文章编号：2095-2163(2022)01-0069-06 中图分类号：TP301.6 文献标志码：A

# 基于混沌剑鱼算法的 K_means 算法

唐 辉，刘晓波，韩祥民，邱 知，徐邦贤
（贵州大学 电气工程学院，贵阳 520025）

**摘 要**：传统 K-means 聚类算法容易受到初始聚类中心影响，从而导致聚类准确度较差的问题，本文利用剑鱼优化算法全局搜索能力强、收敛速度快的优势，提出一种基于改进剑鱼算法的 K-means 聚类算法。为增强剑鱼优化算法全局搜索能力，采用 Tent 混沌序列初始化种群，利用 Tent 混沌序列遍历性、随机性和规律性提高初始解的质量；为了提升算法搜索的精度，引入高斯变异，以此增强算法局部搜索能力；为了促使算法在跳出限制后继续搜索，在搜索停滞的解的基础上生成 Tent 混沌序列，用 Tent 混沌序列对部分陷入局部最优的个体进行扰动。最后，在 9 个标准测试函数上进行仿真实验，验证了所提算法的优越性；通过与传统 K_means 聚类算法在 UCI 数据集上聚类结果的对比，证明所提出的聚类算法具有更好的聚类性能，可以有效降低初始聚类中心对 K-means 算法的影响。

**关键词**：K_means 聚类算法；剑鱼算法；Tent 混沌；高斯变异；聚类中心

## K-means algorithm based on chaotic sailfish optimizer

TANG Hui, LIU Xiaobo, HAN Xiangmin, QIU Zhi, XU Bangxian
(School of Electrical Engineering, Guizhou University, Guiyang 550025, China)

**【Abstract】** For the problem that the traditional K-means clustering algorithm is easy to be affected by the initial clustering center, resulting in poor clustering accuracy, a k-means clustering algorithm based on improved Sailfish Optimizer is proposed by taking advantage of the strong global search ability and fast convergence speed of Sailfish Optimizer. In order to enhance the global search ability of Sailfish Optimizer, ten chaotic sequence is used to initialize the population, and the ergodicity, randomness and regularity of ten chaotic sequence are used to improve the quality of initial solution; In order to improve the search accuracy of the algorithm, Gaussian mutation is introduced to enhance the local search ability of the algorithm; In order to make the algorithm continue to search after jumping out of the limit, a tent chaotic sequence is generated on the basis of searching the stagnant solution, and the tent chaotic sequence is used to disturb some individuals who fall into local optimization. Finally, simulation experiments are carried out on 9 standard test functions, and the results verify the superiority of the proposed algorithm; By comparing the clustering results of traditional K-means clustering algorithms on UCI data sets, the proposed clustering algorithm has better clustering performance and can effectively reduce the impact of initial clustering center on K-means algorithm.

**【Key words】** K-means clustering algorithm; Sailfish optimizer; Tent chaos; Gaussian mutation; Cluster centers

## 0 引 言

聚类是人类一项基本的认知手段，是数据挖掘中的一个重要方法[1]。聚类被大量应用于模式识别、数据挖掘、图像分析、生物基因信息处理等领域，其目的是将一组未知分布的数据集划分为若干类，使不在同一类之间的数据相似性尽可能小，在同一类之间的数据相似性尽可能大。

11-means 算法作为经典的聚类方法，拥有快速、易行等特点，然而此方法对于初始聚类中心点的选取比较敏感，从而容易陷入局部最优解[2]。针对 K-means 容易受到初始聚类中心点影响的问题，群智能优化算法的出现解决了 K-means 算法对初始中心点过度依赖的问题[3-4]。文献[5]为避免 K-means 算法中初始聚类中心点随机选取这一问题，使用了一种基于近邻密度选取的方法；文献[6]采取万有引力优化算法（gravitational search algorithms, GSA）对 K-means 算法进行优化，来减少由于初始聚类中心点的随机或凭经验选取而导致聚类精度不高的问题。

群智能算法是一种模拟自然界中一些生物或事物行为的人工智能算法，常见算法有粒子群算法

**基金项目**：国家自然科学基金(51867005)。

**作者简介**：唐 辉(1993-)，男，硕士研究生，主要研究方向：电力系统及其自动化；刘晓波(1964-)，女，博士，副教授，主要研究方向：高电压技术、电力系统规划；韩祥民(1995-)，男，硕士研究生，主要研究方向：电气工程；邱 知(1995-)，男，硕士研究生，主要研究方向：电气工程；徐邦贤(1994-)，男，硕士研究生，主要研究方向：电气工程。

**通讯作者**：刘晓波 Email：799797284@qq.com

**收稿日期**：2021-09-22

(Particle Swarm Algorithm, PSO)、蚁狮算法(Ant Lion Optimizer, ALO)、灰狼算法(Grey Wolf Optimization, GWO)、剑鱼算法(The Sailfish Optimizer,SFO)等。其中,S. Shadravan 等于 2019 年提出的剑鱼算法(SFO)是一种新型的仿生智能优化算法[7]。SFO 无论是在收敛速度上还是在寻优能力上,都表现的很优秀,而且鲁棒性还强。但是,SFO 也存在着和其他群体智能优化算法一样的缺点,当其搜索将要达到全局最优的时候,仍然会出现种群多样性减少,容易陷入局部最优等问题。为改善群体智能优化算法,文献[8]提出将 Logistic 映射引入 PSO 算法,使得 PSO 算法跳出局部最优的能力得到提升;文献[9]和文献[10]在 GWO 算法的种群初始化时,分别使用混沌 Logistic 映射和 Tent 映射,来避免随机初始种群的缺点,从而提高算法的收敛性。

考虑到 SFO 算法和 K-means 聚类算法各自的优缺点,本文首先对 SFO 算法作出改进,优化 K-means 算法中聚类中心的位置,消除初始聚类中心的影响和陷入局部最优解的可能。

## 1 研究方法

### 1.1 K-means 聚类算法

K-means 算法目的是将集合分成若干个类别,使所属不同类对象尽可能不同,同时使所属同一类对象尽可能相同[11]。在含有 $N$ 个点的集合里,K-means 算法首先任意选取 $K$ 个点作为初始聚类中心,接着根据所有点到聚类中心点的距离大小,将其分配到距中心点最近的类别中,然后重新计算出每个类别的聚类中心,不断重复,直到聚类中心不再改变,或者聚类准则函数收敛为止[12]。

### 1.2 剑鱼算法

SFO 算法是受剑鱼捕杀沙丁鱼行为的启发所提出的智能算法,具有寻优能力强,收敛速度快的特点。该算法首先对剑鱼和沙丁鱼分别初始化,剑鱼种群位置用 $X_{SF}$ 表示,沙丁鱼种群位置用 $X_S$ 表示;初始化后,用 $X_{injuredS}$ 表示沙丁鱼中适度值最优的种群位置,用 $X_{eliteSF}$ 表示剑鱼中适应度值最优的种群位置;剑鱼位置的变动用式(1)表示:

$$X^i_{newSF} = X^i_{eliteSF} - \lambda_i * \left[rand(0,1) * \left(\frac{X^i_{eliteSF} + X^i_{injuredS}}{2}\right) - X^i_{oldSF}\right] \tag{1}$$

其中,$X^i_{injuredS}$ 表示当迭代次数为 $i$ 时,沙丁鱼所处的最优位置;$X^i_{eliteSF}$ 表示当迭代次为 $i$ 数时,剑鱼所处的最优位置;$X^i_{oldSF}$ 表示当迭代次数为 $i$ 时,剑鱼所处的位置;其中 $\lambda_i$ 的系数定义如式(2):

$$\lambda_i = 2 * rand(0,1) * PD - PD \tag{2}$$

$$PD = 1 - \frac{N_{SF}}{N_{SF} + N_S} \tag{3}$$

其中:$N_S$,$N_{SF}$ 分别代表沙丁鱼和剑鱼的数量。

沙丁鱼的位置更新如式(4)所示:

$$X^i_{newS} = (AP + X^i_{eliteSF} - X^i_{oldS}) * \gamma \tag{4}$$

其中,$AP$ 代表剑鱼攻击系数,其计算方式为公式(5);$X^i_{oldS}$ 表示当迭代次数为 $i$ 时,沙丁鱼所处的位置;$\gamma$ 为介于 0~1 之间的随机数。

$$AP = A * (1 - 2 * Itr * \varepsilon) \tag{5}$$

其中,$A$,$\varepsilon$ 控制攻击系数 $AP$ 的变换,使攻击系数 $AP$ 线性变换到 0,$Itr$ 为当前迭代次数。

当 $AP$ <0.5 时,更新沙丁鱼部分位置,沙丁鱼部分位置的范围由式(6)、式(7)确定;当 $AP$ >0.5 时,用式(8)更新沙丁鱼所有位置。

$$\alpha = N_S * AP \tag{6}$$

$$\beta = d_i * AP \tag{7}$$

其中,$\alpha$ 代表需要更新沙丁鱼的数量,$\beta$ 表示需要更新的维度数量。

如果 $f(X^i_S) < f(X^i_{SF})$,表示沙丁鱼被剑鱼猎杀,沙丁鱼的位置将被剑鱼取代,用式(8)表示:

$$X^i_{SF} = X^i_S \tag{8}$$

其中,$X^i_{SF}$,$X^i_S$ 分别表示在第 $i$ 次迭代时剑鱼和沙丁鱼的位置,$f(X^i_S)$,$f(X^i_{SF})$ 分别表示在第 $i$ 次迭代时沙丁鱼和剑鱼的适应度。

### 1.3 Tent 混沌映射

由于 Tent 混沌映射拥有随机性、规律性和遍历性的特点[13],因此被许多学者用于算法过程中的扰动或者用来产生算法的初始种群。Tent 混沌映射的数学表达式为式(9):

$$x_{i+1} = \begin{cases} \dfrac{x_i}{0.7}, & x_i < 0.7 \\ \dfrac{10}{3} \times (1 - x_i), & x_i \geqslant 0.7 \end{cases} \tag{9}$$

### 1.4 Tent 混沌扰动

为防止 SFO 算法陷入局部最优,改善全局搜索能力和寻优精度,引入 Tent 混沌扰动。步骤如下[14]:

**Step 1** 由式(9)产生混沌变量 $z_i$;

**Step 2** 用式(10)计算 $newx_i$;

$$newx_i = \min_i + (\max_i - \min_i) \times z_i \tag{10}$$

其中,$\min_i$ 和 $\max_i$ 分别表示 $newx_i$ 中第 $i$ 维变

量的最小值和最大值；

**Step 3**　用式(11)对个体产生混沌扰动。

$$newx' = \frac{x'}{2} + \frac{newx}{2} \tag{11}$$

其中，$newx$ 表示混动扰动量；$x'$ 表示需要执行混沌扰动的个体；而 $newx'$ 表示执行混沌扰动后的个体。

### 1.5　高斯变异

高斯变异源于高斯分布。高斯分布具有局部搜索能力强，鲁棒性好的特点[15]。而高斯变异是用一个符合均值为 $\mu$，方差为 $\sigma^2$ 的正态分布随机数来代替原参数值[16]。通过正态分布的特点可知，高斯变异将把原个体附近所在的局部区域当作重点搜索目标。高斯变异公式(12)为：

$$mutation(x) = x(1 + N(0,1)) \tag{12}$$

其中，$N(0,1)$ 表示符合标准正态分布的随机数；$x$ 为原来的参数值；$mutation(x)$ 为经历高斯变异后的值。

## 2　混沌剑鱼算法

考虑到 Tent 混沌搜索和高斯变异的优点，将其引入剑鱼算法，提出一种混沌剑鱼算法(Chaotic Sailfish Optimizer，CSFO)。引入的 Tent 混沌搜索和高斯变异，丰富了种群的多样性，改善了算法的搜索性能和开拓性能，避免算法陷入局部最优，算法流程如下：

**Step 1**　初始化参数：剑鱼数量 $N_{SF}$、沙丁鱼数量 $N_S$、目标函数的维数 $D$、参数 $A$、$\varepsilon$、最大迭代次数 $T_{\max}$；

**Step 2**　应用式(9)初始化剑鱼和沙丁鱼种群，分别生成 $N_{SF}$ 个 $D$ 维向量 $z_i^{sf}$ 和 $N_S$ 个 $D$ 维向量 $z_i^s$，并将其各分量用式(10)映射到原问题的空间变量的取值范围内；

**Step 3**　计算每条沙丁鱼和剑鱼的适应度值，并记录下两者的最优适应度值及其所对应位置；

**Step 4**　分别用式(1)和式(4)更新剑鱼和沙丁鱼的位置，在更新沙丁鱼位置时，当攻击系数 $AP > 0.5$ 时，更新全部沙丁鱼位置，否则更新部分沙丁鱼位置；

**Step 5**　当剑鱼猎杀到沙丁鱼时，沙丁鱼的位置将被剑鱼所占领；

**Step 6**　更新被猎杀的沙丁鱼，利用其他位置代替；

**Step 7**　当一次迭代次数完成时，重新计算每条剑鱼的适应度值 $f_i^{sf}$ 以及其平均适应度值 $f_{av}^{sf}$，当 $f_i^{sf} \geqslant f_{av}^{sf}$ 时，用式(11)对个体执行扰动，如果执行扰动后的新个体性能更优，那么就用生成的新个体替代之前的旧个体，否则不变；当 $f_i^{sf} < f_{av}^{sf}$ 时，用式(12)对个体执行高斯变异，如果生成的变异个体更优，则用变异个体代替未变异的个体，否则不变；

**Step 8**　计算所有剑鱼与沙丁鱼的适应度值，并将最优适应度值及其位置记录下来；

**Step 9**　如果达到迭代条件，则输出最优适应度值及其位置，否则转到 Step 4。

## 3　基于混沌剑鱼算法的 K-means 算法

基本思想：按照 K-means 算法的聚类原理，使用混沌剑鱼算法来优化聚类中心，以此得到不同聚类数下的聚类划分。

由于 CSFO 算法是一种随机寻优算法，不会受到初始解的干扰，而且拥有较强的全局搜索能力和较快的收敛速度，这些优点能够使其在整个搜索区域内得到使聚类目标函数尽可能小的聚类中心，能有效地减少 K-means 算法对初始中心的依赖，提高算法的聚类精度。将寻找到的全局最优位置(即聚类中心)用作 K-Means 聚类，当聚类划分整个数据集时，依据样本与聚类中心欧氏距离最近的原则进行划分。

在算法实现的过程中，有两点需要注意：

(1)使用公式(13)计算适应度函数：

$$E = \sum_{i=1}^{k} \sum_{c_i} |y - m_i|^2 \tag{13}$$

其中，$E$ 代表误差平方和；$y$ 代表数据集中的点；$m_i$ 代表每类数据 $c_i$ 的中心。

(2)初始化的过程中 CSFO 算法里每条鱼的位置是由 $k$ 个聚类中心点所组成。

算法流程如下：

**Step 1**　输入参数：样本数据总个数为 $S$，单个样本数据维数 $p$，聚类中心个数 $k$；

**Step 2**　初始化 SFO 参数：剑鱼数量 $N_{SF}$、沙丁鱼数量 $N_S$，参数 $A$、$\varepsilon$ 以及最大迭代次数 $T_{\max}$，使用式(9)和式(10)分别随机生成 $k$ 个数据(维数为 $P$)作为一条剑鱼和一条沙丁鱼，重复上述过程直到生成剑鱼 $N_{SF}$ 条，沙丁鱼 $N_S$ 条为止；

**Step 3**　采用式(13)计算适应度值，然后保存其中最优的适应度值和最优适应度所处的位置；

**Step 4**　分别用式(1)和式(4)更新剑鱼和更新沙丁鱼的位置。在使用式(4)更新沙丁鱼位置时，

当攻击系数 $AP > 0.5$ 时,更新全部沙丁鱼位置,否则更新部分沙丁鱼位置;

**Step 5** 当剑鱼猎杀到沙丁鱼时,沙丁鱼的位置将被剑鱼所占领;

**Step 6** 更新被猎杀的沙丁鱼,利用其他位置代替;

**Step 7** 当一次迭代次数完成时,重新计算每条剑鱼的适应度值 $f_i^{sf}$ 以及其平均适应度值 $f_{av}^{sf}$。当 $f_i^{sf} \geqslant f_{av}^{sf}$ 时,采用式(11)对个体执行扰动,如果经历执行扰动后的新个体更优,那么就用新个体替代扰动前的旧个体,否则不变。当 $f_i^{sf} < f_{av}^{sf}$ 时,用式(12)对个体执行高斯变异,如果变异个体更优,则代替之前未变异的个体,否则不变;

**Step 8** 计算所有剑鱼与沙丁鱼的适应度值,并将最优适应度值及其位置记录下来;

**Step 9** 是否达到迭代条件,如果达到则输出最优适应度值及其位置,否则转到 Step 4;

**Step 10** 输出最优适应度值的鱼,将其所包含的 $k$ 个样本点作为 K_means 的初始聚类中心;

**Step 11** 输出最优聚类结果。

## 4 实验仿真与分析

### 4.1 CSFO 算法的性能分析

将本文提出的 CSFO 算法与 SFO、PSO、ALO 算法相比较,使用一系列的基准函数作为测试算法性能的基准,基准函数详细表达见表 1。

**表 1 基准函数**

**Tab. 1 Benchmark function**

| 函数 | 维度 | 取值范围 | 最优值 |
|---|---|---|---|
| $f_1(x) = \sum_i^n x_i^2$ | 30 | [-100,100] | 0 |
| $f_2(x) = \sum_{i=1}^n (\sum_{j-1}^i x_j^i)^2$ | 30 | [-100,100] | 0 |
| $f_3(x) = \sum_{i=1}^n [(x_i + 0.5)]^2$ | 30 | [-100,100] | 0 |
| $f_4(x) = \frac{1}{400}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$ | 30 | [-600,600] | 0 |
| $f_5(x) = \sin^2(\pi w_1) + \sum_{i=1}^{29}(w_i - 1)^2[1 + 10\sin^2(\pi w_i + 1)] + (w_{30} - 1)^2[1 + \sin^2(2\pi w_d)]$ $w_i = 1 + \frac{x_i - 1}{4}$ | 30 | [-10,10] | 0 |
| $f_6(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$ | 30 | [-5.12,5.12] | 0 |
| $f_7(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1)$ | 4 | [-10,10] | 0 |
| $f_8(x) = \frac{1}{2}\sum_{i=1}^2 (x_i^4 - 16x_i^2 + 5x_i)$ | 2 | [-5,5] | 78.332 |
| $f_9(x) = \sum_{i=1}^5 [(x - a_i)(x - a_i)^r + c_i]^{-1}$ | 4 | [0,10] | -10.402 8 |

实验中选取种群规模 $N = 30$,最大迭代次数 $T_{\max} = 500$,$A = 4$,$\varepsilon = 0.001$,目标函数的维数 $D$ 和初始值的上下界 ub 和 lb 按照表 1 中各基准函数定义域选定,剑鱼数量 $N_{SF}$ 取种群规模的 30%,沙丁鱼数量 $N_S$ 取种群规模的 70%。为了证明 CSFO 的稳定性,降低寻优结果的偶然因素,将 9 个基准函数分别独立运行 30 次的寻优结果作为实验数据,使用每个优化算法在标准测试函数运行 30 次寻优结果的平均值和标准差作为评价标准,结果见表 2。

表 2　各基准函数寻优结果比较

Tab. 2　Comparison of optimization results of each benchmark function

| 算法 | 函数 | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ |
|---|---|---|---|---|---|---|---|---|---|---|
| PSO | 均值 | 668.67 | 612 200.36 | 5.67 | 1.04 | 1.66 | 1.35e-3 | 4.81e-2 | -76.92 | -10.14 |
| | 标准差 | 336.21 | 517 417.85 | 1.75 | 0.01 | 1.14 | 3.96e-4 | 0.11 | 4.24 | 1.41 |
| | 排名 | 4 | 4 | 3 | 3 | 3 | 3 | 2 | 3 | 3 |
| ALO | 均值 | 34.34 | 559.43 | 42.73 | 3.97 | 49.83 | 212.51 | 2.34 | -78.29 | -6.59 |
| | 标准差 | 14.14 | 929.68 | 27.23 | 5.01 | 11.73 | 30.27 | 2.3 | 0.16 | 2.97 |
| | 排名 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 2 | 4 |
| SFO | 均值 | 2.26e-9 | 2.29e-15 | 1.08 | 1.78e-10 | 4.45e-8 | 6.48e-7 | 0.39 | -74.3 | -10.4 |
| | 标准差 | 4.89e-9 | 1.23e-14 | 0.63 | 7.24e-10 | 1.86e-7 | 1.39e-6 | 1.19 | 5.66 | 3.2e-3 |
| | 排名 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 4 | 2 |
| CSFO | 均值 | 4.3e-13 | 8.11e-25 | 0.72 | 1.01e-14 | 6.34e-9 | 4.38e-11 | 9.34e-7 | -78.31 | -10.4 |
| | 标准差 | 1.11e-12 | 1.46e-24 | 0.22 | 1.87e-14 | 1.61e-8 | 6.55e-11 | 2.01e-6 | 4.25e-2 | 6.99e-4 |
| | 排名 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

表 2 的实验结果表明：对于函数 $f_1 \sim f_9$，CSFO 除了在函数 $f_3$ 上寻优结果不理想外，其余的函数无论是寻优结果的平均值上还是稳定性上都优于其他算法，而且在函数 $f_2$ 的寻优平均值和标准差值相较于 SFO 算法提升了 10 个数量级。

### 4.2　基于 CSFO 算法的 K-means 算法性能分析

实验选取改进 K-means 算法和原始 K-means 算法，分别在 UCI 数据库里的 Iris 和 Wine 两个数据集上独立重复 20 次实验的平均准确率来验证算法性能，结果见表 3。Iris 数据集共有 150 条数据，每条数据均为 4 维，分为 3 类，每类均有 50 个数据；Wine 数据集共有 178 条数据，每条数据均为 13 维，分为 3 类，各类别数目分别为 59、71、48 条。实验中改进的 K-means 算法的参数 $A = 4, \varepsilon = 0.001$，种群规模 100（其中剑鱼 30），最大迭代次数为 100。

表 3　平均准确率

Tab. 3　Average accuracy　　%

| 数据集 | 原始 K-means 算法 | 基于 CSFO 的 K-means 算法 |
|---|---|---|
| Wine | 67.74 | 71.91 |
| Iris | 85.7 | 90 |

从表 3 可以看出在 K-means 算法引入 CSFO 算法后，相比于原始的 K-means 算法，在 Wine 和 Iris 两个数据集上，分类准确率分别提高了 4.17%、4.3%。

## 5　结束语

本文提出了一种改进的剑鱼优化算法，引入 Tent 混沌搜索和高斯变异丰富了种群的多样性，改善了算法的搜索性能和开拓性能，避免算法陷入局部最优。通过与 3 种智能优化算法在 9 个基准函数的寻优实验，结果表明 CSFO 算法的寻优精度有所提升，且稳定性强。将 CSFO 算法与 K-means 算法相结合，降低了 K-means 算法对初始聚类中心选取的依赖，在 UCI 数据集上的实验说明了将 CSFO 与 K-means 结合能够提高 K-means 算法的分类的准确性，今后将对如何使用数学表达式来证明 CSFO 收敛性作进一步的研究。

## 参考文献


[1] Han Jiawei, Kamber M. 数据挖掘概念与技术[M]. 北京：机械工业出版社，2008.

[2] 曹永春，蔡正琦，邵亚斌. 基于 K-means 的改进人工蜂群聚类算法[J]. 计算机应用，2014，34(1)：204-207.

[3] ZULVIA F E, MEI C H, TSAI C Y, et al. Anapplication of a metaheuristic algorithm-based clustering ensemble method to APP customer segmentation [J].Neurocomputing, 2016, 205: 116-129.

[4] 陈小雪，尉永清，任敏，等. 基于萤火虫优化的加权 K-means 算法[J]. 计算机应用研究，2018，35(2)：466-470.

[5] Khan H S, Ahmad A. Cluster center initialization algorithm for K-means clustering[J]. Patter Recognition Letter, 2004, 25 (11): 1293-1302.

[6] Hatamlou A, Abdullah S, Nezamabadi-Pour H. A combined approach for clustering based on K-means and gravitational search algo- rithms[J]. Swarm & Evolutionary Computation, 2012, 6: 47-52.

[7] S.Shadravan, H.R. Naji, V. K. Bardsiri. The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems [J]. Engineering Applications of Artificial Intelligence, 2019, 80:80.


（下转第 79 页）