COMP9414: 人工智能讲座3a: 约束满足

韦恩-沃布克
电邮：w.wobcke@unsw.edu.au

本讲座

- 约束满足问题(CSP)
- 标准搜索方法
  - 逆向搜索和启发式方法
  - 正向检查和电弧一致性

约束满足问题

- 约束满足问题是由一组变量X_i，每个变量都有一个可能值的域D_i，以及一组约束条件C所定义的。
- 目的是为每个变量X_i（从域D_i）找到一个分配，使所有的约束条件C得到满足。

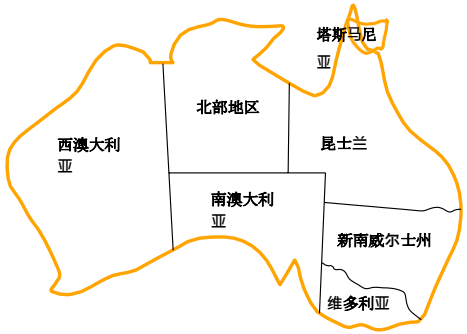
例子。地图 着色

- 域拆分和弧形一致性
- 消灭变量
- **本地搜索**
  - 爬坡
  - 模拟退火

变量。Wa, nt, q, nsw, v, sa, t

领域。D_i = {红、绿、蓝}

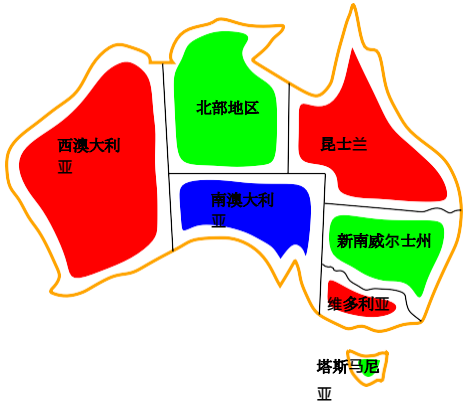
限制条件。相邻的区域有不同的颜色（WA /= NT，等等）。



新南威尔士大学
©W.Wobcke等人，2019-2022年

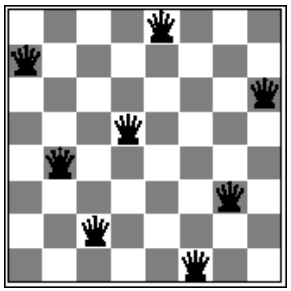
例子。地图涂色

- 解决方案是一个满足所有约束条件的任务



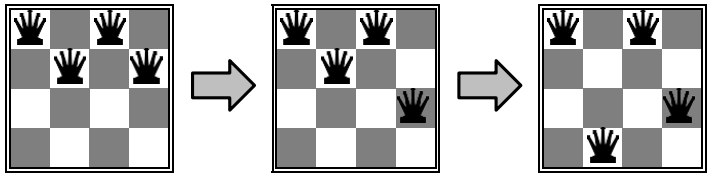
{WA=红色, NT=绿色, Q=红色, NSW=绿色, V=红色, SA=蓝色, T=绿色}。

例如：n-Queens拼图



n-作为CSP的皇后区之谜

假设每一列有一个皇后。域是皇后在一列中的可能位置。赋值是指每个域有一个元素。



变量。Q_1，Q_2，Q_3，Q_4
域。D_i = {1, 2, 3, 4}。约束条件。
$Q_i \neq Q_j$ (不能在同一行)
$|Q_i - Q_j| \neq |i - j|$（或同一对角线）。

例子。密码运算

- 在$n \times n$棋盘上放置n个皇后，这样就不会有两个皇后互相攻击。

$$\begin{array}{cccccc} & S & E & N & D \\ + & M & O & R & E \\ \hline M & O & N & E & Y \end{array}$$

变量。
D E M N O R S Y

领域。

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

限制因素。

$M /= 0, S /= 0$（单数约束）。

$Y=D+E$或$Y=D+E-10$，等等。

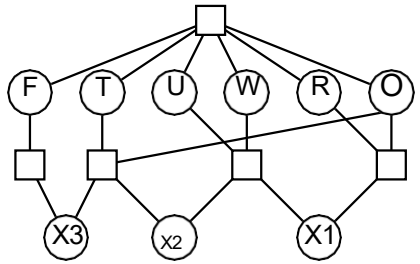
$D /= E, D /= M, D /= N$，等等。

新南威尔士大学
©W.Wobcke等人，2019-2022年

隐性变量的加密运算

我们可以添加 "隐藏 "的变量来简化约束。

```
   T W O
 + T W O
---------
基金会
```



变量。F T U W R O X_1 X_2 X_3

领域。{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

限制条件。AllDifferent(F, T,U,W,R,O) O + O = R + 10-X_1， 等等。

例子。数独

9				6				3
1		5		9	3	2		6
	4			5				9
8						4	7	1
		4	8	7				
7		2	6		1			8
2								
5				3	2		9	4
	8	7		1	6	3	5	

真实世界的CSP

- 作业问题（例如，谁教什么课）。
- 时间安排问题（例如，何时何地提供哪种课程？）
- 硬件配置（如尽量减少电路布局的空间）。
- 运输调度（例如，快递员送货，车辆路线）。
- 工厂调度（优化工作分配给机器）。
- 闸口分配（将闸口分配给飞机，使过境时间最小化）许多

现实世界的CSP也是优化问题

CSP的种类

离散变量

- 有限域；大小$d \Rightarrow O(d^n)$完整的赋值
  - 例如，布尔CSP，包括布尔可满足性（NP-complete）。
- 无限域（整数、字符串等）。
  - 工作车间调度，变量是每项工作的开始/结束日期
  - 需要一种约束性语言，例如$StartJob_1 + 5 \leq StartJob_3$
  - 线性约束可解，非线性不可解 连续变量
- 例如，哈勃望远镜观测的开始/结束时间

- **可通**过**LP方法在多**项式时间内解决的线性约束

新南威
尔士大
学
©W.Wobcke等人，2019-
2022年

约束的类型

- 一元约束涉及一个单一的变量
  - M /= 0
- 二元约束涉及成对的变量
  - SA /= WA
- 高阶约束涉及3个或更多的变量
  - $Y = D + E$ 或 $Y = D + E - 10$
- 连续变量的不等式约束
  - $\text{EndJob}_1 + 5 \leq \text{StartJob}_3$
- 软约束（偏好）。
  - 上午11点的讲座比8点的讲座好!

回溯搜索

CSP可以通过给变量逐一赋值，以不同的组合来解决。每当一个约束条件被违反时，就回到最近分配的变量上，给它分配一个新的值。

可以在一种特殊的状态空间上使用深度优先搜索来实现，其中状态是由迄今分配的值来定义的。

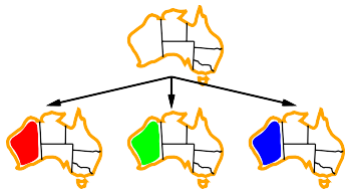
- 初始状态。空任务
- 继任函数。为一个未分配的变量分配一个值，该值不会与之前分配的其他变量的值冲突
- 目标状态。所有的变量都被分配了一个值，所有的约束条件都得到了满足

路径搜索与约束条件满足

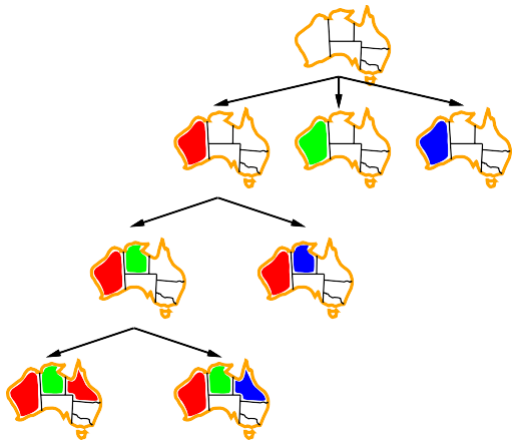
路径搜索问题和CSP之间的重要区别

- 限制性满足问题（如*n-Queens*）。
  - 困难的部分是知道最终状态
  - 如何到达那里很容易
- 路径搜索问题（如魔方）。
  - 知道最终状态很容易
  - 困难的部分是如何到达那里

逆向搜索实例

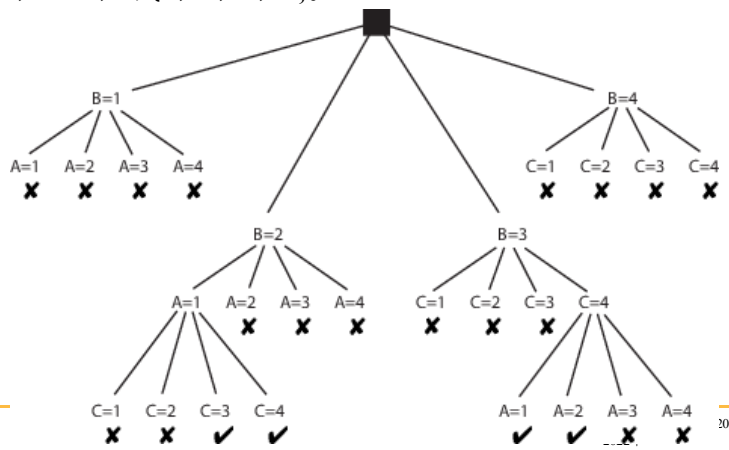


逆向搜索实例



回溯搜索的问题

$A<B$，$B<C$，域{1，2，3，4}。



逆向搜索空间属性

搜索空间具有非常具体的属性

- 如果有*n*个变量，每个解决方案的深度正好是*n*个
- 变量赋值是换算的
  [WA = red then NT = green] 同[NT = green then WA = red]

回溯搜索可以解决*n*≈25的*n-Queens问题*

对回溯搜索的改进

通用的启发式方法可以带来巨大的速度提升

1. 接下来应该分配哪个变量？
2. 应该按照什么顺序尝试其价值？
3. 能否及早发现不可避免的失败？

最小剩余值

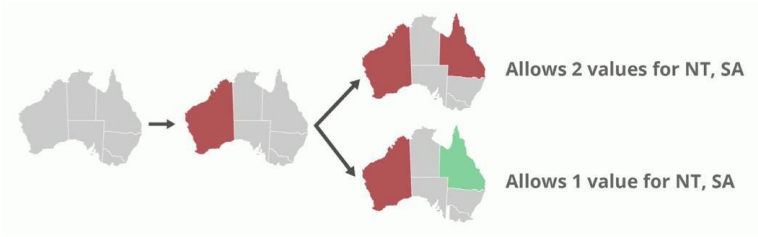
- 最小剩余值(MRV)
  - 选择合法值最少的变量
  - 如果多于一个，则在它们之间随机选择
  - 应用约束条件来消除其他变量的值

学位启发式

- MRV变量之间的并列关系
  - 选择对其余变量制约最大的变量

最小约束值

- 给定一个变量，选择约束性最小的值
  - 排除其余变量中最少值的一个变量

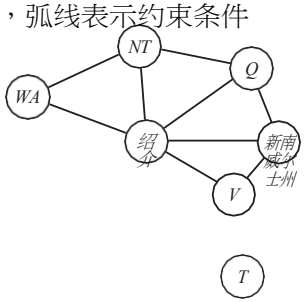


更为普遍的是，3个允许值会比2个更好，等等。将这些启发式方法结合起来，使得1000个阈值是可行的

约束图

二元CSP：每个约束条件最多涉及两个变量

约束图。节点是变量，弧线表示约束条件

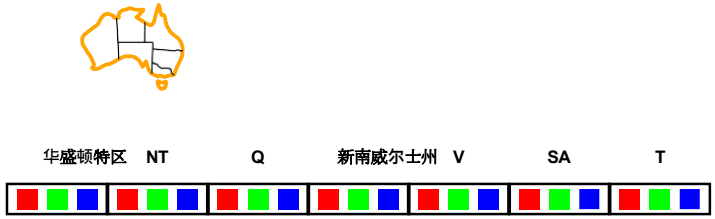


通用的CSP算法使用图结构来加快搜索速度，例如，Tasmania是一个独立的子问题!

新南威尔士大学
©W.Wobcke等人，2019-2022年

前瞻性检查

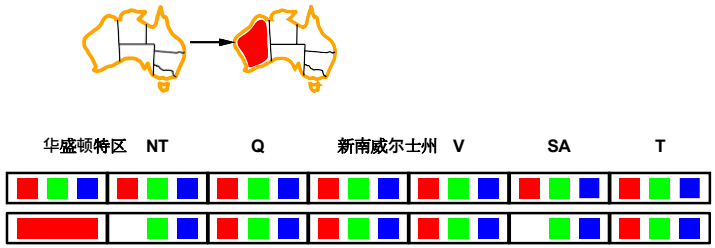
想法。追踪未分配变量的剩余合法值



前瞻性检查

想法。跟踪未分配变量的剩余合法值

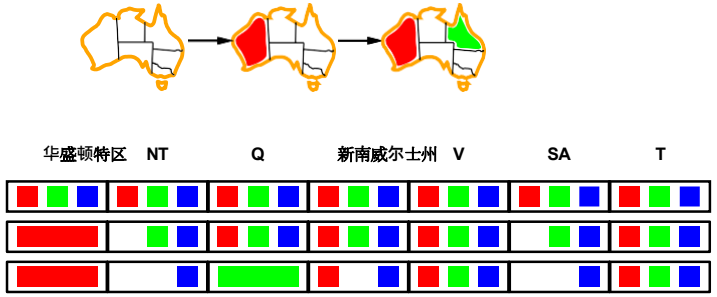
当任何变量没有合法值时，终止搜索



前瞻性检查

想法。跟踪未分配变量的剩余合法值

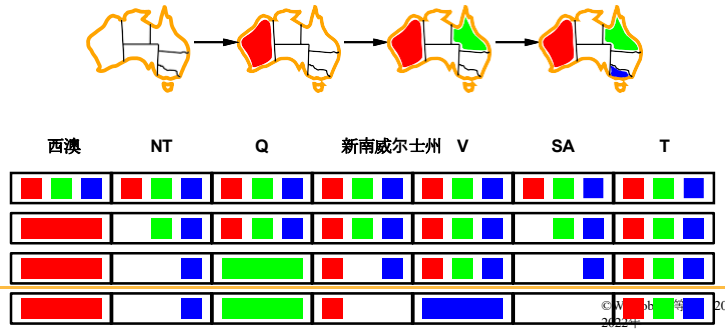
当任何变量没有合法值时，终止搜索



前瞻性检查

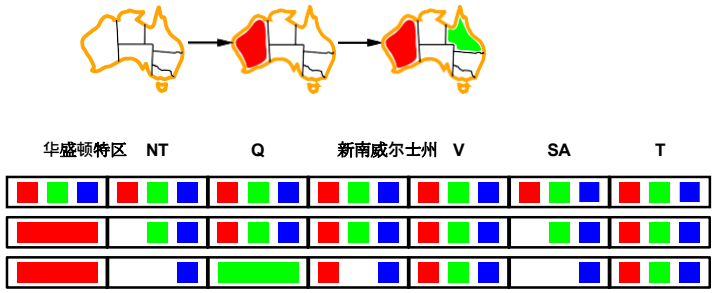
想法。跟踪未分配变量的剩余合法值

当任何变量没有合法值时，终止搜索



约束传播

前向检查将信息从分配的变量传播到未分配的变量，但并不能为所有的故障提供早期检测。



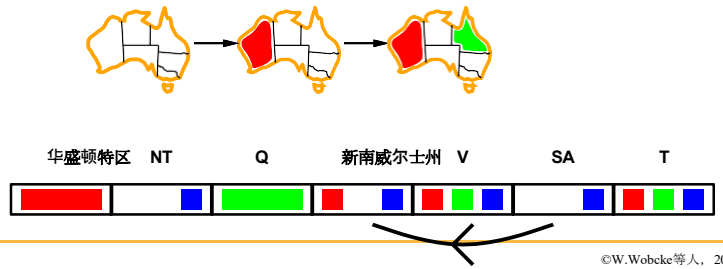
NT和SA不可能都是蓝色的!

约束传播重复地在本地执行约束

弧形的连贯性

约束传播的最简单形式是弧形一致

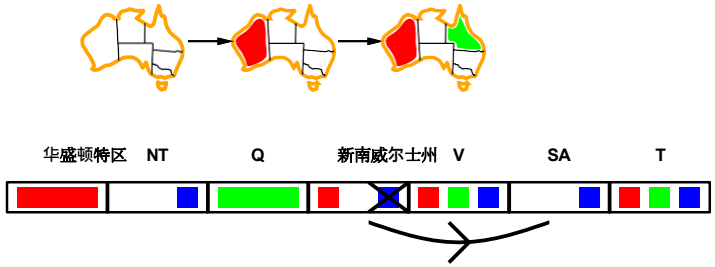
弧形（约束）$X \rightarrow Y$是弧形一致的，如果
对于$dom(X)$中的每个值x，$dom(Y)$中都有一些允许的y。



弧形的连贯性

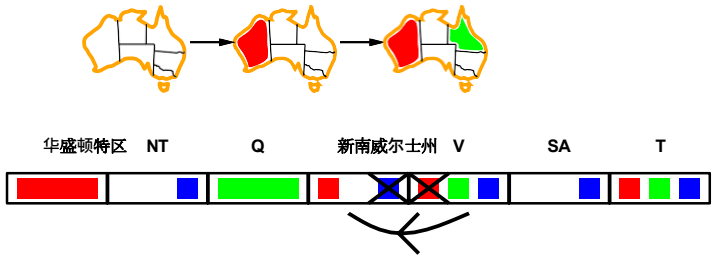
约束传播的最简单形式是弧形一致

弧形（约束）$X \rightarrow Y$是弧形一致的，如果
对于$dom(X)$中的每个值x，$dom(Y)$中都有一些允许的y。



弧形的连贯性

弧形（约束）$X \rightarrow Y$是弧形一致的，如果
对于$dom(X)$中的每个值x，$dom(Y)$中都有一些允许的y。



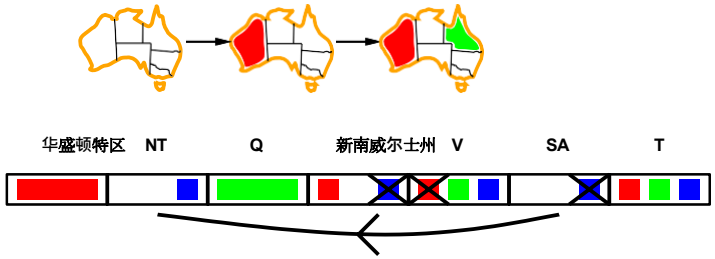
通过从$dom(X)$中删除任何这样的x，使$X \rightarrow Y$弧形一致。

如果X失去了一个值，X的邻居需要被重新检查

新南威尔士大学
©W.Wobcke等人，2019-2022年

弧形的连贯性

弧形（约束）$X \rightarrow Y$是弧形一致的，如果
对于$dom(X)$中的每个值x，$dom(Y)$中都有一些允许的y。



弧形一致性比前向检查更早发现故障
对于某些问题，它可以极大地加快搜索速度
对其他人来说，由于计算上的开销，它可能会减慢搜索速度

域拆分和弧形一致性

国家是整个CSP（而不是部分分配）。

- 使CSP领域一致和弧形一致
  - 领域一致 = 满足所有单项约束条件
- 使用深度优先搜索来解决CSP
  - 选择一个域内有多个值的变量v
  - 将v的域分成两个子集
  - 这就得到了两个较小的CSP
  - 使每个CSP弧线一致（如果可能）。

限制性优化问题

国家是整个CSP（而不是部分分配），有成本

- 使CSP领域一致和弧形一致
- 将CSP添加到优先队列中
- 使用贪婪搜索来解决CSP
  - 从优先级队列中删除具有最小h的CSP
  - 选择一个域内有多个值的变量v
  - 将v的域分成两个子集
  - 这就得到了两个较小的CSP
  - 使每个CSP弧线一致（如果可能）--添加到优先级队列中
- $cost$(CSP)≈违反软约束的成本之和

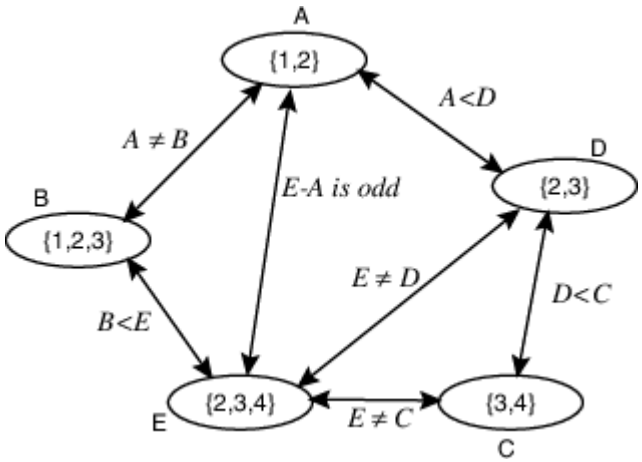
变量消除

- 解决每一个产生的CSP（如果无法解决，则回溯）。

- 如果只有一个变量，则返回其（单项）约束的交集
- 否则
  - 选择一个变量X
  - 将X出现在其中的约束连接起来，形成约束R1
  - 将R1投射到X以外的其他变量上，形成R2
  - 用R2替换所有X出现的约束条件
  - 递归解决简化问题，形成R3
  - 将R1与R3连接起来返回

新南威尔士大学
©W.Wobcke等人，2019-2022年

变量消除实例



变量消除实例

$r_1 : C /= E$	C	E
	3	2
	3	4
	4	2
	4	3

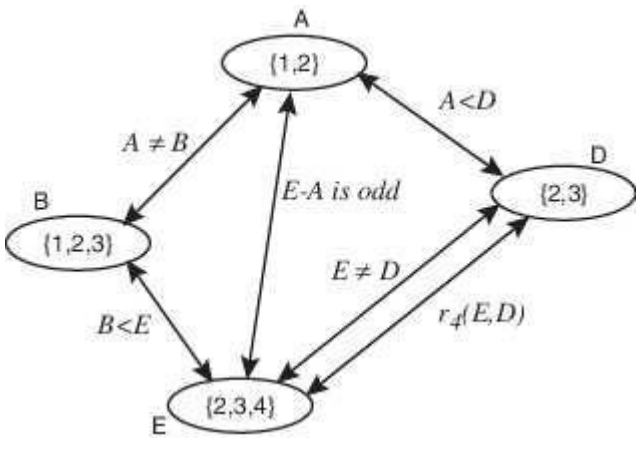
$r_2 : C > D$	C	D
	3	2
	4	2
	4	3

$r_3 : r_1 \lhd\rhd r2$	C	D	E
	3	2	2
	3	2	4
	4	2	2
	4	2	3
	4	3	2
	4	3	3

$r4 :$ $\pi\{D,E\}r3$	D	E
	2	2
	2	3
	2	4
	3	2
	3	3

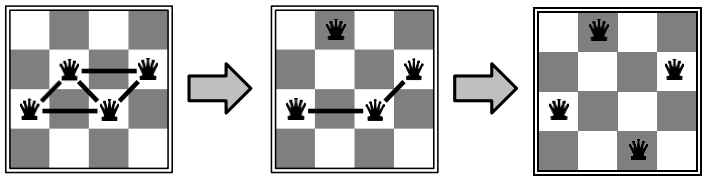
新的制约因素

变量消除实例



本地搜索

- 迭代式改进
  - 随机分配所有变量（可能违反约束条件）
  - 每次改变一个变量，试图减少每一步的违规次数
  - 贪婪搜索，h = 违反的约束数量

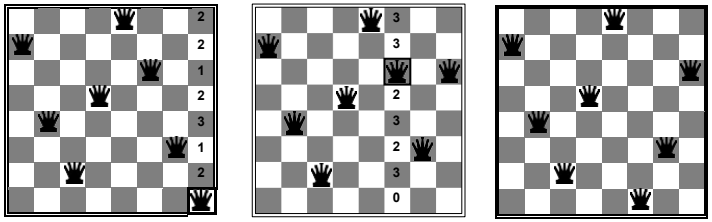


新南威
尔士大
学

©W.Wobcke等人，2019-
2022年

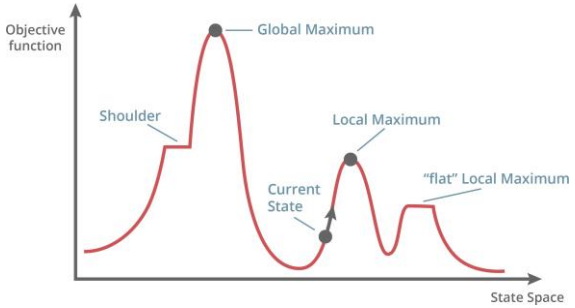
h = 5h **= 2h** **= 0**

小冲突的爬坡活动



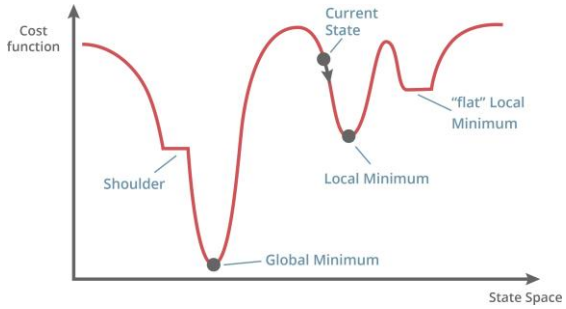
- 变量选择：随机选择任何有冲突的变量
- 通过最小冲突启发式的价值选择
  - 选择违反最小约束的值
  - 可以（经常）解决n≈10，000，000的*n-Queens问题*

高原和当地人的选择



有时，必须从侧面甚至倒退，才能在实现全局最优解决方案方面

颠倒的观点



当最小化违反限制条件时，考虑从山脊顶部开始，向下爬到山谷中是有意义的

模拟退火

取得进展。

- 基于前一状态(h_0)和新状态(h_1)的评估差异的随机爬坡法
  - 如果$h_1 < h_0$，一定要进行修改。
  - 否则，以概率的方式做出改变

e-($h1$-$h0$)/T

其中T是一个 "温度 "参数

- 当T=0时，减少到普通的爬坡运动
- 当$T \rightarrow \infty$时成为完全随机的搜索
- 有时，在搜索过程中逐渐减少T的值

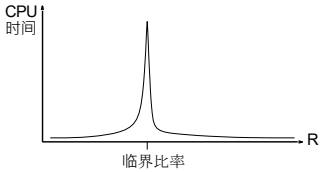
新南威尔士大学
©W.Wobcke等人，2019-2022年

CSP中的相变

给定随机初始状态，通过随机重启的最小冲突进行爬坡，可以在几乎恒定的时间内解决任意*n*的高概率问题（如*n*=10 ， 000 ， 000）。

如果约束条件非常少或非常多，随机生成的CSP往往很容易，但在一个狭窄的比率范围内会变得特别难。

$$R = \frac{\text{制约因素的数量}}{\text{变量的数量}}$$



摘要

- 对CSP在现实世界的应用很感兴趣
- 回溯=深度优先搜索，每个节点分配一个变量
- 变量和值排序启发式方法有很大帮助
- 前瞻性检查有助于及早发现不可避免的故障
- 小冲突的爬坡在实践中往往是有效的
- 模拟退火可以帮助摆脱局部优化的影响
- 哪种方法是最好的，因不同的任务而异!