COMP90046 Constraint Programming

Peter Stuckey

Critical Information #1

▸ Lecture times:
- Mondays 12:00 - 13:00
- Alan Gilbert 109 (Theatre 2)

▸ Workshop times:
- Tuesdays 11:00 - 12:00 Alice Hoy 101
- Fridays 11:00 - 12:00 Alice Hoy 3.33
- workshops commence in week 2

▸ How much of Cryptarithm assignment one have you completed.

- A: WHAT there is an ASSIGNMENT!
- B: seen it
- C: thought about it
- D: tried it
- E: finished it.

Week 3 Survey 3B

▸ How many lectures in Sets have you watched
- – A: none
- – B: 1
- – C: 2-3
- – D: 4
- – E: all

Survey 3C

▸ How much of the Workshop 1 have you attempted

– A: none

– B: 1 question

– C: 2 questions

– D: 3 questions

– E: all

Survey 3D

▸ How much of the Workshop 1 have you successfully completed

- – A: none
- – B: 1 question
- – C: 2 questions
- – D: 3 questions
- – E: all

‣ Which constraint is preferable for using a MIP solver.

– A: sum(i in OBJ)(size[i] * x[i]) <= capacity

– B: sum(i in OBJ)(size[i] * bool2int(x[i])) <= capacity

– C: sum(i in OBJ)(size[i] * bool2int(i in x)) <= capacity

– D: sum(i in x)(size[i]) <= capacity

– E: sum(i in ub(x))(size[i] * bool2int(i in x)) <= capacity

▸ Which declaration is best to represent a set of numbers from 0..100000 of cardinality at most 10

  – A: array[1..10] of var 0..100000: x;

  – B: var set of 0..100000: x;

  – C: array[0..100000] of var 1..10: x;

  – D: array[0..100000] of var bool: x;

  – E: var set of 1..10: x;

Survey 3G

▸ Which declaration is best to represent a set of numbers from 1..10 of cardinality at most 100000

- A: array[1..10] of var 0..100000: x;
- B: var set of 0..100000: x;
- C: array[0..100000] of var 1..10: x;
- D: array[0..100000] of var bool: x;
- E: var set of 1..10: x;

Survey 3H

▸ Which declaration is best to represent a set of numbers from 1..100000 of cardinality between 3 and 10

- A: array[1..10] of var 0..100000: x;
- B: var set of 0..100000: x;
- C: array[1..100000] of var 3..10: x;
- D: array[1..100000] of var bool: x;
- E: array[3..10] of var 1..100000: x;

TeamSelect Example

‣ Xavier, Yuri, and Zena each have to pick indoor soccer teams from the players:

  – Goalies: Ant, Bee

  – Defence: Chu, Deb, Eve, Fin

  – Offence: Ged, Hel, Ila, Jan, Kim

‣ Each team has to have one goalie, two defence, and two offence and one reserve (of any type)

‣ The teams of Xavier and Yuri can only have two common members, similarly for Xavier and Zena

‣ Each captain has a perceived value of each player. Choose the teams which maximise the total perceived value.

‣ Captains: Xavier, Yuri, Zena

‣ Players:

  – Goalies: Ant, Bee

  – Defence: Chu, Deb, Eve, Fin

  – Offence: Ged, Hel, Ila, Jan, Kim

‣ One goalie, two defence, and two offence and one reserve (of any type)

‣ At most 2 common members except Yuri + Zena

‣ value = [| 2, 5, 6, 8, 9, 5, 8, 7, 7, 4, 6

‣ | 9, 8, 4, 7, 6, 4, 5, 3, 5, 5, 7

‣ | 8, 4, 3, 3, 6, 2, 5, 5, 3, 2, 5 |];

- On the island of knights and knaves you meet three brother natives who say
  - Larry: I am a knight like my brother Liam
  - Liam: At least one of us is a knave
  - Tim: All brothers whose name starts with T are knights
- Who is a knight?
  - A: [no, no, yes]
  - B: [yes, yes, no]
  - C: [no, no, no]
  - D: [yes, yes, yes]
  - E: [yes, no, yes]

Knights and Knaves 3

‣ On the island of knights and knaves you meet three brother natives who say

> ‣ Larry: I am a knight like my brother Liam
>
> ‣ Liam: At least one of us is a knave
>
> ‣ Tim: All brothers whose name starts with T are knights

‣ Build a MiniZinc model to answer!

▸ Island of tall and short knights and knaves

– tall knights and short knaves always speak truth

– short knights and tall knaves always lie

▸ You meet a short and tall native, they say

– X: my partner is a knight

– Y: we are both knights

▸ but you aren't sure which said which

– A: X is a tall knight, Y is a short knight

– B: X is a tall knave, Y is a short knight

– C: X is a short knight, Y is a tall knave

– D: X is a short knave, Y is a tall knight

– E: X is a short knight, Y is a short knight

EOF

Dynamic Lot Sizing

▸ Given T time periods and for each period t

- – a demand for the product d[t]
- – a setup cost for producing any product s[t]
- – a cost per unit product c[t]
- – a hold cost for storing a unit product h[t]

▸ Determine how much product to produce in each time period to minimize costs and meet demand

```
T = 10;

d = [10,12,13,20,8,12,8,4,18,3];
s = [10,20,10,20,10,20,10,20,0,20];
c = [3,6,7,8,6,6,4,5,10,4];
h = [1,1,1,1,2,2,1,1,1,2];
```