基于并行协同进化的属性约简

王立宏[1),2)] 吴耿锋[1)]

[1)](上海大学计算机工程与科学学院 上海 200072)

[2)](烟台大学计算机工程与技术学院 烟台 264005)

摘 要 提出一种求属性集合最小约简的新方法,即基于并行协同进化的属性约简方法.该方法将并行遗传算法和协同进化算法相结合,能有效地处理具有大量属性的信息系统.对各类实验数据的测试表明,该方法得到的属性约简量与基于属性重要性的约简方法相似,在某些情况下求得最小约简的可能性要高于属性重要性方法.

关键词 属性约简;粗糙集合;协同进化;遗传算法;区分矩阵

中图法分类号 TP18

Attribute Reduction Based on Parallel Symbiotic Evolution

WANG Li-Hong[1),2)] WU Geng-Feng[1)]

[1)](*School of Computer Engineering and Science*, *Shanghai University*, *Shanghai* 200072)

[2)](*School of Computer Engineering and Technology*, *Yantai University*, *Yantai* 264005)

Abstract A new approach to attribute reduction based on parallel symbiotic evolution is proposed. Combining parallel genetic algorithm with symbiotic evolution, this approach can make efficient reduction for an information system with a large number of attributes. In symbiotic evolution, a (full) solution to an optimal problem will be divided into several partial solutions that constitute a population, which is going to evolve to find the optimal solution for each partial solution. Due to the diversity of optimal patterns for partial solutions, the population can maintain diversity and the optimal solutions will be found more promisingly. However, the position information of partial solutions in a full solution does not be used in symbiotic evolution and then an optimal pattern for one position may be settled in another place. The abuse of these optimal patterns leads to a lower fitness value and then threatens the survival of them. Considering the position information as well as the population diversity, this paper proposes an algorithm named Parallel Symbiotic Evolution Algorithm (PSEA). In order to get the minimal attribute reduction, authors encode the attribute subsets into binary strings, cut them into several sections and create a population for each section position. These populations experience evolution in a parallel way, exchanging patterns by the elitist strategy. The experiment tests seven kinds of discernibility matrices and shows that the minimal attribute reduction can be found in some cases with a higher probability than that of other approaches.

Keywords attribute reduction; rough set; symbiotic evolution; genetic algorithm; discernibility matrix

---

收稿日期:2001-07-10;修改稿收到日期:2002-06-12.本课题得到国家自然科学基金(69985004,60275022)和上海市科委基础研究项目(01JC14022)资助.**王立宏**,女,1970 年生,副教授,博士研究生,研究方向为知识发现、智能信息处理和计算机网络体系结构等.E-mail:yeshzh@vip.sina.co.**吴耿锋**,男,1945 年生,教授,博士生导师,研究方向为数据挖掘与知识发现、智能控制、智能信息处理以及 CSCW 等.

1 引　言

在基于粗糙集合的数据分析中,通常将数据集合称为信息系统[1,2].信息系统可以表示为 $S=(U,A)$,其中 $U=\{O_1,O_2,\cdots,O_n\}$ 是对象集合,$A=\{a_1,a_2,\cdots,a_m\}$ 是属性集合,$a_k(O_i)$ 表示对象 O_i 在属性 a_k 上的取值($i=1,2,\cdots,n,k=1,2,\cdots,m$).若 $a_k(O_i)\neq a_k(O_j)$,则称 O_i 与 O_j 是可由属性 a_k 区分的.如果存在属性 $a_k\in A$,使得 O_i 与 O_j 可以由 a_k 区分,则称 O_i 与 O_j 是可由 A 区分的.

通过比较任意两个对象之间不相等的属性,可得出信息系统的区分矩阵[2](又称可辨识矩阵).区分矩阵 $\boldsymbol{M}$ 是 $n\times n$ 的矩阵,元素 M_{ij} 是一个属性集合,可表示为

$$M_{ij}=\{a\mid a(O_i)\neq a(O_j)\wedge a\in A\},\quad i,j=1,2,\cdots,n.$$

显然 $\boldsymbol{M}$ 是对称矩阵,主对角线上的元素全是∅,因此该矩阵只需保存 $|M|=n(n-1)/2$ 个元素.

通常,对象的各属性之间并不是完全独立的,部分属性区分对象的能力与所有属性共同的区分能力可能完全相同.因此,如果能抽取这些有代表性的属性就可以缩减属性的数量,为进一步的知识发现算法减少时间与空间的复杂度.对于任意 $B\subset A$,若信息系统中任意两个可由 A 区分的对象,也可以由 B 区分,则称 B 为 A 的约简,具有最小基数的约简称为 A 的最小约简.

如何从区分矩阵得出属性集合的最小约简是我们感兴趣的问题.人们已经提出几种求最小约简的算法[2~4],如基于正区域的约简算法,基于区分矩阵中属性频率的约简算法、基于信息的约简算法等等.这些算法通过为属性的重要性给出不同的定义来求最小约简,算法本身都是不完备的[2].另外,还有基于区分矩阵和逻辑运算得到最小约简的方法[5],但这种方法在属性较多时需进行大量的逻辑运算.

研究证明,求属性集合的最小约简是 NP-难问题[2].对于 NP-难问题,遗传算法往往能有效地解决.遗传算法的基本出发点是达尔文的"优胜劣汰,适者生存"思想,而在研究生物与环境的关系时[6]发现,物种之间的协调发展是使各物种同时生存下去的重要因素,协同作用往往比竞争更重要.模仿生物种群之间的协调发展来求解最优化问题是协同进化算法的主要思想.文献[7~9]将这种思想应用于神经网络的连接权及网络结构的调整,成功地控制了小车倒立摆[7,9]和机械手的运动[8].本文结合并行遗传算法和协同进化算法的基本思想提出并行协同进化算法,并用该算法求解属性集合的最小约简.实验表明,该算法适用于具有大量属性(如属性数>100)的信息系统,在某些情况下求到最小约简的可能性要大于属性重要性方法.

2 属性约简的表示与评价

用遗传算法求属性集合的最小约简时涉及到染色体的定义和适应值的评价等问题.

遗传算法的种群由染色体 ch(chromosome)组成,每条染色体用 m 位二进制数 $c_1c_2\cdots c_m$ 表示,其中 $m=|A|$.染色体 ch 对应属性集合 A 的一个子集 CH,二者之间可以建立一一对应关系:$c_j=1$ 当且仅当 $a_j\in CH,j=1,2,\cdots,m$.

对染色体适应值的评价是整个进化策略的基础.染色体的适应值应能体现出该染色体表示的属性子集是不是原属性集合的约简,如果是约简,该属性子集中包含的元素越少,适应值应越高.因此本文为每条染色体定义如下的适应值函数:

$$Fitness(ch)=\begin{cases}1+\dfrac{|A|-|CH|}{|A|}, & CH\in RED(A)\\ \dfrac{match(ch,M)}{|M|}, & \text{否则}\end{cases}\tag{1}$$

其中,ch 是一条染色体,CH 是 ch 对应的属性子集.$RED(A)$ 表示属性集合 A 的所有约简组成的集合,$match(ch,M)$ 表示 ch 匹配的区分矩阵 $\boldsymbol{M}$ 中元素的个数.

染色体 ch 与区分矩阵的元素 M_{ij} 匹配是指:ch 对应的属性子集 CH 和 M_{ij} 的交集不为空集,即存在属性 $a\in M_{ij}\cap CH$,a 是 CH 中的元素但可以把 M_{ij} 对应的两个对象 O_i 与 O_j 区分开.ch 与 M_{ij} 匹配说明这两个对象可以由 A 区分,也可以由 CH 区分.

如果一条染色体匹配区分矩阵的所有元素,则信息系统中任意两个可由 A 区分的对象,也可以由该染色体对应的属性子集 CH 区分,即 CH 是 A 的约简,而且 CH 中包含的属性越少越接近最小约简,因此适应值随 CH 中包含的属性个数减少而增加.如果 CH 不是 A 的约简,则 CH 匹配区分矩阵的元素越多,说明 CH 的区分能力越接近 A 的区分能力,因此适应值越高.

如上定义的适应值函数是多峰值的,这是问题本身决定的,因为属性集合的约简有很多,即使是最小约简也不一定唯一.

3 协同进化算法 SEA(Symbiotic Evolutionary Algorithm)

文献[7～9]称最优化问题的一个解为完全解,组成完全解的各个部分称为部分解,如图 1 所示.一个部分解反映问题的一个方面,如一个决策变量的取值.只有当它与其它部分解成功合作,组成的完全解才是最优解.如果在遗传算法中只进化部分解形成的种群,通过部分解的组合来搜索完全解的空间,在处理复杂问题时就能有效控制算法的搜索空间.

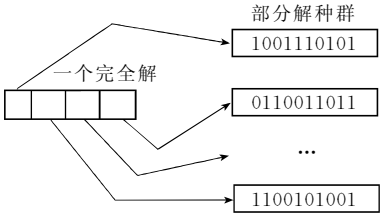

图 1 完全解的构成

协同进化算法是根据以上思想改进简单遗传算法得到的.算法运行之前,先将完全解分解成长度相同的部分解,形成初始的部分解种群,然后开始进化过程:从种群中随机抽取部分解放在完全解的各位置上形成完全解;构造好若干个完全解之后,根据具体问题评价每个完全解,之后评价它的各个部分解;接着在部分解种群中进行选择、交叉、变异等进化操作,然后进行下一轮进化.

部分解的适应值由它参与的完全解的适应值来确定.完全解的适应值高则它的每个部分解的适应值也高.进化算法为每个部分解记录它曾经参与的完全解的适应值,将这些适应值的最大值定义为该部分解当前的适应值.

由于在完全解的不同位置上,部分解的最优模式一般不同,因此部分解种群在进化过程中能保持多样性,有效地防止种群进化到一定代数时,因全部染色体完全相同而停留在某个局部最优解[7],这是协同进化算法的优点.但是算法不考虑部分解位置信息,可能将一个位置上的最优模式安排到其它位置上,这样降低了完全解的适应值,从而影响这个最优模式的生存.本文考虑部分解的位置信息兼顾部分解种群的多样性,提出了并行协同进化算法.

4 并行协同进化算法 PSEA(Parallel Symbiotic Evolutionary Algorithm)

在并行协同进化算法中,为完全解的每个位置设立一个对应的部分解种群.构造完全解时从第 1 个种群中选取第 1 个部分解,从第 2 个种群中选取第 2 个部分解,以此类推,这样可以保留位置信息.在求最小约简时,可以将所有属性分成几组.每组属性是一个部分解,在相应的部分解种群中由一个个体表示,算法流程如下.

算法 1. 并行协同进化算法 PSEA(如图 2 所示).

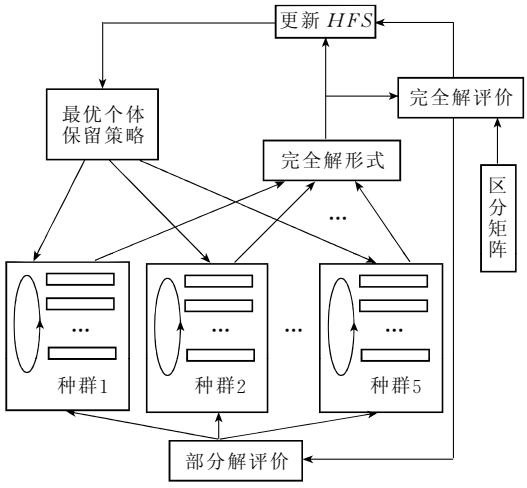

图 2 PSEA 算法原理图

1. 初始化

a. 从文件中读入区分矩阵,每个元素是一个属性的集合.

b. *HFS* (The Highest Fitness Solution) 的初始适应值设为 0,初始化 L 个种群,种群编号为 $1\cdots L$(*HFS* 为目前发现的适应值最高的完全解,一个完全解由 L 个部分解组成).

c. L 个种群中每个个体适应值为 0.

2. 形成完全解

从每个种群中随机选取一个个体,按种群序号形成完全解.

3. 评价完全解

按式(1),结合给定的区分矩阵数据,计算当前完全解的适应值.测试完全解 ch 是否与区分矩阵元素 M_{ij} 匹配时:

a. 将 M_{ij} 表示成长度为 m 的二进制位串 $e_1e_2\cdots e_m$,其中 $e_k=1$ 当且仅当 $a_k\in M_{ij}$,$k=1,2,\cdots,m$,$m=|A|$.

b. 将 ch 与 $e_1e_2\cdots e_m$ 按位求与,如结果为 0,则不匹配,否则匹配.

4. 更新 *HFS*

若该完全解的适应值$>$*HFS* 的适应值,则将当前完全解的各部分解记录到 *HFS* 中,*HFS* 的适应值调整为当前完全解的适应值.

5. 评价部分解

对该完全解的每个部分解进行评价,如果它的适应值小于完全解的适应值,则将它的适应值调整为该完全解的适应值.

6. 重复 2～5 足够多次(如 100 次),使每个部分解有充分的机会参与组成完全解.

7. 执行最优个体保留策略

若在步 2～6 中没有更新 HFS,则将 HFS 记录的各部分解取出(共 L 个),插入每个种群的队头.每个种群中较差的 L 个个体被淘汰.

8. 部分解种群 i 独立进化,$i=1,2,\cdots,L$

a. 种群中的个体按适应值从大到小排序.

b. 按线性排序方法进行选择,计算每个个体复制到下一代的个数,复制后得到中间种群 1.

c. 中间种群 1 中的个体两两配对,按交叉概率 P_c 执行单点交叉运算(即随机选择一个交叉点,两个个体互换交叉点后面的信息),形成中间种群 2.

d. 对中间种群 2 中的每个个体按变异概率 P_m 执行单点变异运算(即随机选择一个变异点,该点的信息求反).

9. 回到步 2,直到迭代超过指定代数(如 1000 代).

部分解种群在进化中采用线性排序选择算子,它能仔细地区分两个适应值相近的个体,赋予它们不同的复制比例[10].个体按适应值从大到小排序后得到一个队列,队列中排序为 i 的个体复制到下一代的个数为

$$P(i)=\eta_{\max}-(\eta_{\max}-\eta_{\min})\frac{i-1}{N-1},\quad i=1,2,\cdots,N \tag{2}$$

满足 $\sum_{i=1}^{N}P(i)=N$,其中 $\eta_{\max}(\eta_{\min})$ 表示种群中最好(最差)个体复制到下一代的个数期望值,满足 $\eta_{\max}=2-\eta_{\min}$,$N$ 表示种群的大小.

$\eta_{\max}$ 和 $\eta_{\min}$ 的选择反映算法对目前适应值较高个体的偏好程度.如果 $\eta_{\max}$ 和 $\eta_{\min}$ 相差较大(如 $\eta_{\max}$ 为 1.8,$\eta_{\min}$ 为 0.2),则适应值低的个体很快被淘汰,种群中的个体很快趋于相同;若二者相差不大(如 $\eta_{\max}$ 为 1.2,$\eta_{\min}$ 为 0.8),则适应值低的个体也有很大的机会进入下一代,可以保持种群的多样性;若二者完全相同,则选择算子不起作用,选择前后的种群完全相同.在实现线性排序选择算子时,可以将 $\eta_{\max}$ 和 $\eta_{\min}$ 设为定值(如本文取 $\eta_{\max}$ 为 1.2,$\eta_{\min}$ 为 0.8),也可以动态调整.

另外,在执行最优个体保留策略时,将历史最好解 HFS 的各个部分解加入到每个种群中以增加种群的多样性.为了使新加入的个体在后续的进化中不被立即淘汰,将它们加入到各种群队列的队头.

5　模拟实验

比较成熟的求属性约简的方法是基于属性重要性的方法,文献[2]中的算法将区分矩阵的元素按照其基数大小分类,认为出现在具有较小基数的类中的属性更重要一些,区分对象的能力更强一些,因此选它作为约简属性.从后边的实验中可以看出,这个算法效率很高,但本身也是不完备的.

本文设计了大量数据用于测试并行协同进化算法 PSEA,并与文献[2]中"改进的属性约简算法"进行了对比.实验省略了从信息系统到区分矩阵的转换,直接从模拟的区分矩阵开始.最小约简的验证用枚举法完成.

5.1　一个例子

某信息系统有 120 个属性,10 个对象,区分矩阵元素的编码如图 3 所示.区分矩阵共有 45 个元素,每个元素编码用 30 个十六进制数表示,元素排列不计顺序.用并行协同进化算法 PSEA、属性重要性方法和枚举法分别对该区分矩阵求解最小约简.

1632f60f861013d22d84b726b6a278	d802d1eeaf1321ba5929dec6a62a65	cff4473621ebdd4ba26a99a812c0e0
3491206ec7624f3d84eea0a8690c22	d0247f183f28c0a9cb01ffd9bde464	b8338b9ad9c3408e06613774328320
0caf83fd23a244315cb318da3009e1	61347577509aed4a63fb3d49e154f9	022c4eb60a97c2799f797c708c803f
e04d470a5f742a9a434b59305abb2b	94dc6cb55a9d2e9c64879827a848ee	98334848bd6edef1c2a1c30e49b5c7
80bc3ec20be2b9ceb7b111b3766469	a308cd15281cebe95ab8bfb755da5c	13ea603e599525dcf2af6d80d2e583
5745e9a0c4ec670ffa023b8fcae4e1	f9b9d12d2256ee3c03cb8daa17b1ae	050529c6827f28c0ef6a1242e93f8b
314fb18a77f790ae049fedd612267f	ecaefc450174d76d9f9aa7755a30cd	90a9a5874bf48eaf70eea3a62a250a
8b6bd8d9b08b08d64e32d1817777fb	544d49cd49720e219dbf8bbed33904	e1fd40a41d370a1f65745095687d47
ba1d36d2349e23f644392c8ea9c49d	40c13271aff264d0f24841d6465f09	96ff84e65fc517c53efc3363c38492
ab08a3aa3ff03f1c55ad514fc48596	585ed5881e81568cbbe99f6d25c8eb	090d191d4a07310158ec97d50d7c15
08aa480f41c8d014a391e8b3502f60	902b85e3b7e31d202f2d6228d35010	175de7e8f7c4e2a8e1c8cf3a65ca58
2c2de20c60dc2c62053c62fac599b0	274068c3abba2d24c1109bc461f1fc	d8bf4ad3e61502c020a2e8a5f2f2ea
07d76187b770db87b1d7e5e94431e1	1d73828d739cc6ced4573dacb0a106	9d373aef06cc4b8cffb64c86571925
36d7259372cb8ecea7bf3c69288743	79c68215f9a11ff76d3e9fb1c6d91d	8a86fccc7324508183b2b471a3bd8c
3b8b755b29ed0d95b2ef65ae44dfe7	774122afaa486eeec3b53a90126b72	1c0fd16edced1a253f72ca9e7b0575

图 3　区分矩阵的元素

首先为 PSEA 设定参数,每代评价的完全解个数为 100,进化的代数设为 1000. 染色体分为 5 段(即 $L=5$),每个部分解长度为 24. 每个种群大小 N 为 32,交叉概率 P_c 为 0.6,变异概率 P_m 为 0.1,$\eta_{\max}$ 为 1.2,$\eta_{\min}$ 为 0.8.

然后运行 PSEA,在每代进化结束时输出目前已发现的最好约简 *HFS*,部分 *HFS* 数据记录在图 4. 图 4 显示:开始 30 代左右 *HFS* 的值变化很快,之后变化减慢. 到 140 代时出现了基数为 3 的约简,该约简保持最高适应值直到 1000 代. 用属性重要性方法和枚举法也得出了相应的结果:

并行协同进化 00000000100000400000008000000,约简的基数为 3.

属性重要性法 80000008020000000000000004000,约简的基数为 4.

枚举法 08082000000000000000000000000,约简的基数为 3.

用枚举法求得的约简不同,说明最小约简不唯一.

代数	到当前代为止,适应值最高的约简 *HFS*	代数	到当前代为止,适应值最高的约简 *HFS*
0	245e0d4440660d890a245e0da61f03	20	00408600408600400600400600 4086
2	12493244404a124c87124c8744404a	22	00400600400600400640400600 4006
4	304066124c8744404a44406644404a	24	00400600400600400600400600 4006
6	30404a30406644408730406404066	26	00400600400200400200400600 4006
8	44404740408644400740406404066	28	00400600400000400000400000 4006
10	44400740406640408640408740 4086	30	00400200400000400200400000 4000
12	40408640408640408640406640 4086	34	00400000400200400000400000 4000
14	40408640408640408640408600448a	72	00000000400000400001400000 4000
16	40408600408600408600408640 4006	82	00040001400000000000400000000
18	40400600400600408600408600 4086	140	00000000100000400000008000000

图 4 进化过程记录

5.2 7 种不同的区分矩阵

从属性约简的定义可知,区分矩阵元素中 1 占的比例越大,求得的约简基数越小. 这说明各对象在很多属性上取值不同,因此几个属性就能将所有不同的对象区分开. 但是当区分矩阵元素中 1 占的比例较小时,需要多个属性配合才能完成对象的区分. 因此本文根据区分矩阵元素中 1 占的比例设计了 7 个随机数据类(每类 100 个区分矩阵),区分矩阵的大小及运行参数设置与 5.1 相同. 用简单遗传算法 SGA(Simple Genetic Algorithm)、协同进化算法 SEA、并行协同进化算法 PSEA、基于属性重要性算法分别进行测试,求解结果见表 1.

表 1 4 种算法对各类区分矩阵数据的平均约简量对比

算法	平均约简量(%)						
	Ⅰ类(4.2)	Ⅱ类(6.3)	Ⅲ类(12.5)	Ⅳ类(12.5)	Ⅴ类(25)	Ⅵ类(37.5)	Ⅶ类(50)
SGA	82.63	85.20	89.88	91.38	93.65	95.38	96.32
SEA	76.94	83.50	90.74	93.42	94.74	96.48	97.47
PSEA	84.95	87.93	92.38	94.33	95.49	96.68	97.48
属性重要性	86.77	89.49	93.42	94.95	95.91	96.81	97.32

注:每个数据类标有该类区分矩阵数据中 1 占的百分比.

表 1 中,约简量定义为(属性总数-约简的基数)/属性总数,表示去掉的属性占的比例,显然约简量越大,相应的约简基数越小. 从表 1 中可以看出,

(1) 随着 1 所占比例的增加,就每种算法来讲,属性的约简量都在增加.

(2) 当 1 占的比例≥12.5%(即 120 个属性中有 15 个属性可区分)时,PSEA 与属性重要性方法的约简量差异在 1%以内,可以认为完全类似.

(3) PSEA 对 SEA 的改进是明显的,尤其在前 3 类中.

(4)虽然在Ⅲ、Ⅳ类数据中 1 占的比例相同,但在Ⅲ类中 1 均匀分布在 120 位中,而在Ⅳ类中 1 在 1～24 位中均匀出现 5 次,在 25～48 中出现 2 次,49～72 位中出现 1 次,73～96 中出现 4 次,在 97～120 中出现 3 次. Ⅲ、Ⅳ类数据中 1 出现的模式不同,PSEA 求到的平均约简量也不同,说明 PSEA 适合于求解各个部分解有不同最优模式的问题.

针对第Ⅶ类数据,本文进行了几种算法的性能对比,得到表 2. 从表 2 中可以看出,并行协同进化算法为 96 个区分矩阵找到了最小约简,成功率最

高.就评价的完全解的个数而言,并行协同进化算法的性能明显优于其它算法.属性重要性方法不评价完全解,它为 78 个区分矩阵找到了最小约简.

表 2　几种算法性能对比(第Ⅶ类数据)

算法名称	评价完全解的平均个数	找到最小约简的比例(%)	平均约简量(%)
枚举法	147680	100	97.51
属性重要性	—	78	97.32
SGA	18924	2	96.32
SEA	17852	95	97.47
PSEA	7424	96	97.48

6　结　论

本文将并行遗传算法和协同进化算法相结合,提出了一种求属性约简的新方法 PSEA,为属性约简问题提供了新思路.实验表明 PSEA 能有效地找出集合的最小约简,为后续的知识发现算法减少了时间与空间复杂性.

参　考　文　献

1　Pawlak Z. Drawing conclusions from data——The rough set way. International Journal of Intelligent Systems, 2001,16:3～11

2　Miao Duo-Qian. Rough set theory and its application in machine learning[Ph D dissertation]. Institute of Automation. Chinese Academy of Sciences, Beijing, 1997(in Chinese)
(苗夺谦. Rough Set 理论及其在机器学习中的应用研究[博士学位论文]. 中国科学院自动化研究所,北京,1997)

3　Hu X H, Cercone N. Learning in relational databases: A rough set approach. International Journal of Computational Intelligence, 1995, 11(2):323～328

4　Miao Duo-Qian, Hu Gui-Rong. A heuristic algorithm for reduction of knowledge. Journal of Computer Research and Development, 1999, 36(6):681～684 (in Chinese)
(苗夺谦,胡桂荣.知识约简的一种启发式算法.计算机研究与发展, 1999,36 (6):681～684)

5　Chang Li-Yun, Wang Guo-Yin, Wu Yu. An approach for attribute reduction and rule generation based on rough set theory. Journal of Software, 1999,10(11):1206～1211(in Chinese)
(常犁云,王国胤,吴　渝. 一种基于 Rough set 理论的属性约简及规则提取方法. 软件学报,1999, 10(11):1206～1211)

6　Xu Gui-Rong, Wang Yong-Biao, Gong Shu-Yun. Mutually developing: A global view of life evolution. Geological Science and Technology Information, 1998,17(2):102～105(in Chinese)
(徐桂荣,王永标,龚淑云. 协同进化——生物发展的全球观. 地质科技情报,1998,17(2):102～105)

7　Moriarty D E, Miikkulainen R. Efficient reinforcement learning through symbiotic evolution. Department of Computer Sciences, University of Texas at Austin, Austin: Technical Report AI94-224,1994

8　Moriarty D E, Miikkulainen R. Hierarchical evolution of neural networks. Department of Computer Sciences, University of Texas at Austin, Austin: Technical Report AI96-242, 1996

9　Wu Geng-Feng, Wang Li-Hong, Zhu Ying-Zhuang. Symbiotic evolution based fuzzy neural controller. In: Proceedings of the 2002 International Conference on Control and Automation, Xiamen, 2002. 50～54

10　Herrera F, Lozano M. Gradual distributed real-coded genetic algorithms. IEEE Transactions on Evolutionary Computation, 2000,4(2):43～62

WANG Li-Hong, born in 1970, Ph. D. candidate, associate professor. Her main research interests include knowledge discovery, intelligent information processing and the architecture of computer network.

WU Geng-Feng, born in 1945, professor, Ph. D. supervisor. His current research interests include data mining and knowledge discovery in database, intelligent control, intelligent information processing and CSCW (computer supported cooperative work).