人工智能导论

腾讯云人工智能特色班课程

主讲人：高　灿，致腾楼936

(davidgao@szu.edu.cn)

时　　间：周三晚上11-12节 致理楼L1-306（理论）

周三晚上13-14节(单) 致腾楼318（实验）

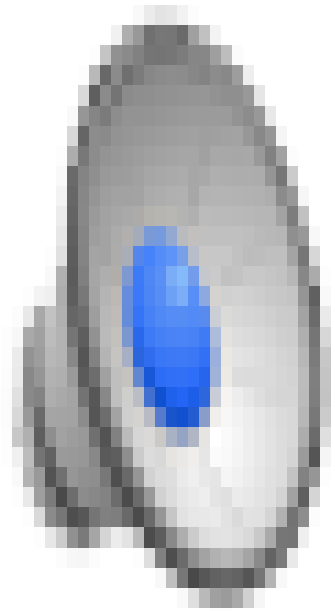
深圳大学 计算机与软件学院

College of Computer Science and Software Engineering of Shenzhen University

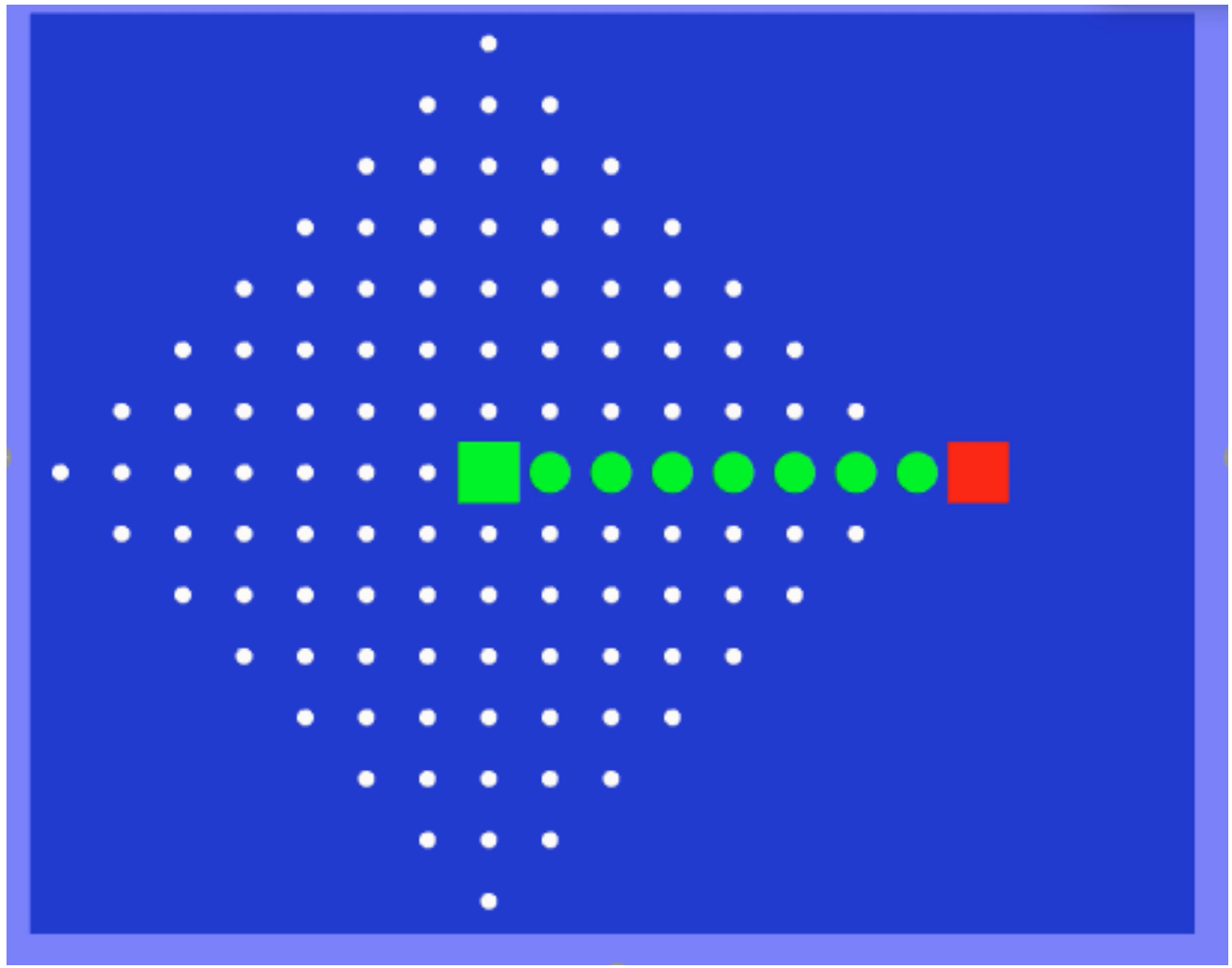
内容回顾

	后向代价	前向代价	备注
BFS	g(n)：深度层次	N/A	
DFS	g(n)：深度层次	N/A	
UCS	g(n)：真实代价	N/A	退化为BFS
GBFS	N/A	h(n)：估计代价	退化为DFS
A*	g(n)：真实代价	h*(n)：估计代价	
…			

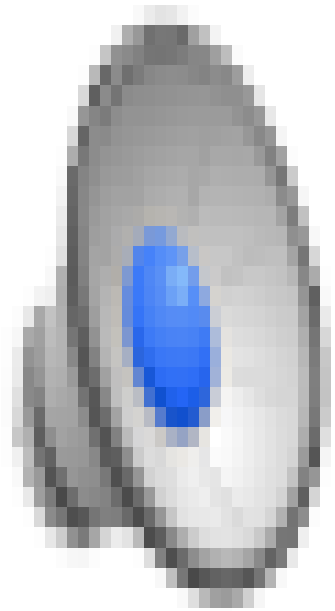
内容回顾



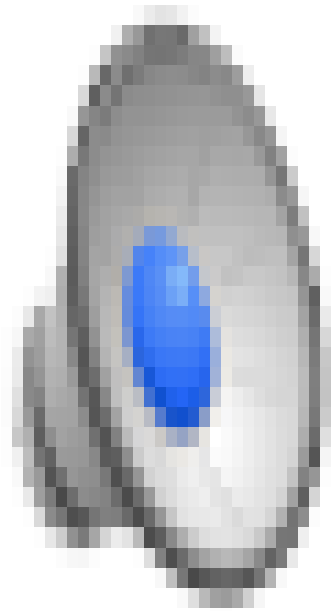
内容回顾



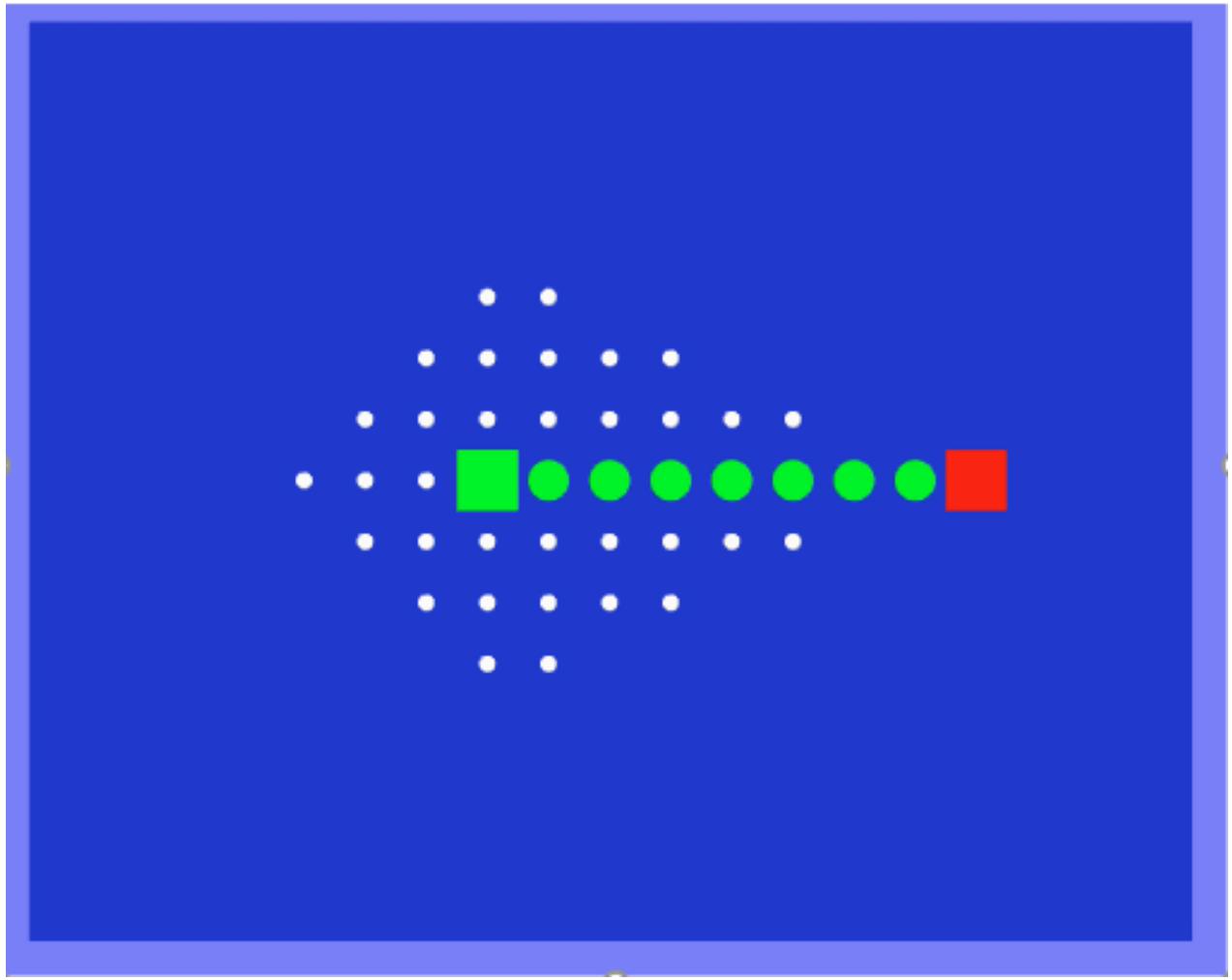
内容回顾



内容回顾



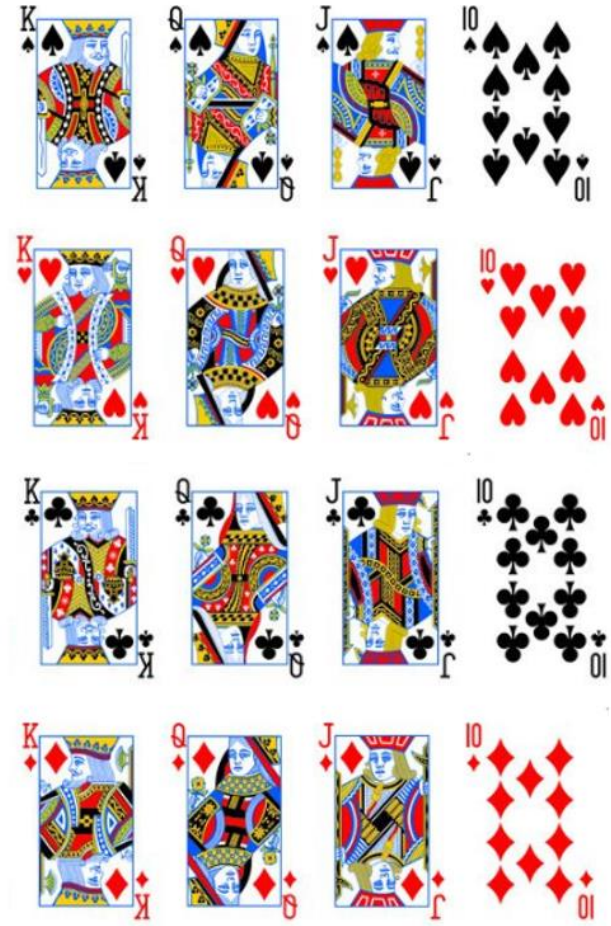
内容回顾



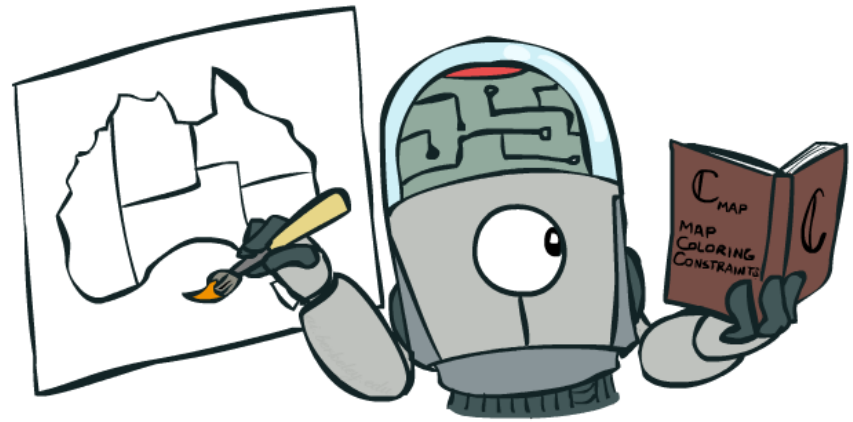
内容回顾

练习题

扑克牌问题：给定红心、黑桃、梅花和方块各花色的10，J，Q，K牌，共计16张。试将牌排列组合，使每行、每列、正反对角线都由不重复的10，J，Q，K构成。



目录



1. 约束满足问题

标准搜索问题：

状态表示：数据结构

后继函数：操作或动作

初始及目标状态：测试

问题的解：将开始状态转换为目标状态的一系列动作

约束满足问题：

搜索问题的特殊子集

约束满足问题包含：

变量的集合$X = \{X_1, X_2, \dots, X_n\}$

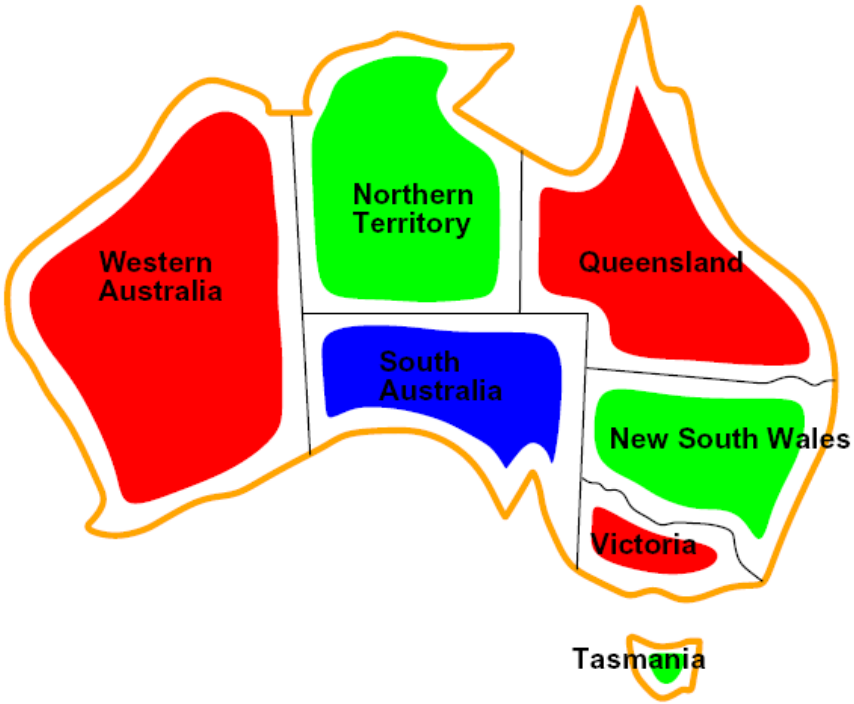
值域的集合$D = \{D_1, D_2, \dots, D_n\}$，每个变量有自己的值域

约束的集合：描述变量取值的约束

问题的解：满足约束的所有变量分配

1. 约束满足问题

问题实例



地图着色

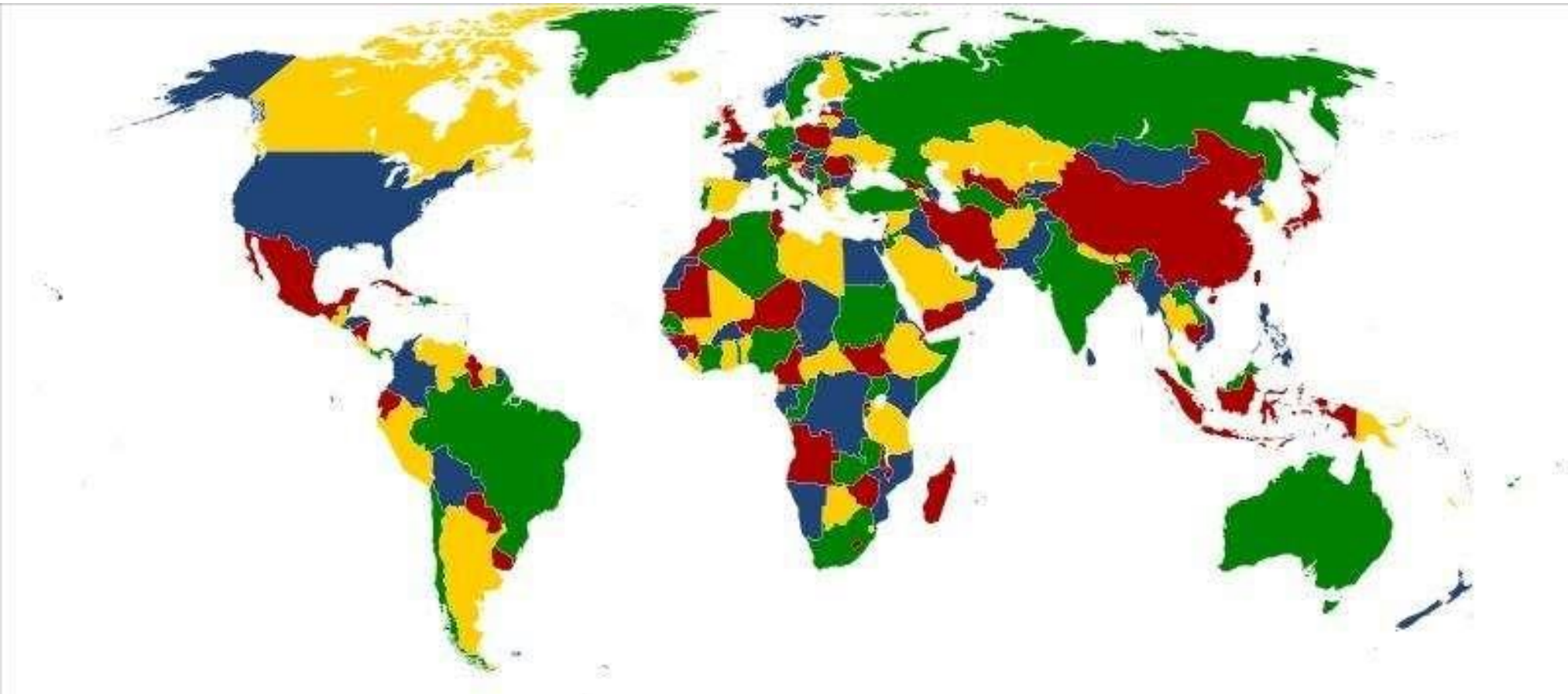
形式化描述

- 变量: WA，NT，Q，NSW，V，SA，T
- 值域: Color={red, green, blue}
- 约束条件: **相邻区域颜色不同**

  Color(WA)≠Color(NT)
- 满足条件的解:

  {WA=red，NT=green，Q=red，NSW=green，V=red，SA=blue，T=green}

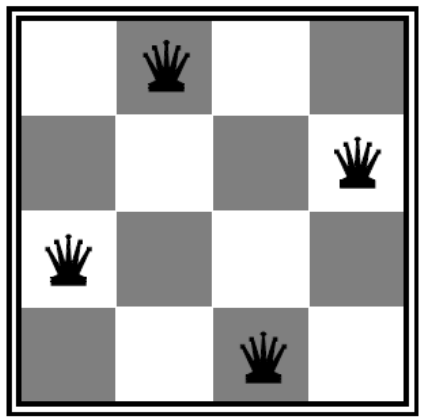
1. 约束满足问题



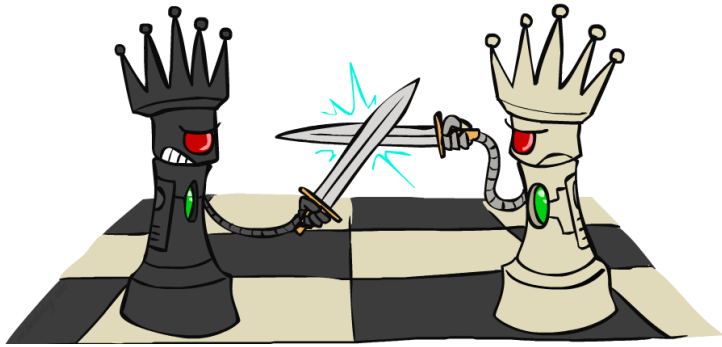
四色地图

1. 约束满足问题

问题实例



N皇后



形式化描述1

- 变量：X_{ij}
- 值域：{0,1}
- 约束条件

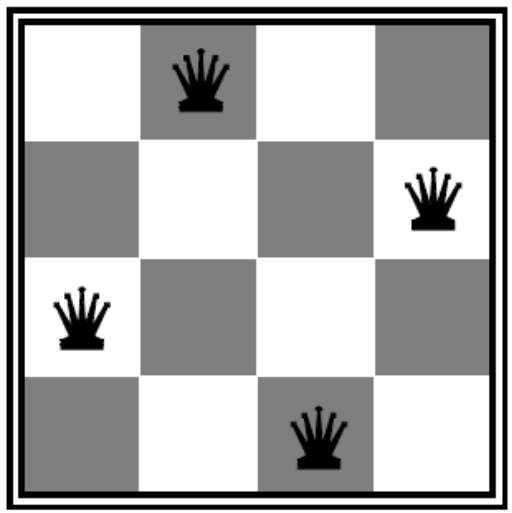
$$\sum_{i,j} X_{ij} = N$$

$$\forall i,j,k \quad (X_{ij}, X_{ik}) \in \{(0,0),(0,1),(1,0)\}$$
$$\forall i,j,k \quad (X_{ij}, X_{kj}) \in \{(0,0),(0,1),(1,0)\}$$
$$\forall i,j,k \quad (X_{ij}, X_{i+k,j+k}) \in \{(0,0),(0,1),(1,0)\}$$
$$\forall i,j,k \quad (X_{ij}, X_{i+k,j-k}) \in \{(0,0),(0,1),(1,0)\}$$

1. 约束满足问题

问题实例

Q_1
Q_2
Q_3
Q_4



形式化描述2

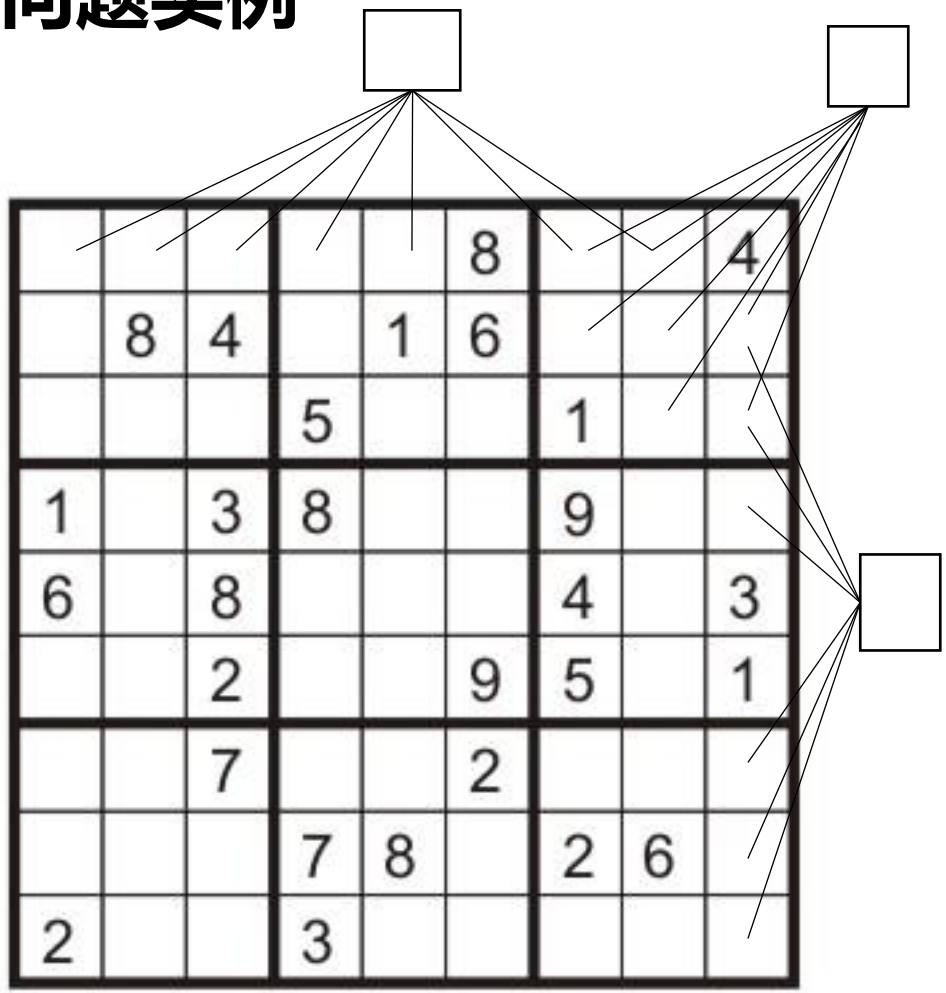
- 变量：Q_k
- 值域：$\{1, 2, 3, \ldots, N\}$
- 约束条件

$$\forall i, j \quad \text{non-threatening}(Q_i, Q_j)$$

$$(Q_1, Q_2) \in \{(1,3), (1,4), \ldots\}$$

1. 约束满足问题

问题实例



形式化描述

- 变量：空白方格
- 值域：$\{1,2,3,\ldots,9\}$
- 约束条件

  每行9个不同数字

  每列9个不同数字

  每块9个不同数字

1. 约束满足问题

变化形式

变量变化

有限值域：大小为d意味着$O(d^n)$种可能

连续值域：如任务规划，变量为每个任务的起始/终止时间

约束变化

单变量的一元约束(值域缩小)，如SA ≠ green

成对变量的二元约束，如SA ≠ WA

涉及3个或更多变量的高阶约束

全局约束（所有变量约束Alldiff），如数独

约束偏好

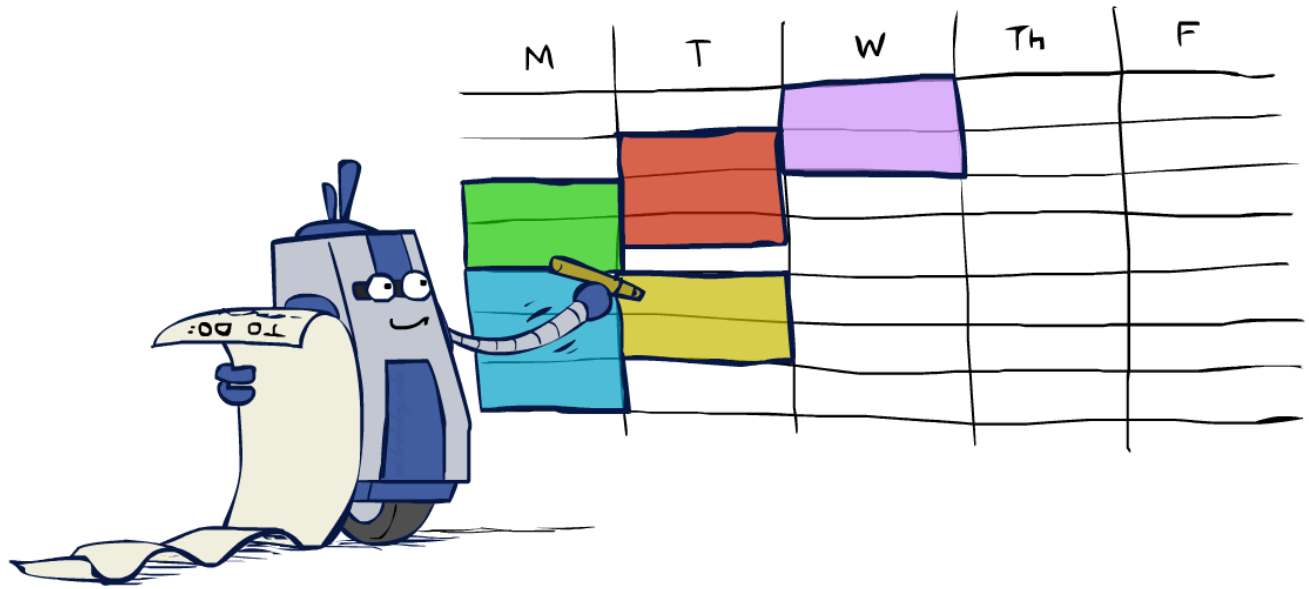
地图着色：红色比绿色更好

每个变量分配涉及不同代价

1. 约束满足问题

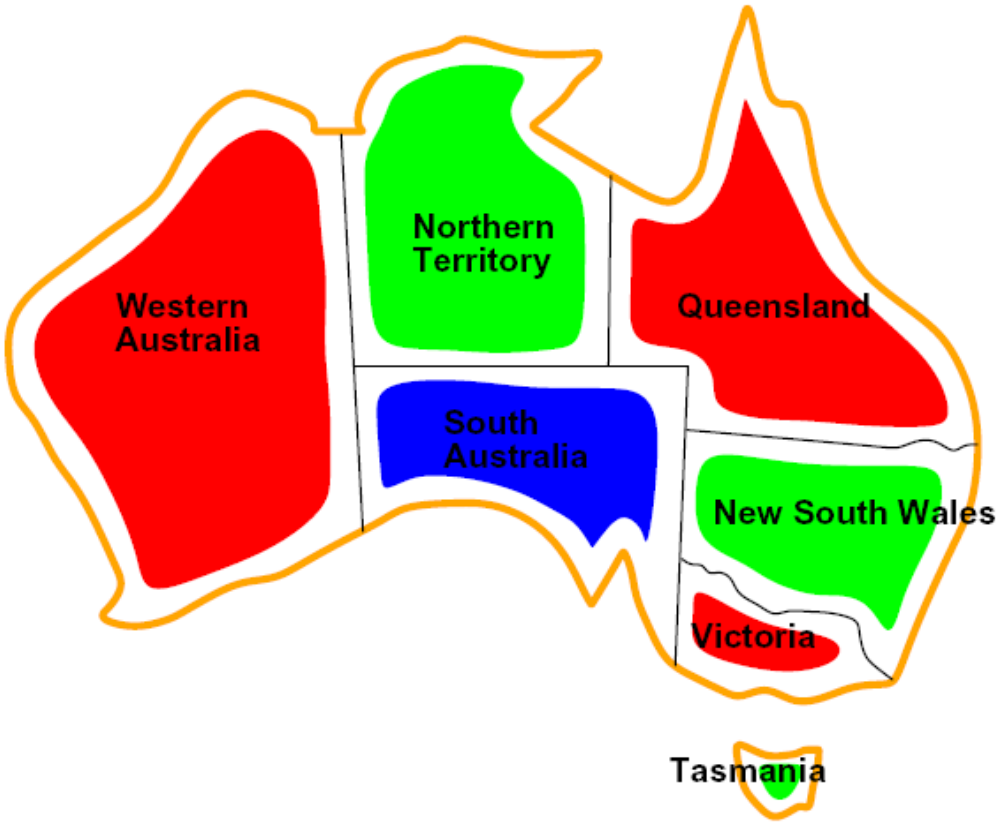
实际问题

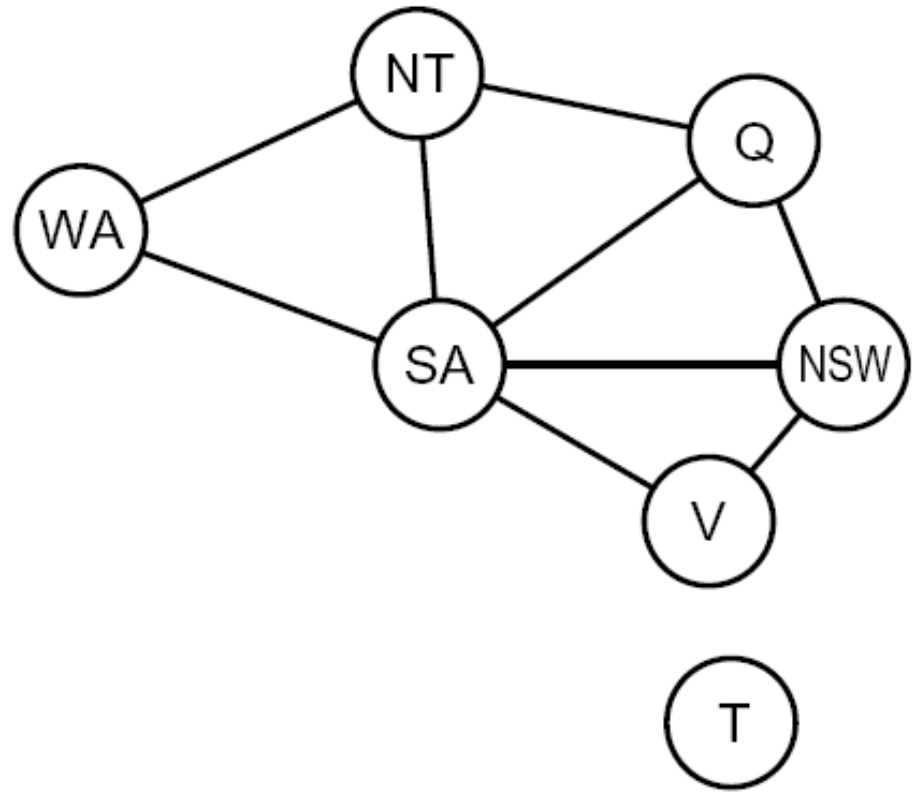
课程分配问题
排课问题
资源分配问题
产品装配
列车时刻表
工厂排班
等...



2. 回溯搜索求解

图着色-约束图





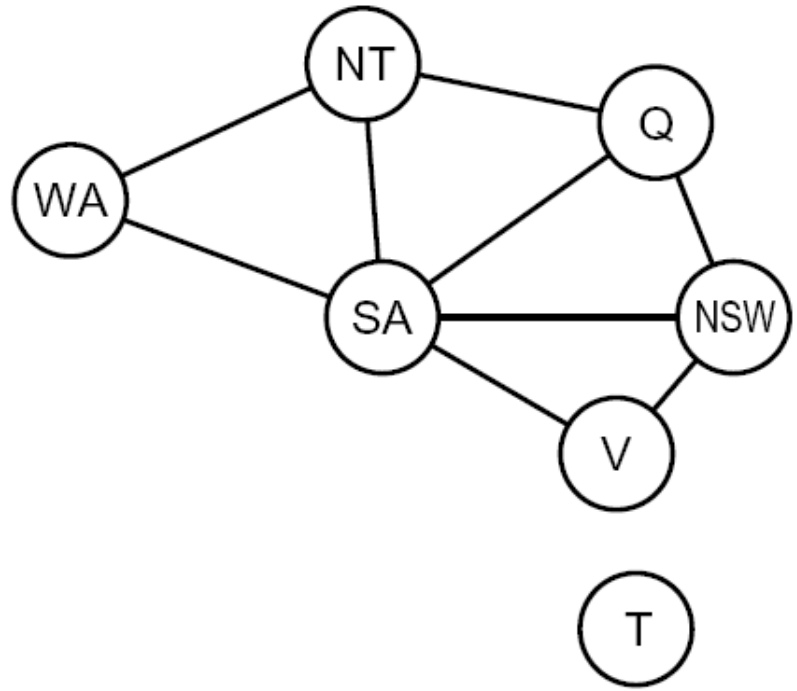
2. 回溯搜索求解

图着色-约束图

二元约束满足问题(CSP):每个约束最多与两个变量相关

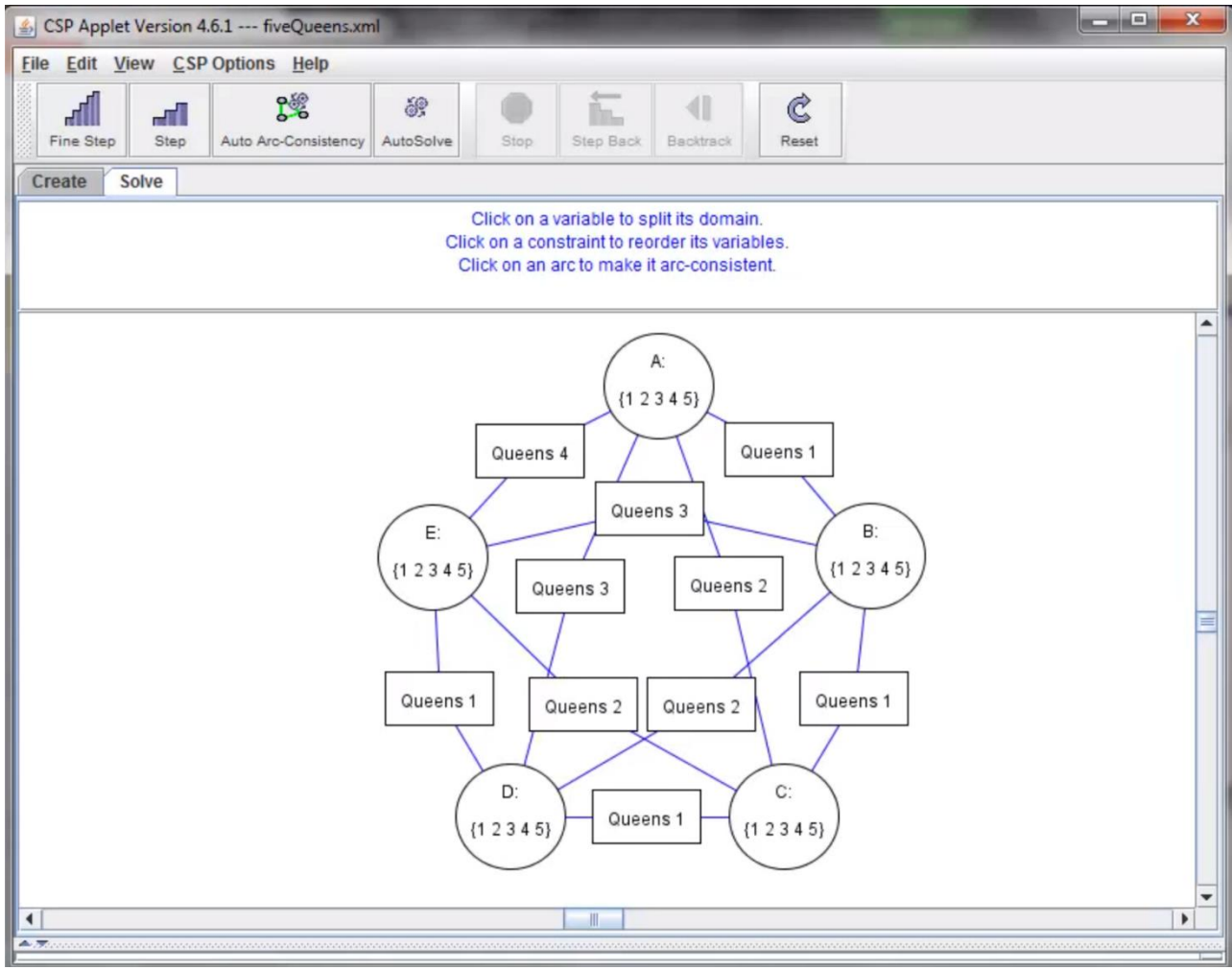
二元约束图：结点为变量，边表示约束

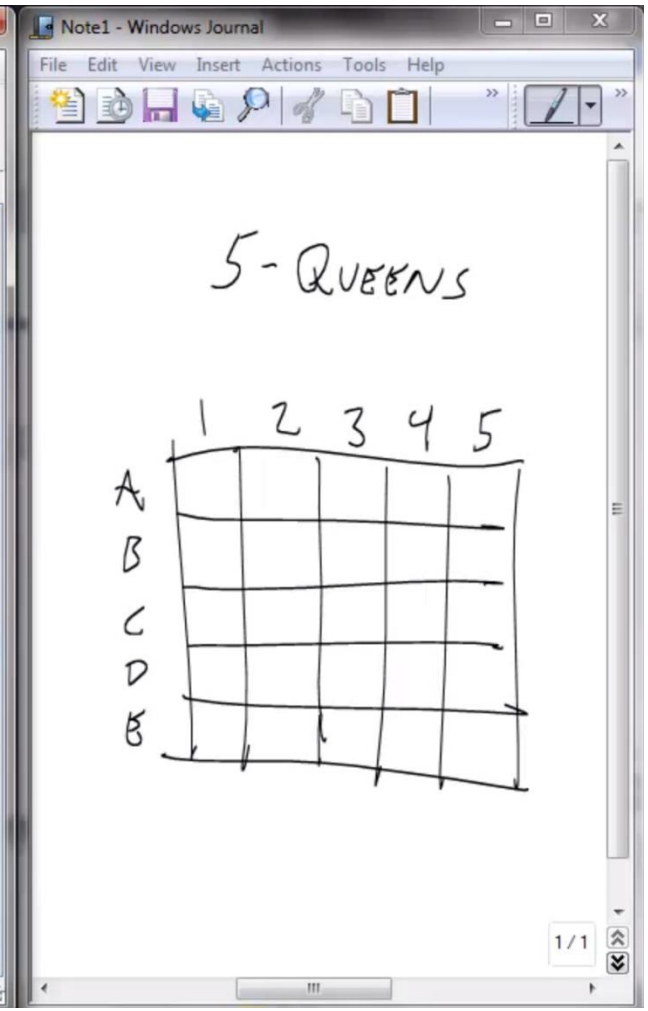
一般CSP算法利用图结构加速搜索



2. 回溯搜索求解

N皇后-约束图





2. 回溯搜索求解

算式谜-约束图

变量：$F, T, U, W, R, O, X_1, X_2, X_3$

值域：$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

约束：F, T, U, W, R, O值不同

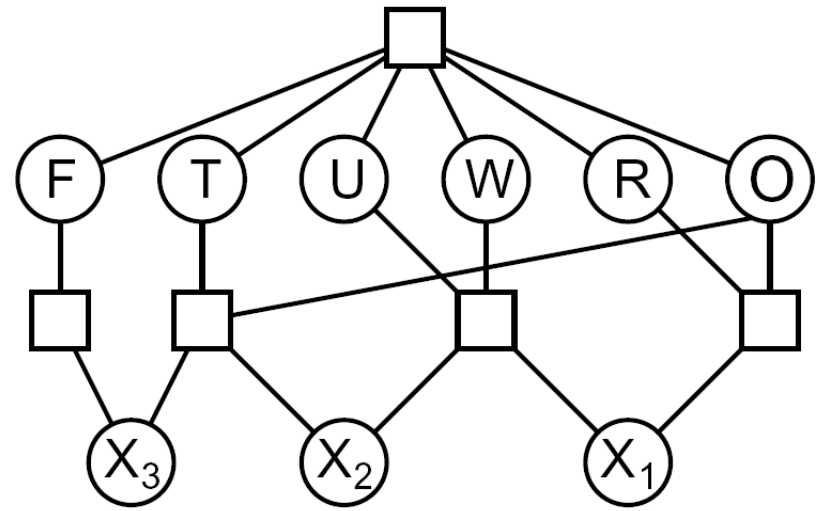
$$O + O = R + 10 \cdot X_1$$

$$W + W + X_1 = U + 10 \cdot X_2$$

$$T + T + X_2 = O + 10 \cdot X_3$$

$$F = X_3$$

$$\begin{array}{r} \text{T W O} \\ +\ \text{T W O} \\ \hline \text{F O U R} \end{array}$$



2. 回溯搜索求解

状态：

初始状态：所有变量未分配{}

后继函数：对未分配的变量分配一个值

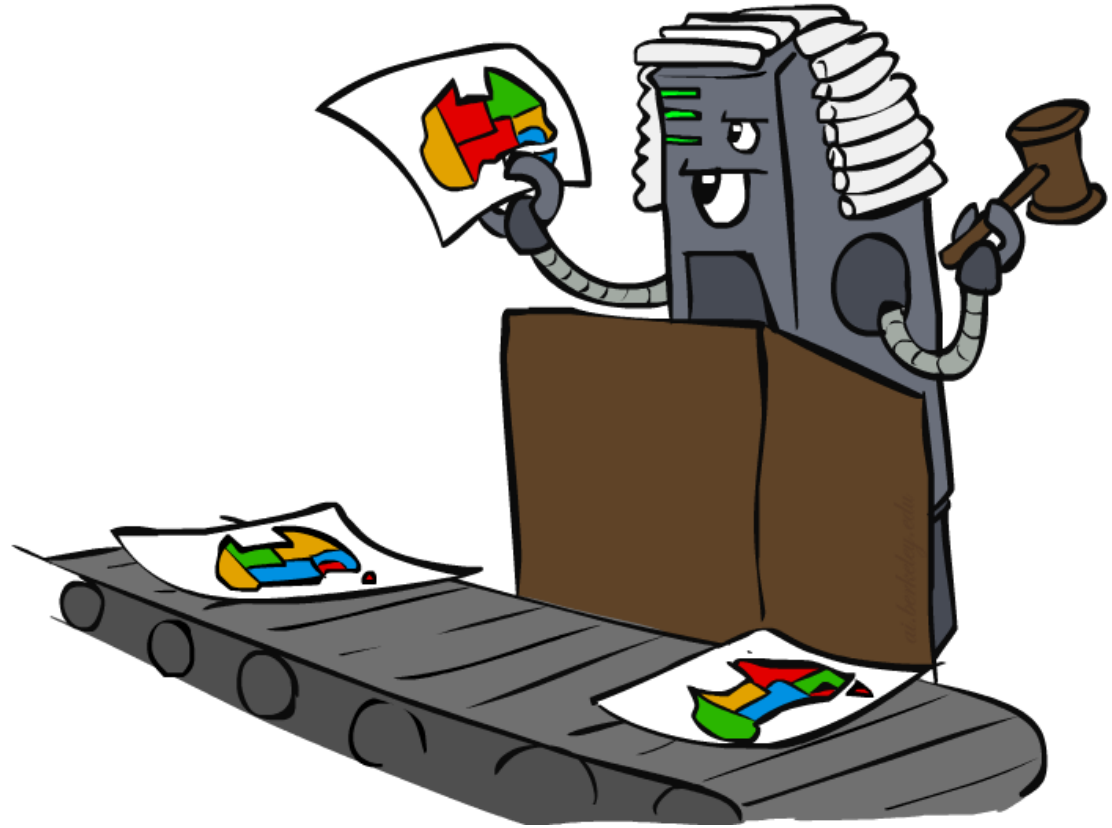
目标测试：变量都已分配值而且满足约束条件

搜索：

宽度优先?

深度优先?

简单搜索?



2. 回溯搜索求解

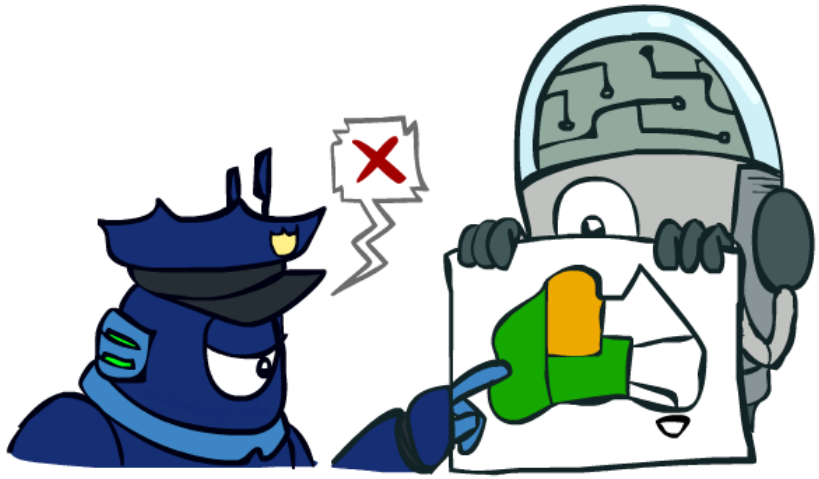
无信息搜索-回溯法

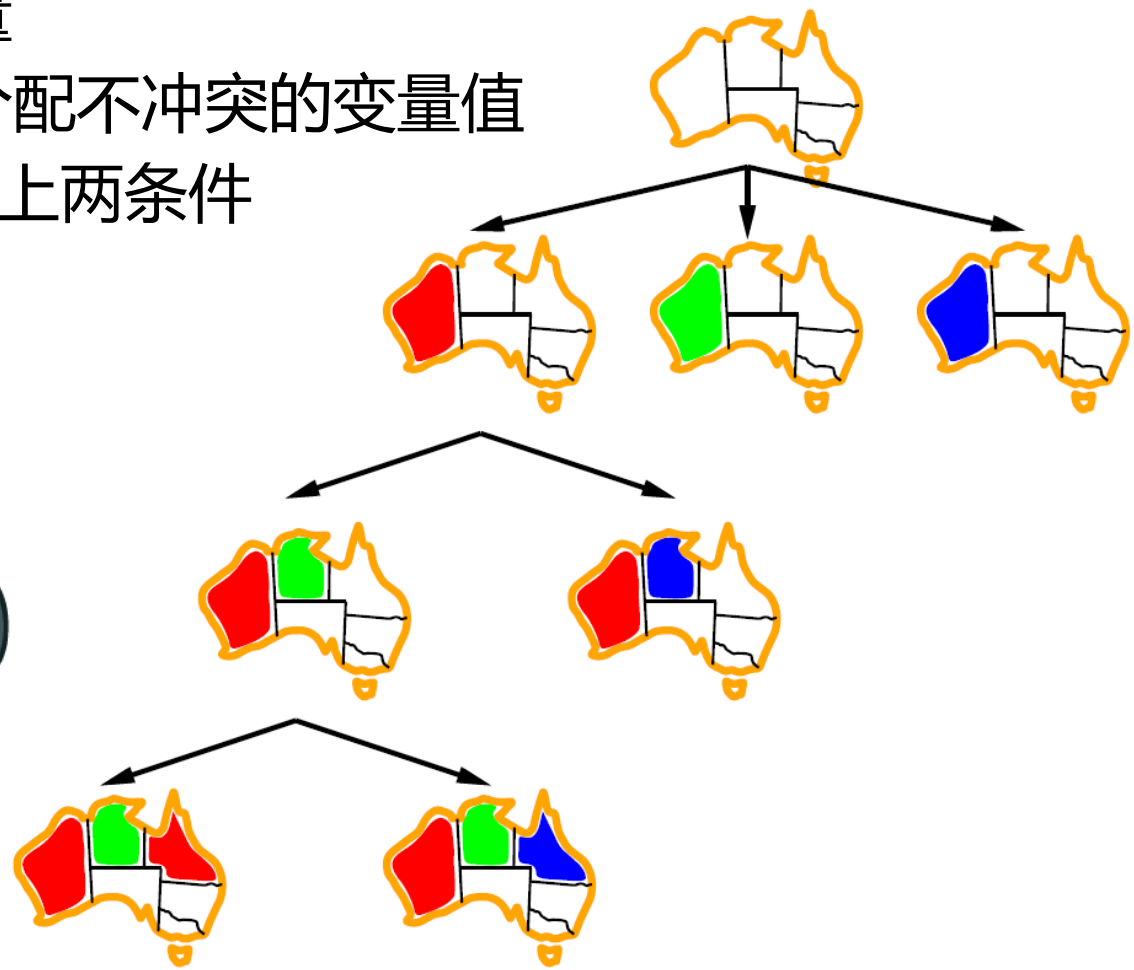
回溯搜索是求解约束满足问题最基本的无信息搜索算法

变量分配：每次一个变量

约束检查：考虑与前面分配不冲突的变量值

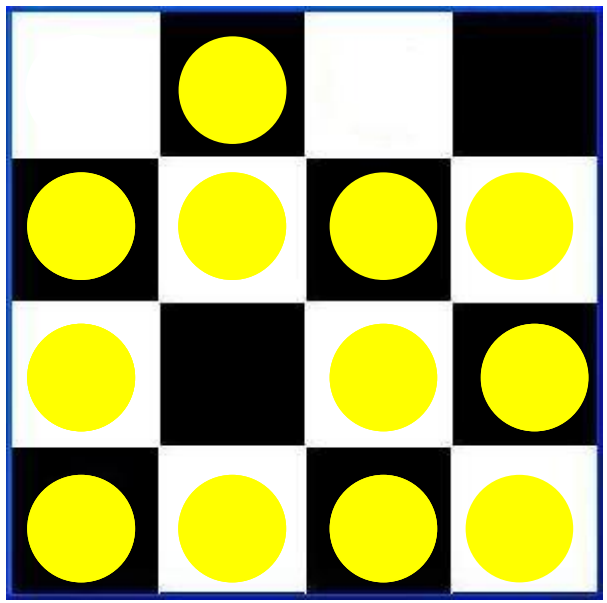
回溯搜索：深度优先+以上两条件

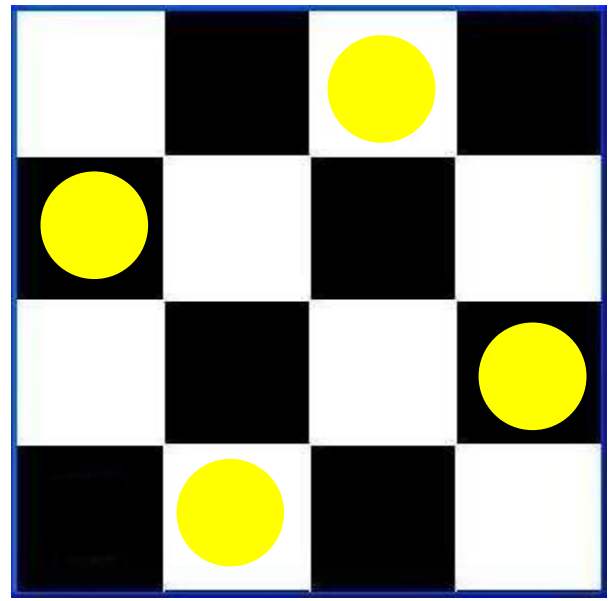




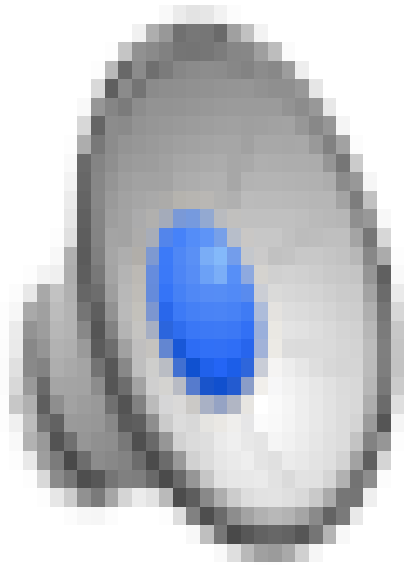
2. 回溯搜索求解

无信息搜索-回溯法

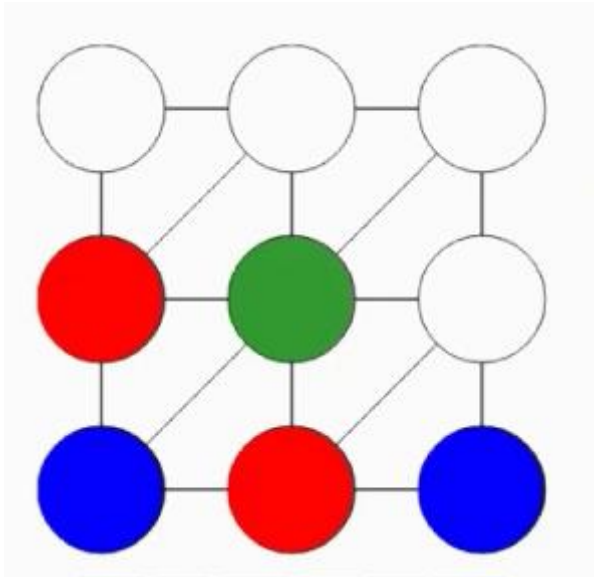


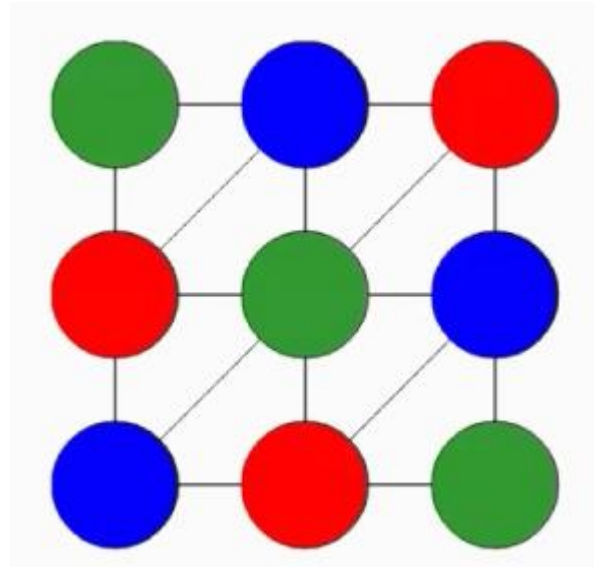


2. 回溯搜索求解



2. 回溯搜索求解





2. 回溯搜索求解

```
function BACKTRACK(csp, assignment ) returns a solution or failure
    if assignment is complete then return assignment
    var ←SELECT-UNASSIGNED-VARIABLE(csp, assignment )
    for each value in ORDER-DOMAIN-VALUES(csp, var, assignment ) do
        if value is consistent with assignment then
            add {var = value} to assignment
            inferences ←INFERENCE(csp, var , assignment )
            if inferences ≠ failure then
              add inferences to csp
              result ←BACKTRACK(csp, assignment )
              if result ≠ failure then return result
              remove inferences from csp
            else remove {var = value} from assignment
return failure
```

3. 搜索优化

简单的启发式策略可大大提高速度

排序策略

如何决定下一个需分配的变量 SELECT-UNASSIGNED-VARIABLE

如何选择尝试分配的值 ORDER-DOMAIN-VALUES

推理策略

能否尽早的发现可避免的分配尝试 INFERENCE

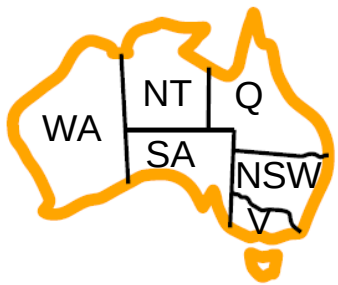
结构特性

能否利用问题结构特性 CSP

3. 搜索优化

变量排序-最少可取值（MRV）

变量排序：选择值域最少可取值的变量



该如何选择
下一个变量？

为什么不是最多值的变量？

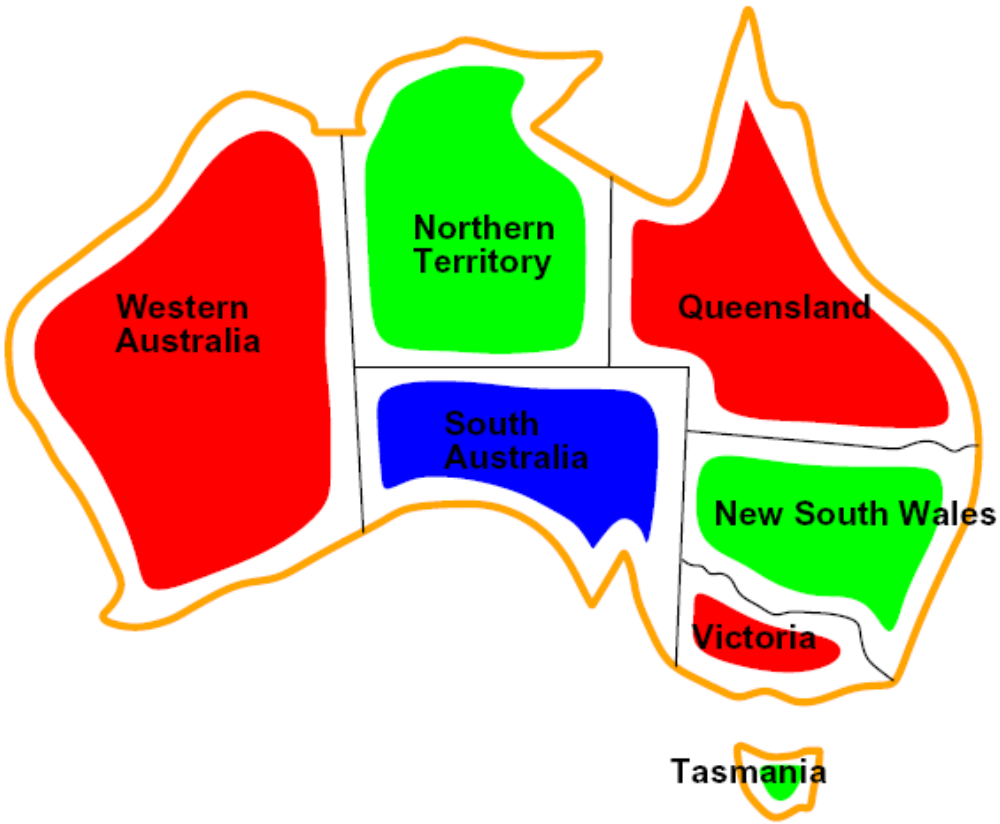
最少值变量也称为最多限制变量（变量空值，直接退出）

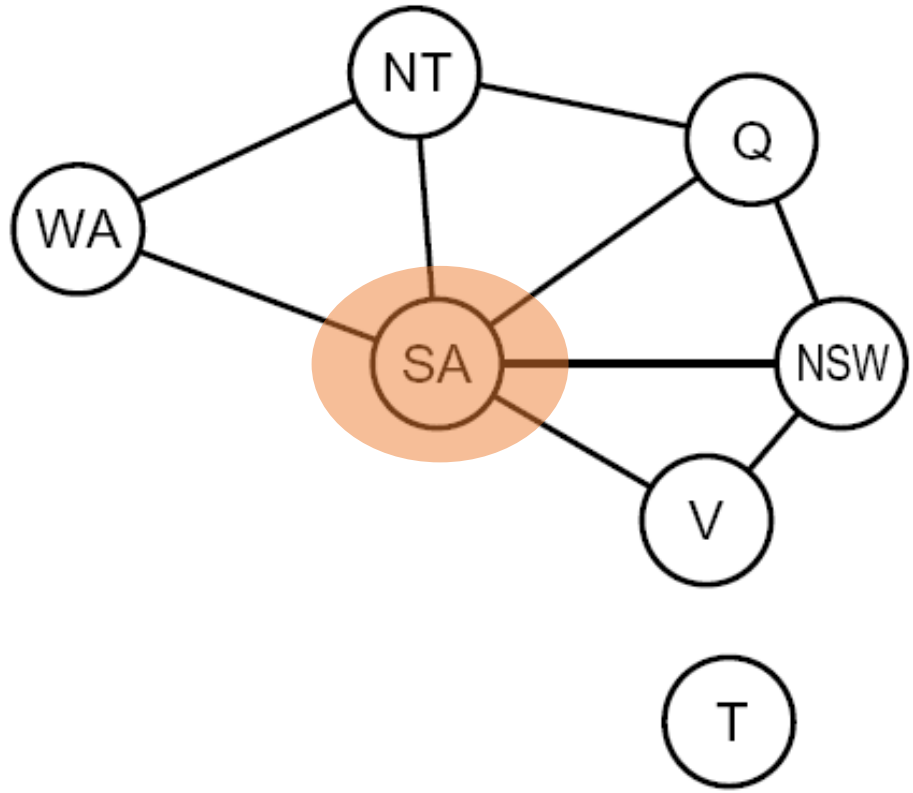
“失败-最快”排序-预剪枝，性能提升有时可达1000倍

3. 搜索优化

变量排序-度启发式

变量可取值数目相同时，选择**度较大**的变量

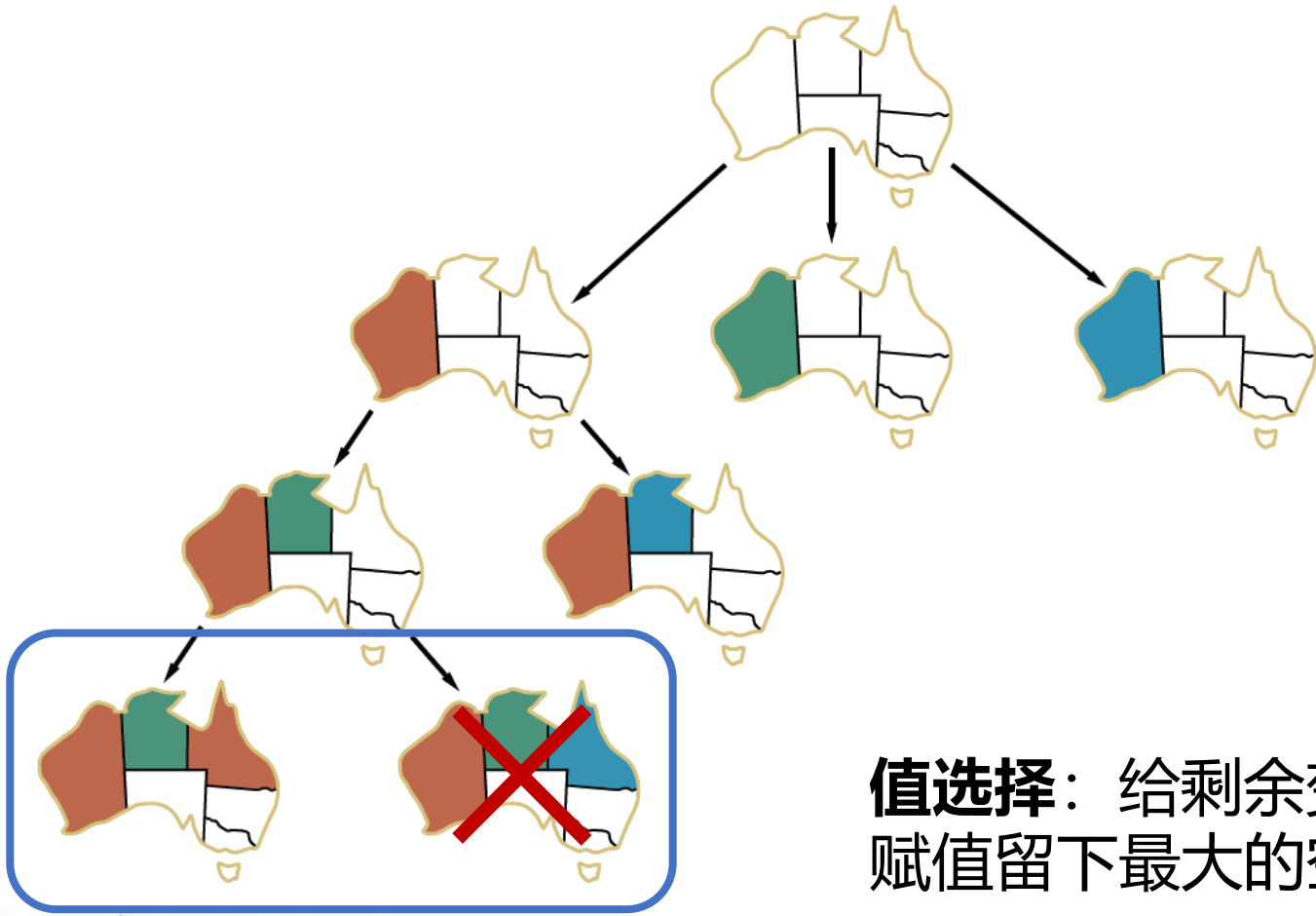




3. 搜索优化

值排序-最少约束值（MCV）

优先选择的值：给近邻变量留下更多选择的值



值选择：给剩余变量赋值留下最大的空间

3. 搜索优化

推理策略-约束传播

搜索过程进行**约束传播**的特殊**推理-局部相容性**。

使用约束来减小一个变量的合法取值范围，从而影响与此变量有约束关系的其它变量的取值。

约束传播与搜索可交替进行，亦可作为搜索前的预处理步骤。

结点相容

单个变量值域中的所有取值满足它的一元约束，则称此变量是**结点相容**的。

结点约束(偏好选择)：SA不喜欢绿色，则SA值域为{红色、蓝色}

3. 搜索优化

推理策略-约束传播

边相容

某变量值域中的所有取值满足它的所有二元约束，则称此变量是**边相容**的。

对于变量X和Y，若对变量X的每个取值X_i在变量Y中都存在Y_j满足边(X_i, Y_j)的二元约束，则称X相对Y是边相容的。

如果每个变量相对其他变量都是边相容的，则称该网络是边相容的。

路径相容

边相容通过边（二元约束）缩紧值域（一元约束）。路径相容通过观察变量得到隐式约束并以此来加强二元约束。

两变量X和Y对于第三个变量Z是相容的，指对每一个相容赋值(X_i, Y_j)，Z都有合适的取值使得(X_i, Z_k)和(Y_j, Z_k)是相容的。