15-381 Spring 05 Midterm Solution

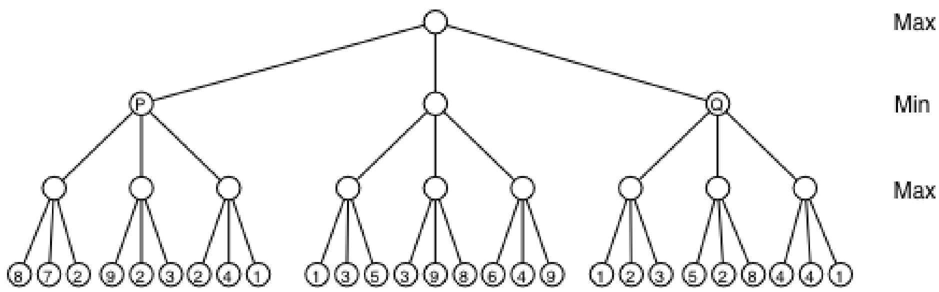
Tuesday March 1, 2005

Name: ______________________________ Andrew ID: ______________________________

- This is an open-book, open-notes examination.You have 80 minutes to complete this examination.
- This examination consists of 6 questions, each worth 20 points. For each student, **only the top 5 scoring questions** will be considered. The worst-scoring question will be discarded (thus you may choose to ignore a question and still potentially get full marks). The maximum possible score is 100.
- Write your answers legibly *in the space provided* on the examination sheet. If you use the back of a sheet, indicate clearly that you have done so on the front.
- Write your name and Andrew ID on this page and your andrew id on the top of each successive page in the space provided.
- Calculators are allowed but laptops and PDAs are not allowed.
- Good luck!

1 Problem 1: Game Tree Search (20 pts)

The figure below is the game tree of a two-player game; the first player is the maximizer and the second player is the minimizer. Use the tree to answer the following questions:



(a) What is the final value of this game?

Answer: 5, which can be determined from a quick application of the minimax algorithm.

(b) Is the final value of beta at the root node (after all children have been visited) $+\infty$? (**T/F**)

Answer: True. The root node is a maximizing node, and the value of beta never changes at a maximizing node.

(c) What is the final value of beta at the node labeled P (after all of P's children have been visited)?

Answer: Since we are at a minimizing node and alpha is negative infinity, beta will take on the smallest value returned by any of P's children. In this case, P's children would return 8, 9, and 4, so the final value of beta at P will be 4.

Suppose we are in the middle of running the algorithm. The algorithm has just reached the node labeled Q. The value of alpha is 5 and the value of beta is $+\infty$.

(d) Will any nodes be pruned?

Answer: Yes. If we work out the algorithm on the leftmost subtree of Q, we see that it returns a value of 3. Beta at Q will then become 3, and 3 is less than the alpha value at Q, so Q will immediately return a value to its parent. This means that the other two subtrees of Q are pruned.

(e) What value will Q return to its parent?

Answer: Since its beta value after visiting the leftmost subtree was less than its alpha value of 5, Q will return its alpha value of 5.

2 Problem 2: Game Theory (20 pts)

Two players, A and B, play a game, in which they each shout out an integer: 1, 2 or 3. If they both shout the same number, they receive a prize:

- They each get one dollar if they both shouted "1".
- They each get two dollars if they both shouted "2".
- They each get three dollars if they both shouted "3".

If they shouted different numbers, they get nothing.

(a) Is it a Nash Equilibrium to both shout "1"? (**T/F**)

Answer: Yes, because if I know that you will shout "1", then I will want to shout "1" too and vice versa.

Consider the mixed strategy of

I1: shout "1" with probability $\frac{1}{3}$
I2: shout "2" with probability $\frac{1}{3}$
I3: shout "3" with probability $\frac{1}{3}$

(b) If both players use this mixed strategy, is that a Nash Equilibrium? (**T/F**)

Answer: No, because if you are going to shout at random, I am better off always shouting "3" so that on the offchance we match, I will get \$3.
If you shout at random and I shout at random, my expected payoff is

$$1 \times 1/9 + 2 \times 1/9 + 3 \times 1/9 = 2/3$$

If you shout at random and I shout "3", my expected payoff is

$$1/3 \times 0 + 1/3 \times 0 + 1/3 \times 3 = 1$$

Now consider a very different game. Two companies, A and B, both make elbow warmers. The more they spend on advertising, the more sales they get, but there are diminishing returns. A's advertising somewhat helps B, and B's advertising somewhat helps A. the exact fomulas are

$$\begin{aligned} P_a &= \overbrace{\log(2a+b)}^{revenue} - \overbrace{a}^{expense} \\ P_b &= \log(a+2b) - b \end{aligned}$$

where

$$\begin{aligned} \text{a} &= \text{\# dollars that A spends on advertising} \\ \text{b} &= \text{\# dollars that B spends on advertising} \\ P_a &= \text{profit to A} \\ P_b &= \text{profit to B} \end{aligned}$$

(c) Work out the nash equilibrium.

Hint: $\frac{\partial P_a}{\partial a} = \frac{2}{2a+b} - 1 \quad \frac{\partial P_a}{\partial b} = \frac{1}{2a+b}$
$\frac{\partial P_b}{\partial a} = \frac{1}{a+2b} \quad \frac{\partial P_b}{\partial b} = \frac{2}{a+2b} - 1$

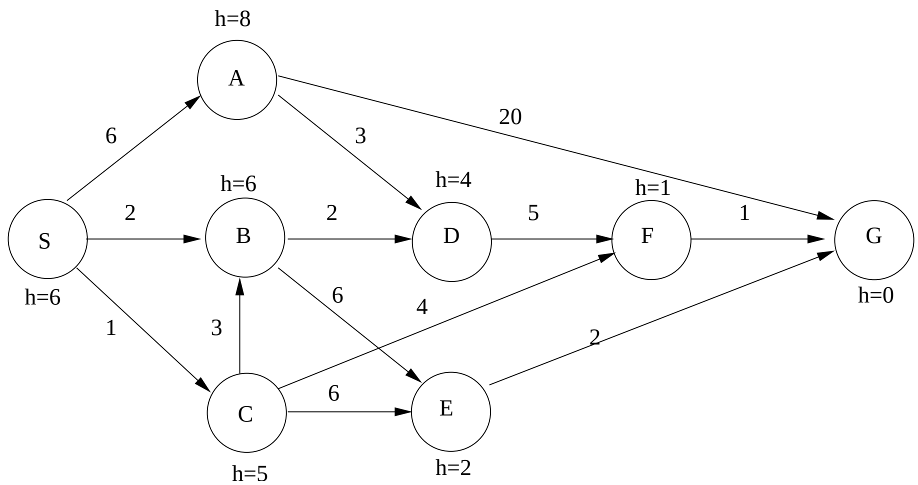
Answer: At N.E.

$$\begin{aligned}
a^* &= argmax_a \; P_a(a, b^*) \\
&= a \; such \; that \; \frac{\partial P_a(a, b*)}{\partial a} = 0 \; (assume \; P_a \; has \; maximum \; at \; point \; of \; zero \; derivative) \\
&= a \; such \; that \; \frac{2}{2a + b^*} - 1 = 0 \\
&= a \; such \; that \; 2a + b^* = 2 \\
&= 1 - \frac{b^*}{2}
\end{aligned}$$

Similarly, at N.E. $B^* = 1 - \frac{a^*}{2}$
Slove these two equations in a^* and b^* gives $a^* = b^* = 2/3$

3 Problem 3: Search (20 pts)

Consider the search problem below with start state S and goal state G. The transition costs are next to the edges, and the heuristic values are next to the states.



If we use Uniform-Cost Search:

(a) What is the final path for this search?

Answer: $S \to C \to F \to G$

If we use Depth First Search, and it terminates as soon as it reaches the goal state:

(b) What is the final path for this DFS search? If a node has multiple successors, then we always expand the successors in increasing alphabetical order.

Answer: $S \to A \to D \to F \to G$

If we use A* search:

(c) What is the final path for this A* search?

Answer: $S \to C \to F \to G$

(d) Is the heuristic function in this example admissible?

Answer: Yes, since $h(s) \leq h^*(s)$ for all states s.

4 Problem 4: Hill Climbing, Simulated Annealing and Genetic Algorithm (20 pts)

An N-Queens problem is to place N Queens on an NxN chess board such that no queen attacks any other (a queen can attack any other piece in the same row, column or diagonal). Let's consider one slightly efficient complete-state formulation as below:

- State: All N queens are on the board, one queen per row and per column. In this way, we only need to worry about the attacks along the diagonal, and this simplifies the evaluation function calculation.
- Evaluation: Number of *nonattacking* pairs of queens in this state.
- Successor Function: Swap of *adjacent* columns. For example, swap (1,2) means swap the column#1 and column#2

Let's study the 5-Queens problem:

(a) Given the definition above, how many states are there in total?

Number of states: 5! *or* n!

```
. . Q . .
Q . . . .
. Q . . .
. . . Q .
. . . . Q
```

Figure 1: Initial state

```
. . Q . .    . Q . . .    . . . Q .    . . Q . .
. Q . . .    Q . . . .    Q . . . .    Q . . . .
Q . . . .    . . Q . .    . Q . . .    . Q . . .
. . . Q .    . . . Q .    . . Q . .    . . . . Q
. . . . Q    . . . . Q    . . . . Q    . . . Q .
```

Figure 2: Successors

(b) If we carry out steepest ascent hill-climbing starting from the initial state in Figure 1, what is the final state, and is it a solution? (The evaluation function values for the initial state and its four successors are given as below.)

InitState: Eval = 8
swap(1,2): Eval = 4
swap(2,3): Eval = 6
swap(3,4): Eval = 6
swap(4,5): Eval = 6

Answer: the initial state; not a solution.

(c) Consider the relations between simulated annealing and variants of hill-climbing in a general setting:

When T=∞, simulated annealing is: *E*

- A. steepest ascent
- B. stochastic hill climbing

- C. first-choice hill climbing
- D. random-restart hill climbing
- E. none of the above

When the temperature decay rate = 1, simulated annealing is: *B*

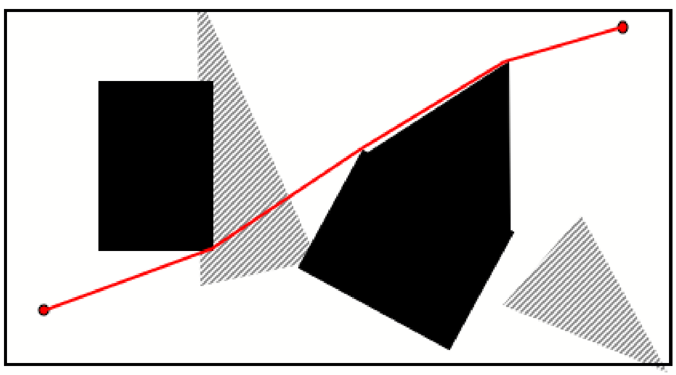
- A. steepest ascent
- B. stochastic hill climbing
- C. first-choice hill climbing
- D. random-restart hill climbing
- E. none of the above

5 Problem 5: Search and Motion Planning (20 pts)

(a) Description:

The vertices of the visibility graph are the vertices of the obstacles (black regions) plus A and B. An link is generated between 2 vertices if the straight line segment between them does not intersect any obstacle region. The path is found by using any standard optimal search algorithm (e.g., A*).

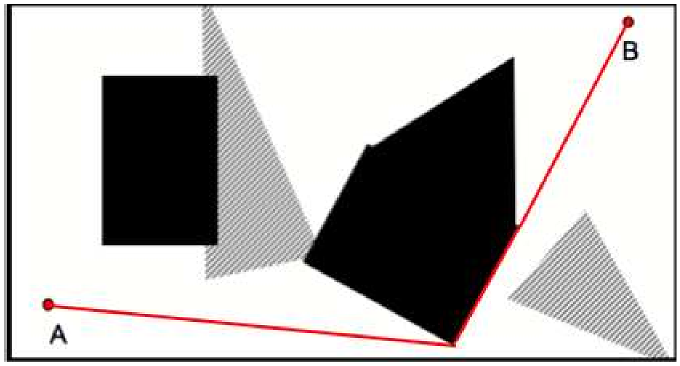
Output:



(b) Description:

The vertices of the fields of view are added to the visibility graph, which is constructed as in (a), as if the fields of view were obstacles. The best path is found as in (a).

Output:



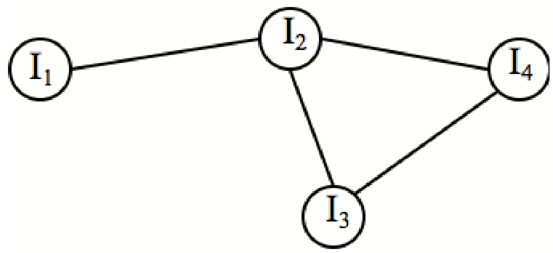
(c) As indicated there are several possible answers in this case. Listed below are three possible classes of approaches that were proposed and accepted.

Build the visibility graph by including the vertices of the obstacles and of the fields of view, but still considering only the black regions as obstacles. In the default visibility graph, the cost of an edge is equal to the distance between the two vertices linked by the edge. Modify that cost so that the edge cost is increased by an amount proportional to the length of the part of edge intersecting any of the fields of view. Effectively, this will penalize (but allow) edges that intersect the fields of view. Note that a weaker version was suggested in which the cost of an edge is computed by counting the number of intersections with fields of view regions. This will not do the job in general (although it might do a reasonable job in this particular example) because 1) an edge can be fully contained in a field of view region (zero intersections!) and 2) an edge can intersect fields of view many times and still have only a small percentage of its length inside the field of view regions.

Discretize the space by using a regular grid. The cells that intersect the black obstacles are forbidden and removed from consideration. If there were no additional fields of view constraints, all we need to do is a regular search on the graph formed by the resulting graph of cells by using a constant cost for all the cells (or more precisely, transitions between cells). Instead, we can give a higher cost to those cells that intersect the fields of view (for example, by adding the area of the cell that intersects the fields of view). The resulting algorithm is still a standard search in this grid as usual. Note that, as in any of the approximate cell decomposition methods, the algorithm is no longer guaranteed to find a path. It is guaranteed only up to the resolution of the grid.
If one prefers to use potential fields, then one possibility is to add a potential that keeps the robot away from the field of view regions, in addition to the attractive potential to the goal and the negative potential from the obstacle regions. The simplest approach would be to use the same form as the obstacle potential: $1/(distance\ to\ closest\ field\ of\ view)^2$.
Note that, as with all fields approaches, the algorithm may get stuck in a local minimum and we will need to apply the usual tricks to get out of it.

6 Problem 6: Constraint Satisfaction (20 pts)

(a) The question indicated that the variables were the instructors and the values were the rooms. The resulting constraint graph is:



Variable Instantiated	I1	I2	I3	I4
Initial Domains	R1,R2	R1,R2 R3	R1	R1,R2 R3
$I1 \leftarrow R1$ $I2 \leftarrow R2$	R1 R1	R2,R3 R2	R1 R1	R1,R2,R3 R1,R3
$I3 \leftarrow R1$	R1	R2	R1	R3
$I4 \leftarrow R3$	R1	R2	R1	R3

Final solution: (I1, I2, I3, I4) = (R1, R2, R1, R3)

b) Yes. We did take into the fact that the question turned out to be ambiguous because of the confusion constraint satisfaction/constraint propagation in grading.