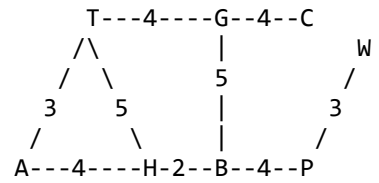
Lecture 3

1. Optimal Path
    1. A*
2. Games
    1. Minimax
    2. Alpha-beta pruning
    3. Progressive Deepening
    4. Heuristic pruning
    5. Heuristic continuation

A*

As well as a cost associated with each path, there may be a minimum possible distance to the goal. For example consider the following:



Problem: find the optimum path from A to W

The minimum possible distance from any node to W is the straight line distance to W. We can use this to help the branch and bound algorithm.

Instead of using the cost, use the cost + minimum distance left. This will discourage steps away from the solution.

Some paths may be redundant e.g. A->T->H is a valid path but A-> H has a lower cost and so the former is redundant.

The A* procedure is a branch and bound search with the inclusion of a lower-bound estimate of remaining cost and removal of redundant paths.

Search Procedure:

1. Form a queue of partial paths. Let the initial queue consist of the zero length, zero step path from the root node to nowhere.
2. Until the queue is empty or the goal has been reached, determine if the first path in the queue reaches the goal node.
    1. If the first path reaches the goal node then do nothing
    2. otherwise:
        1. Remove the first path from the queue.
        2. Form new paths from the removed path by extending one step.
        3. Add the new paths to the queue.
        4. Sort the queue by the sum of total cost *and a lower-bound estimate of the cost remaining*, with least cost at the front.
        5. If two or more paths reach a common node, delete all those paths except for the one that reaches the common node with minimum cost.
3. If the goal has been found then success otherwise failure.

Example: The 8-puzzle

In the 8-puzzle problem 8 differently numbered tiles are fitted into 9 spaces on a 3x3 grid. One space has no tile so that any tile adjacent to the empty space can be moved to occupy it by sliding from its original position. For instance, here is an initial and a goal state:

```
initial state        goal state
2 8 1                1 2 3
4   6                4 5 6
7 5 3                7 8
```

Here are some different heuristic functions:

- h1 (n) = number of misplaced tiles
- h2(n) = P(n) + 2 * R(n) where P(n) is the sum of the Manhattan distances that each tile is from home, and R(n) is the number of direct tile reversals (when two adjacent tiles must be exchanged to be in the order of the goal).
- h3(n) = P(n) + 3 * S(n) where P(n) is the sum of the Manhattan distances that each tile is from home, and S(n) is a sequence score obtained by counting 1 for the central tile (if not empty) and counting 2 for each tile on the board perimeter that is not followed (in clockwise order) by its correct successor.

A solution to the problem may be found by using a best-first search where the heuristic is used to choose which position to search next. (don't forget to exclude loops from the search).

In order to find the optimal solution, branch and bound or A* could be used. In this case the path length is the number of moves so far and the underestimate of distance left could be the heuristic.

Games

Games require different search methods since we are often dealing with an adversary.

Game Tree

A tree in which the nodes denote board configurations, and the branches indicate how one board configuration can be transformed into another by a single move.

An exhaustive search of the game tree for a solution is usually not possible. Need a way to rank board positions. If there is an infallible way to rank board positions then no search is necessary, just choose the move that gives the best position. Unfortunately, this is not possible for the vast majority of games. The answer is to use a board ranking after play has been extended through several levels of move and counter move.

Minimax Search

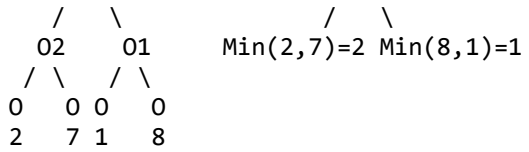
Suppose that board positions can be ranked by a situation analyser to give a single number.

The machine will want to maximise this number for its turn and the opponent will want to minimise it.

The procedure by which the scoring information passes up the game tree is called the MINIMAX procedure,because the score at each node is either the minimum or the maximum of the scores at the nodes immediately below:

Minimax Example

```
          O              Max(2,1)=2
        / \                 / \
```



The numbers in this diagram represent the board ranking. One player tries to get higher numbers while the other tries to get lower numbers.

Minimax procedure:

For each node in the games tree:

1. If the limit of the search has been reached then compute the value of the current position and report the result.
2. If the level is a minimising one, then use minimax on the children of the current position and report the minimum of the results.
3. If the level is a maximising one, then use minimax on the children of the current position and report the maximum of the results.

Alpha-Beta Pruning Principle

If you have an idea that is surely bad, do not take time to see how terrible it is.

Consider the example: It is not necessary to evaluate the branch at the right (with the 8) since the 1 is already lower than the 2 and will not be chosen.

Procedure To perform minimax search with the ALPHA-BETA procedure,

1. If the level is the top level, let alpha be negative maximum and let beta be positive maximum.
2. If the limit of search has been reached, compute the value of the current position relative to the appropriate player. Report the result.
3. If the level is a minimizing level,
    1. Until all children are examined with ALPHA-BETA or until alpha is equal to or greater than beta,
        1. Use the ALPHA-BETA procedure, with the current alpha and beta values, on a child; note the value reported.
        2. Compare the value reported with the beta value; if the reported value is smaller, reset beta to the new value.
    2. Report beta.
4. Otherwise, the level is a maximizing level:
    1. Until all children are examined with ALPHA-BETA or alpha is equal to or greater than beta,
        1. Use the ALPHA-BETA procedure, with the current alpha and beta value, on a child; note the value reported.
        2. Compare the value reported with the alpha value; if the reported value is larger, reset alpha to the new value.
    2. Report alpha.