

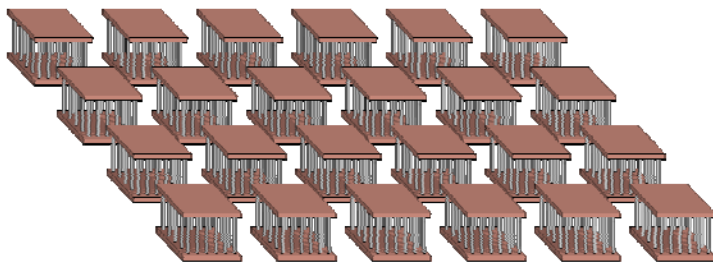


另一个分配问题例子

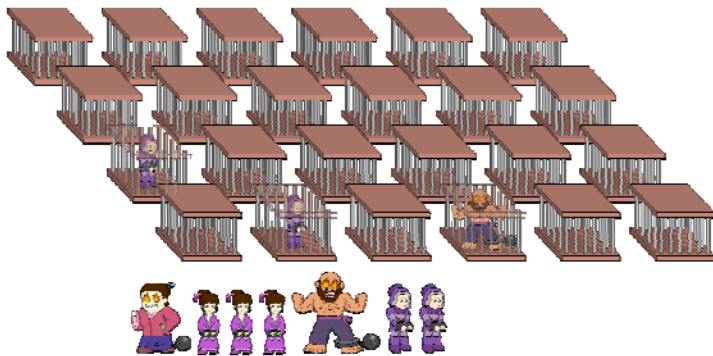
李浩文、彼得·斯塔基



牢房问题



同一个牢房不能有两个囚犯



3

危險囚犯



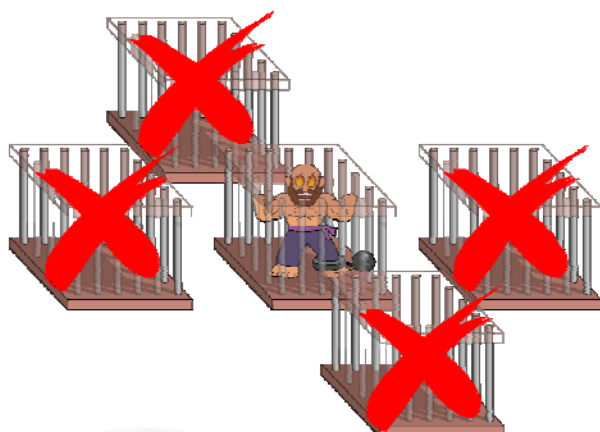
4

危險囚犯



5

危險囚犯

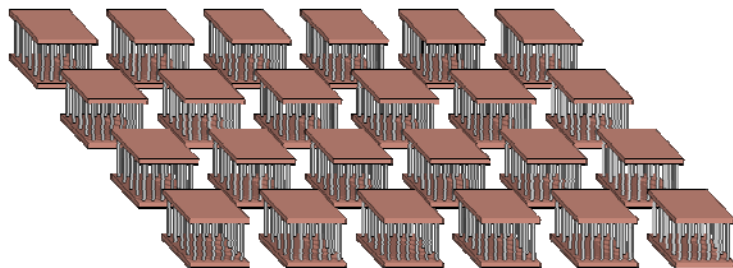


6

除非另有声明，本材料为©墨尔本大学与香港中文大学 版权所有。只能在个人参考、研究或学习的情况下，方可分享、复印或下载本材料。



男性与女性分隔开



7

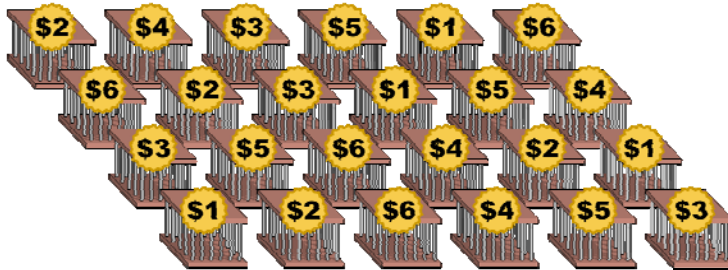
男性与女性分隔开



8

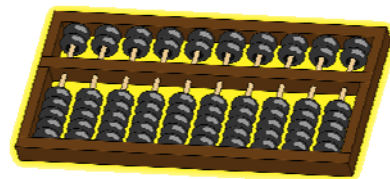
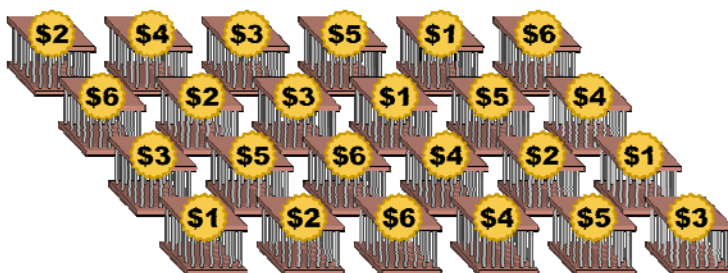


最小化费用



9

最小化费用



10

牢房问题的数据 (cellBlock.mzn)

```
enum PRISONER;  
  
int: n;  
set of int: ROW = 1..n;  
int: m;  
set of int: COL = 1..m;  
array[ROW,COL] of int: cost;  
  
set of PRISONER: danger;  
set of PRISONER: female;  
set of PRISONER: male = PRISONER diff female;
```

依赖参数声明

11

牢房问题的决策变量 (cellBlock.mzn)

定义域的对象是什么？

• DOM = PRISONER

值域的对象是什么？

• COD = ROW x COL

表示：两个函数

```
array[PRISONER] of var ROW: r;  
array[PRISONER] of var COL: c;
```

12

牢房问题的约束 (cellBlock.mzn)

同一个牢房不能有两个囚犯

```
forall(p1, p2 in PRISONER where p1 < p2)
  (abs(r[p1]-r[p2]) + abs(c[p1]-c[p2]) > 0);
```

难道我们不能使用alldifferent吗？

可以

```
alldifferent([r[p] * m + c[p] | p in PRISONER]);
```

将每个牢房映射到一个独有的数

危险囚犯不能有邻居

```
forall(p in PRISONER, d in DANGER where p != d)
  (abs(r[p] - r[d]) + abs(c[p] - c[d]) > 1);
```

13

牢房问题的约束 + 目标 (cellBlock.mzn)

性别约束

```
forall(p in female)(r[p] <= (n + 1) div 2);
forall(p in male)(r[p] >= n div 2 + 1);
```

注意male的使用

- 比使用定义的方式更清晰

目标函数

```
var int: tCost =
  sum(p in PRISONER)(cost[r[p],c[p]]);
solve minimize tCost;
```

14



求解模型

Female: P1 in cell[1,1]
Female: P6 in cell[2,4]
Female: P7 in cell[1,5]
Female: P10 in cell[2,2]

Male: P2 in cell[3,1]
Male: P3 in cell[4,6]
Male: P4 in cell[4,4]
Male: P5 in cell[4,2]
Male: P8 in cell[3,5]
Male: P9 in cell[4,1]

Total Cost: 21

15

小结

- 分配子问题在很多应用中都很常见
- 在以后一个关于刘备的问题中，也有一个分配子问题，但是那个函数是双射的
 - 这个就是匹配问题了

16



图像引用

所有图像由Marti Wong设计提供, © 香港中文大学与墨尔本大学 2016

17