随机约束满足问题的回溯算法分析*

许可，李未

(北京航空航天大学计算机科学与工程系，北京 100083)

E-mail: kexu@nlsde.buaa.edu.cn; ke.xu@263.net

http://kexu.home.chinaren.net/

摘要：提出一种新的随机CSP(constraint satisfaction problem)模型，并且通过研究搜索树的平均节点数，分析了回溯算法求解该模型的平均复杂性.结果表明，这种模型能够生成难解的CSP实例，找到所有的解或证明无解所需的平均节点数即随变量数的增加而指数增长.因此，该模型可以用来研究难解实例的性质和CSP算法的性能等问题，从而有助于设计出更为高效的算法.

关键词：算法分析；平均复杂性；回溯算法；约束满足

中图法分类号：TP18 **文献标识码**：A

一个约束满足问题(constraint satisfaction problem，简称CSP)由一个变量的有限集$\{u_1,\ldots,u_n\}$和一个约束集合组成.每个变量都定义了相应的值域，变量u_i的赋值只能从其值域D_i的元素中选取.对于$2\leqslant k\leqslant n$，一个k元约束$C_{i1,i2,\ldots,ik}$包含了变量集的一个子集$\{u_{i1},u_{i2},\ldots,u_{ik}\}$和一个约束关系$R_{i1,i2,\ldots,ik}\subseteq D_{i1}\times\ldots\times D_{ik}$(其中$i1$，$i2$，...，$ik$各不相同)，约束关系$R_{i1,i2,\ldots,ik}$限制了变量$u_{i1}$，...，$u_{ik}$的取值.给定一个约束$C_{i1,i2,\ldots,ik}$，一组赋值是协调的，当且仅当这组赋值属于约束关系$R_{i1,i2,\ldots,ik}$，否则就是不协调的.一组关于$u_{i1}$，...，$u_{ik}$的赋值满足约束$C_{i1,i2,\ldots,ik}$当且仅当该组赋值是协调的.一组关于所有变量的赋值是CSP的一个解，当且仅当这组赋值使得该CSP的所有约束均被满足.有时候，人们希望判断一个给定的CSP是否有解，本文研究的是找到CSP实例所有的解或证明其无解的平均复杂性.

CSP是人工智能中的一个基本问题，它在模式识别、语言理解、规划和诊断等领域都有非常广泛的应用[1].在一般情况下，CSP是一个NP完全问题.近年来，通过大量的实验研究人们发现，在CSP和SAT等随机组合问题中，只要选取适当的控制参数(通常是约束的个数与变量数之比)，就存在某个临界值，使得当控制参数小于该临界值时，实例有解的概率随变量数的增大而趋近于1，而当控制参数大于该临界值时，则有解的概率趋近于0.同时，随着控制参数的增加，求解实例的难度也发生了从易到难再到容易的变化，难度曲线的尖峰正好位于有解概率突然转变的区域内.人们把这种现象称为相变现象[2].自从发现相变现象以来，随机CSP就一直是研究的热点之一，人们在算法设计、模型选择和算法测试等方面进行了大量而深入的研究[3~8].但是，到目前为止，有关CSP算法理论分析的文章还非常少见.最近，Kondrak[5]已基本完成了CSP算法的最坏复杂性研究.但是，最坏复杂性通常不能反映算法的平均性能.为了解算法在通常情况下的表现，就必须研究其平均复杂性.1997年，Purdom[7]首先讨论了回溯算法求解一种随机CSP模型的平均复杂性.该模型的每个约束包含k个不同的变量，对于一个给定的约束，每组赋值以概率p成为协调赋值.分析表明，当约束个数与变量数之比较小时，回溯算法求解该模型所需的时间与变量数呈指数关系；而当约束个数与变量数之比较大时，则只需要多项式的时间.在模型B[3]的基础上，本文提出了一种新的随机CSP模型，并分析了该模型在回溯算法下的平均复杂性.结果表明，这种模型能够生成难解的CSP实例，回溯算法求解该模型具有指数型的平均复杂性.因此，这种模型可以帮助

* 收稿日期：1999-04-09；修改日期：1999-09-15

基金项目：国家重点基础研究发展规划资助项目(G1999032701)；教育部博士点基金资助项目(1999000613)

作者简介：许可(1971—)，男，四川渠县人，博士，主要研究领域为算法设计与分析，人工智能；李未(1943—)，男，北京人，教授，博士生导师，中国科学院院士，主要研究领域为并发程序设计语言，操作语义，类型理论，人工智能.

我们更好地研究难解实例和算法性能等问题.

1 随机 CSP 模型和回溯算法

在实验研究中,通常有若干种方法用来产生随机 CSP 实例.本文所用的随机 CSP 模型与随机图中常见的 $G_{n,m}$ 模型类似,其具体过程如下:

步骤 1. 随机、均匀、可重复地选取 $t=rn$ 个约束,每个约束恰好包含从 n 个变量中选取的 k 个不同的变量.其中 r 表示约束个数与变量数之比.

步骤 2. 对于每个给定的约束,均匀而不重复地选取 $p\cdot d^k$ 组的赋值为不协调赋值.换句话说,每个约束关系恰好包含 $(1-p)\cdot d^k$ 组赋值.其中 p 和 d 分别表示约束强度和变量值域的大小.

在本文中,假定所有变量都有相同大小的值域 $d\geqslant 2$,并且约束强度 $0<p<1/d^{k-1}$.为了论述方便,我们将以上的随机 CSP 模型简称为模型 G.值得注意的是,当前 SAT 研究中应用非常广泛的随机 k-SAT 只是模型 G 的一个特例(分别令 $d=2$ 和 $p=1/2^k$).因为随机 k-SAT 能够生成难解的实例,所以人们在实验研究中经常采用它来测试算法的性能.模型 G 包含了比随机 k-SAT 更丰富的特征.通过对该模型进行研究,可以帮助我们进一步了解随机 CSP 的性质,并加深对难解实例的认识.回溯算法是一种求解 CSP 的基本算法,同时也是许多搜索算法的基础.本文分析所用的回溯算法[6]如下:

(1) Set $i\leftarrow 0$.

(2) If $F(u_1/a_1,\ldots,u_i/a_i,\ldots,u_i/a_i)$ is false, go to (6)($F(u_i/a_i)$表示将 CSP 实例 F 中的变量 u_i 赋值为 a_i,$a_i\in D_i$,$i=1,2,\ldots,i$,$F(u_1/a_1,\ldots,u_i/a_i,\ldots,u_i/a_i)$ is false 表示在当前赋值下,至少有一个子句不可满足).

(3) Set $i\leftarrow i+1$.

(4) If $i>n$, then $a_1,a_2,\ldots,a_n$ is a solution. Go to (7).

(5) Set $u_i\leftarrow$ the first value of u_i and go to (2).

(6) If u_i has more values, set $u_i\leftarrow$ the next value for u_i and go to (2).

(7) Set $i\leftarrow i-1$. If $i>0$, go to (6); otherwise stop.

2 搜索树的平均节点数

为了研究搜索树的平均节点数,我们首先来研究在搜索树的第 i 层(根节点为第 0 层,第 i 层已有 i 个变量被赋值),一个随机约束可满足的概率,记为 $g(i)$.

当 $i\leqslant k-1$ 时,由于每个约束都有 k 个变量,因此,至少有一个变量未被赋值.此时,该约束的 k 个变量至少有 d 组可能的赋值,注意,有 $p<1/d^{k-1}$,即不协调赋值的总数小于 d,因此,至少有一组赋值满足该约束,故可得

$$g(i)=1. \tag{1}$$

当 $i\geqslant k$ 时,一个随机约束中 k 个变量全被赋值的概率等于 C_i^k/C_n^k.此时,这组赋值满足约束的概率等于 $(1-p)$;否则,约束至少有一个变量未被赋值,与前面的分析相同,约束可满足的概率等于 1.因此有

$$g(i)=\frac{C_i^k}{C_n^k}(1-p)+\left(1-\frac{C_i^k}{C_n^k}\right)=1-p\frac{C_i^k}{C_n^k}=1-p\frac{i(i-1)\ldots(i-k+1)}{n(n-1)\ldots(n-k+1)}. \tag{2}$$

显然,表达式(1)和表达式(2)可以统一为一个表达式,即

$$g(i)=1-p\frac{i(i-1)\ldots(i-k+1)}{n(n-1)\ldots(n-k+1)},\quad 0\leqslant i\leqslant n-1. \tag{3}$$

由于模型 G 的每一个约束都是随机独立产生的,因此,$t=rn$ 个约束均可满足的概率等于 $[g(i)]^m$.这也是第 i 层的一个节点扩展到第 $i+1$ 层的概率,每一个节点的扩展将生成 d 个附加的节点,第 i 层总共可能有 d^i 个节点.因此,当采用回溯算法求解模型 G 时,找到实例所有的解或证明其无解所需的平均节点数为

$$T_{av}=1+d\sum_{i=0}^{i=n-1}d^i[g(i)]^m. \tag{4}$$

在研究平均节点数的变化规律之前,我们首先来分析$g(i)$的性质.易证:$\frac{i-l}{n-l}\leqslant\frac{i}{n}$,其中$0\leqslant l\leqslant k-1$.因此有

$$\frac{i(i-1)\cdots(i-k+1)}{n(n-1)\cdots(n-k+1)}\leqslant\left(\frac{i}{n}\right)^k. \tag{5}$$

将式(5)代入$g(i)$的表达式可得

$$g(i)=1-p\frac{i(i-1)\cdots(i-k+1)}{n(n-1)\cdots(n-k-1)}\geqslant 1-p\left(\frac{i}{n}\right)^k. \tag{6}$$

将式(6)代入式(4)得

$$T_{av}=1+d\sum_{i=0}^{i=n-1}d^i[g(i)]^r=1+d\sum_{i=0}^{i=n-1}e^{i\ln d}e^{r\cdot n\cdot g(i)}\geqslant 1+d\sum_{i=0}^{i=n-1}e^{n\left\{\frac{i}{n}\ln d+r\ln\left[1-p\left(\frac{i}{n}\right)^k\right]\right\}}. \tag{7}$$

令$x=i/n$,$f(x)=x\ln d+r\ln(1-px^k)$,其中$0\leqslant x\leqslant 1$,则式(7)可写为

$$T_{av}\geqslant 1+d\sum_{x=0}^{x=\frac{n-1}{n}}e^{rf(x)}. \tag{8}$$

以上的不等式给出了平均节点数的一个下界,不等式右端的性质主要由指数项确定.为了得到平均节点数的变化规律,我们首先来研究函数$f(x)$的性质.

引理 1. 给定r,若$r>r_0=\frac{(1-p)\ln d}{pk}$,则函数$f(x)$在区间$[0,1]$上有唯一的最大值点$0<\zeta<1$;若$r\leqslant r_0$,则$f(x)$在端点$x=1$处取最大值.

证明:给定r,为了求出函数$f(x)$的最大值,首先来分析$f(x)$的导函数:

$$f'(x)=\ln d-\frac{rpkx^{k-1}}{1-px^k},\quad f''(x)=-rpk\frac{(k-1)x^{k-2}+px^{2k-2}}{(1-px^k)^2}, \tag{9}$$

$$f'(0)=\ln d,\quad f'(1)=\frac{pk}{1-p}\left[\frac{(1-p)\ln d}{pk}-r\right]. \tag{10}$$

若$r>r_0$,则由式(9)、式(10)易得$f'(0)>0$,$f'(1)<0$,在区间$(0,1)$上有$f''(x)<0$,因此,$f'(x)$是一个单调减函数.根据介值定理可知,存在唯一的点$0<\zeta<1$使得$f'(\zeta)=0$,注意:因为有$f''(\zeta)<0$,故ζ是$f(x)$唯一的最大值点.

若$r\leqslant r_0$,则$f'(1)\geqslant 0$,由于$f'(x)$是一个单调减函数,因此,在区间$[0,1)$上有$f'(x)>0$,即$f(x)$是一个单调增函数,故$f(x)$在端点$x=1$处取最大值. □

引理 2. 给定r,当$r>r_0$时,则函数$f(x)$在区间$[0,1]$上的最大值$f(\zeta)>0$.

证明:由引理 1,给定$r>r_0$,函数$f(x)$有唯一的最大值点ζ.定义一个以r为自变量,最大值点ζ为因变量的函数$\zeta(r)$.依定义,$\zeta(r)$满足如下方程:

$$f'(\zeta(r))=0\Rightarrow\ln d-\frac{rpk\zeta^{k-1}(r)}{1-p\zeta^k(r)}=0. \tag{11}$$

若$r>r_0$,则有

$$f(\zeta(r))=\zeta(r)\ln d+r\ln[1-p\zeta^k(r)]. \tag{12}$$

从式(11)中解出r并代入式(12)得

$$f(\zeta))=\frac{\ln d}{kp\zeta^{k-1}(r)}\{kp\zeta^k(r)+[1-p\zeta^k(r)]\ln[1-p\zeta^k(r)]\}. \tag{13}$$

令$y=p\zeta^k(r)$,$H(y)=ky+(1-y)\ln(1-y)$,则

$$F(r)=\frac{2\ln 2}{kp\zeta^{k-1}(r)}H(y),\quad 0<y\leqslant p. \tag{14}$$

显然,$H(0)=0$,$H'(y)=k-1-\ln(1-y)>0$.因此,$H(y)>0$,故$f(\zeta(r))>0$. □

3 平均复杂性的分析结果

定理 1. 给定r,平均节点数T_{av}随变量数n的增加而呈指数增长.

证明:分以下两种情况来加以证明:

情况 1. 若 $r>r_0$,由引理 2,函数 $f(x)$有唯一的最大值点 ζ,并且 $f(\zeta)>0$. 显然,$f(x)$是一个连续函数,取 δ 充分小,使得当 $0<\zeta-\delta<x<\zeta+\delta<1$ 时,有

$$f(x)>\frac{f(\zeta)}{2}>0. \tag{15}$$

当 n 充分大以后,我们总可以找到正整数 i_M,使其满足 $\zeta-\delta<\frac{i_M}{n}<\zeta+\delta$,因此有 $f\left(\frac{i_M}{n}\right)>\frac{f(\zeta)}{2}>0$. 于是,

$$T_{av}=1+d\sum_{x=0}^{x=\frac{n-1}{n}}e^{nf(x)}\geqslant 1+de^{nf(i_M/n)}>1+de^{n\frac{f(\zeta)}{2}}. \tag{16}$$

情况 2. 若 $r\leqslant r_0$,$f(x)$在端点 $x=1$ 处取最大值. 首先分析函数 $g(n-1)$的性质.

$$g(n-1)=1-p\frac{n-1}{n}\cdot\frac{n-2}{n-1}\cdot\ldots\cdot\frac{n-k}{n-k+1}=1-p+\frac{pk}{n}>1-p. \tag{17}$$

因此有

$$T_{av}=1+d\sum_{i=0}^{i=n-1}d^i[g(i)]^{rn}\geqslant 1+d\cdot d^{n-1}[g(n-1)]^{rn}\geqslant 1+e^{n[\ln d+r\ln(1-p)]}. \tag{18}$$

下面,我们来证明 $\ln d+r\ln(1-p)>0$. 当 $r\leqslant r_0$ 时,有

$$\ln d+r\ln(1-p)\geqslant\ln d+r_0\ln(1-p)=\left[1-\frac{1}{k}\frac{(1-p)}{p}\ln\left(1+\frac{p}{1-p}\right)\right]\ln d. \tag{19}$$

易证:$\frac{1-p}{p}\ln\left(1+\frac{p}{1-p}\right)<1$,因此,$\ln d+r\ln(1-p)>\left(1-\frac{1}{k}\right)\ln d>0$.

综合以上两种情况,平均节点数随变量数的增加至少将以指数的速度增长. 从另一方面来看,即使在最坏情况下,搜索树的平均节点数也不会超过搜索树可能有的全部节点数,即 $T_{av}\leqslant 1+d+d^2+\ldots+d^n=(d^{n+1}-d)/(d-1)$. 因此,平均节点数随变量数的增加而呈指数增长. 故定理 1 得证. □

4 结 论

定理 1 表明,当采用回溯算法求解本文所提出的随机 CSP 模型时,找到实例所有的解或证明其无解需要指数型的平均时间. 也就是说,这种模型生成的实例在回溯算法下通常具有难解的性质. 因此,该模型可以用来研究难解实例的特征和 CSP 算法的性能等问题,从而帮助我们设计出更为高效的算法.

致谢 在本文的写作过程中,曾与 Patras 大学的 Y. C. Stamatiou 和 L. M. Kirousis 进行过有启发性的讨论,特致谢意.

References:

[1] Dechter, R. Constraint satisfaction. In: The MIT Encyclopedia of the Cognitive Sciences (MITECS). 1998. ftp://ftp.ics.uci.edu/pub/CSP-repository/papers/R68.ps.

[2] Borow, D. G., Brady, M. Special volume on frontiers in problem solving: phase transitions and complexity. Artificial Intelligence, 1996,81(1~2):1~15.

[3] Smith, B. M., Dyer, M. E. Locating the phase transition in binary constraint satisfaction problems. Artificial Intelligence, 1996,81(1~2):155~181.

[4] Frost, D. Algorithms and heuristics for constraint satisfaction problems[Ph. D. Thesis]. University of California, Irvine, 1997. ftp://ftp.ics.uci.edu/pub/CSP repository/papers/R69.ps.

[5] Kondrak, G., Beek, P. V. A theoretical valuation of selected algorithms. Artificial Intelligence, 1997,89(1~2):365~387.

[6] Prosser, P. An empirical study of phase transitions in binary constraint satisfaction problems. Artificial Intelligence, 1996,81(1~2):81~109.

[7] Purdom, P. W. Backtracking and random constraint satisfaction. Annals of Mathematics and Artificial Intelligence, 1997,20(1~4):393~410.

[8] Purdom, P. W., Brown, C. A. An average time analysis of backtracking. SIAM Journal on Computing, 1981,10(3):583~593.

An Average Time Analysis of Backtracking on Random Constraint Satisfaction Problems

XU Ke, LI Wei

(*Department of Computer Science and Engineering, Beijing University of Aeronautics and Astronautics, Beijing* 100083, *China*)

E-mail: kexu@nlsde.buaa.edu.cn; ke.xu@263.net

http://kexu.home.chinaren.net/

Received April 9, 1999; accepted September 15, 1999

Abstract: A new random CSP (constraint satisfaction problem) model is proposed in this paper. By analyzing the expected number of nodes in a search tree, the average running time used by the backtracking algorithm on random constraint satisfaction problems is studied. The results show that the model can generate hard CSP instances, and the expected number of nodes required for finding all solutions or proving that no solution exists becomes exponentially large as the number of variables grows. Therefore, the model can be used to analyze the nature of hard instances and evaluate the performance of CSP algorithms, and hence it helps the researchers to design more efficient algorithms.

Key words: analysis of algorithm; average complexity; backtracking algorithm; constraint satisfaction