



Predicates

Jimmy Lee & Peter Stuckey



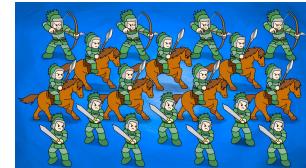
Defending the Jing Province



What is in an Army Unit?



Elite Soldiers



Selecting Team Leaders



Perceived Valuation of Elite Soldiers

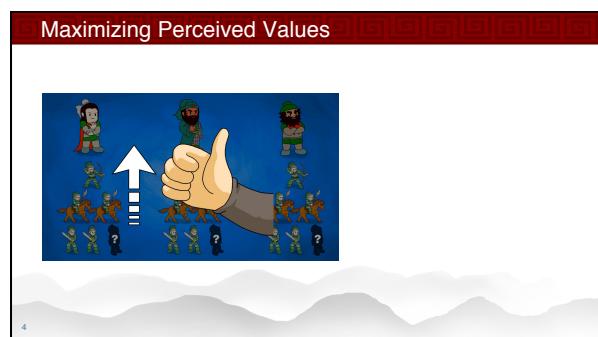
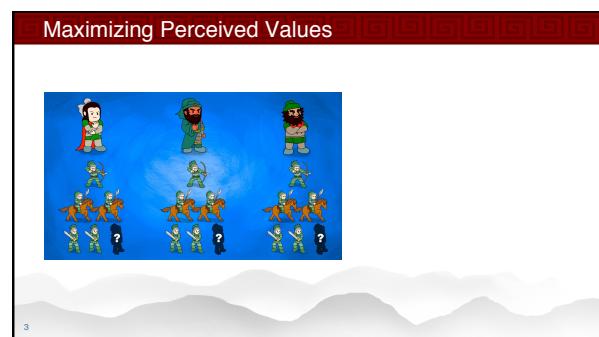
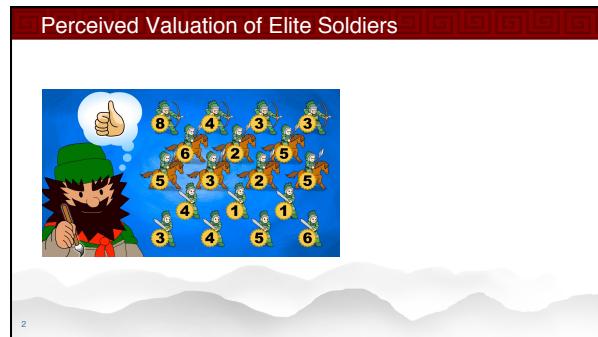
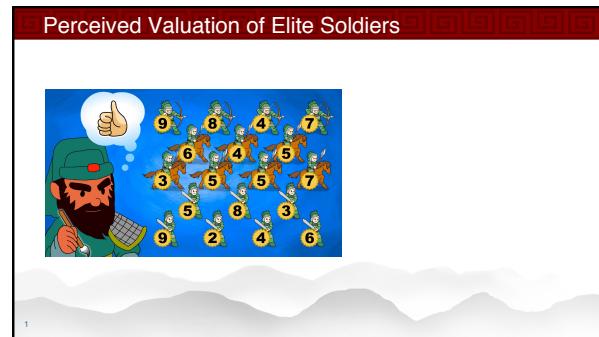




THE UNIVERSITY OF
MELBOURNE



香港中文大學
The Chinese University of Hong Kong



Unless otherwise indicated, this material is © The University of Melbourne and The Chinese University of Hong Kong. You may share, print or download this material solely for your own information, research or study.



ArmySelect Data and Decisions (armySelect.mzn)

⌘ Data

```
enum ELITE; % all the elite soldiers
set of ELITE: ARCHER;
set of ELITE: CAVALRY;
set of ELITE: INFANTRY;
array[ELITE] of int: liu_value;
array[ELITE] of int: guan_value;
array[ELITE] of int: zhang_value;
```

⌘ Decisions

```
var set of ELITE: Liu;
var set of ELITE: Guan;
var set of ELITE: Zhang;
```

11

Constraints (armySelect.mzn)

⌘ Army units do not overlap. Don't forget this one!

```
include "globals.mzn";
all_disjoint([Liu, Guan, Zhang]);
```

⌘ Army formation constraints

```
card(Liu intersect ARCHER) >= 1;
card(Liu intersect CAVALRY) >= 2;
card(Liu intersect INFANTRY) >= 2;
card(Liu) = 6;
```

⌘ Similarly, for Guan and Zhang, but that is tedious!

12



Predicates

- ⌘ Here, we are repeatedly building a system of constraints of the same form. We want a way to encapsulate this.

Predicates

- a definition of a complex constraint
- reusable in the model

13

Predicates

- ⌘ A predicate is a constraint relation whose definition is hidden/elsewhere
- ⌘ Analogous to a procedure/macro
- ⌘ In fact, all global constraints in MiniZinc are defined as predicates.
- ⌘ A predicate has type `var bool`
- ⌘ It can be used “anywhere” a `var bool` can be used
 - `not quite correct, more on this later`

14



Using Predicates (armySelectPred.mzn)

⌘ Predicate Definition

```
predicate form(var set of ELITE: team) =  
    card(team intersect ARCHER) >= 1  
    /\ card(team intersect CAVALRY) >= 2  
    /\ card(team intersect INFANTRY) >= 2  
    /\ card(team) = 6;
```

⌘ Predicate Use

```
form(Liu); form(Guan); form(Zhang);
```

⌘ More correctly (slides often omit keyword)

```
constraint form(Liu);  
constraint form(Guan);  
constraint form(Zhang);
```

15

Objective

⌘ Maximizing the sum of perceived values

```
var int: obj =  
    sum(e in Liu) (liu_value[e]) +  
    sum(e in Guan) (guan_value[e]) +  
    sum(e in Zhang) (zhang_value[e]);  
solve maximize obj;
```

16



Summary

- ⌘ If you are repeatedly writing the same constraint code for different variables
- ⌘ Define a **predicate**
- ⌘ Please refer to the **reference materials** for more precise descriptions of predicates and related concepts.

17

Image Credits

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

18