论 文

一种基于变量熵求解约束满足问题的置信传播算法

赵春艳*, 郑志明

北京航空航天大学数学与系统科学学院, 数学、信息与行为教育部重点实验室, 北京 100191

* 通信作者. E-mail: zhaocy@ss.buaa.edu.cn

收稿日期: 2011-07-28; 接受日期: 2011-09-01

国家自然科学基金 (批准号: 60973033) 和国家国际科技合作专项 (批准号: 2010DFR00700) 资助项目

摘要 在置信传播 (belief propagation, BP) 算法中, 提出一种基于变量熵来挑选变量从而固定变量赋值的策略, 用于求解一类具有增长定义域的随机约束满足问题. RB 模型是一个具有增长定义域的随机约束满足问题的典型代表, 已经严格证明它不仅存在精确的可满足性相变现象, 而且可以生成难解实例. 在 RB 模型上选取两组不同的参数进行数值实验. 结果表明: 在接近可满足性相变点时, BP 引导的消去算法仍然可以非常有效地找到随机实例的解; 不断增加问题的规模, 算法的运行时间呈指数级增长; 并且当控制参数 (约束紧度) 增加时, 变量的平均自由度逐渐降低.

关键词 约束满足问题 RB 模型 相变 置信传播 算法 熵

1 引言

约束满足问题 (constraint satisfaction problem, CSP) 是人工智能中的一个重要研究领域. 现实生活中的许多问题, 如调度、资源分配、模式识别、语言理解和诊断等等都可以用约束满足问题来建模. 近年来, 约束满足问题在理论计算机科学、信息论和统计物理中都受到了广泛关注和深入研究 [1~6].

一个 CSP 由 N 个变量和 M 个约束组成. 每个变量从其定义域 D 中取值; 每个约束包含从变量集合中随机挑选的 k 个不同的变量, 并且有一个相应的不协调集合来限制这些变量的某些赋值为不可满足的取值 [7]. 从可能的 C_N^k 个约束中随机挑选 M 个就构成这个 CSP 的一个随机实例. 求解这个随机实例就是至少找到一个解, 也就是这 N 个变量的一组赋值, 使得这 M 个约束同时被满足; 或者证明无解. 一般情况下, 随机 CSP 是 NP 完全问题. 大量的理论分析和实验研究表明: NP 完全问题存在可满足性相变现象, 即控制参数 (通常为约束密度 M/N) 存在一个阈值, 几乎所有的随机实例在小于这个阈值时都是可满足的 (satisfiable, SAT); 相反, 几乎所有的实例在大于这个阈值时都是不可满足的 (unsatisfiable, UNSAT) [8~13]. 自从相变现象被发现以来, 统计物理中的自旋玻璃理论就提供了一个非常有力的工具来研究随机 CSP, 尤其是像随机 K -satisfiable (K -SAT) 问题, 随机图中的 q -染色问题以及顶点覆盖问题等 [14~19]. 特别地, 基于无序系统中空腔场方法 (cavity method) 的置信传播 (belief propagation, BP) 算法和调查传播 (survey propagation, SP) 算法在约束密度接近相变点时, 仍

引用格式: 赵春艳, 郑志明. 一种基于变量熵求解约束满足问题的置信传播算法. 中国科学: 信息科学, 2012, 42: 1170–1180, doi: 10.1360/112011-693

然有非常出色的表现^[20~22], 弥补了以往的模拟退火算法、随机游走算法等在接近相变区域时就会失效的缺憾.

RB 模型是一个典型的具有增长定义域的随机约束满足问题. 在 RB 模型中, 变量定义域的规模随着变量数的增加呈多项式增长^[23], 这与以往研究的固定定义域规模的随机 CSP (如随机 K -SAT 问题等) 具有不同的特点. RB 模型是为了克服标准 CSP 模型中 B 模型的平凡渐近不可满足性^[24~26], 由 Xu 等于 2000 年提出^[23]. 具体来讲, RB 模型是对 B 模型的一个修改, 通过对变量定义域和约束紧度的限制, 避免了 B 模型不能产生难解实例的缺点. 文献 [23] 中严格证明了 RB 模型不但具有可满足性相变现象, 而且可以得到精确的相变点. 在文献 [27] 中, 分析了在有限规模尺度效应下, RB 模型的相变区域随着问题规模的增加而发生的变化. 文献 [28~30] 分别从理论和实验上说明在相变的阈值点附近, 求解 RB 模型随机实例的难度随着问题规模的增加呈指数级增长. 因此, 基于 RB 模型的难解算例被广泛地应用于各种算法竞赛. 最近, 文献 [31] 中提出了两种基于 BP 的信息传递算法用以求解具有增长定义域的随机 CSP. 这两种算法都是启发式的不完备算法. 将随机实例表示成因子图形式^[32], 算法是通过定义在因子图上的信息传递来实现的. 第一种算法中, 利用 BP 方程的结果, 一次就将所有的变量进行赋值, 然后结合局部搜索算法来求解; 另一种则是利用 BP 方程的结果, 每执行一次算法仅挑选一个变量进行赋值的一个逐步消去变量的过程.

本文基于文献 [31] 中第二种算法的思想, 提出一种新的挑选变量和固定变量赋值的策略. 在算法逐步消去变量的过程中, 每次都挑选具有最小熵的变量, 并将此变量的值固定到其具有最大边际概率的分量上. 针对 RB 模型, 选取两组不同的参数进行数值模拟. 实验结果表明: 这种消去策略与挑选具有最小分量熵的变量, 并将变量的取值固定到此分量上的策略相比, 具有显著的优越性. 本文中的算法在临近相变阈值时, 仍然可以快速有效地找到解. 增加问题规模, 算法的效率几乎不受影响. 事实上, 该算法不仅适用于具有增长定义域的随机 CSP, 而且对于那些变量定义域规模固定的随机 CSP 仍然有效.

2 问题描述

设有一个变量集合 $\{X_1, X_2, \dots, X_N\}$, 每个变量都从定义域 $D = \{s_1, s_2, \dots, s_{d_N}\}$ 中取值. 我们从变量集合中随机选取 k ($k \geq 2$) 个不同的变量 $\{X_a^1, X_a^2, \dots, X_a^k\}$ 构成约束 C_a . 对于每个约束 C_a , 都有一个相应的不协调赋值集合 $Q_a \subset D^k$ 来限制这 k 个变量的取值. 若这 k 个变量的联合取值属于集合 Q_a , 则称此约束是不可满足的; 反之, 则称约束是可满足的. 我们由以下两步生成 RB 模型:

- 从可能的 C_N^k 个约束中随机挑选 $M = rN \ln N$ 个约束构成约束集合 $\{C_a\}_{a=1}^M$, 其中 $r > 0$ 是常数;
- 对于每个约束 C_a , 从可能的 d_N^k 个赋值中随机不重复地挑选 $|Q_a| = pd_N^k$ 个作为不协调赋值集合, 其中 $0 < p < 1$ 称为约束紧度, $d_N = N^\alpha$, $\alpha > 0$ 为常数.

这 M 个约束就组成 RB 模型的一个随机实例. 如果存在这 N 个变量的一组赋值, 使得所有的约束同时被满足, 则这组赋值就是一个解. 在文献 [23] 中, Xu 等用一阶矩和二阶矩的方法, 严格证明了 RB 模型存在精确的可满足性相变现象. 具体来讲, 用 $\text{Pr}(\text{SAT})$ 表示 RB 模型的随机实例是可满足的, 则有以下结论成立:

定理 1^[23] 设 $p_s = 1 - e^{-\frac{\alpha}{k}}$. 如果 $\alpha > \frac{1}{k}$, $r > 0$ 均为常数, 并且 k, α, r 满足 $ke^{-\frac{\alpha}{k}} \geq 1$, 则

$$\lim_{N \rightarrow \infty} \text{Pr}(\text{SAT}) = \begin{cases} 0, & \text{如果 } p > p_s; \\ 1, & \text{如果 } p < p_s. \end{cases} \quad (1)$$

图 1 由二元 RB 模型生成的一个随机实例的因子图表示

Figure 1 The factor graph of a random instance of binary model RB

由此可见, 当变量数 N 充分大以后, RB 模型的随机实例在 p_s 处发生了可满足性的突变. 也就是说, 当 $p < p_s$ 时, RB 模型的随机实例以趋近于 1 的概率有解; 而当 $p > p_s$ 时, RB 模型的随机实例以趋近于 1 的概率无解. 文献 [28 ~ 30] 中指出, 几乎所有实例在相变点附近具有指数级的求解复杂度.

3 基于变量熵的置信传播算法

RB 模型在 $k \geq 2$ 时都是 NP 完全问题, 因此为了方便讨论, 我们以二元 RB 模型 ($k = 2$ 的情形) 为研究对象. 文中算法和结果可以推广到 $k > 2$ 的情形或者将 $k > 2$ 的情形归约为二元 RB 模型来进行研究.

3.1 二元 RB 模型的因子图表示

二元 RB 模型的随机实例可以表示成因子图形式, 如图 1 所示. 因子图是一个二部图, 包括变量结点集合 (图中用圆圈表示, 记作 $i, j, k, \dots$) 和约束结点集合 (图中用方块表示, 记作 $a, b, c, \dots$), 边 (i, a) 表示约束 a 包含变量 i . 在 RB 模型的因子图中, 变量结点的平均度 ($\equiv M/N$) 为 $r \ln N$. 由此可见, 因子图不是稀疏的, 而是带有许多圈结构的稠密图. 但是, 由于在平均意义下, 圈的长度随着变量数 N 的增加以 $\ln N / \ln \ln N$ 增长. 因此, 尽管二元 RB 模型的因子图不是严格的树状图, 但是当 N 充分大时, 它是局部树状结构的. 这一特性是得到 BP 方程的重要依据.

3.2 BP 迭代方程

在因子图的每条边 (i, a) 上, 我们可以定义变量发送给约束的信息 $u_{i \rightarrow a}(s_i)$ 以及约束发送给变量的信息 $\eta_{a \rightarrow i}(s_i)$. 其中, $u_{i \rightarrow a}(s_i)$ 表示在没有约束 a 的情况下, 变量 i 取值为 $s_i \in D$ 的概率; $\eta_{a \rightarrow i}(s_i)$

图 2 算法中变量的分类

Figure 2 Classification of the variable nodes

表示约束 a 要求变量 i 取值为 $s_i \in D$ 的概率. 根据统计物理中的空腔场理论, 可以得到下面的 BP 迭代方程:

$$u_{i \rightarrow a}^{(t)}(s_i) = \frac{1}{Z^{i \rightarrow a}} \prod_{b \in \partial i \setminus a} \eta_{b \rightarrow i}^{(t)}(s_i), \quad (2)$$

$$\eta_{a \rightarrow i}^{(t+1)}(s_i) = \frac{1}{Z^{a \rightarrow i}} \sum_{s_j \in D, j \in \partial a \setminus i} \delta_{(s_i, s_j) \in Q_a} u_{j \rightarrow a}^{(t)}(s_j), \quad (3)$$

其中, $Z^{i \rightarrow a}$ 和 $Z^{a \rightarrow i}$ 是归一化因子, $b \in \partial i \setminus a$ 和 $j \in \partial a \setminus i$ 分别表示与变量 i 相连接的约束结点集合除去 a 和与约束 a 相连接的变量结点集合除去 i , 函数 $\delta_{(s_i, s_j) \in Q_a}$ 取值为 0 如果 $(s_i, s_j) \in Q_a$ 成立, 否则取值为 1. 如果 BP 迭代方程收敛, 我们就可以根据信息 $\eta_{a \rightarrow i}(s_i)$ 计算出变量 i 取值的边际概率分布.

3.3 基于变量熵的置信传播算法

基于变量熵的 BP 算法是利用 BP 迭代方程的结果, 得到所有变量取值的边际概率分布, 计算变量熵; 然后挑选熵最小的变量, 将其值固定到概率最大的分量上. 不断地重复挑选变量进行赋值操作这一过程, 逐步消去所有的变量. 在算法执行过程中, 我们将变量分为 3 种类型, 如图 2 所示:

- A 类型: 已经被固定赋值的变量, 在图中变量上用 A 标记;
- B 类型: 与 A 类型的变量出现在同一个约束中的变量, 在图中变量上用 B 标记;
- C 类型: 其他变量, 在图中变量上用 C 标记.

基于变量熵的 BP 消去算法步骤如下所示:

输入: 二元 RB 模型一个随机实例的因子图, 最大迭代次数 $t_{\max}$ 和精度 ϵ .

输出: 实例的解或者算法不收敛.

Step 1 $t \leftarrow 0$, 对于每条边 (i, a) , 随机初始化约束到变量的信息 $\eta_{a \rightarrow i}^{(0)}(s_i) \in [0, 1]$;

Step 2 $t \leftarrow t + 1$, 对于每条边 (i, a) , 更新信息 $\eta_{a \rightarrow i}^{(t)}(s_i)$. 首先由 $\eta_{a \rightarrow i}^{(t-1)}(s_i)$, 根据方程 (2) 得到 $u_{i \rightarrow a}^{(t-1)}(s_i)$ (如果 $\partial i \setminus a = \emptyset$, 则我们令 $u_{i \rightarrow a}^{(t-1)}(s_i) = 1/d_N$). 然后将 $u_{i \rightarrow a}^{(t-1)}(s_i)$ 的值输入方程 (3) 得到 $\eta_{a \rightarrow i}^{(t)}(s_i)$;

Step 3 如果 $|\eta_{a \rightarrow i}^{(t)}(s_i) - \eta_{a \rightarrow i}^{(t-1)}(s_i)| < \epsilon$, 则令 $\eta_{a \rightarrow i}^*(s_i) = \eta_{a \rightarrow i}^{(t)}(s_i)$ 并转到Step 5; 如果 $t > t_{\max}$, 则输出不收敛;

Step 4 返回到Step 2;

Step 5 计算每个变量 i 取值的边际概率分布

$$\eta_i(s_i) = \frac{1}{Z^i} \prod_{a \in \partial i} \eta_{a \rightarrow i}^*(s_i), \quad (4)$$

其中 $Z^i = \sum_{s_i \in D} \prod_{a \in \partial i} \eta_{a \rightarrow i}^*(s_i)$;

Step 6 挑选变量进行赋值. 对于每个变量, 计算其熵 $S_i = -\sum_{s_i \in D} \eta_i(s_i) \log(\eta_i(s_i))$, 并挑选熵最小的变量 $i^* = \min_i S_i$, 将 i^* 的取值固定到分量 $s_i^* = \max_{s_i} \eta_{i^*}(s_i)$ 上;

Step 7 $t \leftarrow 0$, 对于每条边 (i, a) , 初始化信息 $\eta_{a \rightarrow i}^{(0)}(s_i)$

(a) 对于 A 类型的变量: 跳过;

(b) 对于 B 类型的变量: 如果 i 的取值 s_i 与固定变量 j^* 的取值有 $(s_i, s_j^*) \in Q_a$, 则我们令 $\eta_{a \rightarrow i}^{(0)}(s_i) = 0$; 否则 $\eta_{a \rightarrow i}^{(0)}(s_i) = 1$;

(c) 对于 C 类型的变量: 随机初始化 $\eta_{a \rightarrow i}^{(0)}(s_i) \in [0, 1]$;

Step 8 $t \leftarrow t + 1$

(a) 对于 A 类型的变量: 跳过;

(b) 对于 B 类型的变量: $\eta_{a \rightarrow i}^{(t)}(s_i) = \eta_{a \rightarrow i}^{(t-1)}(s_i)$;

(c) 对于 C 类型的变量: 对于每条边 (i, a) , 更新信息 $\eta_{a \rightarrow i}^{(t)}(s_i)$. 首先由 $\eta_{a \rightarrow i}^{(t-1)}(s_i)$, 根据方程 (2) 得到 $u_{i \rightarrow a}^{(t-1)}(s_i)$ (如果 $\partial i \setminus a = \emptyset$, 则我们令 $u_{i \rightarrow a}^{(t-1)}(s_i) = 1/d_N$). 然后将 $u_{i \rightarrow a}^{(t-1)}(s_i)$ 的值输入方程 (3) 得到 $\eta_{a \rightarrow i}^{(t)}(s_i)$;

Step 9 (a) 对于 A 类型的变量: 跳过;

(b) 对于 B 类型和 C 类型的变量: 如果 $|\eta_{a \rightarrow i}^{(t)}(s_i) - \eta_{a \rightarrow i}^{(t-1)}(s_i)| < \epsilon$, 则令 $\eta_{a \rightarrow i}^*(s_i) = \eta_{a \rightarrow i}^{(t)}(s_i)$ 并转到Step 11; 如果 $t > t_{\max}$, 输出不收敛;

Step 10 返回到Step 8;

Step 11 对于自由变量 (没有被固定的变量), 计算它取值的边际概率分布

$$\eta_i(s_i) = \frac{1}{Z^i} \prod_{a \in \partial i} \eta_{a \rightarrow i}^*(s_i), \quad (5)$$

其中 $Z^i = \sum_{s_i \in D} \prod_{a \in \partial i} \eta_{a \rightarrow i}^*(s_i)$;

Step 12 挑选变量进行赋值. 对于每个自由变量, 计算其熵 $S_i = -\sum_{s_i \in D} \eta_i(s_i) \log(\eta_i(s_i))$, 并挑选熵最小的变量 $i^* = \min_i S_i$, 将 i^* 的取值固定到分量 $s_i^* = \max_{s_i} \eta_{i^*}(s_i)$ 上;

Step 13 返回到Step 7, 直到所有变量都被赋值;

Step 14 对于每个约束 a , 如果变量的取值都有 $(s_i^*, s_j^*) \notin Q_a$, 则输出这组赋值作为解; 否则输出算法不收敛.

4 数值结果及分析

4.1 算法在二元 RB 模型上的应用

二元 RB 模型是具有可变定义域的随机 CSP 模型. 在数值实验中, 我们选取两组不同的参数, 分

表 1 数值实验中用到的第一组参数 (取 $\alpha = 0.8$, $r = 3$, 对于不同的 N , 相应的 d_N 和 M 值)

Table 1 The first group of parameters used in the numerical experiments*

N	α	d_N	r	M
20	0.8	11	3	180
40	0.8	19	3	443
60	0.8	26	3	737
80	0.8	33	3	1052
100	0.8	40	3	1382

* For given $\alpha = 0.8$ and $r = 3$, we show the domain size d_N of each variable and the number of constraints M for different N in this table

表 2 数值实验中用到的第二组参数 (取 $\alpha = 0.6$, $r = 1$, 对于不同的 N , 相应的 d_N 和 M 值)

Table 2 The second group of parameters used in the numerical experiments*

N	α	d_N	r	M
100	0.6	16	1	461
150	0.6	20	1	752
200	0.6	24	1	1060
250	0.6	27	1	1380
300	0.6	31	1	1711

* For given $\alpha = 0.6$ and $r = 1$, we show the domain size d_N of each variable and the number of constraints M for different N in this table

别为 $\alpha = 0.8$, $r = 3$ 和 $\alpha = 0.6$, $r = 1$. 对于不同的问题规模 N , 相应的其他参数如表 1 和 2 所示. 在算法中, 我们选取 $t_{\max} = 10^3$, $\epsilon = 10^{-4}$.

对于第一组参数 $\alpha = 0.8$, $r = 3$, 由定理 1 可以得到相变点 $p_s = 0.234$. 也就是理论上当 $N \rightarrow \infty$ 时, 若 $p < p_s$, 几乎所有二元 RB 模型的随机实例都有解; 反之, 若 $p > p_s$, 则几乎所有的随机实例都无解. 我们在 100 个随机生成的二元 RB 模型实例上运行算法, 结果如图 3(a) 所示. 我们可以看到, 算法在 $p < 0.155$ 时, 都能有效地找到随机实例的解; 而当约束紧度 p 越过 0.155 时, 算法便进入了 hard-SAT 区域. 此外, 算法的效率几乎不受实例规模 N 的影响. 对于另一组参数 $\alpha = 0.6$, $r = 1$, 相变点为 0.451. 图 3(b) 显示, 在 $p > 0.360$ 时, 算法几乎找不到任何随机实例的解.

文献 [31] 中, 采用了不同的策略对变量进行逐个消去. 即对于所有变量的每个分量, 计算分量熵 $S'_i(s_i) = -\eta_i(s_i) \log(\eta_i(s_i))$, 挑选值最小的变量 $i^* = \min_i S'_i(s_i)$, 并将 i^* 的值固定到分量 $\min_{s_i} S'_{i^*}(s_i)$ 上. 我们取 $N = 20$, 在由第一组参数 $\alpha = 0.8$, $r = 3$ 生成的 100 个随机实例上运行算法. 结果如图 3(a) 中插入图所示, 在 $p > 0.06$ 时, 算法就已经失效. 与之相比, 本文中使用的消去策略极大地提高了算法的性能, 使得算法在接近相变点时, 仍然可以快速有效地找到解.

图 4(a) 和 4(b) 分别展示了算法在 100 个随机实例上的运行时间随着问题规模 N 的增加而发生的变化. 其中图 4(a) 是针对 $\alpha = 0.8$, $r = 3$, $p = 0.120$, $N \in [20, 100]$ 的结果, 图 4(b) 是针对 $\alpha = 0.6$, $r = 1$, $p = 0.230$, $N \in [100, 300]$ 的结果. 从图中我们可以看出, 算法的运行时间随着 N 的增加以指数级形式增长.

在算法消去变量的过程中, 我们依次挑选一个熵最小的变量进行赋值, 并将赋值固定到其具有最

图 3 算法在 100 个随机实例上找到解的成功概率

Figure 3 The successful probability of the algorithm averaged over 100 random instances of binary model RB. (a) Results on the 1st group of parameters (the inset is cited in Ref. [31], in which a different decimation strategy is used in the algorithm); (b) results on the 2nd group of parameters

图 4 算法在 100 个随机实例上的运行时间随着 N 的增加而发生的变化

Figure 4 The running time (100 random instances) of the algorithm for different N . (a) For $\alpha = 0.8$, $r = 3$, we choose $p = 0.120$; (b) for $\alpha = 0.6$, $r = 1$, we choose $p = 0.230$

大概率分布的分量上. 如果这组变量的取值使得所有约束都被满足, 那么这组赋值就是一个解. 我们定义

$$f(p) = \frac{1}{N} \sum_{i^*=1}^N S_{i^*} \quad (6)$$

为变量的平均自由度, 用来度量随着约束紧度 p 的增加, 变量的赋值被凝固到某个分量上的强度. 图 5(a) 和 5(b) 分别是对 $\alpha = 0.8$, $r = 3$, $N = 20$ 和 $\alpha = 0.6$, $r = 1$, $N = 100$ 的运行结果. 图中表明所有变量的平均自由度随着 p 的增强逐渐减弱, 也就是当 p 在 0 附近时, 我们挑选一组变量赋值的自由度相对较高, 并且这组赋值成为解的概率也较大. 然而, 随着 p 的增加, 某些变量的赋值可能被凝固到极少数的分量上, 从而使得算法找到解的难度大大增加. 凝固变量的出现在目前被认为是使算法陷入困境的主要原因.

图 5 在算法运行过程中, 被固定赋值的变量的平均自由度随着 p 的增加发生的变化

Figure 5 In the decimation procedure, the average freedom of the N variables is plotted against p . (a) For $\alpha = 0.8$, $r = 3$ and $N = 20$; (b) for $\alpha = 0.6$, $r = 1$ and $N = 100$

表 3 算法在由 $\langle N, 5, 0.2, p_2 \rangle$ 生成的 100 个随机实例上的成功概率和运行时间

Table 3 Given $d = 5$, $p_1 = 0.2$, we show the successful probability and running time of the algorithm on 100 random instances generated following $\langle N, 5, 0.2, p_2 \rangle$ of model B

$\langle 20, 5, 0.2, p_2 \rangle$			$\langle 60, 5, 0.2, p_2 \rangle$			$\langle 100, 5, 0.2, p_2 \rangle$		
p_2	Success	Time (s)	p_2	Success	Time (s)	p_2	Success	Time (s)
0.00	1.00	0	0.00	1.00	19	0.00	1.00	125
0.05	1.00	2	0.05	1.00	56	0.05	1.00	407
0.10	1.00	1	0.10	1.00	82	0.10	0.00	1301
0.15	1.00	2	0.15	0.05	136	—	—	—
0.20	1.00	1	0.20	0.00	169	—	—	—
0.25	1.00	2	—	—	—	—	—	—
0.30	0.08	2	—	—	—	—	—	—
0.35	0.00	3	—	—	—	—	—	—

4.2 算法在 B 模型上的应用

事实上, 算法不仅在具有增长定义域的 RB 模型上有效, 在固定定义域的随机 CSP 模型上仍然具有良好的性能. 我们以标准 CSP 模型中的 B 模型^[26]为例, 来测试算法的效率. 用四元组 $\langle N, d, p_1, p_2 \rangle$ 来表示 B 模型: 其中 N 表示变量的数目; d 表示每个变量取值的个数, 是个与 N 无关的常数; 随机选取 $p_1 C_N^2$ 个二元约束组成 B 模型的一个随机实例; 对于每个约束, 随机不重复地选取 $p_2 d^2$ 个赋值对作为不协调赋值集合. 然而, Achlioptas 等在文献 [25] 中指出 B 模型存在一个缺点, 也就是它具有平凡的渐近无解性. 即若 $p_2 \geq 1/d$, 当 $N \rightarrow \infty$ 时, B 模型几乎必然存在一个有缺陷的 (flawed) 变量, 这个变量的任何取值都不能使其所在的约束是可满足的.

在数值实验中, 我们分别在由 $\langle N, 5, 0.2, p_2 \rangle$ 和 $\langle N, 10, 0.1, p_2 \rangle$ 生成的 100 个随机实例上运行算法. 取 $N = 20, 60, 100$, 对于不同的 p_2 , 算法成功找到解的概率和运行时间 (以秒计) 结果如表 3 和 4 所示. 从表中可以看出, 当 N 较小时, 在较大的 p_2 范围内, 算法可以快速有效地找到实例的解. 由于 B

表 4 取 $d = 10$, $p_1 = 0.1$, 算法在由 $\langle N, 10, 0.1, p_2 \rangle$ 生成的 100 个随机实例上的成功概率和运行时间

Table 4 Given $d = 10$, $p_1 = 0.1$, we show the successful probability and running time of the algorithm on 100 random instances generated following $\langle N, 10, 0.1, p_2 \rangle$ of model B

$\langle 20, 10, 0.1, p_2 \rangle$			$\langle 60, 10, 0.1, p_2 \rangle$			$\langle 100, 10, 0.1, p_2 \rangle$		
p_2	Success	Time (s)	p_2	Success	Time (s)	p_2	Success	Time (s)
0.00	1.00	1	0.00	1.00	20	0.00	1.00	99
0.05	1.00	1	0.05	1.00	48	0.05	1.00	227
0.10	1.00	2	0.10	1.00	49	0.10	1.00	286
0.15	1.00	1	0.15	1.00	52	0.15	1.00	336
0.20	1.00	1	0.20	1.00	56	0.20	1.00	448
0.25	1.00	2	0.25	1.00	63	0.25	0.71	633
0.30	1.00	1	0.30	1.00	75	0.30	0.00	1165
0.35	1.00	2	0.35	1.00	91	—	—	—
0.40	1.00	1	0.40	0.04	144	—	—	—
0.45	1.00	2	0.45	0.00	268	—	—	—
0.50	1.00	1	—	—	—	—	—	—
0.55	1.00	2	—	—	—	—	—	—
0.60	1.00	2	—	—	—	—	—	—
0.65	0.18	3	—	—	—	—	—	—
0.70	0.00	6	—	—	—	—	—	—

模型具有平凡渐近无解性, 因此随着 N 的增加, 显然算法有效的范围会减小.

4.3 算法分析

文中基于变量熵求解随机 CSP 的 BP 算法是一种启发式的不完备算法. 一旦 BP 方程收敛到不动点, 我们就可以得到每个变量取其分量值的边际概率分布, 从而可以计算每个变量的熵. 在算法固定变量的过程中, 我们每次均挑选具有最小熵的变量进行赋值. 变量的熵最小意味着在所有变量中, 这个变量取其分量值的边际概率分布最不均匀, 出现了具有极大边际概率的分量. 因此, 我们将熵最小的变量固定到其具有最大边际概率的分量上. 事实上, 每固定一个变量, 就相当于在赋值空间中选取了一个可能包含解的子集合, 直到所有变量都被固定, 我们就找到了一组可能为解的变量赋值. 如果这组赋值使得所有约束都被满足, 则为解; 否则可能我们将某个变量错误地固定到了其某个分量上. 若要进一步提高算法的效率, 我们可以结合更加复杂的策略, 如回溯等.

5 结论

本文研究了一类具有增长定义域的随机约束满足问题, 引入统计物理中基于空腔场方法的 belief propagation 来进行求解. 针对一个具有代表性的 RB 模型, 提出了由 BP 引导的信息传递算法. 在算法逐步消去变量 (固定变量的赋值) 的过程中, 我们采用的策略是挑选具有最小熵的变量, 并将变量赋值固定到边际概率最大的分量上. 实验结果表明, 这种策略极大地提高了算法的效率. 在将来的工作中, 我们将进一步利用 BP (如 reinforced BP) 来深入研究具有增长定义域的随机约束满足问题. 我们

希望这些方法的引入可以提供一种新的思路, 来更好求解随机约束满足问题和探讨其解空间的结构特征, 从而揭露 NP 完全问题的难解本质.

参考文献

- 1 Martin D, Alan F, Michael M. A probabilistic analysis of randomly generated binary constraint satisfaction problems. *Theor Comput Sci*, 2003, 290: 1815–1828
- 2 Creignou N, Daudé H. The SAT-UNSAT transition for random constraint satisfaction problems. *Discrete Math*, 2009, 309: 2085–2099
- 3 Martin O C, Monasson R, Zecchina R. Statistical mechanics methods and phase transitions in optimization problems. *Theor Comput Sci*, 2001, 265: 3–67
- 4 Achlioptas D, Moore C. Random k -SAT: two moments suffice to cross a sharp threshold. *SIAM J Comput*, 2006, 36: 740–762
- 5 Krzakala F, Montanari A, Ricci-Tersenghi F, et al. Gibbs states and the set of solutions of random constraint satisfaction problems. *Proc Natl Acad Sci*, 2007, 104: 10318–10323
- 6 Mézard M, Montanari A. *Information, Physics and Computation*. New York: Oxford University Press, 2009
- 7 Tsang E. A glimpse of constraint satisfaction. *Artif Intell Rev*, 1999, 13: 215–227
- 8 Xu K, Li W. The SAT phase transition. *Sci China Ser E: Tech Sci*, 1999, 42: 494–501
- 9 Fan Y, Shen J. On the phase transitions of random k -constraint satisfaction problems. *Artif Intell*, 2011, 175: 914–927
- 10 Hartmann A K, Weigt M. Statistical mechanics perspective on the phase transition in vertex covering of finite-connectivity random graphs. *Theor Comput Sci*, 2001, 265: 199–225
- 11 Achlioptas D, Peres Y. The threshold of random k -SAT is $2^k \log k - O(k)$. *J Am Math Soc*, 2004, 17: 947–973
- 12 Achlioptas D, Naro A, Peres Y. Rigorous location of phase transitions in hard optimization problems. *Nature*, 2005, 435: 759–764
- 13 Mertens S, Mézard M, Zecchina R. Threshold values of random k -SAT from the cavity method. *Random Struct Alg*, 2006, 28: 340–373
- 14 Monasson R, Zecchina R. Statistical mechanics of the random K -satisfiability model. *Phys Rev E*, 1997, 56: 1357–1370
- 15 Mézard M, Mora T, Zecchina R. Clustering of solutions in the random satisfiability problem. *Phys Rev Lett*, 2005, 94: 197205
- 16 Zdeborová L, Krzakala F. Phase transitions in the coloring of random graphs. *Phys Rev E*, 2007, 76: 031131
- 17 Weigt M, Zhou H. Message passing for vertex covers. *Phys Rev E*, 2006, 74: 046110
- 18 Zhou J, Zhou H. Ground-state of the random vertex-cover problem. *Phys Rev E*, 2009, 79: 020103(R)
- 19 Dall'Asta L, Pin P, Ramezanpour A. Statistical mechanics of maximal independent sets. *Phys Rev E*, 2009, 80: 061136
- 20 Braunstein A, Mézard M, Zecchina R. Survey propagation: an algorithm for satisfiability. *Random Struct Alg*, 2005, 27: 201–226
- 21 Chavas J, Furtlehner C, Mézard M, et al. Survey-propagation decimation through distributed local computations. *J Stat Mech*, 2005: P11016
- 22 Sulc P, Zdeborová L. Belief propagation for graph partitioning. *J Phys A: Math Theor*, 2010, 43: 285003
- 23 Xu K, Li W. Exact phase transitions in random constraint satisfaction problems. *J Artif Intell Res*, 2000, 12: 93–103
- 24 Smith B M, Dyer M E. Locating the phase transition in binary constraint satisfaction problems. *Artif Intell*, 1996, 81: 155–181
- 25 Achlioptas D, Molloy M S O, Kirousis L M, et al. Random constraint satisfaction: a more accurate picture. *Constraints*, 2001, 6: 329–344
- 26 Gent I P, Macintyre E, Prosser P, et al. Random constraint satisfaction: flaws and structure. *Constraints*, 2001, 6: 345–372
- 27 Zhao C, Zheng Z. Threshold behaviors of a random constraint satisfaction problem with exact phase transitions. *Inform Process Lett*, 2011, 111: 985–988
- 28 Xu K, Li W. Many hard examples in exact phase transitions. *Theor Comput Sci*, 2006, 355: 291–302

- 29 Liu T, Lin X, Wang C, et al. Large hinge width on sparse random hypergraphs. In: Walsh T, ed. Proceedings of the 22nd International Joint Conference on Artificial Intelligence. Barcelona: AAAI, 2011. 611–616
- 30 Xu K, Boussemart F, Hemery F, et al. Random constraint satisfaction: easy generation of hard (satisfiable) instances. *Artif Intell*, 2007, 171: 514–534
- 31 Zhao C, Zhou H, Zheng Z, et al. A message-passing approach to random constraint satisfaction problems with growing domains. *J Stat Mech*, 2011: P02019
- 32 Kschischang F R, Frey B J, Loeliger H A. Factor graphs and the sum-product algorithms. *IEEE Trans Inf Theory*, 2001, 47: 498–519

A belief-propagation algorithm based on variable entropy for constraint satisfaction problems

ZHAO ChunYan* & ZHENG ZhiMing

School of Mathematics and Systems Science, Key Laboratory of Mathematics, Informatics and Behavioral Semantics, Ministry of Education, Beihang University, Beijing 100191, China

*E-mail: zhaocy@ss.buaa.edu.cn

Abstract Belief propagation (BP) is a powerful technique that has been applied to solve constraint satisfaction problems (CSPs). In this paper, for solving random CSPs with growing domains, we propose a new strategy based on the variable entropy to fix variables in the procedure of BP decimation. It has been proved that model RB, a representative random CSP with growing domains, exhibits an exact satisfiability phase transition phenomenon, and all instances of model RB are hard at the threshold. We perform the algorithm on the instances of model RB with two different groups of parameters. Numerical results show that the algorithm guided by BP can find solutions efficiently for instances in the regime that is close to the threshold. The running time of the algorithm grows exponentially with the problem size. Besides, the average freedom of the variables decreases as the control parameter (constraint tightness) increases.

Keywords constraint satisfaction problems, model RB, phase transition, belief propagation, algorithm, entropy

ZHAO ChunYan was born in 1982. She received the Ph.D. degree in mathematics from Beihang University, Beijing in 2012. Currently, she is an Assistant Professor at University of Shanghai for Science and Technology. Her research interests include computational complexity, design and analysis of algorithms, constraint satisfaction problems and random graphs.

ZHENG ZhiMing was born in 1953. He received the Ph.D. degree in mathematics from Peking University, Beijing in 1987. Currently, he is a Professor and Vice President of Beihang University. His research interests include dynamical systems, computational complexity, information security and software trustworthiness.