Too Many Solutions

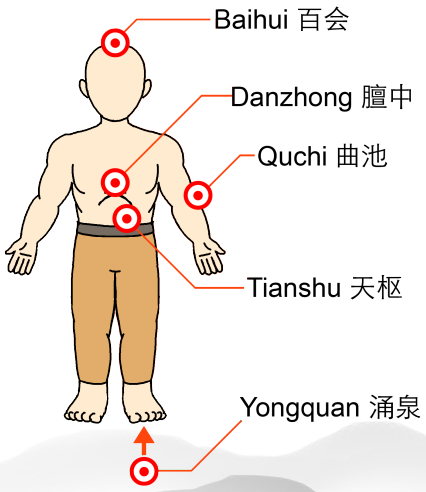
Jimmy Lee & Peter Stuckey

香港中文大學 The Chinese University of Hong Kong
THE UNIVERSITY OF MELBOURNE

Combating Two Fearsome Generals

2

The Generals' Weak Spots



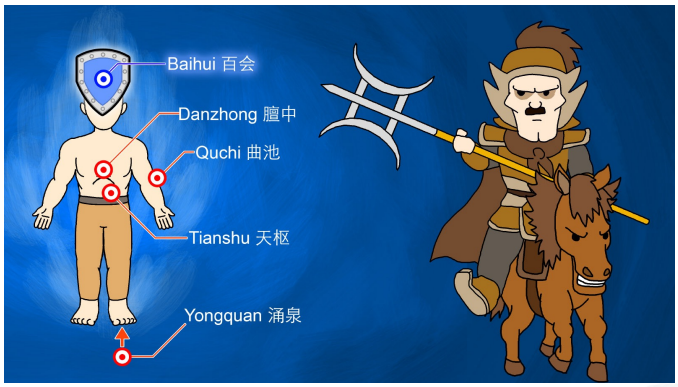
3

Damage Matrix

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

4

Yan Liang's Invincible Helmet



5

Wen Chou's Lower Body Kung Fu

6

Attacking Two Generals

- Guan, Zhang, and Xu will each attack a **different** spot of each general in turn
- The first general has an impenetrable helmet, so cannot be attacked on the head
- If they attack the first general in the lower body the second general will be able to defend the lower body perfectly, so they cannot attack his lower body
- Find out the spots they should attack in order to maximize the total damage to the generals

7

Data, Decisions and Objective (twogenerals.mzn)

- Data

```
enum HERO;
enum SPOT;
array[HERO,SPOT] of int: damage;
```

- What are the decisions?

```
array[HERO] of var SPOT: pos1;
array[HERO] of var SPOT: pos2;
```

- What is the objective?

```
solve maximize sum(h in HERO)
   (damage[h,pos1[h]] +
    damage[h,pos2[h]]);
```

8

Constraints (twogenerals.mzn)

- Each strikes each general in a different position

```
alldifferent(pos1);
alldifferent(pos2);
```

- No one strikes the first generals head, and if striking the first general low, they must strike the second general high

```
set of SPOT: LO = {TIANSHU,YONGQUAN};
set of SPOT: HI = SPOT diff LO;
forall(h in HERO)
   (pos1[h] != BAIHUI /\
    pos1[h] in LO -> pos2[h] in HI);
```

9

Two Generals

- Running the model we obtain
  - `pos1 = [YONGQUAN, BAIHUI, QUCHI]`
  - `pos2 = [DANZHONG, BAIHUI, QUCHI]`
- **What!?**
- No one is meant to strike the first general's head?

10

Model Debugging

- How do you tell that the model is correct!?
- Examples where you know the correct solution(s)
- What can go wrong?
  - Too many solutions, superoptimal answer
  - Missing solutions, suboptimal answer
  - No solutions, ?????

11

Too Many Solutions

- The easier case :-)
- Two reasons
  - A problem with the model allowing solutions which do not satisfy the "desired" constraints
  - The problem has lots of solutions that are not "interesting"

12

Too Many Solutions Debugging Strategy

- Isolate which constraint definition should remove the solution
- Examine the definition + fix it!

13

Too Many Solutions Example Fix (twogeneralsFixed.mzn)

- What is wrong with

```
forall(h in HERO)
  (pos1[h] != BAIHUI /\
    pos1[h] in LO -> pos2[h] in HI);
```

- Remember precedence of operators
  - this is equivalent to

```
forall(h in HERO)
  ((pos1[h] != BAIHUI /\ pos1[h] in LO)
    -> pos2[h] in HI);
```

- Correct version

```
forall(h in HERO)
  (pos1[h] != BAIHUI /\
    (pos1[h] in LO -> pos2[h] in HI));
```

14

Correctly Too Many Solutions

- You will not notice this unless you ask for all solutions to be printed
- E.g. lots.mzn

```
array[1..10] of var 1..10: a;
solve satisfy;
output ["\(a)\n"];
```

```
$ minizinc lots.mzn -a
```

- <<massive numbers of solutions>>

15

Correctly Too Many Solutions

- Add constraints to reduce the solutions
  - E.g. symmetry elimination
  - `var_sym(a);`
- Or add an objective function
- E.g.
  - `solve minimize sum(a);`

16

Summary

- Too many solutions
- Solution of the model is not a solution of the problem
  - find the constraint(s) that should remove the erroneous solution, fix them
- Model has too many correct solutions
  - don't ask for all solutions
  - add constraints to select only some solutions or even an objective

17

Image Credits

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

18