



THE UNIVERSITY OF  
MELBOURNE



香港中文大學  
The Chinese University of Hong Kong

# Choosing a Set Representation

Jimmy Lee & Peter Stuckey



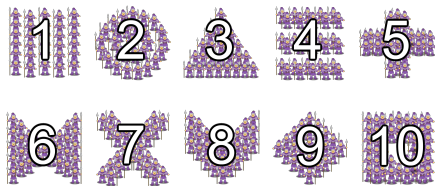
香港中文大學  
The Chinese University of Hong Kong



THE UNIVERSITY OF  
MELBOURNE

## Bagua Problem Analysis

- 10 possible attack spots



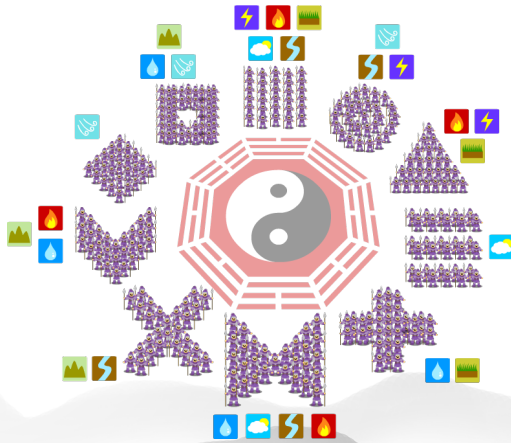
- 8 Bagua properties





## Bagua Problem Analysis

- Spots are backed up by symbols



3

## Bagua Problem Analysis

- Spots are backed up by symbols

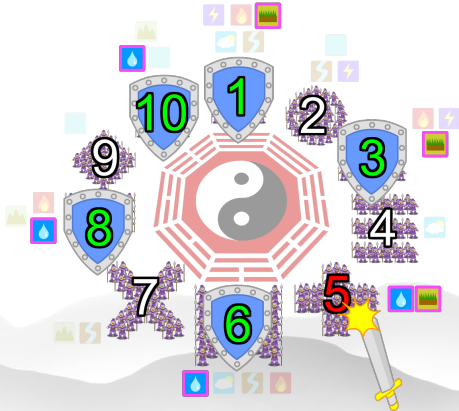
☰ 4 6	☷ 2 8	☲ 3 9	☵ 1 7	☴ 5 10	☳ 6 11	☱ 7 12	☶ 8 13

4



## Bagua Problem Analysis

- Once a spot associated with a symbol is attacked, **all** other spots supported by the same symbol will be **reinforced**. Thus, **only one spot of a symbol** can be attacked



5

## Bagua Problem Analysis

- Some spots are **weaker**!

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

- Goal:** produce the biggest damage!
- In other words, pick a subset of the spots to attack
  - at most 1 spot for each symbol
  - maximize the damage

6



## SetSelect Question (bagua-10-8.dzn)

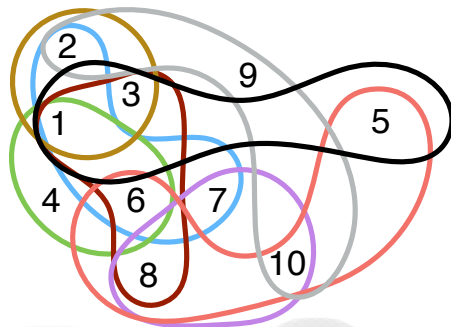
- Given a subset of numbers  $1..nSpots$  for each symbol in SYMB, choose a subset of  $1..nSpots$  which includes at most one from each subset and maximize the damage points of the chosen set

```
nSpots = 10;  
damage = [10, 8, 4, 2, 6, 9, 5, 3, 8, 10];  
SYMB = {'天', '澤', '火', '雷', '風', '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

## Simple Set Select Algorithm

- Greedy algorithm
  - choose the largest damage available element
  - eliminate choices that are no longer valid



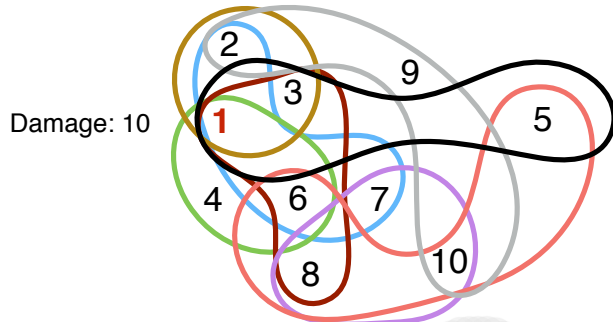
```
damage = [10, 8, 4, 2, 6, 9, 5, 3, 8, 10];  
group = [{1,4,6}, {1,2,6,7}, {1,3,6,8}, {1,2,3},  
         {2,9,10}, {5,6,8,10}, {7,8,10}, {1,3,5}];
```

8

## Simple Set Select Algorithm

### Greedy algorithm

- choose the largest damage available element
- eliminate choices that are no longer valid



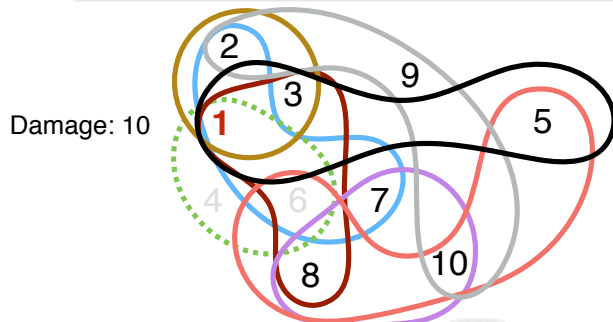
```
damage = [10, 8, 4, 2, 6, 9, 5, 3, 8, 10];
group = [{1,4,6}, {1,2,6,7}, {1,3,6,8}, {1,2,3},
        {2,9,10}, {5,6,8,10}, {7,8,10}, {1,3,5}];
```

9

## Simple Set Select Algorithm

### Greedy algorithm

- choose the largest damage available element
- eliminate choices that are no longer valid



```
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}];
```

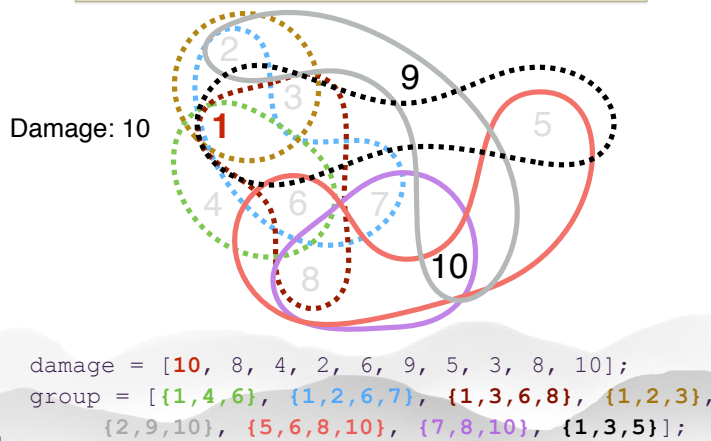
10



## Simple Set Select Algorithm

### Greedy algorithm

- choose the largest damage available element
- eliminate choices that are no longer valid

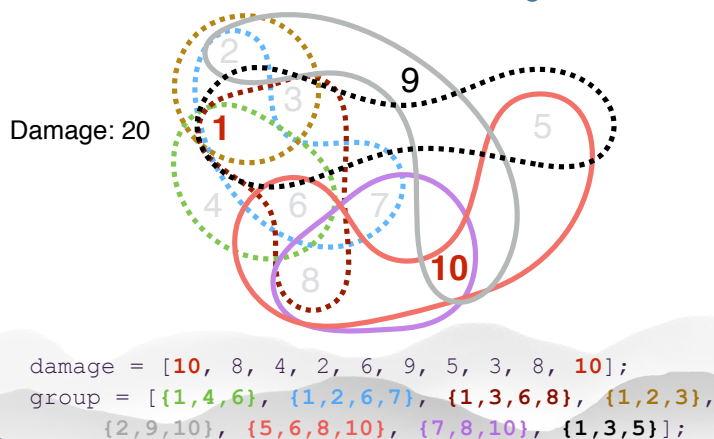


11

## Simple Set Select Algorithm

### Greedy algorithm

- choose the largest damage available element
- eliminate choices that are no longer valid



12

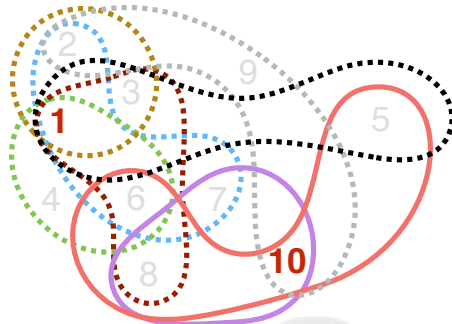


## Simple Set Select Algorithm

### Greedy algorithm

- choose the largest damage available element
- eliminate choices that are no longer valid

Damage: 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

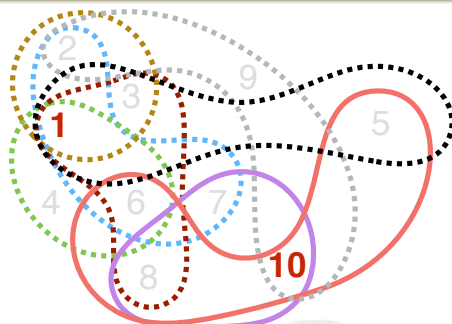
## Simple Set Select Algorithm

### Greedy algorithm

- choose the largest damage available element
- eliminate choices that are no longer valid

Damage: 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



## Bagua Problem Data + Decisions (bagua.mzn)

### ⌘ Data

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

### ⌘ Decisions

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

15

## SetSelect Constraints + Objective (bagua.mzn)

### ⌘ At most one intersection

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

### ⌘ Objective

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

16





## Solving the Model

### ⌘ Executing the model

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

### ⌘ **Better** than the greedy algorithm!!

17

## Image Credits

All graphics by Marti Wong, ©The Chinese University of Hong Kong and the University of Melbourne 2016

18