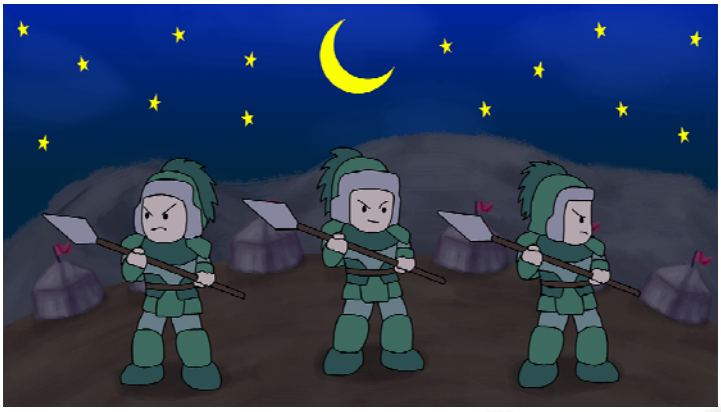
划分建模

李浩文、彼得·斯塔基

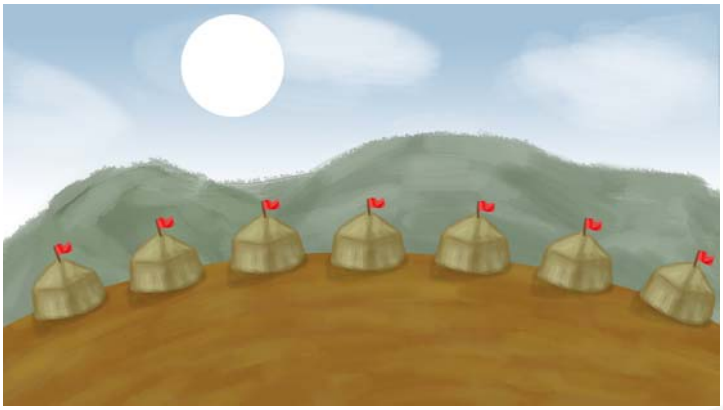
巡逻轮换问题

夜晚时需要三个士兵



3

白天不需要巡逻



4

傍晚时需要1个或2个士兵

5

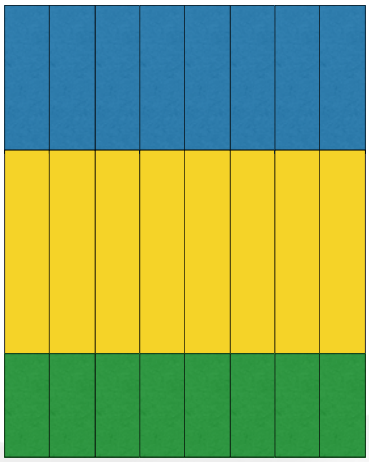
傍晚时需要1个或2个士兵

6

巡逻轮换问题

夜晚
傍晚
休息

1 2 3 4 5 6 7 8

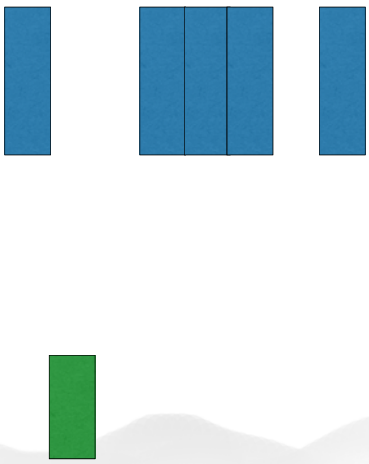


7

不能连续多于两个晚上值班

夜晚
傍晚
休息

1 2 3 4 5 6 7 8



8

不能连续多于两个晚上值班

夜晚
傍晚
休息

1 2 3 4 5 6 7 8

9

不能连续多于两个晚上值班

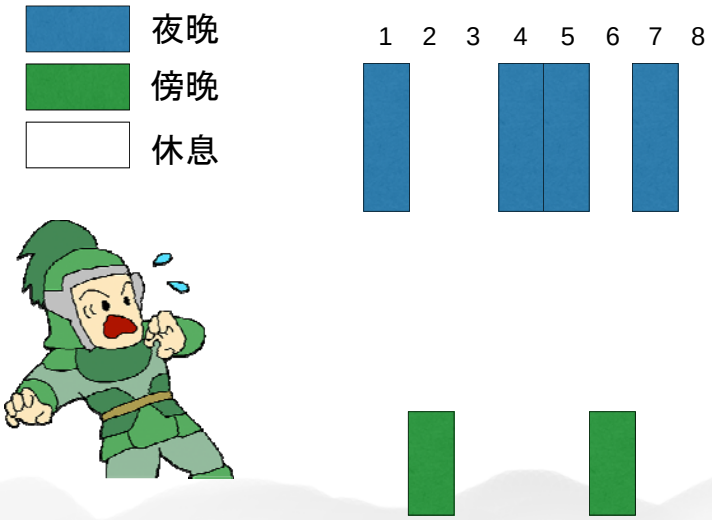
夜晚
傍晚
休息

1 2 3 4 5 6 7 8

X

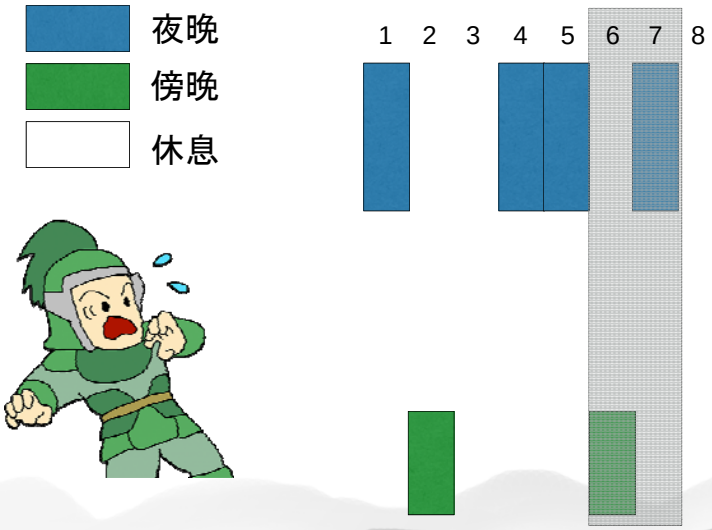
10

傍晚值班后夜晚不能再值班



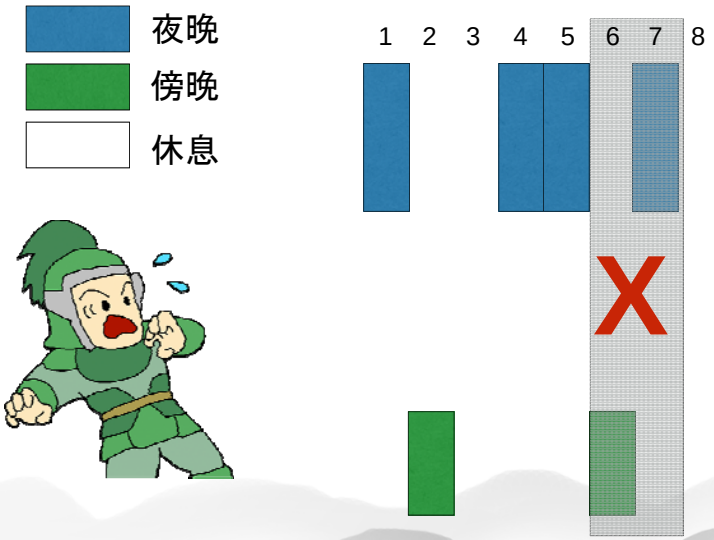
11

傍晚值班后夜晚不能再值班



12

傍晚值班后夜晚不能再值班



13

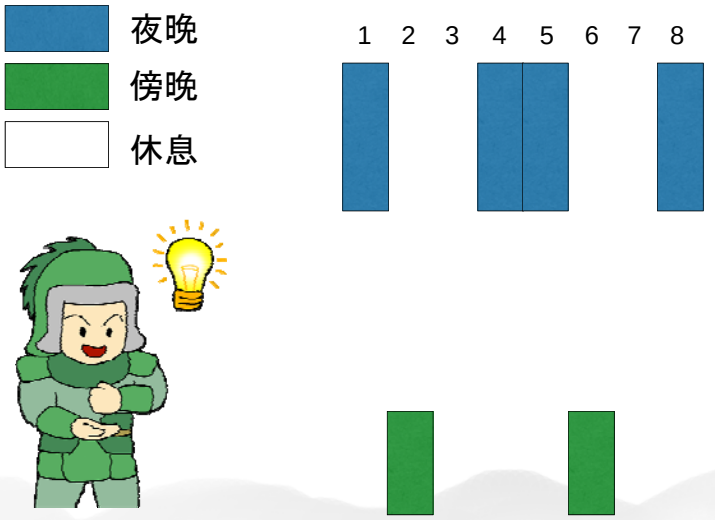
最大化傍晚值班的士兵数量

14

最大化傍晚值班的士兵数量

15

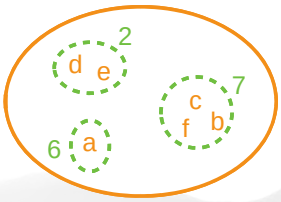
巡逻轮换问题



16

划分问题

- 确定一个函数 f: DOM ➔ COD
- 可以看成是一个划分问题
- 若有以下条件，这可以给我们额外的洞察
  - 我们想对集合进行约束或者操作
  - F(c) = { d in DOM | f(d) = c },
  - 其中 F: COD ➔ 2^{DOM} (f 的伪逆函数)



17

排班对象

- 定义我们想要推理的对象集合

```
enum SOLDIER;
enum SHIFT;
int: nDays;
set of int: DAY = 1..nDays;

int: o; % required number for NIGHT
int: l; % lower bound for EVE
int: u; % upper bound for EVE
```

18

排班决策变量

- 定义域中的对象是什么？
  - DOM = SOLDIER × DAY
- 值域中的对象是什么？
  - COD = SHIFT

- 对每一天每一个士兵，选择班次

```
array[SOLDIER,DAY] of var SHIFT: roster;
```

- 可以看成是一个划分问题
  - 根据班次类型来划分士兵
  - 对每个班次的士兵集合进行推理

19

排班模式约束

- 每一个士兵都不能连续多于两个晚上值班
- 我们如何来表达它？

```
forall(s in SOLDIER, k in 1..nDays-2)
   (sum(d in k..k+2)(roster[s,d] = NIGHT) <= 2);
```

- 呀, 不是很清楚啊!!
- 使用逻辑联结词来使得它更明确

```
forall(s in SOLDIER, d in 1..nDays-2)
   (roster[s,d] = NIGHT /\ roster[s,d+1] = NIGHT
      -> roster[s,d+2] != NIGHT);
```

- 每一个士兵都不能傍晚值班后夜晚再值班

```
forall(s in SOLDIER, d in 1..nDays-1)
   (roster[s,d] = EVE -> roster[s,d+1] != NIGHT);
```

20

逻辑联结词

- 布尔表达式
  - true
  - false
  - /\ 合取
  - \/ 析取
  - -> 蕴含
  - <-> 双蕴含或者等价
  - not 非
- 允许我们更有效地结合约束
  - 但是结合逻辑约束和全局约束时要注意！后面会有更多介绍 ...

21

排班需求约束

- 每个夜晚有o个士兵值班

```
forall(d in DAY)
   (sum(s in SOLDIER)
      (roster[s,d] = NIGHT) = o);
```

- 每个傍晚有l到u个士兵值班

```
forall(d in DAY)
   (sum(s in SOLDIER)
      (roster[s,d] = EVE) >= l);
forall(d in DAY)
   (sum(s in SOLDIER)
      (roster[s,d] = EVE) <= u);
```

22

目标函数

- **傍晚**时越多士兵巡逻越好

```
var int: tOnEve = sum(d in DAY)
   (sum(s in SOLDIER)(roster[s,d] = EVE));
solve maximize (tOnEve);
```

23

巡逻模型 (patrolV1.mzn)

```
enum SOLDIER;
enum SHIFT;
int: nDays;
set of int: DAY = 1..nDays;
int: o;
int: l;
int: u;

array[SOLDIER,DAY] of var SHIFT: roster;

constraint forall(d in 1..(nDays-2),
   s in SOLDIER)((roster[s,d] = NIGHT) /\
      (roster[s,d+1] = NIGHT)
   -> (roster[s,d+2] != NIGHT));
constraint forall(d in 1..(nDays-1),
   s in SOLDIER)((roster[s,d] = EVE) ->
      (roster[s,d+1] != NIGHT));
```

24

巡逻模型 (patrolV1.mzn)

```
constraint forall(d in DAY)(sum (s in SOLDIER)
   ((roster[s,d] = NIGHT)) = o);
constraint forall(d in DAY)(sum (s in SOLDIER)
   ((roster[s,d] = EVE)) >= l);
constraint forall(d in DAY)(sum (s in SOLDIER)
   ((roster[s,d] = EVE)) <= u);

var int: tOnEve = sum(d in DAY)
   (sum(s in SOLDIER)(roster[s,d] = EVE));
solve maximize (tOnEve);
```

25

求解模型

- **求解 … 求解 … 求解 …**
- 5分钟后还是没有任何结果
- **在10.5分**时终于得到了一个结果
- **更改天数**为6，**只需要7秒**这个模型就可以找到**一个最**优解

哪个地方不对？

26

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

27