

8-13-2020

An Improved Particle Swarm Optimization Algorithm and Its Application in Clustering analysis

Chuang Wang

1. Institute of Complex System and Advanced Control Northeast Petroleum University, Daqing 163318, China;;2. Heilongjiang Provincial Key Laboratory of Networking and Intelligent Control; Daqing 163318, China;;

Zhang Yong

3. School of Electronic Science, Northeast Petroleum University, Daqing 163318, China;;

Xuegui Li

4. School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China;

Hongli Dong

1. Institute of Complex System and Advanced Control Northeast Petroleum University, Daqing 163318, China;;2. Heilongjiang Provincial Key Laboratory of Networking and Intelligent Control; Daqing 163318, China;;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research](#), [Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

An Improved Particle Swarm Optimization Algorithm and Its Application in Clustering analysis

Abstract

Abstract: In this paper, a novel artificial fish swarm particle swarm optimization algorithm (AF-PSO) is proposed corresponding to the shortcomings of the standard particle swarm algorithm including the fast convergence speed in the initial stage, the easiness to fall into premature convergence in the late, the local optimization and the poor ability to global search. *This paper firstly introduces the crowding factor δ and the Markov chain, and then adds the artificial fish swarm algorithm to the particle swarm optimization algorithm. By calculating the crowding factor, the velocity model is updated to switch among four modes: foraging, clustering, following and random.* The simulation results show that the proposed AF-PSO algorithm has better performance compared with other improved PSO algorithms in synthesis. To further illustrate the application potential, *the AF-PSO algorithm is successfully applied to the clustering analysis of oil pipeline leakage data.* Experiment results demonstrate that the performance of the AF-PSO based K-means method is better than other clustering algorithms.

Keywords

particle swarm optimization, artificial fish swarm algorithm, AF-PSO, K-means, Markov chain

Recommended Citation

Wang Chuang, Zhang Yong, Li Xuegui, Dong Hongli. An Improved Particle Swarm Optimization Algorithm and Its Application in Clustering analysis[J]. Journal of System Simulation, 2020, 32(8): 1577-1587.

改进粒子群优化算法及其在聚类分析中应用

王闯^{1,2}, 张勇^{3*}, 李学贵⁴, 董宏丽^{1,2}(1. 东北石油大学复杂系统与先进控制研究院, 黑龙江 大庆 163318; 2. 黑龙江省网络与智能控制重点实验室, 黑龙江 大庆 163318;
3. 东北石油大学电子科学学院, 黑龙江 大庆 163318; 4. 东北石油大学计算机与信息技术学院, 黑龙江 大庆 163318)

摘要: 针对标准粒子群优化算法初期收敛速度快, 后期容易陷入早熟收敛, 局部寻优, 全局搜索能力差等缺点, 提出了一种新的鱼群-粒子群优化算法(AF-PSO)。引入拥挤因子 δ 和马尔可夫链, 将鱼群算法加入到粒子群优化算法中, 通过计算拥挤因子, 来更新速度模型, 使其在觅食, 聚群, 追尾, 随机 4 种模式下进行切换。仿真结果表明了所提出的 AF-PSO 算法的综合性能优于其他改进的 PSO 算法。为进一步说明算法的实用性, 将 AF-PSO 算法成功应用于输油管道泄露数据的聚类分析上。实验结果表明基于 AF-PSO 的 K-means 算法性能是优于其他聚类算法。

关键词: 粒子群优化算法; 人工鱼群算法; 鱼群-粒子群优化算法; K-means; 马尔可夫链

中图分类号: TP273

文献标识码: A

文章编号: 1004-731X (2020) 08-1577-11

DOI: 10.16182/j.issn1004731x.joss.19-0006

An Improved Particle Swarm Optimization Algorithm and Its Application in Clustering analysis

Wang Chuang^{1,2}, Zhang Yong^{3*}, Li Xuegui⁴, Dong Hongli^{1,2}

(1. Institute of Complex System and Advanced Control Northeast Petroleum University, Daqing 163318, China; 2. Heilongjiang Provincial Key Laboratory of Networking and Intelligent Control; Daqing 163318, China; 3. School of Electronic Science, Northeast Petroleum University, Daqing 163318, China; 4. School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China)

Abstract: In this paper, a novel artificial fish swarm particle swarm optimization algorithm (AF-PSO) is proposed corresponding to the shortcomings of the standard particle swarm algorithm including the fast convergence speed in the initial stage, the easiness to fall into premature convergence in the late, the local optimization and the poor ability to global search. This paper firstly introduces the crowding factor δ and the Markov chain, and then adds the artificial fish swarm algorithm to the particle swarm optimization algorithm. By calculating the crowding factor, the velocity model is updated to switch among four modes: foraging, clustering, following and random. The simulation results show that the proposed AF-PSO algorithm has better performance compared with other improved PSO algorithms in synthesis. To further illustrate the application potential, the AF-PSO algorithm is successfully applied to the clustering analysis of oil pipeline leakage data. Experiment results demonstrate that the performance of the AF-PSO based K-means method is better than other clustering algorithms.

Keywords: particle swarm optimization; artificial fish swarm algorithm; AF-PSO; K-means; Markov chain

引言

石油是重要的能源, 日常生活, 工业生产, 航



收稿日期: 2019-01-07 修回日期: 2019-03-19;
基金项目: 国家自然科学基金(61873058), 中国博士后基金(2017M621242), 中国石油科技创新基金(2018D-5007-0302), 黑龙江省自然基金(F2018005);
作者简介: 王闯(1994-), 男, 辽宁昌图, 博士生, 研究方向为进化计算和深度学习。

天军工都需要石油作为能源和原料。经过石油分离出的汽油、沥青等化工产品广泛应用到人们日常生活中, 随着经济的快速发展, 石油的需求日益增多, 其输送问题逐渐进入人们的视线。管道作为一种使用最有效、最安全、最快捷的运输方式, 广泛应用于石油运输行业, 截至到目前, 我国输油管线总长达 20×10^4 km。但是由于管线老化, 自然灾害和

人为盗油等原因,造成了大量的石油泄漏,严重污染环境,甚至危害人的生命财产。因此,对输油管道定期检测是十分必要的,最早是人工进行巡线,人工巡线不但效率低,成本高,而且有些险峻地区人无法巡线,因此,提出了间接检测法,通过传感器采集管道数据,然后经过小波变换等操作对管道进行检测,但是操作过于繁琐,在文献[1]中提出了一种简单的基于模拟退火 K-Means 算法的石油管道工况条件聚类。

聚类分析是数据挖掘最重要的功能之一。聚类将一组样本划分为若干类,并最大化类别之间的差异,最大限度地减少同一类中的样本差异。通过使用聚类分析发现未知对象类别并从原始数据中提取有价值的信息。聚类技术已经在机器学习,模式识别和统计方法等方面得到了广泛的研究和应用。经过十多年的发展,聚类分析已经产生了大量的算法。经典的是 K-means 算法及其改进算法。然而,聚类算法严重依赖于聚类质心的初始状态^[2-3],并且可能陷入局部最优。因此,文献[1]提出模拟退火 K-Means 算法,文献[4]提出改进人工蜂群 K-means 算法,文献[5-6]提出粒子群优化算法来优化 K-means 参数。

优化技术引起了计算机科学,信息科学和通信等各种研究领域的研究兴趣。进化计算(EC, Evolutionary Computation)作为一个强大的算法家族,可以通过使用随机启发式搜索策略或元启发式搜索策略有效地用于解决全局优化问题。一些重要的 EC 方法,包括遗传算法、进化程序设计、进化规划和进化策略,都是由生物进化驱动的,并已成功应用于人工智能等各种研究领域,文献[7]中提出的粒子群优化(PSO, Particle Swarm Optimization)算法的灵感来自于模拟鸟群觅食行为。PSO 算法被认为是一种特别有竞争力的进化算法,它在处理优化问题时已经得到了广泛的应用。

1995 年,美国 James Kennedy 和 Russell Eberhart 在 IEEE 国际神经网络学术会议上发表的题为 Particle Swarm Optimization 的文章中提出粒

子群优化算法(PSO)的概念。该算法具有建模简单,收敛速度快,容易实现等优点。但是粒子群优化算法具有易早熟收敛,容易陷入局部最优,全局搜索性能差等缺陷,国内外的学者对此进行改进: Clerc 等^[8]提出了带收缩因子粒子群优化算法, Ratnaweera 等^[9]提出了时变加速因子粒子群优化算法, Zeng 等^[10]提出了切换时滞粒子群优化算法,都取得了一定的效果,但是仍没有完全解决算法容易早熟收敛的问题。

李晓磊^[11]提出了人工鱼群算法(AFSA, Artificial Fish Swarm Algorithm)。人工鱼群算法是一种自上而下的优化算法,是群智能在优化领域的又一个实际应用。经典人工鱼群算法主要包括人工鱼的 4 个基本行为:觅食、聚群、追尾和随机行为。该算法具有对搜索空间适应能力强,全局寻优性能好,鲁棒性好等优点,但是该算法也具有前期搜索盲目性大,后期收敛缓慢,很难找到高精度最优值等缺点。目前,单一的优化算法,如遗传算法,粒子群优化算法,人工鱼群算法,人工蜂群算法等很难满足人们解决某些问题的需要,因此,一些融合算法开始显示出优越性,如蜂群-粒子群优化算法^[12],粒子群-遗传算法^[13],遗传-蜂群算法^[14]等。因此,本文提出了人工鱼群-粒子群优化算法(AF-PSO),将鱼群算法的全局寻优能力强与粒子群优化算法局部寻优能力强相结合,AF-PSO 算法的提出不仅弥补了基本 AFSA 的收敛速度慢,而且克服了标准 PSO 算法难以跳出局部极值的缺点。

1 粒子群优化算法及改进

1.1 标准粒子群优化算法

粒子群优化算法是通过模拟鸟群觅食行为,群体中的每个个体通过自身学习和信息交换,来调整搜索策略,最终实现群体趋向最优位置^[10]。PSO 算法作为一种启发式智能算法,整个群体由 D 维搜索空间中移动的 S 个粒子组成。第 i 个粒子的位置可以用向量表示, $\mathbf{x}_i(k) = (x_{i1}(k), x_{i2}(k), \dots, x_{iD}(k))$

其中 $x_{i,n}(k) \in [x_{\min,n}, x_{\max,n}] (1 \leq n \leq D)$, 其中 $x_{\min,n}$ 和 $x_{\max,n}$ 分别是第 n 维的上下界, 在搜索过程中, 粒子根据以下两个因素向全局最优连续调整位置: 自身遇到的最佳位置 (p_{best}) 记为 $\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{iD})$, 在整个群体中的最优位置 ($\mathbf{g}_{\text{best}}$) 表示为 $\mathbf{p}_g = (p_{g1}, p_{g2}, \dots, p_{gD})$ 。在第 k 次迭代中第 i 个粒子的速度由 $\mathbf{v}_i(k) = (v_{i1}(k), v_{i2}(k), \dots, v_{iD}(k))$ 表示, 并且被限制在最大速度下 $\mathbf{v}_{\max} = (v_{\max,1}, v_{\max,2}, \dots, v_{\max,D})$ 。下一次迭代中粒子的速度和位置根据式(1)进行更新:

$$\begin{aligned} \mathbf{v}_i(k+1) &= \omega \mathbf{v}_i(k) + c_1 r_1 (\mathbf{p}_i(k) - \mathbf{x}_i(k)) + \\ & c_2 r_2 (\mathbf{p}_g(k) - \mathbf{x}_i(k)) \\ \mathbf{x}_i(k+1) &= \mathbf{x}_i(k) + \mathbf{v}_i(k+1) \end{aligned} \quad (1)$$

式中: k 为迭代次数; ω 为惯性权重; c_1 和 c_2 为加速因子, 代表认知系数和社会系数。 r_1 和 r_2 是 $(0, 1)$ 之间的随机数。标准粒子群优化算法存在早熟收敛, 易陷入局部最优等缺点, 针对此不足, 提出了各种改进 PSO 算法来优化传统的 PSO 算法的性能。例如, 文献[15]提出了惯性权重线性递减的 PSO 算法(PSO-LDIW), 较大的惯性权重有助于粒子跳出局部极小点, 以便于全局搜索。较小的惯性权重有助于对当前的搜索区域进行精细搜索, 以便于算法收敛。对于当前的迭代, 惯性权重计算如下:

$$\omega = (\omega_1 - \omega_2) \times (\text{iter}_{\max} - \text{iter}) / \text{iter}_{\max} + \omega_2 \quad (2)$$

式中: ω_1 和 ω_2 为惯性权重的初始值和最终值; 当前迭代次数和最大迭代次数表示为 iter 和 $\text{iter}_{\max}$ 。Ratnaweera 等^[9]提出了一个具有随时间变化的加速度系数的 PSO 算法(PSO-TVAC)。2 个加速度系数 c_1 和 c_2 计算如下:

$$c_1 = (c_{1i} - c_{1f}) \times (\text{iter}_{\max} - \text{iter}) / \text{iter}_{\max} + c_{1f} \quad (3)$$

$$c_2 = (c_{2i} - c_{2f}) \times (\text{iter}_{\max} - \text{iter}) / \text{iter}_{\max} + c_{2f} \quad (4)$$

式中: $c_{1i}(c_{2i})$ 和 $c_{1f}(c_{2f})$ 分别为加速度系数 $c_1(c_2)$ 的初始值和最终值。

为了提高搜索的性能, 文献[8]中提出了具有收缩因子的 PSO 算法(PSO-CK), 推荐惯性权重和加速度系数分别为 0.729 和 1.49。然而, 上述改进后的 PSO 算法的参数并不能自适应地改变。为了

克服上述的缺陷, 文献[16]提出了一种自适应切换粒子群优化算法(PSPO), 利用进化因子来决定整个粒子群的进化状态, 由进化因素定义 4 个状态, 从而自适应地选择惯性权重和加速度系数。此外, 文献[10]提出了一种切换时滞粒子群优化算法(SDPSO), 它利用马尔可夫链从当前状态预测下一个状态, 根据新的进化状态在不同模式之间切换来更新速度模型。同时时滞信息可以保证算法初期收敛不是过快, 防止早熟。时滞信息的加入也可以保证粒子更加充分利用更多的历史最佳粒子信息, 扩大了粒子搜索空间, 使粒子更准确进入全局最优位置, 避免陷入局部最优。

1.2 人工鱼群算法

在自然界的水域当中, 鱼类大多是群居生活, 而且为了生存, 它们往往能够找到食物充足、水草丰美的地方。模仿鱼类的这一特性, 李晓磊^[11]提出了一种新的群智能优化算法—人工鱼群算法。人工鱼群算法是一种自上而下的优化算法, 是群智能在优化领域的又一个实际应用。经典人工鱼群算法主要包括人工鱼的 4 个基本行为: 觅食、聚群、追尾和随机行为。

(1) 觅食行为: 自然界中鱼类为了生存会不断地寻找食物, 而这种觅食行为是通过身体的感官感知食物浓度从而选择游动方向。在算法中, 人工鱼 X_i 在其视野范围内随机选择一个状态 X_j , 然后比较它所在位置以及 X_i 处的食物浓度大小, 如果 Y_j 的状态比 Y_i 优, 则 X_i 向 X_j 的位置移动; 否则, X_i 重复上述行为, 当超过一定的尝试次数 try_number 后仍然不能找到食物浓度更高的状态 X_j , 则人工鱼随机选择一个状态。其数学描述为:

设人工鱼当前状态为 X_i , 其所在位置适应度值为 Y_i , 按照公式(5)感知其视野范围内的状态 X_j , 计算 Y_j 的值, 若 $Y_i < Y_j$, 则向 X_j 位置移动。

$$X_j = X_i + \text{rand} \times \text{Visual} \quad (5)$$

按照公式(6)进行位置更新, 使得 X_i 到达新的状态 X_{i+1} , 反之, 若 $Y_i > Y_j$, 则继续按照公式(5)

进行位置更新。

$$X_{i+1} = X_i + rand \times step \times \frac{X_j - X_i}{\|X_j - X_i\|} \quad (6)$$

这样反复尝试 try_number 次后, 仍然不能找到更优的状态 X_j , 则按照公式(7)进行位置更新,

$$X_{i+1} = X_i + rand \times step \quad (7)$$

(2) 聚群行为: 鱼类为了更好地适应自然环境, 大多是群居并聚群行动。在鱼群算法中, 人工鱼 X_i 在其视野范围内搜索相邻同伴的数量 n_f 及相邻同伴的中心位置 X_c , 如果 $Y_c/n_f > \delta Y_i$, 则表明 X_c 处的食物浓度较高且并不十分拥挤, 这时 X_i 就会朝 X_c 移动一段距离; 如果 $Y_c/n_f \leq \delta Y_i$, 则人工鱼执行觅食行为, 其数学描述为:

设人工鱼的此时状态为 X_i , 食物浓度值为 Y_i , 其视野范围内相邻同伴的数量 n_f 及相邻同伴中心位置 X_c , 当前人工鱼和相邻同伴之间的距离为 d_{ij} 。若存在 $d_{ij} \leq Visual$, 则表明 X_i 的视野范围内存在其他同伴, 按照公式(8)计算中心位置 X_c ,

$$X_c = \sum_{j=1}^{n_f} x_j / n_f \quad (8)$$

若满足条件 $Y_c/n_f > \delta Y_i$, 则按照公式(9)朝 X_c 移动一段距离; 如果 $Y_c/n_f \leq \delta Y_i$, 则人工鱼执行觅食行为,

$$X_{i+1} = X_i + rand \times step \times \frac{X_j - X_i}{\|X_j - X_i\|} \quad (9)$$

(3) 追尾行为: 在自然界中, 当鱼群中某个个体发现某个位置有较多的食物时, 附近的其他鱼会迅速朝这个方向游来。在算法中, 人工鱼 X_i 搜索其视野内所处位置食物浓度最高的同伴位置 X_j , 如果 $Y_j/n_f > \delta Y_i$, 表明此位置食物浓度较高且并不十分拥挤, 这时 X_i 就会朝 X_j 移动一段距离; 如果 $Y_j/n_f \leq \delta Y_i$, 则人工鱼执行觅食行为, 其数学描述为: 设人工鱼此时的状态为 X_i , 所处位置食物浓度值为 Y_i , 人工鱼 X_i 视野范围内食物浓度最大的同伴为 X_j , 如果满足 $Y_j/n_f > \delta Y_i$, 则按照公式(10) X_i 向 X_j 移动一段距离; 如果 $Y_j/n_f \leq \delta Y_i$, 则人工鱼执行觅食行为。

$$X_{i+1} = X_i + rand \times step \times \frac{X_j - X_i}{\|X_j - X_i\|} \quad (10)$$

(4) 随机行为: 人工鱼在执行觅食行为的时候, 当尝试次数超过设定的 try_number 次后, 仍然不能找到更优的状态, 就会随机选择一个方向进行移动。

2 鱼群-粒子群优化(AF-PSO)算法

本文提出了一种新的鱼群-粒子群优化算法(AF-PSO)来改善 PSO 的搜索性能。文章引入拥挤因子 δ 和马尔可夫链, 对拥挤因子进行评估, 速度更新模型将根据 δ 从一种模式切换到另一种模式。这种策略能够有效地避免 PSO 算法的早熟收敛, 对于处理多模态和高维问题特别有效。

拥挤因子 δ 由马尔可夫链 $\xi(k)(k \geq 0)$ 决定, 马尔可夫链的值取在一个限定状态空间中: $S = \{1, 2, \dots, N\}$, 马尔可夫链的转移矩阵被定义为

$$\prod^{(k)} = (\pi_{ij}^{(k)})_{N \times N} \quad (11)$$

马尔可夫概率转移矩阵定义如下:

$$\prod = \begin{pmatrix} x & 1-x & 0 & 0 \\ (1-x)/2 & x & (1-x)/2 & 0 \\ 0 & (1-x)/2 & x & (1-x)/2 \\ 0 & 0 & 1-x & x \end{pmatrix} \quad (12)$$

式中: $\pi_{ij}^{(k)} \geq 0 (i, j \in S)$, $\sum_{j=1}^N \pi_{ij}^{(k)} = 1$ 。

在 AF-PSO 算法中, 矩阵 $\prod^{(k)}$ 通过评估拥挤因子 δ 进行自适应调整, 该拥挤因子被引入来描述种群分布特性。根据搜索过程的特点, δ 可以用来定义 4 种行为: 觅食, 聚群, 追尾和随机。这 4 个行为在马尔可夫链中分别由 $\xi(k)=1$, $\xi(k)=2$, $\xi(k)=3$ 和 $\xi(k)=4$ 表示。

每个鱼之间的平均距离定义如下:

$$d_i = \frac{1}{S} \sum_{i=1}^S \sqrt{\sum_{k=1}^D (x_i^k - x_j^k)^2} \quad (13)$$

其中: S 为群体数; D 为群体维数。

鱼群之间的拥挤度即拥挤因子 δ 被定义为:

$$\delta = (d_g - d_{\min}) / (d_{\max} - d_{\min}) \quad (14)$$

式中: d_g 为全体鱼的中心位置; $d_{\max}, d_{\min}$ 为代表离中心位置的最远距离和最近距离。其中 $\zeta(k)$ 被定义如下:

$$\zeta(k) = \begin{cases} 1 & 0 \leq \delta < 0.25 \\ 2 & 0.25 \leq \delta < 0.5 \\ 3 & 0.5 \leq \delta < 0.75 \\ 4 & 0.75 \leq \delta \leq 1 \end{cases} \quad (15)$$

(1) 觅食行为 ($\zeta(k)=1$), 速度更新模型如下:

$$\begin{aligned} v_i(k+1) &= \omega v_i(k) + c_1 r_1 (p_i(k) - x_i(k)) + \\ & c_2 r_2 (p_g(k) - x_i(k)) \\ x_i(k+1) &= x_i(k) + v_i(k+1) \end{aligned} \quad (16)$$

所有的粒子(鱼)希望尽可能快的到达全局最优位置, 当所有粒子到达全局最优位置时, 粒子自身最优 p_{best} 和全局最优 g_{best} 重合, 即 $p_{\text{best}} = g_{\text{best}} = x_i(k)$ 。又因为 ω 是线性递减的, 最后速度 $v_i(k+1)=0$, 推出 $x(k+1)=x(k)$ 所有粒子都在全局最优位置。

(2) 聚群行为 ($\delta(k)=2$), 速度更新模型如下:

$$\begin{aligned} v_i(k+1) &= \omega v_i(k) + c_1 r_1 \frac{(p_i(k) - x_i(k))}{\| (p_i(k) - x_i(k)) \|} + \\ & c_2 r_2 (p_g(k) - x_i(k)) \\ x_i(k+1) &= x_i(k) + v_i(k+1) \end{aligned} \quad (17)$$

式中: $\| (p_i(k) - x_i(k)) \|$ 代表粒子当前位置和自身最优位置之间的距离。当粒子进入聚群行为, 粒子希望精细搜索自身附近空间, 由于粒子趋于全局最优, 所以 $p_{\text{best}}, g_{\text{best}}, x_i(k)$ 三者相接近, $v_i(k)=0$, 但是由于第 2 项引入距离, 当粒子接近自身最优位置时 $(p_i(k) - x_i(k))$ 趋近于 0, 粒子间距离变小, 由于距离在分母上, 所以会导致整项变大, 从而导致 $v_i(k+1)$ 不为 0, 为粒子精细搜索自身附近空间提供了速度。

(3) 追尾行为 ($\delta(k)=3$), 速度更新模型为:

$$\begin{aligned} v_i(k+1) &= \omega v_i(k) + c_1 r_1 (p_i(k) - x_i(k)) + \\ & c_2 r_2 \frac{(p_g(k) - x_i(k))}{\| (p_g(k) - x_i(k)) \|} \\ x_i(k+1) &= x_i(k) + v_i(k+1) \end{aligned} \quad (18)$$

当粒子进入追尾行为, 粒子想进一步扩大搜索的空间, 怕自己现在找到的全局最优位置不是最优的。由于粒子趋于全局最优, 所以 $p_{\text{best}}, g_{\text{best}}, x_i(k)$ 三者

相接近, $v_i(k)=0$, 但是由于第 3 项引入的距离, 当粒子接近全局最优位置时 $(p_g(k) - x_i(k))$ 趋近 0, 粒子间距离变小, 由于距离在分母上, 所以会导致整项变大, 从而导致 $v_i(k+1)$ 不为 0, 为粒子找寻全局最优空间提供了速度。

(4) 随机行为 ($\delta(k)=4$), 速度更新模型为:

$$\begin{aligned} v_i(k+1) &= \omega v_i(k) + c_1 r_1 \frac{(p_i(k) - x_i(k))}{\| (p_i(k) - x_i(k)) \|} + \\ & c_2 r_2 \frac{(p_g(k) - x_i(k))}{\| (p_g(k) - x_i(k)) \|} \\ x_i(k+1) &= x_i(k) + v_i(k+1) \end{aligned} \quad (19)$$

当粒子进入随机行为, 粒子可能陷入局部最优状态, 由于粒子趋于全局最优, 所以 $p_{\text{best}}, g_{\text{best}}, x_i(k)$ 近似认为是同一个值, 同时由于第 2 项和第 3 项引入距离, 当粒子接近最优位置时 $(p_i(k) - x_i(k))$ 和 $(p_g(k) - x_i(k))$ 趋近于 0, 粒子间距离变小, 由于距离在分母上, 所以会导致整项变大, 从而导致 $v_i(k+1)$ 不为 0, 为粒子逃离局部最优提供了速度。

AF-PSO 算法流程图如图 1 所示。

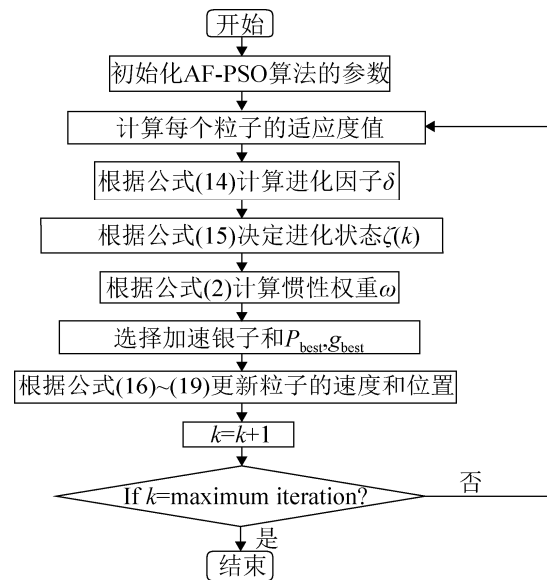


图 1 AF-PSO 算法流程图

Fig. 1 AF-PSO algorithm flow chart

3 AF-PSO 算法仿真分析

本节采用大量仿真实验来验证提出的 AF-PSO

算法比 5 种经典改进的 PSO 算法的性能是优越的。包括 PSO-LDIW^[15], PSO-TVAC^[9], PSO-CK^[8], SPSO^[16]和 SDPSO^[10]。仿实验的参数设置如下: 粒子群数 $S=20$, 群体维数 $D=20$ 。最大迭代 $N=20000$, 每个实验的重复次数 $T=50$ 。本文选取了 8 种常用的测试函数来评估 AF-PSO 和其他改进 PSO 算法的性能, 如图 2~9 所示。测试函数由方程式(20)~(27)给出, 其中 Sphere 函数是典型的单峰优化函数, 通常用于检验优化算法的收敛速度; Rosenbrock 函数可以看作是一个多峰优化函数, 它很难在狭窄的波形谷中获得最优值; 其他典型的单峰和多峰函数, 都很难找到全局最优。同时在表 1 中给出了各个测试函数的维数, 搜索范围, 阈值和极值。

$$\text{Sphere: } f_1(x) = \sum_{i=1}^D x_i^2 \quad (20)$$

Rosenbrock:

$$f_2(x) = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (21)$$

$$\text{Ackley: } f_3(x) = \sum_{i=1}^D (i \sum_{j=1}^i x_j^2) \quad (22)$$

Griewank:

$$f_4(x) = \frac{1}{4000} \sum_{i=1}^D (x_i)^2 + 1 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (23)$$

表 1 基准函数配置

Tab. 1 Benchmark function configuration

函数	名称	维数	搜索范围	阈值	极值
F1	Sphere	20	[- 100 100]	0.01	0
F2	Rosenbrock	20	[- 30 30]	100	0
F3	Ackley	20	[- 32 32]	0.01	0
F4	Griewank	20	[- 600 600]	0.01	0
F5	Penalized 1	20	[- 50 50]	0.01	0
F6	Schwefel 2.21	20	[- 100 100]	0.01	0
F7	Step	20	[- 100 100]	0.01	0
F8	Penalized 2	20	[- 50 50]	0.01	0

$$\begin{aligned} \text{Penalized1: } f_5(x) = & \frac{\pi}{D} (10 \sin^2(\pi x(i) + \\ & \sum_{i=1}^D (x(i) - 1)^2 (1 + \sin^2(10\pi x(i+1))) + \\ & (x(i) - 1)^2) + \sum_{i=1}^D u(x(i), 10, 100, 4) \end{aligned} \quad (24)$$

Schwefel2.21:

$$f_6(x) = \max |x_i|, 1 \leq i \leq D \quad (25)$$

$$\text{Step: } f_7(x) = \sum_{i=1}^D (|x_i + 0.5|)^2 \quad (26)$$

$$\begin{aligned} \text{Penalized2: } f_8(x) = & \sum_{i=1}^D 0.1 (\sin^2(3\pi x(i))) + \\ & \sum_{i=1}^{D-1} (x(i) - 1)^2 (1 + \sin^2(3\pi x(1+i))) + \\ & (x(D) - 1)^2 \times (1 + \sin^2(2\pi x(D))) + \\ & \sum_{i=1}^D u(x(i), 5, 100, 4) \end{aligned} \quad (27)$$

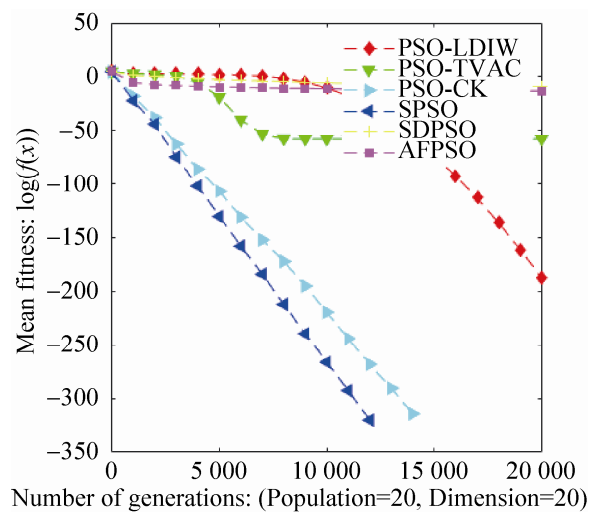


图 2 Sphere 函数性能测试

Fig. 2 Sphere function performance test

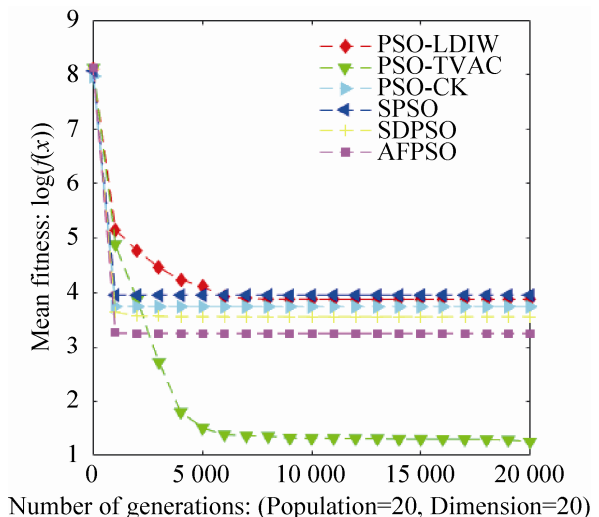


图 3 Rosenbrock 函数性能测试

Fig. 3 Rosenbrock function performance test

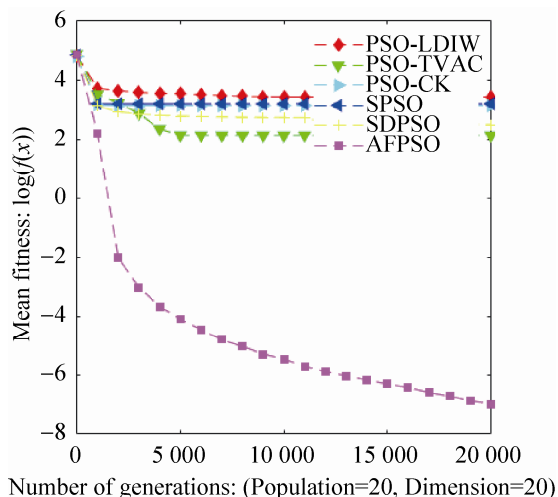


图 4 Schwefel 1.2 函数性能测试

Fig. 4 Schwefel 1.2 function performance test

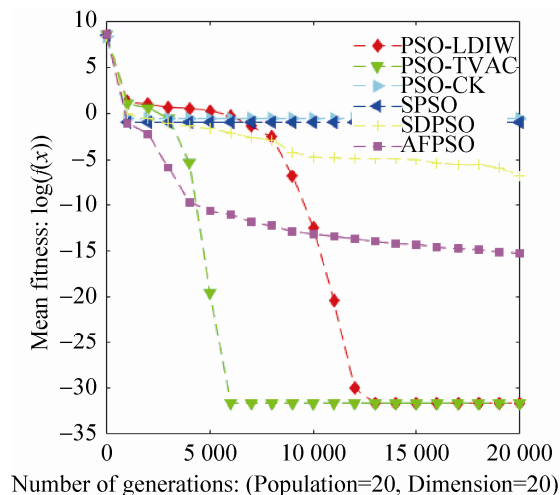


图 7 Penalized1 函数性能测试

Fig. 7 Penalized1 Function Performance Test

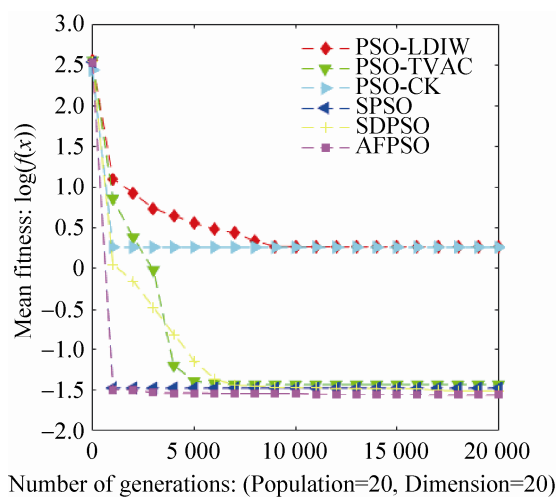


图 5 Griewank 函数性能测试

Fig. 5 Griewank function performance test

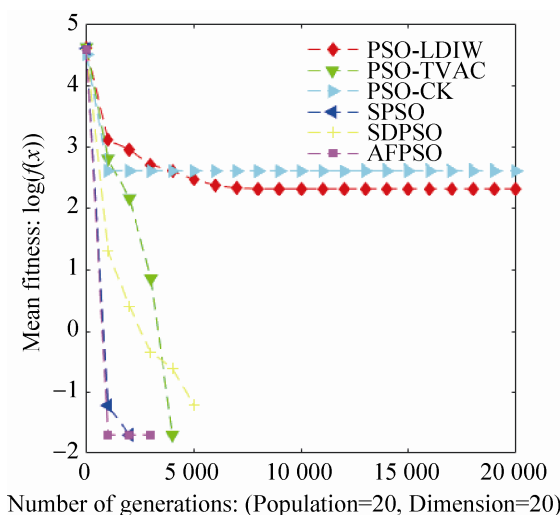


图 8 Step 函数性能测试

Fig. 8 Step function performance test

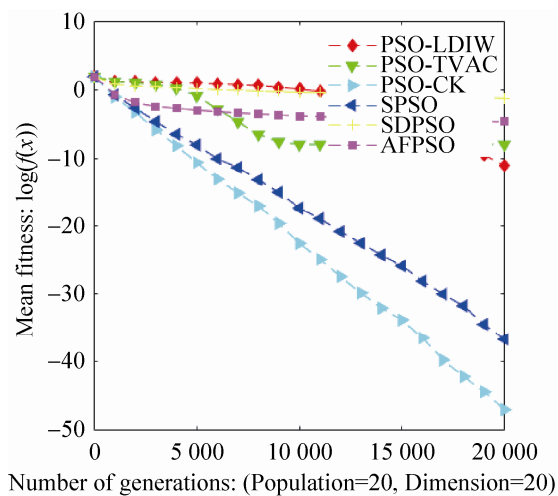


图 6 Schwefel 2.21 函数性能测试

Fig. 6 Schwefel 2.21 function performance test

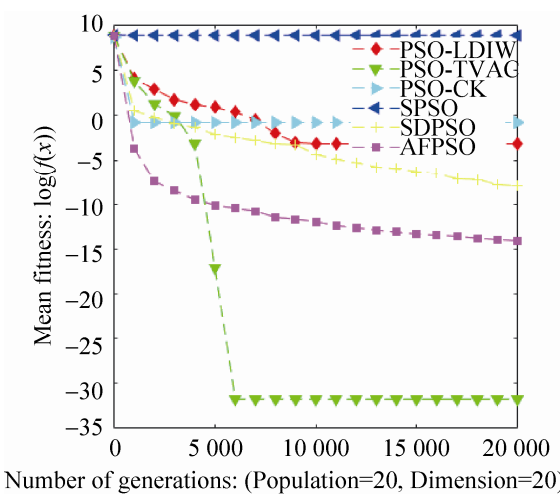


图 9 Penalized 函数性能测试

Fig. 9 Penalized function performance test

上述改进的 PSO 算法对每个测试函数的适应度值如图 2~9 所示。纵坐标代表平均适应值的对数形式，横坐标是迭代次数。另外，在表 2 中给出了各种算法的最小值、均值、标准差和收敛率几个参数。整体上看所提出的 AF-PSO 算法在均值、最小

值和标准差指标方面表现出优于其他 PSO 算的优越性。具体地，AF-PSO 算法的平均适应值较小，都是负指数，这证明了 AF-PSO 能到达全局最优解。此外，收敛率用于证明优化算法收敛性的重要指标。

表 2 测试结果统计
Tab. 2 Test results statistics

函数名称	性能指标	PSO-LDIW	PSO-TVAC	PSO-CK	SPSO	SDPSO	AF-PSO
$f_1(x)$	Minimum	2.45×10^{-202}	8.45×10^{-152}	0	0	2.71×10^{-13}	0
	Mean	1.90×10^{-188}	3.50×10^{-58}	0	0	1.08×10^{-9}	4.14×10^{-14}
	Std,Dev	0	2.47×10^{-57}	0	0	2.37×10^{-9}	1.11×10^{-13}
	Ratio/%	100%	100%	100%	100%	100%	100%
$f_2(x)$	Minimum	4.32×10^{-5}	1.69×10^{-3}	1.54×10^{-12}	1.89×10^{-3}	3.39×10^{-1}	8.16×10^{-5}
	Mean	$7.59 \times 10^{+3}$	$1.79 \times 10^{+1}$	$5.55 \times 10^{+3}$	$9.03 \times 10^{+3}$	$3.62 \times 10^{+3}$	1.28
	Std,Dev	$2.46 \times 10^{+4}$	$4.69 \times 10^{+1}$	$2.15 \times 10^{+4}$	$2.73 \times 10^{+4}$	$1.78 \times 10^{+4}$	3.41
	Ratio/%	76%	96%	82%	86%	96%	100%
$f_3(x)$	Minimum	4.32×10^{-27}	4.15×10^{-33}	2.70×10^{-104}	6.17×10^{-7}	2.42×10^{-1}	7.40×10^{-10}
	Mean	$2.57 \times 10^{+3}$	$1.33 \times 10^{+2}$	$1.33 \times 10^{+3}$	$1.53 \times 10^{+3}$	$3.05 \times 10^{+2}$	1.28
	Std,Dev	$3.31 \times 10^{+3}$	$9.43 \times 10^{+2}$	$2.49 \times 10^{+3}$	9.05×10^{-1}	$1.20 \times 10^{+3}$	3.41
	Ratio	64%	98%	76%	72%	0%	100%
$f_4(x)$	Minimum	0	0	0	0	1.89×10^{-10}	0
	Mean	1.84	3.69×10^{-2}	1.83	3.36×10^{-2}	3.08×10^{-2}	2.32×10^{-2}
	Std,Dev	$1.27 \times 10^{+1}$	2.93×10^{-2}	$1.27 \times 10^{+1}$	3.84×10^{-2}	2.48×10^{-2}	2.43×10^{-2}
	Ratio/%	12%	14%	28%	26%	22%	40%
$f_5(x)$	Minimum	2.36×10^{-32}	2.36×10^{-32}	2.36×10^{-32}	2.36×10^{-32}	1.74×10^{-13}	3.56×10^{-18}
	Mean	2.36×10^{-32}	2.43×10^{-32}	2.61×10^{-1}	1.09×10^{-1}	1.55×10^{-7}	1.24×10^{-15}
	Std,Dev	2.47×10^{-57}	4.49×10^{-33}	5.17×10^{-1}	2.11×10^{-1}	1.09×10^{-6}	4.47×10^{-15}
	Ratio/%	100%	100%	52%	72%	100%	100%
$f_6(x)$	Minimum	1.55×10^{-15}	5.31×10^{-12}	1.25×10^{-55}	9.14×10^{-43}	6.81×10^{-3}	5.32×10^{-6}
	Mean	8.16×10^{-12}	9.96×10^{-9}	1.10×10^{-47}	1.96×10^{-37}	5.86×10^{-2}	1.93×10^{-5}
	Std,Dev	5.35×10^{-11}	2.10×10^{-8}	3.92×10^{-47}	1.06×10^{-36}	4.11×10^{-2}	1.01×10^{-5}
	Ratio/%	100%	100%	100%	100%	6%	100%
$f_7(x)$	Minimum	0	0	0	0	0	0
	Mean	$2.00 \times 10^{+2}$	0	$4.01 \times 10^{+2}$	0	0	0
	Std,Dev	$1.41 \times 10^{+3}$	0	$1.98 \times 10^{+3}$	0	0	0
	Ratio/%	98%	100%	62%	100%	100%	100%
$f_8(x)$	Minimum	1.35×10^{-32}	1.35×10^{-32}	1.35×10^{-32}	2.50×10^{-32}	4.30×10^{-12}	2.94×10^{-17}
	Mean	6.59×10^{-4}	1.79×10^{-32}	1.50×10^{-1}	$6.22 \times 10^{+8}$	1.15×10^{-8}	4.29×10^{-15}
	Std,Dev	2.64×10^{-3}	2.683×10^{-32}	6.78×10^{-1}	$2.27 \times 10^{+8}$	4.08×10^{-8}	4.87×10^{-15}
	Ratio/%	94%	100%	50%	2%	100%	100%

每种改进粒子群优化算法的收敛率并非都是 100%，因为每种算法不能总是找到测试函数的全局最优，从表 2 中可以看出 AF-PSO 算法比其他 PSO 算法收敛率高。除 Griewank 函数外，AF-PSO 算法的收敛成功率为 100%。注意，Griewank 函数具有非常多局部最小值，因此难以检测到全局最优，导致收敛率非常低。我们可以看到，所有测试 PSO 算法的函数都很低，分别为 12%，14%，28%，26%，22%，36%，40%。然而，提出的 AF-PSO 算法仍然是所有算法的最佳收敛，这证明了提出算法的收敛性优于其他 PSO 算法。测试算法的收敛速度曲线如图 2~9 所示。很明显，AF-PSO 算法的收敛速度是优于其它算法，因此，AF-PSO 算法有令人满意的收敛速度和收敛精度来解决优化问题。

4 基于 AF-PSO 算法的 K-means 算法在输油管道泄露数据应用及分析

在石油领域，管道广泛用于石油运输。由于自然因素、人为因素等管道运输中很容易造成漏油。管道泄漏造成的损失和危害使人们不得不对输油管道泄漏检测注意。因此，石油管道泄漏检测技术的研究已成为国内外工业界和学术界研究的热点。目前，石油管道泄漏检测已有成熟的方法，但随着工业技术的发展，石油管道泄漏检测准确性要求也在增加。因此，对输油管道泄漏检测技术提出了更高的要求。本节采用轮廓聚类验证方法，对提出的基于 AF-PSO 的聚类算法的性能进行评估。实验的数据来自东北石油大学复杂系统与先进控制研究院的油气管道泄漏检测仿真实验平台。共 470 组 23 500 个数据，包括 195 组正常数据 9 750 个，泄漏数据 13 750 个。实验参数设置如下：粒子种群规模 $S=20$ ，最大迭代 $N=1000$ ，惯性权重线性递减 $\omega[0.9, 0.5]$ ，加速因子 $c_1[2.5, 0.5]$ ， $c_2[0.5, 2.5]$ 。

4.1 K-means 算法

聚类问题可以描述为：给定一个样本数为 n 的数据集 $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ ，其中， $i = 1, 2, \dots, n$ ，

将其化为 k 个类，即

$$X = C_1 \cup \dots \cup C_k \cup C_{\text{outliers}} \quad (28)$$

式中： $C_1 \dots C_k$ 为所分的类别，如果群集不重叠，则： $C_i \cap C_j = \emptyset$ ，K-means 是最常用的聚类算法。它通常使用欧氏距离测量，由式(29)给出：

$$Dis = \sqrt{\sum_{i=1}^d (x_i - y_i)^2} \quad (29)$$

式中： x_i 和 y_i 是 d 维欧氏空间中的 2 个点。本文使用均方误差(MSE)作为聚类的目标函数，由式(30)给出：

$$MSE = \frac{1}{n} \sum_{j=1}^k \sum_{y_i \in C_j} \|y_i - z_j\|^2 \quad (30)$$

式中： z_j 为聚类中心。

4.2 基于 AF-PSO 算法的 K-means 算法

针对 K-means 算法严重依赖质心的初始状态和易陷入局部最优缺点，提出了基于 AF-PSO 算法的 K-means 算法，利用 AF-PSO 算法来优化质心的初始状态，因为 AF-PSO 是一种启发式智能优化算法，不受初始状态干扰，能够有效搜索整个空间找到使 MSE 尽可能小的值，达到最优的聚类效果。

算法步骤如下：

step 1: 对粒子群进行初始化操作，从样本中随机选取 K 个中心点，将其作为粒子位置的初值，同时，初始化粒子的速度 v_i ，个体最优 p_{best} ，全局最优 g_{best} 。

step 2: 根据式(29)判断每个数据点归属的聚类中心。

step 3: 根据式(30)计算每个粒子的适应度值。

step 4: 将每个粒子的适应度值与粒子的最好位置比较，如果更优，则将当前值作为粒子最好位置。比较当前所有的 p_{best} 和 g_{best} ，更新 g_{best} 。

step 5: 根据式(16)~(19)调整所有粒子的移动速度和位置。

step 6: 重复执行 step 2~step 7，直到满足终止条件。若终止条件成立，则输出每一类别的聚类中心坐标。

4.3 基于 AF-PSO 算法的 K-means 算法在输油管道泄露数据应用及分析

仿真实验软件使用 LABVIEW 开发环境, 使用 NI-9215 采集卡收集数据。采样频率为 5 kHz, 压电声波传感器用于将数据传输到计算机。仿真实验数据主要包括正常数据和泄漏数据。通过快速切换 4 点球阀开关来模拟管道泄漏, 获得泄漏数据。在泄漏点安装一个 10 m 的高压声波衰减器, 在末端安装一个 4 点球阀和一个漏孔为 1 mm 的插头, 以减少切换开关阀对数据自身造成的振动的影响衰减。

Silhouette^[17]提出的一种流行的聚类验证方法。为验证所提出的基于 AF-PSO 算法的 K-means 算法的聚类性能, 本文比较了基于 AF-PSO 算法的 K-means 算法与基于 PSO 算法 K-means 算法的轮廓系数。基于 PSO 算法的 K-means 算法和基于 AF-PSO 算法的 K-means 算法轮廓图见图 10~11。

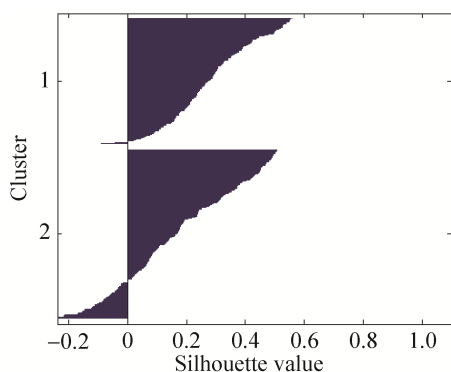


图 10 基于 PSO 算法的 K-means 算法轮廓系数图
Fig. 10 Contour coefficient diagram of k-means algorithm based on PSO algorithm

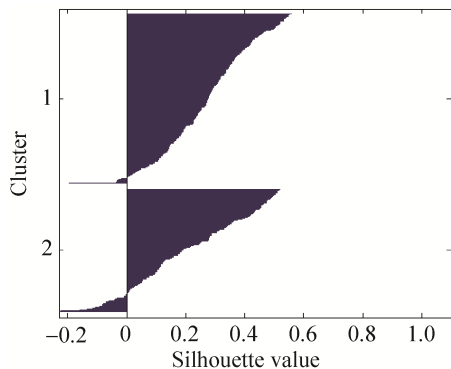


图 11 基于 AF-PSO 算法的 K-means 算法轮廓系数图
Fig. 11 Contour coefficient diagram of k-means algorithm based on AF-PSO algorithm

它们的平均轮廓系数为 0.223 1 和 0.248 2。本文可以看到, 在图 10~11 中, 2 种算法的绝大多数轮廓值都是正的, 这表明大多数数据被分配给适当的类别。然而, 在图 11 中基于 AF-PSO 算法的 K-means 算法的平均轮廓值是 0.248 2, 其高于基于 PSO 算法的 K-means 算法。此外, 使用基于 AF-PSO 的聚类算法的负轮廓值比基于 PSO 算法的 K-means 算法更少, 表明较少的数据点被分配给不适当的聚类。最后, 泄露数据是 13 750 个, 正常数据 9 750 个, 比大约是 1.41:1, 图 11 轮廓比符合 1.41:1。因此, 证明了所提出的基于 AF-PSO 算法的 K-means 算法的优越性和可行性, 以及生成分类的合理性。

5 结论

本文提出了一种新的 AF-PSO 算法, 将提出的算法优化 K-means 参数, 并成功应用于输油管道泄漏检测数据的聚类问题上。AF-PSO 算法的速度更新模型根据进化状态自适应地调整。值得一提的是, 在速度更新模型中引入拥挤因子和马尔可夫链, 通过对拥挤因子进行评估, 速度更新模型从一种模式切换到另一种模式, 这种策略充分利用鱼群算法的全局搜索能力和粒子群的局部搜索能力的优点, 通过这种策略, AF-PSO 可以更彻底地探索整个搜索空间, 从而更有可能跳出局部最优并达到全局最优解。仿真结果表明了所提出的 AF-PSO 算法的综合性能优于其他改进的 PSO 算法。最后, AF-PSO 算法用于优化 K-means 并成功应用于石油管道泄漏检测数据的聚类。通过将平均轮廓值与 PSO-K-means 聚类算法进行比较, 证明了所提出的基于 AF-PSO 的聚类算法的优越性和实用性。

参考文献:

- [1] Yingchun Y, Laibin Z, Wei L, et al. Oil pipeline work conditions clustering based on simulated annealing k-Means algorithm[C]. Computer Science and Information Engineering, 2009 WRI World Congress on. IEEE, 2009, 3: 646-650.
- [2] 李春生, 王耀南. 聚类中心初始化的新方法[J]. 控制理论与应用, 2010, 27(10): 1435-1440.

- Li Chunsheng, Wang Yaonan. A new method for cluster center initialization[J]. Control Theory and Applications, 2010, 27(10): 1435-1440.
- [3] 钱线, 黄莹菁, 吴立德. 初始化 K-means 的谱方法[J]. 自动化学报, 2007, 33(4): 342-346.
- Qian Xian, Huang Xuanjing, Wu Lide. Initialization of K-means Spectral Method[J]. Acta Automatica Sinica, 2007, 33(4): 342-346.
- [4] 于佐军, 秦欢. 基于改进蜂群算法的 K-means 算法[J]. 控制与决策, 2018, 33(1): 181-185.
- Yu Zuojun, Qin Huan. K-means algorithm based on improved bee colony algorithm[J]. Control and Decision, 2018, 33(1): 181-185.
- [5] 杨广全, 朱昌明, 王向红, 等. 基于粒子群 K 均值聚类算法的电梯交通模式识别[J]. 控制与决策, 2007, 22(10): 1139-1142.
- Yang Guangquan, Zhu Changming, Wang Xianghong, et al. Elevator traffic pattern recognition based on particle swarm K-means clustering algorithm[J]. Control and Decision, 2007, 22(10): 1139-1142.
- [6] 刘衍民, 隋常玲, 赵庆祯. 基于 K-均值聚类的动态多种群粒子群算法及其应用[J]. 控制与决策, 2011, 26(7): 1019-1025.
- Liu Yanmin, Sui Changling, Zhao Qingzhen. Dynamic multi-population particle swarm optimization algorithm based on K-means clustering and its application[J]. Control and Decision, 2011, 26(7): 1019-1025.
- [7] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory[C]. Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on. IEEE, 1995: 39-43.
- [8] Clerc M, Kennedy J. The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space[J]. IEEE Transactions on Evolutionary Computation (S1089-778X), 2002, 6(1): 58-73.
- [9] Ratnaweera A, Halgamuge S K, Watson H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[J]. IEEE Transactions on Evolutionary Computation (S1089-778X), 2004, 8(3): 240-255.
- [10] Zeng N Y, Wang Z D, Zhang H, et al. A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay[J]. Cognitive Computation (S1866-9956), 2016, 8(2): 143-152.
- [11] 李晓磊. 一种新型的智能优化方法-人工鱼群算法[D]. 杭州: 浙江大学, 2003.
- Li Xiaolei. A new intelligent optimization method-artificial fish swarm algorithm [D]. Hangzhou: Zhejiang University, 2003.
- [12] 匡芳君, 金忠, 徐蔚鸿, 等. Tent 混沌人工蜂群与粒子群混合算法[J]. 控制与决策, 2015, 30(5): 839-847.
- Kuang Fangjun, Jin Zhong, Xu Weihong, et al. Tent chaotic artificial bee colony and particle swarm hybrid algorithm[J]. Control and Decision, 2015, 30(5): 839-847.
- [13] Shi X H, Liang Y C, Lee H P, et al. An improved GA and a novel PSO-GA-based hybrid algorithm[J]. Information Processing Letters (S0020-0190), 2005, 93(5): 255-261.
- [14] 胡玉霞, 马留洋, 张锐, 等. 基于遗传蜂群算法的运动想象 BCI 系统导联选择[J]. 计算机应用研究, 2018, 35(8): 2374-2378.
- Hu Yuxia, Ma Liuyang, Zhang Rui, et al. Selection of lead in motion imaging BCI system based on genetic colony algorithm[J]. Journal of Computer Applications, 2018, 35(8): 2374-2378.
- [15] Shi Y H, Eberhart R C. Parameter selection in particle swarm optimization[C]. International conference on evolutionary programming. Berlin: Springer, 1998: 591-600.
- [16] Tang Y, Wang Z D, Fang J. Parameters identification of unknown delayed genetic regulatory networks by a switching particle swarm optimization algorithm[J]. Expert Systems with Applications (S0957-4174), 2011, 38(3): 2523-2535.
- [17] Rousseeuw P J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis[J]. Journal of Computational and Applied Mathematics (S0377-0427), 1987, 20: 53-65.