Global Cardinality Constraint

Jimmy Lee & Peter Stuckey

Solving the Model

- Solving … solving … solving …
- Nothing came out after 5 mins
- A solution finally came out in 10.5 minutes
- Changing the number of days to 6, the model can find an optimal solution only after 7 seconds

What went wrong?

Shift Requirement Constraints

- There are `o` soldiers on night shift each day

```
forall(d in DAY)
   (sum(s in SOLDIER)
      (roster[s,d] = NIGHT) = o);
```

- There are between `l` and `u` soldiers on each evening shift

```
forall(d in DAY)
   (sum(s in SOLDIER)
      (roster[s,d] = EVE) >= l);
forall(d in DAY)
   (sum(s in SOLDIER)
      (roster[s,d] = EVE) <= u);
```

3

Shift Requirement Constraints

- There are `o` soldiers on day shift each night

```
forall(d in DAY)
   (sum(s in SOLDIER)
      (roster[s,d] = NIGHT) = o);
```

- There are between `l` and `u` soldiers on each evening shift

```
forall(d in DAY)
   (sum(s in SOLDIER)
      (roster[s,d] = EVE) >= l);
forall(d in DAY)
   (sum(s in SOLDIER)
      (roster[s,d] = EVE) <= u);
```

Common Subexpressions

4

Intermediate Variables

- Intermediate variables
  - Store values of expressions that are reused
  - Are dependent on decisions
  - (note: intermediate parameters too!)

```
array[DAY] of var 0..card(SOLDIER): onEve;
onEve = [sum(s in SOLDIER)(roster[s,d] = EVE)
   | d in DAY];
forall(d in DAY)
   (onEve[d] >= l /\ onEve[d] <= u);
```

- Choose bounds for intermediates well
- Or simply

```
array[DAY] of var l..u: onEve;
onEve = [sum(s in SOLDIER)(roster[s,d] = EVE)
      | d in DAY];
```

5

Another Common Subexpression

- Objective

```
var int: tOnEve = sum(d in DAY)
   (sum(s in SOLDIER)(roster[s,d] = EVE));
solve maximize (tOnEve);
```

6

The Patrol Model (patrolV2.mzn)

```
enum SOLDIER;
enum SHIFT;
int: nDays;
set of int: DAY = 1..nDays;
int: o;
int: l;
int: u;

array[SOLDIER,DAY] of var SHIFT: roster;

constraint forall(d in 1..(nDays-2),
   s in SOLDIER)((roster[s,d] = NIGHT) /\
      (roster[s,d+1] = NIGHT)
   -> (roster[s,d+2] != NIGHT));
constraint forall(d in 1..(nDays-1),
   s in SOLDIER)((roster[s,d] = EVE) ->
      (roster[s,d+1] != NIGHT));
```

7

The Patrol Model (patrolV2.mzn)

```
constraint forall(d in DAY)(sum (s in SOLDIER)
   ((roster[s,d] = NIGHT)) = o);
array[DAY] of var l..u: onEve;
constraint onEve = [sum (s in SOLDIER)
   (roster[s,d]=EVE) | d in DAY];

solve maximize sum(onEve);
```

8

Solving the New Model

- The new version of the model after removing the common subexpression is solved in 4.5 seconds

9

Partitioning Problems

- Many times when we are partitioning a set we have to partition it with bounds on the size of the partitions
  - e.g. `#NIGHT = o, l ≤ #EVE ≤ u`
- We have special constraints for partitioning with size bounds
  - `global_cardinality(x, v, c)`
  - Sizes of v and c are the same
  - Constrains $c_i = \Sigma_{j \text{ in } 1..n}(x_j = v_i)$
  - Collects the counts, and bounds the count of the number of occurrences of v_i
  - e.g. `global_cardinality(x,[1,2],[2,1])`
    `x = [1,1,2,3]` ✅, `[1,2,3,4]` ❌

10

Global Cardinality

- Replace

```
forall(d in DAY)(sum(s in SOLDIER)
   (roster[s,d] = NIGHT) = o);
forall(d in DAY)(sum(s in SOLDIER)
   (roster[s,d] = EVE) >= l);
forall(d in DAY)(sum(s in SOLDIER)
   (roster[s,d] = EVE) <= u);
```

- By

```
array[DAY] of var l..u: onEve;
constraint forall(d in DAY)
   (global_cardinality(
      [roster[s,d] | s in SOLDIER],
      [NIGHT, EVE],
      [o, onEve[d]]));
```

11

Global Cardinality Variants

- There are a number of variants of global cardinality
- Collecting counts, and requiring every value is counted
  - `global_cardinality_closed(x, v, c)`
  - Values for x_i must be from v, i.e. $\forall i.\exists j.\ x_i = v_j$
- Bounding the count of the number of occurrences
  - `global_cardinality_low_up(x, v, lo, hi)`
  - Constrains $lo_i \leq (\Sigma_{j \text{ in } 1..n} (x_j = v_i)) \leq hi_i$
- Bounding the count, and requiring that every value is counted

`global_cardinality_low_up_closed(x,v,l,u)`

12

The Patrol Model (patrolV3.mzn)

```
enum SOLDIER;
enum SHIFT;
int: nDays;
set of int: DAY = 1..nDays;
int: o;
int: l;
int: u;

array[SOLDIER,DAY] of var SHIFT: roster;

constraint forall(d in 1..(nDays-2),
   s in SOLDIER)((roster[s,d] = NIGHT) /\
      (roster[s,d+1] = NIGHT)
   -> (roster[s,d+2] != NIGHT));
constraint forall(d in 1..(nDays-1),
   s in SOLDIER)((roster[s,d] = EVE) ->
      (roster[s,d+1] != NIGHT));
```

13

The Patrol Model (patrolV3.mzn)

```
array[DAY] of var l..u: onEve;
constraint forall(d in DAY)(global_cardinality(
   [roster[s,d] | s in SOLDIER],
   [NIGHT, EVE], [o, onEve[d]]));

solve maximize sum(onEve);
```

14

Solving the Global Constraint Model

- The global constraint version of the model is solved in 1.5 seconds
- In subsequent courses, we will explain how modeling can affect solving efficiency

15

Summary

- Many discrete optimization problems involve
  - deciding a function f: DOM ➔ COD
  - this can be seen as partitioning DOM
- The partition view is useful when we reason about the sets F(c) (pseudo-inverse of f)
- Partitioning with cardinality constraints is a substructure captured by the
  - `global_cardinality` family
- Common subexpressions
- Logical connectives

16

Image Credits

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