

基于边际概率分布重新进行单变量选取的 置信传播算法求解约束满足问题

刘梦圆

上海理工大学理学院, 上海

收稿日期: 2024年3月13日; 录用日期: 2024年4月18日; 发布日期: 2024年5月31日

摘要

针对RB模型这一类具有增长取值域的随机约束满足问题, 提出一种基于边际概率分布重新进行单变量选取的置信传播算法。该算法在置信传播方程不收敛时, 通过边际概率分布顺序由大到小找到下一个变量进行重新赋值, 从而消去变量的过程。实验结果表明: 这种重新挑选变量进行赋值的置信传播算法能在可满足相变区域找到问题的解, 有效地提高了置信传播地求解效率。

关键词

约束满足问题, RB模型, 置信传播, 边际概率分布

Belief Propagation Algorithm for Solving Constraint Satisfaction Problem Based on Marginal Probability Distribution with Re-Selection of Univariate

Mengyuan Liu

School of Science, University of Shanghai for Science and Technology, Shanghai

Received: Mar. 13th, 2024; accepted: Apr. 18th, 2024; published: May 31st, 2024

Abstract

Aiming at the RB model which has a growing value range of stochastic constraint satisfaction problem, a belief propagation algorithm based on marginal probability distribution is proposed.

文章引用: 刘梦圆. 基于边际概率分布重新进行单变量选取的置信传播算法求解约束满足问题[J]. 理论数学, 2024, 14(5): 335-343. DOI: 10.12677/pm.2024.145190

When the belief propagation equation does not converge, the algorithm finds the next variable to be reassigned by the order of marginal probability distribution from large to small, so as to eliminate the process of variables. Experimental results show that the belief propagation algorithm based on reselecting variables for assignment can find the solution of the problem in the region that can satisfy the phase change, and effectively improves the efficiency of belief propagation.

Keywords

Constraint Satisfaction Problem, RB Model, Belief Propagation, Marginal Probability Distribution

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

约束满足问题(Constraint Satisfaction Problem, CSP)是人工智能领域的一个重要研究方向。约束满足问题的求解涉及到计算复杂性理论、图论、组合优化等多个学科领域,不仅在计算机科学、统计物理学、离散数学、信息论等领域中都具有非常重要的理论研究价值;而且在视觉处理、资源分配、时序推理、自然语言处理、机器学习等实际问题中也得到了广泛关注和研究。

CSP 涉及一组变量和这些变量需要满足的一组约束条件。每个变量都有一个非空的可能取值域,每个约束描述了一个变量子集以及子集之间各变量的不相容赋值集合, CSP 的目标是在满足所有约束条件的情况下,找到一组变量值(解)。N 皇后、图着色、合取式范式的可满足性(satisfiability, SAT)和最小问题点覆盖等都是 CSP。一般情况下,随机 CSP 是 NP 完全问题。为了研究 CSP,早期的许多理论研究和算法实验基本都是围绕四个标准的二元 CSP 模型 A、B、C、D 展开的[1] [2] [3], Achlioptas [4]等人指出这四个模型具有平凡渐近不可满足性的缺点,为了克服这一缺点, Molloy [5] [6]等人先后提出了一些改进模型,这些模型在理论上能产生非平凡的难解实例,但是它们约束的产生并不是简单而自然的。2000 年,许可和李未提出了 RB 模型[7] (revised B), RB 模型是一个典型的具有增长取值域的随机约束满足问题,在 RB 模型中,变量的取值域随问题规模的增大呈多项式级增长,这与以往具有固定增长域的随机约束满足问题存在不同的特点,克服了 B 模型不能产生难解实例的缺点,引起了国内外学者的广泛关注。许可等人[7]利用二阶矩方法严格证明 RB 模型不仅存在可满足性相变现象,而且可以得到精确的相变点。文献[8] [9]分别从理论和实验上说明了求解 RB 模型随机实例的难度随着问题规模的增加呈多项式级增长,证明了 RB 模型在相变点附近可以产生大量的难解实例。因此, RB 模型是一个非常值得研究的具有精确相变现象并且能从相变区域产生大量难解实例的 NP 完全问题。

RB 模型可满足相变现象已经被证明,但求解 RB 模型的随机实例仍然具有挑战性。2011 年,文献赵春艳等人引入统计物理学理论中的空腔场方法,提出了由置信传播算法引导的信息传播(Belief Propagation, 简称 BP)算法求解具有增长取值域的约束满足问题[10],这些算法在接近理论相变点时都能有效找到实例的解。赵春艳等人提出了一种基于变量熵来挑选变量的置信算法,且进一步探讨了 RB 模型的解空间[11]。王晓峰等人从理论上分析了置信传播算法在 RB 模型实例上的收敛性[12]。原志强等人提出了两种改进的模拟退火(Revised Simulated Annealing, 简称 RSA)算法求解增长值域的约束满足问题[13]。吴拔荣等人将置信传播和模拟退火相结合的算法求解 RB 模型[14]。李飞龙等人提出了基于禁忌搜索(Tabu Search, 简称 TS)算法求解 RB 模型并将禁忌搜索算法与置信传播算法相结合[15]。范如梦等人提出了基

于度启发式和最少约束值启发式的回溯算法(Heuristic Backtracking Algorithm, HBT), 通过度启发式来确定待赋值变量顺序, 然后利用最少约束值启发式对选择的变量进行赋值, 来求解 RB 模型[16]。赵春艳等人提出了不同约束紧度下 p-RB 模型存在精确的可满足性相变现象[17]。2022 年, 张学才等人提出两种基于动态度的回溯算法[18]求解 RB 模型, 与经典回溯算法相比, 具有显著的优越性。2023 年, 林童提出基于回溯的置信满足算法[19], 实验结果表明该算法能够更有效地找到问题的解。大量算法研究表明, 如何挑选变量和如何对变量赋值是解决问题的关键。

本文基于文献[10]中的算法, 提出了根据边际概率分布重新进行单变量选取的置信传播算法(New-selected belief propagation, NBP)来求解 RB 模型这一类具有可变量定义域的随机 CSP。在 NBP 算法中, 若 BP 方程收敛, 则利用变量的边际概率分布选取最大边际概率的变量作为待赋值的变量, 并令该变量取其最大边际概率分量的值。同时将待赋值变量根据边际概率分布进行降序排列, 以此作为挑选变量的顺序; 若迭代不收敛, 则重新挑选变量, 选择当前变量的后一位待赋值变量, 并将其值固定到该变量对应的最大边际概率的分量上, 再重新进行迭代。数值实验结果表明: 与 BP 算法相比, NBP 算法通过重新挑选待赋值变量可以找到更多实例的解, 提高了随机实例的求解概率。

2. RB 模型

RB 模型是由 N 个变量组成的变量集合 $X = \{x_1, x_2, x_3, \dots, x_N\}$ 和 t 个约束组成的约束集合 $C = \{C_1, C_2, C_3, \dots, C_t\}$ 构成的, 每个变量都从它们对应的定义域 $D = \{d_1, d_2, d_3, \dots, d_N\}$ 中取值, 其中 $|d_i| = d (i=1, 2, 3, \dots, N)$, $d = N^\alpha$, $\alpha > 0$ 为常数。RB 模型的随机实例 $RB(N, k, \alpha, r, p)$ 由以下步骤产生:

从 C_N^k 中随机可重复地选取 t 个约束, 其中 $t = rN \ln N$, $r > 0$ 为常数。每个约束 $C_a (1 \leq a \leq t)$ 包含了从变量集中随机挑选的 $k (k \geq 2)$ 个不同的变量构成的集合 X_a 和一个相应的不相容赋值集合来限制 X_a 中 k 个变量的某些赋值为不可满足的取值。其中 X_a 是从变量集合 X 中随机挑选的不重复的 k 个变量组成的; Q_a 是从 X_a 中所有可能的 d^k 个取值中随机挑选不重复的 pd^k 个 k 元赋值, $0 < p < 1$ 表示约束紧度。这 t 个约束就构成了 RB 模型的一个随机实例。求解这个随机实例就是至少找到一个解, 也就是找到同时满足 t 个约束的 N 个变量的一组赋值。

文献[7]中利用二阶矩方法严格证明了 RB 模型存在可满足性相变现象, 可以得到精确的相变点。用 $Pr(SAT)$ 表示 RB 模型生成的随机实例可满足的概率, 则有以下结论成立:

定理 1 [7] 设 $P_{cr} = 1 - e^{-\alpha/r}$, 若 $\alpha > 1/k$, $r > 0$ 是两个常数, 且 $ke^{-\alpha/r} \geq 1$, 则

$$\lim_{N \rightarrow \infty} Pr(SAT) = \begin{cases} 1 & p < p_{cr} \\ 0 & p > p_{cr} \end{cases} \quad (1)$$

定理 2 [7] 设 $r_{cr} = -\alpha/\ln(1-p)$, 若 $\alpha > 1/k$, $0 < p < 1$ 是两个常数, 且 $k \geq 1/(1-p)$, 则

$$\lim_{N \rightarrow \infty} Pr(SAT) = \begin{cases} 1 & r < r_{cr} \\ 0 & r > r_{cr} \end{cases} \quad (2)$$

这两个定理表明: 当 N 充分大时, 随着参数 p 或者 r 的增加, 当 $p > p_{cr} (r > r_{cr})$ 时, RB 模型随机实例可满足的概率趋近于 0; 当 $p < p_{cr} (r < r_{cr})$ 时, RB 模型随机实例可满足的概率趋近于 1。由此可见, RB 模型随机实例可满足的概率发生了从 1 到 0 的突变, 这就是可满足性相变现象, p_{cr} (或 r_{cr}) 为相变点。

3. 根据边际概率分布重新进行单变量选取的置信传播算法

3.1. RB 模型的因子图和 BP 迭代方程

RB 模型在 $k \geq 2$ 时都是 NP 完全的, 因此我们取 $k = 2$ 。将二元 RB 模型随机表示成因子图形式, 如

图 1 所示。圆圈表示变量节点，记作 i, j, k, \dots ；方框表示约束节点，记作 a, b, c, \dots ；若变量 i 在约束 a 中，则变量节点 i 与约束节点 a 之间用边连接，记作 (α, i) 。由于 RB 模型约束数为 $O(N \ln N)$ ，所以因子图具有局部树状结构。

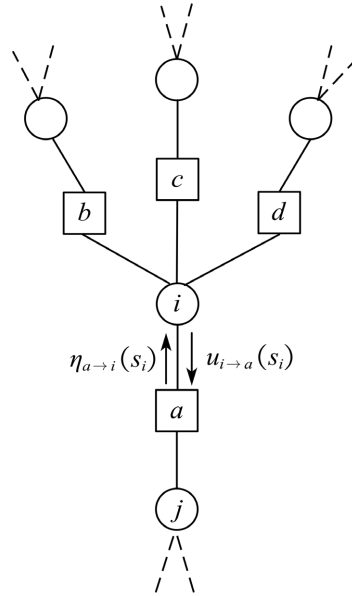


Figure 1. Factor graph of binary RB model
图 1. 二元 RB 模型的因子图

每条边上定义两种信息 $\eta_{a \rightarrow i}(s_i)$ 和 $u_{i \rightarrow a}(s_i)$ ， $\eta_{a \rightarrow i}(s_i)$ 是约束 a 发送给变量 i 的信息，表示约束 a 传递信息给变量 i 令其取值为 $s_i (s_i \in D)$ 的概率； $u_{i \rightarrow a}(s_i)$ 是变量 i 传递给约束 a 的信息，表示变量 i 在没有约束 a 的信息下取值为 $s_i (s_i \in D)$ 的概率。根据统计物理学中的空腔场理论，可得到 BP 迭代方程：

$$u_{i \rightarrow a}^{(t)}(s_i) = \frac{1}{Z^{i \rightarrow a}} \prod_{b \in \delta(i) \setminus a} \eta_{b \rightarrow i}^{(t)}(s_i), \quad (3)$$

$$\eta_{a \rightarrow i}^{(t+1)}(s_i) = \frac{1}{Z^{a \rightarrow i}} \sum_{s_j \in D, j \in \delta(a) \setminus i} \delta(s_i, s_j) u_{j \rightarrow a}^{(t)}(s_j), \quad (4)$$

其中：

$$\delta(s_i, s_j) = \begin{cases} 0, & (s_i, s_j) \in Q_a \\ 1, & (s_i, s_j) \notin Q_a \end{cases}, \quad (5)$$

这里 $Z^{i \rightarrow a}$ 和 $Z^{a \rightarrow i}$ 是归一化因子， $\delta(i)$ 表示与变量 i 相关的所有约束， $\delta(i) \setminus a$ 表示与变量 i 相关的约束除去约束 a ； $\delta(a)$ 表示约束 a 中包含的所有变量， $\delta(a) \setminus i$ 表示约束 a 除去变量包含的其他变量。

3.2. NBP 算法

事实上，文献[10]中提出的第二种算法是利用 BP 方程的结果，每次挑选一个变量进行赋值，逐步消去变量，但是 BP 方程会在接近相变点时出现不收敛的现象，从而使得算法失效，为了提高算法的求解效率，NBP 算法根据迭代方程计算边际概率，将概率最大的边际概率所对应的变量固定，令该变量取其最大边际概率分量的值，并按照每个变量的最大边际概率大小对变量进行排序，若迭代收敛，算法运行；若算法迭代不收敛或赋值不满足约束，将按照变量顺序对下一个变量进行赋值，将其固定到最大边际概

率分量上, 若变量顺序中无变量可挑选, 则结束算法, 通过边际概率重新挑选单变量进行赋值, 来提高算法收敛的可能性, 从而找到满足约束的一组赋值。

置信传播算法是在 BP 方程收敛后, 计算每个变量取值的边际概率分布, 通过先按照最大边际概率挑选变量再利用最大概率分量进行赋值方法执行算法, 在执行算法中, 将变量分为三类, **A 类型**: 已赋值的变量; **B 类型**: 与 A 类型的变量在同一个约束中的变量; **C 类型**: 其它变量。

NBP 算法步骤:

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

输出: 实例的解或者算法不收敛。

a) $n=1, t=0$, 随机初始化每条边上约束发送给变量的信息 $\eta_{a \rightarrow i}^{(0)}(s_i)$ 。

b) $t=t+1$, 将 $\eta_{a \rightarrow i}^{(t-1)}(s_i)$ 代入公式(3)计算得到每条边上变量传递给约束的信息 $u_{i \rightarrow a}^{(t-1)}(s_i)$ (如果 $\delta(i) \setminus a = \phi$, 则令 $u_{i \rightarrow a}^{(t-1)}(s_i) = 1/d$), 然后将 $u_{i \rightarrow a}^{(t-1)}(s_i)$ 代入公式(4)更新得到 $\eta_{a \rightarrow i}^{(t)}(s_i)$ 。

c) 如果 $|\eta_{a \rightarrow i}^{(t)}(s_i) - \eta_{a \rightarrow i}^{(t-1)}(s_i)| < \varepsilon$, 则令 $\eta_{a \rightarrow i}^*(s_i) = \eta_{a \rightarrow i}^{(t)}(s_i)$, 并执行步骤 e); 否则, 执行步骤 b)。

d) 如果 $t = t_{\max}$, 则输出不收敛。

e) 计算每个变量 i 的取值的边际概率

$$\eta_i(s_i) = \frac{\prod_{a \in \delta i} \eta_{a \rightarrow i}^*(s_i)}{\sum_{s_i \in D} \prod_{a \in \delta i} \eta_{a \rightarrow i}^*(s_i)}, \quad (6)$$

f) 挑选所有 $\eta_i(s_i)$ 中的最大值, 将其所对应的变量 i 作为第一个赋值的变量 x_1 , 对应的取值分量 s_i 赋给 x_1 , 并将变量按照 $\eta_i(s_i)$ 中最大边际概率降序排列, 得到所有变量的一个变量顺序 V_1 。

g) $n=n+1, t=0$, 初始化每条边上约束发送给变量的信息 $\eta_{a \rightarrow i}^{(0)}(s_i)$:

对于 A 类型的变量: 跳过;

对于 B 类型的变量: 如果变量 i 的取值 s_i 与约束中 A 类型的变量 j 的取值满足 $(s_i, s_j) \in Q_a$, 则 $\eta_{a \rightarrow i}^{(0)}(s_i) = 0$; 否则, $\eta_{a \rightarrow i}^{(0)}(s_i) = 1$;

对于 C 类型的变量: 随机初始化 $\eta_{a \rightarrow i}^{(0)}(s_i)$ 。

h) $t=t+1$

对于 A 类型的变量: 跳过;

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

对于 C 类型的变量: 将 $\eta_{a \rightarrow i}^{(t-1)}(s_i)$ 代入公式(3)计算得到每条边上变量发送给约束的信息 $u_{i \rightarrow a}^{(t-1)}(s_i)$ (如果 $\delta(i) \setminus a = \phi$, 则令 $u_{i \rightarrow a}^{(t-1)}(s_i) = 1/d$), 然后将 $u_{i \rightarrow a}^{(t-1)}(s_i)$ 代入公式(4)更新得到 $\eta_{a \rightarrow i}^{(t)}(s_i)$ 。

i) 如果 $|\eta_{a \rightarrow i}^{(t)}(s_i) - \eta_{a \rightarrow i}^{(t-1)}(s_i)| < \varepsilon$, 则令 $\eta_{a \rightarrow i}^*(s_i) = \eta_{a \rightarrow i}^{(t)}(s_i)$, 并执行步骤 k); 否则, 执行步骤 h)。

j) 如果 $t = t_{\max}$, 则 BP 方程不收敛, 执行步骤 n)。

k) 对于 B 类型和 C 类型的变量, 利用公式(6)计算变量取值的边际概率。

l) 挑选所有 $\eta_i(s_i)$ 中的最大值, 将其所对应的变量 i 作为第 n 个赋值的变量 x_n , 对应的取值分量 s_i 赋给 x_n , 并将 $\eta_i(s_i)$ 中所有边际概率降序排列, 再按照每个变量的最大边际概率排列, 得到此时的一个变量顺序 V_n , 且如果该变量的赋值不满足约束则执行步骤 n);

m) 如果 $n < N$, 执行步骤 g); 如果 $n = N$, 则找到上满足约束的一组赋值, 并输出这组赋值作为实例的解。

n) 根据变量 x_n 的变量顺序 V_n , 如果 V_n 中在变量 x_n 后还有变量可取, 则将 V_n 中当前赋值变量的后一位变量进行赋值, 若赋值满足约束, 则该变量成为第 n 个赋值变量, 并执行步骤 g); 如果该赋值不满足约束则执行步骤 n); 如果 V_n 无变量可取, 则输出未找到实例的解。

4. 数值结果及分析

在数值实验中, 取 $N \in \{20, 40, 60, 80\}$, $k=2$, $\alpha=0.8$, $r=3$ 。本文随机生成 50 个二元 RB 模型的实例。根据定理 1 可知问题的可满足性相变点是 $p_{cr}=0.234$ 。对于不同的变量数目 N , 变量定义域的大小 d 和约束数目 t 如表 1 所示。在算法中, 取最大迭代次数 $t_{\max}=10^3$, 精度 $\epsilon=10^{-4}$ 。

Table 1. Parameters of the binary RB model corresponding to different numbers of variables N

表 1. 二元 RB 模型对应不同的变量数目 N 时的参数

| N | α | r | d | M |
|-----|----------|-----|-----|------|
| 20 | 0.8 | 3 | 11 | 180 |
| 40 | 0.8 | 3 | 19 | 443 |
| 60 | 0.8 | 3 | 26 | 737 |
| 80 | 0.8 | 3 | 33 | 1052 |

4.1. 算法结果

本文在随机生成的 50 个二元 RB 模型随机实例上运行 NBP 算法, 算法找到解的概率如图 2 所示:

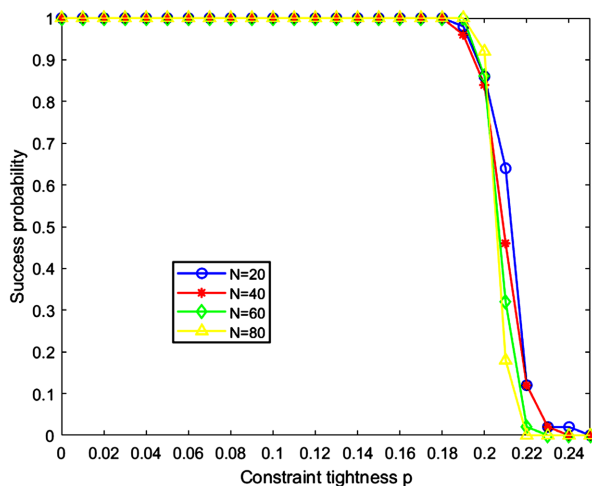


Figure 2. Probability of NBP algorithm solving on 50 random instances

图 2. NBP 算法在 50 个随机实例上的求解概率

从图 2 可以看出, 在 $N=20$ 时, NBP 算法在 $p \leq 0.18$ 时找到解的概率是 100%, 当 $p \geq 0.25$ 时找不到任何解; 在 $N=80$ 时, NBP 算法在 $p \leq 0.19$ 时找到解的概率是 100%, 当 $p \geq 0.22$ 时找不到任何解。

4.2. 算法分析

$N = \{20, 40, 60, 80\}$ 时, NBP 算法和 BP 算法在 50 个实例上的求解概率对比分别如图 3~6 所示。从图中可以看出, 与 BP 算法相比, NBP 算法可以明显提高对实例的求解概率, 尤其是在区域 $0.19 \leq p \leq 0.22$ 内, NBP 算法可以有效地找到更多有解实例。

在 $p=0.19$ 和处, NBP 算法和 BP 算法在求解二元 RB 模型的随机实例的运行结果对如图 7 所示。结果表明, 算法运行时间随变量数目的增加呈指数型增长。由于 NBP 算法重新挑选变量进行赋值, 则其运行时间明显多于 BP 算法的运行时间。

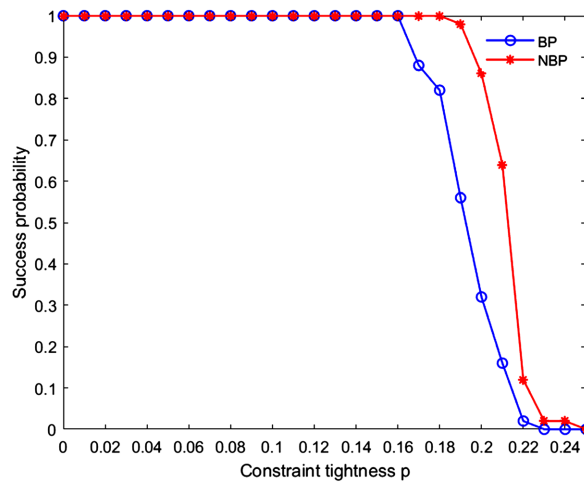


Figure 3. Comparison of results between NBP algorithm and BP algorithm at $N = 20$

图 3. $N = 20$ 时 NBP 算法和 BP 算法结果对比

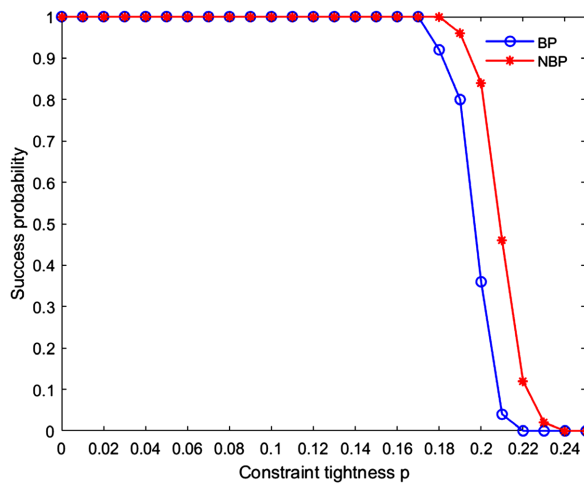


Figure 4. Comparison of results between NBP algorithm and BP algorithm at $N = 40$

图 4. $N = 40$ 时 NBP 算法和 BP 算法结果对比

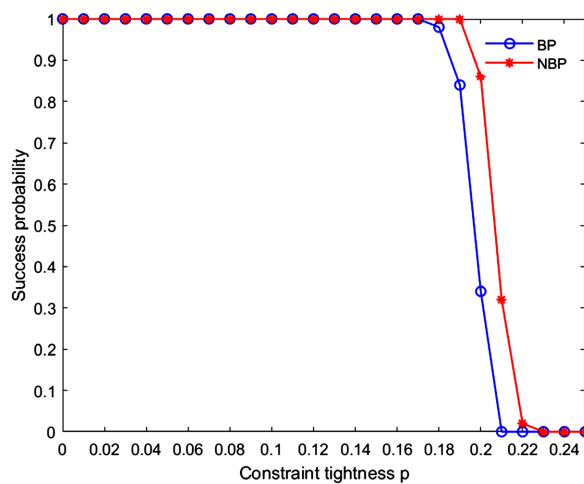


Figure 5. Comparison of results between NBP algorithm and BP algorithm at $N = 60$

图 5. $N = 60$ 时 NBP 算法和 BP 算法结果对比

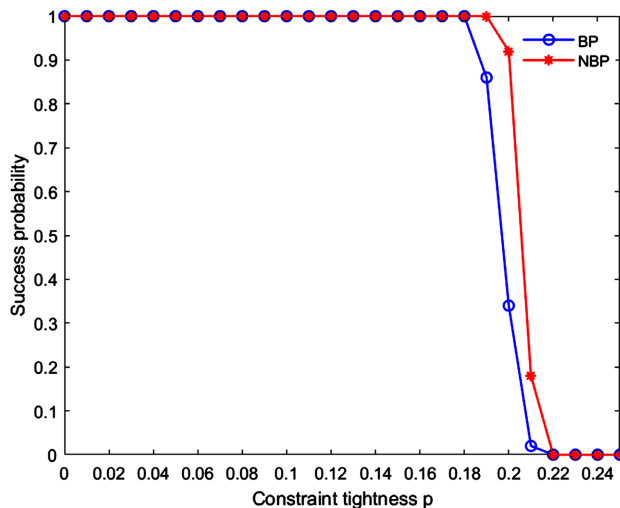


Figure 6. Comparison of results between NBP algorithm and BP algorithm at $N = 80$
图 6. $N = 80$ 时 NBP 算法和 BP 算法结果对比

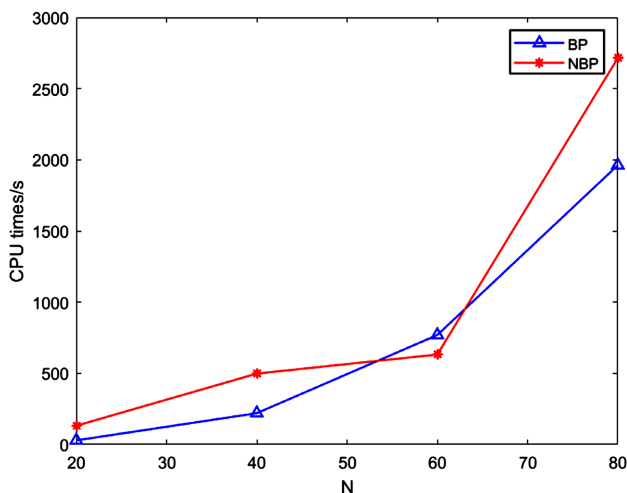


Figure 7. Comparison of runtime between NBP algorithm and BP algorithm for solving at $p = 0.19$
图 7. $p = 0.19$ 时 NBP 算法和 BP 算法求解的运行时间对比

5. 总结

本文提出了根据边际概率分布重新进行单变量选取的置信传播算法来求解具有可变定义域的随机约束满足问题。在 NBP 算法中，若 BP 方程迭代不收敛，则重新挑选变量，选择当前变量的后一位待赋值变量，并将其值固定到该变量对应的最大边际概率的分量上，再重新进行迭代，保证算法继续进行。数值实验结果表明，与 BP 算法相比，NBP 算法通过重新挑选待赋值变量可以找到更多实例的解，提高了随机实例的求解概率。但因为重新挑选变量赋值，增加了算法的运行时间，在以后的研究中可以通过进一步改进算法，提高算法的运行速度。

参考文献

- [1] Prosser, P. (1996) An Empirical Study of Phase Transitions in Binary Constraint Satisfaction Problems. *Artificial Intelligence*, **81**, 81-109. [https://doi.org/10.1016/0004-3702\(95\)00048-8](https://doi.org/10.1016/0004-3702(95)00048-8)
- [2] Smith, B.M. and Dyer, M.E. (1996) Locating the Phase Transition in Binary Constraint Satisfaction Problems. *Artifi-*

- cial Intelligence*, **81**, 155-181. [https://doi.org/10.1016/0004-3702\(95\)00052-6](https://doi.org/10.1016/0004-3702(95)00052-6)
- [3] Gent, I.P., Macintyre, E., Prosser, P., *et al.* (2001) Random Constraint Satisfaction: Flaws and Structure. *Constraints*, **6**, 345-372. <https://doi.org/10.1023/A:1011454308633>
- [4] Achlioptas, D., Molloy, M.S.O., Kirousis, L.M., *et al.* (2001) Random Constraint Satisfaction: A More Accurate Picture. *Constraints*, **6**, 329-344. <https://doi.org/10.1023/A:1011402324562>
- [5] Molloy, M. (2003) Models for Random Constraint Satisfaction Problems. *SIAM Journal of Computing*, **32**, 935-949. <https://doi.org/10.1137/S0097539700368667>
- [6] Frieze, A.M. and Molloy, M. (2006) The Satisfiability Threshold for Randomly Generated Binary Constraint Satisfaction Problems. *Random Structures and Algorithms*, **28**, 323-339. <https://doi.org/10.1002/rsa.20118>
- [7] Xu, K. and Li, W. (2000) Exact Phase Transitions in Random Constraint Satisfaction Problems. *Journal of Artificial Intelligence Research*, **12**, 93-103. <https://doi.org/10.1613/jair.696>
- [8] Xu, K. and Li, W. (2006) Many Hard Examples in Exact Phase Transitions. *Theoretical Computer Science*, **355**, 291-302. <https://doi.org/10.1016/j.tcs.2006.01.001>
- [9] Xu, K., Boussemart, F., Hemery, F. and Lecoutre, C. (2007) Random Constraint Satisfaction: Easy Generation of Hard (Satisfiable) Instances. *Artificial Intelligence*, **171**, 514-534. <https://doi.org/10.1016/j.artint.2007.04.001>
- [10] Zhao, C., Zhou, H., Zheng, Z. and Xu, K. (2011) A Message-Passing Approach to Random Constraint Satisfaction Problems with Growing Domains. *Journal of Statistical Mechanics: Theory and Experiment*, **2**, P02019. <https://doi.org/10.1088/1742-5468/2011/02/P02019>
- [11] 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[J]. 中国科学: 信息科学, 2012, 42(9): 1170-1180.
- [12] 王晓峰, 许道云. RB 模型实例集上置信传播算法的收敛性[J]. 软件学报, 2016, 27(11): 2712-2724.
- [13] 原志强, 赵春艳. 两种改进的模拟退火算法求解大值域约束满足问题[J]. 计算机应用研究, 2017, 34(12): 3611-3616.
- [14] 吴拔荣, 赵春艳, 原志强. 置信传播和模拟退火相结合求解约束满足问题[J]. 计算机应用研究, 2019, 36(5): 1297-1301.
- [15] 李飞龙, 赵春艳, 范如梦. 基于禁忌搜索算法求解随机约束满足问题[J]. 计算机应用, 2019, 39(12): 3584-3589.
- [16] 范如梦, 赵春艳, 李飞龙. 启发式回溯算法求解约束满足问题[J]. 计算机应用研究, 2020, 38(4): 1438-1442.
- [17] 赵春艳, 范如梦, 刘雅楠. 不同约束紧度下约束满足问题的相变现象[J]. 计算机应用研究, 2020, 37(12): 2739-2743.
- [18] 张学才, 赵春艳. 基于动态度的回溯算法求解大值域约束满足问题[J]. 计算机应用研究, 2022, 39(5): 1427-1431.
- [19] 林童. 一种基于回溯求解约束满足问题的置信传播算法[J]. 应用数学进展, 2023, 12(3): 993-1002. <https://doi.org/10.12677/aam.2023.123101>