



# 第二个模型

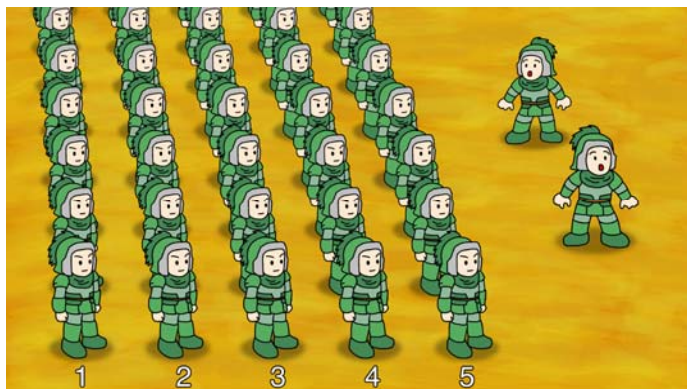
李浩文、彼得·斯塔基



## 艰苦训练的弊端

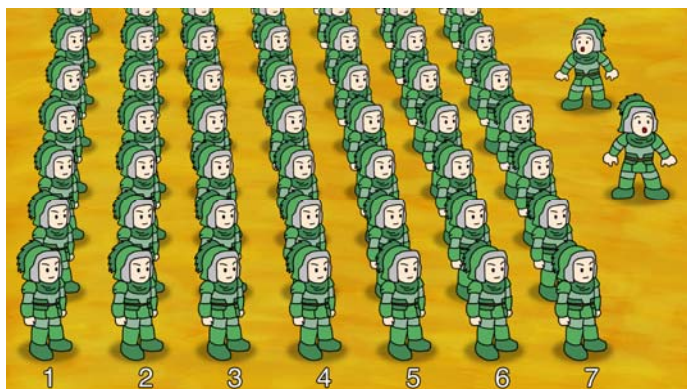


## 5列



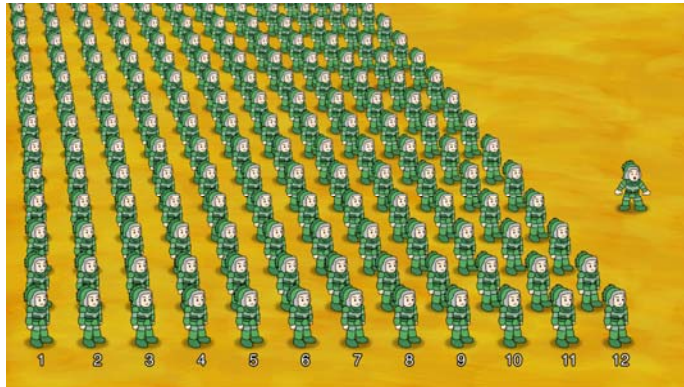
3

## 7列



4

## 12列



5

## 点兵问题



6

## 点兵问题 (count.mzn)

```
var 100..800: army;  
  
constraint army mod 5 = 2;  
constraint army mod 7 = 2;  
constraint army mod 12 = 1;
```

```
solve satisfy;
```

没有目标

没有输出

7

## 点兵问题

- 我们可以运行如下命令求解MiniZinc模型

```
$ minizinc count.mzn
```

- 运行结果为

```
army = 457;
```

```
-----
```

- 直线 ----- 标示解

- 直线 ===== 未出现指明或许还有其他解

8

## 点兵问题

- 我们可以运行如下命令求解MiniZinc模型

```
$ minizinc -all-solutions count.mzn
```

- 运行结果为

```
army = 457;
```

```
-----
```

```
=====
```

- 直线 ----- 标示解

- 直线 ===== 标示没有更好的解了。士兵数量即是457

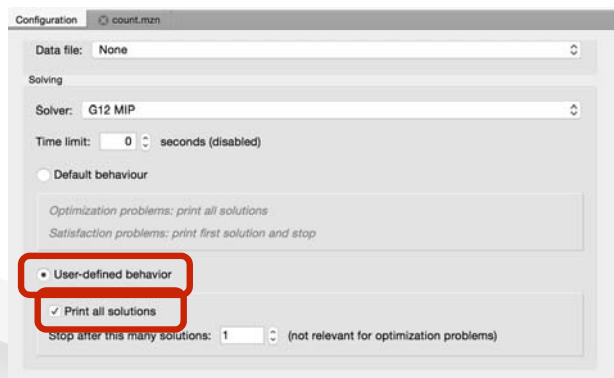
- 可以从IDE的Configurations处设置得到所有的解

- 对于优化问题，这是默认设置

9

## 点兵问题

- 在IDE中求解一个满足问题，求出所有的解
  - 在Configuration页
  - 设置user-defined behaviour
    - print all solutions



10





## 默认输出

- 注意 `count.mzn` 没有输出项
- 默认情况下，MiniZinc输出
  - 所有声明的变量
  - 且没有被表达式赋值
- 我们会在简单的模型中利用这个特性

11

## 中国剩余定理

- 点兵问题的约束可以利用用于解决同余方程组的**中国剩余定理**直接解决
- 一种代数解法（如下所示）
  - (A)  $\text{army} = 5t + 2 \pmod{5}$
  - (B)  $\text{army} = 7u + 2 \pmod{7}$
  - (C)  $\text{army} = 12v + 1 \pmod{12}$
  - $5t + 2 = 2 \pmod{7}$  (将 A 代入 B)
  - (D)  $t = 0 \pmod{7}$  即  $t = 7w$  (化简)
  - (E)  $\text{army} = 2 + 35w$  (将 D 代入 A)
  - $2 + 35w = 1 \pmod{12}$  (将 E 代入 C)
  - (F)  $w = 1 \pmod{12}$  即  $w = 1 + 12x$  (化简)
  - $\text{army} = 37 + 420x$  (将 F 代入 E)

12



## 小结

- 满足问题
  - 不需要求得一个最优解
- 约束未必只能是线性等式和不等式
  - 取模 (mod), 乘法, 除法
  - 非等式 ( $\neq$ )
  - 之后会有更复杂的约束！

13

## 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

14