

# Survey Propagation: 一种求解 SAT 的高效算法

李韶华 张 健

(中科院软件所 计算机科学实验室 北京100080)

**摘 要** Survey propagation 是一种新生的 SAT(CSP)算法。它基于统计物理的 spin glass 模型,针对具体问题进行纵览(survey),从而极大地降低求解的复杂度。但 sp 算法在某些时候不收敛,或引导向错误的解。对此,G. Parisi 提出一种复杂回溯(backtrack)算法,而作者在 sp 中加入简单回溯,也使一部分此类问题得到解决。

**关键词** 不完备搜索方法,可满足性问题,求解算法

## Survey Propagation: An Effective Algorithm for Solving SAT

LI Shao-Hua ZHANG Jian

(Computer Science for Lab., Institute of Software, Chinese Academy of Sciences, Beijing 100080)

**Abstract** The Survey propagation is a new SAT(CSP)algorithm. It is a spin glass model based on statistic physics, which makes survey for specific problems, so as to greatly reduce the complexity of solving. But the sp algorithm sometime is convergence, or leads to erroneous solutions. For this reason, in this paper, a simple backtracking is inducted into SP, thereby partially solves this kind of SAT problems.

**Keywords** Incomplete search, SAT, Solving algorithm

## 1 简介

命题逻辑公式的可满足性问题(SAT)是计算机科学和人工智能中的一个重要问题。过去几十年,很多学者在这方面做了大量的工作,提出了各种各样的算法<sup>[1]</sup>。其中最经典的是 Davis 等人提出的 DPLL 算法。它可看成是一种回溯搜索方法,也就是一个一个地确定命题变量的值,直到所有变量都被赋值并且公式成真,或者所有可能的赋值都被试过但都不合适。从理论上讲,这种算法总能判定一个给定的命题公式是否可满足。但是,其最坏复杂度是指数级的,因此往往难以解决规模较大的问题实例。

从20世纪90年代起,很多人开始研究不完备的搜索过程,比如以 GSAT 为代表的局部搜索(local search)方法。这种方法不一定能判定任何公式的可满足性,但对某些可满足的公式能很快地给出答案。最近几年,欧洲有些学者(M. Mezard, G. Parisi 等)从统计物理的模型出发,又提出了一种称为 survey propagation(SP)的不完备搜索方法<sup>[2~4]</sup>。实验表明,在难解区域内,这种算法可以固定随机 SAT 实例的10%~50%的变量,然后交给 walksat 等算法求解<sup>[3,5]</sup>。他们声称在变量数为 $10^7$ 的问题实例上,SP 仍有不错的表现,交给 walksat 的子问题在短时间内能解出。但这一算法在 SAT/UNSAT 临界区(第2节),常常不收敛,而不收敛时得到的子问题常常 UNSAT,或 walksat 等算法在短期内无法解出。本文主要基于这些人提出的 SP 算法<sup>[3]</sup>,介绍其基本思想,对它进行分析,并对不收敛行为提出了使用回溯的改进办法。对100个随机实例的实验表明,使用作者的回溯策略,大约1/6的不收敛问题可以很快(5秒内)求出解。

## 2 SAT 问题及其相变现象

### 2.1 预定义

给出一个 SAT 问题,其变量数为  $N$ ,子句数为  $M$ 。所有

变量记为  $\{x_i \in \{0,1\}\}, i \in \{1, \dots, N\}$ 。在一个子句  $a$  里,它包含  $K$  个变量  $i_1, \dots, i_k$ (记为  $V(a)$ ),这其中变量的正负用  $K$  个数  $J_r^a \in \{\pm 1\}$  区别。特别地,对于 3-SAT,  $K=3$ 。子句写为

$$(z_{i_1} \vee \dots \vee z_{i_r} \vee \dots \vee z_{i_k}) \quad (1)$$

其中如果  $J_r^a = -1, z_{i_r} = x_{i_r}$ ; 如果  $J_r^a = 1, z_{i_r} = \bar{x}_{i_r}$ 。

记它的一组变量取值为  $\vec{x} = (x_i)_{i \in I}, I = \{1, \dots, N\}$ 。每个变量有 0, 1 两种状态,故  $\vec{x} \in X = \{0, 1\}^N$ 。向量  $\vec{x}$  被称为一个配置。子句  $a$  可满足,记作  $C_a = 0$ ,不可满足,记作  $C_a = 1$ 。

定义代价方程:  $C[\vec{x}] = \sum_{a \in A} C_a[(x_i)_{i \in V(a)}]$ 。其中  $A$  为所有子句集合。它统计了所有未满足子句数。

则解集  $S_C \subset X$  可如下定义:

$$S_C = \{\vec{s} | \vec{s} \in X, C[\vec{s}] = 0\}$$

### 2.2 相变

当变量数  $N$  足够大,而子句与变量比例  $\alpha = M/N$  保持不变时,随机  $k$ -SAT 问题表现出有趣的相变现象。此比例在某个值  $\alpha_c(K)$  处,问题出现相变。对  $\alpha < \alpha_c(K)$ ,问题一般是可满足(SAT)的,而对于  $\alpha > \alpha_c(K)$ ,问题总的来说不可满足(UNSAT)。严格来说,随机选择一个变量数  $N$ ,子句数  $\alpha N$  的问题实例  $ins$ ,当  $N \rightarrow \infty$ ,如果  $\alpha_c(K), Pr_{SAT}(ins) \rightarrow 1$ ; 如果  $\alpha > \alpha_c(K), Pr_{SAT}(ins) \rightarrow 0$ 。特别地,  $k=3$  时,  $\alpha_c(3) \approx 4.267$ 。

不仅如此,进一步的研究发现,可以假定存在另一个相变点  $\alpha_{cluster}$ (又记做  $\alpha_d$ ),小于  $\alpha_c$ ,在此值之上,解组成的空间会劈分到一些互不连接的簇(在文[4]中称为状态)里。一个簇是指具有相同不满足子句数的一些解的配置,它们彼此之间可以通过一次变量翻转相互达到。通过统计物理“one step replica symmetry breaking”(1-RSB)方法,可以计算得  $\alpha_d(3) = 3.921$ 。簇的数目  $N_c$  关于  $N$  有如下关系:

$$N_c(E) = \exp(N\Sigma(\alpha, \epsilon))$$

其中  $\Sigma(\alpha, \epsilon)$  被称为复杂度。 $\epsilon = E/N$ ,  $E$  是不满足子句个数。注意  $N\Sigma$  并非单增,故簇数目也并非单增(图1)。

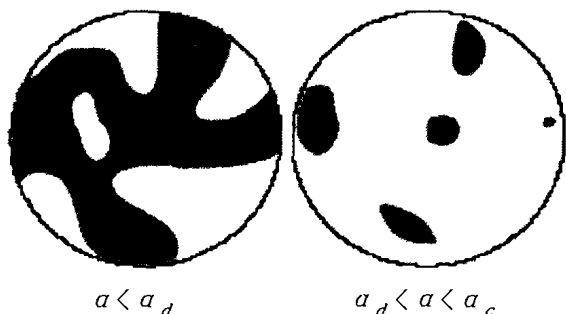


图1 在 STA 区,解集连成若干大的区域。当  $a$  越过分簇阈值  $a_d$  时,解集劈分到许多互不连接的簇里。 $a$  增加,簇继续缩小,最后在 SAT 阈值  $a_c$  处消失。全空间是个  $N$ -维的超立方体,图中用圈表示。

### 2.3 SAT 问题的因子图(Factor Graph)表示

SAT 问题可表示为因子图的形式<sup>[6]</sup>。而因子图是下面算法的重要数据结构。从历史上讲,因子图的结点和 spin glass 模型的自旋(spin)相对应。

因子图是个 2 分无向图  $G=(V, E)$ ,它有两种类型的结点:变量结点  $i \in I$  和子句结点  $a \in A$ 。图中有边  $(i, a)$ ,当且仅当  $i$  出现在  $a$  中(图2)。

在图中,我们总是用圈代表变量结点,用方块代表子句结点。

变量结点记为  $i, j, k, \dots$ ,子句结点记为  $a, b, c, \dots$ 。对每个变量结点  $i$ ,我们记它连接的子句结点集合为  $V(i)$ ,它的连通度为  $n(i) = |V(i)|$ 。 $V_+(i)$  表示  $V(i)$  的由变量在其中为正的子句结点  $a$  组成的子集,(边  $a-i$  为实线), $V_-(i)$  代表  $(V_+(i))$  对于  $V(i)$  的补集,其中变量为负(边  $a-i$  为虚线)。 $V(i)$ ,  $b$  指除去结点  $b$  的集合  $V(i)$ 。类似地,对每个子句结点  $a$ ,我们用  $V(a) = V_+(a) \cup V_-(a)$  表示邻接变量结点的集合,由连接  $a$  和  $i$  的边的类型分解开, $n(a)$  为其连通度。

因子图的一个重要特性是:它局部来看是类似树(tree like)的。这一点是下面算法能正常工作的重要依据。

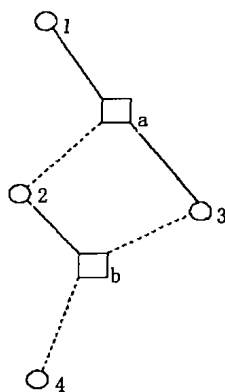


图2 一个具有4个变量  $i=1, \dots, 4$  和2个子句结点  $a, b$  的因子图的例子。原来的公式是:  $(x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_4)$

## 3 统计物理和 SAT

统计物理方法曾用来解决优化问题,即众所周知的模拟退火算法(simulated annealing, SA)。它是模仿了 spin glass 冷却时,spin 的翻转总是以较大概率趋向于减小“受挫”(frustration)数目(或者系统能量,可近似等价于 CSP 中约束未满足的数目),以较小概率倾向于增加系统能量。这样经过较长的时间,系统能量会达到一个很低的水平<sup>[7,8]</sup>。

但正如现实的 spin glass 系统不总处于最低能态一样,SA 也不能保证总是能达到最低能态。

而 BP 和 SP 用到的统计物理思想则远远超出了上面的对物理过程的简单模拟。与 SA 本质上是一种局部搜索策略(而且这种策略是问题无关的)相对,SP 使用源于统计物理(或与统计物理方法等价)的方法计算全局信息,用它来引导搜索过程,自然会高效得多<sup>[3]</sup>。大概来说,统计物理方法似乎并不适合计算每个 SAT 配置的性质,但它却发展出了有效的方法,计算所有解的统计特性。如每个变量 在所有解里的平均值,在统计物理里称作自旋  $i$  处的磁场  $\langle i \rangle$ 。在高温  $T > T_c$  (对应于允许有较多不满足子句,或约束不足的 SAT 的解空间),系统处于顺磁态,“自旋”没有极化( $\langle i \rangle = 0$ ),也即平均而言, $i$  取 1 和取 -1 的解的数目大致相当(这里取值 1 和 -1 分别对应于 SAT 解中的 1 和 0),低于  $T_c$  时,系统进入自旋玻璃态。在这个状态中,解空间劈分到若干簇里,而在每个簇  $l$  里,“自旋”倾向于固定在某个方向上:

$$\langle i \rangle_l = m_l^i \neq 0$$

也就是在簇里, $i$  倾向于取 1 或 -1。但对所有变量取平均后,这个磁场可能仍趋近于 0<sup>[9]</sup>:

$$\frac{1}{N} \sum_i \langle i \rangle_i \approx 0 (\pm O(1/\sqrt{N}))$$

容易想到,这一信息可用来引导搜索算法朝解较多的子空间搜索。这也是 BP 和 SP 的基本工作原理。

## 4 BP 算法简介

作为 SP 算法的前身和一种简化(不准确地说),这里先介绍 BP(Belief propagation)算法。本文中的 BP 算法实际上是标准 BP 算法在 SAT 问题上的改编,在文[3]中称为警告传播(warning propagation, WP)。而标准 BP 算法则是“Sum-Product”算法的一种受限情况。“Sum-Product”算法被广泛用于计算图中相互作用的离散变量的边缘概率<sup>[6]</sup>。

### 4.1 空腔偏差(cavity bias)和空腔场(cavity field)

给出一个 SAT 问题和配置(configuration,即各变量的一组取值)  $\vec{x} \in X$ ,定义下面的三个与  $\vec{x}$  相关的量:

从子句结点  $a$  传到变量结点  $i$  的基本消息是个布尔量  $u_{a \rightarrow i} \in \{0, 1\}$ ,称为空腔偏差(文[2]称为警告,本文一些地方也用这个词代表空腔偏差)。从变量结点  $j$  传到子句结点  $a$  的基本消息是一个整数  $h_{j \rightarrow a}$ ,称为空腔场。

为计算空腔场  $h_{j \rightarrow a}$ ,变量  $j$  考虑从它连接的所有子句结点  $b$ ,除了  $b=a$ ,接收到的消息(既然称为“空腔”),求和:

$$h_{j \rightarrow a} = \left( \sum_{b \in V_+(j), b \neq a} u_{b \rightarrow j} \right) - \left( \sum_{b \in V_-(j), b \neq a} u_{b \rightarrow j} \right) \quad (2)$$

如果  $j$  除  $a$  外没有邻结点,则  $h_{j \rightarrow a} = 0$ 。

为计算空腔偏差  $u_{a \rightarrow i}$ ,子句结点  $a$  考虑它从所有变量结点  $j$ ,除了  $j=i$ ,接收到的消息。如果至少有一个  $j \in V(a)$ ,  $i$  使  $h_{j \rightarrow a} J_{ji}^* \leq 0$ ,则  $u_{a \rightarrow i} = 0$ 。否则  $u_{a \rightarrow i} = 1$ 。如果  $a$  没有  $i$  以外其他邻结点,则  $u_{a \rightarrow i} = 1$ 。故而:

$$u_{a \rightarrow i} = \prod_{j \in V(a), j \neq i} \theta(h_{j \rightarrow a} J_{ji}^*) \quad (3)$$

这里  $\theta(x)$  是梯度函数。 $\theta(x) = 0$  对  $x \leq 0$ ,  $\theta(x) = 1$  对  $x > 0$ 。

给出一个子句结点  $a$  和它的一个变量  $i$ ,空腔场  $h_{j \rightarrow a}$  ( $j \neq i$ ) 可被视为输入,  $u_{a \rightarrow i}$  则是输出。消息和消息传递过程可如下解释。空腔偏差  $u_{a \rightarrow i} = 1$  可被解释为从子句结点  $a$  送出的警告,告诉变量  $i$  它应该取使子句  $a$  满足的值。这由  $a$  从与之相连的其他结点接收到的消息决定:如果  $h_{j \rightarrow a} J_{ji}^* > 0$ ,这意味着结点  $j$  (在  $a$  被移除的情况下)的趋势是取使  $a$  不满足的值。如果所有邻结点  $j \in V(a)$ ,  $i$  都处于这样的处境,则  $a$  给  $i$  发送

一个警告(如果  $i$  仍不满足  $a$ , 则  $a$  就不可满足, 故而  $i$  必须满足  $a$ )。

## 4.2 迭代动态过程(Iteration Dynamics)

BP 是一种迭代算法(和 SP 一样), 迭代动态过程使用空腔偏量作为基本消息。它们被初始化为随机值(比如  $u_{a \rightarrow i} = 0$ , 1, 取概率 1/2)。每次迭代我们随机游历边的集合, 并更新所有的相应空腔偏量。

如果这个过程收敛, 这个动态过程定义了一组固定的空腔偏量  $u_{a \rightarrow i}$ 。它们可用于计算对每个变量  $i$  的局部场(local field)  $H_i$  和“矛盾数”  $c_i$ , 二者是两个如下定义的整数:

$$H_i = \sum_{\delta \in V_+(i)} u_{\delta \rightarrow i} - \sum_{\delta \in V_-(i)} u_{\delta \rightarrow i} \quad (4)$$

$$c_i = 1 \text{ if } \left( \sum_{\delta \in V_+(i)} u_{\delta \rightarrow i} \right) \left( \sum_{\delta \in V_-(i)} u_{\delta \rightarrow i} \right) > 0 \quad (5)$$

$$= 0 \text{ otherwise} \quad (6)$$

局部场  $H_i$  是变量  $i$  的偏向的指示:  $x_i = 1$  如果  $H_i > 0$ ,  $x_i = 0$  如果  $H_i < 0$ 。矛盾数指出了是否变量  $i$  收到了相抵触的消息。

## 4.3 BP 算法流程

BP 算法可总结为:

BP algorithm:

INPUT: 一个 SAT 问题的因子图表示以及最大迭代数  $t_{\max}$

OUTPUT: UN-CONVERGED, 如果 BP 经过  $t_{\max}$  步后仍不收敛。如果它收敛, 输出所有警告的集合  $u_{a \rightarrow i}$ 。

0. 在时间  $t=0$ : 对因子图每边  $a \rightarrow i$ , 随机初始化警告  $u_{a \rightarrow i}(t=0) \in \{0, 1\}$

1. For  $t=1$  to  $t=t_{\max}$ :

1.1 以随机顺序游历边的集合, 并依次更新图的所有边上的警告值。用子程序 UPDATE 生成值  $u_{a \rightarrow i}(t)$ 。

1.2 如果所有边上,  $u_{a \rightarrow i}(t) = u_{a \rightarrow i}(t-1)$ , 则迭代收敛了, 并生成  $u_{a \rightarrow i} = u_{a \rightarrow i}(t)$ ; go to 2.

2. If  $t=t_{\max}$  return UN-CONVERGED. If  $t < t_{\max}$  return 固定点警告集合  $u_{a \rightarrow i} = u_{a \rightarrow i}(t)$

Subroutine UPDATE( $u_{a \rightarrow i}$ ).

INPUT: 所有变量结点  $j \in I(a)$ ,  $i$  上到达的警告值

OUTPUT: 警告  $u_{a \rightarrow i}$  的新值。

1. 对每一  $j \in I(a)$ ,  $i$ , 计算空腔场

$$h_{j \rightarrow a} = \left( \sum_{\delta \in I_+(j), \delta \neq a} u_{\delta \rightarrow j} \right) - \left( \sum_{\delta \in I_-(j), \delta \neq a} u_{\delta \rightarrow j} \right)$$

2. 用空腔场  $h_{j \rightarrow a}$ , 计算警告  $u_{a \rightarrow i} = \prod_{j \in I(a), j \neq i} \theta(h_{j \rightarrow a}, J_j^i)$

BP 算法的重要性很大程度上来自于这样的事实: 它可以给出因子图是树的问题的确切解。对 SAT 情况, 这被总结为下面的定理:

**定理** 考虑一个因子图为树的 SAT 问题实例, 则 WP 算法收敛到一个唯一的固定空腔偏量集合  $u_{a \rightarrow i}$  上, 与初始空腔偏量无关。如果至少有一个相应的矛盾数  $c_i$  等于 1, 则问题是 UNSAT, 否则是 SAT。

**推论** 在问题是 SAT 的情况下, 局部场  $H_i$  可被用于找出满足所有子句的一组变量的赋值。当  $H_i \neq 0$ , 变量  $i$  是受约束的, 我们应该固定  $x_i = 1$  当  $H_i > 0$ ,  $x_i = 0$  当  $H_i < 0$ 。一个  $H_i = 0$  的变量是约束过少的, 这意味着存在一些解, 其中它为 1, 另一些解中它为 0。为找到一组满足赋值, 我们可以:

i) 固定所有受约束变量, 并消除所有被满足子句, 简化剩余公式。

ii) 随机选择一个约束过少的变量, 固定它的值为  $\{0, 1\}$  中一个值。消除所有被满足子句, 并简化剩余公式。

iii) 再次在新的实例上运行 BP 算法, 并反复整个过程。

上面的算法在文[3]中被称为警告启发算法(Warning Inspired Decimation, WID)。可总结为:

WID algorithm:

INPUT: 一个 SAT 问题的因子图表示

OUTPUT: UN-CONVERGED, 或公式的状态(SAT 或 UNSAT); 如果公式为 SAT, 输出一组赋值, 它满足所有子句。

1. While 未固定变量数  $>$ , do:

1.1 Run WP

1.2 如果 WP 不收敛, return UN-CONVERGED. 否则计算局部场  $H_i$  和矛盾数  $c_i$ , 使用公式 (4, 6)。

2. 如果至少一个矛盾数  $c_i = 1$ , return UNSAT. 否则:

2.1 如果至少一个局部场  $H_i \neq 0$ : 固定所有  $H_i \neq 0$  的变量 ( $H_i > 0 \Rightarrow x_i = 1$  和  $H_i < 0 \Rightarrow x_i = 0$ ), 使用“单子句规则”清理图。GOTO label 1. 否则:

2.2 选择一个未固定变量, 固定它到任意值, 清理图。GOTO label 1

\*return 所有变量赋值

定理证明见文[3]。

## 5 SP 算法简介

在接近 SAT/UNSAT 阈值区域, 遗憾的是, BP 算法常常无法收敛。这可能是由于在此区域, 解被分散在许多相距较远的簇里的缘故。分簇现象使 BP 算法陷入无能为力。这是因为局部交换的信念只能传递一个解簇的信息, 图的相距较远的部分的信念通常涉及了不同的簇, 从而算法无法得到全局信息。简单的迭代规则找到一组与某特定簇相对应的空腔偏量的概率非常小。原因可如下量化的理解: 当在图的一部分区域, 局部迭代规则会产生在一个簇方向上“极化”的(polarized)空腔偏量, 图的其他区域会被极化到别的方向, 对应于不同的簇。故而多次迭代后, 方程无法收敛。

在这些簇里, 我们可以区分两种变量: 一种是簇里所有配置中, 都被固定在某个单一状态的变量, 另一种是在簇里不同的解之间, 值波动的变量。注意在某个簇里固定的变量, 可能到了别的簇里, 值仍然会波动, 即变成非固定变量。

针对这种情况, Mezard 等人<sup>[2]</sup>在 BP 算法里增添了一个新的状态, 称为百搭态(joker state), 一个变量处于百搭态, 意味着在一个解簇里, 它可以取 1 或 0 (也就是未约束变量)。

与 BP 算法不同的是, SP 算法不再试图算出每一簇解对应的一组信念, 而是采用了一种更加复杂的消息传递机制。每条边上传递的消息包含了对这条边上属于不同解簇的所有信念的一个纵览(survey)。这样不同“极化”方向的消息不再相互矛盾, 而是都在纵览中得到了一定体现。这样算法就能处理这种分簇的情况(作者看来, 这是对标准 BP 算法作了某种弱化)。

### 5.1 纵览(survey)的定义

我们如下定义空腔偏量  $u_{a \rightarrow i}$  的纵览(survey):

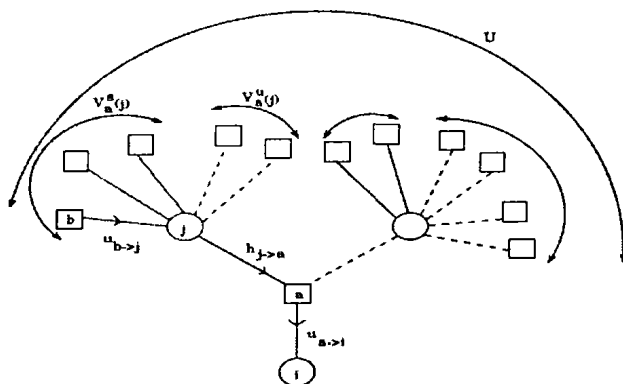


图3 一个子句结点  $a$  和它的邻结点。空腔偏量  $u_{a \rightarrow i}$  的纵览可由集合  $U$  里所有空腔偏量, 即, 那些发到  $a$  的所有的邻变量结点  $j$  (除了  $j=i$ ) 的警告的联合概率分布的知识计算出。

$$Q_{a \rightarrow i}(u) = \frac{1}{N_{cl}} \sum_l \delta(u, u'_{l \rightarrow i}) \quad (7)$$

其中  $l$  跑遍所有的解簇,  $u'_{l \rightarrow i}$  是在簇  $l$  中上边  $a \rightarrow i$  的空腔偏量。 $\delta$  指离散变量的 Kronecker delta 方程,  $\delta(x, y) = 1$  如果  $x = y$ , 否则为 0。

注意  $u \in \{0, 1\}$ , 因此空腔偏量纵览  $Q_{a \rightarrow i}(u)$ , 作为一个概率分布, 可用一个参数  $\eta_{a \rightarrow i}$  表示:

$$Q_{a \rightarrow i}(u) = (1 - \eta_{a \rightarrow i}) \delta(u, 0) + \eta_{a \rightarrow i} \delta(u, 1) \quad (8)$$

## 5.2 纵览迭代方程

由式(3), 概率  $\eta_{a \rightarrow i}$ , 即  $u_{a \rightarrow i} = 1$  等于概率:  $\forall j \in V(a), i: h_{j \rightarrow a} J_j^a > 0$ 。

记变量  $J_j^a$  严格正的的概率为  $\rho_{j \rightarrow a}$ 。定义集合  $V_+^a(j)$  和  $V_-^a(j)$  ( $j$ ), 指标  $S$  和  $u$  分别指倾向于使变量  $j$  满足或不满足子句  $a$  的邻结点, 定义为(见图 3):

$$\text{if } J_j^a = 1: V_+^a(j) = V_+(j); V_-^a(j) = V_-(j), a \quad (9)$$

$$\text{if } J_j^a = -1: V_+^a(j) = V_-(j); V_-^a(j) = V_+(j), a \quad (10)$$

为计算发到子句结点  $a$  的所有消息的联合分布, 我们假定注入子句结点的消息都是不相关的。这样它们的联合分布就可以分解。在此假定下, 有:

$$\eta_{a \rightarrow i} = \prod_{j \in V(a), i} \rho_{j \rightarrow a} \quad (11)$$

而

$$\rho_{j \rightarrow a} = C_{j \rightarrow a} \sum_{\{u_{b \rightarrow j}\}} \left[ \prod_{b \in V(j), a} Q_{b \rightarrow j}(u_{b \rightarrow j}) \right] \theta \left( \sum_{b \in V_+^a(j)} u_{b \rightarrow j} \right) \prod_{b \in V_-^a(j)} \delta(u_{b \rightarrow j}, 0) \quad (12)$$

我们只考虑非矛盾消息, 即  $V_+^a(j)$  与  $V_-^a(j)$  不同时发出警告的情景。这样概率分布的计算需要归一化。所有非矛盾消息可分为 3 类:

Class u: 那些至少有一个  $u_{b \rightarrow j}, b \in V_+^a(j)$  等于 1 的(这些消息是贡献到  $\rho$  上的那些)。Class s: 那些至少有一个  $u_{b \rightarrow j}, b \in V_-^a(j)$  等于 1 的。Class 0: 那些所有  $u_{b \rightarrow j}, b \in V_+^a(j)$  等于 0 的。我们计算每一类的权重,  $\prod_{j \rightarrow a}^{u, s, 0}$ :

$$\prod_{j \rightarrow a}^u = \sum_{\{u_{b \rightarrow j}\}} \left[ \prod_{b \in V(j), a} Q_{b \rightarrow j}(u_{b \rightarrow j}) \right] \theta \left( \sum_{b \in V_+^a(j)} u_{b \rightarrow j} \right) \prod_{b \in V_-^a(j)} \delta(u_{b \rightarrow j}, 0) \quad (13)$$

$$\prod_{j \rightarrow a}^s = \sum_{\{u_{b \rightarrow j}\}} \left[ \prod_{b \in V(j), a} Q_{b \rightarrow j}(u_{b \rightarrow j}) \right] \theta \left( \sum_{b \in V_+^a(j)} u_{b \rightarrow j} \right) \prod_{b \in V_-^a(j)} \delta(u_{b \rightarrow j}, 0) \quad (14)$$

$$\prod_{j \rightarrow a}^0 = \sum_{\{u_{b \rightarrow j}\}} \left[ \prod_{b \in V(j), a} Q_{b \rightarrow j}(u_{b \rightarrow j}) \right] \left[ \prod_{b \in V(j), a} \delta(u_{b \rightarrow j}, 0) \right] \quad (15)$$

则式(12)中的归一化常量为:

$$C_{j \rightarrow a} = 1 / (\prod_{j \rightarrow a}^u + \prod_{j \rightarrow a}^s + \prod_{j \rightarrow a}^0) \quad (16)$$

式(12)和归一化式(14)提供了关于  $\eta_{a \rightarrow i}$  一组闭合的等式, 故可以用来迭代。

## 5.3 SP 算法流程

SP 迭代等式可直接写为:

$$\eta_{a \rightarrow i} = \prod_{j \in V(a), i} \left[ \frac{\prod_{j \rightarrow a}^u}{\prod_{j \rightarrow a}^u + \prod_{j \rightarrow a}^s + \prod_{j \rightarrow a}^0} \right] \quad (17)$$

这里:

$$\begin{aligned} \prod_{j \rightarrow a}^u &= \left[ 1 - \prod_{b \in V_+^a(j)} (1 - \eta_{b \rightarrow j}) \right] \prod_{b \in V_-^a(j)} (1 - \eta_{b \rightarrow j}) \\ \prod_{j \rightarrow a}^s &= \left[ 1 - \prod_{b \in V_+^a(j)} (1 - \eta_{b \rightarrow j}) \right] \prod_{b \in V_-^a(j)} (1 - \eta_{b \rightarrow j}) \prod_{j \rightarrow a}^0 = \\ &\prod_{b \in V(j), a} (1 - \eta_{b \rightarrow j}) \end{aligned} \quad (18)$$

连乘  $\prod_{b \in V_+^a(j)} (1 - \eta_{b \rightarrow j})$  给出了从子句结点  $b \in V_+^a(j)$  没

有警告抵达  $j$  的概率; 如果集合  $V_+^a(j)$  为空, 积定义为 1。如果  $V(a), i$  为空, 则  $\eta_{a \rightarrow i} = 1$ 。这里的  $\prod_{j \rightarrow a}^{u, s, 0}$  和等式(13)中的是等价的。

SP 算法可总结如下:

SP algorithm:

INPUT: 一个 SAT 问题的因子图表示以及最大迭代数  $t_{\max}$ , 要求精度  $\epsilon$ 。

OUTPUT: UN-CONVERGED, 如果经过  $t_{\max}$  步后仍不收敛。如果它收敛, 输出所有警告纵览的集合  $\eta_{a \rightarrow i}^*$ 。

0. 在时间  $t=0$ : 对因子图每一边  $a \rightarrow i$ , 随机初始化空腔偏量  $\eta_{a \rightarrow i}(t=0) \in [0, 1]$
1. For  $t=1$  to  $t=t_{\max}$ :
  - 1.1 游历边的集合, 并依次更新图的所有边上的警告纵览值, 用子程序 CBS-UPDATE 生成值  $\eta_{a \rightarrow i}(t)$ 。
  - 1.2 上,  $|\eta_{a \rightarrow i}(t) - \eta_{a \rightarrow i}(t-1)| < \epsilon$ , 则迭代收敛了, 并生成  $\eta_{a \rightarrow i}^* = \eta_{a \rightarrow i}(t)$ ; GOTO label 2.
2. If  $t=t_{\max}$  return UN-CONVERGED. If  $t < t_{\max}$  return 固定点警告纵览集合  $\eta_{a \rightarrow i}^* = \eta_{a \rightarrow i}(t)$

Subroutine CBS-UPDATE( $\eta_{a \rightarrow i}$ ).

INPUT: 所有变量结点  $j \in V(a), i$  上到达的警告纵览值

OUTPUT: 警告纵览  $\eta_{a \rightarrow i}$  的新值。

1. 对每一  $j \in V(a), i$ , 使用式(16)计算值  $\prod_{j \rightarrow a}^u, \prod_{j \rightarrow a}^s, \prod_{j \rightarrow a}^0$ 。
2. 使用式(15)计算  $\eta_{a \rightarrow i}$ 。

## 5.4 SP 启发的消解过程(Survey Inspired Decimation, SID)

SP 迭代得到的一组警告纵览可用于估计每个变量的在所有解中的统计特性(称为总偏量), 并用于新的启发式算法 SID 中。

5.4.1 偏量 假定 SP 收敛到了一组警告纵览  $\eta_{a \rightarrow i}^*$ 。它可用于估计每个变量的属性(总偏量)。当随机选取一个簇时, 变量  $i$  被约束(“固定”)到  $x_i = 1$  的概率称为它的正偏量:

$$W_i^{(+)} = \text{Prob} \left( \sum_{a \in V(i)} u_{a \rightarrow i} > 0 \right) \quad (19)$$

通过分解因数近似, 这写作:

$$W_i^{(+)} = C_i \sum_{\{u_{b \rightarrow j}\}} \left[ \prod_{b \in V(j), i} Q_{b \rightarrow j}(u_{b \rightarrow j}) \right] \theta \left( \sum_{a \in V_+^i(i)} u_{a \rightarrow i} \right) \prod_{a \in V_-^i(i)} \delta(u_{a \rightarrow i}, 0) \quad (20)$$

为用收敛的权值  $\{\eta_{a \rightarrow i}^*\}$  来表示偏量, 引入这些量是方便的:

$$\prod_i^+ = \left[ 1 - \prod_{a \in V_+^i(i)} (1 - \eta_{a \rightarrow i}^*) \right] \prod_{a \in V_-^i(i)} (1 - \eta_{a \rightarrow i}^*)$$

$$\prod_i^- = \left[ 1 - \prod_{a \in V_-^i(i)} (1 - \eta_{a \rightarrow i}^*) \right] \prod_{a \in V_+^i(i)} (1 - \eta_{a \rightarrow i}^*)$$

$$\prod_i^0 = \prod_{a \in V(i)} (1 - \eta_{a \rightarrow i}^*) \quad (21)$$

则正偏量可表为:

$$W_i^{(+)} = \frac{\prod_i^+}{\prod_i^+ + \prod_i^- + \prod_i^0} \quad (22)$$

类似地, 当随机选取一个簇时, 变量  $i$  被约束(“固定”)到  $x_i = 0$  的概率称为它的负偏量

$$W_i^{(-)} = \frac{\prod_i^-}{\prod_i^+ + \prod_i^- + \prod_i^0} \quad (23)$$

选得变量在其中为约束不足或未固定(unfrozen)的簇的概率

$$W_i^{(0)} = 1 - W_i^{(+)} - W_i^{(-)} \quad (24)$$

5.4.2 变量类型 计算得总偏量  $\{W_i^{(\pm)}, W_i^{(0)}\}$  后, 我们可以区分三种变量结点(当然也包括中间类型): 约束不足变量, 其  $W_i^{(0)} \square 1$ ; 偏倚变量, 它的  $W_i^{(+)} \square 1$  或  $W_i^{(-)} \square 1$  和平衡变量, 它的  $W_i^{(+)} \approx W_i^{(-)}, W_i^{(0)}$  小。

固定一个偏倚变量不会改变解簇的结构,复杂度(解簇数目的对数)变化是平滑的(砍掉的簇很少)。固定一个约束不足变量对复杂度影响也不大。但固定一个平衡变量会几乎减少一半的解簇。自然地,消解算法在每一步将尽量固定最偏倚的,即具有最大 $|W_i^{(+)} - W_i^{(-)}|$ 的变量。

5.4.3 消解算法 使用 SP 提供的信息的一个策略是固定尽可能多的变量,同时不除去过多的簇。在每个消解步,待固定变量的选择依照偏量  $W$ 。问题被简化,SP 在这个子问题上重新运行,并导向一组新的  $\eta$ ,和新的偏量  $W$ 。这个过程反复进行下去。

#### SID algorithm

INPUT: 一个 SAT 问题的因子图表示以及最大迭代数  $t_{\max}$ , SP 中的要求精度  $\epsilon$ 。

OUTPUT: 一组满足所有子句的配置,或‘SP UN-CONVERGED’,或‘可能 UNSAT’

0. 随机初始化因子图每一边的警告纵览值

1. Run SP

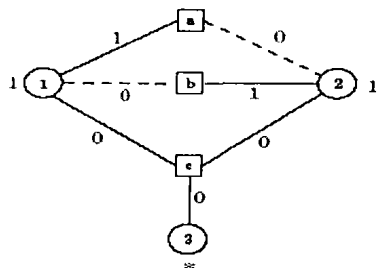
2. 使用公式(19-22)计算所有偏量  $\{W_i^{(+)}, W_i^{(-)}, W_i^{(0)}\}$ 。

3. 简化和搜索:

3.3 如果发现非平凡纵览( $\eta \neq 0$ ),则 固定具有最大 $|W_i^{(+)} - W_i^{(-)}|$ 的变量,并简化问题。

3.4 如果所有纵览是平凡的( $\eta = 0$ ),则输出简化后的公式,并用别的局部搜索工具(如 walksat)求解。

3.5 如果 SP 不收敛,输出‘可能 unsat’并停止(或重新开始,即 go to 0.)



4. 如果问题被单子句规则彻底解决,输出‘SAT’并停止。如果没有发现矛盾,消解过程在子问题上继续(go to 1.),否则(如果发现矛盾)停止。

消解算法可能有3种表现:

1. 算法几乎固定了所有变量的值(一些变量的值可能直到问题解决的时候,仍然未固定)。

2. 纵览在某个阶段收敛到空消息,对所有边。这种情况下,SP 不能提供任何更多的信息。幸运的是,这些问题通常是约束过少的(underconstrained),并且容易用其他方法(如 walksat)加以解决。这种阶段也称为顺磁态(paramagnetic)。

3. SP 算法在某个阶段,不能收敛,即使原来的问题可满足。

在较大的随机 SAT 实例上,在难满足区域,但不是非常接近 SAT/UNSAT 阈值的时候,实验证明算法表现为第二种行为。

当问题离 SAT/UNSAT 相变区很接近时,经常出现第3种情况。

#### 5.5 SP 算法的不完备性

SP 算法是否总可以收敛到一个唯一的的状态?回答是否定的。Ref<sup>[3]</sup>给出了一个简单的反例(图4)。

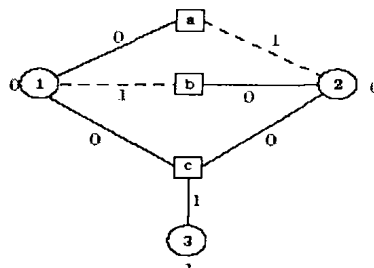


图4 一个简单的 SAT 问题,3个变量,2个2-子句,一个3-子句,和环。每个图给出了 BP 算法两个可能的解中的一个。每边  $a-i$  上的数字是警告  $u_{a-i}$ ,每个变量结点相邻的数是在两个“簇”上的各自相应的一般化状态。此图上 SP 方程有无穷个解,  $\eta_{a \rightarrow 1} = \eta_{b \rightarrow 2} = x$ ,  $\eta_{a \rightarrow 2} = \eta_{b \rightarrow 1} = y$ ,  $\eta_{c \rightarrow 1} = \eta_{c \rightarrow 2} = 0$ , and  $\eta_{c \rightarrow 3} = (1-x^2)y^2/(1-xy)^2$

## 6 不收敛情况分析和回溯

在 SAT/UNSAT 临界区(如  $\alpha=4.3$ ),SP 算法常常无法收敛,针对这种情况,G. Parisi 提出一种回溯策略<sup>[10]</sup>。其基本思想是,估计当前部分赋值下解空间大小  $N$ ,以及给每个变量  $i$  赋值后解空间大小  $N(i)$  占此空间的比例  $P(i) = N(i)/N$ ,选择变量  $i_M$ (使  $P(i_M) = P_M = \max_i P(i)$ ),记剩下的解空间大小  $N'$ ,再计算回溯每个变量  $k$  后解空间大小  $N'(k)$ ,二者的比例  $I(k) = N'/N'(k)$ 。如果

$$P_M = N'/N > I_m \equiv \min_k I(k) = \min_k N'/N'(k)$$

做一个联合移动,即给变量  $i_M$  赋值,并取消变量  $k_m$  ( $I(k_m) = I_m$ ) 的赋值。这一估量在每一步都进行。这样可能可以纠正由前面计算偏差导致的错误赋值。

而作者则采用一种更为简单的回溯策略并进行了实验,发现它是有效的。

具体地说,发现一个实例不收敛时,取消它最后  $m$  个变量赋值(作者实验采用  $m = \lceil N/100 \rceil$ ,  $\lceil \cdot \rceil$  为取整),然后交给 zchaff 求解。

对 sp 代码随机生成的100个实例( $n=2000-6000$ ,  $\alpha=4.1-4.3$ )进行求解,发现其中42个不收敛。这里面,又有9个实例,其残余问题(记为 noconvergence.tmp.cnf)用 zchaff 能几乎立即( $<5$ 秒)判断其为 UNSAT(其余33个,zchaff 运行5秒后 TIME OUT)。

对不收敛实例,在发现它不收敛时进行回溯,发现在 zchaff 能立即判断其为 UNSAT 的这类问题里,有7个经回溯后,zchaff 几乎立即( $<5$ 秒)求出它的一组解。而对于 zchaff TIME OUT 的不收敛问题,则通过回溯也未能得到解决。

这个现象,似乎也揭示出子问题收敛性和它的归约序列在 SAT/UNSAT 之间转化临界点(即文[3]等中的复杂度由正到负的转化)的某种关系。或者说,在可能的解很少(或没有)时,子问题不收敛的可能性较大。这和相变区附近 SP 的表现也是一致的。

## 7 实验结果

|     | 2000    | 3000    | 4000    | 5000    | 6000  |
|-----|---------|---------|---------|---------|-------|
| 4.1 | 10/10   | 8/8     | 8/8     | 6/6     | 6/6   |
| 4.2 | 1+1/1+2 | 4+2/4+2 | 3+1/3+2 | 5+1/5+3 | 6/7   |
| 4.3 | 0+1/0+7 | 0+1/0+6 | 0/0+7   | 0/0+6   | 0/0+7 |

说明:在形如  $a+b/c+d$  的表示中, $a$  表示 SP 收敛到顺磁态后,walksat 解出的实例数, $b$  表示到某阶段不收敛时,zchaff 解出的问题数, $c$  表示收敛的问题数, $d$  表示不收敛问题数。 $b$  与  $d$  为0时,省略不记。显见,总有  $a \leq c, b \leq d$ 。

这其中,zchaff 解出的问题均是通过回溯解决的。

而单纯的 walksat 算法,对于 SAT/UNSAT 相变区附近的较大问题,表现得相当无力(如  $n=2000-4000$ ,  $\alpha=4.2$ )。

**结论** 作为一类概率算法,消息传递算法(如 BP,SP)有其快速、高效的特点。但 BP 算法在 Hard-SAT 区域常常无法收敛。而改进后的 SP 算法则表现得相当出色。但 SP 算法在 SAT/UNSAT 相变区附近,出现不收敛,越过 SAT/UNSAT 相变区后(如  $\alpha=4.3$ ),则绝大情况仍然无法收敛。也许这是由于在此区域,解的个数过少(或者无解?),以至无法对这些解作有效的统计分析。作者采用的回溯策略,也只能是在一定程度上解决这个问题,而且这纯粹是经验性的。在 satlib 的一些难解结构化实例上(如 g125.18.cnf, g250.15.cnf 等),SP 也没有好的表现。常常出现 0/0,作者估计这可能是公式(20, 21)计算出的问题,即所有警告为 1。也可能在 SP 算法眼里,这些结构化实例是约束过强的。

## 参考文献

- 1 张健. 逻辑公式的可满足性判定一方法、工具及应用. 北京: 科学出版社, 2000
- 2 Braunstein A, Mezard M, Weigt M, Zecchina R. Constraint satisfaction by survey propagation. <http://fr.arxiv.org/abs/cond-mat/0212451>, Dec. 2002
- 3 Braunstein A, Mezard M, Zecchina R. Survey propagation: an algorithm for satisfiability. submitted to Random structures and algorithms. <http://fr.arxiv.org/abs/cs.CC/0212002>, Dec. 2002
- 4 Mezard M, Parisi G, Zecchina R. Analytic and Algorithmic Solution of Random Satisfiability Problems. Science, 2002, 297, 812

- 5 Parisi G. Some remarks on the survey decimation algorithm for K-satisfiability. <http://arxiv.org/abs/cs.CC/0301015>, Jan. 2003
- 6 Kschischang F R, Frey B J, Loeliger H A. Factor Graphs and the Sum-Product Algorithm. IEEE Trans. Inform. Theory 2002, 47: 498
- 7 Kirkpatrick S, Gelatt CD, Jr., Vecchi MP. Optimization by simulated annealing. Science, 1983(4598)
- 8 酆庆增. 自旋玻璃和复杂系统. 物理, 1992, 21(7)
- 9 Mezard M. Spin glasses: an introduction. In: Proc. of the Cargese summer institute on. From statistical physics to statistical inference and back, Sep. 1992
- 10 Parisi G. A backtracking survey propagation algorithm for K-satisfiability. <http://arxiv.org/abs/cond-mat/0308510> v1, Aug. 2003
- 11 Zhou Haijun. Vertex cover problem studied by cavity method: Analytics and population dynamics. Eur. Phys. J., 2003, B 32: 265~270
- 12 Weiss Y. Approximate inference using loopy belief propagation. Tutorial given at UAI 2001
- 13 Mezard M. Constraint satisfaction networks: Belief propagation and beyond. Invited Talks at NIPS 2003 Workshop on Robust Communication Dynamics in Complex Networks
- 14 Mezard M. Passing Messages Between Disciplines. SCIENCE, 2003, 301
- 15 Mertens S, Mezard M, Zecchina R. Threshold values of Random K-SAT from the cavity method. <http://fr.arxiv.org/abs/cs.CC/0309020>, Sep. 2003
- 16 Survey Propagation: Constraint Satisfaction via Message Passing, Constantine Caramanis, slides of MIT course 6.454: Graduate Seminar in Area I, Fall 2003

(上接第129页)

```

规则1:
%VAR[
    aimport, port1:Port
// 定义 Port 类型的变量, 代表目标端口
]
%Proc Rule1(aimport)[
    LB=START-Rule1=>$OLB=l1;
    LB=l1 & aimport==port1
    =>abandon() & $OLB=EXIT;
// 调用 abandon() 函数, 丢弃包
// port1 是用户定义的目标端口号
    LB=l1 & ~(aimport==port1)
    =>accept() & $OLB=EXIT;
// 调用 accept() 函数, 接收包
]

规则2:
%VAR[
    aimport, port1:Port
]
%Proc Rule2(aimport)[
    LB=START-Rule2=>$OLB=l1;
    LB=l1 & aimport<port1=>Rule35()
    & $OLB=EXIT;
// 调用规则35, 规则35的作用及描述省略
// port1 是用户定义的端口号
    LB=l1 & aimport>=port1=>$OLB=EXIT;
]

规则3:
%VAR[
    sourceaddress, low, high:IP Address,
// 定义 IP Address 类型的变量, 表示源地址
]
%Proc Rule3(sourceaddress, low, high)[
    LB=START-Rule3=>$OLB=l1;
    LB=l1 & AddBetween(low, high)
    =>accept() & $OLB=EXIT;
// AddBetween(low, high) 函数的作用是判断 IP 地址
// 是否在 low 和 high 之间, 这两个参数由用户设置
    LB=l1 ~ AddBetween(low, high)
    =>abandon() & $OLB=EXIT;
]

```

这三个规则要同时满足, 则要用公式(1)进行判断。由于  $(aimport == port1 \wedge AddBetween(low, high) == \$T) \wedge (abandon() \wedge accept() == \$F)$

所以有可能产生特征干扰。例如: 当一个包来自于 202.112.58.1, 目标端口为 21 时, 路由器根据规则 1 将其丢弃, 但这种行为破坏了规则 3 希望该包通过的意图, 即发生了特征

干扰问题。

**结束语** 本文中我们提出了一种基于构件组装方式的特征干扰检测方法。这种方法是根据构件组装的不同方式, 采用不同的规则检测特征干扰。需要指出的是, 由于对不同的应用系统和不同的用户, 其需求各不相同, 对构件间交互的期望也各不相同。所以我们在检测过程中采用的方法是尽量找出组装过程中可能出现的特征干扰, 让用户决定哪些是他所期望的, 哪些则是必须排除的。

在以后的研究中, 我们还有许多工作要做。一是我们只给出了几种组装方式下检测特征干扰的规则, 这些规则本身还不完善; 而其它构件组装方式中的特征干扰问题的方式和检测规则还有待研究。二是我们研究的只是简单构件组装过程中的特征干扰的检测, 在这种情况下的特征干扰一般都是比较明显的、容易被发现的; 在此基础上, 我们还要研究多个构件以多种不同方式组装时特征干扰的检测规则。

## 参考文献

- 1 李敏, 费丽娟, 胡金柱. 软件系统中特征干扰问题的研究. 计算机科学, 2003, 30(8, 增)
- 2 Pomakis K P, Atlee J M. Reachability Analysis of Feature Interactions: A Progress Report. In: The international symposium on software testing and analysis. California, United States, 1996
- 3 Heisel M, Souquieres J. A heuristic approach to detect feature interactions in requirements. In: Fifth Intl. Workshop on Feature Interactions in Telecommunications and Software Systems. Lund, Sweden, 1998
- 4 任洪敏, 钱乐秋. 构件组装及其形式化推导研究. 软件学报, 2003, 14(6): 1065~1072
- 5 唐稚松, 等. 时序逻辑程序设计与软件工程(上). 科学出版社, 1999
- 6 唐稚松, 等. 时序逻辑程序设计与软件工程(下). 科学出版社, 2002
- 7 任洪敏, 朱承, 钱乐秋. 基于时序逻辑软件构架形式化方法研究. 计算机科学, 2003, 30(5): 1~3
- 8 梅宏, 黄昱, 邢岩, 彭枫. 特征交互问题导论. 电子学报, 2002, 30(12A): 1923~1927