Transport: Foundations and Applications, 2008: 1-41.

[17] Yang X S, Suash Deb. Cuckoo search via Lévy flights[C]. World Congress on Nature & Biologically Inspired Computing. Coimbatore: IEEE, 2009: 210-214.

[18] 王学武, 严益鑫, 顾幸生. 基于莱维飞行粒子群算法的焊接机器人路径规划[J]. 控制与决策, 2017, 32(2): 373-377.
(Wang X W, Yan Y X, Gu X S. Welding robot path planning based on Lévy-PSO[J]. Control and Decision, 2017, 32(2): 373-377.)

[19] 张新明, 王霞, 涂强, 等. 趋优算子和Lévy flight混合的粒子群优化算法[J]. 电子科技大学学报, 2018, 47(3): 103-111.
(Zhang X M, Wang X, Tu Q, et al. Particle sarm optimization algorithm based on combining global-best operator and Lévy flight[J]. Journal of University of Electronic Science and Technology of China, 2018, 47(3): 103-111.)

[20] Tawhid M A, Ali A F. Simplex particle swarm optimization with arithmetical crossover for solving global optimization problems[J]. Opsearch, 2016, 53: 705-740.

[21] Chen Y, Li L, Xiao J, et al. Particle swarm optimizer with crossover operation [J]. Engineering Applications of Artificial Intelligence, 2018, 70: 159-169.

[22] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[C]. Proceedings of the 6th International Symposium on Micro Machine and Human Science. Nagoya: IEEE, 1995: 39-43.

[23] Mirjalili S. SCA: A sine cosine algorithm for solving optimization problems[J]. Knowledge-Based Systems, 2016, 96: 120-133.

[24] Mirjalili S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm[J]. Knowledge-Based Systems, 2015, 89: 228-249.

作者简介

王秋萍(1964−), 女, 教授, 博士, 从事预测技术与决策分析、智能计算、灰色系统理论等研究, E-mail: wqp566@sina.com;

丁成(1994−), 男, 硕士生, 从事群智能优化算法、聚类分析的研究, E-mail: 511265378@qq.com;

王晓峰(1966−), 女, 教授, 博士, 从事智能计算、统计学习、图像认证等研究, E-mail: xfwang66@sina.com.

(责任编辑: 郑晓蕾)