

快速求解 CSP 问题的方法——以八皇后问题为例

08022311 陈鲲龙

2024 年 11 月 7 日

1 问题描述

八皇后问题要求在一个 8×8 的棋盘上摆放 8 个皇后，使其不能互相攻击，即任意两个皇后不处于同一行、同一列或同一斜线。为解决此问题，基于回溯搜索、节点相容、弧相容、变量排序、值排序、前向检验、弧相容保持、独立子问题、树结构 CSP 等模块设计一种快速求解 CSP 问题的方法。

2 问题建模

2.1 变量

定义变量 Q_1, Q_2, \dots, Q_8 ，每个变量 Q_i 表示第 i 行上的皇后所处的列位置，即变量 Q_i 的值表示第 i 行的皇后位于第几列。而每行只允许一个皇后，所以共有 8 个变量，分别代表棋盘的每一行。

2.2 域

每个变量 Q_i 的取值范围是 $\{1, 2, 3, \dots, 8\}$ ，表示棋盘上的列位置，即变量 Q_i 的值可以是任意一个从 1 到 8 的整数。

2.3 约束

八皇后问题的约束是确保任何两个皇后不能位于同一行、同一列或同一斜线上。

- **行约束**：每个变量 Q_i 只表示第 i 行，因此不同皇后自动在不同的行上。
- **列约束**：任何两个变量 Q_i 和 Q_j 的取值不能相同，即 $Q_i \neq Q_j$ 。如果两个变量的值相同，则它们在同一列。
- **对角线约束**：对于任意两个变量 Q_i 和 Q_j ，如果 $|i - j| = |Q_i - Q_j|$ ，则这两个皇后位于同一对角线上，因此需要避免这种情况。

总结约束条件为：

$$Q_i \neq Q_j \quad (\text{不同列})$$

$$|i - j| \neq |Q_i - Q_j| \quad (\text{不同对角线})$$

3 解法框架

3.1 回溯搜索

回溯搜索从第一行开始尝试在每一列放置皇后，在放置每个皇后之前，检查当前列和两条对角线上是否已有其他皇后，如果没有冲突，就将皇后放置在该位置，并递归尝试下一行，如果在某行中无法放置皇后（即所有列都冲突），则回溯到上一行，改变上一行皇后的位置，继续尝试新的解。这个过程不断进行，直到所有皇后都成功放置在棋盘上，或者所有可能的放置都被尝试过。如果找到一个有效解，就返回；否则，继续探索其他可能的解。

3.2 节点相容

节点相容指每个节点的状态（即皇后放置的位置）必须满足一定的约束条件，对于每一行的皇后放置位置，如果当前节点的放置没有违反列冲突或对角线冲突（即所有已经放置的皇后都不会互相攻击），则该节点是相容的。在回溯搜索过程中，只有当节点相容时，才会进一步探索该节点的子节点（即尝试在下一行放置皇后）。如果某个节点不满足相容性条件，则直接回溯，不再继续探索该路径，避免不必要的计算。

3.3 弧相容

对于任意两个皇后 Q_i 和 Q_j ，弧相容性要求它们在同一列或同一对角线上的情况被提前排除。在放置每个皇后时，如果某个放置位置与之前放置的皇后形成冲突，搜索过程就会进行剪枝，避免进入不合法的状态。通过弧相容性约束的提前检查，可以有效减少搜索空间，提高回溯搜索的效率。

3.4 变量排序

采用“最小剩余值”启发式，优先选择限制最多的变量，以减少回溯次数。

3.5 值排序

优先选择最少对其他变量产生冲突的值，即“最少约束值”启发式。

3.6 前向检验

在为某个皇后选择位置后，进行前向检验，若其他行或列无合法位置，则回溯。

3.7 弧相容保持

动态地在每步回溯后进行弧相容检查，以保证后续变量的合法值存在。

4 框架流程

Algorithm 1 八皇后问题 CSP 解法框架流程

```
1: 初始化棋盘和变量
2: while 未找到解且未回溯至根节点 do
3:   选择下一个变量（行）并按变量排序启发式进行尝试
4:   for 每个可行的位置（列） do
5:     为当前变量（行）选择一个位置（列）
6:     if 满足前向检验和弧相容保持条件 then
7:       递归到下一个变量
8:     else
9:       回溯到上一行并重新选择位置
10:    end if
11:  end for
12: end while
```
