Advances in Applied Mathematics 应用数学进展, 2023, 12(3), 993-1002
Published Online March 2023 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2023.123101

Hans汉斯

一种基于回溯求解约束满足问题的置信传播算法

林　童

上海理工大学，理学院，上海

收稿日期：2023年2月13日；录用日期：2023年3月8日；发布日期：2023年3月15日

摘　要

针对一个典型的具有可变取值域的随机约束满足问题模型即RB模型，提出一种基于回溯的置信传播的算法。该算法在置信传播方程不收敛时，通过回溯对上一个变量进行重新赋值，从而将消去变量的过程继续进行下去。数值结果表明：这种基于回溯的置信传播算法能在可满足性相变区域找到问题的解，有效地提高了置信传播算法的求解效率。

关键词

约束满足问题，RB模型，置信传播，回溯算法

A Belief Propagation Algorithm for Constraint Satisfaction Problem Based on Backtracking

Tong Lin

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

Received: Feb. 13th, 2023; accepted: Mar. 8th, 2023; published: Mar. 15th, 2023

Abstract

In this paper, we propose a belief propagation algorithm based on backtracking for a typical model of random constraint satisfaction problem with variable range, namely RB model. In this algorithm, when the belief propagation equation does not converge, it reassigns the previous variable by backtracking, so as to continue the process of variable elimination. The numerical results show

文章引用: 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.
DOI: 10.12677/aam.2023.123101

that the backtracking belief propagation algorithm can find the solution of the problem in the satisfiability phase transition region, and effectively improve the efficiency of the belief propagation algorithm.

Keywords

Constraint Satisfaction Problem, RB Model, Belief Propagation, Backtracking Algorithms

1. 引言

消息传递算法的迭代特性捕获了复杂系统中相互关联变量之间的复杂交互，并从迭代消息的不动点中提取信息，为处理优化、推理和学习问题中的困难计算任务提供了一个强大的工具包。约束满足问题(Constraint Satisfaction Problems, CSPs)是复杂系统研究中最典型的计算任务之一，CSPs 的一个基本计算问题是，在给定实例中找到满足所有约束的变量赋值的存在性和数量。在理论计算机科学的计算复杂性理论中，许多原型 NP-难问题可以被表述为 CSPs，因此 CSPs 被广泛研究和讨论。

许多学者开始利用消息传递算法求解约束满足问题，如利用消息传递算法中的置信传播(belief propagation, BP)算法求解约束满足问题，得到了很好的求解效率。由于 RB 模型是一个典型的具有可变取值域的随机约束满足问题模型，且具有精确相变，所以 RB 模型一经提出，就受到广泛关注，得到国内外学者的重视，并针对 RB 模型进行了大量的理论和实验研究。文献[1] [2]提出几种不同的置信传播(belief propagation, BP)算法求解 RB 模型产生的随机实例。文献[3]提出改进的置信传播算法，在 BP 方程中加入惩罚值，使得算法可以更有效的求解最大约束满足问题。文献[4]提出置信传播和模拟退火相结合的算法求解 RB 模型。相关文献[5]中，提出一种基于异步更新的置信传播算法求解 RB 模型，改变 BP 算法消息更新过程，极大的提高了算法的求解效率。文献[6]提出一种基于残差的消息传递算法，选择具有最大残差的边更新信息，算法提高了 BP 算法的收敛性和性能。

本文基于文献[1]中的算法 3，考虑置信传播方程不收敛时，为了使算法可以继续下去，提出基于回溯的置信传播算法(backtracking belief propagation, BBP)来提高算法的收敛性。在 BBP 算法中，若迭代收敛，则根据变量的边际概率分布选取具有最大边际概率的变量，将其值固定在边际概率最大的分量上，同时将该变量的各取值分量按照边际概率降序依次排序，以此作为该变量的取值顺序；若迭代不收敛后，则进行回溯，改变上一个变量的值，重新进行迭代。数值结果表明：与 BP 算法相比，BBP 算法通过回溯可以找到更多的有解实例，有效提高随机实例的求解概率。

2. RB 模型

过去几十年，针对 CSP 的研究，主要集中在 A，B，C，D 四种标准模型[7] [8] [9]上。后来 Achlioptas 等人发现随着问题规模的增大，很难找到一组赋值同时满足所有约束，即模型会表现出平凡渐近无解性[10]。为解决这一问题，学者们先后从多个角度提出各种不同改进模型[10] [11] [12] [13] [14]，其中包括 RB (Revised B)模型[11]，RB 模型对 B 模型约束数量和取值域大小进行了限制。文献[11]证明了 RB 模型存在精确的可满足性相变现象。而且 RB 模型在相变区域会产生大量的难解实例[15] [16]。

RB 模型的一个随机实例由变量集合 $X=\{x_1,x_2,\cdots,x_N\}$ 和约束集合 $C=\{C_1,C_2,\cdots,C_M\}$ 组成。每个变量都有对应的定义域 $D=\{s_1,s_2,\cdots,s_d\}$，其中 $|D|=d$，$d=N^{\alpha}$ ($\alpha>0$ 是常数)。每个约束 C_i 包含 $k(k\geq 2)$ 个不同的变量。对于每个约束 C_i，定义 Q_i 为 k 个变量的不相容赋值，即若这 k 个变量的取值属于集合 Q_i，则约束是不可满足的；否则约束就是可满足的。求解 RB 模型的一个实例，相当于找到实例中 N 个变量的一组赋值，使得所有约束同时被满足。生成 RB 模型的随机实例的步骤如下[11]：

a) 从可能的 C_N^k 个约束中随机、可重复地挑选 $M=rN\ln N$ 个约束构成约束集合 $\{C_i\}_{i=1}^{M}$，每个约束 C_i 都包含从 N 个变量中随机、不可重复挑选的 k 个变量；

b) 对每个约束 C_i，从 d^k 个可能赋值中随机、不可重复地选取 pd^k 个构成不协调赋值集合 Q_i，这里 $0<p<1$ 表示约束紧度。

文献[11]中严格证明了 RB 模型存在精确的可满足性相变现象。用 $\Pr(\mathrm{SAT})$ 表示 RB 模型的随机实例有解的概率，则有以下结论成立：

定理 1 [11] 令 $p_s=1-\mathrm{e}^{-\frac{\alpha}{r}}$，如果 $\alpha>\frac{1}{k},r>0$ 是两个常数，且 $k\mathrm{e}^{-\frac{\alpha}{r}}\geq 1$，则

$$\lim_{N\to\infty}\Pr(\mathrm{SAT})=\begin{cases}1, & p<p_s\\ 0, & p>p_s\end{cases} \tag{1}$$

定理表明 RB 模型在临界值 p_s 处的可满足概率发生突变，当 $N\to\infty$ 时，若 $p<p_s$，RB 模型的随机实例有解的概率趋近于 1；若 $p>p_s$，RB 模型的随机实例有解的概率趋近于 0。

3. 基于回溯的置信传播算法

3.1. RB 模型的因子图

因为 $k\geq 2$ 时 RB 模型都是 NP-完全问题，所以为了方便讨论，本文讨论 $k=2$ 时的 RB 模型。二元 RB 模型的部分因子图如图 1 所示，图中的圆圈表示变量结点，记作 $i,j,k,\cdots$，图中的方块表示约束结点，记作 $a,b,c,\cdots$，约束与约束中包含的变量的结点之间用线相连，表示一条边。

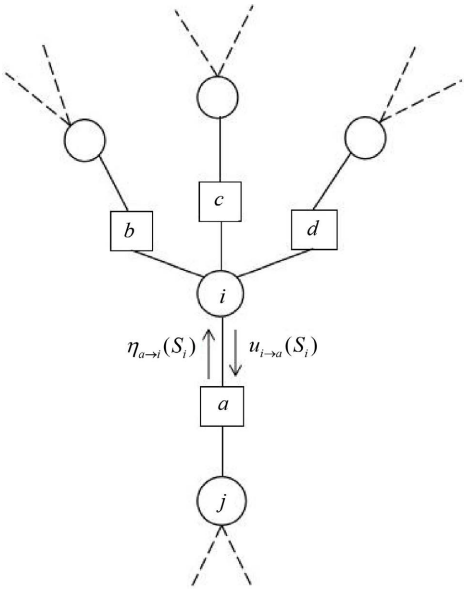

Figure 1. Partial factor graphs of random instances of binary RB model
图 1. 二元 RB 模型随机实例的部分因子图

3.2. BP 迭代方程

因子图中每条边(a,i)上定义了两种信息，分别是约束 a 发送给变量 i 的信息$\eta_{a\to i}(s_i)$和变量 i 发送给约束 a 的信息$u_{i\to a}(s_i)$，$\eta_{a\to i}(s_i)$表示约束 a 要求变量 i 取值为s_i的概率；$u_{i\to a}(s_i)$表示变量 i 在没有约束 a 的情况下取值为s_i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：

$$u_{i\to a}^{t}(s_i)=\frac{1}{Z^{i\to a}}\prod_{b\in V(i)\backslash a}\eta_{b\to i}^{t}(s_i) \tag{2}$$

$$\eta_{a\to i}^{t+1}(s_i)=\frac{1}{Z^{a\to i}}\sum_{s_j\in D,\, j\in V(a)\backslash i}\delta(s_i,s_j)u_{j\to a}^{t}(s_j) \tag{3}$$

其中

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

这里$Z^{i\to a}$和$Z^{a\to i}$是归一化因子，$V(i)$表示与变量 i 相关的所有约束，$V(i)\backslash a$表示与变量 i 相关的约束除去约束 a；$V(a)$表示约束 a 中包含的所有变量，$V(a)\backslash i$表示约束 a 包含的变量除去变量 i。

3.3. BBP 算法

事实上，文献[1]中提出的算法，在接近相变点会出现 BP 方程不收敛的情况，导致算法失效，为了提高算法地求解效率，在算法中加入回溯算法，得到 BBP 算法。当 BP 算法不收敛时，通过回溯增加算法收敛的可能性，最后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，每一步只将一个具有最大边际概率的变量的值固定到其概率最大的分量上，因此在算法执行过程中，将变量分为三种类型，如图 2 所示：

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；

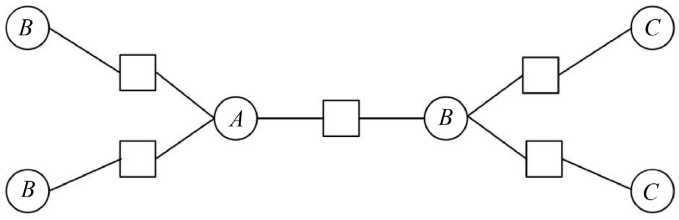

Figure 2. Type of variable during algorithm execution
图 2. 算法执行过程中的变量类型

BBP 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$t_{\max}$，精度ε和最大回溯次数T_m。

输出：实例的解或者算法不收敛。

Step 1： $n=1$，$T=0$，运行子程序 BP 算法得到$\eta_{a\to i}^{*}(s_i)$；

Step 2：计算每个变量 i 的取值的边际概率

$$\eta_i(s_i)=\frac{\prod_{a\in\partial i}\eta^*_{a\to i}(s_i)}{\sum_{s_i\in D}\prod_{a\in\partial i}\eta^*_{a\to i}(s_i)} \tag{5}$$

Step 3：挑选所有 $\eta_i(s_i)$ 中的最大值，将其所对应的变量 i 作为第一个待赋值的变量 x_1，将 x_1 的值固定在分量 s_i 上，并将 x_1 的所有取值按边际概率降序排列，得到该变量的一个取值顺序 V_{x_1}。

Step 4：$n=n+1$，$t=0$，初始化每条边上约束发送给变量的信息 $\eta^{(0)}_{a\to i}(s_i)$：

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 $\eta^{(0)}_{a\to i}(s_i)=0$；否则，$\eta^{(0)}_{a\to i}(s_i)=1$；

对于 C 类型的变量：随机初始化 $\eta^{(0)}_{a\to i}(s_i)$；

Step 5：$t=t+1$

对于 A 类型的变量：跳过；

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

对于 C 类型的变量：将 $\eta^{(t-1)}_{a\to i}(s_i)$ 代入公式(2)计算得到 $u^{(t-1)}_{i\to a}(s_i)$（如果 $V(i)\backslash a=\phi$，则令 $u^{(t-1)}_{i\to a}(s_i)=1/d$），然后将 $u^{(t-1)}_{i\to a}(s_i)$ 代入公式(3)更新得到 $\eta^{(t)}_{a\to i}(s_i)$；

Step 6：如果 $\left|\eta^{(t)}_{a\to i}(s_i)-\eta^{(t-1)}_{a\to i}(s_i)\right|<\varepsilon$，则令 $\eta^*_{a\to i}(s_i)=\eta^{(t)}_{a\to i}(s_i)$，并执行 **Step 8**；否则，执行 **Step 5**；

Step 7：如果 $t=t_{\max}$，则 BP 方程不收敛，令 $n=n-1$，执行 **Step 10**；

Step 8：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；

Step 9：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 个待赋值的变量 x_n，并将 x_n 的值固定在分量 s_i 上。同时将 x_n 的所有取值按边际概率降序排列，得到该变量的一个取值顺序 V_{x_n}。检查此次赋值是否满足所有约束，如果不满足约束则输出未找到实例的解；

Step 10：$T=T+1$，如果 $T>T_m$，则输出没找到实例的解；否则，执行 **Step 11**；

Step 11：根据变量 x_n 的取值顺序 V_{x_n}，如果 V_{x_n} 中还有值未被取过，则将 V_{x_n} 中当前取值的后一位赋给 x_n，并检查此次赋值是否满足所有约束，如果该不满足约束则执行 **Step 10**；如果 V_{x_n} 中所有值都被取过，则清除该变量的取值，且 $n=n-1$，如果 $n<1$，则输出没找到实例的解，如果 $n\geq 1$，则执行 **Step 10**；

Step 12：如果 $n<N$，执行 **Step 7**；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。

子程序 BP 算法如下：

Step 1：$t=0$，随机初始化每条边上约束发送给变量的信息 $\eta^{(0)}_{a\to i}(s_i)$；

Step 2：$t=t+1$，将 $\eta^{(t-1)}_{a\to i}(s_i)$ 代入公式(2)计算得到 $u^{(t-1)}_{i\to a}(s_i)$（如果 $V(i)\backslash a=\phi$，则令 $u^{(t-1)}_{i\to a}(s_i)=1/d$），然后将 $u^{(t-1)}_{i\to a}(s_i)$ 代入公式(3)更新得到 $\eta^{(t)}_{a\to i}(s_i)$；

Step 3：如果 $\left|\eta^{(t)}_{a\to i}(s_i)-\eta^{(t-1)}_{a\to i}(s_i)\right|<\varepsilon$，则令 $\eta^*_{a\to i}(s_i)=\eta^{(t)}_{a\to i}(s_i)$，输出 $\eta^*_{a\to i}(s_i)$；否则，执行 **Step 2**；

Step 4：如果 $t=t_{\max}$，则输出不收敛。

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 100 个二元 RB 模型的随机实例。对不同的变量数 N，变量的定义域规模 d 和约束的数目 M 如表 1 所示。在算法中，取最大迭代次数 $t_{\max}=10^3$，精度 $\varepsilon=10^{-4}$。

Table 1. Parameter values of binary RB model under different number of variables
表 1. 二元 RB 模型在不同变量数 N 下的参数值

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
100	0.8	3	40	1382

4.1. 算法结果

在算法中，取最大回溯次数 $T_m = 500$。在 100 个随机实例上运行 BBP 算法，算法求解概率如图 3 所示。

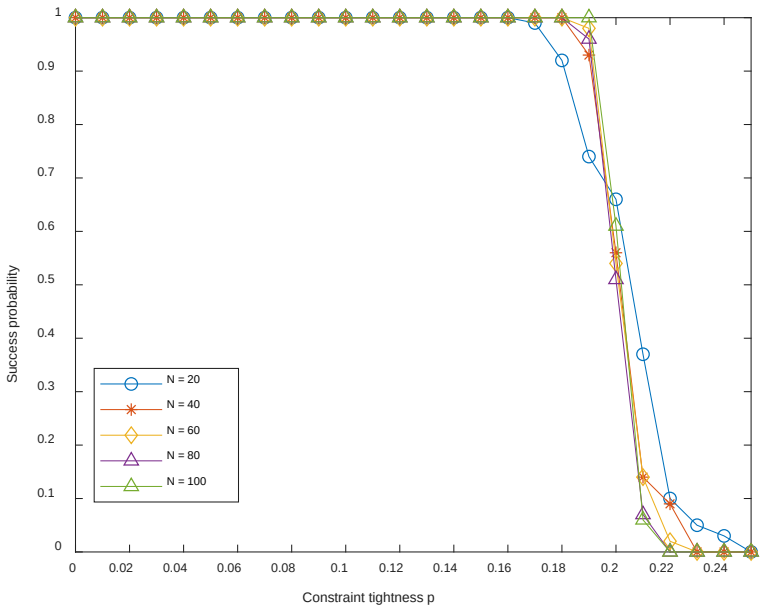

Figure 3. The result of solving 100 random instances by BBP algorithm
图 3. BBP 算法求解 100 个随机实例的结果

从图 3 可以看出，在 $N = 20$ 时，BBP 算法在 $p \le 0.16$ 时找到解的概率是 100%，当 $p \ge 0.25$ 时算法找不到任何解；在 $N = 100$ 时，BBP 算法在 $p \le 0.19$ 时找到解的概率是 100%，当 $p \ge 0.22$ 时算法找不到任何解。

4.2. 算法分析

$N = \{20, 40, 60\}$ 时，BBP 算法和 BP 算法的求解效率对比分别如图 4~6 所示。

从图中可以看出，与 BP 算法相比，BBP 算法可以明显提高对实例的求解概率，尤其是在区域 $0.18 < p < 0.22$ 内，BBP 算法可以通过回溯找到更多的有解实例。

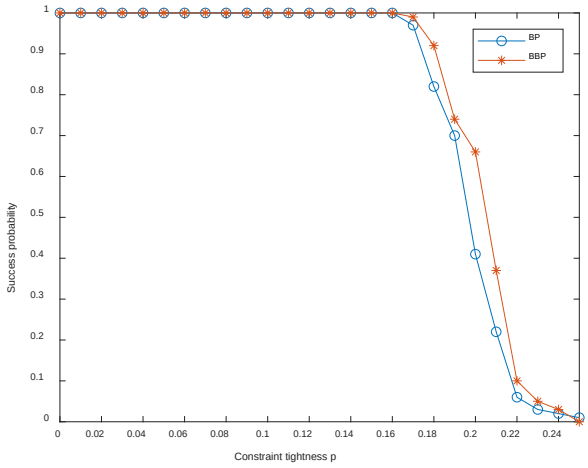

Figure 4. Comparison of the results of the two algorithms for $N = 20$
图 4. $N = 20$ 时两种算法的结果对比

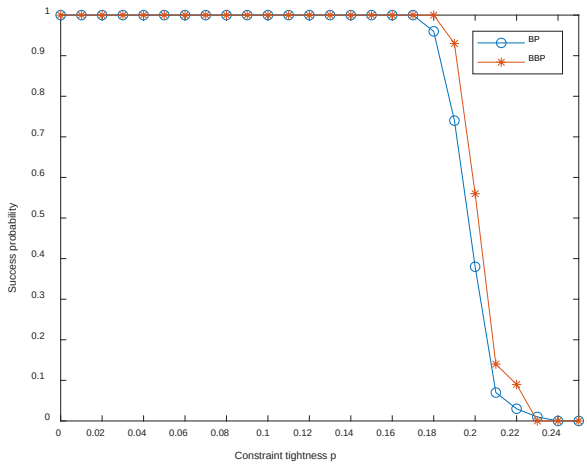

Figure 5. Comparison of the results of the two algorithms for $N = 40$
图 5. $N = 40$ 时两种算法的结果对比

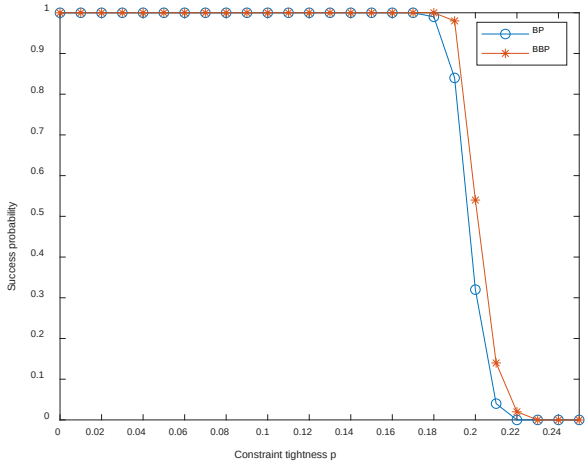

Figure 6. Comparison of the results of the two algorithms for $N = 60$
图 6. $N = 60$ 时两种算法的结果对比

$N=\{20,40,60\}$ 时，BBP 算法求解 100 个实例，未发生回溯找到解和发生回溯找到解的情况如图 7~9 所示。从图中可以看出，对于 $N=\{20,40,60\}$，在区域 $0.17\le p\le 0.23$ 内，BBP 算法通过回溯有效提高算法的求解效率，当 $p>0.23$ 时，即使发生回溯也无法找到实例的解。值得注意的是，随着 p 的增大，发生回溯找到解的实例的数目呈现先增加再减少的趋势，在 $p=0.2$ 时，发生回溯找到解的实例的数目达到最大。

接下来，我们比较 BBP 算法和 BP 算法的收敛性。取 $N=20$，BBP 算法和 BP 算法求解 100 个实例时收敛的实例数对比如图 10 所示。图 10 表明 BBP 算法的收敛性优于 BP 算法的收敛性，说明通过回溯可以提升算法的收敛性，从而提高算法的求解效率。

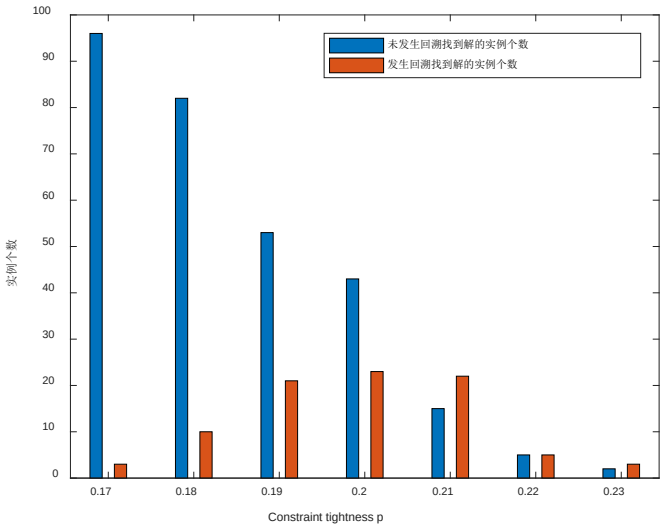

Figure 7. The backtracking when the BBP algorithm finds the solution for $N=20$

图 7. $N=20$ 时 BBP 算法找到解的回溯情况

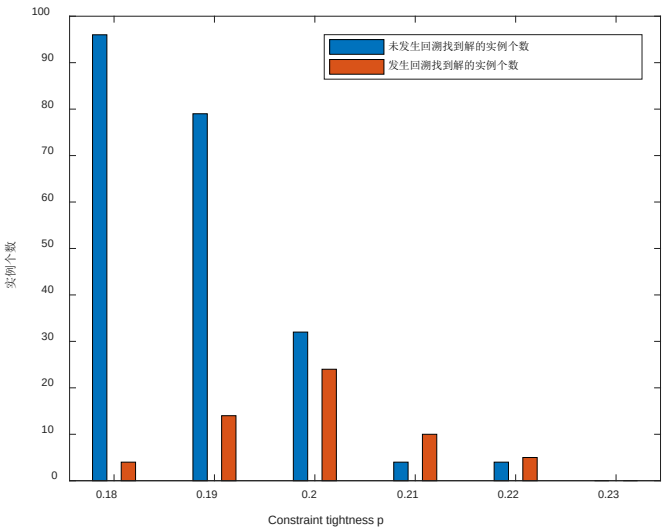

Figure 8. The backtracking when the BBP algorithm finds the solution for $N=40$

图 8. $N=40$ 时 BBP 算法找到解的回溯情况

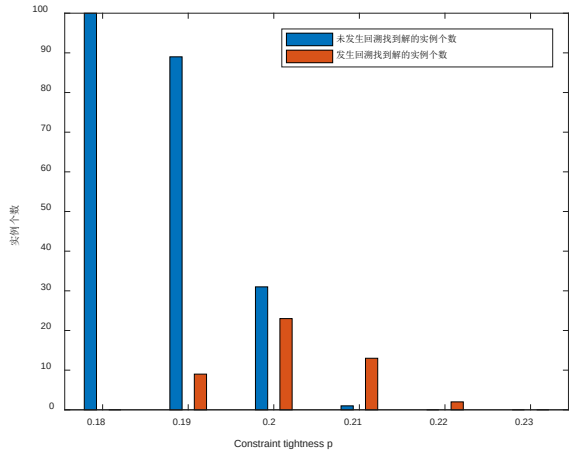

Figure 9. The backtracking when the BBP algorithm finds the solution for $N = 60$
图 9. $N = 60$ 时 BBP 算法找到解的回溯情况

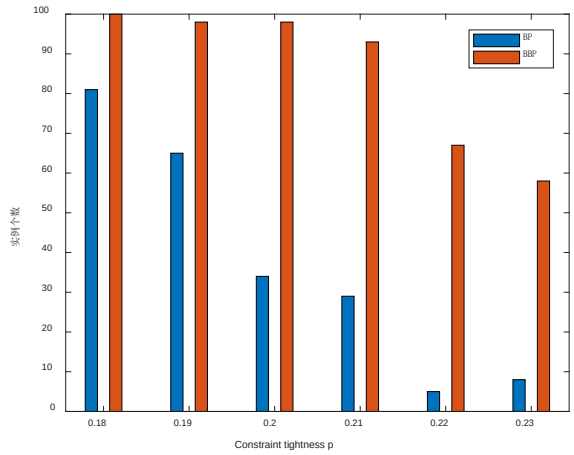

Figure 10. Comparison of the number of convergence cases between BP algorithm and BBP algorithm for $N = 20$
图 10. $N = 20$ 时 BP 算法和 BBP 算法收敛的实例数目对比

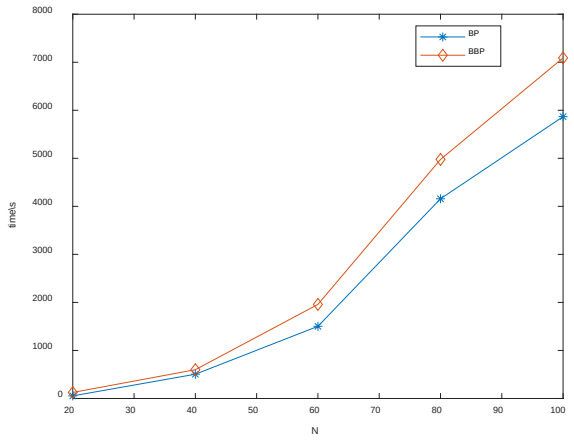

Figure 11. The running time of the two algorithms for p = 0.19
图 11. $p = 0.19$ 时两种算法求解的运行时间

在 $p = 0.19$ 处 BBP 算法和 BP 算法求解 RB 模型的随机实例的运行时间对比如图 11 所示。

从图 11 可以看出，算法运行时间随着问题规模 N 的增加呈指数形式增长。BBP 算法因为回溯策略的加入，算法的运行时间也明显增加，而且随着问题规模 N 的增加，BBP 算法较 BP 算法增加的运行时间越来越多。

5. 总结

本文提出基于回溯的置信传播算法求解大值域的随机约束满足问题。算法在 BP 迭代方程不收敛时，通过回溯改变上一个变量的值，并重新进行迭代，保证消去变量的过程继续进行。数值结果表明，BBP 算法可以提高 BP 算法的收敛性，与 BP 算法相比可以有效提高算法的求解效率。因为回溯的加入，算法的时间复杂度会增加，所以在未来的工作中，可以优化算法，减少算法的时间复杂度。

参考文献

[1] Zhou, H.J., Zheng, Z.M. and Xu, K. (2011) A Message-Passing Approach to Random Constraint Satisfaction Problems with Growing Domains. *Journal of Statistical Mechanics*: *Theory and Experiment*, **2011**, Article ID: P02019. https://doi.org/10.1088/1742-5468/2011/02/P02019

[2] 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.

[3] 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.

[4] 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.

[5] 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 ID: 033408. https://doi.org/10.1088/1742-5468/abe6fe

[6] 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 ID: 035601. https://doi.org/10.1088/1572-9494/ac4896

[7] Gent, I.P., Macintyre, E., Prosser, P., Smith, B.M. and Walsh, T. (2001) Random Constraint Satisfaction: Flaws and Structure. *Constraints*, **6**, 345-372. https://doi.org/10.1023/A:1011454308633

[8] 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

[9] 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

[10] 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

[11] 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

[12] Frieze, A. 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

[13] 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

[14] Molloy, M. (2003) Models for Random Constraint Satisfaction Problems. *SIAM Journal of Computing*, **32**, 935-949. https://doi.org/10.1137/S0097539700368667

[15] 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

[16] 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