DOI:10.11992.tis.201505043
网络出版地址:http://www.cnki.net/kcms/detail/23.1538.tp.20151110.1354.006.html

融合并行混沌萤火虫算法的 K-调和均值聚类

朱书伟,周治平,张道文
(江南大学 物联网工程学院,江苏 无锡 214122)

摘　要:针对 K-调和均值算法易陷于局部最优的缺点,提出一种基于改进萤火虫算法(firefly algorithm, FA)的 K-调和均值聚类算法。将基于 FA 的粗搜索与基于并行混沌优化 FA 的精细搜索相结合,其中精细搜索部分首先通过 FA 搜索到当前最优解及次优解,然后通过改进的 logistic 映射与并行混沌优化策略产生混沌序列在其附近直接搜索,以增强算法的寻优性能。最终,将这种改进的 FA 用于 K-调和均值算法聚类中心的优化。实验结果表明:该算法不但对几种测试函数具有更高的搜索精度,而且对 6 种数据集的聚类结果均有一定的改善,有效地抑制了 K-调和均值算法陷于局部最优的问题,提高了聚类准确性和稳定性。

关键词:K-调和均值;局部最优;萤火虫算法;聚类;并行混沌优化;混沌局部搜索;映射模型;种群多样性
中图分类号:TP18　**文献标志码**:A　**文章编号**:1673-4785(2015)06-0872-09

中文引用格式:朱书伟,周治平, 张道文. 融合并行混沌萤火虫算法的 K-调和均值聚类[J]. 智能系统学报, 2015, 10(6): 872-880.
英文引用格式:ZHU Shuwei, ZHOU Zhiping, ZHANG Daowen. K-harmonic means clustering merged with parallel chaotic firefly algorithm[J]. CAAI Transactions on Intelligent Systems, 2015, 10(6): 872-880.

K-harmonic means clustering merged with parallel chaotic firefly algorithm

ZHU Shuwei, ZHOU Zhiping, ZHANG Daowen
(School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China)

Abstract: The K-harmonic means algorithm (KHM) has the disadvantage of easily falling into a local optimum. To solve this problem, we propose a hybrid KHM based on an improved firefly algorithm (FA). In this paper, we combined raw FA-based searching with parallel chaotic FA-based elaborate searching. In the elaborate searching, we found the current best and second-best solutions using the FA, then we used an improved logistic map model combined with parallel chaotic optimization to search this area in order to enhance the searching ability of the algorithm. Finally, we used the improved FA to optimize the cluster centers obtained by the KHM. Experimental results demonstrate that the proposed algorithm not only had higher search precision for several test functions, but also improved the clustering accuracy and stability of six datasets, effectively avoiding being trapped into a local optimum.
Keywords: K-harmonic means; local optimum; firefly algorithm; clustering; parallel chaotic optimization; chaotic local search; map model; diversity of population

聚类分析是一种广泛使用的数据分析方法,一直被应用于多个领域,特别是在数据挖掘、模式识别、图像处理等领域应用十分广泛。K-means[1]是最经典且使用最为广泛的聚类算法,其过程简单快捷,容易实现。为了克服 K-means 对初始聚类中心敏感的缺陷,Zhang 等[2]于 1999 年提出一种 K-调和均值(K-harmonic means, KHM)算法,具有较高的稳定性、收敛速度快,但由于其与 K-means 同样基于划分的原理,仍存在易陷于局部最优的问题。

目前,对于 KHM 算法的研究主要是结合智能

收稿日期:2015-05-27.　**网络出版日期**:2015-11-10.
基金项目:江苏省产学研联合创新资金-前瞻性联合研究基金资助项目(BY2013015-33).
通信作者:朱书伟. E-mail:6131905056@vip.jiangnan.edu.cn.

优化算法进行改进,以充分利用其全局搜索能力,如融合粒子群[3]、变邻域搜索[4]、改进候选组搜索[5]等混合聚类算法。此外,将模糊概念引入 KHM 中也得到了一定的关注[6-7]。目前,各种群智能优化算法已被广泛地应用于各个领域中[8-11],并且依据没有免费的午餐定律,本文提出新的混合聚类算法。萤火虫算法(firefly algorithm, FA)是由剑桥学者 Yang 等[12-13]在 2008 年提出的一种新颖的群智能算法,具有结构简单、可调参数少、宜于并行处理等特点,可以有效解决各种优化问题,并能够成功应用到聚类问题中提高算法的准确性和鲁棒性[14]。很多学者已经对它开展了不少研究工作,引入混沌原理改进的 FA 具有一定的优势,Fister 等[15]对现有的混沌萤火虫算法(chaos-based firefly algorithm, CFA)进行了总结,它们的主要思想都是基于算法参数的改进,其中 Gandomi 等[16]采用各种混沌映射模型进行了比较全面的对比分析。然而,仅对参数的调整无法更全面有效地利用混沌优化的优点,混沌局部搜索(chaotic local search, CLS)[9-10]是一种能够有效提高算法优化性能的策略。

本文从进一步提高 FA 的优化性能出发,提出一种新颖的 CFA,并将其融入到 KHM 以获得一种更有效的混合聚类方法。在 FA 中引入一种并行混沌局部搜索策略,将 CLS 与并行混沌优化(parallel chaotic optimization, PCO)[17-18]相结合,提高 FA 的局部搜索能力,具有更高的搜索效率,并能够有效避免局部最优。将这种改进的 CFA 融入到 KHM 中优化其目标函数,通过对实际数据集的实验可以看出本文所提的聚类算法能够获得更好的性能指标,有效抑制了陷入局部最优的问题。

1　算法概念与定义

1.1　K-调和均值算法

K-调和均值算法的原理基本上与 K-means 是相似的,不同的是其使用调和均值(harmonic means, HM)代替算术均值来计算目标函数,能够有效解决对初始类中心点选取的敏感性问题。假定数据集 $\boldsymbol{X}=[\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \cdots \quad \boldsymbol{x}_n]$ 包含 n 个数据,它们被划分到 k 个聚类簇,每个簇的中心用 $\boldsymbol{c}_j(j=1,2,\cdots,k)$ 表示,KHM 的目标函数为[3]

$$\mathrm{KHM}(\boldsymbol{X},\boldsymbol{C})=\sum_{i=1}^{n}\frac{k}{\sum_{j=1}^{k}\frac{1}{\|\boldsymbol{x}_i-\boldsymbol{c}_j\|^{p}}},\forall i=1,2,\cdots,n \tag{1}$$

这里采用欧式距离计算样本到聚类中心的距离,参数 p 对算法的性能具有重要的影响,且当 $p\geqslant 2$ 时聚类的效果比较好[2]。算法通过不断地迭代使目标函数值不断减小并保持稳定,每次迭代过程中,各个簇的中心点 $\boldsymbol{c}_j(j=1,2,\cdots,k)$ 的更新如下[3]。

$$\boldsymbol{c}_j^{\mathrm{new}}=\frac{\sum_{i=1}^{n}m_{\mathrm{KHM}}(\boldsymbol{c}_j/\boldsymbol{x}_i)\times w_{\mathrm{KHM}}(\boldsymbol{x}_i)\times\boldsymbol{x}_i}{\sum_{i=1}^{n}m_{\mathrm{KHM}}(\boldsymbol{c}_j/\boldsymbol{x}_i)\times w_{\mathrm{KHM}}(\boldsymbol{x}_i)} \tag{2}$$

式中:成员函数 m_{KHM} 和权重函数 w_{KHM} 的定义分别为式(3)和式(4)。

$$m_{\mathrm{KHM}}(\boldsymbol{c}_j/\boldsymbol{x}_i)=\frac{\|\boldsymbol{x}_i-\boldsymbol{c}_j\|^{-p-2}}{\sum_{j=1}^{K}\|\boldsymbol{x}_i-\boldsymbol{c}_j\|^{-p-2}} \tag{3}$$

$$w_{\mathrm{KHM}}(\boldsymbol{x}_i)=\frac{\sum_{j=1}^{k}\|\boldsymbol{x}_i-\boldsymbol{c}_j\|^{-p-2}}{\left(\sum_{j=1}^{k}\|\boldsymbol{x}_i-\boldsymbol{c}_j\|^{-p}\right)^2} \tag{4}$$

1.2　萤火虫算法的相关定义

在 FA 中萤火虫彼此吸引主要取决于 2 个因素:亮度和吸引度。亮度决定了个体所处位置的好坏及其移动方向,吸引度决定了移动的距离,通过亮度和吸引度的不断更新,实现目标优化。通常直接利用目标函数值的大小表示萤火虫 i 的亮度 I_i,即 $I_i=f(\boldsymbol{x}_i)$, $\boldsymbol{x}_i=[x_{i1} \quad x_{i2} \quad \cdots \quad x_{id}]$。FA 的相关定义如下[12-13]:

定义 1　萤火虫 i 与 j 之间的吸引度为

$$\beta=\beta_0\mathrm{e}^{-\gamma r_{ij}^2} \tag{5}$$

式中:β_0 为在 $r=0$ 处的吸引度,一般可取值为 1;γ 为光强吸收系数,对算法的性能具有重要的影响,通常情况下可以取 $\gamma=1$;r_{ij} 为萤火虫 i 与 j 之间的空间距离,一般采用欧氏距离计算。

定义 2　萤火虫 i 被更亮的萤火虫 j 吸引而移动的位置为

$$\boldsymbol{x}_i^{\mathrm{new}}=\boldsymbol{x}_i+\beta(\boldsymbol{x}_j-\boldsymbol{x}_i)+\alpha\boldsymbol{\varepsilon}_i \tag{6}$$

式中:$\boldsymbol{x}_i$、$\boldsymbol{x}_j$ 为萤火虫 i 和 j 的位置;α 为步长因子,可设为常数;$\boldsymbol{\varepsilon}_i$ 为服从均匀分布的随机数向量。

2　基于改进 FA 的 K-调和均值聚类

2.1　并行混沌局部搜索策略改进的 FA

基本的 FA 缺乏变异机制,当处于局部极值时难以摆脱,且当前最优解 $\boldsymbol{x}_{pg}$ 周围是搜索到更优解的最有利的区域,而 FA 在优化过程中采用对其随机扰动的方式,搜索效率不高。混沌优化方法能够有效地跳出局部最优并搜索到全局最优解,现有文献

对混沌模型的研究非常广泛,如 logistic 映射、Sinusoidal 映射、Gaussian 映射等[16,19]。文献[9-10]中采取一种改进 logistic 映射分别与粒子群(particle swarm optimization, PSO)算法和差分进化算法融合提出 2 种有效的基于 CLS 的混合优化算法,成功用于短期梯级水电系统调度问题,并且在文献[19]中验证了这种混沌映射的优势,它具有较大的李雅普诺夫指数。logistic 映射模型为[19]

$$y(l+1)=4y(l)(1-y(l)),\quad y(l)\in(0,1) \tag{7}$$

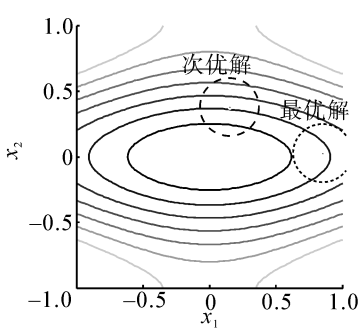
式中:l 表示迭代次数,需要注意的是混沌变量初始值 $y(0)\notin\{0.25,0.5,0.75\}$,若设置 $y(l)=(z(l)+1)/2$,则可以获得改进的 logistic 映射如式(8)[19]:

$$z(l+1)=1-2(z(l))^2, z(l)\in(-1,1) \tag{8}$$

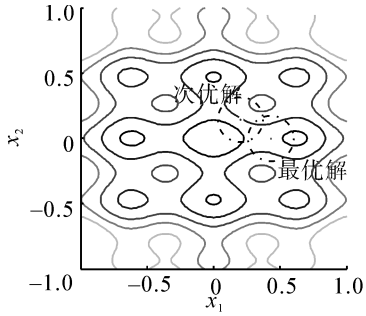
并且其概率密度分布表达式为

$$f(z)=\begin{cases}\dfrac{1}{\pi\sqrt{1-z^2}}, z\in(-1,1)\\ 0, z\notin(-1,1)\end{cases} \tag{9}$$

由式(9)可以看出改进 logistic 映射可以将混沌变量的搜索空间拓展到(-1,1),在接近边界-1 和 1 处具有较大的概率密度值,因此具有更好的遍历性、随机性。因此,本文利用改进 logistic 映射在当前最优解附近直接搜索,其本质上属于一种混沌干扰法,即产生许多局部最优解的邻域点,以增强搜索到全局最优解的概率。与此同时,适应度值仅次于最优解 $\boldsymbol{x}_{pg}$ 的次优解 $\boldsymbol{x}_{ps}$ 同样对搜索到更优解具有一定的价值,文献[20]中以最优点和次优点为基础进行反射、延伸、收缩等步骤的单纯形法也为本文提供一定的启发。为了更直观地分析,在图 1 中分别给出二维的单峰和多峰搜索空间的 2 种特殊情况的次优解与最优解局部搜索区域,它们的局部搜索半径均相等,并且假定越往内适应度值越好。从图 1(a)、(b)中可见 2 种特殊情况下次优解相对于最优解均具有更好的搜索潜力。



(a)单峰



(b)多峰

图 1 2 种特殊情况的最优解与次优解局部搜索区域

Fig.1 Two particular types of local search region around the best and second best solutions

为了进一步提高搜索效率,提出一种并行混沌局部搜索(parallel chaotic local search, PCLS)策略,采用并行混沌优化的思想产生 N 个混沌局部变量对次优解和最优解并行扰动,不但克服传统 CLS 的串行机制搜索精确解效率不高、收敛稳定性不强等缺点[9-10],还能够有效地兼顾最优解与次优解。当最优解和次优解接近时可将它们的作用视为相等,不接近时则能够有效地拓展局部搜索空间。每次迭代后取 N 个并行变量与 $\boldsymbol{x}_{pg}$ 和 $\boldsymbol{x}_{ps}$ 综合排序获得新的最优解和次优解,有效地提高算法搜索能力。

考虑到文献[17-18]中 PCO 结合了粗搜索与细搜索的策略以平衡算法的探索与开发性能,为了使并行混沌局部搜索萤火虫算法(parallel chaotic local search firefly algorithm, PCLSFA)在前期进行一定的粗搜索,可在前 $T_{\max_1}$ 次迭代直接执行 FA,PCLSFA 的具体过程为:

1)初始化萤火虫个体的位置并计算其对应的目标函数值 I_i 作为亮度,初始化迭代次数 $t=0$,最大迭代次数设为 $T_{\max}$,粗搜索迭代次数 $T_{\max_1}$。

2)执行 FA 不断更新亮度,最亮的个体即为当前最优解 $\boldsymbol{x}_{pg}$,并且次优解为 $\boldsymbol{x}_{ps}$,若 $t>T_{\max_1}$,则采用 PCLS 在它们附近寻优作为细搜索。

3)设置当前混沌搜索次数 $l=0$,在上文几个断点外的区域初始化混沌变量 $-1<z_{ij}^{(0)}<1$ ($i=1,2,\cdots,N$; $j=1,2,\cdots,m$),N 为并行变量数,m 为单变量维数,则 z_{ij} 表示第 i 个并行变量的第 j 维。此外,中间变量矩阵为 $\boldsymbol{Y}$,PCLS 最大迭代次数为 $C_{\max}$。

①考虑到大多数情况下 $\boldsymbol{x}_{pg}$ 具有更好的搜索潜力,令 $\boldsymbol{y}_i^{(l)}=\boldsymbol{x}_{pg}(l)$,$i=1,2\cdots,\dfrac{2N}{3}$,且 $\boldsymbol{y}_i^{(l)}=\boldsymbol{x}_{ps}(l)$,$i=\dfrac{2N+1}{3},\dfrac{2N+2}{3},\cdots,N$,使用式(8)确定第 l+1 次迭代的混沌扰动变量 $z_{ij}^{(l+1)}$。

②混沌变量与收缩因子 β_t 成比例,通过混沌扰动产生 N 个新变量如式(10)所示。

$$y_{ij}^{(l+1)} = y_{ij}^{(l)} + \beta_t z_{ij}^{(l+1)} \boldsymbol{l}_{\mathrm{PCLS}} \tag{10}$$

式中:$\boldsymbol{l}_{\mathrm{PCLS}}$ 为并行混沌局部搜索的范围,可将其设置为 $0.01\boldsymbol{l}$ ~$0.1\boldsymbol{l}$,$\boldsymbol{l}$ 为变量尺度,若 $\boldsymbol{u}_b$ 、$\boldsymbol{l}_b$ 分别为变量的上下界,则取 $\boldsymbol{l} = (\boldsymbol{u}_b - \boldsymbol{l}_b)/2$,收缩因子 β_t 为

$$\beta_t = \mathrm{e}^{-C*t/T_{\max}} \tag{11}$$

式中:C 是一个用于控制 PCLS 精度的正数,根据实验分析可在[1,10]内选取,一般对于较难搜索到全局最优的问题取较小值。求得 N 个新变量组成的矩阵为

$$\boldsymbol{Y}^{(l+1)} = \begin{bmatrix} y_{11}^{(l+1)} & y_{12}^{(l+1)} & \cdots & y_{1m}^{(l+1)} \\ y_{21}^{(l+1)} & y_{22}^{(l+1)} & \cdots & y_{2m}^{(l+1)} \\ \vdots & \vdots & & \vdots \\ y_{N1}^{(l+1)} & y_{N2}^{(l+1)} & \cdots & y_{Nm}^{(l+1)} \end{bmatrix} \tag{12}$$

③计算每个新变量所对应的目标函数值为 $f(\boldsymbol{y}_i^{(l+1)})$,并将 $\boldsymbol{Y}^{(l+1)}$ 与 $\boldsymbol{x}_{pg}^{(l+1)}$ 、$\boldsymbol{x}_{ps}^{(l+1)}$ 合并,对这 N+2 个变量的适应度值进行排序,得到第 l+1 次迭代中的最优解 $\boldsymbol{x}_{pg}^{(l+1)}$ 和次优解 $\boldsymbol{x}_{ps}^{(l+1)}$ 。

④$l=l+1$,若 $l<C_{\max}$,转向①;否则转向 4)。

4)$t=t+1$,若 $t<T_{\max}$,转向 2),且随机选取一个萤火虫个体用 3)中获得的 $\boldsymbol{x}_{pg}$ 替换并更新其亮度;否则停止迭代,输出全局最优解。

2.2　提高种群多样性的策略

由于 FA 缺乏保持种群多样性的操作,降低了算法探索到全局最优解的能力,因此需要采取一定的措施来解决这一问题。本文中算法每迭代 N_p 次时,找出适应度值最差的 n_c% 的个体并采用混沌重构法生成新的个体替代它们。对于各维尺度相等的优化问题,直接计算出当前种群所有维空间的最大值 $x_{\max}$和最小值 $x_{\min}$作为各维的统一边界。对于各维尺度不相等的优化问题,对边界向量不断地收缩,初始时第 j 维的边界等于定义域$[a_j,b_j]$,当达到第 N_p次迭代的最优个体为 $\boldsymbol{x}^*$,根据式(13)收缩边界。

$$\begin{cases} a_j^{\mathrm{new}} = x_j^* - \varphi(b_j - a_j) \\ b_j^{\mathrm{new}} = x_j^* + \varphi(b_j - a_j) \end{cases} \tag{13}$$

式中:$\varphi \in (0,0.5)$,并且为了保证新的边界范围不会越界,对其进行相应的处理为:若 $a_j^{\mathrm{new}} < a_j$,则 $a_j^{\mathrm{new}} = a_j$;若 $b_j^{\mathrm{new}} > b_j$,则 $b_j^{\mathrm{new}} = b_j$。然后根据式(7)的 logistic 映射生成比例为 n_c% 的 N_c 个在(0,1)上的向量 $\boldsymbol{Cx}_i(i = 1,2,\cdots,N_c)$ 如式(14)所示。

$$\boldsymbol{Cx}_i = 4 \times \boldsymbol{y} \times (1 - \boldsymbol{y}), \boldsymbol{y} \in (0,1) \tag{14}$$

最后再将其转换到当前种群变量的取值空间如式(15)所示,获得替换种群并更新其适应度值。

$$\boldsymbol{x}_i^{\mathrm{new}} = \boldsymbol{b} + \boldsymbol{Cx}_i(\boldsymbol{b} - \boldsymbol{a}) \tag{15}$$

这里随着迭代次数的增加对边界范围不断收缩,在各个不同阶段生成不同尺度的混沌变量,能够避免直接根据初始的定义域随机生成替代个体时效率不高的问题,且同样能够改善种群多样性。

2.3　改进 FA 的收敛性分析及复杂度分析

目前,FA 还没有很完备的数学理论基础[12-13],但已有的仿真实验结果表明 FA 具有较高的寻优精度和收敛速度,是一种有效的优化方法。本文改进算法与基本 FA 的不同之处为迭代 $T_{\max_1}$次后增加了 PCLS 过程,故只需证明 3)过程的收敛性,即可证明 PCLS-FA 的收敛性优于 FA。从测度论上进行分析,由于 PCLS 属于下降算法,并且它具有很好的遍历性,因而设 $\boldsymbol{R}_g$ 表示全局最优点 $\boldsymbol{x}^*$ 的可行域。总迭代次数为 t 时($t>T_{\max_1}$),在执行 2)后的当前最优解 $\boldsymbol{x}_{pg}$ 和次优解 $\boldsymbol{x}_{ps}$ 落入 $\boldsymbol{R}_g$ 的事件集合为 A_0,$P(A_0) \leqslant 1$,PCLS 每次迭代后产生的序列矩阵 $\boldsymbol{y}^{(l)}$ 且与 $\boldsymbol{x}_{pg}^{(l)}$ 、$\boldsymbol{x}_{ps}^{(l)}$ ($l=1,2,\cdots,C_{\max}$)合并后落入 $\boldsymbol{R}_g$ 的事件集合为 A_l,因此$A_1 \subset A_2 \subset \cdots \subset A_{C_{max}}$,概率测度单调不减,故 $P(A_{C_{\max}}) \geqslant \cdots \geqslant P(A_2) \geqslant P(A_1)$。可知执行 3)之后具有更高的概率落入全局最优点 $\boldsymbol{x}^*$ 的可行域,故 PCLSFA 的收敛性优于 FA,接下来通过对基准函数的仿真实验能够进一步验证其收敛性。此外,当忽略对目标函数的计算时,FA 的时间复杂度为 $O(T_{\max} \cdot N_{\mathrm{pop2}})$,且 PCLSFA 的时间复杂度为 $O(T_{\max} \cdot N_{\mathrm{pop2}} + T_{\max 2} \cdot C_{\max} \cdot N)$,$(T_{\max 2} = T_{\max} - T_{\max 1})$。

2.4　KHM-PCLSFA 算法流程

本文采用 K-调和均值的目标函数 KHM(X,C)作为萤火虫 i 的亮度 I_i,并以此确定其移动方向,其本质上是将聚类问题转化为一种优化问题。若 k 为聚类的数量,m 为数据的维数,则用一个 $k\times m$ 列的一维向量 $\boldsymbol{x} = (\boldsymbol{x}_{11},\boldsymbol{x}_{12},\cdots,\boldsymbol{x}_{1m},\cdots,\boldsymbol{x}_{k1},\boldsymbol{x}_{k2},\cdots,\boldsymbol{x}_{km})$ 来表示一个聚类中心,即一个萤火虫个体。由于算法对初始值不敏感,可从数据集中随机选择 k 个不同的点并对其进行较小的扰动以构成一个中心向量 $\boldsymbol{x}$,确定 P_{size}个这样的向量作为种群初始位置。由于本文算法的总迭代次数 Itermax 较小,不需要执行粗搜索。

综上所述,本文算法 KHM-PCLSFA 的流程为:

1)初始化算法的基本参数 γ、α、β、$C_{\max}$、N、l 并随机初始化萤火虫种群的位置。

2)根据萤火虫的位置计算其目标函数值作为亮度,初始化当前迭代次数 gen=0。

3)执行 PCLSFA 进行搜索,迭代运行 gen_1 次,求出当前的最优个体 $\boldsymbol{G}_{best}$ 以及对应的最优目标函数值 F_g,进入下一步操作。并且,选出占种群比例为 n_c%的最差个体并采用混沌重构法将其替换。

4)以 $\boldsymbol{G}_{best}$ 为聚类中心执行 KHM 操作,迭代运行 gen_2 次,得到目标函数值 KHM($\boldsymbol{X}$,$\boldsymbol{C}$)和聚类中心并将其转化为一维向量 $\boldsymbol{x}_{KHM}$,若 KHM($\boldsymbol{X}$,$\boldsymbol{C}$)< F_g,则用 $\boldsymbol{x}_{KHM}$ 代替 $\boldsymbol{G}_{best}$,并以 $\boldsymbol{x}_{KHM}$ 随机替换一个萤火虫。

5)gen = gen+1,若 gen<Itermax,则转到 3)继续执行,否则停止迭代得出聚类结果。

若数据集中有 n 个数据,则 KHM 每次迭代的时间复杂度为 $O(knm)$,本文聚类算法 FA 部分采用的是同步的适应度更新方式[15],故 3)中 PCLSFA 的时间复杂度为 $O(gen_1 \cdot (P_{size} \cdot (P_{size}+knm)+C_{max} \cdot N \cdot knm))$,4)中 KHM 的时间复杂度为 $O(gen_2 \cdot knm)$,并且 $P_{size}<knm$,$gen_2<gen_1 \cdot P_{size}$,因此本文算法的时间复杂度为 $O(Itermax \cdot gen_1 \cdot (P_{size}+C_{max} \cdot N) \cdot knm)$。

3 实验数据与分析

3.1 PCLSFA 的性能测试

选取了 4 个标准的无约束测试函数 $f_1 \sim f_4$[17-18]:

Ackley($x_i \in [-30, 30]$)、Rosenbrock($x_i \in [-2.048, 2.048]$)、Rastrigin($x_i \in [-5.12, 5.12]$)、Griewank($x_i \in [-600, 600]$)进行仿真测试,它们的最优解都是 0。通过 FA、采用串行 CLS 分别改进 PSO 和 FA 算法的 CLSPSO[9] 和 CLSFA 进行对比分析,以验证 PCLSFA 的收敛性能及寻优能力。各算法种群规模都为 $N_{pop}=40$,最大迭代数 $T_{max}=2\,000$,3 种具有 CLS 机制的算法中取 $C_{max}=10$,$C=5$。考虑 FA 对不同函数的收敛性能不同,在 CLSFA 和 PCLSFA 前期执行粗搜索的迭代数也不相同,对 f_1 和 f_4 取 $T_{max_1}=500$,对 f_2 取 $T_{max_1}=0$,对 f_3 取 $T_{max_1}=200$。CLSPSO 中采用线性递减的惯性权重 w[9],且 $w_{max}=0.9$,$w_{min}=0.4$,学习因子为 $c_1=c_2=1.496$。FA 型算法中统一设置 $\gamma=1$,随机步长 α 随着迭代次数 t 的增加不断减小为

$$\alpha^{t+1}=\alpha^t((10-4/0.9)^{\wedge}(b/T_{max}))$$

式中:α 的初始值为 1,b 是控制收敛精度的参数,对算法的收敛性能具有较大的影响,偏大会导致早熟收敛,偏小则会使算法无法更精确地搜索到全局最优解,经过实验对比分析本文取 $b=3$。此外,为防止距离太大使算法失效,还需对 β 进行调整,即为 $\beta=(\beta_{max}-\beta_{min})e^{-\gamma r_{ij}^2}+\beta_{min}$,其中 $\beta_{max}=1$,$\beta_{min}=0.2$。对搜索空间较小的函数 $f_1 \sim f_3$ 取 $\boldsymbol{l}_{PCLS}=0.1\boldsymbol{l}$,对搜索空间较大的函数 f_4 取 $\boldsymbol{l}_{PCLS}=0.01\boldsymbol{l}$。此外,PCLSFA 中的 $N=15$,$N_p=50$,$n_c=20$,$\phi=0.4$。

仿真实验基于 MATLAB2010b 平台,计算机的硬件配置为:Intel Core i5-4 200 M CPU 2.5 GHz、4 GB RAM。各函数的维数均为 30,每种算法独立运行 30 次,计算各自的最大值、最小值、平均值和标准差,记录至表 1。对各函数的收敛曲线为 30 次运行的平均结果,分别如图 2 所示,为了更明显的比较,图中纵坐标是对最优解求 lg(f)后的平均值。

表 1 4 个基准函数的实验结果

Table 1 The experiment results for four test functions

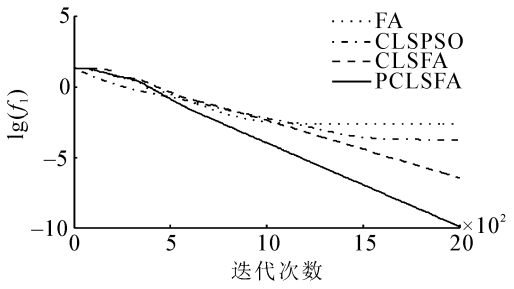
函数	算法	最小值	最大值	平均值	标准差
f_1	FA	3.763×10^{-4}	6.981×10^{-3}	3.084×10^{-3}	1.962×10^{-3}
	CLSPSO	7.994×10^{-15}	2.660 5	1.291 6	0.934 4
	CLSFA	1.139×10^{-10}	0.052 2	0.020 1	0.026 0
	PCLSFA	1.052×10^{-10}	1.496×10^{-10}	1.259×10^{-10}	1.212×10^{-11}
f_2	FA	26.575 0	29.210 0	28.306 0	0.804 6
	CLSPSO	7.919×10^{-4}	4.222 1	0.265 1	0.717 8
	CLSFA	0.030 2	32.154 2	4.076 0	9.609 3
	PCLSFA	2.208×10^{-3}	0.213 5	0.130 8	0.066 9
f_3	FA	20.895 0	47.820 0	30.911 6	7.167 8
	CLSPSO	59.697 0	112.431 0	88.551 7	12.532 2
	CLSFA	10.861 7	38.601 4	21.039 2	5.803 0
	PCLSFA	6.574 8	22.109 1	13.212 9	4.387 2
f_4	FA	3.089×10^{-5}	3.441×10^{-4}	1.030×10^{-4}	7.068×10^{-5}
	CLSPSO	7.116×10^{-14}	0.048 9	9.674×10^{-3}	0.011 7
	CLSFA	1.112×10^{-16}	3.296×10^{-4}	9.873×10^{-5}	1.508×10^{-4}
	PCLSFA	1.110×10^{-16}	5.541×10^{-16}	3.331×10^{-16}	1.655×10^{-16}

根据表 1 可见,PCLSFA 对各函数求出的最优解的平均值及标准差均为最小,表明算法具有较高的寻优精度与稳定性。虽然对 f_1 和 f_2,CLSPSO 能搜索到更佳的最小值,但相应的概率较小,从其偏大的平均值和标准差可以看出。并且,CLSFA 对于 f_1 和 f_4 能够获得的最小值与 PCLSFA 接近,但其很不稳定使其平均值相对较差,有效验证了并行 CLS 相对于串行 CLS 的优势。由图 2 可见 PCLSFA 对 f_1 和 f_4 的收敛性均取得了显著的提高,对于相对较难寻优的 f_2 和 f_3 也取得了一定的提高。因此,表 1 和图 2 中的实验结果有效验证了本文算法的收敛性。尽管 PCLSFA 对复杂函数的寻优精度方面还有待改进,

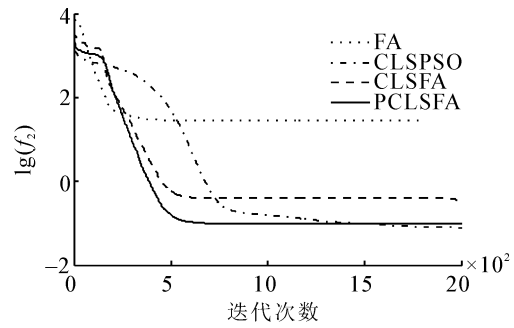
但是本文中其拥有最好的寻优能力,表明了 PCLS 机制的引入是有效的。

3.2　KHM-PCLSFA 的实验数据与分析

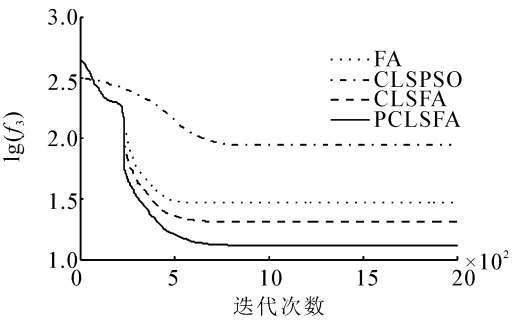
为了验证本文算法的聚类性能,选取了 UCI 数据库中的 6 个常用的数据集 Iris、Ionosphere、Wine、Image Segmentation(本文简称为 Image)、CMC 和 Satellite 进行测试,它们的特性如表 2 所示。



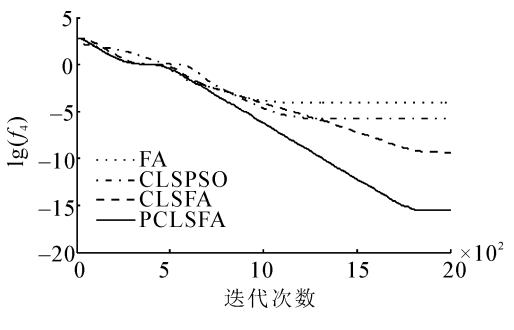
(a) f_1: Ackley



(b) f_2: Rosenbrock



(c) f_3: Rastrigin



(d) f_4: Griewank

图 2　各函数的收敛曲线

Fig.2　The convergence curves for test functions

表 2　实验数据集的特性

Table 2　The feature of experimental data set

数据集	类数	维数	数据个数
Iris	3	4	150
Ionosphere	2	33	351
Wine	3	13	178
Image	7	19	210
CMC	3	9	1473
Satellite	6	33	6435

本文以目标函数值 KHM($\boldsymbol{X},\boldsymbol{C}$)作为聚类的内部评价指标,其值越小则聚类结果越好;F-measure 作为聚类的外部评价指标,其值越大则聚类效果越好。若已知类 i 中的样本数目 n_i,簇 j 中的样本数目 n_j,以及簇 j 中属于已知类 i 的样本数目 n_{ij},则判准率为 $p(i,j)=\dfrac{n_{ij}}{n_j}$,查全率为 $r(i,j)=\dfrac{n_{ij}}{n_i}$,样本数为 n 的数据集的总体 F-measure 值为

$$F=\sum_i \frac{n_i}{n}\max_j\{F(i,j)\}$$

$$F(i,j)=\frac{2\times p(i,j)\times r(i,j)}{p(i,j)+r(i,j)}$$

分别采用 KHM、KHM-FA、KHM-PSO[3] 和本文的 KHM-PCLSFA 对几种数据集进行实验,其中 KHM-FA 与文本算法的不同体现在 2.1 节的 3)中执行的是 FA。各算法参数设置为:种群规模与文献[3]保持一致 $P_{\text{size}}=18$;KHM 的最大迭代次数 Maxgen=100,且在迭代过程中若目标函数值不再变化则停止;3 种混合聚类算法各部分迭代次数统一设置为 Itermax = 5, $\text{gen}_1=8$, $\text{gen}_2=10$。KHM-PSO 中 PSO 的相关参数为 $w=0.729\,8$, $c_1=c_2=1.496$, FA 及 PCLSFA 的相关参数为 $\gamma=1$, $\alpha=0.1$, β 同式(17)。这里为了控制 PCLS 的执行时间,取 $C_{\max}=4$, $N=6$, $C=3$, $\boldsymbol{l}_{\text{PCLS}}=0.1\boldsymbol{l}$。本文分别取 $p=2.5$、3、3.5时对聚类结果进行比较,每种算法独立运行 20 次,计算各自的 KHM($\boldsymbol{X},\boldsymbol{C}$)、F-measure 和运行时间的平均值,并分别记录在表 3~表 5 中,需注意其中数据集 Ionosphere 用 Sphere 表示。

从表 3~5 可以看出对不同特性的数据集,3 种混合聚类算法所得的 KHM($\boldsymbol{X},\boldsymbol{C}$)和 F-measure 值相对于 KHM 算法均得到了一定的改善。从 KHM($\boldsymbol{X},\boldsymbol{C}$)值的降低看出,Iris 和 Ionosphere 在 $p=3.5$ 时最大程度均降低了 0.56%;Wine 在 $p=3.5$ 时最大程度降低了 47.77%;Image 在 $p=3.5$ 时最大程度降低了 78.91%;CMC 在 $p=3$ 时最大程度降低了0.13%;Satellite

在 $p=3.5$ 时最大程度降低了 0.28%。从 F-measure 值的提升可以看出,Wine 在 $p=3$ 时最大程度提高了 5.79%;Image 在 $p=3$ 时最大程度提高了 1.63%;CMC 在 $p=3.5$ 时最大程度提高了1.10%;Satellite 在 $p=3$ 时最大程度提高了 2.23%。对于 Iris 和 Ionosphere,3 种混合算法的 F-measure 难以获得提高,这里主要通过 KHM($\boldsymbol{X},\boldsymbol{C}$)值的降低看出 3 种混合聚类算法的性能改善,其中本文算法能够获得最小的值,表现出更好的寻优能力。对于 Wine 和 Satellite,几种混合聚类算法的 F-measure 均取得比较明显的提高。虽然对于 CMC 和 Image 的 F-measure 提高有时比较有限,但是对 KHM($\boldsymbol{X},\boldsymbol{C}$)的降低取得了不错的效果,尤其是对于 Image,比如在 $p=2.5$ 时发现 KHM 算法总是会早熟收敛于 KHM($\boldsymbol{X},\boldsymbol{C}$)= $9.636\,4\times10^7$ 左右的状态,而实验中算法可获得的实际最优值在 $2.232\,6\times10^7$ 左右,这时使得其最终的均值较大,而 3 种混合算法能够有效减少陷入局部最优的次数,可以看出其均值都有较大程度的降低。

表 3 p = 2.5 时 4 种算法实验结果

Table 3 The results of four algorithms when p = 2.5

数据集	算法	KHM($\boldsymbol{X},\boldsymbol{C}$)	F-measure	时间/s
Iris	KHM	148.91	0.885	0.176
	KHM-FA	148.84	0.885	1.322
	KHM-PSO	148.89	0.885	1.227
	本文算法	148.83	0.886	2.236
Sphere	KHM	2 805.6	0.706	0.691
	KHM-FA	2 804.8	0.706	5.108
	KHM-PSO	2 805.2	0.706	5.105
	本文算法	2 803.7	0.706	9.668
Wine	KHM	75 338 585.3	0.689	0.272
	KHM-FA	75 336 750.4	0.704	2.185
	KHM-PSO	75 336 842.8	0.701	2.216
	本文算法	75 335 247.3	0.707	3.854
Image	KHM	54 906 284.3	0.595	0.921
	KHM-FA	40 937 350.7	0.597	6.658
	KHM-PSO	40 178 361.8	0.597	6.883
	本文算法	29 926 352.6	0.599	12.034
CMC	KHM	96 201.47	0.465	1.987
	KHM-FA	96 165.23	0.465	14.814
	KHM-PSO	96 185.28	0.464	14.924
	本文算法	96 160.39	0.465	26.683
Satellite	KHM	$1.954\,3\times10^8$	0.762	12.462
	KHM-FA	$1.953\,6\times10^8$	0.775	92.112
	KHM-PSO	$1.953\,7\times10^8$	0.771	99.459
	本文算法	$1.953\,3\times10^8$	0.772	175.71

此外,对于 Satellite 在 $p=3.5$ 时,KHMPSO 的目标函数值为最小,而其 Fmeasure 值却低于其他聚类算法,其中的原因值得进一步研究。经过综合对比分析,比较不同 p 值下的聚类结果可以看出,本文算法在总体上具有最佳的聚类准确性和稳定性,并且对于较复杂数据的改进效果更明显,能够有效避免陷入局部最优解。3 种混合聚类算法中都引入了群智能算法的搜索过程,因此它们的运行时间大于 KHM,并且本文算法中又引入了 PCLS 策略,使得其运行时间更长一些,这无法满足于数据规模非常大的聚类问题。在时间效率要求不是很高时适当增加 PCLSFA 的搜索次数能够进一步获得更佳的结果,并且在很多情况下算法精度方面的要求也显得更为重要。

表 4 p = 3 时 4 种算法实验结果

Table 4 The results of four algorithms when p = 3

数据集	算法	KHM($\boldsymbol{X},\boldsymbol{C}$)	F-measure	时间/s
Iris	KHM	126.08	0.892	0.181
	KHM-FA	125.86	0.892	1.296
	KHM-PSO	125.99	0.892	1.198
	本文算法	125.81	0.893	2.195
Sphere	KHM	2648.0	0.700	0.693
	KHM-FA	2643.5	0.700	5.087
	KHM-PSO	2646.5	0.700	5.067
	本文算法	2642.3	0.700	9.630
Wine	KHM	$1.049\,1\times10^9$	0.622	0.262
	KHM-FA	$1.049\,1\times10^9$	0.656	2.193
	KHM-PSO	$1.049\,0\times10^9$	0.632	2.136
	本文算法	$1.049\,1\times10^9$	0.658	3.831
Image	KHM	$1.790\,6\times10^8$	0.551	0.876
	KHM-FA	$1.783\,5\times10^8$	0.560	6.592
	KHM-PSO	$1.788\,4\times10^8$	0.559	6.703
	本文算法	$1.771\,3\times10^8$	0.560	12.210
CMC	KHM	187 018.21	0.458	1.937
	KHM-FA	186 824.39	0.461	14.885
	KHM-PSO	186 943.89	0.460	14.869
	本文算法	186 778.24	0.464	26.389
Satellite	KHM	$8.805\,4\times10^8$	0.763	12.428
	KHM-FA	$8.802\,7\times10^8$	0.776	91.250
	KHM-PSO	$8.802\,8\times10^8$	0.774	99.784
	本文算法	$8.796\,5\times10^8$	0.780	175.18

表 5　p = 3.5 时各算法实验结果

Table 5　The result of four algorithms when p = 3.5

数据集	算法	KHM($\boldsymbol{X},\boldsymbol{C}$)	F-measure	时间/s
Iris	KHM	109.84	0.892	0.178
	KHM-FA	109.53	0.892	1.304
	KHM-PSO	109.61	0.892	1.235
	本文算法	109.22	0.892	2.311
Sphere	KHM	2 567.9	0.700	0.684
	KHM-FA	2 558.9	0.700	5.096
	KHM-PSO	2 564.4	0.700	5.088
	本文算法	2 553.5	0.700	9.662
Wine	KHM	$2.717\,5\times10^{10}$	0.630	0.273
	KHM-FA	$1.420\,4\times10^{10}$	0.655	2.201
	KHM-PSO	$1.441\,9\times10^{10}$	0.636	2.140
	本文算法	$1.419\,3\times10^{10}$	0.661	3.945
Image	KHM	$7.089\,9\times10^{9}$	0.535	0.928
	KHM-FA	$2.324\,2\times10^{9}$	0.537	6.667
	KHM-PSO	$1.668\,5\times10^{9}$	0.538	6.843
	本文算法	$1.494\,6\times10^{9}$	0.540	12.162
CMC	KHM	380 733.23	0.455	1.966
	KHM-FA	380 013.07	0.458	14.892
	KHM-PSO	380 381.58	0.458	14.855
	本文算法	379 782.74	0.460	26.632
Satellite	KHM	$4.171\,4\times10^{9}$	0.715	12.437
	KHM-FA	$4.163\,0\times10^{9}$	0.721	92.174
	KHM-PSO	$4.150\,6\times10^{9}$	0.709	97.873
	本文算法	$4.159\,7\times10^{9}$	0.724	175.75

4　结束语

由于传统的 KHM 算法具有易陷于局部最优解的问题,本文基于一种高效的群智能优化算法提出了一种混合的聚类算法,在 KHM 中融合了混沌优化改进的萤火虫算法,不断优化其聚类中心。实验结果表明,本文算法的综合性能优于 KHM 以及 2 种混合聚类算法 KHM-FA 和 KHM-PSO,具有更高的聚类准确性和稳定性,能够有效地避免陷入局部最优。但是本文算法的运行时间相对比较长,在数据量较大的情况下具有较大的计算开销而影响了算法的效率,接下来可以针对算法效率的改善开展进一步研究工作。此外,可以尝试将 PCLSFA 应用于其他的优化问题中。

参考文献:

[1]JAIN A K. Data clustering: 50 years beyond K-means[J]. Pattern Recognition Letters, 2010, 31(8): 651-666.

[2]ZHANG Bin, HSU M, DAYAL U. K-harmonic means-a data clustering algorithm. Technical Report HPL-1999-124 [R]. Hewlett-Packard Laboratories, 1999.

[3]YANG Fengqin, SUN Tieli, ZHANG Changhai. 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.

[4]ALGUWAIZANI A, HANSEN P, MLADENOVIC N, et al. Variable neighborhood search for harmonic means clustering [J]. Applied Mathematical Modelling, 2011, 35(6): 2688-2694.

[5]HUNG C H, CHIOU H M, YANG Weining. Candidate groups search for K-harmonic means data clustering[J]. Applied Mathematical Modelling, 2013, 37(24): 10123-10128.

[6]汪中, 刘贵全, 陈恩红. 基于模糊 K-harmonic means 的谱聚类算法[J]. 智能系统学报, 2009, 4(2): 95-99.
WANG Zhong, LIU Guiquan, CHEN Enhong. A spectral clustering algorithm based on fuzzy K-harmonic means[J]. CAAI Transactions on Intelligent Systems, 2009, 4(2): 95-99.

[7]WU Xiaohong, WU Bin, SUN Jun, et al. A hybrid fuzzy K-harmonic means clustering algorithm[J]. Applied Mathematical Modelling, 2015, 39(12): 3398-3409.

[8]王建峰, 孙超, 姜守达. 基于粒子群优化的组合测试数据生成算法[J]. 哈尔滨工程大学学报, 2013, 34(4): 477-482.
WANG Jianfeng, SUN Chao, JIANG Shouda. Improved algorithm for combinatorial test data generation based on particle swarm optimization[J]. Journal of Harbin Engineering University, 2013, 34(4): 477-482.

[9]HE Yaoyao, YANG Shanlin, XU Qifa. Short-term cascaded hydroelectric system scheduling based on chaotic particle swarm optimization using improved logistic map[J]. Communications in Nonlinear Science and Numerical Simulation, 2013, 18(7): 1746-1756.

[10]HE Yaoyao, XU Qifa, YANG Shanlin, et al. A novel chaotic differential evolution algorithm for short-term cascaded hydroelectric system scheduling[J]. International Journal of Electrical Power & Energy Systems, 2014, 61: 455-462.

[11]廖煜雷, 刘鹏, 王建, 等. 基于改进人工鱼群算法的无人艇控制参数优化[J]. 哈尔滨工程大学学报, 2014, 35(7): 800-806.
LIAO Yulei, LIU Peng, WANG Jian, et al. Control parameter optimization for the unmanned surface vehicle with the improved artificial fish swarm algorithm[J]. Journal of Harbin Engineering University, 2014, 35(7): 800-806.

[12]YANG Xinshe. Firefly algorithm, stochastic test functions

and design optimisation[J]. International Journal of Bio-Inspired Computation, 2010, 2(2): 78-84.

[13]赵玉新, YANG X S, 刘利强. 新兴元启发式优化方法[M]. 北京: 科学出版社, 2013: 148-157.

[14]SENTHILNATH J, OMKAR S N, MANI V. Clustering using firefly algorithm: performance study[J]. Swarm and Evolutionary Computation, 2011, 1(3): 164-171.

[15]FISTER Jr I, PERC M, KAMAL S M. A review of chaos-based firefly algorithms: perspectives and research challenges[J]. Applied Mathematics and Computation, 2015, 252: 155-165.

[16]GANDOMI A H, YANG X S, TALATAHARI S, et al. Firefly algorithm with chaos[J]. Communications in Nonlinear Science and Numerical Simulation, 2013, 18(1): 89-98.

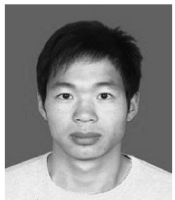
[17]YUAN Xiaofang, ZHAO Jingyi, YANG Yimin, et al. Hybrid parallel chaos optimization algorithm with harmony search algorithm[J]. Applied Soft Computing, 2014, 17: 12-22.

[18]YUAN Xiaofang, ZHANG Ting, XIANG Yongzhong, et al. Parallel chaos optimization algorithm with migration and merging operation[J]. Applied Soft Computing, 2015, 35: 591-604.

[19]YANG Dixiong, LIU Zhenjun, ZHOU Jilei. Chaos optimization algorithms based on chaotic maps with different probability distribution and search speed for global optimization[J]. Communications in Nonlinear Science and Numerical Simulation, 2014, 19(4): 1229-1246.

[20]莫愿斌, 马彦追, 郑巧燕, 等. 单纯形法的改进萤火虫算法及其在非线性方程组求解中的应用[J]. 智能系统学报, 2014, 9(6): 747-755.
MO Yuanbin, MA Yanzhui, ZHENG Qiaoyan, et al. Improved firefly algorithm based on simplex method and its application in solving non-linear equation groups[J]. CAAI Transactions on Intelligent Systems, 2014, 9(6): 747-755.

作者简介:



朱书伟,男,1990 年生,硕士研究生,主要研究方向为人工智能与模式识别。

周治平,男,1962 年生,教授,博士,主要研究方向为智能检测、自动化装置、网络安全等。

张道文,男,1989 年生,硕士研究生,主要研究方向为数据挖掘与人工智能。