

# 一种基于置信传播的算法求解随机约束满足问题

刘梦圆

上海理工大学理学院, 上海

收稿日期: 2024年4月16日; 录用日期: 2024年5月17日; 发布日期: 2024年6月14日

## 摘要

为了求解具有增长域的随机约束满足问题(CSP), 提出一种基于置信传播的算法即NBP\* (new-selected belief propagation\*, NBP\*)。在置信传播算法中, 当BP方程不收敛时, 算法就会终止。然而算法在经过多次迭代之后, 虽然约束发送给变量的信息没有达到收敛条件, 但是仍有部分信息是准确的, 所以当算法的BP方程不收敛时, 提出利用最后一次迭代得到的约束发送给变量的信息来计算变量的边际概率, 当赋值不满足约束时, 根据边际概率确定的变量顺序挑选下一个变量进行赋值, 得到NBP\*算法。数值实验表明: 这种算法可以在可满足性相变区域找到解, 并且有效提高了置信传播算法的求解效率。

## 关键词

约束满足问题, 置信传播算法, BP方程, 最后一次迭代信息

# A Belief Propagation Based Algorithm for Solving Stochastic Constraint Satisfaction Problem

Mengyuan Liu

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

Received: Apr. 16<sup>th</sup>, 2024; accepted: May 17<sup>th</sup>, 2024; published: Jun. 14<sup>th</sup>, 2024

## Abstract

In order to solve the stochastic constraint satisfaction problem (CSP) with a growing domain, a belief propagation based algorithm called NBP\* (new selected belief propagation\*, NBP\*) is pro-

posed. In the belief propagation algorithm, when the BP equation does not converge, the algorithm will terminate. However, after multiple iterations, although the information sent by the constraint to the variable did not meet the convergence condition, some information was still accurate. Therefore, when the BP equation of the algorithm does not converge, it is proposed to use the information sent by the constraint to the variable from the last iteration to calculate the marginal probability of the variable. When the assignment does not meet the constraint, the next variable is selected according to the variable order determined by the marginal probability for assignment, resulting in the NBP\* algorithm. Numerical experiments have shown that this algorithm can find solutions in the satisfiable phase transition region and effectively improve the efficiency of the confidence propagation algorithm.

## Keywords

Constraint Satisfaction Problem, Belief Propagation Algorithm, BP Equation, Last Iteration Information

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. 绪论

约束满足问题(Constraint Satisfaction Problem, CSP)是人工智能和计算机科学领域一个非常重要的研究对象。在计算复杂性理论中, NP-完全问题都可以表述为 CSP, 例如子集加总问题、布尔可满足性问题、顶点涵盖问题、图着色问题、N-puzzle 问题等, 因此 CSP 具有非常高的研究价值。

对于 CSP 的研究, 早期主要是在 A, B, C, D 四种的模型[1] [2] [3]的基础上展开的。但之后 Achlioptas 等人指出随着问题规模的增大, 这四种模型会表现出平凡渐近无解性[4]。为解决这一问题, 学者们先后提出了许多改进的模型[4] [5] [6] [7] [8]。2000 年, Xu 等对 B 模型进行了修改, 提出了 RB (Revised B)模型[5], 该模型克服了 B 模型的平凡渐近无解性, 证明了 RB 模型存在精确的可满足性相变现象, 并且 RB 模型可以在相变区域产生大量难解的实例[9] [10]。

由于 RB 模型具有精确相变, 并且它的取值域会随着问题规模增大而增大, 所以受到了国内外学者的广泛关注, 成为国际上应用最为广泛的难解实例的产生模型, 被广泛应用于 ACM 等国际算法竞赛。自从相变现象被发现后, 统计物理学中的自旋玻璃理论就提供了一个有力的研究工具, 在接近相变点时, 基于物理学中的无序系统的置信传播(belief propagation, BP)算法展现出惊人的结果[11], 越来越多的学者利用消息传递算法中的置信传播算法来解决 CSP。文献[12]中提出两种基于置信传播的信息传递算法求解 RB 模型产生的随机实例。文献[13]赵春艳等人提出了一种基于变量熵的 BP 算法。文献[14]中提出在 BP 方程加入惩罚值的置信传播算法。2019 年, 文献[15]提出置信传播和模拟退火相结合的算法求解 RB 模型。文献[16]中, 提出一种基于异步更新的置信传播算法求解 RB 模型。文献[17]中提出一种基于残差的消息传递算法。以上算法都不同程度地提高了置信传播算法(belief propagation algorithm, BP)的性能。

本文基于文献[12]的算法, 在置信传播方程迭代不收敛时, 提出一种基于置信传播的算法即 NBP\* (new-selected belief propagation\*, NBP\*)。若算法迭代收敛, 则将具有最大边际概率的变量固定, 并令该变量取边际概率最大的分量的值, 同时按照边际概率由大到小排列, 确定变量的顺序, 若迭代不收敛, 则利用最后一次迭代信息的结果计算边际概率, 当赋值不满足约束时, 根据边际概率确定的变量顺序挑

选下一个变量进行赋值，并将其固定到边际概率最大的分量上，从而保证算法继续进行。数值实验结果表明，与 BP 算法相比，NBP\*算法可以找到更多的有解实例。

## 2. 预备知识

### 2.1. RB 模型

CSP 定义为一个三元组  $\langle X, D, C \rangle$ ，其中  $X$  是一组变量，记为  $X = \{x_1, x_2, x_3, \dots, x_N\}$ ， $D$  表示值域，记为  $D = \{d_1, d_2, d_3, \dots, d_N\}$ ， $C$  表示  $t = rN \ln N$  个约束组成一约束，记为  $C = \{C_1, C_2, C_3, \dots, C_t\}$ ，每个变量  $x_i$  都从它们对应的取值域  $d_i$  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  $RB(N, k, \alpha, r, p)$  由以下两个步骤产生：

a) 从  $C_N^k$  中随机可重复地选取  $t$  个约束，每个约束  $C_a (1 \leq a \leq t)$  包含了从变量集合中随机挑选的  $k (k \geq 2)$  个不同的变量构成的集合  $X_a$ 。

b) 对每个约束  $C_a (1 \leq a \leq t)$ ，选定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找到同时满足  $t$  个约束的变量的一组赋值。

### 2.2. RB 模型可满足相变现象

文献[5]中利用二阶矩方法严格证明了 RB 模型存在精确的可满足性相变现象。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 [5] 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

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

定理表明：当  $N$  充分大时，随着参数  $p$  的增加，当  $p > p_{cr}$  时，RB 模型随机实例可满足的概率趋近于 0；当  $p < p_{cr}$  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 降到 0，这就是可满足性相变现象， $p_{cr}$  是相变点。

### 2.3. RB 模型的因子图

由于 RB 模型在  $k \geq 2$  时都是 NP 完全的，一般地，我们在生成二元 RB 模型随机实例时，取  $k = 2$  即可。将二元 RB 模型随机表示成因子图形式，如图 1 所示。用圆形表示变量节点，记作  $i, j, k, \dots$ ；用正方形表示约束节点，记作  $a, b, c, \dots$ ；若约束  $\alpha$  包含变量  $i$  中，则用边连接变量节点  $i$  与约束节点  $\alpha$ ，记作  $(\alpha, i)$ 。

每条边上包含了约束  $a$  发送给变量  $i$  的信息  $\eta_{a \rightarrow i}(s_i)$  和变量  $i$  传递给约束  $a$  的信息  $u_{i \rightarrow a}(s_i)$ ， $\eta_{a \rightarrow i}(s_i)$  表示约束  $a$  传递信息让变量  $i$  取值为  $s_i (s_i \in D)$  的概率； $u_{i \rightarrow a}(s_i)$  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。

### 2.4. BP 迭代方程

根据统计物理学中的空腔场理论，可得到 BP 迭代方程：

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

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

其中:

$$\delta(s_i, s_j) = \begin{cases} 0, & (s_i, s_j) \in Q_a \\ 1, & (s_i, s_j) \notin Q_a \end{cases}, \quad (4)$$

$Z^{i \rightarrow a}$  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  $i$  相关的不包括约束  $a$  的其余相关的约束;  $\delta(a)$  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  $i$  的其他变量。

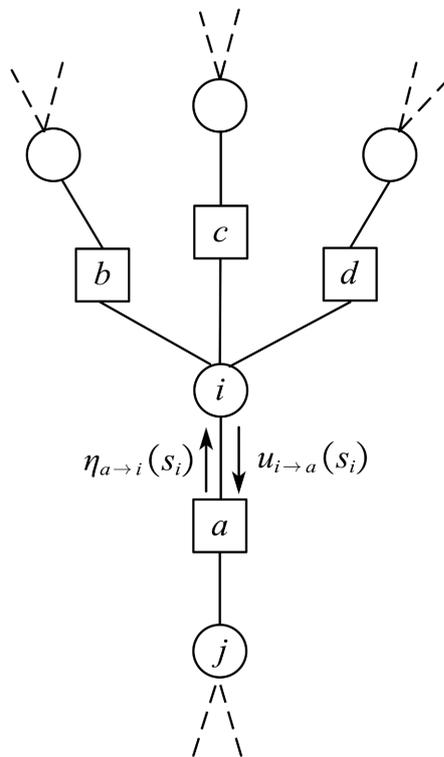


Figure 1. Factor graph of binary RB model

图 1. 二元 RB 模型的因子图

### 3. NBP\*算法

文献[12]中的算法,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,并且提出的算法表明BP方程会在接近相变点时出现不收敛现象,从而使得算法失效。然而在BP方程迭代到最大迭代次数时,虽然约束传递给变量的信息没有达到收敛条件,但是有一部分信息是准确的,此时根据最后一次迭代的信息计算边际概率分布,对变量进行赋值,再判断赋值是否满足所有约束;当赋值不满足约束时,根据边际概率确定的变量顺序挑选下一个变量进行赋值,从而增加算法收敛的可能性,最后找到满足约束的一组赋值。

在执行算法时,将变量分为三种类型,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  $\varepsilon$ 。

输出: 实例的解或者算法不收敛。

- 1)  $n=1$ ,  $t=0$ , 随机初始化每条边上约束发送给变量的信息  $\eta_{a \rightarrow i}^{(0)}(s_i)$ 。
- 2)  $t=t+1$ , 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  $\eta_{a \rightarrow i}^{(t)}(s_i)$ 。
- 3) 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 5); 否则, 执行步骤 2)。
- 4) 如果  $t = t_{\max}$ , 则令  $\eta_{a \rightarrow i}^*(s_i) = \eta_{a \rightarrow i}^{(t)}(s_i)$ 。
- 5) 计算每个变量  $i$  的取值的边际概率

$$\eta_i(s_i) = \frac{\prod_{a \in \delta(i)} \eta_{a \rightarrow i}^*(s_i)}{\sum_{s_i \in D} \prod_{a \in \delta(i)} \eta_{a \rightarrow i}^*(s_i)} \quad (5)$$

6) 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  $x_1$ , 并将每个变量按照  $\eta_i(s_i)$  中的边际概率降序排列, 再取每个变量的最大边际概率降序排列, 得到所有变量的一个顺序  $V_1$ 。

7)  $n=n+1$ ,  $t=0$ , 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  $\eta_{a \rightarrow i}^{(0)}(s_i)$ 。

8)  $t=t+1$

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  $\eta_{a \rightarrow i}^{(t)}(s_i)$ 。

9) 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 11); 否则, 执行步骤 8)。

10) 如果  $t = t_{\max}$ , 则令  $\eta_{a \rightarrow i}^*(s_i) = \eta_{a \rightarrow i}^{(t)}(s_i)$ 。

11) 对于 B 类型和 C 类型的变量, 利用公式(5)计算变量取值的边际概率。

12) 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  $x_n$ , 并将每个变量按照  $\eta_i(s_i)$  中的边际概率降序排列, 再取每个变量的最大边际概率降序排列, 得到所有变量的一个顺序  $V_n$ , 且如果该变量  $i$  的赋值不满足约束则执行步骤 n)。

13) 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。

14) 根据确定变量  $x_n$  时的变量顺序  $V_n$ , 如果  $V_n$  有变量未赋值, 则将  $V_n$  中当前变量的后一位变量进行赋值, 并且将该变量作为第  $n$  个待赋值的变量, 如果该赋值不满足约束则执行步骤 14); 如果该赋值满足约束则执行步骤 7); 如果  $V_n$  中无变量可取, 则输出没找到实例的解。

#### 4. 数值结果及分析

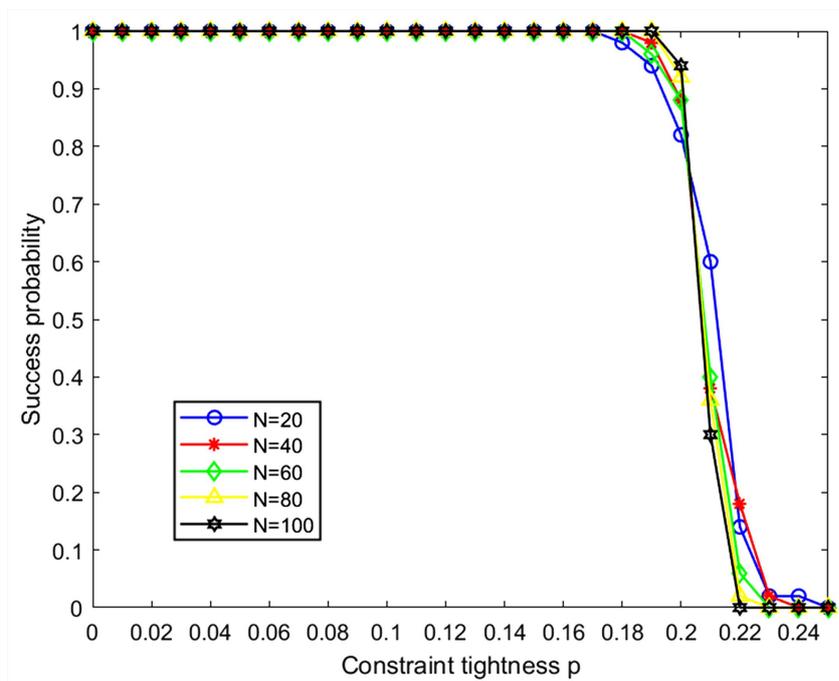
在数值实验中, 本文取  $N \in \{20, 40, 60, 80, 100\}$ ,  $k=2$ ,  $\alpha=0.8$ ,  $r=3$  生成 50 个二元 RB 模型的实例。由定理 1 可以推出二元 RB 模型的可满足性相交点是  $p_s \approx 0.234$ 。对于不同的变量数目  $N$ , 变量定义域的取值域  $d$  和约束的数目  $t$  如表 1 所示。在算法中, 取最大迭代次数  $t_{\max} = 10^3$ , 精度  $\epsilon = 10^{-4}$ 。

**Table 1.** The parameters of binary RB model under different problem scales  $N$   
**表 1.** 二元 RB 模型在不同问题规模  $N$  下的参数值

$N$	$\alpha$	$r$	$d$	$t$
20	0.8	3	11	180
40	0.8	3	19	443
60	0.8	3	26	737
80	0.8	3	33	1052
100	0.8	3	40	1382

#### 4.1. NBP\*算法的结果

在数值实验中, 取  $N \in \{20, 40, 60, 80, 100\}$  时, 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.** The 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  $p \leq 0.17$  时所有实例都可以找到解, 当  $p \geq 0.25$  时算法失效; 在  $N = 100$  时, NBP\*算法在  $p \leq 0.19$  时所有实例都可以找到解, 当  $p \geq 0.22$  时算法失效。

#### 4.2. 算法分析

NBP\*算法和 BP 算法在 50 个实例上的求解概率对比分别如图 3~7 所示。

从图 3~7 中可以清楚地看出, 与 BP 算法相比, NBP\*算法的求解效率有明显提高。当  $0.19 \leq p \leq 0.22$  内, NBP\*算法可以找到更多有解实例, 这就说明了当置信传播方程不收敛, 仍有部分信息是准确的, 可以利用这些信息来提高算法的求解效率。当变量数目  $N = \{20, 40, 60, 80\}$ , 可以通过图 3~6 看出, 当 BP 算法在接近可满足性相变点时失效, 而 NBP\*算法却仍然可以找到一部分实例的解。

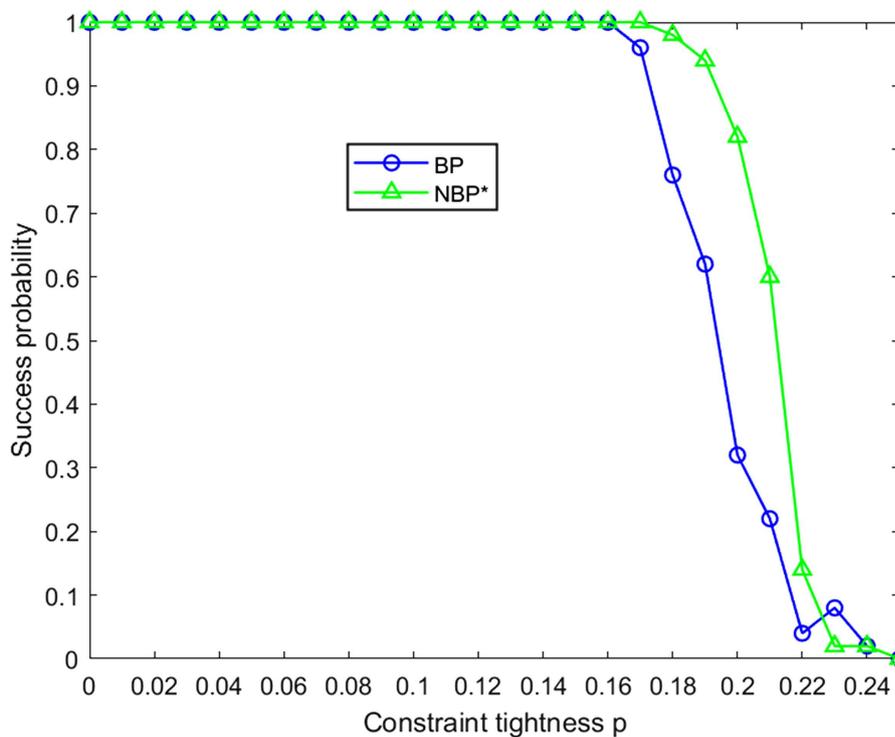


Figure 3. Comparison of NBP\* algorithm and BP algorithm results at  $N = 20$   
图 3.  $N = 20$  时 NBP\*算法和 BP 算法结果对比

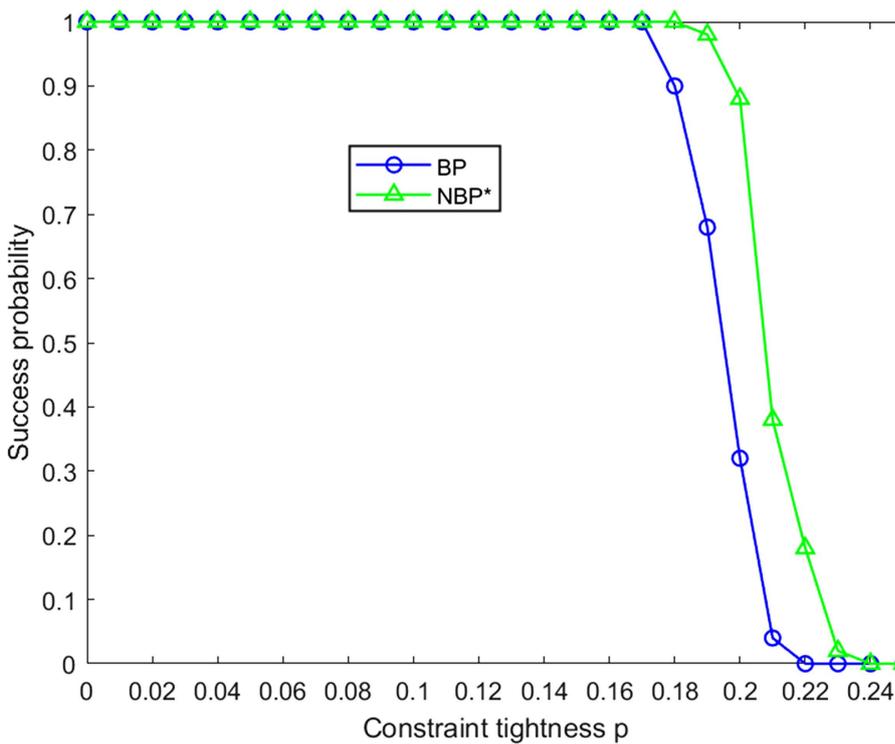


Figure 4. Comparison of NBP\* algorithm and BP algorithm results at  $N = 40$   
图 4.  $N = 40$  时 NBP\*算法和 BP 算法结果对比

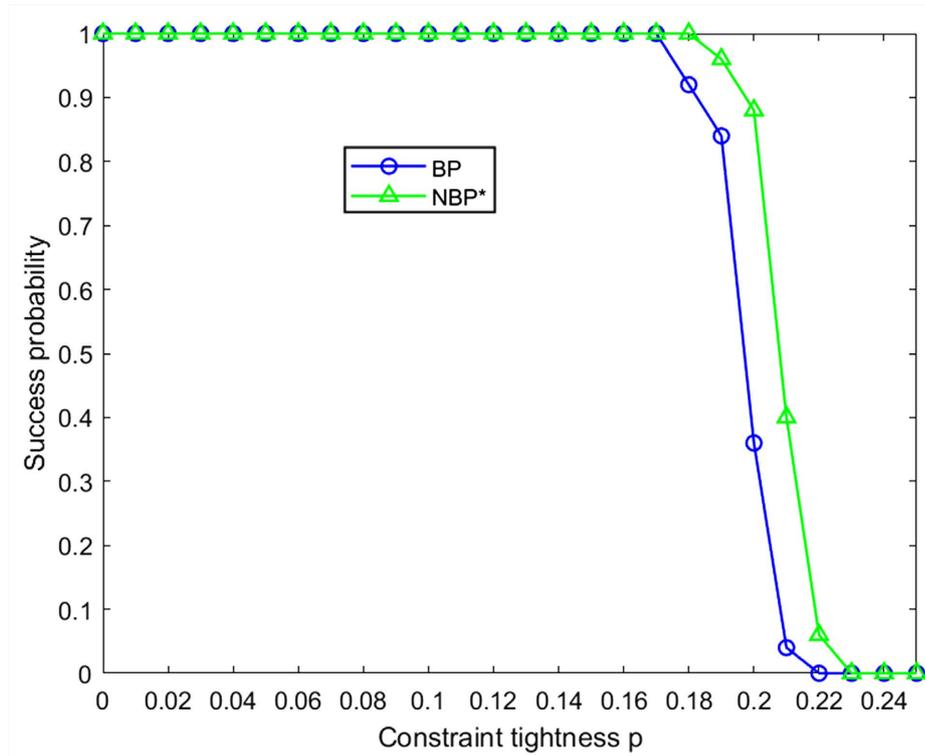


Figure 5. Comparison of NBP\* algorithm and BP algorithm results at  $N = 60$

图 5.  $N = 60$  时 NBP\*算法和 BP 算法结果对比

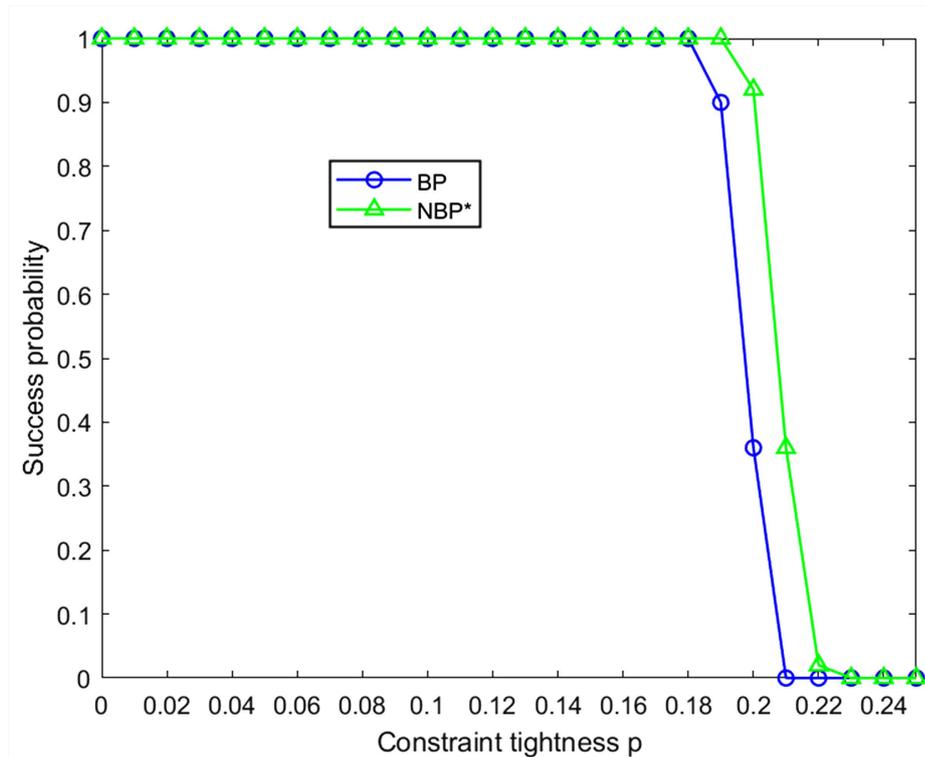


Figure 6. Comparison of NBP\* algorithm and BP algorithm results at  $N = 80$

图 6.  $N = 80$  时 NBP\*算法和 BP 算法结果对比

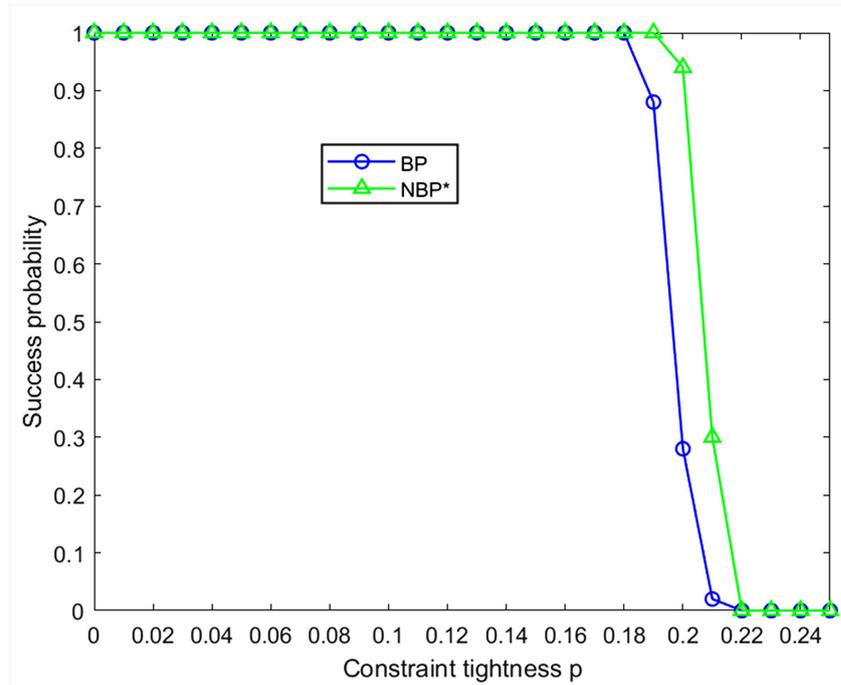


Figure 7. Comparison of NBP\* algorithm and BP algorithm results at  $N = 100$

图 7.  $N = 100$  时 NBP\*算法和 BP 算法结果对比

$N = 60$ , NBP\*算法在策略影响下求解 50 个实例时, 在  $0.18 \leq p \leq 0.23$  处利用最后一次迭代信息策略找到解和未利用最后一次迭代信息找到解的情况如图 8 所示。

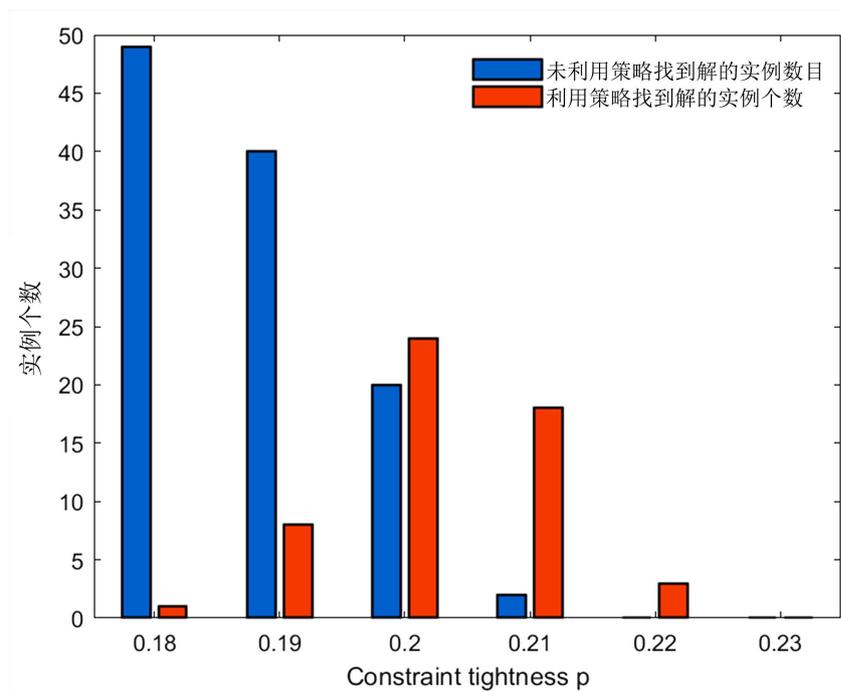


Figure 8. The situation of finding solutions under the influence of NBP\* algorithm strategy when  $N = 60$

图 8.  $N = 60$  时 NBP\*算法策略影响下找到解的情况

从图 8 可以看出, 当  $N=60$  时, 随着  $p$  的增大, 在找到解的实例中, NBP\* 算法利用最后一次迭代的结果找到解的实例与未利用最后一次信息找到解的比值逐渐增大,  $p=0.22$  时, BP 算法失效时, NBP\* 算法利用最后一次迭代结果仍能找到解, 这说明了在置信方程不收敛时, 最后一次迭代的结果其中有部分信息是准确的, 可以用来寻找实例的解。  $p \geq 0.23$  时, NBP\* 算法利用策略也无法找到解。

在  $p=0.19$  处, NBP\* 算法和 BP 算法在求解二元 RB 模型的随机实例的运行时间结果对如图 9 所示。

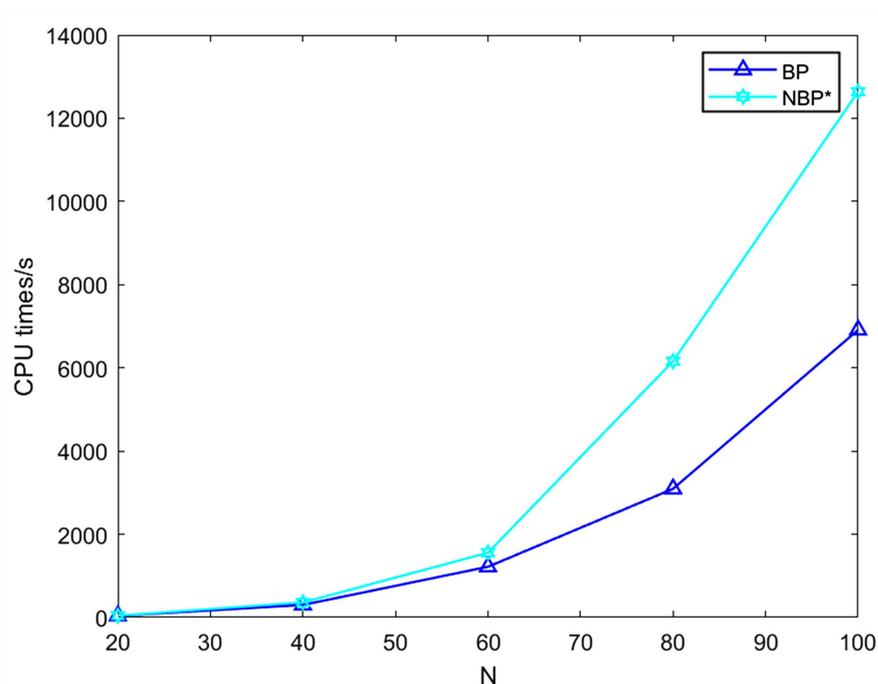


Figure 9. Comparison of runtime between NBP\* algorithm and BP algorithm when  $p=0.19$   
图 9.  $p=0.19$  时 NBP\* 算法和 BP 算法求解的运行时间对比

图 9 结果表明, 算法运行时间随问题规模的增加呈指数型增长。NBP\* 算法运行时间明显多于 BP 算法的运行时间。

## 5. 总结

本文基于置信传播算法, 在置信传播方程迭代不收敛时, 利用最后一次迭代信息的结果计算边际概率, 当赋值不满足约束时, 根据边际概率确定的变量顺序挑选下一个变量进行赋值, 保证算法继续进行。数值实验结果表明, 与 BP 算法相比, NBP\* 算法确实提高了随机实例的求解概率, 但增加了运行时间。将来的工作可以结合这一思路, 进一步优化算法的结果。

## 参考文献

- [1] 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>
- [2] 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)
- [3] Smith, B.M. and Dyer, M.E. (1996) Locating the Phase Transition in Binary Constraint Satisfaction Problems. *Artificial Intelligence*, **81**, 155-181. [https://doi.org/10.1016/0004-3702\(95\)00052-6](https://doi.org/10.1016/0004-3702(95)00052-6)
- [4] Achlioptas, D., Molloy, M.S.O., Lirousis, L.M., *et al.* (2001) Random Constraint Satisfaction: A More Accurate Pic-

- ture. *Constraints*, **6**, 329-344. <https://doi.org/10.1023/A:1011402324562>
- [5] 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>
- [6] Frieze, A.M. and Molloy, M. (2006) The Satisfiability Threshold for Randomly Generated Binary Constraint Satisfaction Problems. *Random Structure and Algorithms*, **28**, 323-339. <https://doi.org/10.1002/rsa.20118>
- [7] Smith, B.M. (2001) Constructing an Asymptotic Phase Transition in Random Binary Constraint Satisfaction Problems. *Theoretical Computer Science*, **265**, 265-283. [https://doi.org/10.1016/S0304-3975\(01\)00166-9](https://doi.org/10.1016/S0304-3975(01)00166-9)
- [8] Molloy, M. (2003) Models for Random Constraint Satisfaction Problems. *SIAM Journal of Computing*, **32**, 935-949. <https://doi.org/10.1137/S0097539700368667>
- [9] 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>
- [10] Sulc, P. and Zdeborová, L. (2010) Belief Propagation for Graph Portioning. *Journal of Physics A: Mathematical and Theoretical*, **43**, Article 285003. <https://doi.org/10.1088/1751-8113/43/28/285003>
- [11] Xu, K., Boussemart, F., Hemery, F., *et al.* (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>
- [12] Zhao, C.Y., Zhou, H.J., Zheng, Z.M., *et al.* (2011) A Message Passing Approach to Random Constraint Satisfaction Problems with Growing Domains. *Journal of Statistical Mechanics: Theory and Experiment*, **2011**, P02019. <https://doi.org/10.1088/1742-5468/2011/02/P02019>
- [13] 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.
- [14] 任雪亮. 改进的置信传播算法在求解最大约束满足问题的应用[D]: [硕士学位论文]. 长春: 东北师范大学, 2015.
- [15] 吴拨荣, 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.
- [16] Zhao, C.Y. and Fu, Y.R. (2021) Belief Propagation Guided Decimation Algorithms for Random Constraint Satisfaction Problems with Growing Domains. *Journal of Statistical Mechanics: Theory and Experiment*, **2021**, Article 033408. <https://doi.org/10.1088/1742-5468/abe6fe>
- [17] Zhao, C.Y., Fu, Y.R. and Zhao, J.H. (2022) A Residual-Based Message Passing Algorithm for Constraint Satisfaction Problems. *Communications in Theoretical Physics*, **74**, Article 035601. <https://doi.org/10.1088/1572-9494/ac4896>