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

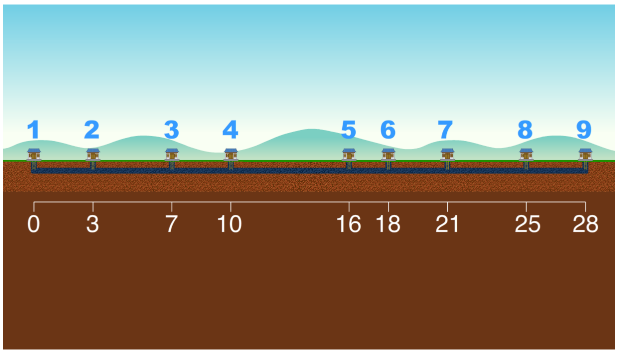
More Permutation Problem

Jimmy Lee & Peter Stuckey

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

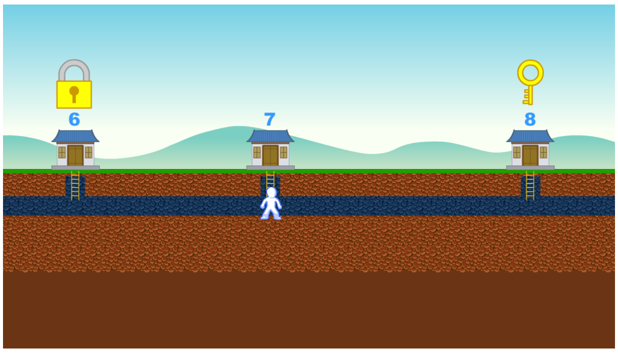
Delivering Messages in a Tunnel

The Tunnel Problem



3

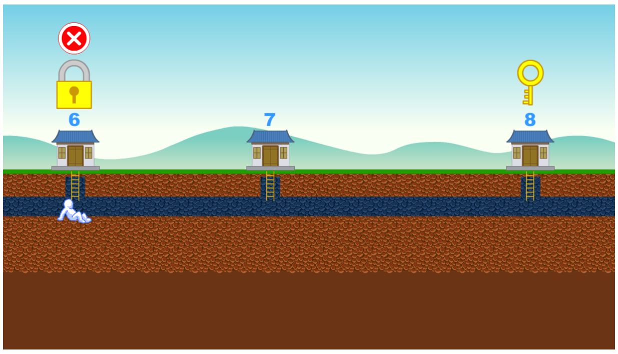
Point Precedence



4

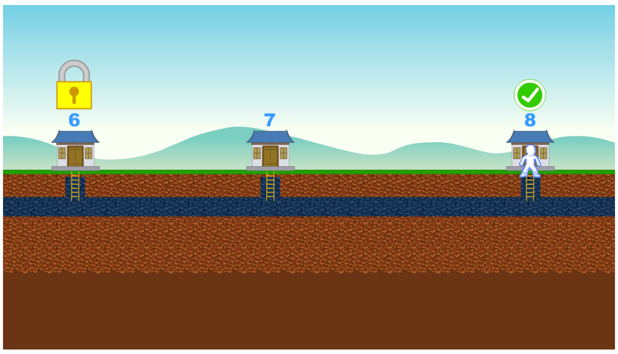
THE UNIVERSITY OF MELBOURNE 香港中文大學 The Chinese University of Hong Kong

Point Precedence



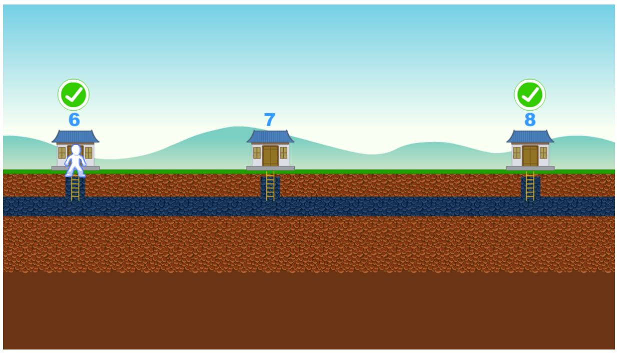
5

Point Precedence



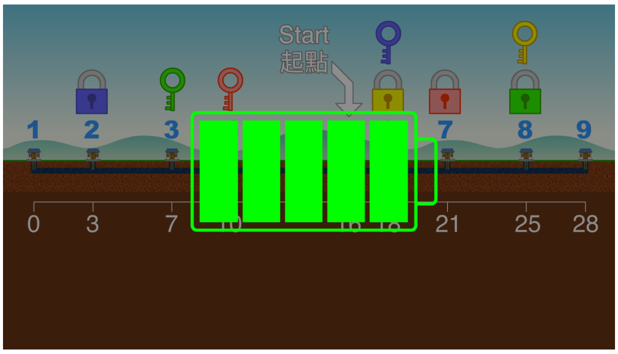
6

Point Precedence



Point Precedence

Objective



9

TSP on a Line

- Given a set of military pivotal points on a line, and a set of precedences amongst the pivotal points, visit each pivotal point in turn starting from the 5th pivotal point to
  - satisfy the precedence requirement, and
  - minimize the total distance travelled

10

Permutation Problems

- An important class of matching problems are permutation problems
  - Place a set of objects OBJ in an order
- This is a matching of OBJ with 1..*n*
  - where *n* is the cardinality of OBJ
- At least two viewpoints
  - DOM = OBJ and COD = 1..*n*
  - DOM = 1..*n* and COD = OBJ
- The Belt problem and this Tunnel problem are both permutation problems

11

The Tunnel Model (tunnel.mzn)

- Data

```
enum PIVOT;
PIVOT: first;

set of int: POS = 1..card(PIVOT);
array[PIVOT] of int: coord; % coord of pivot

int: m; % number of precedences
set of int: PREC = 1..m;
array[PREC] of PIVOT: left;
array[PREC] of PIVOT: right;
```

12

THE UNIVERSITY OF MELBOURNE

香港中文大學 The Chinese University of Hong Kong

The Tunnel Model (tunnel.mzn)

- Decisions of the two viewpoints
  - order: the posn of each pivotal point

```
array[PIVOT] of var POS: order;
```

  - route: the pivotal point of each step in the route

```
array[POS] of var PIVOT: route;
```

- Constraints

```
route[1] = first;
inverse(order,route);
forall(i in PREC)
   (order[left[i]] < order[right[i]]);
```

- Objective

```
solve minimize sum(i in 1..card(PIVOT)-1)
   (abs(coord[route[i]] - coord[route[i+1]]));
```

13

Solving the Model

```
Route: [P5, P4, P3, P9, P8, P7, P6, P2, P1]
Total Distance: 58
```

14

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

Summary

- The Tunnel problem is a simplification of the classic Traveling Salesman Problem (TSP) in computer science but our example has also side constraints (precedence)
- The TSP is an important problem in graph theory and has applications to routing and optimisation in general
- In our example, some requirements are **impossible** to express in a certain viewpoint, making the combined model the only way to formulate the complete problem

Summary

- Permutation problems
  - always have at least two viewpoints
- Choose the viewpoint that is:
  - **possible/easy** to express constraints and objective
- Otherwise, choose both viewpoints and add
  - inverse constraint

Photo Problem

- Given *n* people line them up for a photo with the most friendliness, defined as the sum of the friendliness between each pair of people adjacent in the line.

```
int:n ;
set of int: PERSON = 1..n;
set of int: POS = 1..n;
array[PERSON,PERSON] of int: friend;
```

- How should this be modeled?

17

PhotoProblem Model One

- Variables: the position of each person

```
array[PERSON] of var POS: x;
```

- Constraints:

```
alldifferent(x);
```

- Objective ???????

```
solve maximize …
```

- Hard to see how to express objective
- This is the wrong viewpoint

18

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

PhotoProblem Model Two

- Variables

```
array[POS] of var PERSON: y;
```

- Constraints

```
alldifferent(y);
```

- Objectives

```
solve maximize sum(i in 1..n-1)
                  (friend[y[i],y[i+1]]);
```

- Easy to express constraints, and objective!

19

Image Credits

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

20