



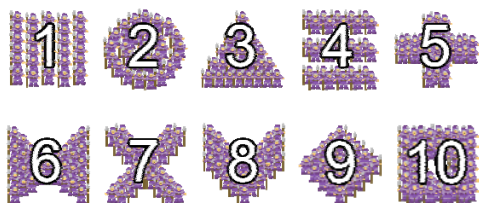
# 集合表示的选择

李浩文、彼得·斯塔基



## 八卦问题的分析

☳ 10个攻击点

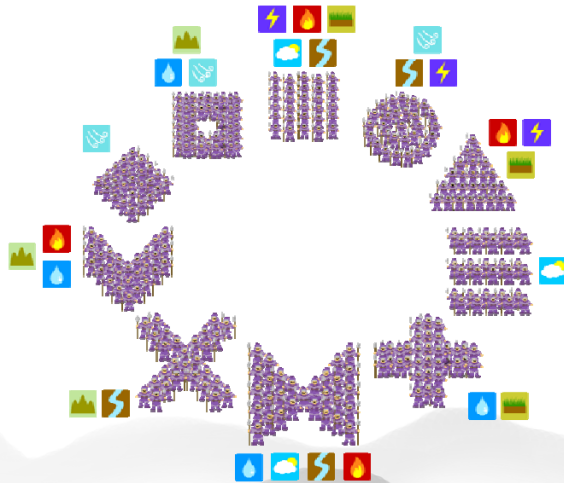


☳ 8种八卦属性



## 八卦问题的分析

### 攻击点具有八卦属性



3

## 八卦问题的分析

### 攻击点具有八卦属性

							
☳	☲	☴	☵	☶	☱	☴	☷
1	2	3	4	5	6	7	8
4 6	6 7	6 8	2 3	9 10	8 10	8 10	3 5

4

## 八卦问题的分析

- 一旦一个与某个属性相关联的攻击点被攻击，**所有**其他具有此属性的攻击点都会被**强化**。因此，对于**每个属性**，**只有一个攻击点**可以被攻击



5

## 八卦问题的分析

- 有些攻击点会**比较弱**！

1	2	3	4	5	6	7	8	9	10
10	8	4	2	6	9	5	3	8	10

- 目标**: 产生最大的伤害！
- 换句话说，选择所有攻击点的一个子集去攻击
  - 每个属性最多有一个攻击点
  - 最大化伤害

6

## 集合选择问题 (bagua-10-8.dzn)

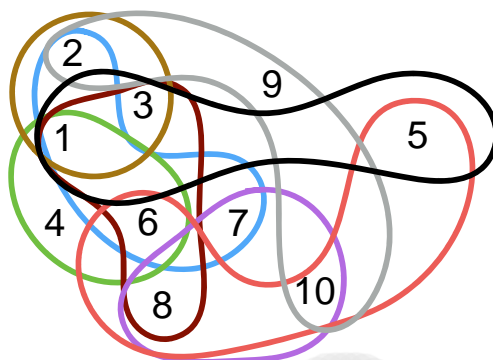
- 每个SYMB中的属性，给定一个数字  
1..nSpots的子集。选择一个1..nSpots  
的子集，使得每个属性的子集中最多有一个  
元素在其中，并且最大化选择的集合的伤害  
值

```
nSpots = 10;  
damage = [10, 8, 4, 2, 6, 9, 5, 3, 8, 10];  
SYMB = {'天', '泽', '火', '雷', '风', '水', '山', '地'};  
group = [{1,4,6}, {1,2,6,7}, {1,3,6,8}, {1,2,3},  
         {2,9,10}, {5,6,8,10}, {7,8,10}, {1,3,5}];
```

7

## 简单集合选择算法

- 贪心算法
  - 选择伤害最大的可选元素
  - 去掉不再有效的选择



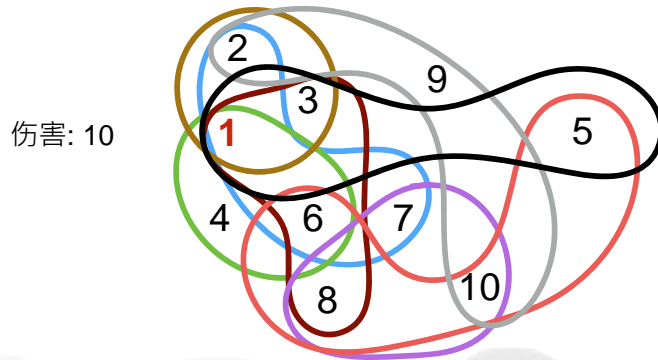
```
damage = [10, 8, 4, 2, 6, 9, 5, 3, 8, 10];  
group = [{1,4,6}, {1,2,6,7}, {1,3,6,8}, {1,2,3},  
         {2,9,10}, {5,6,8,10}, {7,8,10}, {1,3,5}];
```

8

## 简单集合选择算法

### 贪心算法

- 选择伤害最大的可选元素
- 去掉不再有效的选择



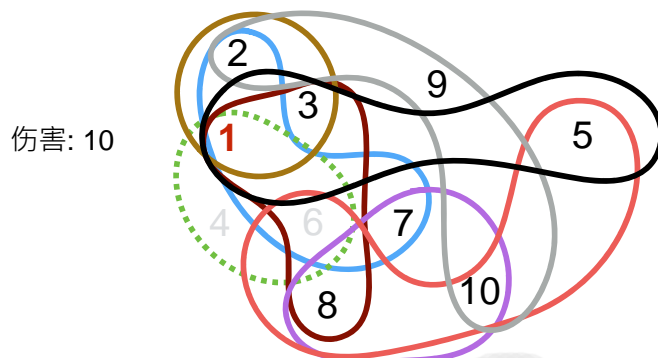
```
damage = [10, 8, 4, 2, 6, 9, 5, 3, 8, 10];  
group = [{1, 4, 6}, {1, 2, 6, 7}, {1, 3, 6, 8}, {1, 2, 3},  
         {2, 9, 10}, {5, 6, 8, 10}, {7, 8, 10}, {1, 3, 5}];
```

9

## 简单集合选择算法

### 贪心算法

- 选择伤害最大的可选元素
- 去掉不再有效的选择



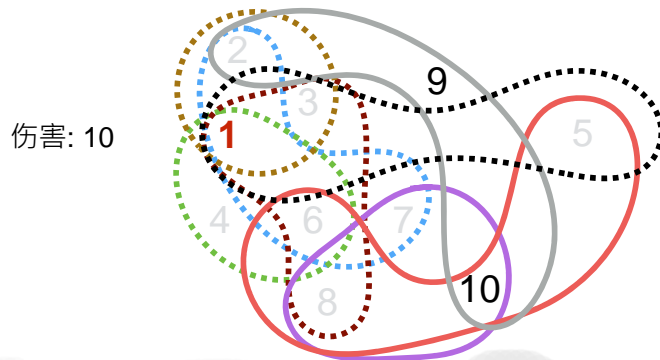
```
damage = [10, 8, 4, 2, 6, 9, 5, 3, 8, 10];  
group = [{1, 4, 6}, {1, 2, 6, 7}, {1, 3, 6, 8}, {1, 2, 3},  
         {2, 9, 10}, {5, 6, 8, 10}, {7, 8, 10}, {1, 3, 5}];
```

10

## 简单集合选择算法

### 贪心算法

- 选择伤害最大的可选元素
- 去掉不再有效的选择



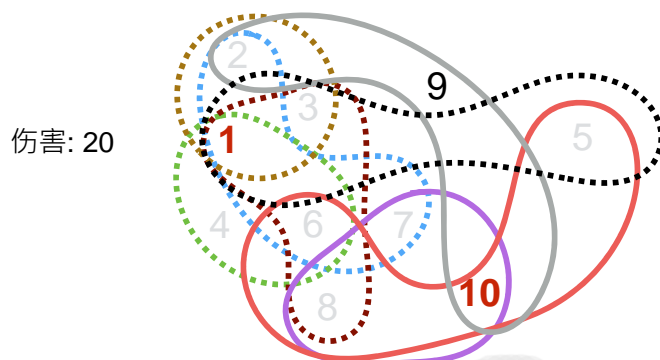
```
damage = [10, 8, 4, 2, 6, 9, 5, 3, 8, 10];  
group = [{1,4,6}, {1,2,6,7}, {1,3,6,8}, {1,2,3},  
         {2,9,10}, {5,6,8,10}, {7,8,10}, {1,3,5}];
```

11

## 简单集合选择算法

### 贪心算法

- 选择伤害最大的可选元素
- 去掉不再有效的选择



```
damage = [10, 8, 4, 2, 6, 9, 5, 3, 8, 10];  
group = [{1,4,6}, {1,2,6,7}, {1,3,6,8}, {1,2,3},  
         {2,9,10}, {5,6,8,10}, {7,8,10}, {1,3,5}];
```

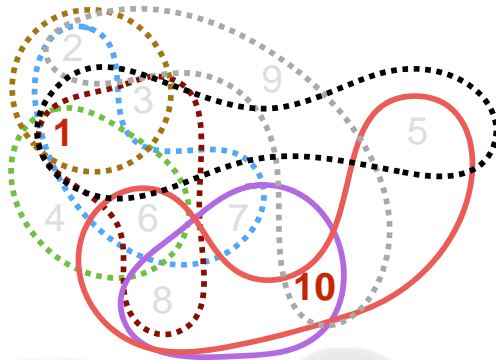
12

## 简单集合选择算法

### 贪心算法

- 选择伤害最大的可选元素
- 去掉不再有效的选择

伤害: 20



```
damage = [10, 8, 4, 2, 6, 9, 5, 3, 8, 10];  
group = [{1,4,6}, {1,2,6,7}, {1,3,6,8}, {1,2,3},  
         {2,9,10}, {5,6,8,10}, {7,8,10}, {1,3,5}];
```

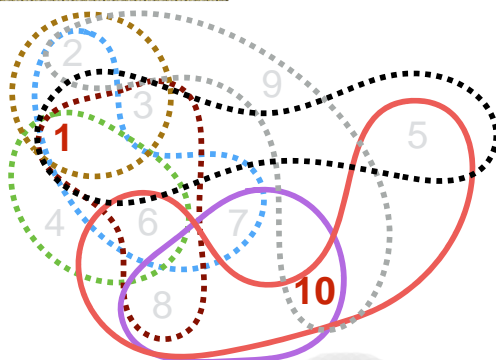
13

## 简单集合选择算法

### 贪心算法

- 选择伤害最大的可选元素
- 去掉不再有效的选择

伤害: 20  
**{1,10}**



```
damage = [10, 8, 4, 2, 6, 9, 5, 3, 8, 10];  
group = [{1,4,6}, {1,2,6,7}, {1,3,6,8}, {1,2,3},  
         {2,9,10}, {5,6,8,10}, {7,8,10}, {1,3,5}];
```

14





## 八卦问题 数据 + 决策变量 (bagua.mzn)

### 数据

```
int: nSpots;  
set of int: SPOT = 1..nSpots;  
array[SPOT] of int: damage;  
  
enum SYMB;  
array[SYMB] of set of SPOT: group;
```

### 决策变量

```
var set of SPOT: attacks;
```

15

## 集合选择约束 + 目标 (bagua.mzn)

### 交集最多只有一个元素

```
forall(s in SYMB)  
  (card(attacks intersect group[s])  
   <= 1);
```

### 目标

```
var int: totalDamages =  
  sum(p in attacks)(damage[p]);  
solve maximize (totalDamages);
```

16





## 求解模型

- 对模型求解

attacks: {4,5,7,9} & damages: 21;

- 比贪心算法好！

17

## 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

18