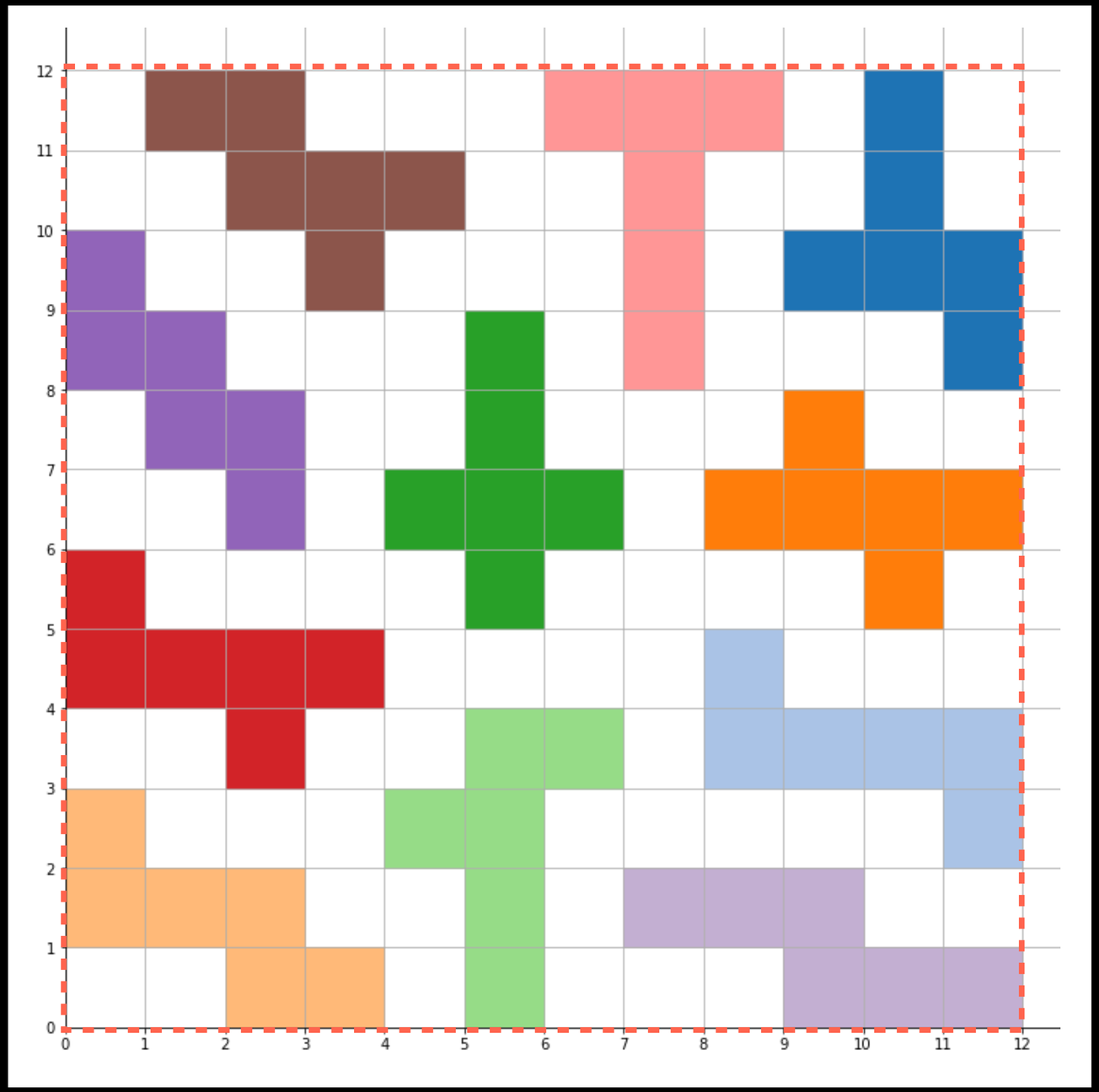
利用线性规划求解“拼图嵌挤”问题

我建模并利用程序解决了“Untouchable 11”问题!

by 极客鸭鸭 aka the Friedbees

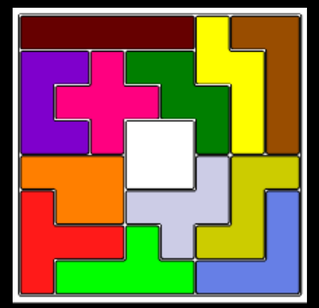


提纲&时间线
Outline & Timeline

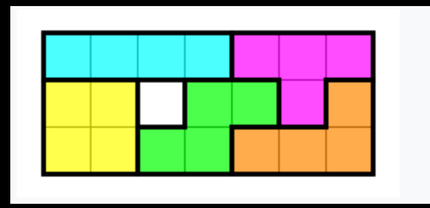
- 问题概述与简介
- 线性规划建模
- 线性规划求解
- 图解定义，深度优先搜索/回溯法（DFS/Backtracking）算法的思路
- Part 2 附录：线性规划，线性代数简介与更多线性规划建模的例子

“拼图嵌挤”问题介绍
Intro about the problem

- 很常见的一种Puzzle
- 早期研究案例：Tetromino/Pentomino/Polyominoes（四连方/五连方/多连方）
  - Tetris-（俄罗斯方块，四连方）由四个方块连接组成的图形，共五种可能（不考虑镜像）
  - -mino 骨牌，如domino（多米诺骨牌）
- “嵌挤问题”：给定有限的地块(tile)/棋盘，是否能够将所有给定的连方放入地形之内
- Polyominoes: Tetromino(4), Pentomino(5), etc.



Pentomino
五连方骨牌“嵌挤”问题

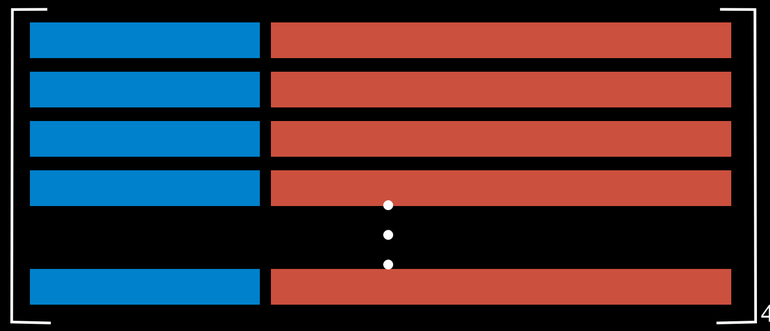


四连方“嵌挤”问题

利用数组表示一个图形的放置方案（Pattern）

Use array of numbers to model plan (pattern)

- 数组表示:
  - [4 7 8 9 11]
- 0-1长数组精确表示:
  - [0 0 0 1 0 0 1 1 1 0 1 0, …, 0]
- 穷举所有可行解(All possible patterns)：
  - 矩阵 A =



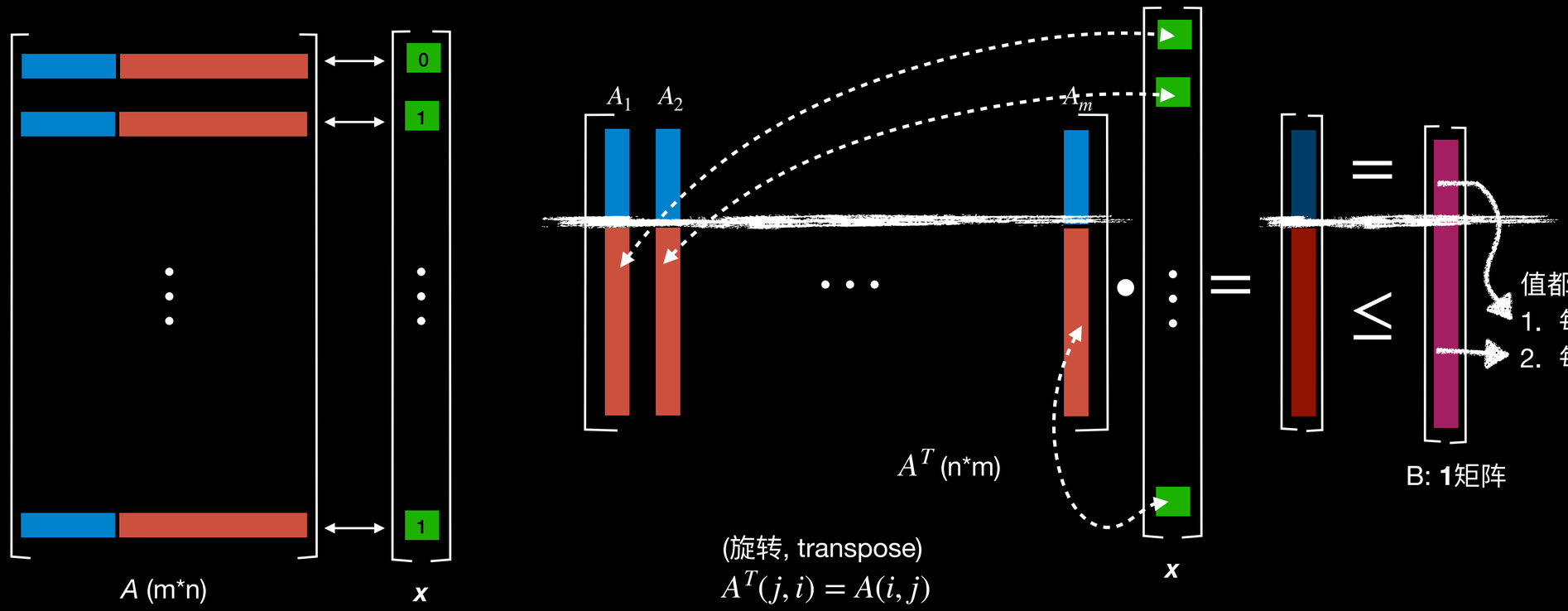
6	7	8	9
10	11	12	13
14	15	16	17

棋盘/地块

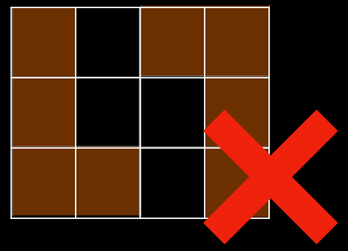
图块编号1-5

决策变量，限定条件
Decision variables, Constraints

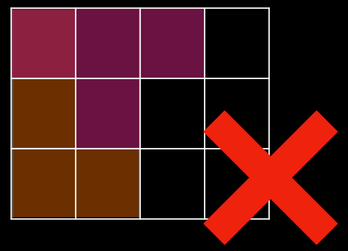
6	7	8	9
10	11	12	13
14	15	16	17



值都为1的矩阵：
1. 每一种蓝色对应的图形必须使用一次
2. 每一种红色对应的地块最多使用一次



反例1



反例2

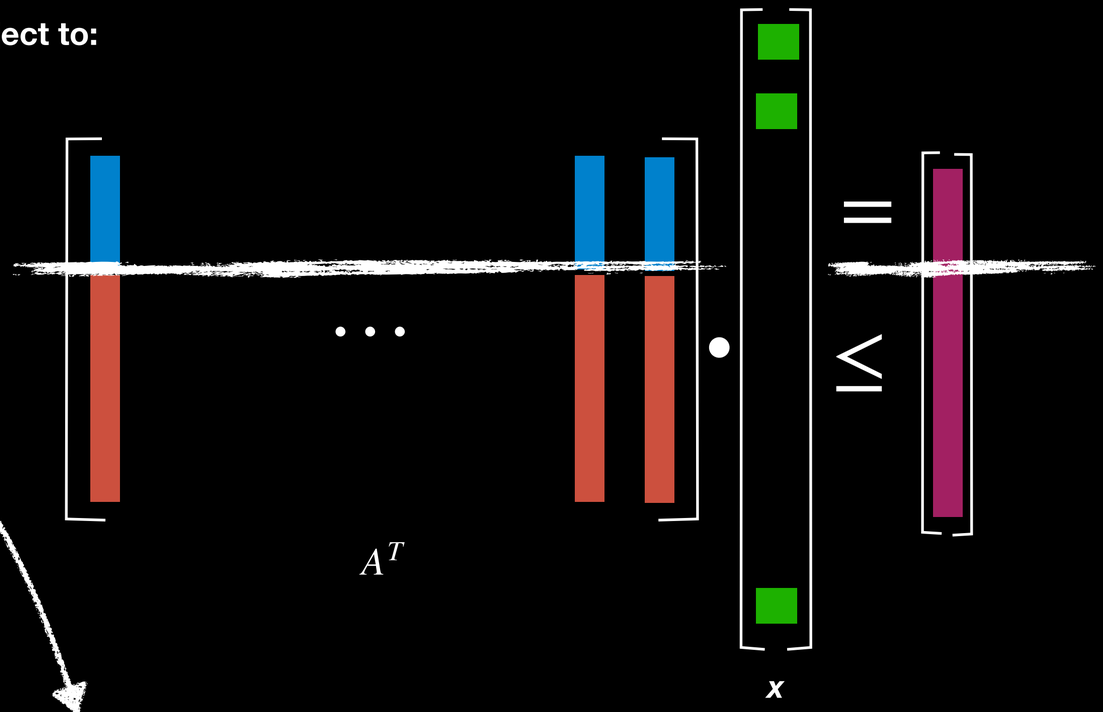
- 决策变量 (Decision Variables)：一个可行解包含 【是/否】 使用每一种模式
  - 任意模式（A中每一行A_i），对应一个决策变量（x_i=0或1）
- 构建限制条件 (Constraints)

线性规划/线性优化
Linear Programming/ Linear Optimization

- 目标函数 (Objective Function)
- 变量的限制条件 (Constraints)
  - **x**：决策变量限制条件
- 直接使用现成的线性规划求解器（LP Solver）求解， 如Gurobipy(商业软件但是对学术开放)

Minimize: $0^T \cdot x = 0$

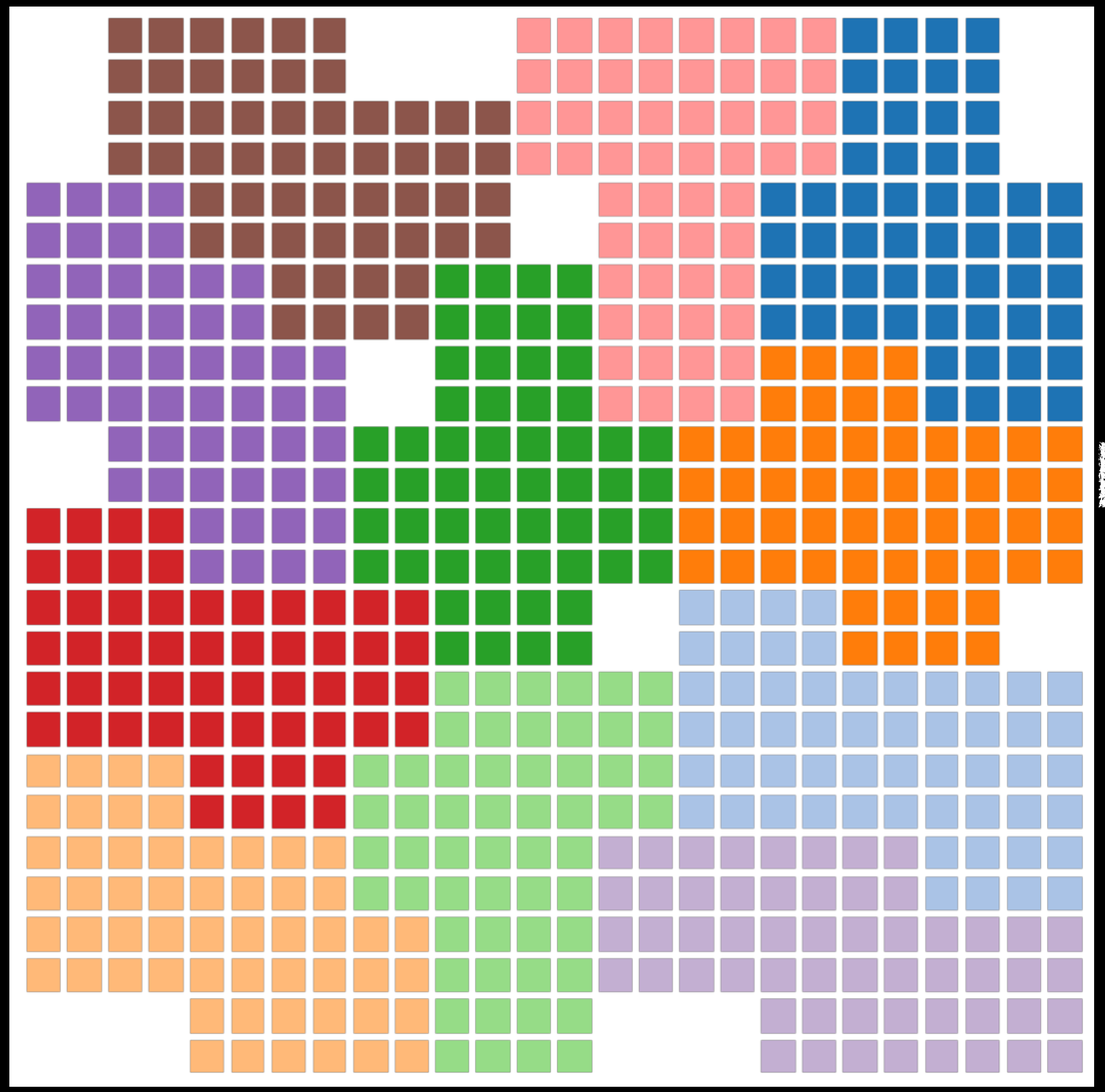
Subject to:



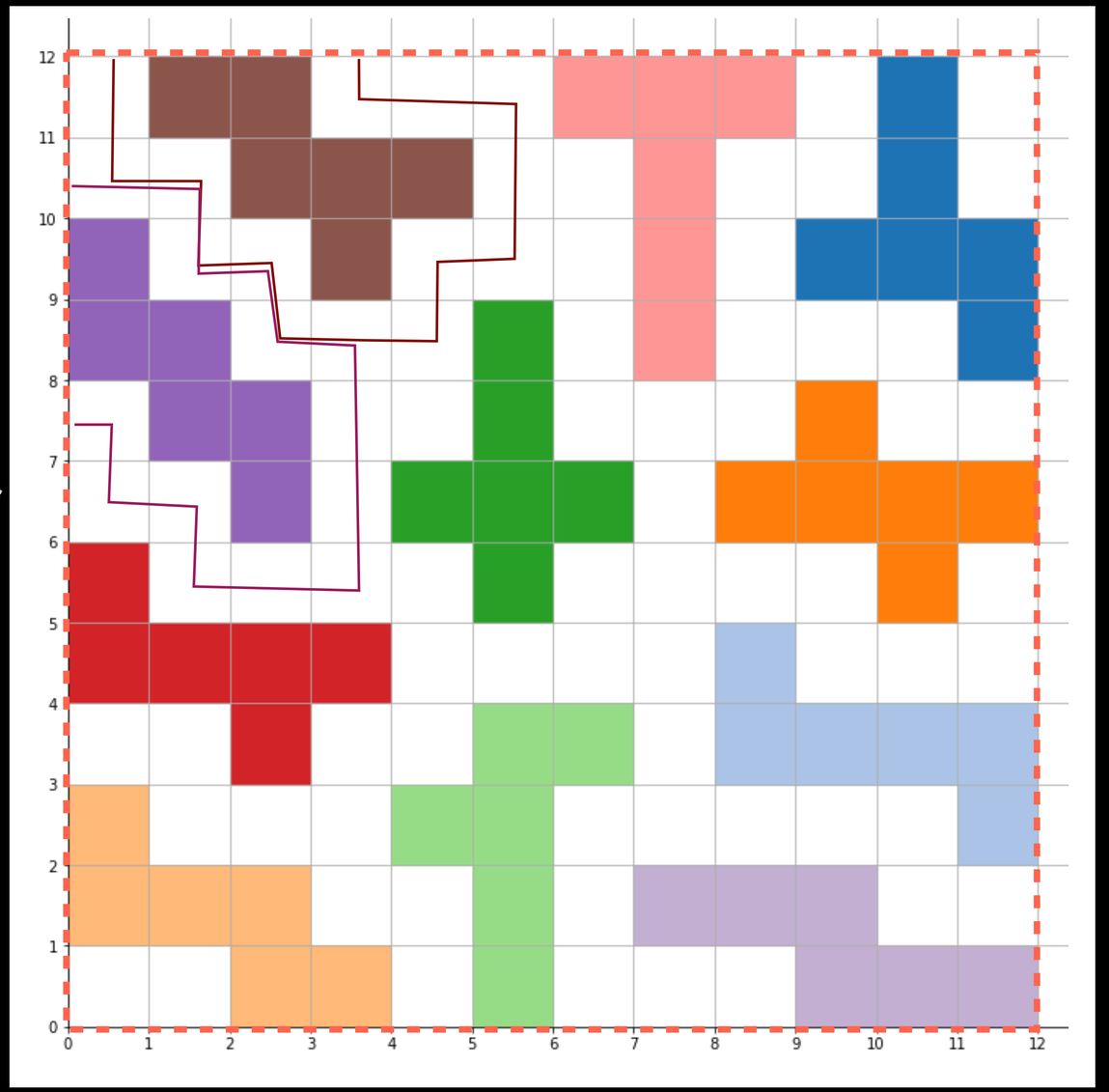
$$\boldsymbol{x} \geq 0, x_i \in \{0,1\}, \forall i$$

Untouchable 11建模

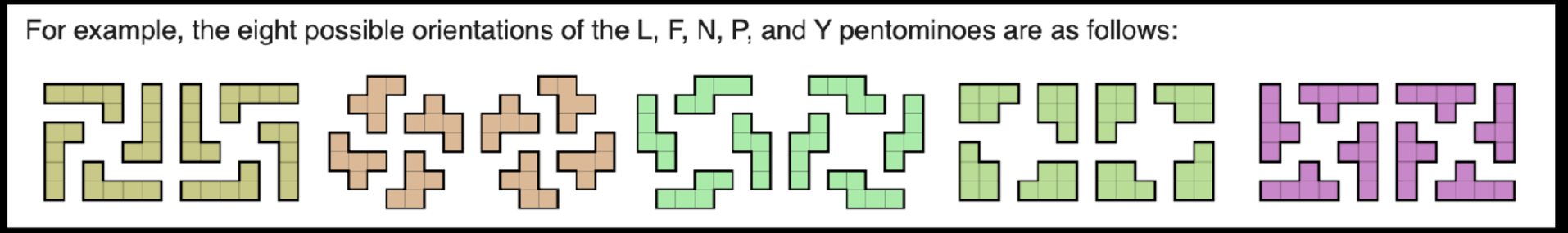
原图形往外扩展半个格即可



我找到的解（程序求解运行仅用<10秒钟）



感悟
Thoughts

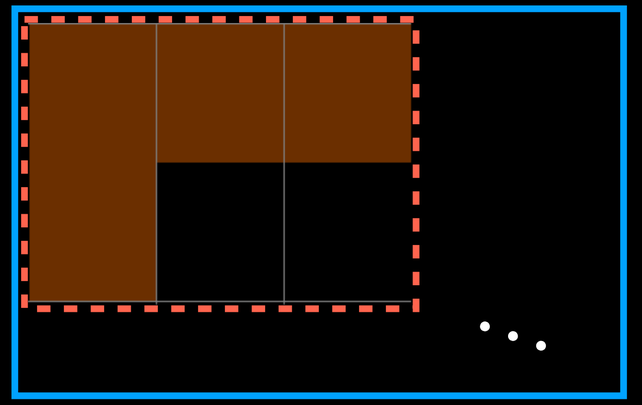


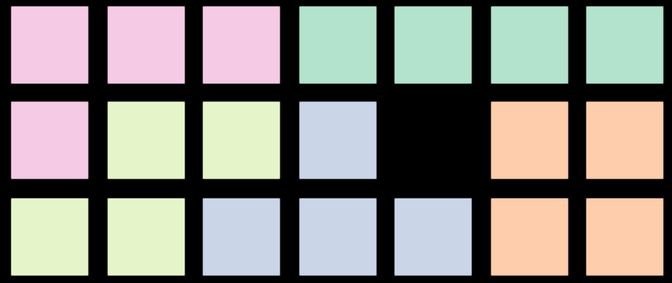
编程难点：

- 对每个连方列举全部可能的八种翻转（镜像翻转+“翻滚”）
- 找出对应的“最小长方体”列举位置
- 先研究本质相同但是结构简单的问题

本质：从所有模式中选择正确的模式组合，以满足限定条

- Untouchable 11所有的不重复模式（矩阵A^T的列数）： 5392
- 因为只有11个连方，最多选定11个列，对应11个不同的连方





先解决简化版问题（a toy/mini example）

深度优先搜索算法/回溯法

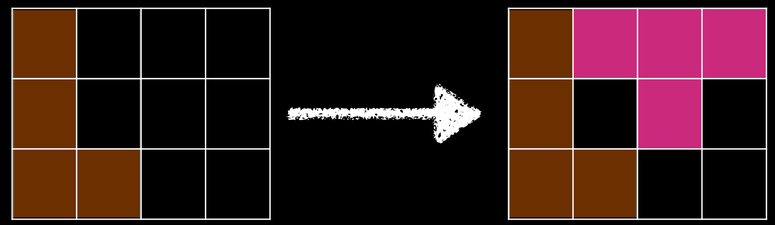
Depth first search (DFS)/ Backtracking

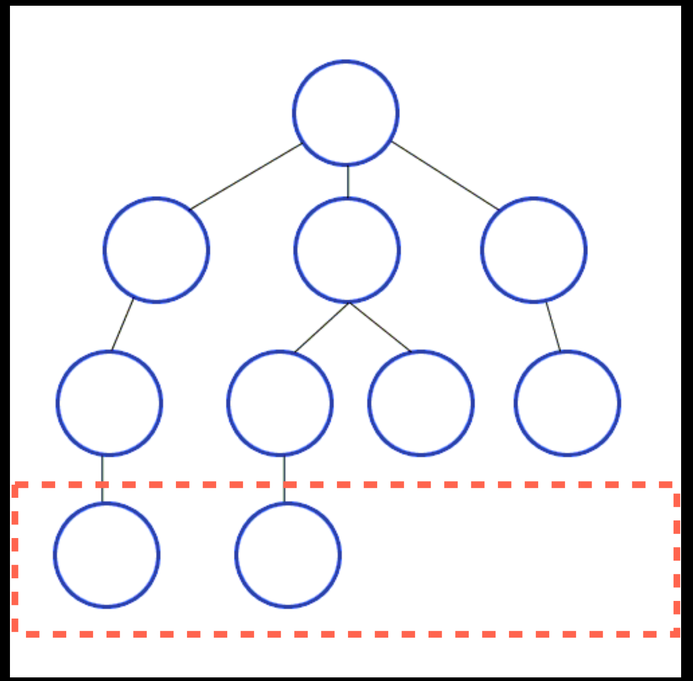
- 图结构：搜索树（Searching tree）
- 构建搜索树：

  根Root（搜索起始点）：即为空棋盘

  从部分解开始寻找最终解

  有向线段 directed link：





本问题中：

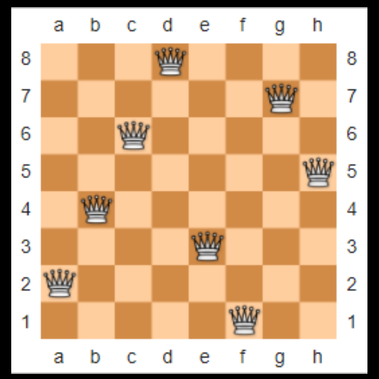
- 每一层增加一个模式（连方）
- 最底层全部11个模式/连方都输入即为可行解

- 通过DFS/回溯法穷举搜索，我们可以找到所有的可行解

  类比：用{🧶，✏️}标记迷宫

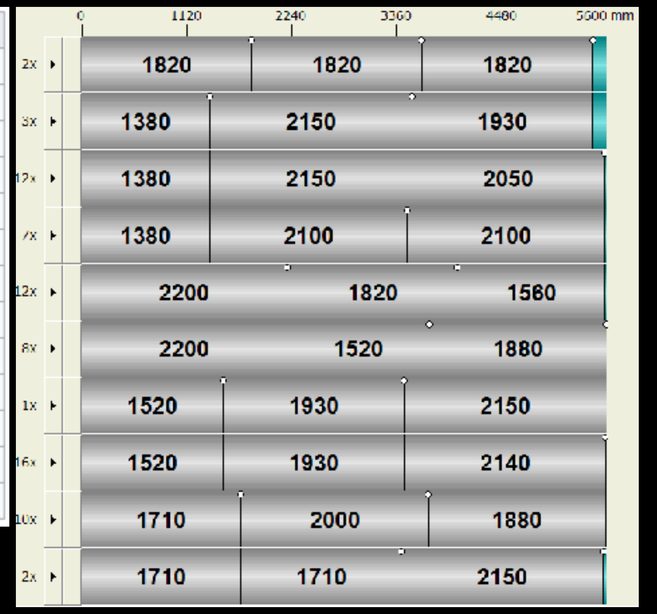
其他类似的问题
Other Similar Problems

- ♟国际象棋皇后问题(n queen problem)
- 🧻工厂裁剪纸张问题（The cutting stock problem）
  - 经典优化题目：建模中引入了模式Pattern的概念
  - LP implementation topics: (Delayed) column generation （理解为什么这个问题使用LP求解效率高）（例如：>5000模式选择11个模式）
- DFS (backtracking):
  - 解数独问题(Suduko)
  - 连方体问题拓展到三维

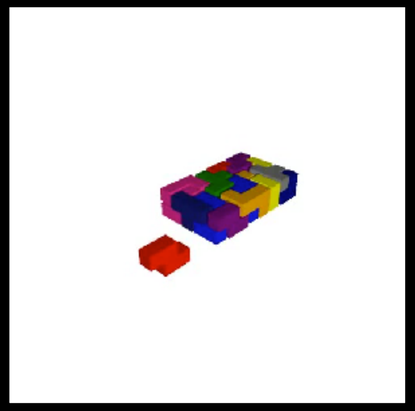


N queen problem

Width	#Items
1380	22
1520	25
1560	12
1710	14
1820	18
1880	18
1930	20
2000	10
2050	12
2100	14
2140	16
2150	18
2200	20



The cutting stock problem
(一个最优解) 模式的运用


Backtracking求解三维连方

求解数独

参考资料
References

- Untouchable 11论文：https://www.gathering4gardner.org/g4g13gift/puzzles/HoffCarl-GiftExchange-FromUntouchable11toHazmatCargo-G4G13.pdf
- Dancing Links论文：https://arxiv.org/pdf/cs/0011047.pdf
- 五连方: https://en.wikipedia.org/wiki/Pentomino
- 四连方: https://en.wikipedia.org/wiki/Tetromino
- 深度优先搜索：https://en.wikipedia.org/wiki/Depth-first_search
- 象棋皇后问题：https://en.wikipedia.org/wiki/Eight_queens_puzzle
- 工厂裁纸问题：https://en.wikipedia.org/wiki/Cutting_stock_problem
- Gurobipy：https://www.gurobi.com/documentation/9.1/quickstart_mac/cs_grbpy_the_gurobi_python.html
- Gurobi：API支持很多软件：C++, Java, Python, C, Matlab, R, Excel, …
  - 首页：https://www.gurobi.com/
  - 编程实例：https://www.gurobi.com/resource/functional-code-examples/