

# Correction TD IA - 2026

Équipe Pédagogique IA

## 1 TD1 : Résolution de problèmes et Recherche

### Exercice 1 : Le problème des 8 dames

**Énoncé :** Modélez le problème de 8 dames en détaillant l'espace d'états, les actions, le modèle de transition, les états but. Détaillez les premiers deux niveaux de l'arbre de recherche. Analysez l'arbre correspondant.

---

#### Proposition de Corrigé

Il existe plusieurs modélisations. Nous présentons ici la modélisation **incrémentale par colonnes** (plus efficace que la modélisation naïve par placement libre).

#### 1. Modélisation formelle

- **Hypothèse :** On sait qu'une solution valide contient exactement une dame par colonne. On peut donc restreindre la recherche à placer la  $k^{\text{ème}}$  dame dans la colonne  $k$ .
- **État initial ( $s_0$ ) :** Une liste vide  $[]$  (aucune dame sur l'échiquier).
- **Espace d'états ( $S$ ) :** Un état est représenté par une liste (ou un vecteur) de longueur  $k$  (avec  $0 \leq k \leq 8$ ), où le  $i$ -ème élément représente la ligne de la dame dans la  $i$ -ème colonne.  
*Exemple :*  $p = [1, 5, 8]$  signifie qu'il y a des dames en (Col 1, Ligne 1), (Col 2, Ligne 5) et (Col 3, Ligne 8).
- **Actions ( $A(s)$ ) :** Ajouter une dame dans la colonne de gauche la plus vide (colonne  $k+1$ ).  
*Contrainte :* On ne place la dame que sur une ligne qui n'est pas déjà attaquée par les dames présentes (pas de conflit de ligne). Plus formellement, pour tout  $i$  (nouvelle dame) et pour tout  $j < i$  (dame déjà placée),  $p_i \neq p_j$ .
- **Modèle de transition :** Passe de l'état  $[p_1, \dots, p_k]$  à l'état  $[p_1, \dots, p_k, p_{k+1}]$  avec, pour tout  $i$ ,  $p_i \in [1, 8]$ .
- **Test de but :** La liste contient 8 dames ( $k = 8$ ) et aucune attaque mutuelle n'est détectée.  
*Formalisation :* Si on représente la configuration par un vecteur  $p$  où  $p[i]$  est la ligne de la reine dans la colonne  $i$ , la condition pour que deux reines  $i$  et  $j$  s'attaquent diagonalement est :

$$|p[i] - p[j]| = |i - j|$$

Le but est atteint si cette égalité n'est jamais vérifiée pour aucune paire  $(i, j)$  distincte.

# TD IA-2026修正案

IA教学团队

## 1 TD1：问题解决与研究

### 练习1：八皇后问题

**陈述：**建立八皇后问题的数学模型，需详细说明状态空间、动作集、转移模型及目标状态。具体阐述搜索树的前两个层级结构，并对对应搜索树进行分析。

---

#### 修正提案

存在多种建模方法。本文重点介绍**列式增量建模**（相较于自由放置的朴素建模更为高效）。

#### 1. 形式化建模

—**假设：**已知有效解中每列恰好包含一位女士。因此可以将搜索范围限制为将第 $k$ 位女士放置在第 $k$ 列。

—**初始状态 ( $s_0$ ) :**空列表 [] (棋盘上没有棋子)。

—**状态空间 ( $S$ ):**一个状态由长度为 $k$  (其中  $0 \leq k \leq 8$ ) 的列表 (或向量) 表示, 其中第 $i$ 个元素代表第 $i$ 列中的皇后所在行。例如:  $p = [1, 5, 8]$  表示皇后位于 (第1列, 第1行)、(第2列, 第5行) 和 (第3列, 第8行)。

—**操作 ( $A(s)$ ):**在最左侧的空列 (列 $k+1$ ) 中添加一个棋子。约束条件: 仅将棋子放置在未被现有棋子攻击的行上 (无行冲突)。更正式地说, 对于任意 $i$  (新棋子) 和任意 $j < i$  (已放置棋子),  $p_i \neq p_j$ 。

—**转移模型:**从状态 $[p_1, \dots, p_k]$  转移到状态 $[p_1, \dots, p_k, p_{k+1}]$ , 其中对于任意 $i$ ,  $p_i \in [1, 8]$ 。

—**目标测试:**列表包含8个皇后 ( $k=8$ ) 且未检测到相互攻击。形式化: 若用向量 $p$ 表示配置, 其中 $p[i]$  表示第 $i$ 列中的皇后位置, 则两个皇后 $i$ 和 $j$ 对角线攻击的条件为:

$$|p[i] - p[j]| \neq |i - j| \text{ 若该等式对任意一对 } (i, j) \text{ 均不成立, 则目标达成。}$$

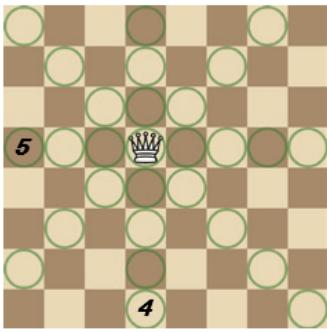


Figure 1: Une dame et les cases qu'elle menace.

FIGURE 1 – 4ème dame :  $p[4] = 5$

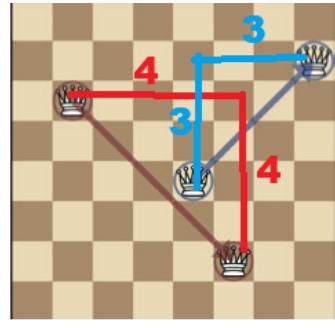


Figure 2: En rouge, deux dames se menacent sur une diagonale SE et en bleu, deux sur une diagonale NE.

FIGURE 2 – Diagonale :  $|p[i] - p[j]| = |i - j|$

## 2. Arbre de recherche (2 premiers niveaux)

Si l'on considère la modélisation par *permutations* (on s'interdit uniquement de mettre deux dames sur la même ligne pour simplifier la visualisation de l'arbre, les diagonales étant vérifiées ensuite) :

- **Niveau 0 (Racine)** : État vide [].
- **Niveau 1** : On place la 1ère dame dans la colonne 1. Il y a **8 actions possibles** (lignes 1 à 8).  
→ États : [1], [2], [3], [4], [5], [6], [7], [8].
- **Niveau 2** : Pour chaque noeud du niveau 1, on place une dame dans la colonne 2. Comme on ne peut pas réutiliser la ligne déjà prise :
  - Pour le noeud [1], les fils sont : [1, 2], [1, 3], ..., [1, 8] (7 fils).
  - Pour le noeud [2], les fils sont : [2, 1], [2, 3], ..., [2, 8] (7 fils).
 Total au niveau 2 :  $8 \times 7 = 56$  noeuds.

## 3. Analyse de l'arbre

- **Profondeur ( $d$ )** : 8. (On s'arrête dès que les 8 colonnes sont remplies).
- **Facteur de branchage ( $b$ )** : Il n'est pas constant.
  - À la racine :  $b = 8$ .
  - Au niveau 1 :  $b = 7$ .
  - Au niveau  $k$  :  $b = 8 - k$ .
- **Nombre de feuilles** : Correspond au nombre de permutations possibles des 8 lignes :

$$8! = 40\,320 \text{ feuilles.}$$

(Note : C'est beaucoup plus efficace que la modélisation naïve qui autorise n'importe quelle case, donnant  $64!/56! \approx 1.7 \times 10^{14}$  feuilles).

- **Nombre d'états total** : La somme des arrangements :  $\sum_{k=0}^8 \frac{8!}{(8-k)!} = 69\,281$  états.

## 2 TD2 : Recherche non informée

**Sujet** : Implémentation et comparaison de DFS et de BFS pour le jeu du taquin.

Voir le notebook Jupyter de correction (Python) disponible sur l'espace Moodle du cours pour le code source et les courbes de performance.

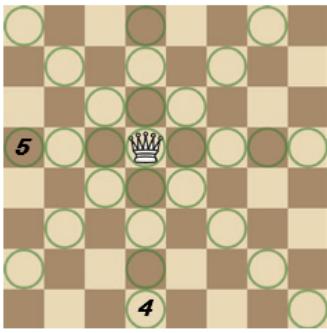


Figure 1: Une dame et les cases qu'elle menace.

图1 - 第4位女士:  $p[4] = 5$

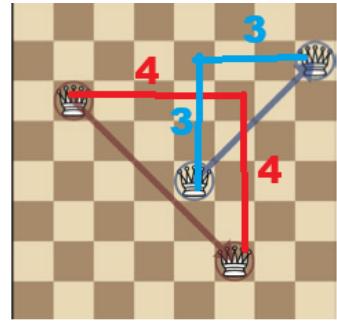


Figure 2: En rouge, deux dames se menacent sur une diagonale SE et en bleu, deux sur une diagonale NE.

图2 - 对角线:  $|p[i] - p[j]| = |i - j|$

## 2. 检索树（前两个层级）

若采用排列组合建模法（为简化树状结构可视化，仅禁止将两个皇后棋子置于同一条线上，后续将对对角线进行验证）：

—**级别0（根）**：空状态 []。

—**第1级**：将第一夫人放入第1列。共有**8种可能的行动**（第1至8行）。

→状态: [1]、[2]、[3]、[4]、[5]、[6]、[7]、[8]。

—**第二层**：对于第一层的每个节点，需在第二列放置一个棋子。由于已占用的行无法重复使用：

→节点[1]的子节点为: [1,2]、[1,3]、...、[1,8]（共7个子节点）。

→节点[2]的子节点为: [2,1]、[2,3]、...、[2,8]（共7个子节点）。

第2级总计:  $8 \times 7 = 56$ 个节点。

## 3. 树分析

—**深度 ( $d$ ) : 8**。（当8列填满时停止）。

—**分支因子 ( $b$ )**：不是常数。

— 根部:  $b=8$ 。

— 第1级:  $b=7$ 。

— 在 $k$ 层面上:  $b=8-k$ 。

—**叶片数量**：对应8行排列的可能组合数：

$$8! = 40320 \text{ 张纸。}$$

（注：这比允许任意单元格的朴素建模高效得多，可生成 $64! / 56! \approx 1.7 \times 10^{14}$ 张纸）。

—**总状态数**：排列的总和:  $P_{8k=0}^{8!} = 69281$ 个州。

## 2 TD2: 非知情研究

**主题：**棋盘游戏DFS与BFS的实现与比较

请查阅课程Moodle空间中提供的Python Jupyter 调试笔记本，其中包含源代码和性能曲线。