University of Malta
BSc(Hons.) IT Year II

Problem Solving Techniques in Artificial Intelligence

Lecture 8

Agenda Based Search

- Coursework Description
- Best First Search

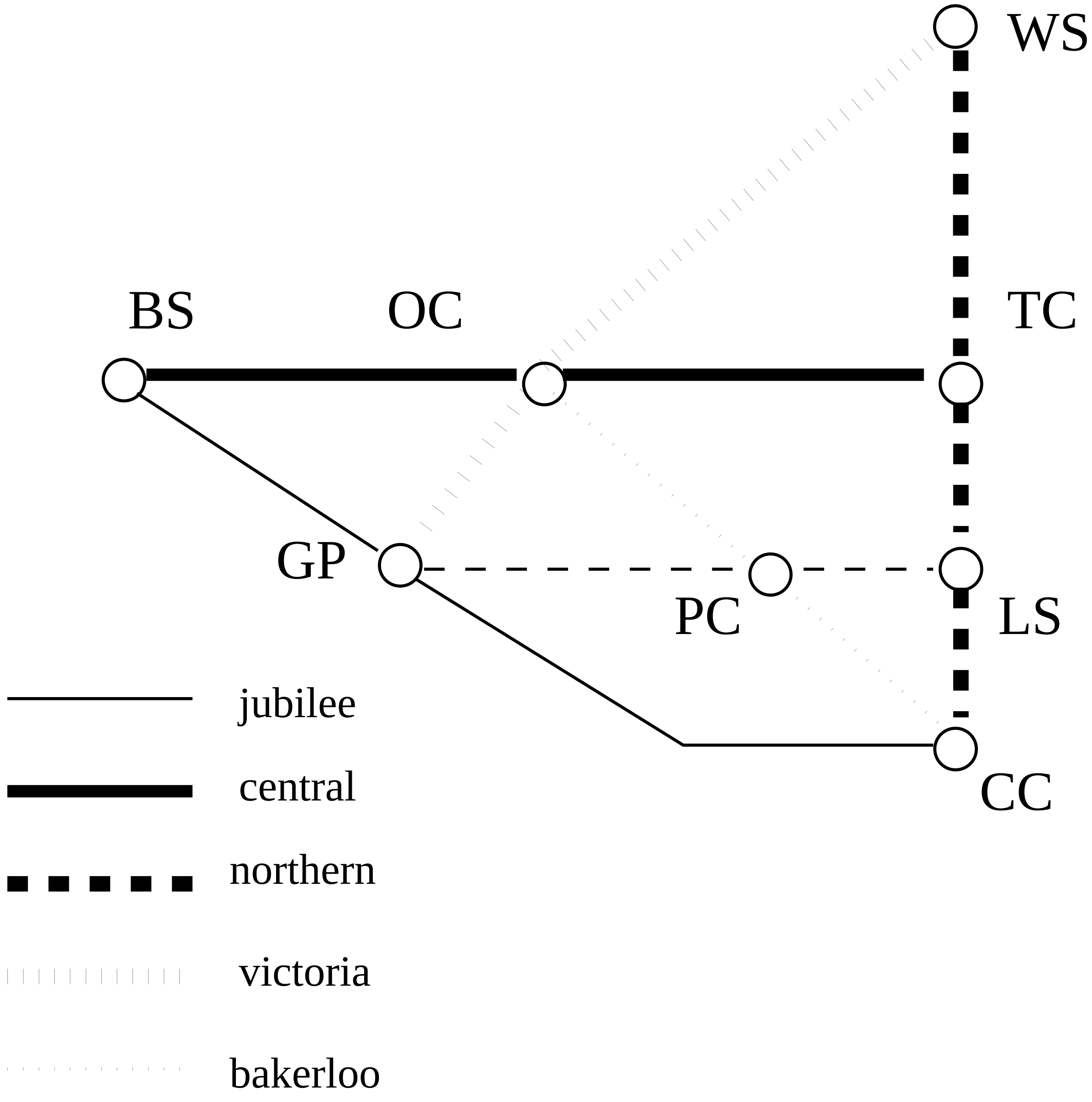
Department of Computer Science and AI
1998/9

Coursework

Build a general purpose agenda-based search system that can be cleanly configured to handle

- **Different Search Strategies**: Depth First, Breadth First, Best First.
- **Different Problem Domains**: Underground Map, Eight Puzzle
- **Different Cost Measures**: Path Cost, Heuristic Cost.
- **Performance Measurement**:
  - Search Cost (number of nodes generated/number of nodes examined)
  - Quality of Solution (path cost of solution)

Problem Domain 1
London Underground with Lines



Remarks

- This is just like the domain we used in the lectures except that there are different lines and interchanges. To get from A to B may involve a change of line.
- Path costs are assumed to be 1 for each stop.
- Line changes cost 1 except at Piccadilly Circus (PC) where they costs 2.
- Aim is to find minimum cost path between arbitrary stations.

The Eight Puzzle

The puzzle consists of eight sliding tiles numbered by digits from 1 - 8 arranged in a 3x3 array of nine cells.

One of the cells is always empty, and any adjacent cell can be moved into the empty cell. We can say that the empty cell is allowed to move around.

The aim of the puzzle is to discover a sequence of moves for reaching a goal state, some special arrangement of tiles, from an arbitrary starting position e.g.

$$\begin{array}{|c|c|c|}\hline 1 & & 3 \\ \hline 8 & 2 & 4 \\ \hline 7 & 6 & 5 \\ \hline \end{array} \quad ?\Rightarrow? \quad \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline 8 & & 4 \\ \hline 7 & 6 & 5 \\ \hline \end{array}$$

Representing the Board State

The board can be regarded as comprising 9 *tiles* (8 numbered tiles and one blank), each in a given *position*. A *state* of the board encodes the position of each of these 9 tiles We will consider 2 possible board state representations:

- As a 9-element list whose i^{th} element encodes the i^{th} tile position (cf. Bratko)
- As a single integer (cf Russell and Norvig).

Bratko's scheme is easier to understand, but much less efficient: states take up much more room and are harder to compare.

List Representation of States

In this representation a state is a list of coordinates, each of which encodes the position of a tile according to the following scheme:

1/3	2/3	3/3
1/2	2/2	3/2
1/1	2/1	3/1

- Element 0 contains the coordinates of the empty square.
- Element 1 contains the coordinates of the tile marked 1
- Element n ($0 < n < 9$)contains the coordinates of the tile marked n

For instance, the state

1	2	3
8		4
7	6	5

would be

[2/2,1/3,2/3,3/3,3/2,3/1,2/1,1/1,1/2]

Numerical Representation of States

We represent each tile as an integer from 0 to 8 (0 for the blank). We also represent each position as an integer from 0 to 8, arranged as follows:

0	1	2
3	4	5
6	7	8

A state is the sum of the tile numbers times 9 to the power of the tile's position number. For example, the state

1	2	3
8		4
7	6	5

```
1*9^0 +  2*9^1  + 3*9^2 +
8*9^3 +  0*9^4  + 4*9^5 +
7*9^6 +  6*9^7  + 5*9^8 = 247893796
```

Generating Successors

The successors to a given state are obtained by moving the blank one square in any legal direction. In this case there are 4 but this is not always so.

```
1 2 3        1   3   1 2 3   1 2 3   1 2 3
8   4   =>   8 2 4     8 4   8 4     8 6 4
7 6 5        7 6 5   7 6 5   7 6 5   7   5
```

To find the successors of a given state:

- Find the position occupied by the blank. Call that position PB.
- Find all the states in which tiles at a Manhatten Distance of 1 from PB are swapped with PB.

Successors: Bratko

Suppose a certain state has this shape

[$\underline{\text{A}}$,B,$\underline{\text{C}}$,D,E,F,$\underline{\text{G}}$,H]

and that C and G are at a MD of 1 from A.

Then the two successor states will be:

[$\underline{\text{C}}$,B,$\underline{\text{A}}$,D,E,F,G,H] and [$\underline{\text{G}}$,B,C,D,E,F,$\underline{\text{A}}$,H]

Main predicate:

swap(PB,PT,State,SState):

PB is the position of blank in State PT is the position of a Tile at MH 1. SState is the successor state arrived at by swapping PB and PT.

(see Bratko p.289)

Heuristics

Possible heuristics include:

h1(S) = number of tiles not in goal state position

h2(S) = sum of Manhatten distances of each tile from its goal position.

h3(S) = sequence score (S). This measures the degree to which tiles are already correctly ordered.

- A tile in the centre scores 1
- A tile on a non-central square scores 0 is the tile is, in a clockwise direction, followed by its propoer successor.
- Such a tile scores 2 if it is not followed by its proper predecessor.

The estimate is obtained by multiplying h3 by 3 (se Bratko p. 291).