Disjunctive Scheduling

Jimmy Lee & Peter Stuckey

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

Quality Assurance

Resources

- Critical to most scheduling problems are limited resources
  - unary resource (at most one task at a time)
  - cumulative resource (a limit on the amount of resource used at any time)
- Liu Bei is a resource

3

Unary Resources

- The Project Scheduling problem with non-overlap involved a unary resource
  - number of tasks executing at one time
- Unary resources are common
  - machine
  - nurse, doctor, worker in a roster
  - track segment (one train at a time)
  - …
- Liu Bei is a unary resource!

4

Scheduling Concepts (so far)

- Tasks
  - start time, duration (and end time)
  - other attributes

```
array[TASK] of var int: start;
array[TASK] of (var) int: duration;
```

- Precedences
  - one task can only start after another finishes
  - task t1 precedes t2

```
start[t1] + duration[t1] <= start[t2]
```

5

Nonoverlap (disj_sched_1.mzn)

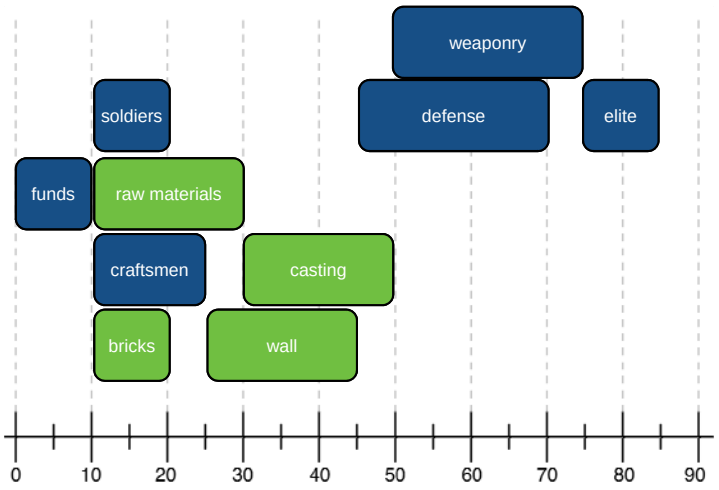
- Liu Bei wants to be in charge of quality assurance of the production materials to buy and the finished products

```
predicate nonoverlap(var int:s1, var int:d1,
                     var int:s2, var int:d2) =
          s1 + d1 <= s2 \/ s2 + d2 <= s1;

set of TASK: LIU =
   {CASTING, RAW_MATERIALS, BRICKS, WALL};
forall(t1, t2 in LIU where t1 < t2)
      (nonoverlap(start[t1],duration[t1],
                  start[t2],duration[t2]));
```

6

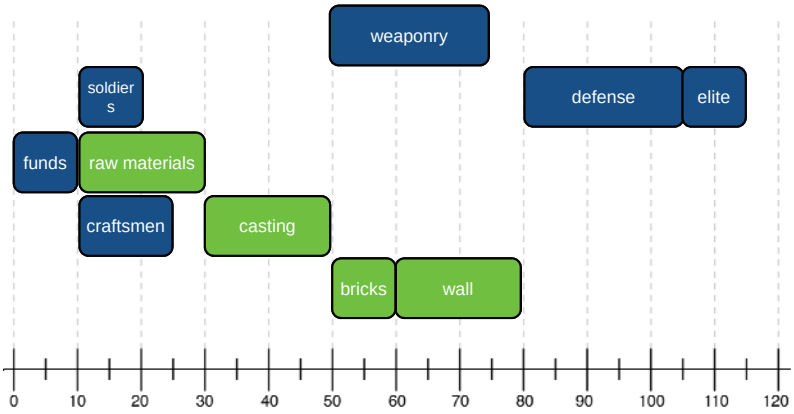
The Basic Scheduling Solution




```
makespan    F   S   D  We   E   R  Cr  Ca   B  Wa
85        = [0, 10, 45, 50, 75, 10, 10, 30, 10, 25]
```

7

Scheduling With Quality Control




```
makespan    F   S   D  We   E   R  Cr  Ca   B  Wa
115       = [0, 10, 80, 50, 105, 10, 10, 30, 50, 60]
```

8

Combat Expert: Guan Yu

9

Warfare Expert: Zhang Fei

10

More Resources (disj_sched_2.dzn)

- Quality Assurance: Liu Bei
  - Raw materials, bricks, casting, wall

```
LIU = {CASTING, RAW_MATERIALS, BRICKS, WALL};
```

- Warfare Expertise: Zhang Fei
  - Soldiers, craftsmen, raw materials, bricks

```
ZHANG = {SOLDIERS, RAW_MATERIALS, BRICKS,
   CRAFTSMEN};
```

- Combat Expertise: Guan Yu
  - Weaponry training, defense training, elite army, wall

```
GUAN = {WEAPONRY, DEFENSE, ELITEARMY,
   WALL};
```

11

More Nonoverlaps (disj_sched_2.mzn)

```
predicate nonoverlap(var int:s1, var int:d1,
                     var int:s2, var int:d2) =
          s1 + d1 <= s2 \/ s2 + d2 <= s1;
predicate exclusive(set of TASK: tasks) =
   forall(t1, t2 in tasks where t1 < t2)
      (nonoverlap(start[t1],duration[t1],
                  start[t2],duration[t2]));

set of TASK: LIU;
constraint exclusive(LIU);
set of TASK: ZHANG;
constraint exclusive(ZHANG);
set of TASK: GUAN;
constraint exclusive(GUAN);
```

12

The "disjunctive" Global Constraint

- Nonoverlap only considers two tasks at a time
  - a unary resource requires non overlap for all pairs of tasks that use it
- Disjunctive constraint
  - disjunctive(<start time array>, <duration array>)
    - ensure no two tasks in the array overlap in execution

```
predicate disjunctive(array[int] of var int:s,
                      array[int] of var int:d) =
   forall(i1,i2 in index_set(s) where i1 < i2)
         (nonoverlap(s[i1],d[i1],s[i2],d[i2]));
```

13

Unary Resource Problem Revisited (disj_sched_glo.mzn)

- Replace nonoverlap with disjunctive
- We need to build the start times and durations for all tasks using a resource
  - perfect for a local variable

```
include "disjunctive.mzn";
predicate exclusive(set of TASK: tasks) =
   let {array[int] of var int:
      ss = [start[t] | t in tasks];
        array[int] of int:
      dd = [duration[t] | t in tasks];}
   in disjunctive(ss,dd);
```

14

Solving the Model

```
makespan    F   S   D  We   E    R  Cr  Ca   B  Wa
130 =      [0, 35, 55, 95, 120, 55, 10, 75, 25, 35]
```

- That is taking too long!!!

15

Job Shop Scheduling

- The story problem is an adaptation of the well-known Job Shop Scheduling Problem
  - remarkably hard
- For some 10x10 instances from 1963
  - we did not know the optimal solution until 1989!
- There are a lot of approximation algorithms
- The online version is also heavily studied
  - where we have to schedule a job, given an existing schedule, then schedule the next job

16

Summary

- Disjunctive scheduling
  - allows us to express that two tasks do not overlap in execution
  - without specifying the relative order
- disjunctive global constraint
  - capture a set of tasks on a unary resource
- Many classic scheduling problems
  - job shop scheduling
  - open shop scheduling

17

Image Credits

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

18