粗等价粒度下基于多种加速策略的增量式求核算法

赵 洁[1] 张恺航[2] 董振宁[1] 梁俊杰[3] 徐克付[2]
(广东工业大学 广州 510520)[1] (中国科学院信息工程研究所 北京 100093)[2]
(华南理工大学 广州 510006)[3]

摘 要 提出一种全新的渐增式求核算法。首先基于全局等价类提出粗等价类概念并分析其性质,研究粗等价类下的求核与约简;深入研究3类粗等价类与核属性的内在联系,设计粗等价类下判断核属性的等价方法和渐增式求核方法,通过该方法可在一次增量计算中求得多个非核属性,从而设计双向剪枝策略;可从属性和实体双方面缩减计算域,无需遍历全部属性和实体,在无核情况下,剪枝策略仍然有效。设计多次Hash的属性增量划分算法来完成上述增量式计算,基于此给出完整的渐增式求核算法。最后用UCI中20个决策表及海量、超高维3类数据集从多个角度进行验证,实验结果证明了所提算法的有效性和高效性,其尤其适用于大型决策表,大多数情况下优于现有算法。算法可进一步作为新型约简和优化算法的基础。

关键词 粗糙约简,粗等价类,渐增式求核,Hash
中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.01.043

Granularity of Rough Equivalence Class Based Incremental Attribute Core Computation Using Multiple Accelerating Strategies Pruning and Multiple Hashing

ZHAO Jie[1] ZHANG Kai-hang[2] DONG Zhen-ning[1] LIANG Jun-jie[3] XU Ke-fu[2]
(Guangdong University of Technology, Guangzhou 510520, China)[1]
(Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China)[2]
(South China University of Technology, Guangzhou 510006, China)[3]

Abstract A new incremental core computation algorithm was proposed. Firstly, rough equivalence class (REC) was proposed based on the smallest computational granularity of global equivalences, the character of REC was analyzed and core and reduction computation under REC was studied. Then relationship of core attributes and REC were studied and then an equal method of judging core attribution and incremental core computation method based on 0-REC were designed, through which multiple non-core attributions can be gained in one calculation. Based on which, bilateral pruning strategies were proposed to reduce calculation field of both attributes and entities, so it need not travel all the attributes and entities. The pruning strategies still work even there is no core. At last, 20 decision sets of UCI, massive and ultra-high dimension were used to verify the strategies and algorithms. The results show that the algorithm is effective and efficient, and in most conditions, the algorithm of this paper is superior to the current algorithms, and fit for massive decision table especially. The algorithm can be the basis of new reduction and optimization algorithms.

Keywords Reduction under rough set, Rough equivalence class, Incremental core computation, Hash

1 引言

属性约简是粗糙集理论[1,2]研究的核心问题之一,高效的约简算法研究一直备受关注。近年来约简算法研究向纵深方向发展,基于粗糙集的扩展[3,4]或融合多种理论进行算法设计,如模糊集[5,6]、证据理论[7]或多种理论综合使用[8],算法趋向精细化设计,如研究不一致数据下算法如何增速[9];当多个属性具有相同重要度时如何挑选[10]等。求所有约简与最小约简是典型的NP-Hard问题,一般来说求某个可行解可满足普通应用需要,启发式方法在这方面具有较大优势,成果众多[11-16],但很多应用需要求最小约简或多个次优约简,因此优化技术被大量应用在约简算法中[17-20]。在上述两类研究中求核是基础且重要的核心算法。基于优化技术的约简算法研究中,核可作为寻优的初始解,尤其对于某些优化方法,初始解的质量对寻优效率影响较大[17,19];核更新算法[21,22]则是基于分辨矩阵约简算法的热点问题。求核算法[16,23-25]是基于正区域约简研究的核心算法[10]。

基于正区域的约简时间和空间复杂度均较低,一直是研究热点。求核往往始于等价类和正区域算法,求核与约简的高效性依赖于基础算法的时间复杂度降低。刘少辉等[11]基于快速排序求正区域,时间复杂度和空间复杂度为$O(|C||U|\log|U|)$和$O(|U|+1)$,从而求核算法的时间复杂度为

到稿日期:2015-11-03 返修日期:2016-03-31 本文受国家自然科学基金资助项目:DS证据推理下抗信誉共谋攻击的行为信任研究(71401045)资助。

赵 洁(1979—),女,博士生,副教授,主要研究方向为智能算法、数据挖掘;E-mail:zhaojie@gdut.edu.cn(通信作者);张恺航(1993—),男,硕士生,主要研究方向为信息安全、数据挖掘;董振宁(1978—),男,博士生,主要研究方向为物流金融。

$O(|C|^2|U|\log|U|)$，该求核方法通过判断 $POS_C(D)\neq POS_{C-\{a_i\}}(D)$ 逐个验证 a_i，需要遍历全部属性。徐章艳等[13]引入基数排序，在计算中不断丢掉不影响计算正区域结果的对象，求正区域的算法时间复杂度为 $O(|C||U|)$。葛浩[23]提出一种分布计数基数排序，等价类算法的时间复杂度和空间复杂度为 $O(|C||U|)$ 和 $O(|U|)$，缩减计算域的方法优于文献[13]，求核时间复杂度和空间复杂度为 $O(|C|^2|U|)$ 和 $O(|U|)$。刘勇等[14]提出的基于 Hash 的求等价类的算法时间复杂度为 $O(|C||U|)$，约简中把正区域计算转化为非正区域计算，并在后续算法中以全局等价类作为基本计算单位，计算域从 $|U|$ 降为 $|U/C|$，此思路与文献[13]的相同。与文献[14]思路相似，葛浩等[23]把正区域计算转化为冲突域计算，并以全局等价类压缩决策系统，求核时间复杂度和空间复杂度降为 $O(|C|^2|U/C|)$ 和 $O(|U|/C)$，但需要遍历全部属性和多次遍历全部实体。赵洁等[25]提出一种全局正区域不一致性检测方法，一旦发现引起不一致的实体就立即停止搜索，无需遍历全部实体，使求核时间复杂度下降至 $O(|U||C|^2)-O(\frac{|C||U||Core(A)|}{2})$，但在无核情况下，需遍历全部实体。

在现有研究的基础上，本文提出一种全新的渐增式求核算法。首先，以全局等价类为最小粒度，提出粗等价类概念并分析其性质，研究在粗等价类粒度下决策系统的求核与约简，使计算域从 $|U|$ 降低为 $|U/C|$。然后，把粗等价类细分为 3 类，并深入研究其性质及其与核属性的内在关联，把求核的过程进一步分解为细粒度计算，设计 0-粗等价类下识别核属性的等价方法和渐增式求核方法，可在一次计算中求得多个非核属性，从而提出横向和纵向的剪枝策略。在每次增量计算中，通过横向剪枝策略删减 1 和－1 粗等价类，有效缩减计算域，使用纵向剪枝策略，可删减非核属性，因此可有效压缩计算域。即使在无核情况下，剪枝策略仍有效。基于上述研究，设计多次 Hash 的属性渐增划分算法，最后给出完整的渐增式求核算法。

本文第 2 节介绍 Rough 集的基本概念；第 3 节给出粗等价类的定义和相关性质以及粗等价类下的约简概念；第 4 节设计基于 Hash 的粗等价类算法；第 5 节给出基于粗等价类的核属性的等价计算方法和渐增式求核方法，证明横向和纵向剪枝策略的有效性；第 6 节给出完整的双向剪枝多次 Hash 渐增式求核算法；第 7 节使用 UCI 中 20 个决策表及超高维和海量数据集进行验证，并分析实验结果；最后对全文进行总结。

2　粗糙集的基本概念

本节给出粗糙集的基本概念，详见文献[1]。

定义 1(决策系统)　决策系统 $S=(U,A,V,f)$，其中 $U=\{x_1,x_2,x_3,\cdots\}$ 是论域，是实体的集合。A 是属性集，$A=\{a_1,a_2,\cdots,a_m\}$，$A=C\cup D$ 且 $C\cap D=\emptyset$，C 称为条件属性，D 称为决策属性。V 是属性的值域，$V=\{V_{a1},V_{a2},\cdots,V_{am}\}$。$f$ 是一个信息函数 $f:U\times A\in V$。$f(x,a)$ 通常也记为 $a(x)$。假设 $P=\{a_1,a_2,\cdots,a_k\}\subseteq A$，$(a_1(x),a_2(x),\cdots,a_k(x))$ 记为 $P(x)$。

定义 2(不可区分关系)　对 $P\subseteq A$，$Ind(P)=\{(x_i,x_j)|P(x_i)=P(x_j)\}$ 称为不可区分关系，或等价关系，表示关于属性集 P 是不可区分的。根据 $IND(P)$ 可导出一个等价划分 U/P，该划分中包含对象 x 的等价类记为 $[x]_P$。

定义 3(正区域)　决策系统 $S=(U,C\cup D,V,f)$，令 P，$Q\subseteq C$，$POS_P(Q)=\{x|[x]_P\subseteq[x]_Q\}$，$POS_P(Q)$ 称为 P 相对于 Q 的正区域。$POS_C(D)$ 称为全局正区域。

定理 1[26]　$POS_P(Q)=\bigcup_{Y\in U/P\wedge|Y/Q|=1}Y$

定理 1 是求正区域的等价方法，当 $\forall x_i,x_j\in[x]_P$，若 $Q(x_i)=Q(x_j)$，即可得到 $[x]_P\subseteq POS_P(Q)$，可简化计算。

定义 4(约简)　在决策系统 $S=(U,C\cup D,V,f)$ 中，若 $R\subseteq C$，$POS_R(D)=POS_C(D)$，且对于 $R'\subset R$，都有 $POS_{R'}(D)\neq POS_C(D)$，则称 R 是 C 相对于 D 的属性约简，记为 $R=Red(C)$。

定义 5(核)　在决策系统 $S=(U,C\cup D,V,f)$ 中，C 中所有不可省略属性的集合称为 C 的核，即 $Core(C)=\cap Red(C)$。

3　粗等价类性质及粗等价类下的求核与约简

3.1　粗等价类的定义及性质

定义 6(全局等价类及特征)　基于定义 2，$Ind(C)$ 可导出一个等价划分 $U/C=\{e_1,e_2,\cdots,e_k\}$，其中 e 是基于条件属性 C 的等价类，称为全局等价类，定义其 2 个特征。

1)$e.count$：等价类 e 中的实体 x 的个数被记为 $e.count$；

2)$cons$ 和 dec：对于 $x,y\in e$，显然 $C(x)=D(y)$，若同时 $D(x)=D(y)$，则 $e.cons=true$，$e.dec=D(x)$；否则，若存在 x，$y\in e$，但 $D(x)\neq D(y)$，则 $e.cons=false$，$e.dec=$'/'，表示 e 的 D 值不存在或无意义。

定义 7(基于 S 全局等价类的决策系统)　$S^G=(U^G,C\cup D,V,f^G)$，其中 $U^G/C=\{e_1,e_2,\cdots,e_k\}$，$f^G$ 是 $U^G\times(C\cup D)$ 到 V 的映射，设 $x\in e\in U^G$，其中 $C(e)=C(x)$，$D(e)=e.dec$。称 S^G 是基于 S 全局等价类的决策系统。

同时，对于任意的属性集 $P\subseteq C$，若 $x\in e$，定义 $P(e)=P(x)$。

定义 8(粗等价类)　给定属性集 $P\subseteq C$，若 $e\in U^G$，则 $E^P=[e]_P=\{e'|P(e')=P(e)\}$ 被称作 P 的粗等价类 $RIND$，$U^G/P=\{E_1^P,E_2^P,\cdots,E_n^P\}$，$E^P$ 称为 P 的粗等价类。

定义 8 说明，在决策系统 S^G 中，属性集 $P\subseteq C$，U^C 中的元素 $\{e_1,e_2,\cdots,e_k\}$ 可以被合并为几个大的集合 $\{E_1^P,E_2^P,\cdots,E_n^P\}$，$e$ 是 S^G 中粒度最小的单位。若 $P\subseteq C$，根据 $e.cons$，P 的粗等价类可以分为 3 类，同时定义粗等价类的 2 个特征 $cons$ 和 dec：

1)若 $\forall e,e'\in E^P$，均有 $e.cons=e'.cons=$ true，且 $e.dec=d.dec$，记 $E^P.cons=1$，$E^P.dec=e.dec=e'.dec$，称 E^P 为 P 的 1-粗等价类；

若 $\forall e\in E^P$，均有 $e.cons=false$，记 $E^P.cons=-1$，$E^P.dec='/'$，表示 $D(E^P)$ 不存在或无意义，称 E^P 为 P 的 1-粗等价类.

2)否则，如果 E^P 不满足上述两种情况，则称 E^P 为 P 的 0-粗等价类。

定义 9　S^G 中 D 的 C-正区域在 $S^G=(U^G,C\cup D,V,f^G)$ 中，D 的 C-正区域 $POS_C^G(D)=\{e|[e]_P\subseteq[e]_D\}$。

定理 2　S^G 中，令 $P\subseteq C$，$\forall e\in E^P$，$\forall x\in e$，则 $x\in POS_P(D)\Leftrightarrow e\in POS_C^G(D)$

(1) 先证若 $x\in POS_P(D)$，则 $e\in POS_P^G(D)$。任取 $e'\in[e]_P=E^P$，满足 $P(e)=P(e')$，取 $\forall x\in e$，取 $\forall y\in e'$，由已知 $x,y\in POS_P(D)$，根据定理 1 有 $D(x)=D(y)$，根据定义 6

有 $D(e)=D(x)$，$D(e')=D(y)$，由上面推理得到 $e'\in[e]_D$，即对 $\forall e'\in[e]_P$，有 $e'\in[e]_D$，因此 $[e]_P\in[e]_D$，从而根据定义 9，$e\in POS_P^G(D)$。证毕。

(2)再证若 $e\in POS_P^G(D)$，则 $x\in POS_P(D)$。取 $\forall x,y\in e$，则 $P(x)=P(y)$，由已知 $e\in PSO_P^G(D)$，$[e]_p\subseteq[e]_D$，e 的 D 值存在，根据定义 6，有 $D(e)=D(x)=D(y)$，即 $\forall x,y\in e$，有 $D(x)=D(y)$，根据定理 1，$x\in POS_P(D)$。证毕。

定理 2 说明，S 与 S^G 中的正区域是对应的，若任意 $x\in e$，当 $x\in POS_P(D)$，则必有 $e\in POS_P^G(D)$，而当 $\forall e\in POS_P^G(D)$，若 $x\in e$，则必有 $x\in POS_P(D)$。

定理 3 在 $S^G=(U^G,C\cup D,V,f^G)$ 中，令 $P\subseteq C$，则 $POS_P^G(D)=\bigcup_{|E^P/D|=1}^{P}E_{|E^P/D|=1}$

证明：(1)先证 $\forall e\in\bigcup_{|E^P/D|=1}E^P$，有 $e\in POS_P^G(D)$。取 $\forall e,e'\in\bigcup_{|E^P/D|=1}E^P$，同时取 $\forall x\in e$，$\forall y\in e'$，有 $P(x)=P(e)=P(e')=P(y)$，已知 $|E^P/D|=1$，有 $D(x)=D(e)=D(e')=D(y)$，即 $\forall x,y$，均有 $P(x)=P(y)$ 且 $D(x)=D(y)$，根据定理 1，$x,y\in POS_P(D)$，根据定理 2，$e,e'\in POS_P^G(D)$。证毕。

(2)再证 $\forall e\in POS_P^G(D)$，均有 $e\in\bigcup_{|E^P/D|=1}E^P$。$\forall e\in POS_P^G(D)$，必定存在 E^P，使得 $e\in E^P$，由定义 9 知 $[e]_P\subseteq[e]_D$，取 $\forall e'\in[e]_P$，有 $D(e)=D(e')$，即 $E^P/D=E^P$，故 $|E^P/D|=1$，因此 $e\in\bigcup_{|E^P/D|=1}E^P$。证毕。

与定理 1 作用相似，定理 3 是用于 S^G 中求正区域的等价算法，$\forall e,e'\in E^P$，若 $e.dec=e'.dec$，则可判断 $E^P\in POS_P^G(D)$。

定理 4 在 $S^G=(U^G,C\cup D,V,f^G)$ 中，令 $P\subseteq C$，则 $POS_P^G(D)=\bigcup_{E^P.cons=1}E^P$

证明：已知 $E^P.cons=1$，根据定义 8，$\forall e,e'\in E^P$ 均有 $e.dec=e'.dec$，即 $D(e)=D(e')$，所以 $|E^P/D|=1$，即当 $E^P.cons=1$，必有 $|E^P/D|=1$，根据定理 3，上式成立。证毕。

定理 4 说明，基于粗等价类定义，正区域的计算可得到进一步简化，对 $\forall E^P.cons=1$，即可判断 E^P 属于 P 正区域。

推论 1 $POS_C^G(D)=\{e|e.cons=\text{true}\}$。

根据定理 4，上式成立。

3.2 粗等价类下的求核与约简

定理 5 在 $S^G=(U^G,C\cup D,V,f^G)$ 中，令 $P,Q\subseteq A$，则 $POS_P(D)=POS_Q(D)\Leftrightarrow POS_P^G(D)=POS_Q^G(D)$。

(1)先证 $POS_P(D)=POS_Q(D)\Rightarrow POS_P^G(D)=POS_Q^G(D)$，设 $x\in e\in POS_P^G(D)$，已知 $\forall x\in POS_P(D)$，$POS_P(D)=POS_Q(D)$，有 $x\in POS_Q(D)$，根据定理 2，$e\in POS_P^G(D)$，即 $POS_P^G(D)\subseteq POS_Q^G(D)$。

同理，$POS_Q^G(D)\subseteq POS_P^G(D)$，定理成立。证毕。

(2)再证 $POS_P^G(D)=POS_Q^G(D)\Rightarrow POS_P(D)=POS_Q(D)$。$\forall e\in POS_P^G(D)$，对于 $\forall x\in e$，根据定理 2，均有 $x\in POS_Q(D)$。已知 $POS_Q^G(D)=POS_P^G(D)$，故 $e\in POS_Q^G(D)$，因此 $x\in POS_Q(D)$，$POS_P(D)\subseteq POS_Q(D)$。

同理，$POS_Q(D)\subseteq POS_P(D)$，定理成立。证毕。

定理 5 说明，S 和 S^G 正区域存在映射关系。

定理 6 在 $S^G=(U^G,C\cup D,V,f^G)$ 中，$\forall R\subseteq C$，若有 $POS_R^G(D)=POS_C^G(D)$，且对于 $\forall R'\subset R$，都有 $POS_{R'}^G(D)\neq POS_C^G(D)$，则 R 是 S 中 C 相对于 D 的属性约简，记为 $R=Red(C)$。

证明定理 6，则要证明 $POS_P(D)=POS_Q(D)\Leftrightarrow POS_P^G(D)=POS_Q^G(D)$，根据定理 5，定理得证。

基于上述分析可知，对 S^G 求约简与对 S 求约简是完全等价的，从而根据属性核定义，由于 S^G 与 S 的约简等价，可知在 S^G 上求核与在 S 上求核等价。

4 基于 Hash 的粗等价类算法

粗等价类是本文算法对于所有计算的基础，基于全局等价类的定义，本节先设计全局等价类下基于 Hash 的压缩决策表算法，然后给出粗等价类算法，后续例子使用决策表 1[16] 中的数据。

表 1 决策表

U	a_1	a_2	a_3	a_4	a_5	D
x_1	0	1	1	0	1	1
x_2	0	1	1	0	1	0
x_3	0	1	1	1	1	0
x_4	0	1	1	1	1	1
x_5	0	0	0	1	0	0
x_6	0	0	1	1	0	0
x_7	1	1	0	1	1	1
x_8	1	1	0	1	1	0
x_9	1	1	0	0	1	1
x_{10}	1	0	1	0	1	0
x_{11}	1	0	0	1	1	0
x_{12}	1	0	0	1	1	0

算法 1 计算属性集 P 的等价类 U/P

输入：S=(U,C∪D,V,f)，P(P⊆C)

输出：SG=(U^G,C∪D,V,f^G)

1. 为 P 初始化一个 Hash 表 H
2. 对每一个 x∈U，key=P(x)，执行以下步骤
   2.1. 若 H 中无当前 key
   则创建 h，h. count=1，h. cons=true，h. dec=D(x)
   2.2. 若 H 中存在 h
   2.2.1. 获取对应 h，h_k. count++
   2.2.2. 若 h. cons=true 且 D(x)≠h. dec，则 h. cons=false
3. 返回 H

根据算法 1，当 $P=C$ 时，可求基于 S 全局等价类的压缩决策表 S^G，它是一个 Hash，其子项即为全局等价类，全局等价类是各种计算的基本单位，只进行一次全局计算。Hash 表 H 中的每个分项 h，3 个属性 $h.cons$，$h.count$ 统计 h_k，$h_k.dec$ 如定义 6 中的含义。每个 a 按算法 1 中步骤 2 遍历所有实体，时间复杂度为 $O(|U|)$。使用 Hash 方法，执行 *put* 和 *get* 方法的时间复杂度均为 $O(|key|)$，故步骤 2.1 和步骤 2.2 的时间复杂度均为 $O(|key|)$，步骤 2 的时间复杂度为 $O(|U|)\times O(|key|)=O(|P||U|)$，最差情况下与多个文献[14,23,24,27]相同，优于刘少辉[11]的 $O(|A||U|\log|U|)$。空间复杂度也与多个文献[11,23,24]相同，优于 Hu[27] 的 $O(|U|+p|C|)$。

算法 2 计算属性集 P 的粗等价类 U^G/P

输入：H_C，P(P⊂C)// H_C 是 S 的全局等价类 Hash

输出：U^G/P

1. 为 P 初始化一个 Hash 表 H_P
2. 对每一个 e∈H_C，key=P(e)，执行以下操作
   2.1. 若 H_P 中无当前 key
   2.1.1. 创建 h，h=h∪{e}，h. dec= e. dec，h. cons=−1
   2.1.2. 若 e. cons=true，则 h. cons=1，否则 h. cons=−1
   2.2. 若 H_P 中存在对应 key
   2.2.1. 若获取对应 h，h= h∪{e}，h. count++

2.2.2. 若 h.cons=1 &&e.cons=true && h.dec= e.dec，转步骤 2

2.2.3. 若 h.cons=−1 &&e.cons =false，转步骤 2

2.2.4. 否则令 h.cons=0，h.dec='/'

3. 返回 H_P

算法 2 基于算法 1 的计算结果，步骤 2 遍历 U^G，每次需要形成属性集合 P 的 *key*，时间复杂度为 $O(|P||U/C|)$。算法 1 对表 1 进行计算所得的全局等价类如表 2 所列。

表 2　决策表 1 的全局等价类在算法 1 下的结果

IND	e.key	e.cons	e.dec	e.count	对应实体
e_1	[01101]	false	/	2	x_1、x_2
e_2	[01111]	false	/	2	x_3、x_4
e_3	[11011]	false	/	2	x_7、x_8
e_4	[10011]	true	0	2	x_{11}、x_{12}
e_5	[10101]	true	0	1	x_{10}
e_6	[11001]	true	1	1	x_9
e_7	[00110]	true	0	1	x_6
e_8	[00010]	true	0	1	x_5

设 $P=\{a_1,a_3\}$，根据算法 2 计算得到粗等价类如表 3 所列。

表 3　属性集 $P=\{a_1,a_3\}$ 在算法 2 下的结果

RIND	E.key	E.cons	E.dec	对应 e
E_1	[0 * 1 * *]	0	/	e_1，e_2，e_7
E_2	[1 * 0 * *]	0	/	e_3，e_4，e_6
E_3	[1 * 1 * *]	1	0	e_5
E_4	[0 * 0 * *]	1	0	e_8

5　粗等价类下求核的等价方法

5.1　粗等价类下求核的增量式计算

5.1.1　基于 0-粗等价类的核/非核属性计算方法

在决策系统 S^G 中，$P\subseteq C$，由粗等价类分类可得到：

$$U^R/P=\bigcup_{E_i^P.cons=1}E_i^P\cup\bigcup_{E_j^P.cons=-1}E_j^P\cup\bigcup_{E_k^P.cons=0}E_k^P \tag{1}$$

定理 7　在 S^G 中，$U^G/C=\bigcup_{E_i^C.cons=1}E_i^C\cup\bigcup_{E_j^C.cons=-1}E_j^C$，即 $|\bigcup_{E_k^C.cons=0}E_k^C|=0$。

证明：根据定义 8，$E^C=\{e\}$，即 $|E^C|=1$，而任意 e 的 *cons* 值仅有两种情况：true 或者 false，故 E^C 的 *cons* 值也仅有两种：1 或者 −1，从而 $|\bigcup_{E_k^C.cons=0}E_k^C|=0$ 成立。证毕。

由式(1)可得到：

$$\bigcup_{E_k^P.cons=0}E_k^P=U^G/P-\bigcup_{E_i^P.cons=1}E_i^P-\bigcup_{E_j^P.cons=-1}E_j^P \tag{2}$$

根据定理 7 和上述公式通过 1 和 −1-粗等价类可计算 0-粗等价类大小。下面讨论 $\bigcup_{E_i^P.cons=1}E_i^P$ 和 $\bigcup_{E_j^P.cons=-1}E_j^P$ 的计算。

定理 8　在 S^G 中，$U^G/(C-\{a\})=\bigcup_{E_i^{C-\{a\}}.cons=1}E_i^{C-\{a\}}\cup\bigcup_{E_j^{C-\{a\}}.cons=-1}E_j^{C-\{a\}}\cup\bigcup_{E_k^{C-\{a\}}.cons=0}E_k^{C-\{a\}}$，若 $|\bigcup_{E_k^{C-\{a\}}.cons=0}E_k^{C-\{a\}}|\neq0$，则 $a\in Core(C)$。

证明：已知 $|\bigcup_{E_k^{R-\{a\}}.cons=0}E_k^{C-\{a\}}|\neq0$，必 $\exists E_k^{C-\{a\}}.cons=0$，且 $\exists e.cons=$ true，$e\in E_k^{C-\{a\}}$，根据定理 4，$e\in E_k^{C-\{a\}}\not\subset POS_{R-\{a\}}^G(D)$，但根据定理 2，$e\in POS_C^G(D)$，则必有 $POS_{C-\{a\}}^G(D)\neq POS_C^G(D)$，因此 $a\in Core(C)$。证毕。

定理 9　在 S^G 中，$U^G/(C-\{a\})=\bigcup_{E_i^{C-\{a\}}.cons=1}E_i^{C-\{a\}}\cup\bigcup_{E_j^{C-\{a\}}.cons=-1}E_j^{C-\{a\}}\cup\bigcup_{E_k^{C-\{a\}}.cons=0}E_k^{C-\{a\}}$，若 $|\bigcup_{E_k^{C-\{a\}}.cons=0}E_k^{C-\{a\}}|=0$，则 $a\notin Core(C)$。

同理可证明定理 9。

根据定理 8 和定理 9，得到核属性和非核属性的等价计算方法，即通过计算 0-粗等价类的大小即可判断 a 是否为核属性。下面讨论 $|\bigcup_{E_i^P.cons=1}E_i^P|$ 和 $|\bigcup_{E_j^P.cons=-1}E_j^P|$ 的计算。

5.1.2　属性增量划分下 1-粗等价类的增量式计算

令 $P\subset Q\subseteq C$，下面分析 P 和 Q 的正区域关系。首先在属性簇 $P\subseteq C$ 上定义关系 $\preceq$。

定义 10(关系≥或≤)　令 $P,Q\subseteq C,U/P=\{P_1,P_2,\cdots,P_n\},0<n\leqslant|U|,U/Q=\{Q_1,Q_2,\cdots,Q_n\},0<m\leqslant|U|$ 当且仅当任意 $Q_i\subseteq U/Q$，存在一个 $P_j\subseteq U/Q$，使 $Q_i\subseteq P_j$。

基于定义 1，当满足上述条件时，可以认为属性组 P 比 Q 粒度更粗，Q 粒度比 P 更细。若 $Q\preceq P$ 且 $U/Q=U/P$，则 P 严格粗于 Q(或者 Q 严格细于 P)，此时记为 $Q\succ P$ 或者 $P\prec Q$。

定理 10　在 S^G 中，令 $P=\{R_1,R_2,\cdots,R_n\}$ 是一簇属组，其中 $R_1\geq R_2\geq\cdots\geq R_n(R_n\subseteq 2^C)$，则有 $POS_{R_1}^G(D)\subseteq POS_{R_2}^G(D)\subseteq\cdots\subseteq POS_{R_1}^G(D)$。

证明：若 $E^P\subseteq POS_P^G(D)$，则有 $E^{R_i}\subseteq POS_{R_{i+1}}^G(D)$，即要证明 $\forall e\in E^{R_i}$，均有 $e\in POS_R^G i+1(D)$。

已知 $R_i\geq R_{i+1}$，对于 $\forall E^{R^i}\subseteq POS_R^G i+1(D)$，即定义 1，必定有一个或多个 $E^{R_{i+1}}\subseteq U^G/R_{i+1}$，使得 $E^{R_i}=\bigcup_{|E^{R_{i+1}}/R_i|=1}E^{R_{i+1}}$，即有 $E^{R_{i+1}}\subseteq E^{R_i}$。因此对于 $\forall e\in E^{R_i}$ 必有 $e\in E^{R_{i+1}}$。

在同一 E^Q 中任取 e'，有 $Q(e)=Q(e')$。因为 $E^Q\subseteq E^P$，必有 $e,e'\in E^P$。已知 $E^P\subseteq POS_P^G(D)$，根据定义 8 有 $e.dec=e'.dec$，再根据定理 3，$E^Q\subseteq POS_Q^G(D)$。即 $\forall e\in E^P$，均有 $e\in POS_Q^G(D)$。因此 $E^P\subseteq POS_Q^G(D)$。证毕。

基于上述分析，当 $P\subset Q\subseteq C$，即满足 $Q\prec P$，可判断 P 正区域属于 Q 正区域，而属性 $Q-P$(Q 比 P 多出的部分)对 P 的 0-粗等价类可能划分出新的 1 和 −1-粗等价类，因此 Q 的正区域可能大于 P 的正区域，基于上述分析可知，要求 Q 正区域，可使用 $Q-P$ 在 P 的 0-粗等价类上进行增量计算即可，方法如下。

定理 11　S^G 中，任意属性 $P\subset Q\subseteq C$，则 $POS_Q^G(D)=POS_P^G\cup I^{Q-P}$，其中

$$I^{Q-P}=\bigcup_{E^{Q-P}\subseteq E^P\text{ and }E^{Q-P}.cons=1\text{ and }E^P.cons=0}E^{Q-P}$$

证明：(1)先证明 $\forall e\subseteq POS_Q^G(D)$ 均有 $e\in POS_P^G(D)\cup I^{Q-P}$。

已知 $P\subset Q$，必有 $POS_P^G(D)\subseteq POS_Q^G(D)$，分两种情况讨论：

1) $e\in POS_Q^G(D)$ 且 $e\in POS_P^G(D)$，上式成立。

2) $e\in POS_Q^G(D)$ 且 $e\notin POS_P^G(D)$，此时对于 $\forall e,e'\subseteq POS_Q^G(D)$，由于 $P\subset Q$，有 $(Q-P)e=(Q-P)e'$，必 $\exists E^{Q-P}$ 使得 $e\in E^{Q-P}$，同时使得 $\forall e'\in E^{Q-P}$，必有 $(Q-P)e=(Q-P)e'$，由定义 8 可得 $ec.ons=e''.cons=$ true，$e.dec=e''.dec$，再根据粗等价类分类有 $E^P.cons=1$。

同理，因为 $P\subset Q$，必有 $P(e)=P(e'')$，因此在条件 $e\in POS_Q^G(D)$ 下，当 $e\in E^{Q-P}$，均有 $e\in E^P$，因此 $E^{Q-P}\subseteq E^P$。

由上面证明可知 $e\in E^{Q-P}\subseteq E^P\not\subset POS_P^G(D)$，假设 $E.cons=1$，根据定理 3，必有 $E^P\subseteq POS_P^G(D)$，这与已知矛盾。

若 $E^P.cons=-1$，必有 $e.cons=$ false，与已知 $e\in POS_Q^G(D)$ 矛盾，因此必有 $E^P.cons=0$。

由上证明可得 $e\in I^{Q-P}$。证毕。

(2)再证明 $\forall e\in POS_P^G(D)\subseteq I^{Q-P}$，均有 $e\in POS_Q^G(D)$。

已知，$\forall e\in POS_P^G(D)\subseteq I^{Q-P}$，分两种情况讨论：

1)若 $e\in POS_P^G(D)$，由定理 7，必有 $e\in POS_Q^G(D)$。

2)若 $e\in I^{Q-P}$，由已知 $\forall e,e'\in E^{Q-P}\in I^{Q-P}$ 且 $E^{Q-P}\subseteq E^P$，则 $\forall e,e'\in E^P$；由于 $P(e)=P(e')$ 且 $(Q-P)e=(Q-P)e'$，有 $Q(e)=Q(e')$，故 $E^{Q-P}\subseteq U^G/Q$；已知 $E^{Q-P}.cons=1$，由定理 4，$E^{Q-P}\subseteq POS_Q^G(D)$。证毕。

根据定理 8 和定理 9，若 $A_1\subset A_2\subset\cdots\subset A_i\subset A_n\subseteq C$，则有 $\bigcup\limits_{E^A.cons=1}E^A$，则计算可转化为以下增量式计算：

$$\begin{cases}\bigcup\limits_{E^{A_{i+1}}.cons=1}E^{A_{i+1}}=\bigcup\limits_{E^A.cons=1}E^{A_i}+I^{A_{i+1}-A_i}, & i\geqslant 1\\ I^{A_1-A_0}=\varnothing, & A_0=\varnothing\end{cases}\tag{3}$$

其中，

$$I^{A_{i+1}-A_i}=\bigcup_{E^{A_{i+1}}\subseteq E^{A_i}\text{ and }E^{A_{i+1}-A_i}.cons=1\text{ and }E^{A_i}.cons=0}E^{A_i}$$

5.1.3 属性增量划分下－1-粗等价类的增量式计算

与 1-粗等价类的增量式计算相似，下面给出相关证明。

定理 12 在 S^G 中，$P\subset Q\subseteq C$，若 $E^P.cons=-1$，则 $E^P\not\subset POS_Q^G(D)$。

证明：已知 $E^P.cons=-1$，根据定义 8，$\forall e\in E^P$，有 $e.cons=false$，对于任意的 E^P，均存在一个或多个 E^P，使得 $E^P=\bigcup\limits_{|E^Q/P|=1}E^Q$ 取 $\forall e,e'\in E^P$ 均有 $e.cons=e'.cons=false$，则 $D(x)=D(x')$ 不成立。根据定理 3，$E^Q\not\subset POS_Q^G(D)$，因此 $\bigcup\limits_{|E^Q/P|=1}E^Q=E^P\not\subset POS_Q^G(D)$。

定理 12 说明，所有被 P 划分出来的－1-粗等价类都不可能成为 Q 的正区域，计算公式及证明如下。

定理 13 在 S^G 中，$P\subset Q\subseteq C$，则 $\bigcup\limits_{E^Q.cons=1}E^Q=\bigcup\limits_{E^P.cons=1}E^P\cup N^{Q-P}$，其中：

$$N^{Q-P}=\bigcup_{E^{Q-P}\subseteq E^P\text{ and }E^{Q-P}.cons=-1\text{ and }E^P.cons=0}E^{Q-P}.$$

证明与定理 12 相似，根据定理 13，同理 $\bigcup\limits_{E^A.cons=-1}E^A$ 可增量计算：

$$\begin{cases}\bigcup\limits_{E^{A_{i+1}}.cons=1}E^{A_{i+1}}=\bigcup\limits_{E^A.cons=1}E^{A_i}+N^{A_{i+1}-A_i}, & i\geqslant 1\\ N^{A_1-A_0}=\varnothing, & A_0=\varnothing\end{cases}\tag{4}$$

其中：

$$N^{A_{i+1}-A_i}=\bigcup_{E^{A_{i+1}}\subseteq E^{A_i}\text{ and }E^{A_{i+1}-A_i}.cons=-1\text{ and }E^{A_i}.cons=0}E^{A_i}$$

5.1.4 属性增量划分 0-粗等价类的核/非核属性增量式计算

$$\begin{cases}\bigcup\limits_{E^{A_0}.cons=0}E^{A_0}=U^G, & A_0=\varnothing\\ \bigcup\limits_{E^{A_1}.cons=0}E^{A_1}=U^G-\bigcup\limits_{E^{A_1}.cons=1}E^{A_1}-\bigcup\limits_{E^{A_1}.cons=-1}E^{A_1}\\ \bigcup\limits_{E^{A_{i+1}}.cons=0}E^{A_{i+1}}=U^G-I^{A_{i+1}-A_i}-N^{A_{i+1}-A_i}, & i\geqslant 1\end{cases}\tag{5}$$

根据式(3)、式(4)可得，求核计算可转化为一种增量式计算，最后通过判断 0-粗等价类的大小，即可根据定理 8 或定理 9 判断属性是否为核属性。

5.2 粗等价类下的双向剪枝策略

根据定理 10 和定理 12，$P\subset Q\subseteq C$，所有 P 的 1 和－1-粗等价类具有传递性，保留 Q 的 1 和－1-粗等价类，因此计算中仅保留 P 的 0-粗等价类，基于此设计双向剪枝策略，每次计算均可缩减搜索空间。下面给出剪枝的定义。

定义 11(剪枝) 求核属性的过程中，根据定理 10、定理 12 和式(5)，不断缩减计算域的方法称为剪枝。

本文提出两种剪枝策略，对实体的删减称为横向剪枝，对属性的删减成为纵向剪枝。

横向剪枝策略 1：增量式求正区域时，根据定理 10，删除所有 1-粗等价类。

横向剪枝策略 2：在增量式求约简时，根据定理 12 删除所有－1-粗等价类。

在增量式计算中，忽略当前属性 a_i，把 $a_j\in C-\{a_i\}$ 逐个加入划分属性集，同时删减 1 和－1-粗等价类，求解到第 a_j 个属性时若计算域 $U'=\varnothing$，则 $O=\{a_i,a_{j+1},a_{j+1}\cdots a_n\}$ 可判断为非核属性，此时根据定理 7，$POS_C^G(D)=POS_{C-O-\{a_i\}}^G(D)$，根据定理 9，在集合 O 中的属性都为非核属性。

根据上述分析可得到纵向剪枝策略。

纵向剪枝策略：在增量式求属性的核时，根据定理 9 将非核的属性删除。根据纵向剪枝策略，无需遍历全部属性即可完成求核。

5.3 粗等价类下不一致决策系统的处理

定义 12(不一致决策) 在决策系统 S 中，若 $\exists x_i,x_j\in U,[x_i]_C=[x_j]_C$ 且 $D(x_i)\neq D(x_j)$，则称 S 是不一致的，否则 S 是一致的。

由定义 6 可知，当 S 不一致时，必 $\exists e,e.cons=$false。由定理 10 可知，任意包含约简的属性集合(包括 C)可把所有不一致对象完整划分到－1-粗等价类中，因此不需要增加额外处理，本文求核算法即可用于此类不一致决策系统。

区别于现有方法，本文设计渐增式求核算法，把 $C-\{a\}$ 中的属性逐个加入进行计算，可在一次计算中同时识别多个非核属性，计算中通过双向剪枝策略使遍历属性和实体数不断减少，从而使计算量大大下降，下面给出相关算法。

6 粗等价类下双向剪枝多次 Hash 求核算法

根据前面的多个定理和式(5)，设计多次 Hash 完成渐增计算，每次计算使用双向剪枝策略。

6.1 多次 Hash 的属性增量划分算法

本文的求核基于前述分析中的增量式计算，下面首先给出多次 Hash 的增量划分算法。

算法 3 属性集 Q 相对属性集 P 粗等价类的增量划分

输入：$U^G/P=\bigcup\limits_{E^P.cons=0}E^P$，属性集合 Q(P⊂Q，若 P＝∅，则 $U^G/P=U^G$)// U^G/P 只保留 0-粗等价类，是一个 Hash

输出：U^G/Q

1. 创建一个 hashH，$U'=U^G/P$
2. 依次对 $\forall E^P\subseteq U'$ 执行以下操作//遍历 P 的 0-粗等价类
   2.1. T＝ Q－P，为 E^P 初始化一个 hash H^T//用 Q－P 对 P 的粗等价类进行增量划分
   2.2. 对 $\forall e\in E^P$，执行下列操作
      2.2.1. 令 key＝T(e)，在 H^T 中查找
      2.2.2. 若 H^T 中无对应 key
         2.2.2.1. 创建 h^T，令 $h^T.key=T(e)$，$h^T.dec=e.dec$
         2.2.2.2. 若 e.cons＝true，则 $h^T.cons=1$，否则 $h^T.cons=-1$
      2.2.3. 若 H^T 中已存在对应 key
         2.2.3.1. 获取对应 h^T，若 $h^T.cons=1$ && e.cons＝true && $h^T.cons.dec=e.dec$，转步骤 2.2
         2.2.3.2. 否则，若 $h^T.cons=-1$ && e.cons＝false，转步骤 2.2
         2.2.3.3. 否则令 $h^T.cons=0$，$h^T.dec=$'/'
   2.3. 遍历 H^T，把 H^T 中的粗等价类放入 H 中
3. 返回 H

算法中 Hash 表的子项的 *cons* 和 *count* 的含义如前面定义。

算法 3实质上是根据上面多个定理和式(5)对 0-粗等价类进行增量划分，时间复杂度为 $|\bigcup_{E^P.cons=0}E^P|$，大多数情况下 $|\bigcup_{E^P.cons=0}E^P| \ll |U/C|$，执行 *put* 和 *get* 方式的是时间复杂度为 $O(|key|)$，故 2.2.2 和 2.2.3 的时间复杂度为 $O(|Q-P|)$，算法的时间复杂度为 $O(|Q-P||\bigcup_{E^P.cons=0}E^P|)$，最差情况下为 $O(|Q-P||U/C|)$。

6.2 渐增式求核算法

基于前面多个定理和剪枝策略，给出粗等价类下双向剪枝多次 Hash 求核算法。

算法 4　粗等价类下双向剪枝多次 Hash 求核算法

输入：$S^G=(U^G,C\cup D,V,f)$

输出：Core(C)

1. 初始化 NC =∅，C′=C

2. 对∀ $a_j\in C'$执行以下操作

　2.1. 对∀ $a_j\in C-a_i$ 执行以下操作

　　2.1.1. $U'=U^G$，调用算法 3 (U',a_j)，获取属性 a_j 对 U^G 划分所得到的粗等价类的 Hash$U'/\{a_j\}$，$U'=U'/\{a_j\}$

　　2.1.2. 将 U'的粗等价类中所有 cons=1 和 cons=−1 从 U^G 中删除//横向剪枝

　　2.1.3. 若 $U'=\emptyset$，则 $C'=C'-NC$//纵向剪枝，转步骤 2

　　2.1.4. 否则若 $U'\neq\emptyset$，则跳转步骤 2

3. 返回 Core(C)=C′

现有求核算法一般采用经典方法，每次忽略一个属性 a，然后对 $C-\{a\}$求正区域，通过 $POS_C(D)\neq POS_{C-\{a\}}(D)$判断属性 a 为核属性，此类求核算法的时间复杂度为 $|C|^2|U|$。本文算法中使用双向剪枝策略，U'仅保留 0-粗等价类。步骤 2.1.2 删除搜索空间，即 1-等价类。每增加一个属性后，缩减的空间依次为 $n_1=|\bigcup_{E_i^P.cons=1}E_i^P|+|\bigcup_{E_j^P.cons=-1}E_j^P|$，$n_2=|U^G|-n_1-|\bigcup_{E_k^P.cons=-1}E_k^P|$，当搜索到第 a_j 个属性时，若 $U'=\emptyset$，则停止本次计算。步骤 2 总共执行了 $|C-NC|$次，其中 NC 的数量不确定，与属性次序相关，最坏情况即 $|Red(C)|=|C|$时执行 $|C|$次，其中 $Red(C)$是某一个约简，最理想的情况下为 $|Red(C)|$次。步骤 2.1 总共执行 j 次，j 不确定，最理想情况下执行 $|Red(C)|$次，最坏情况下执行了 $|C|-1$ 次。步骤 2.1.1 对缩减后的 U^G 进行计算，其时间复杂度为 $|U^G-n_1-n_2\cdots-n_j|$，最佳情况下为 $|U^G-n_1-n_2\cdots-n_{|Red|}|$，步骤 2.1.1 每次计算的属性只有 1 个，由算法 3 分析中的 $|key|=1$，得到最坏时间复杂度为 $O(|U^G|)$，最理想的情况下：$O(n)=|Red(C)|^2|U^G-n_1-n_2\cdots-n_{|Red|}|\ll|Red(C)|^2|U/C|<|C|^2|U/C|$，在最坏的情况下等于 $|C|^2|U/C|$。绝大部分情况下，横向剪枝策略生效，使遍历的实体数减少。当决策系统有核时，最理想的情况是 $Core=Red(C)$，而只要 $|Red(C)|<|C|$，无论决策系统是否存在核，纵向剪枝策略生效，使遍历属性数减少。

6.3 数值算例

下面所述为判断表 2 中的 a_1 是否为核属性的过程，如图 1 所示。

步骤 1　首先忽略属性 a_1，使用算法 3，用 $A_1=\{a_2\}$对 U^G 进行划分得到粗等价类 E_1 和 E_2，$U'=\bigcup_{E^{A_i}.cons=0}E^{A_i}$，由于 $E_1.cons=E_2.cons=0$，$U'=U^G$。

步骤 2　使用算法 3，用 $A_2-A_1=a_3$ $(A_2=\{a_2,a_3\})$对 U'的 E_1 和 E_2 进行划分，得到粗等价类 E_{11}，E_{12}，E_{21}，E_{22}。其中 E_{11}，E_{21}，E_{22}符合横向剪枝条件，$U'=\{E_{12}\}$。

步骤 3　使用算法 3，用 $A_3-A_2=a_4$ $(A_2=\{a_2,a_3,a_4\})$对 U'的 E_{12}划分，得到粗等价类 E_{121}和 E_{122}，其全部符合横向剪枝条件。此时 $U'=\emptyset$，算法结束。可以判断 $a_1\notin Core(C)$，$a_5\notin Core(C)$。通过一次计算，根据定理 9 计算得到 2 个非核属性。此时可以执行纵向剪枝策略，$C=C-\{a_1,a_5\}$。

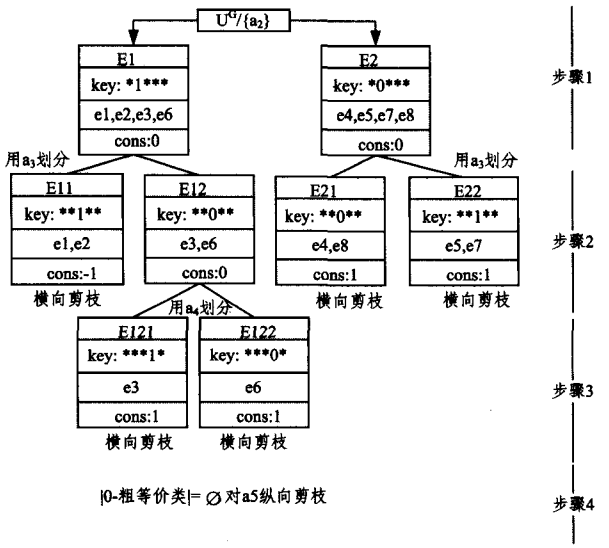

图 1　数值算例

7　实验与结果分析

本文使用 UCI 中的 20 个决策表及海量和超高维数据集进行验证，用本文算法与文献[14-16,25]中的求核算法在求等价类、遍历属性和实体次数以及求核时间上进行对比，计算时间取多次运行的平均值。需要说明的是文献[14]的约简算法中并无求核算法，为了比较基于 Hash 方法的效率，在文献[14]的基于全局等价类计算的思想基础上，按经典求核思路，采用每次忽略一个属性从而判断核属性的方法设计 EHash 求核算法。实验环境为 PC(Intel i7，4 核，2.4GHz，4GB 内存，Win7)，使用 Java 实现所有算法。所使用的 UCI 的决策表如表 4 所列，后续使用表中的 SID 来表示某个决策表。决策表中包含了连续与离散数据，由于离散化方法对约简结果有很大影响，因此本文使用如下方法：某个数值对应一个编码，因此可最大程度保留原决策表的决策能力。

表 4　用于实验的 UCI 决策表

SID	决策表	|U|	|U/C|	|A|	|Core|
1	Arrhythmia	452	452	280	0
2	Australian	690	690	15	1
3	breast1	198	198	35	0
4	breast2	569	569	32	0
5	Car	1728	1728	7	6
6	Credit	690	690	16	1
7	dermatology	366	366	35	0
8	german	1000	1000	21	0
9	letter-recognition	20000	18668	17	3
10	nursery	12960	5784	9	8
11	poker	25010	25008	11	5
12	patient	90	74	9	8
13	shuttle	43500	43500	10	1
14	soybean-large	307	303	36	0
15	tic-tac-toe	985	958	9	0
16	waveform1	5000	5000	22	0
17	wine	178	178	14	0
18	mushroom	8124	8124	23	0
19	zoo	101	59	18	8
20	anneal	798	790	39	1

7.1 等价类算法实验及分析

(1)在 UCI 决策表上实验

各等价类算法对表 4 中决策表求全局等价类的计算时间如表 5 所列,由于文献[14]、文献[25]算法与本文的等价类算法方法均基于 Hash,在计算时间上几乎没有差别,使用"Hash"代表此类等价类算法。文献[15]和文献[16]算法均使用分布基数排序的等价类算法,用"基"表示此类算法。

表 5 各算法求等价类时间比较(ms)

SID	文献[14]	文献[16]	文献[15]	文献[25]	本文	最快
1	3.05	46.8	47.5	2.35	4.6	Hash
2	<0.01	3.15	3.25	0.8	<0.01	Hash
3	<0.01	7	7.8	<0.01	0.75	Hash
4	0.8	50.75	61.6	0.8	0.01	Hash
5	1.6	2.35	1.55	0.8	1.6	/
6	<0.01	3.1	3.85	<0.01	1.55	Hash
7	<0.01	1.55	0.8	0.8	0.75	Hash
8	0.8	7.1	6.25	0.8	0.8	Hash
9	138.15	110.8	106.95	133.55	138.1	基
10	10.85	21.1	23.3	8.35	9.25	Hash
11	546.05	88.2	86.7	582.05	556.1	基
12	<0.01	<0.01	<0.01	0.8	<0.01	/
13	134.95	177.1	184.85	138.1	123.95	Hash
14	6	1.55	0.75	0.01	<0.01	/
15	7.85	0.8	0.8	6.4	8.55	基
16	3.15	367.45	379.65	5.55	5.45	Hash
17	<0.01	1.6	2.35	0.75	<0.01	Hash
18	38.2	31.2	32.7	36.65	36.95	基
19	<0.01	<0.01	<0.01	<0.01	0.75	/
20	<0.01	3.2	3.95	0.8	<0.01	Hash

从表 5 可见,在大部分决策表上,Hash 方法的性能优于基数排序求等价类方法,当决策表实体数与属性个数均较小时,两类方法在效率上无明显差异;但当决策属性个数较多时,如属性个数大于 20 时,除个别决策表外,Hash 方法的优势较为明显。分布基数排序法的时间复杂度为 $O(|C||U|)$,但对每个属性均要遍历 $|U|$ 一次,且每个属性每个实体均要进行多次属性值比较,完成排序后每个实体需按 $|C|$ 个属性对比一次,求全局等价类算法需要访问论域 $|C|+1$ 次。Hash 方法对每个实体的 $|C|-1$ 个属性求 *key*,时间复杂度为 $(|C|-1)|U|$,然后遍历一次论域,求全局等价类仅需遍历论域两次。因此,在求等价类算法上,Hash 的实际运算效率优于基数排序方法。

(2)海量数据和超高维数据集实验

UCI 决策表实验中,在数据量较大的决策表上,Hash 方法和基数排序方法没有明显区别,但不能断言两类方法在海量决策表上的性能相近。下面使用 KDDCup 海量数据集和超高维数据集进行实验,所使用的数据集如表 6 所列。

表 6 海量数据和超高维决策表

SID	决策表	|U|	|U/C|	|A|
21	Kddcup1	500000	366233	42
22	Kddcup2	1000000	582539	42
23	Kddcup3	1500000	731643	42
24	Kddcup4	2000000	731643	42
25	Advertise	3279	2420	1559

各算法在海量和超高维决策表上的求等价类时间比较如表 7 所列。Hash 方法具有较为明显的优势,优于基数方法近 5 倍,在超高维数据集上优于基数方法 10 倍以上。实验数据表明,基于 Hash 的求等价类方法,无论在属性多或少及实体数量多或少的数据集上,均具有较好的性能。

表 7 各算法求等价类的时间比较(ms)

SID	文献[14]	文献[16]	文献[15]	文献[25]	本文	最快
21	2352	10904	10211	1415	1452	Hash
22	4124	21554	20884	4475	4372	Hash
23	6622	31034	31947	6203	6045	Hash
24	9673	43434	44128	9461	9811	Hash
25	114	1835	1849	116	109	Hash

注:由于在海量数据和超高数据上各算法运行时间较长,小数点后两位小数影响较小,鉴于篇幅有限,表 7 运行时间取整表示。

7.2 求核算法实验及分析

(1)剪枝策略验证实验

为深入分析各算法的性能,本实验对本文提出的剪枝策略进行验证,方法如下:对比各算法在求核过程中遍历属性和实体的数量,每次实验均随机改变属性的次序。在 credit 决策表上的 10 次计算结果如表 8 所列。除本文算法外,其他算法均是经典求核思路,需要遍历所有属性,均为 15 个属性。在遍历实体数量上,文献[15]的算法验证每个属性时均需要遍历 $|U|$ 一次,因此遍历的实体总数为 $|C|(|C|-1)|U|=144900$,文献[14]和文献[16]算法对决策表进行压缩,但由于压缩后的实体数等于原始实体数,因此实体数也为 144900,文献[25]算法采用了停止搜索策略,一旦搜索到产生导致全局等价类冲突的实体,即停止当前计算,因此遍历实体次数较少,文献[25]算法遍历实体次数依赖于实体顺序,因此仅改变属性次序情况下,遍历实体数不改变,均为 139874。本文算法在 credit 上的绝大部分计算中均可求得多个非核属性,通过双向剪枝策略,所需要遍历的实体数和属性数均大大缩减。

表 8 各算法在 credit 决策表上遍历次数比较

No	文献[14]		文献[16]		文献[15]		文献[25]		本文算法	
	属性数	实体数	属性数	实体数	属性数	实体数	属性数	实体数	属性数	实体数
1	15	144900	15	144900	15	144900	15	139874	4	5781
2	15	144900	15	144900	15	144900	15	139874	4	6389
3	15	144900	15	144900	15	144900	15	139874	6	8597
4	15	144900	15	144900	15	144900	15	139874	15	41482
6	15	144900	15	144900	15	144900	15	139874	6	11746
6	15	144900	15	144900	15	144900	15	139874	7	11587
7	15	144900	15	144900	15	144900	15	139874	8	12762
8	15	144900	15	144900	15	144900	15	139874	5	6359

为进一步说明各算法的差异,定义下面两个指标。

$$AP=\text{平均遍历属性个数}/|C| \quad (6)$$

$$UP=\text{平均遍历实体数}/|\text{论域}| \quad (7)$$

显然,*AP* 和 *UP* 的取值范围均为(0,1]。*AP* 表示在求核过程中遍历属性的平均次数与 $|C|$ 的比值,*UP* 表示遍历的平均实体数与论域大小的比值。*AP* 和 *UP* 具有相同性质:值

越小则算法遍历的属性和实体数越少，效率越高。算法中若使用原始决策系统，则|论域|=|U|；若使用基于全局等价类的决策系统，则|论域|=|U/C|。各算法在随机抽取的多个UCI决策表上的遍历指标如表9所列。文献[14-16]算法所有AP和UP均为1，和相应的时间复杂度|C||(|C|−1)|U|或者|C||(|C|−1)|U/C|吻合。文献[25]算法使用停止搜索策略使实体访问数至少降低5%左右，最多可减少30%，但对于无核决策表如18-mushroom，则无法缩减计算域。本文算法使用双向剪枝，遍历的属性和实体数大大减少，平均遍历的属性数最多减少55%，平均遍历实体数最多可减少90%。从决策表18-mushroom可验证，即使决策表无核，本算法的双向剪枝策略仍生效，无需遍历全部属性和论域，一次可计算出多个非核属性。

表9　各算法在多个决策表遍历指标比较(AP和UP越小，算法效率越高)

No	|U|	|U/C|	|C|	|C|(|C|−1)|U|	|C|(|C|-1)|U/C|	核数	文献[14]		文献[16]		文献[15]		文献[25]		本文	
							AP	UP	AP	UP	AP	UP	AP	UP	AP	UP
6	690	690	15	144900	144900	1	1	1	1	1	1	1	1	0.9653	0.4583	0.0903
9	20000	18668	16	4800000	4480320	3	1	1	1	1	1	1	1	0.9224	0.9219	0.3847
11	25010	25008	10	2250900	2250720	5	1	1	1	1	1	1	1	0.7197	0.9500	0.6158
18	8124	8124	22	3753288	3753288	0	1	1	1	1	1	1	1	1.0000	0.7216	0.1355
20	798	790	38	1121988	1110740	1	1	1	1	1	1	1	1	0.9781	0.6316	0.1512

(2)UCI决策表实验

各算法在UCI决策表上的求核时间如表10所列。若计算机时间相差在10ms以内，则视为相等。在大部分情况下，文献[15]与文献[16]算法效率接近，两者方法的最大差别在于文献[16]使用了基于全局等价类的压缩决策系统，但大部分UCI中的决策表可压缩空间不大，因此文献[16]算法在UCI决策表上并未显示出较明显优势。而决策系统是否可压缩取决于数据本身具有不可预测性。但可合理估计，当决策表的压缩程度越高，EHash、文献[16]算法和本文算法的效率越高。数据压缩比|U/C|/|U|和文献[15]与文献[16]的比值折线图如图2所示。

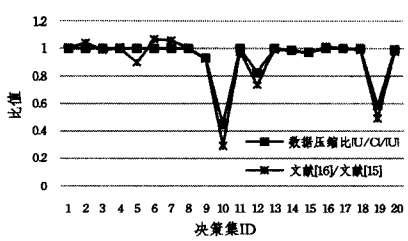

图2　数据压缩比与算法效率关系图

在决策表10,12,19上，数据压缩程度较高，其中决策表10压缩程度最高，同时文献[15,16]算法在此类决策表上的效率差别较大，其中在决策表10上最为明显，图2中数据特征验证了上述分析，数据压缩程度越高，基于数据压缩策略的算法效率越高。

在大部分决策表上，EHash算法优于文献[16]。由于使用停止搜索策略，文献[25]算法具有较佳性能，遍历的实体数量少于文献[15,16]等算法，与EHash算法效率接近，但整体效率稍低于刘算法，在大部分情况下不如本文算法，尤其在无核决策系统上。在决策表1,9,11,13,16,18上，本文效率比较显著地优于其他算法，进而说明本文算法适用于多实体(>8000)和高维度决策表(>100)，说明本文的纵向剪枝策略的有效性。在绝大部分决策表上，本文算法、EHash算法和文献[25]算法效率相近，说明基于Hash的求核算法适用于维度较高(>20)的算法，本文整体上优于EHash和文献[25]算法。由于采用了增量式计算和双向剪枝策略，在大多数情况下，包括在决策系统无核情况下，本文算法均具较佳性能。

表10　各算法求核时间比较(ms)

No	EHash	文献[16]	文献[15]	文献[25]	本文	最快算法
1	378.35	8862.60	8844.75	538	1.90	本文
2	2.55	36.25	34.95	7.70	2.10	本文,EHash,文献[25]
3	3.50	156.65	158.30	8.20	0.10	本文,EHash,文献[25]
4	7.95	476.25	474.65	26.30	0.45	本文,EHash
5	4.60	9.40	10.50	4.30	5.80	/
6	3.25	43.55	40.85	5.80	2.40	本文,EHash,文献[25]
7	5.85	46.90	44.45	11	5.45	本文,EHash,文献[25]
8	7.55	82.90	82.80	14	3.40	本文,EHash,文献[25]
9	1151.70	1739.20	1888.00	1828.60	708.70	本文
10	31.25	80.05	276.45	40.80	58.70	本文,EHash,文献[25]
11	3248.35	905.55	927.50	2980.30	492.25	本文
12	0.10	0.55	0.75	0.30	0.40	/
13	583.40	1638.00	1648.85	991.40	211.30	本文
14	5.250	31.10	31.50	8.40	0.35	本文,EHash,文献[25]
15	25.55	8.7	9.0	30	5.0	本文,文献[16],文献[15]
16	68.3	1507.05	1488.6	94.2	11.05	本文
17	0.4	14.25	14.25	1	0.35	本文,EHash,文献[25]
18	362.35	661.7	667.9	673.7	115.6	本文
19	0.4	1.05	2.15	1.8	1.0	/
20	17.05	122.4	125.3	25.9	10.95	本文,EHash,文献[25]

(3)海量数据和超高维数据实验

在海量和超高维数据上各求核算法的运算时间如表11所列。

表11　海量数据和超高维各算法求核时间比较(ms)

No	EHash	文献[16]	文献[15]	文献[25]	本文	最快算法
21	21752	378573	474881	10915	18814	文献[25]
22	48194	632978	1020473	43167	40994	本文
23	52785	765630	1612256	52823	50233	本文
24	62169	793379	1801586	57192	53589	本文
25	78053	1510889	2816022	455989	53635	本文

注：由于在海量数据和超高数据上各算法运行时间较长，小数点后两位小数影响很小，因篇幅有限，运行时间取整表示。

由于使用全局等价类计算，相对比文献[15]算法，文献[16]算法的性能具有较为明显的优势，计算时间约为文献[15]算法的50%，文献[16]算法的运算时间是EHash的10倍以上，在超高维数据上则为近20倍。EHash方法在速度上明显优于文献[16]算法，但需要遍历全部的属性和数据量为|C|(|C|−1)|U/C|的实体数，因此在计算时间上劣于文献[25]算法，文献[25]算法采用了停止搜索策略，不需要遍历全部实体，因此在海量数据集上的效率优于EHash、文献[16]和文献[15]算法。但在超高维数据集上，EHash效率较大幅

度地优于文献[25]算法。本文算法可一次求得多个非核属性,并采用双向剪枝策略,因此在大部分情况下均优于其他算法,数据量越大优势越明显。在超高维数据上,本文算法亦显示出明显优势,计算时间大大少于其他算法。

结束语 本文提出一种全新的渐增式求核算法。首先以全局等价类为基本计算粒度,并提出粗等价类概念以及多个定理来证明在粗等价类下的求核与约简等价于原始决策系统的求核与约简,使计算空间从$|U|$下降到$|U/C|$,然后深入研究粗等价类的性质与核/非核属性的内在联系,设计基于0-粗等价类的判断核属性的等价方法和渐增式求核方法,把求核过程分解成更小粒度的计算,从而提出双向剪枝策略,其可在每次渐增计算中通过横向剪枝删减实体,通过纵向剪枝减少所需要遍历的属性,不断缩减计算域,从而给出高效求核算法。通过UCI中20个决策表、KDDCup海量数据集和超高维数据集,从等价类算法、求核过程中遍历属性和实体的次数、求核时间等多个方面对本文提出的多种策略与多个算法进行对比和验证,本文算法在大多数情况下,无论是属性多或少,实体数多或少,均具有较好的性能,即使在决策表无核的情况下,剪枝策略仍然生效,无需遍历全部属性和全部实体,具有良好的适应性。本文算法可作为新型约简算法和优化算法的基础,这也是下一步将进行的研究工作。

参 考 文 献

[1] PAWLAK Z. Rough sets[J]. International Journal of Computer & Information Sciences, 1982, 11(5): 341-356.

[2] WANG J, WANG R, MIAO D Q. Data enriching based on rough set theory[J]. Chinese Journal of Computers, 1998, 21(5): 393-400. (in Chinese)
王珏,王任,苗夺谦. 基于 Rough Set 理论的"数据浓缩"[J]. 计算机学报,1998,21(5):393-400.

[3] WANG C, HE Q, CHEN D, et al. A novel method for attribute reduction of covering decision systems[J]. Information Sciences, 2014, 254: 181-196.

[4] WANG C, SHAO M, SUN B, et al. An improved attribute reduction scheme with covering based rough sets[J]. Applied Soft Computing, 2015, 26: 235-243.

[5] Tsang E C, Chen D, Yeung D S, et al. Attributes reduction using fuzzy rough sets[J]. IEEE Transactions on Fuzzy Systems, 2008, 16(5): 1130-1141.

[6] ZHANG Z, TIAN J. On Attribute Reduction with Intuitionstic Fuzzy Rough Sets[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2012, 20(1): 59-76.

[7] CHEN D, LI W, ZHANG X, et al. Evidence-theory-based numerical algorithms of attribute reduction with neighborhood-covering rough sets[J]. International Journal of Approximate Reasoning, 2014, 55(3): 908-923.

[8] FENG T, ZHANG S, MI J. The reduction and fusion of fuzzy covering systems based on the evidence theory[J]. International Journal of Approximate Reasoning, 2012, 53(1): 87-103.

[9] QIAN Y, LIANG J, PEDRYCZ W, et al. An efficient accelerator for attribute reduction from incomplete data in rough set framework[J]. Pattern Recognition, 2011, 44(8): 1658-1670.

[10] ZHENG K, HU J, ZHAN Z, et al. An enhancement for heuristic attribute reduction algorithm in rough set[J]. Expert Systems with Applications, 2014, 41(15): 6748-6754.

[11] LIU S H, SHENG S J, WU B, et al. Research on Efficient Algorithms for Rough Set Methods[J]. Chinese Journal of Computers, 2003, 26(5): 524-529. (in Chinese)
刘少辉,盛秋戬,吴斌,等. Rough 集高效算法的研究[J]. 计算机学报,2003,26(5):524-529.

[12] LIU S H, SHENG Q J, SHI Z Z. A New Method for Fast Computing Positive Region[J]. Journal of Computer Research and Development, 2003, 40(5): 637-642. (in Chinese)
刘少辉,盛秋戬,史忠植. 一种新的快速计算正区域的方法[J]. 计算机研究与发展,2003,40(5):637-642.

[13] XU Z Y, LIU Z P, YANG B R, et al. A Quick Attribute Reduction Algorithm with Complexity of max (O(|C||U|), O(|C^2|U/C|))[J]. Chinese Journal of Computers, 2006, 29(3): 391-399. (in Chinese)
徐章艳,刘作鹏,杨炳儒,等. 一个复杂度为 max(O(|C||U|),O(|C^2|U/C|))的快速属性约简算法[J]. 计算机学报,2006,29(3):391-399.

[14] LIU Y, XIONG R, CHU J. Quick Attribute Reduction Algorithm with Hash[J]. Chinese Journal of Computers, 2009(8): 1493-1499. (in Chinese)
刘勇,熊蓉,褚健. Hash 快速属性约简算法[J]. 计算机学报,2009(8):1493-1499.

[15] GE H, LI L S, YANG C J. An Efficient Attribute Reduction Algorithm Based on Conflict Region[J]. Chinese Journal of Computers, 2012, 35(2): 342-350. (in Chinese)
葛浩,李龙澍,杨传健. 基于冲突域的高效属性约简算法[J]. 计算机学报,2012,35(2):342-350.

[16] GE H, LI L S, YANG C J. Attribute Reduction Algorithm Based on Conflict Region Decreasing[J]. Systems Engineering-Theory & Practice, 2013, 33(9): 2371-2380. (in Chinese)
葛浩,李龙澍,杨传健. 基于冲突域渐减的属性约简算法[J]. 系统工程理论与实践,2013,33(9):2371-2380.

[17] HEDAR A, WANG J, FUKUSHIMA M. Tabu search for attribute reduction in rough set theory[J]. Soft Computing, 2008, 12(9): 909-918.

[18] DING Wei-ping, WANG Jing-dong, GUAN Zhi-jin. Efficient rough attribute reduction based on Quantum Frog-leaping co-evolution[J]. Acta Electronica sinica, 2011, 39(11): 2597-2603. (in Chinese)
丁卫平,王建东,管致锦. 基于量子蛙跳协同进化的粗糙属性快速约简[J]. 电子学报,2011,39(11):2597-2603.

[19] TAO Xin-min, WANG Yan, XU Jing, et al. Minimum rough set attribute reduction algorithm based on virus-coordinative discrete particle swarm optimization[J]. Control and decision, 2012, 27(2): 259-265. (in Chinese)
陶新民,王妍,徐晶,等. 求解最小属性约简的病毒协同进化微粒群算法[J]. 控制与决策,2012,27(2):259-265.

[20] DING W, WANG J. A Novel Approach to Minimum Attribute Reduction based on Quantum-inspired Self-adaptive Cooperative Co-evolution[J]. Knowledge-Based Systems, 2013, 50(3): 1-13.

[21] GE H, LI L S, YANG C J. Incremental updating algorithm of the computation of core based on the collision[J]. Control and Decision, 2011, 26(7): 984-990. (in Chinese)

(下转第 258 页)

参 考 文 献

[1] ZENG Jie, NIE Wei. Novel multi-objective optimization algorithm[J]. Journal of Systems Engineering and Electronics, 2014, 25(4): 697-710.

[2] QI Yu-tao, LIU Fang, CHANG Wei-yuan, et al. Memetic Immune Algorithm for Multiobjective Optimization[J]. Journal of Software, 2013, 24(7): 1529-1544. (in Chinese)
戚玉涛, 刘芳, 常伟远, 等. 求解多目标问题的 Memetic 免疫优化算法[J]. 软件学报, 2013, 24(7): 1529-1544.

[3] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGAII[J]. IEEE Transaction on Evolutionary Computation, 2002, 6(2): 182-197.

[4] ZITZLER E, LAUMANNS M, THIELE L. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization[R]. TIK-Report, 103, Swiss Federal Institute of Technology, 2001: 1-21.

[5] DEB K, MOHAN M, MISHRA S. Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions[J]. Evolutionary Computation, 2005, 13(4): 501-525.

[6] COELLO C C A, PULIDO G T, LECHUGA M S. Handing multiple objectives with particle swarm optimization [J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 256-279.

[7] GONG M G, JIAO L C, DU H F, et al. Multi-Objective immune algorithm with nondominated neighbor-based selection[J]. Evolutionary Computation, 2008, 16(2): 225-255.

[8] YANG D D, JIAO L C, GONG M G, et al. Adaptive ranks and K-nearest neighbor list based multiobjective immune algorithm [J]. Computational Intelligence, 2010, 26(4): 359-385.

[9] ZHANG Shi-hai, Ou Jin-ping. BP-PSO-based intelligent case retrieval method for high-rise structural form selection[J]. Science China Technological Sciences, 2013, 56(4): 940-944.

[10] ZHAO Xin-shuang, WANG Hou-xiang, CAI Yi-chao. Resource scheduling method in antimissile early warning campaign[J]. System Engineering and Electronics, 2015, 37(6): 1300-1305. (in Chinese)
赵新爽, 汪厚祥, 蔡益朝. 反导预警作战资源调度方法[J]. 系统工程与电子技术, 2015, 37(6): 1300-1305.

[11] WANG Hua, ZHU Fu-bao. Site selection model of land consolidation projects based on multi-objective optimization PSO[J]. Transactions of the Chinese Society of Agricultural Engineering, 2015, 31(14): 255-263. (in Chinese)
王华, 朱付保. 基于多目标粒子群的土地整理项目选址模型[J]. 农业工程学报, 2015, 31(14): 255-263.

[12] ZHANG En-ze, WU Yi-fei, CHEN Qing-wei. Particle swarm optimization algorithms for interval multi-objective optimization problems[J]. Control and Decision, 2014, 29(12): 2171-2176. (in Chinese)
章恩泽, 吴益飞, 陈庆伟. 一类区间多目标粒子群优化算法[J]. 控制与决策, 2014, 29(12): 2171-2176.

[13] GAN Xu-sheng, DUANMU Jing-shun, MENG Yue-bo, et al. Wavelet neural network aerodynamic modeling from flight data based on pso algorithm with information sharing and velocity disturbance[J]. Journal of Central South University, 2013, 20(6): 1592-1601.

[14] DRECHSLER N, DRECHSLER R, BECKER B. Multi-objective Optimization based on Relation favor[M]// Evolutionary multi-criterion optimization. Springer Berlin Heidelberg, 2001: 154-166.

[15] FARIAN M, AMATO P. A fuzzy definition of "optimality" for many-criteria optimization problems[J]. IEEE Transactions on Systems, Man, and Cybernetics, 2004, 34(3): 315-326.

[16] HERNANDEZ-DIAZ A G, SANTANA-QUINTERO L V, COELLO C A C, et al. Pareto-adaptive ε-dominance[J]. Evolutionary Computation, 2007, 15(4): 493-517.

[17] GAO Hong-min, ZHOU Hui, XU Li-zhong, et al. Classification of hyperspectral remote sensing images based on simulated annealing genetic algorithm and multiple instance learning[J]. Journal of Central South University, 2014, 21(1): 262-271.

[18] LIU Bo, WANG Ling, JIN Yi-hui. An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers [J]. Computers & Operations Research, 2008, 35(9): 2791-2806.

(上接第 234 页)

葛浩, 李龙澍, 杨传健. 基于冲突的增量式核属性更新算法[J]. 控制与决策, 2011, 26(7): 984-990.

[22] YANG M. An Incremental Updating Algorithm of the Computation of a Core Based on the Improved Discernibility Matrix[J]. Chinese Journal of Computers, 2006, 29(3): 407-413. (in Chinese)
杨明. 一种基于改进差别矩阵的核增量式更新算法[J]. 计算机学报, 2006, 29(3): 407-413.

[23] GE H, LI L S, YANG C J. Quick algorithm for computing core attribute[J]. Control and Decision, 2009, 24(5): 738-742. (in Chinese)
葛浩, 李龙澍, 杨传健. 一种核属性快速求解算法[J]. 控制与决策, 2009, 24(5): 738-742.

[24] GE H, LI L S, YANG C J. Efficient algorithm for computing core attributes [J]. Computer Engineering and Applications, 2010, 46(26): 138-141. (in Chinese)
葛浩, 杨传健, 李龙澍. 一种高效的核属性求解算法[J]. 计算机工程与应用, 2010, 46(26): 138-141.

[25] ZHAO Jie, LIANG Jun-jie, DONG Zhen-ning, et al. Global positive region inconsistency based Attributes Core computation [J]. Computer Science, 2015, 42(8): 259-264. (in Chinese)
赵洁, 梁俊杰, 董振宁, 等. 基于全局正域不一致性的快速求核算法[J]. 计算机科学, 2015, 42(8): 259-264.

[26] ZHAO Jie, DONG Zheng-yu, ZHANG Kan-hang. Rough equivalence Class Based Attribute Reduction Algorithm with Bilateral-pruning strategies and Multiple Hashing[J]. Control and Decision, 2016, 3(11): 1921-1934(in Chinese)
赵洁, 董振宁, 张恺航. 粗等价类双边剪枝策略下多次 Hash 的约简算法[J]. 控制与决策, 2016, 3(11): 1921-1934.

[27] HU F, WANG G, XIA Y. Attribute core computation based on divide and conquer method[M]. Rough Sets and Intelligent Systems Paradigms, Springer, 2007: 310-319.