

# 一种自适应变异二进制粒子群算法

姜磊<sup>1,2</sup>, 刘建华<sup>1,2</sup>, 张冬阳<sup>1,2</sup>, 卜冠南<sup>1,2</sup>

(1.福建工程学院 信息科学与工程学院,福建 福州 350118;

2.福建省大数据挖掘与应用技术重点实验室,福建 福州 350118)

**摘要:** 针对二进制粒子群算法(BPSO)具有过早收敛的缺陷,在粒子位置更新后提出变异概率自适应从大到小的变异操作。同时对算法惯性权重参数采用递增的设置方案,从而得到一种自适应变异BPSO算法(AMBPSO),将其应用于特征选择问题。实验结果表明,提出的新算法前期具有较强的全局搜索能力,后期具有较强的局部搜索能力,能使平均选择特征数量最多从27.6个减少到20.2个,平均分类准确率最多从91.346%提升到94.135%。

**关键词:** 数据预处理;特征选择;二进制粒子群算法;自适应变异;惯性权重

中图分类号: TP301.6

文献标志码: A

文章编号: 1672-4348(2020)03-0273-07

## An adaptive mutation binary particle swarm optimization algorithm

JIANG Lei<sup>1,2</sup>, LIU Jianhua<sup>1,2</sup>, ZHANG Dongyang<sup>1,2</sup>, BU Guannan<sup>1,2</sup>

(1.School of Information Science and Engineering, Fujian University of Technology, Fuzhou 350118, China;

2.Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fuzhou 350118, China)

**Abstract:** Aiming at the defect of premature convergence of the binary particle swarm algorithm (BPSO), a mutation operation with the adaptation of the mutation probability going from large to small was proposed after particle position updating. An incremental setting scheme was adopted for the inertia weight parameters of the algorithm to obtain an adaptive mutation BPSO algorithm (AMBPSO), which was applied to the feature selection problem. Experimental results show the proposed new algorithm has strong global search ability in the early stage, and has strong local search ability in the later stage, and it can reduce the average number of selected features from 27.6 to 20.2 and increase the average classification accuracy from 91.346% to 94.135%.

**Keywords:** data preprocessing; feature selection; BPSO algorithm; adaptive mutation; inertia weight

## 引言

在数据分析中,特征选择是一种重要的数据预处理技术。常见的特征选择方法包括过滤式、包裹式、嵌入式<sup>[1]</sup>。特征选择问题属于NP难问题,其包裹式特征选择方法通常是将智能算法用于特征选择。在智能算法中,二进制粒子群算法(BPSO)具有规则简单、参数设置较少等优点,从而被用到特征选择的问题中。

文献[2]提出了基于二进制粒子群优化的水下目标特征选择算法,并结合k近邻分类算法对3类实测水下目标数据进行了最优特征集的选择及分类实验。文献[3]提出基于离散粒子群算法进行织物疵点特征选择的方法。文献[4]提出了一种基于粒子群优化的入侵特征选择算法,通过分析网络入侵数据特征之间的相关性,可使粒子群优化算法在所有特征空间中优化搜索,自主选择有效特征子集,降低数据维度。

收稿日期: 2020-01-08

基金项目: 福建省自然科学基金项目(2019J01061137); 福建工程学院发展基金(GY-Z17150)

第一作者简介: 姜磊(1991—),男,河南周口人,硕士研究生,研究方向:智能计算,机器学习。

通信作者: 刘建华(1967—),男,江西安福人,教授,博士,研究方向:智能计算、大数据分析、物联网技术。

特征选择一般采用原始的 BPSO 算法,文献[5]从算法的位置改变概率以及遗传算法的模式定理方面对 BPSO 算法进行了分析,发现 BPSO 算法存在搜索能力不足的缺点。文献[6]针对 BPSO 算法提出了改进方法,并验证了改进方法的有效性。文献[7]提出基于特征聚类信息进行种群的初始化策略提高初始化种群的质量,并提出了一种基于决策空间相似性的自适应局部搜索策略,避免算法早熟。文献[8]提出一种鲶鱼效应引入新的粒子来提高 BPSO 算法性能,并用于特征选择。文献[9]提出一种新改进的 BPSO-SVM 算法用于特征选择。

综上所述,BPSO 算法存在过早收敛问题,其权重设置存在不合理的问题。为了将 BPSO 算法更好地用于特征选择,本文研究在粒子位置更新后提出了一种自适应变异操作,同时对算法惯性权重参数采用一种递增的设置方案,从而得到一种自适应变异 BPSO 算法(AMBPSO)。

## 1 相关工作

群智能优化算法源于模仿自然界中生物群体的行为模式,适用于解决大规模复杂性问题。如今,要处理的数据规模越来越大,数据中冗余的特征数量也越来越多,特征选择即成为大规模的 NP 难问题。所以,群智能优化算法适合特征选择问题的处理,尤其在包裹式的特征选择方面,大量的群智能优化算法被应用,例如遗传算法、蚁群算法、蝗虫优化算法、蜻蜓算法等<sup>[10-13]</sup>。群智能优化算法在处理特征选择问题时以生物个体作为特征选择的一个候选解,以分类准确率或分类错误率作为评价函数来引导群体生物中生物个体向群体最优位置移动,从而得到最优解。BPSO 算法的原理、参数设置相对简单,理论成熟度上比较成熟。但是同其他群智能算法一样,BPSO 算法也易陷入局部最优的求解问题。一般的启发式搜索算法要求前期要具有全局搜索能力,后期要具有局部搜索能力,对于 BPSO 算法的改进也需要遵循这个原则,平衡算法的搜索能力。

## 2 基于自适应变异 BPSO 的特征选择

### 2.1 BPSO 算法

BPSO 算法是 Kennedy 于 1997 年在连续性 PSO 算法基础上提出的,用于解决离散的优化

问题<sup>[14]</sup>。

任意粒子  $i(i = 1, 2, \dots, N)$  根据公式(1)更新速度:

$$v_{id} = w \times v_{id} + c_1 \times \text{rand}() \times (p_{id} - x_{id}) + c_2 \times \text{rand}() \times (p_{gd} - x_{id}) \quad (1)$$

式中,  $id$ 、 $gd$  表示坐标变量,  $v_{id}$  表示粒子速度,  $c_1$ 、 $c_2$  表示学习因子,  $\text{rand}()$  表示 0 ~ 1 之间的随机数。  $p_{id}$  表示个体最优粒子位置,  $p_{gd}$  表示群体最优粒子位置。  $x_{id}$  表示粒子位置,  $w$  表示惯性权重, 衡量粒子学习过程中上一代粒子对当代粒子的影响权重, 受 PSO 算法的影响, BPSO 的权重通常采用随机迭代递减的方法。 BPSO 算法用于解决离散空间中的优化问题, 基于二进制编码原则, 算法中粒子位置向量每一维取值只能为 0 和 1。 Kennedy 提出利用 Sigmoid 函数作为转换函数, 将连续空间中的位置向量的每一维映射到二进制编码中的 0 或者 1。 通过公式(2)计算对应的 Sigmoid 函数值:

$$s(v_{id}) = \frac{1}{1 + e^{-v_{id}}} \quad (2)$$

式中,  $v_{id}$  表示粒子速度,  $s(v_{id})$  表示位置  $x_{id}$  取 1 的概率, 粒子通过(3)式改变它的位值:

$$x_{id} = \begin{cases} 1 & \text{if } \text{rand}() \leq s(v_{id}) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

其中,  $\text{rand}()$  是 0~1 之间的随机数。

### 2.2 BPSO 算法的自适应变异操作

BPSO 由于在汉明空间中变化, 存在容易预收敛等问题。公式(2)特性造成当速度  $v_{id}$  很大时,  $s(v_{id})$  很接近 1, 且恒为 1, 致使  $x_{id}$  不易改变, 算法过早收敛; 反之当速度  $v_{id}$  很小, 也易收敛于 0。 综上, BPSO 易于过早收敛。 为了提高 BPSO 算法的多样性, 克服过早收敛问题, 文献[15]对 BPSO 算法提出了一种变异手段, 粒子在(3)式更新位置后引入了变异操作如式(4)。

$$x_{id} = \begin{cases} \sim x_{id} & \text{if } \text{rand}() \leq r_{mut} \\ x_{id} & \text{otherwise} \end{cases} \quad (4)$$

式中,  $r_{mut}$  代表变异概率, 且  $r_{mut} = 1/N$ ,  $N$  为数据维度, 也就是对每个粒子每位都会产生概率为  $r_{mut}$  的变异可能。

但是式(4)的  $r_{mut}$  在算法迭代过程中为定值, 缺少对 BPSO 算法缺陷的针对性。一般启发式算

法前期需要较强的全局搜索能力,后期需要突出局部的搜索能力。基于上述原理,BPSO算法早期变异率要大,晚期变异要小。因此引进如式(5)变异操作。

$$x_{id} = \begin{cases} \sim x_{id} & \text{if rand}() \leq R \\ x_{id} & \text{otherwise} \end{cases} \quad (5)$$

式中, $R$ 表示变异概率,但不是固定值,采用一种递减策略,如式(6):

$$R = \left(1 - \frac{t}{T}\right)^2 \quad (6)$$

式中, $t$ 为当前算法迭代的代数, $T$ 为算法迭代的最大代数。

算法采用式(3)更新后,执行由式(5)(6)组合的变异操作,表示在产生新的粒子时,每一个粒子每位产生一定变异概率。根据分析,式(6)的变异概率 $R$ 值是从1到0,即早期几乎都会发生变异,而晚期几乎不发生变异,变异概率随着代数的增加自适应从1到0变化,而式(4)中 $r_{\text{mut}}$ 固定值表示每次变异概率都只是为 $1/N$ (维度分之一)。

### 2.3 BPSO的惯性权重设置

受到PSO算法影响,BPSO算法也存在惯性权重 $w$ ,其设置采用式(7)的线性递减方案,

$$w = w_{\text{max}} - \frac{(w_{\text{max}} - w_{\text{min}}) \times t}{T} \quad (7)$$

式中, $w_{\text{max}}$ 表示惯性权重最大值, $w_{\text{min}}$ 表示惯性权重最小值, $t$ 表示当前的迭代次数, $T$ 表示最大的迭代次数。

文献[16]通过理论分析研究了惯性权重 $w$ 的设置对算法的影响,认为算法应当在初期有很强的全局探测能力,逐步到后期的时候具有很强的局部探索能力。保持其他参数不变时,较小的惯性权重能够增强全局探测能力较大的惯性权重能够增强局部的探索能力。本研究采用惯性权重线性递增的方案如式(8)。

$$w = \begin{cases} w_{\text{min}} + \frac{(w_{\text{max}} - w_{\text{min}}) * t}{\rho T} & t \leq \rho T \\ w_{\text{max}} & \rho T \leq t \leq T \end{cases} \quad (8)$$

式中, $\rho$ 取0.9,意味着权重变化迭代到最大代数的90%代就结束了。此后的迭代中,权重为固定的值。

PSO算法中,惯性权重 $w$ 一般采用式(7)的

递减策略, $w$ 值从0.9递减到0.4;传统的BPSO采用类似的方案。本研究将式(8)的递增方案引入BPSO算法中, $w_{\text{max}}$ 和 $w_{\text{min}}$ 的取值方法将在试验方法中确定。

### 2.4 kNN算法

将BPSO应用于特征选择,需要一个监督学习器评价其特征选择的适用度函数。最近邻近算法(k-Nearest Neighbor algorithm, kNN)是一种常用的监督学习方法<sup>[17]</sup>,其工作机制简单理论上比较成熟,易于快速实现。其基本原理是:给定测试样本,基于某种距离度量找出训练集中与其最靠近的 $k$ 个训练样本,然后基于这些样本的信息来预测,选择 $k$ 个样本中出现最多的类别标记作为预测结果。本研究采用kNN算法对数据进行分类判别,作为选取数据特征子集的评价函数, $k$ 取5。

### 2.5 带自适用变异操作BPSO的特征选择算法流程

带自适用变异操作的BPSO算法(AMBPSO)在原始BPSO算法执行式(3)位置更新后,再采用式(5)(6)变异操作,并辅以式(8)的权重设置方案,最后将特征选择结果应用于kNN构造分类准确率评价中,因此产生了AMBPSO特征选择算法。其流程包括AMBPSO特征选择算法流程和5-NN算法流程。

#### AMBPSO特征选择算法流程

- 1 开始
- 2 初始化粒子群,粒子维度  $\text{dim}$  = 样本数据维度,种群数量  $\text{popsize} = 20$
- 3  $\text{popul} = \text{rand}(\text{popsize}, \text{dim}) < 0.5$
- 4 记录初始个体最优  $\text{bestpos} = \text{popul}$
- 5 计算各粒子适应度值  $\text{fitvalue}$  by 5-NN()
- 6 记录最佳适应度值  $\text{fbestpos}$ , 以及对应粒子位置  $g$
- 7 While(停止准则为最大迭代次数)
- 8 更新个体最优
- 9 利用(1)式更新粒子速度
- 10 利用(2)和(3)式更新粒子的位值
- 11 利用(5)和(6)式对粒子位值进行变异
- 12 计算新的各粒子适应度值 by 5-NN()
- 13 与现有粒子适应度值作比较,更新全局最优,记录最佳适应度值及更新对应粒子位置  $g$
- 14 End(直到满足最大迭代次数)
- 15 输出最佳适应度值即为分类准确率

### 5-NN 算法流程

```

1 for i=1 to 测试数据的样本数量
2   for j=1 to 训练数据的样本数量
3     for m=1:popsize
4       特征选择方案: A(m) = popul(m, :)
5       if A(1, dim) = 0
6         distij = 0
7       else
8         distij = distij + (dataik - datajk)2
9       end if
10      next m
11     next j
12    对 distij 进行从小到大排序记录其对应类别
13    取出前 k(k=5) 个 distij 及其对应类别
14    对 k 个样本所属类别个数进行统计
15    选出出现次数最多的类别即为第 i 个测试样本的类别 classi
16  next i
17 correct = 0
18 for i=1 to 分类问题的样本数
19   if classi = 第 i 个测试样本 then
20     correct = correct + 1
21   end if
22 next i
23 适应度值 = correct / 测试数据样本数量

```

## 3 实验与结果分析

### 3.1 相关参数设置与原始数据集

在 PSO 算法中学习因子  $c_1$ 、 $c_2$  均取 2, 种群数量为 20, 最大迭代次数为 1 000, kNN 算法中的  $k$  值取 5。用于测试数据为 8 种不同类型的数据集, 这些数据都是从 UCI 机器学习数据库下载 (<https://archive.ics.uci.edu/ml/index.php>)。数据集信息如表 1。

表 1 数据集信息

Tab.1 Dataset information

数据集	特征数	样本数	类别数
Vowel	10	990	11
Wine	13	178	13
Vehicle	18	846	4
SPECT	22	267	2
WDBC	30	569	2
Ionosphere	34	351	2
SPECTF	43	267	2
Snoar	60	208	2

### 3.2 实验方案

从惯性权重和对更新粒子进行变异两个角度分析 BPSO 算法性能的变化。为了决定递增惯性权重设置方案中的  $w_{\max}$  和  $w_{\min}$  的取值, 对原始 BPSO 方案采用几组不同权重取值方案, 对每个数据集分别运行 30 次, 取其平均选择特征数和平均分类准确率作为算法性能的评价标准, 最终确定一个最优的权重方案。

在选择了最佳的权重方案条件下, 通过实验, 比较带固定变异的 BPSO 和本文提出的带自适应变异 BPSO 分别应用特征选择的分类准确率性能。因实验方案分两个内容如下:

(1) 在惯性权重设置方面分别采用原始递减区间 0.9~0.4、递增区间 0.4~0.9、递增区间 0.4~1.0 这 3 种不同的设置方案分别对 8 组不同类型数据进行特征选择实验对比。

(2) 在最佳惯性权重设置方案基础上, 分别用 AMBPSO 与文献[15]提出固定变异 MBPSO 算法对 8 组数据进行特征选择后分类准确率对比。

### 3.3 实验结果分析

#### 3.3.1 基于权重设置不同方案 BPSO 特征选择

在惯性权重设置方面分别采用原始递减区间 0.9~0.4、递增区间 0.4~0.9、递增区间 0.4~1.0 等 3 种不同的设置方案, 分别对 8 组不同类型数据最优的特征选择, 即寻找采用 kNN 分类准确率性能最优的特征选择方案。

表 2 给出了采用 3 种惯性权重设置方案的数据集分类准确率。从表 2 可以看出, 在采用惯性权重递增方案且递增范围与原始递减范围一致时, 数据集 WDBC 分类准确率稍低于原始递减方案, 其他数据集分类准确率稍高于原始递减方案, 但变化较小, 可见算法性能提升较小。把惯性权重递增范围上限提升到 1.0 时, 分类准确率均高于前两种方案, 算法性能得到进一步提升。因此, 本研究的惯性权重设置方案为递增且递增区间为 0.4~1.0。

#### 3.3.2 AMBPSO 算法与 MBPSO 算法性能的比较

AMBPSO 算法的惯性权重设置方案为上节确定的 0.4~1.0 的递增策略, 实验分别用 AMBPSO 和 MBPSO 算法特征选择的分类准确率。为了更对比性能, 实验中也列出两个特征选择时, 分别选择出的特征数量。表 3 给出了研究所采用的 AMBPSO 算法跟文献[15]所用 MBPSO

算法的比较结果。

从表3可见,AMBPSO算法性能在分类准确率和选择特征数量上都要明显优于MBPSO算法,尤其是3个维度较高的数据集 Ionosphere、SPECTF、Snoar 效果更加明显。

表2 不同惯性权重设置方案的分类准确率

Tab.2 Classification accuracy of different inertia weight setting schemes

数据集	不同惯性权重的分类准确率/%		
	0.9~0.4	0.4~0.9	0.4~1.0
Vowel	94.863	94.842	95.042
Wine	99.736	99.664	99.776
Vehicle	79.204	79.224	79.526
SPECT	87.857	88.028	88.086
WDBC	98.320	98.197	98.416
Ionosphere	92.593	93.103	93.846
SPECTF	85.182	85.514	87.266
Snoar	91.327	91.426	93.558

表3 AMBPSO算法与MBPSO算法实验结果比较  
Tab.3 Comparison of experimental results between AMBPSO algorithm and MBPSO algorithm

数据集 (特征数)	MBPSO		AMBPSO	
	分类准确率/%	选择特征数量	分类准确率/%	选择特征数量
Vowel(10)	94.962	8.852	95.076	8.136
Wine(13)	99.664	7.698	99.705	7.216
Vehicle(18)	79.055	9.584	79.723	8.162
SPECT(22)	86.264	11.252	87.985	9.945
WDBC(30)	98.453	16.238	98.537	14.112
Ionospher(33)	92.526	6.521	94.056	5.111
SPECTF(43)	84.965	15.652	87.692	12.252
Snoar(60)	91.015	27.628	94.383	21.422

### 3.3.3 选择特征数量的实验比较

为了对算法性能做更好的对比,表4给出了数据集在3种惯性权重设置方案及变异后选择特征的数量,从表4可以看出,采用区间在0.4~0.9的方案除数据集 Vehicle、SPECT、Snoar 选择特征数量相较原始方案减少1至2个外,其他数据集选择特征数量减少数量变化较小或有所增加。采用区间在0.4~1.0的方案相较前两个均无明显变化,以上分析采用3种惯性权重设置方案所选择

的特征数量总体上无明显变化。但在采用 AMBPSO 算法之后的选择特征数量明显少于3种惯性权重设置方案,尤其是在高维数据的变化更为明显,说明从特征选择数量性能指标分析,自适应变异 BPSO 的特征选择数据不仅优于固定变异操作,而且也提高了原始 BPSO 算法在不同权重方案下的性能。

表4 不同惯性权重设置方案选择特征的数量

Tab.4 Number of different inertia weight setting scheme selection features

数据集	不同惯性权重选择特征的数量			
	0.9~0.4	0.4~0.9	0.4~1.0	AMBPSO
Vowel	8.600	9.023	8.986	8.136
Wine	7.311	7.428	7.859	7.216
Vehicle	12.586	11.021	10.652	8.162
SPECT	10.956	9.756	10.524	9.945
WDBC	13.865	14.014	17.152	14.112
Ionosphere	7.354	7.271	7.684	5.111
SPECTF	17.584	18.385	16.984	12.252
Snoar	27.562	26.859	28.235	21.422

### 3.3.4 实验曲线对比图

为了直观地比较 AMBPSO 算法对特征选择性能的影响,图1给出了原始 BPSO 算法和 AMBPSO 算法在(a)~(h)8个数据集上的分类准确率随迭代变化的对比曲线。从图1可以看出,除了数据集 Wine 后期分类准确率 AMBPSO 算法稍低之外,其他7个数据集的 AMBPSO 算法分类准确率在实验前期低于原始 BPSO 算法,在实验后期随着迭代次数增加都明显高于原始 BPSO 算法,说明在前期,AMBPSO 算法的粒子的变异率较大,全局搜索能力较强,后期随着迭代次数的增加,粒子变异率逐渐减小,算法的局部搜索能力得到增强,最终 AMBPSO 算法分类准确率比 BPSO 算法高。图1曲线图与2.2节所分析的理论完全吻合,并且算法性能得到明显提高。

综上所述,从分类准确率和选择特征数量两个角度对比,结果证明本研究采用的新的惯性权重方案和自适应变异操作相结合的 AMBPSO 算法具有明显优势,从实验曲线图上可以看出 AMBPSO 算法增强了原始算法前期的全局搜索能力和后期局部搜索能力,明显提高了算法性能。

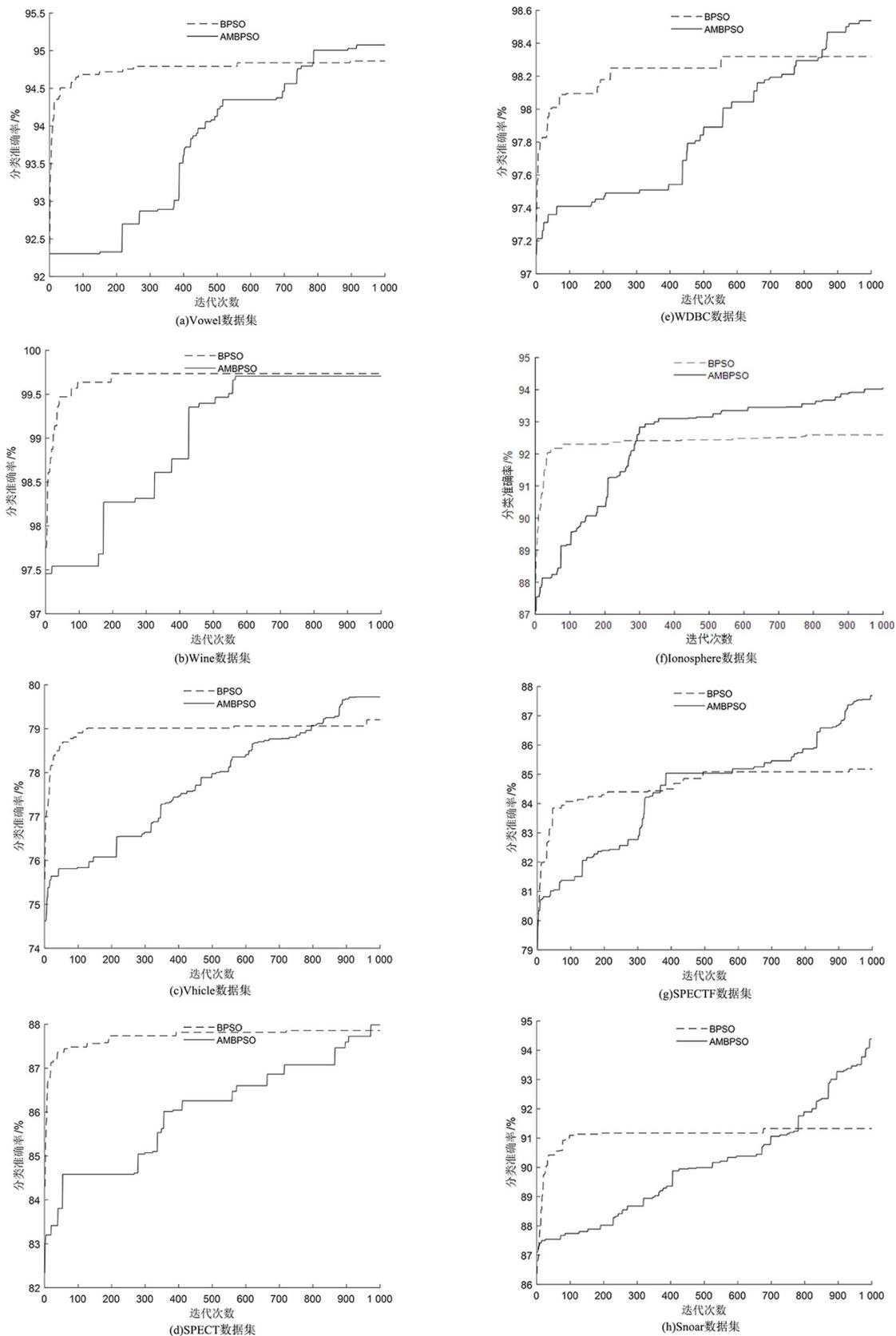


图 1 BPSO 算法和 AMBPSO 算法在 8 个数据集上的实验曲线对比图

Fig.1 Comparison of experimental curves of BPSO algorithm and AMBPSO algorithm on 8 data sets

## 4 结论

本研究提出的 AMBPSO 算法采用了线性递增的惯性权重设置,并通过实验验证的方式确定了惯性权重的取值区间。另外在粒子更新后采用了一种变异操作,随迭代次数变异概率递减,使得

粒子在前期的变异率大,后期变异率小。该算法在前期粒子的多样性更强,具有较强的全局搜索能力,后期具有较强的局部搜索能力,算法性能明显优于原始的 BPSO 算法以及固定变异的二进制粒子群算法(MBPSO)。

## 参考文献:

- [1] 周志华. 机器学习[M]. 北京:清华大学出版社, 2016: 247-266.
- [2] 赵妮,孙进才,梁峰,等. 基于 BPSO 的水下目标特征选择方法[J]. 计算机仿真, 2008(1): 196-199.
- [3] 艾解清,高济,彭艳斌. 基于离散粒子群算法的织物疵点特征选择[J]. 纺织学报, 2011, 32(11): 53-57.
- [4] 吴庆涛,曹继邦,郑瑞娟,等. 基于粒子群优化的入侵特征选择算法[J]. 计算机工程与应用, 2013, 49(7): 89-92.
- [5] 刘建华. 粒子群算法的基本理论及其改进研究[D]. 长沙:中南大学, 2009.
- [6] 刘建华,杨荣华,孙水华. 离散二进制粒子群算法分析[J]. 南京大学学报(自然科学版), 2011, 47(5): 504-514.
- [7] 李炜,巢秀琴. 改进的粒子群算法优化的特征选择方法[J]. 计算机科学与探索, 2019, 13(6): 990-1004.
- [8] CHUANG L, TSAI S, YANG C. Improved binary particle swarm optimization using catfish effect for feature selection[J]. Expert Systems with Applications, 2011, 38(10): 12699-12707.
- [9] WEI J, ZHANG R, YU Z, et al. A BPSO-SVM algorithm based on memory renewal and enhanced mutation mechanisms for feature selection[J]. Applied Soft Computing, 2017, 58: 176-192.
- [10] 张文杰,蒋烈辉. 一种基于遗传算法优化的大数据特征选择方法[J]. 计算机应用研究, 2020, 37(1): 50-52, 56.
- [11] SREEJA N K, SANKAR A. Pattern matching based classification using ant colony optimization based feature selection[J]. Applied Soft Computing, 2015, 31: 91-102.
- [12] 刘亮,何庆. 基于改进蝗虫优化算法的特征选择方法[J]. 南京大学学报(自然科学版), 2020, 56(1): 41-50.
- [13] 庄开元. 基于蜻蜓算法和花朵授粉算法的特征选择方法研究[D]. 长春:吉林大学, 2019.
- [14] KENNEDY J, EBERHART R. A discrete binary version of the particle swarm algorithm[C]//1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation. Orlando: IEEE, 1997:4104-4109.
- [15] LIU J, MEI Y, LI X. An analysis of the inertia weight parameter for binary particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation, 2016, 20(5): 666-681.
- [16] ZHANG Y, WANG S, PHILLIPS P, et al. Binary PSO with mutation operator for feature selection using decision tree applied to spam detection[J]. Knowledge-Based Systems, 2014, 64: 22-31.
- [17] COVER T, HART P. Nearest neighbor pattern classification[J]. IEEE Transactions on Information Theory, 1967, 13(1): 21-27.

(责任编辑:方素华)