

# 基于多目标邻域差分进化和模糊粗糙集的属性约简算法

李兵洋, 肖健梅, 王锡淮<sup>†</sup>

(上海海事大学 物流工程学院, 上海 201306)

**摘要:** 作为粗糙集的一种推广, 模糊粗糙集在属性约简中的应用尤为重要. 约简规模和约简依赖度作为评判约简性能的两个重要指标, 分别对应着约简的效率以及精度. 传统的约简算法通常以追求约简的最大依赖度为导向进行寻优, 并没有直接考虑约简的规模大小. 基于此, 强调所得约简的规模大小在约简运算中的重要性, 并提出一种基于邻域变异信息的多目标差分算法, 在约简运算中将约简的规模也作为单独的优化目标, 将属性约简问题转化为多目标优化问题, 综合考虑约简在属性数量和依赖度两方面的性能. 通过引入目标支配排序, 使得可以从属性数量和依赖度误差两方面对所得约简的性能进行约束, 并得到目标约束内的约简结果. 选取 UCI 上的数据集进行实验分析, 实验结果表明, 所提算法可以在目标约束内得到更加全面的约简结果, 具有一定的可行性, 是一种有效的约简算法.

**关键词:** 模糊粗糙集; 属性约简; 差分算法; 多目标优化; 特征选择

**中图分类号:** TP18

**文献标志码:** A

## Attribute reduction with fuzzy rough set based on multiobjective neighborhood difference algorithm

LI Bing-yang, XIAO Jian-mei, WANG Xi-huai<sup>†</sup>

(Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, China)

**Abstract:** As a generalization of rough set, fuzzy rough set plays a significant role in attribute reduction. Reduct size and dependency degree are two important evaluation criteria to measure reduction performance. Traditional reduction algorithms are mainly designed with the direction of maximizing dependency degree and reduct size is not taken into consideration. This paper emphasizes the significance of reduct size and proposes an multiobjective difference algorithm based on neighborhood mutation information, which makes reduct size a separate objective in attribute reduction. Hence the reduction problem is turned into multiobjective optimization problem, which considers reduction performance from aspects of both attribute number and dependency degree. By using goal-sequence domination scheme, the reduction performance can be stipulated from aspects of attribute number and dependency degree and satisfactory reduction results can be obtained. Several UCI data sets are applied to analyze the algorithm performance. The experimental results show that this method can obtain more comprehensive reduction results, which is feasible and effective.

**Keywords:** fuzzy rough set; attribute reduction; difference algorithm; multiobjective optimization; feature selection

## 0 引言

粗糙集理论<sup>[1]</sup>是由 Pawlak 教授提出的一种新的用于处理不精确、不完备数据的数学工具. 该理论通过上下近似集来描述和挖掘数据集中所蕴含的有效信息, 但是传统的粗糙集理论只能针对离散型数据进行分析. 作为粗糙集的一种重要推广模型, 模糊粗糙集<sup>[2]</sup>可以直接对实值数据进行处理, 避免了数据离散化所带来的信息缺失问题. 一般而言, 在一个决策表中并不是所有的条件属性都是必要的, 属性约简即在

保证决策表有效信息不变或者大致相等的情况下删除其中冗余的属性. 属性约简作为粗糙集理论中的重要问题, 可以有效地应用在特征选择等问题中<sup>[3-5]</sup>, 因此自其提出以来, 许多学者对模糊粗糙集的属性约简算法进行了深入研究<sup>[6-27]</sup>.

最小属性约简问题作为一个 NP 难题, 其计算复杂度会随着属性规模的扩大呈指数规模扩大<sup>[11]</sup>, 因而在实际中, 往往只要求取满足一定目标条件的次优约简即可. 对于一个约简而言, 其性能主要体现在两

收稿日期: 2017-11-02; 修回日期: 2018-03-25.

基金项目: 国家自然科学基金项目(61573240, 61503241); 上海海事大学博士创新基金项目(2017ycx084).

作者简介: 李兵洋(1993-), 男, 博士生, 从事粗糙集理论、智能信息处理的研究; 王锡淮(1962-), 男, 教授, 博士生导师, 从事粗糙集理论、粒计算、智能信息处理等研究.

<sup>†</sup>通讯作者. E-mail: wxh@shmtu.edu.cn.

个方面,一个是约简的分类一致程度,其对应着约简依赖度;另一个是约简所含属性的数量.目前常用的模糊粗糙集属性约简算法主要包括基于辨识矩阵的约简算法<sup>[23-27]</sup>、基于启发式条件的约简算法<sup>[13-22]</sup>以及基于智能算法的约简算法<sup>[6-12]</sup>等.以上几种类型的算法从不同的角度进行约简运算,并且都取得了不错的效果.然而,上述研究都是以追求约简的最大依赖度为目标进行约简计算的,并没有直接考虑所得约简包含属性数量的多少.约简依赖度和约简规模作为评判约简性能的两个重要指标,分别对应着约简的精度和效率.如果一味地追求最大的约简依赖度,则会造成约简规模过大,在某些极端情况下,所得约简甚至会与原决策表的规模相同.在这种情况下,约简也就失去了意义.对于样本数据而言,过多的特征不仅会占用大量存储空间,而且会对计算造成严重负担.因此,在实际的约简过程中,不仅要考虑约简的依赖度大小,同时还要考虑所得约简的规模大小.故本文同时从约简的两个性能指标出发处理属性约简问题.

由于属性约简问题本质上相当于一个非线性的组合优化问题,本文将约简所含属性的数量及依赖度分别设置为单独的目标函数,将属性约简的优化问题扩展为多目标优化问题,以多目标差分算法(简称MODE算法)为基础进行求解.由于在进化过程中随着种群个体差异不断变小,算法容易陷入局部最优且难以保证所得Pareto解集分布的均匀性与多样性,本文使用种群个体的邻域信息生成变异个体,利用改进的多目标邻域差分进化算法(简称IMODE)进行约简运算.此外,为了保证所求约简的质量,从约简依赖度和约简所含属性数量两方面设置目标约束.为了求得满足目标约束的约简结果,引入目标支配排序方法.实验结果表明,所提算法能够提供更加多元化且可行的约简结果.

## 1 模糊粗糙集基本理论

将一个决策表定义为四元组  $DT = \langle U, A, V, f \rangle$ . 其中:  $U$  是一个非空有限样本集,表示论域;  $A = C \cup D$  是样本属性集合,且满足  $C \cap D = \emptyset$ ,  $C$  为条件属性集合,  $D$  为决策属性集合,  $D$  将样本集划分为若干个决策等价类  $\{d_1, d_2, \dots, d_m\}$ ;  $V$  是相应的属性值集合;  $f: U \times A \rightarrow V$  是一个信息函数,它用于确定样本在各个属性下的属性值.

**定义1** 令  $U$  是一个非空有限论域,  $T$  是一个三角范数,如果  $\forall x, y, z \in U$ ,  $R$  满足自反性 ( $R(x, x) = 1$ )、对称性 ( $R(x, y) = R(y, x)$ ) 以及  $T$ -传递性 ( $R(x, z) \geq T(R(x, y), R(y, z))$ ), 则称  $U$  上的二元模糊关系

$R$  为一个模糊  $T$ -相似关系.

不同于传统粗糙集,模糊粗糙集建立在模糊相似关系的基础上来近似一个模糊对象.文献[28]将模糊粗糙集的上下近似算子总结为如下4种形式.

**定义2** 给定决策表  $DT = \langle U, A \rangle$ . 其中:  $U$  为一个非空论域;  $A = C \cup D$ ,  $C$  为条件属性集合,  $D$  为决策属性集合,其将样本集划分为若干个决策等价类  $\{d_1, d_2, \dots, d_m\}$ .  $R$  为  $U$  上的一个模糊  $T$ -相似关系, 则  $d_i (i = 1, 2, \dots, m)$  在  $R$  下的模糊下近似和上近似定义为

$$\begin{cases} \overline{R}_T d_i(x) = \sup_{u \in U} T(R(x, u), d_i(u)), \\ \underline{R}_S d_i(x) = \inf_{u \in U} S(N(R(x, u)), d_i(u)), \\ \overline{R}_\sigma d_i(x) = \sup_{u \in U} \sigma(N(R(x, u)), d_i(u)), \\ \underline{R}_v d_i(x) = \inf_{u \in U} v(R(x, u), d_i(u)). \end{cases} \quad (1)$$

其中: 下近似算子  $\underline{R}_S d_i(x)$  和  $\underline{R}_v d_i(x)$  表示样本  $x \in U$  必然属于决策类  $d_i$  的隶属度, 上近似算子  $\overline{R}_T d_i$  和  $\overline{R}_\sigma d_i$  表示样本  $x \in U$  可能属于决策类  $d_i$  的隶属度. 本文分别采用  $\overline{R}_T d_i$  和  $\underline{R}_v d_i$  计算模糊粗糙集的上下近似, 方便起见, 在下文中用  $\overline{R}$  和  $\underline{R}$  表示该对上下近似算子.

**定义3** 给定决策表  $DT = \langle U, A \rangle$ . 其中:  $A = C \cup D$ ,  $B \subseteq C$ ,  $a \in B$ ,  $D$  将样本集划分为若干个决策等价类  $\{d_1, d_2, \dots, d_m\}$ .  $R$  为  $U$  上的模糊  $T$ -相似关系, 则  $D$  关于  $B$  的模糊正域定义为

$$\text{POS}_B(D) = \bigcup_{i=1}^m \underline{R}_B d_i, \quad (2)$$

其中  $\underline{R}_B$  表示由条件属性集合  $B$  所确定的模糊  $T$ -相似关系. 由定义可知, 模糊正域是一个模糊集合, 样本  $x \in U$  在模糊正域中的隶属度反映该样本隶属于自身决策类的程度. 基于模糊正域, 定义决策属性  $D$  关于条件属性集合  $B$  的依赖度为

$$\gamma_B(D) = \frac{|\text{POS}_B(D)|}{|U|} = \frac{|\bigcup_{i=1}^m \underline{R}_B d_i|}{|U|}, \quad (3)$$

其中  $|\bigcup_{i=1}^m \underline{R}_B d_i| = \sum_i \sum_{x \in d_i} \underline{R}_B d_i(x)$ . 属性  $a$  对  $B$  的重要度为

$$\text{sig}_B(a) = \gamma_B(D) - \gamma_{B-a}(D). \quad (4)$$

从式(3)可以看出, 属性依赖度反映了样本集在属性集  $B$  下与决策属性的分类一致比例, 因此依赖度常作为反映决策表分类能力的指标. 在属性约简中, 追求在保持与原决策表属性集合依赖度最大程度一致的情况下找到拥有最少属性的属性集合.

**定义4** 给定决策表  $DT = \langle U, A \rangle$ ,  $A = C \cup D$ ,  $B \subseteq C$ , 则称  $B$  为决策表  $DT$  的一个属性约简. 若

$\gamma_B(D) = \gamma_C(D)$ , 且对于任意  $B' \subset B$ , 有  $\gamma_{B'} < \gamma_C$ .

## 2 基于IMODE算法的属性约简

MODE算法是在差分算法的基础上扩展而来, 用于处理多目标函数的优化问题. 其基本思想是利用种群个体的差异信息, 根据变异操作来探索新解, 利用交叉操作对个体值进行修正, 并根据非支配排序和拥挤度比较对种群个体进行分级排序, 从而选出Pareto意义下的更优个体. 其最终目的是为了找到Pareto最优解集.

### 2.1 多目标优化

在多目标优化问题中, 往往存在目标函数不可比较或者冲突的情况, 即对一个目标函数而言, 最好的解对于另一个目标函数可能性能很差, 因此多目标优化问题的解表现为Pareto解集的形式. 为了便于描述本文所提算法, 首先简要描述多目标优化的基本概念和原理. 对于一个含有  $m$  个目标函数以及  $n$  个决策变量的最小化问题  $\min F(P), F = \{f_1, f_2, \dots, f_m\}$  为  $m$  个待优化的目标函数,  $P = \{p_1, p_2, \dots, p_n\}$  为  $n$  维向量表决策变量, 非支配解的定义如下.

**定义5** 对于两个目标向量  $F_a$  和  $F_b$ , 如果  $f_{a,i} \leq f_{b,i}, \forall i \in \{1, 2, \dots, m\}$ , 且  $\exists j \in \{1, 2, \dots, m\}$ , 满足  $f_{a,j} < f_{b,j}$ , 则称  $F_b$  被  $F_a$  支配, 表示为  $F_a \prec F_b$ . 若  $F_a$  不被其他任何解所支配, 则称  $F_a$  为非支配解, 即Pareto解. 根据一个解  $F_{ind}$  被支配的情况, 可以将整个解空间内的个体进行分级并排序. 对于解  $F_{ind}$ , 如果支配其的解为  $k$  个, 则令其排序级别为  $k + 1$ <sup>[29]</sup>, 排序级别低的解更优.

### 2.2 编码方式与目标函数

对于条件属性集合  $C = \{c_1, c_2, \dots, c_k\}$ , 用一组二进制编码串  $S = \{s_1, s_2, \dots, s_k\}$  来表示属性的选取. 其中, 若选取第  $i$  条属性, 则  $s_i = 1$ , 否则  $s_i = 0$ . 由于差分算法的编码方式和变异操作是针对浮点数的, 本文采用区间编码映射机制来完成实数编码到离散编码的转换, 其定义如下:

$$M(X_{ij}) = \begin{cases} 0, & X_{ij} \in [0, 0.5); \\ 1, & X_{ij} \in [0.5, 1]. \end{cases} \quad (5)$$

其中  $X_{ij}$  表示种群中第  $i$  个个体在第  $j$  条属性上的值.

在属性约简任务中有两方面因素需要考虑, 一是所得约简能够最大程度地保持与原决策表属性集合依赖度一致, 即约简与原决策表的依赖度之差(简称为依赖度误差)要尽可能小; 二是约简所含属性数量要尽可能少. 因此, 在基于IMODE算法的属性约简算法中, 将约简所含属性的数量以及依赖度误差设置为两个单独的目标函数.

### 2.3 基于目标支配排序的进化约束机制

在实际中, 希望所得约简性能能够满足一定的指标, 例如所得约简依赖度误差应在一定范围内, 约简所含属性数应比原决策表小等. 基于非支配排序的多目标差分算法并不能控制最终所得的Pareto最优解集朝着期望目标方向进化, 也就是说, 所得结果中可能包含不符合条件的约简. 为了解决这个问题, 本文引入目标支配排序方法对算法改进, 以使得到的Pareto解朝着满足目标约束  $G = \{f_{g1}, f_{g2}, \dots, f_{gm}\}$  的方向进化. 在第一阶段, 该方法对种群中所有满足目标约束  $G$  的解进行非支配排序并记录排序级别; 第二阶段则对剩下的不满足  $G$  的解进行排序. 对于不满足  $G$  的解  $F'_a$  和  $F'_b$ , 若  $\text{abs}(F'_a - G) \prec \text{abs}(F'_b - G)$ , 则称  $F'_b$  被  $F'_a$  支配, 表示为  $F'_a \prec_G F'_b$ .

由以上定义可以看出, 若个体不满足目标约束, 则距离目标较远的个体被距离目标较近的个体支配. 因而, 这些个体在进化过程中会不断地向着目标约束范围内靠近. 而满足目标约束的个体则会在进化过程中不断地向着使目标函数更小的方向进化, 从而控制种群最终朝着期望目标的方向进化.

在基于IMODE的约简算法中, 首先利用目标支配排序对种群进行排序. 为了保证种群的解更加多样化, 对于同一排序级别内的解利用拥挤度比较进行进一步的比较和寻优. 拥挤度的定义如下:

$$CD(i) = \begin{cases} \inf, & \text{if } i \text{ is in the boundary;} \\ \sum_{j=1}^m \left| \frac{f'_j(i+1) - f'_j(i-1)}{f'_j(N) - f'_j(1)} \right|, & \text{else.} \end{cases} \quad (6)$$

其中  $f'_j(1), f'_j(2), \dots, f'_j(N)$  为全部解在目标函数  $f_j$  上按照函数值由小到大排列. 当种群中两个个体排序级别相同时, 拥挤度大的个体更优.

### 2.4 邻域进化机制

MODE算法中的变异操作利用种群个体间的差异信息来探索新解. 迭代算法后期, 由于个体间的差异信息逐渐变小, 算法的收敛速度变慢, 容易产生早熟的现象, 因而会影响所得Pareto解的分布多样性与均匀性. 为了降低由差异信息减小而产生的影响, 本文利用种群个体的邻域差异信息来生成变异个体以探索新解, 如图1所示. 对于种群中每个个体  $X_{i,j}$ , 相应的变异个体  $V_{i,j}$  按照下式生成:

$$F = F_{\min} + \text{rand} \times (F_{\max} - F_{\min}), \quad (7)$$

$$V_{i,j} = X_{r1,j} + F(N(X_{r2,j}, \sigma) - N(X_{r3,j}, \sigma)). \quad (8)$$

其中:  $X_{r1}, X_{r2}, X_{r3}$  表示父代种群中3个不同于  $X_i$  的个体;  $N(X_{ri}, \sigma)$  表示以  $X_{ri}$  为均值,  $\sigma$  为标准差的正态分布;  $F_{\max}, F_{\min}$  表示缩放因子的上下限;  $\text{rand}$

表示[0,1]范围内的随机数.

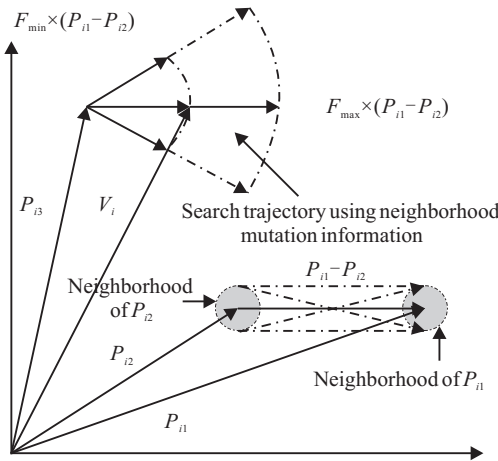


图1 基于个体邻域差异信息的变异操作

从图1可以看出,本文所提的改进变异操作不再局限于种群个体间的差异信息,而是鼓励充分利用种群个体间的邻域差异信息.在算法前期,由于利用的是邻域差异信息,种群将沿着更多可能的搜索方向来探索新解;在迭代后期,随着种群个体间的差异不断减小,变异操作将更多地邻域范围内进行局部寻优搜索,从而提高算法的搜索能力.

在生成变异个体后,通过交叉操作来生成新的子代个体.对于种群中的个体  $X_i$  以及由2.4节中变异操作得到的变异个体  $V_{i,j}$ ,通过下式进行交叉操作来得到子代个体  $X'_{ij}$ :

$$X'_{ij} = \begin{cases} V_{ij}, & \text{rand} \leq \text{CR} \mid \text{rc}(k) = j; \\ X_{ij}, & \text{else.} \end{cases} \quad (9)$$

其中:  $\text{CR} \in [0, 1]$  表示交叉概率,  $\text{rc}(k)$  表示  $\{1, 2, \dots, k\}$  中的随机数.

### 3 算法流程

基于IMODE的属性约简算法流程如图2所示.算法的具体描述如下.

输入: 决策表  $\text{DT} = \langle U, A \rangle, A = C \cup D, C$  为条件属性集合,  $D$  为决策属性集合,目标约束设置为  $G$ ;

输出: 决策表属性约简集合.

**Step 1:** 数据预处理. 首先用  $Z$ -score 标准化对决策表数据进行预处理以减少属性量纲的不同对约简结果的影响.

**Step 2:** 参数初始化. 确定种群规模  $N_{\text{pop}}$ , 最大进化代数  $\text{Gen}$ , 缩放因子  $F_{\text{max}}, F_{\text{min}}$ , 标准差参数  $\sigma$ , 以及交叉概率  $\text{CR}$ .

**Step 3:** 种群初始化. 根据2.2节,随机生成符合条件的  $N_{\text{pop}}$  个解,组成第一代父代种群  $\text{POP}_{\text{init}}$ .

**Step 4:** 进化操作. 通过变异和交叉操作生成新的子代种群.

**Step 5:** 竞争淘汰机制. 将原父代种群与子代种群合并,并对合并种群进行目标支配排序以及拥挤度计算.根据合并种群内个体间的支配关系和个体拥挤度选出 Pareto 意义下最优的  $N_{\text{pop}}$  个个体,组成新一代父代种群  $\text{POP}_{\text{new}}$ .

**Step 6:** 输出. 若达到最大进化代数,则终止计算,输出由当前父代种群  $\text{POP}_{\text{new}}$  得到的 Pareto 解集,否则转到 Step 4.

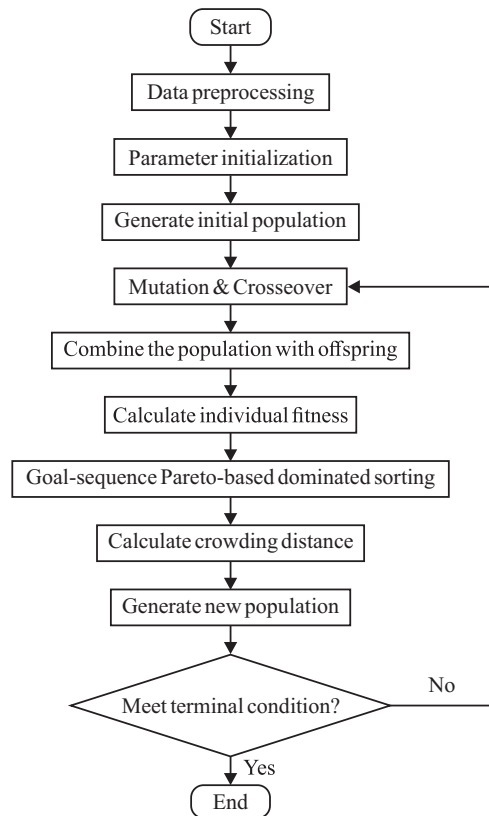


图2 算法流程

## 4 实验分析

### 4.1 实验基本设置

为了验证本文算法求解的效果,从UCI机器学习数据库中选择6组数据进行实验分析.数据集的基本描述如表1所示.本文利用高斯核函数来计算决策表样本间的模糊相似关系  $R(x_i, x_j)$ . 此时,  $R(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / 2\delta^2)$  满足自反性、对称性和  $T_{\text{cos}}$ -传递性. 其中:  $\|x_i - x_j\|$  表示样本  $x_i$  和样本  $x_j$  间的欧氏距离,  $T_{\text{cos}}(a, b) = \max(ab - \sqrt{1 - a^2}\sqrt{1 - b^2}, 0)$ . 算法的基本参数设置见表2.

表1 数据集基本信息

ID	Data	features	class	samples
1	Redwine	11	4	1571
2	Wine	13	3	178
3	Breast	9	2	277
4	Iris	4	3	150
5	ILPD	10	2	583
6	Seeds	7	3	210

表2 算法参数设置

Parameters		Parameters	
Population size	20	Cross factor	0.3
Max iterations	100	Goal function1	Card( $X_{ij}$ )
Max scale factor	0.6	Goal function2	$\gamma_C(D) - \gamma_{X_{ij}}(D)$
Min scale factor	0.2	Goal constraint	$[n - 1, 0.2\gamma_C(D)]$
Standard derivation	0.5		

表2中:  $\text{Card}(X_{ij})$  表示个体  $X_{ij}$  为1的个数, 即个体  $X_{ij}$  确定的条件属性数;  $\gamma_C(D)$  表示原决策表的决策属性关于条件属性的依赖度;  $\gamma_{X_{ij}}(D)$  表示由个体  $X_{ij}$  确定的决策表的决策属性关于条件属性的依赖度;  $n$  表示原决策表条件属性数量. 在实际中, 希望所得约简的属性数要小于原决策表, 否则约简也就失去了意义. 因此, 对于目标函数  $f_1$ , 设置目标约束为  $n - 1$ . 此外, 为了保证约简的分类能力, 要求约简与原决策表间的依赖度误差在一定阈值内, 本文取为

表4 算法求得Pareto解集情况

Data	IMODE			MODE			NSGA-II		
	MinP	MaxP	AveP	MinP	MaxP	AveP	MinP	MaxP	AveP
Redwine	3	4	3.68	2	4	3.06	1	4	2.86
Wine	4	5	4.60	3	4	3.38	2	4	2.74
Breast	1	2	1.86	1	2	1.72	1	2	1.58
Iris	1	1	1.00	1	1	1.00	1	1	1.00
ILPD	5	7	5.90	4	6	4.50	3	5	3.96
Seeds	2	3	2.82	2	3	2.60	2	3	2.38

从表4可以看出, 与MODE和NSGA-II算法相比, 本文所提的IMODE算法容易找到分布更为多样的Pareto解集. Pareto解集的多样性可以保证在满足预期性能指标的前提下, 提供更加多元化并且可行的约简结果. 以数据集Redwine为例, 在50次实验中, IMODE算法所求的Pareto解集最少包含3个有效约简, 最多包含4个有效约简, 50次试验结果平均包含3.68个有效约简. 一般的MODE算法以及NSGA-II算法所求的Pareto解集最少包含的有效约简分别为2个和1个, 且50次实验结果平均包含的约简数也更少, 分别为3.06个和2.86个. 这是因为IMODE算法采用基于邻域差异信息的变异操作, 在迭代寻优的前期能够沿着更多可能的方向搜寻, 增强了算法的全局搜索能力. 在算法后期, 种群内的个体逐渐趋于一致. 此时, 对于MODE和NSGA-II算法而言, 由于个体间差异细微, 难以生成新的变异个体, 因而容易陷入局部最优; 而IMODE算法将在个体的邻域范围内进行搜寻, 因而其局部寻优能力也得到了保证.

4.3 约简结果性能分析

为了进一步验证和比较算法所得约简的性能, 统计了在50次实验中, 所求的Pareto解集中的约简在属

$0.2\gamma_C(D)$ . 表3给出了各数据集具体的目标设置.

表3 各数据集目标约束具体值

	Redwine	Wine	Breast	Iris	ILPD	Seeds
$f_1$	10	12	8	3	9	6
$f_2$	0.0107	0.0400	0.0249	0.0221	0.0279	0.0208

4.2 解集的分布多样性分析

为了验证算法性能, 并避免算法结果的随机性, 分别利用NSGA-II<sup>[30]</sup>、MODE及IMODE算法对每组数据集在相同的目标约束条件下分别进行50次实验. 其中, NSGA-II算法的交叉和变异概率分别取0.9和0.05, 种群规模及最大的迭代次数取表2中设置的参数. 算法的Pareto解集求解情况如表4所示. 在表4中, 分别用MinP、MaxP以及AveP表示算法在50次实验中所求Pareto解集中包含的有效约简的最少个数、最多个数以及平均个数.

性数量和依赖度误差上的具体表现, 并取50次实验的平均结果. 在不同数据集下所求约简的结果如图3和图4所示. 从图中可以看出, 3种算法均能够求得满足目标约束的约简, 但传统的MODE以及NSGA-II算法所得的Pareto解却被IMODE算法所得的Pareto解集支配. 以ILPD数据集为例, 在目标约束范围内, 3个算法所求约简拥有相同数量的属性时, IMODE算法找到的约简具有更低的依赖度误差, 因此其分类一致度更高, 求得的约简更优.

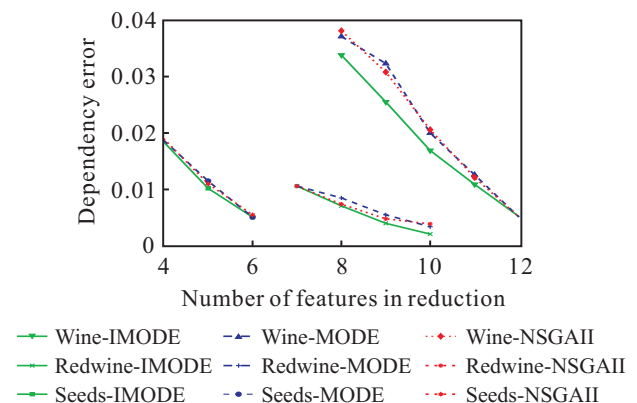


图3 算法在Wine、Redwine、Seeds上的约简性能比较

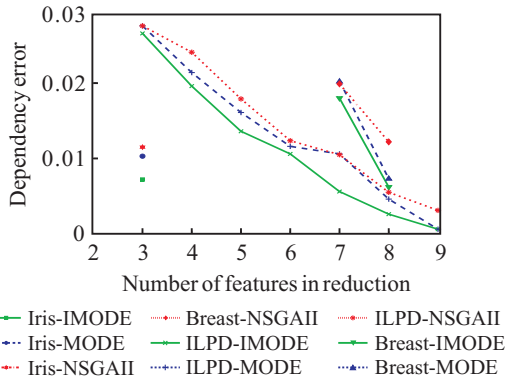


图4 算法在Iris、ILPD、Breast上的约简性能比较

从约简结果上看,IMODE算法找到的是Pareto解,即一个约简集合.每一个约简都对应一个二维的性能指标值,在目标约束内,这些约简彼此互不支配,在约简性能上的表现各有优劣,综合考虑了约简在属性数量和依赖度两个方面的性能,因而所得约简结果更加全面,也更加多元化.

进一步地,以IMODE算法在各数据集上所求的

约简结果为例,在所得约简结果中分别取拥有最少属性数及拥有最大依赖度的约简进行分析,利用CART及BP算法通过十折交叉验证来统计其分类精度,结果如表5所示.其中BP网络为双隐层结构,训练算法采用动量梯度下降算法.表5中: $a_{C_{ini1}}$ 、 $a_{C_{ini2}}$ 分别表示原决策表在CART及BP算法下的分类精度, $a_{C_{max dep1}}$ 、 $a_{C_{max dep2}}$ 分别表示拥有最大依赖度的约简在CART及BP算法下的分类精度, $a_{C_{min num1}}$ 、 $a_{C_{min num2}}$ 分别表示拥有最少属性数的约简在CART及BP算法下的分类精度,Num表示相应的约简所含条件属性的数量.可以看出,算法所求约简的分类精度与原决策表相当,这说明所得约简是有效的.但是,在IMODE算法求得的约简结果中,相比拥有较高依赖度的约简,拥有较少属性的约简,其分类精度也普遍较低,这是因为属性数量减少的同时,相应的依赖度也会减小.而依赖度较高的约简虽然具有更好的分类一致性,但是却牺牲了约简在规模上的性能,这也体现了约简结果在规模和依赖度上的协调.

表5 CART/BP算法下的分类精度结果对比

Data	$a_{C_{ini1}}$	$a_{C_{ini2}}$	Num	$a_{C_{max dep1}}$	$a_{C_{max dep2}}$	Num	$a_{C_{min num1}}$	$a_{C_{min num2}}$	Num
Redwine	0.6264	0.6135	11	0.6156	0.6180	10	0.6053	0.6075	7
Wine	0.9167	0.9674	13	0.9083	0.9625	12	0.8986	0.9347	8
Breast	0.7145	0.6586	9	0.7249	0.6482	8	0.7259	0.6382	7
Iris	0.9433	0.9567	4	0.9400	0.9492	3	0.9400	0.9492	3
ILPD	0.6659	0.7026	10	0.6691	0.6927	9	0.6722	0.6991	3
Seeds	0.9208	0.8952	7	0.9125	0.8929	6	0.8988	0.8810	4

#### 4.4 对比实验分析

选择基于前向特征排序法<sup>[13]</sup>(简称Ranking)以及基于反向特征选择法<sup>[14]</sup>(简称Backward)的约简算法与本文所提算法作进一步对比分析.其中:Ranking根据不同特征相对决策属性的重要度对所有特征排序,并按照重要度大小逐个加入到约简集合中;Backward则计算特征对当前数据集的重要度,删除重要度最小的特征并更新当前数据集,不断迭代以上过程直至满足停止条件.以上两种算法均能在给定的依赖度误差范围内得出拥有不同属性数量的约简集合.为了更加细致地分析不同算法的效果,实验选取数据集ILPD、wine以及两组高维数据集iono和sonar进行分析,数据集基本信息如表6所示.

表6 数据集基本信息

ID	Data	features	class	samples	$f_1$	$f_2$
7	iono	34	2	351	33	0.0692
8	sonar	60	2	208	59	0.0832

图5给出了iono和sonar数据集在IMODE、MODE以及NSGA-II算法下得到的Pareto解集.由

图5可知,相比MODE和NSGA-II算法,在所得Pareto解集的分布上,IMODE算法得到的Pareto解集分布更为多样,因此得到的有效约简也更加全面和多元化.而从所得约简的性能上看,IMODE算法得到的约简拥有更低的依赖度误差.值得注意的是,约简所含的属性数量越少,MODE、NSGA-II与IMODE算法得到约简的性能差异越明显.这是因为不同特征属性所含的有效信息量不同,约简运算删去的属性越多,删去不同属性带来的信息量的差异也越明显.这与4.2节和4.3节中的结论基本一致.

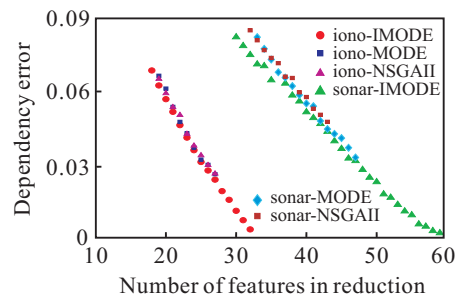


图5 各算法所得约简结果比较(iono & sonar)

接下来,分析对比IMODE以及Ranking算法在

数据集 *iono* 及 *sonar* 上的表现. 根据 Ranking 算法按照重要度大小可以得到加入不同数量的属性时的约简, 并将得到的约简与 IMODE 算法的结果相比, 如图 6 所示. 其中, 水平线表示最大允许误差下的依赖度取值. 分别利用 CART 和 BP 算法通过十折交叉验证统计在约简中加入不同数量的属性时得到的平均分类精度, 并与 IMODE 算法的结果相比较, 如图 7 和图 8 所示.

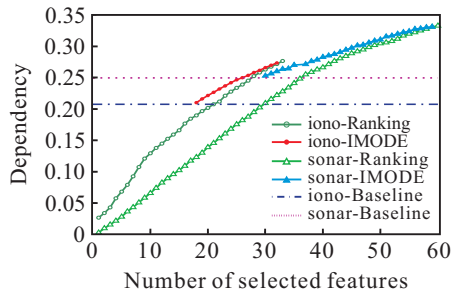


图 6 IMODE 与 Ranking 算法的结果对比 (*iono* & *sonar*)

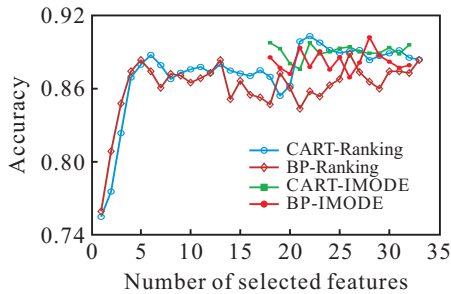


图 7 IMODE 与 Ranking 算法的平均分类精度对比 (*iono*)

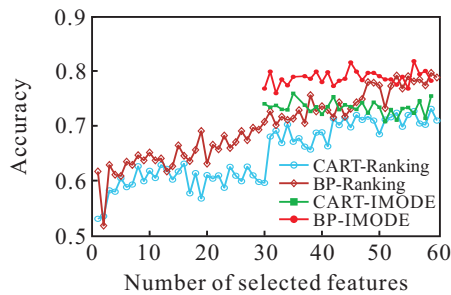


图 8 IMODE 与 Ranking 算法平均分类精度对比 (*sonar*)

从图 6 可以看出, 随着约简所含属性数量的增加, 约简的依赖度呈上升的趋势. 在最大允许的依赖度误差内, 对于数据集 *iono*, Ranking 算法找到的约简最少包含 21 条条件属性, 有效约简共 12 个; IMODE 算法找到的约简最少包含 18 条条件属性, 有效约简共 15 个. 对于数据集 *sonar*, Ranking 算法找到的约简最少包含 37 条条件属性, 有效约简为 24 个; IMODE 算法找到的约简最少只包含 30 条条件属性, 有效约简 29 个. 在依赖度方面, 拥有相同数量的属性时, IMODE 算法得到的约简要比 Ranking 算法拥有更高的依赖度, 并且随着约简所含属性数量的减少, 依赖度间的差异越来越明显. 对于平均分类精度, 从图 7 和图 8

可以看出, 随着属性数量的增加, 平均分类精度总体呈上升的趋势. 当拥有相同属性数量时, 对于数据集 *iono*, IMODE 算法得到的约简在 BP 算法下普遍要比 Ranking 得到的约简拥有更高的分类精度. 而在 CART 算法下, 在属性数量较少时, IMODE 算法得到的约简相比 Ranking 算法有更高的分类精度, 而随着属性数量的增加, 两者的平均分类精度基本保持一致. 对于数据集 *sonar*, 相比 Ranking 算法, IMODE 算法得到的约简不管是在 BP 算法还是 CART 算法下得到的平均分类精度都更高. 此外, 可以发现, 随着属性数量的增加, Ranking 算法所得约简在 CART 和 BP 算法下的平均分类精度逐渐与 IMODE 算法趋于一致, 这与依赖度的趋势相一致. 在约简拥有属性数量较少时, IMODE 与 Ranking 算法得到的约简依赖度相差较大, 因而其平均分类精度差异也较明显. 随着属性数量的增加, 依赖度间的差异越来越小, 因此平均分类精度也逐渐趋于一致.

图 9 给出了 IMODE 算法以及 Backward 算法在数据集 *iono* 以及 *sonar* 上关于约简在规模和依赖度上的表现. 图中标注了在允许的最大依赖度误差下, 各算法所得约简拥有的最少属性数量以及相应的依赖度大小. 对于数据集 *iono*, Backward 算法得到的约简最少包含 26 条条件属性, 共 7 个有效约简. 在拥有相同属性数量时, IMODE 算法所得约简的依赖度要高于 Backward. 对于数据集 *sonar*, Backward 算法得到的约简最少包含 46 条条件属性, 共 14 个有效约简. 在属性数量为 59 时, IMODE 比 Backward 算法所得约简依赖度略低, 其余情况下均要高于 Backward. 在两个数据集上, 随着约简包含属性数量的减少, Backward 与 IMODE 算法得到的约简在依赖度上的差异逐渐变大.

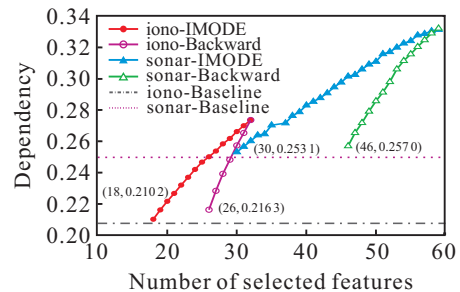


图 9 IMODE 与 Backward 的结果对比 (*iono* & *sonar*)

IMODE 与 Ranking、Backward 算法在 *wine* 和 *ILPD* 数据集上的表现如图 10 和图 11 所示. 由图可知, 对于 *wine* 数据集, Ranking、Backward 以及 IMODE 算法找到的约简最少分别包含 10 条、11 条和 8 条条件属性, 有效约简分别为 3 个、2 个和 5 个. 在拥有相同数量的属性时, IMODE 算法得到的约简依赖度比

Ranking 以及 Backward 算法都要高,其中 Ranking 所得约简依赖度最低.但是,相比 Backward, Ranking 找到了更多的有效约简,而 IMODE 找到的有效约简数量比这两种算法都要多.对于 ILPD 数据集, Backward 和 IMODE 算法所得约简最少,分别包含 9 条以及 3 条条件属性,得到的有效约简分别为 1 个和 7 个,而 Ranking 算法则没有找到有效约简(此时满足目标约束的约简等于原决策表).此外,当拥有相同数量的属性时, IMODE 所得约简的依赖度比 Ranking 以及 Backward 都要高,且随着约简所含属性数量的减少,依赖度的差异越来越明显,这与前面的结论基本保持一致.

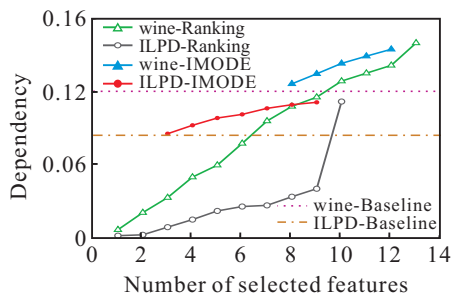


图 10 IMODE 与 Ranking 算法的结果对比 (wine & ILPD)

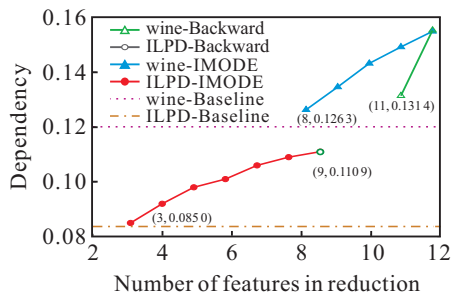


图 11 IMODE 与 Backward 算法结果对比 (wine & ILPD)

从以上分析可以看出,相比 Ranking 和 Backward 算法, IMODE 算法不仅能够求得分布更为多样的约简,并且在拥有相同数量的属性时,求得的约简具有更高的依赖度.另一方面,在探索具有较少属性数量的约简时, IMODE 相比 Ranking 以及 Backward 算法的效果更好.这主要是因为算法的寻优策略不同.属性约简的本质是非线性的组合优化问题<sup>[6]</sup>,属性重要度与属性集依赖度之间并不是单纯的线性关系.也就是说,约简集合并不一定是由重要度最大的前  $n$  条属性构成.由于 Ranking 和 Backward 算法都是在当前属性集合的基础上,根据属性重要度追求使当前属性集合性能最佳化的情况,其本质上属于单调型的贪心寻优方法,因此其并不能得到最优的约简结果.相比之下,结合属性约简的本质和元启发式算法处理非线性优化问题的能力,本文用 IMODE 算法来处理该非线性组合优化问题,邻域差异信息的引入在增强算

法的全局寻优能力的同时也保证了算法在迭代后期具有一定的局部寻优能力,因而所得的约简结果更优.此外,在给定的目标约束范围内, IMODE 算法所求的约简结果更加多元化,不仅包含了拥有最大依赖度的约简,也包含了拥有最少属性数量的约简.这对应着传统约简算法中两种不同的寻优策略:以追求约简最大依赖度为目标,以及追求约简包含属性数量最少为目标.因此, IMODE 算法求解的约简能够提供更加全面的信息和选择,并可以用在以不同需求为导向的约简以及特征提取任务中.

## 5 结论

针对模糊粗糙集的属性约简问题,本文从多目标优化的角度出发,提出了一种多目标邻域差分进化算法,并用于属性约简问题中.为了得到更好的约简结果,文中对多目标差分算法的变异算子作出改进措施,通过种群个体的邻域差异信息来生成变异个体,使得种群个体在算法初期能够沿着更多的方向探索新解,增强了算法的全局搜索能力.与 MODE 和 NSGA-II 算法相比,本文改进的算法具有更好的优化结果.通过引入目标支配排序,使得可以从属性数量和依赖度误差两方面对所得约简的性能进行约束,并得到目标约束内的约简结果.与两种常用的约简算法的对比表明,本文所提算法能够在目标约束内得到更加全面多元化的约简结果,提供了更多的信息和选择空间,且在相同情况下所得的约简更优,具有一定的可行性与实际意义.但是,本文算法主要是针对单标记数据集进行研究的,而目前多标记以及多模态数据集的属性约简以及特征提取也是研究热点之一,因此未来将完善本文算法对不同类型数据集的可行性分析.

## 参考文献 (References)

- [1] Pawlak Z. Rough sets[J]. Int J of Computer and Information Science, 1982, 11(5): 341-356.
- [2] Dubois D, Prade H. Rough fuzzy sets and fuzzy rough sets[J]. Int J of General Systems, 1990, 17(2/3): 191-208.
- [3] An S, Hu Q H, Pedrycz W, et al. Data-distribution-aware fuzzy rough set model and its application to robust classification[J]. IEEE Trans on Cybernetics, 2016, 46(12): 3073-3085.
- [4] 付志耀, 高岭, 孙骞, 等. 基于粗糙集的漏洞属性约简及严重性评估[J]. 计算机研究与发展, 2016, 53(5): 1009-1017.  
(Fu Z Y, Gao L, Sun Q, et al. Evaluation of vulnerability severity based on rough sets and attributes reduction[J]. J of Computer Research and Development, 2016, 53(5): 1009-1017.)
- [5] 张清华, 胡荣德, 姚龙洋, 等. 基于属性重要度的风

- 险决策粗糙集属性约简[J]. 控制与决策, 2016, 31(7): 1199-1205.  
(Zhang Q H, Hu R D, Yao L Y, et al. Risk DTRS attribute reduction based on attribute importance[J]. Control and Decision, 2016, 31(7): 1199-1205.)
- [6] 叶东毅, 陈昭炯. 最小属性约简问题的一个有效的组合人工蜂群算法[J]. 电子学报, 2015, 43(5): 1014-1020.  
(Ye D Y, Chen Z J. An efficient combinatorial artificial bee colony algorithm for solving minimum attribute reduction problem[J]. Acta Electronica Sinica, 2015, 43(5): 1014-1020.)
- [7] Gu X P, Li Y, Jia J H. Feature selection for transient stability assessment based on kernelized fuzzy rough sets and memetic algorithm[J]. Int J of Electrical Power and Energy Systems, 2015, 64: 664-670.
- [8] Chen L, Liu H L, Wan Z L. An attribute reduction algorithm based on rough set theory and an improved genetic algorithm[J]. J of Software, 2014, 9(9): 2276-2282.
- [9] Chebroly S, Sanjeevi S G. Attribute reduction on real-valued data in rough set theory using hybrid artificial bee colony: Extended FTSBPSD algorithm[J]. Soft Computing, 2016, 21(24): 7543-7569.
- [10] Zouache D, Ben A F. A cooperative swarm intelligence algorithm based on quantum-inspired and rough sets for feature selection[J]. Computers and Industrial Engineering, 2018, 115: 26-36.
- [11] 徐雪松, 陈荣元. 融合粗糙集与小生境免疫优化的属性约简方法[J]. 电子学报, 2014, 42(8): 1545-1550.  
(Xu X S, Chen R Y. Attribute decision reduction method based on hybrid rough sets and niche immune optimization[J]. Acta Electronica Sinica, 2014, 42(8): 1545-1550.)
- [12] 续欣莹, 张扩, 谢琚, 等. 基于互信息下粒子群优化的属性约简算法[J]. 电子学报, 2017, 45(11): 2695-2704.  
(Xu X Y, Zhang K, Xie J, et al. An attribute reduction algorithm based on mutual information of particle swarm optimization[J]. Acta Electronica Sinica, 2017, 45(11): 2695-2704.)
- [13] 段洁, 胡清华, 张灵均, 等. 基于邻域粗糙集的多标记分类特征选择算法[J]. 计算机研究与发展, 2015, 52(1): 56-65.  
(Duan J, Hu Q H, Zhang L J, et al. Feature selection for multi-label classification based on neighborhood rough sets[J]. J of Computer Research and Development, 2015, 52(1): 56-65.)
- [14] 张腾飞, 肖健梅, 王锡淮. 粗糙集理论中属性相对约简算法[J]. 电子学报, 2005, 33(11): 2080-2083.  
(Zhang T F, Xiao J M, Wang X H. Algorithms of attribute relative reduction in rough set theory[J]. Acta Electronica Sinica, 2005, 33(11): 2080-2083.)
- [15] Li F, Hu B Q, Wang J. Stepwise optimal scale selection for multi-scale decision tables via attribute significance[J]. Knowledge-Based Systems, 2017, 129: 4-16.
- [16] Hu Q H, Zhang L, Chen D G, et al. Gaussian kernel based fuzzy rough sets: Model, uncertainty measures and applications[J]. Int J of Approximate Reasoning, 2010, 51(4): 453-457.
- [17] Raza M S, Qamar U. Feature selection using rough set-based direct dependency calculation by avoiding the positive region[J]. Int J of Approximate Reasoning, 2018, 92: 175-197.
- [18] 何松华, 康婵娟, 鲁敏, 等. 基于邻域组合测度的属性约简方法[J]. 控制与决策, 2016, 31(7): 1225-1230.  
(He S H, Kang C J, Lu M, et al. Attribute reduction method based on neighborhood combination measure[J]. Control and Decision, 2016, 31(7): 1225-1230.)
- [19] Wang C Z, Qi Y L, Shao M W, et al. A fitting model for feature selection with fuzzy rough sets[J]. IEEE Trans on Fuzzy Systems, 2017, 25(4): 741-753.
- [20] Hu Q H, Zhang L J, Zhou Y C, et al. Large-scale multimodality attribute reduction with multi-kernel fuzzy rough sets[J]. IEEE Trans on Fuzzy Systems, 2018, 26(1): 226-238.
- [21] Du W S, Hu B Q. A fast heuristic attribute reduction approach to ordered decision systems[J]. European J of Operational Research, 2018, 264(2): 440-452.
- [22] 马福民, 陈静雯, 张腾飞. 基于双重粒化准则的邻域多粒度粗糙集快速约简算法[J]. 控制与决策, 2017, 32(6): 1121-1127.  
(Ma F M, Chen J W, Zhang T F. Quick attribute reduction algorithm for neighborhood multi-granulation rough set based on double granulate criterion[J]. Control and Decision, 2017, 32(6): 1121-1127.)
- [23] Wei W, Wu X Y, Liang J Y, et al. Discernibility matrix based incremental attribute reduction for dynamic data[J]. Knowledge-Based Systems, 2018, 140: 142-157.
- [24] Chen D G, Yang Y Y. Attribute reduction for heterogeneous data based on the combination of classical and fuzzy rough set models[J]. IEEE Trans on Fuzzy Systems, 2014, 22(5): 1325-1334.
- [25] He Q, Wu C X, Chen D G, et al. Fuzzy rough set based attribute reduction for information systems with fuzzy decisions[J]. Knowledge-Based Systems, 2011, 24(5): 689-696.
- [26] Liu G L, Hua Z. Partial attribute reduction approaches to relation systems and their applications[J]. Knowledge-Based systems, 2018, 139: 101-107.
- [27] Dai J H, Hu Q H, Zhang J H, et al. Attribute selection for partially labeled categorical data by rough set approach[J]. IEEE Trans on Cybernetics, 2017, 47(9): 2460-2471.
- [28] Yeung D, Chen D G, Tsang E, et al. On the generalization of fuzzy rough sets[J]. IEEE Trans on Fuzzy Systems, 2005, 13(3): 343-361.
- [29] Tan K, Khor E, Lee T, et al. An evolutionary algorithm with advanced goal and priority specification for multi-objective optimization[J]. J of Artificial Intelligence Research, 2003, 18(1): 183-215.
- [30] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Trans on Evolutionary Computation, 2002, 6(2): 182-197.