Assignment 1 – Introductory Concepts

Model Answer

I. Graded Exercises

1. **[Programming Exercise – 7 Marks]** Tic-tac-toe (or Noughts and crosses, Xs and Os) is a zero-sum game where players' preferences are opposed, i.e. the winner's reward or payoff is exactly equal to the defeated player's loss. In this game, a single evaluation function is used to describe the goodness of a board with respect to both players. The given code **tictactoe.zip** contains a Matlab implementation for a 3×3 grid. In this implementation, the player who succeeds in placing three respective marks in a horizontal, vertical, or diagonal row wins the game. This playing strategy in this code is a rule-based strategy considering all the possible actions of the opponent and selecting the appropriate player's action accordingly.

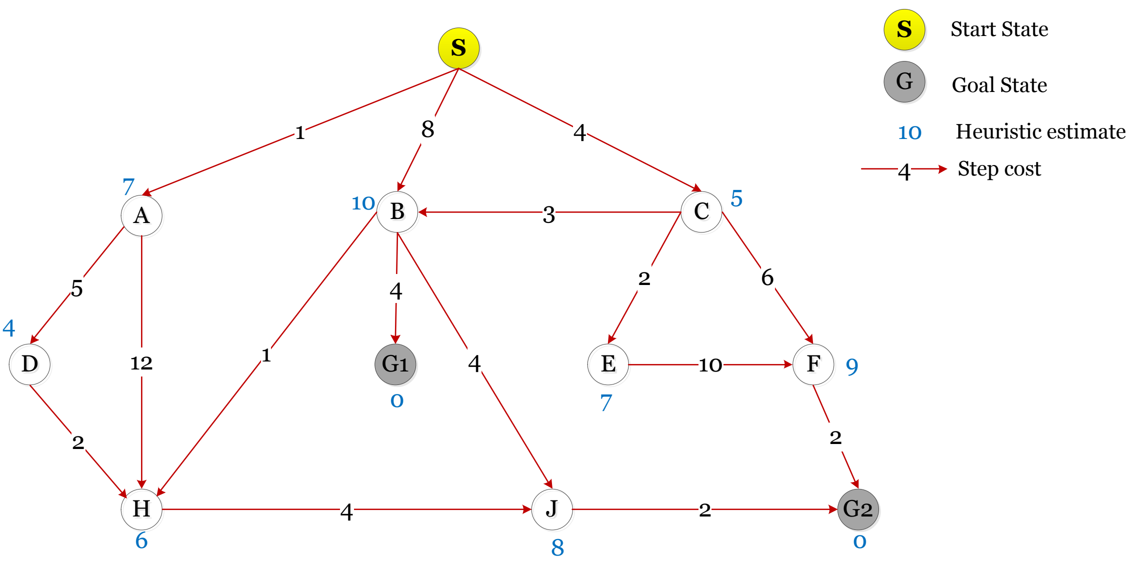
a) Modify the given code by replacing the rule-based strategy by minimax strategy.
b) Modify the implemented minimax strategy by allowing α-β pruning.

Solution [provided by course TA: Keyvan Golestan]: Matlab code is attached.

2. **[Written Exercise - 4 Marks]** Consider the search space below, where S is the start node and G1 and G2 are goal nodes. Arcs are labeled with the value of a *cost function*; the number gives the cost of traversing the arc. Above each node is the value of a *heuristic function*; the number gives the estimate of the distance to the goal. Assume that uninformed search algorithms always choose the left branch first when there is a choice. For each of depth-first search (DFS), breadth-first search (BFS), Greedy and A* search strategies,

(a) indicate which goal state is reached first (if any) and

(b) list *in order,* all the states that are popped off the OPEN list.

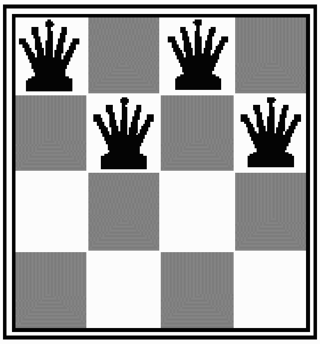


Solution: [provided by course TA: Keyvan Golestan]

Note that edge weights are not considered neither in DFS nor in BFS.

Search Strategy	a)	b)
Depth-first search (DFS) Let us assume that the nodes are expanded based on their lexicographical order (i.e. among the children of S, A is expanded first, since lexicographically, it is ordered before B and C.	G2	S, A, D, H, J, G2
Breadth-first search (BFS) The status of queue evolves such that: [S], [~~S~~,A,B,C], [~~S~~,~~A~~,B,C,D,H],[~~S~~,~~A~~,~~B~~,C,D,H,G1] Note: Since H is already visited by A, it is not pushed again to the queue when being visited by B. However, it is reached twice.	G1	S, A, B, C, D, H, H, G1
Greedy search In greedy search (Best-First-search), only the nodes with minimum edge weights are expanded.	G1	S, A, C, D, E, B, G1
A* search The order of expansion is based on the sum of edge weights g(x) and the heuristic estimation h(x), i.e., f(x) = g(x) + h(x). For example, starting from S, f(A) = 8, f(B) = 18, and f(C) = 9, therefore, the queue would look like: [A,C,B], because f(A) < f(C) < f(B). At the next step when A is popped off from the queue, nodes D and H are evaluated such that f(D) = d(S,A) + d(A,D) + h(D) = 1 + 5 + 4 = 10, and f(H) = d(S,A) + d(A,H) + h(H) = 19, and pushing them onto the queue in order will result in: [~~A~~,C,D,B,H] Therefore, the next node to be expanded would be C which adds E and F to the queue with f(E) = 11 and f(F) = 19: [~~A~~,~~C~~,D,E,B,H,F] Hence, the next node is D which causes an update on f(H)=d(S,A) + d(A,D) + d(D,H) + h(H) = 1 + 5 + 2 + 6 = 14 that subsequently reposition it in the queue: [~~A~~,~~C~~,~~D~~,E,**H**,**B**,F] The same strategy should be followed until one of the goals is found.	G1	S, A, C, D, E, H, B, G1

3. [Written Exercise - 2 Marks] N-queen problem is an example for constraint-satisfaction problem that does not define an explicit objective function. Instead, the objective is to find a solution which satisfies all of a set of constraints. In N-queen problem, the goal is to put *n* queens on an *nxn* board with no two queens on the same row, column, or diagonal. Starting from the given configuration, move the queens to reduce number of conflicts until you reach the final configuration where no queen is attacking another queen.



Solution: [provided by course TA: Keyvan Golestan]

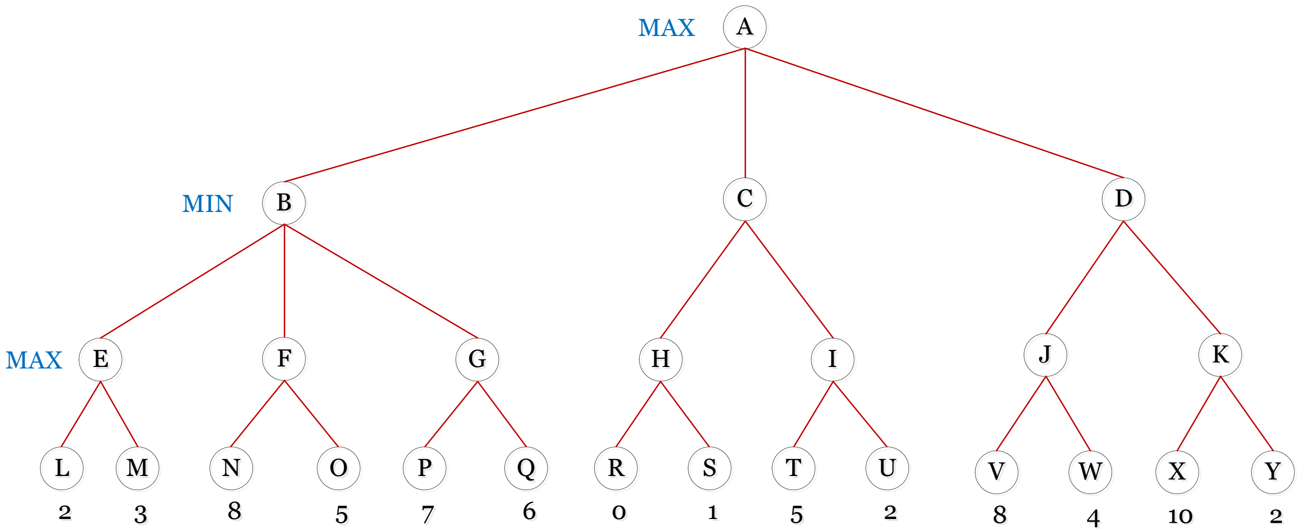
Let's assume that a pair of queens is shown by Qi and Qj in which i and j are indices of columns and the values of them are the rows on which queens reside. Therefore, we define the constraints as:

1. $i \neq j$: column conflict
2. $Q_i \neq Q_j$: row conflict
3. $|i - j| \neq |Q_i - Q_j|$: diagonal conflict

Now we can start from one of the queens and move it in a way to satisfy all the constraints. For instance, if we move the queen placed at (2, 4) one cell down and move it to (3, 4), then all the constraints between this queen and all the others are satisfied. Moving the queen at (1, 3) to any cell it can go to will not satisfy all the constraints, so we leave it at its current position for now. Checking all the moves that the queen at (2, 2) can take results in cell (4, 2) in which all the constraints of this queen is satisfied. Now if we move the queen at (1, 1) one step down and move it to (2, 1) all the constraints will be satisfied and the problem is solved.

Please note that columns satisfaction is already taken into account, and it means that the first constraint is always satisfied.

4. **[Written Exercise - 2 Marks]** Consider the following game tree in which the root corresponds to a MAX node and the values of a static evaluation function, if applied, are given at the leaves.



a) What is the minimax value computed at the root node for this tree?
b) What move should MAX choose?
c) Which nodes are not examined when α-β pruning is performed? Assume children are visited left to right.
d) Is there a different ordering for the children of the root for which more pruning would result by α-β? If so, state the order. If not, say why not.

Solution: [provided by course TA: Keyvan Golestan]

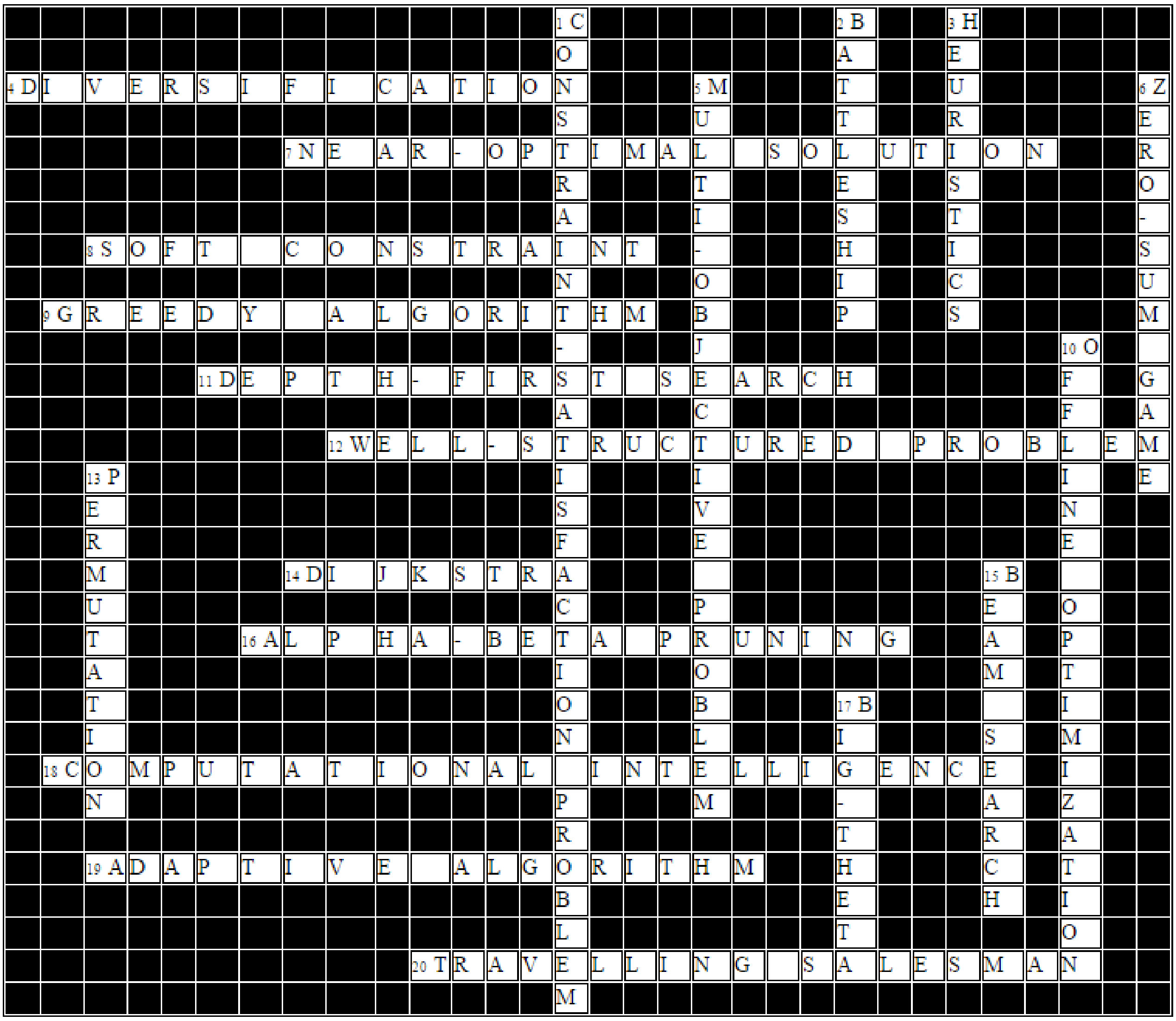
a) What is the minimax value computed at the root node for this tree? **8**
Explanation: *one level before the last level is played by MAX, therefore the values of E, F, G, H, I, J, and K are 3, 8, 7, 1, 5, 8, and 10, respectively. Moreover, one level upper is played by MIN, therefore, B, C, and D are respectively, 3, 5, and 8. The root is played by MAX, so, it chooses 8 (D) among its children.*

b) What move should MAX choose? **D**
Explanation: *it is explained in the previous part.*

c) Which nodes are not examined when α-β pruning is performed? Assume children are visited left to right. **O, Q, I, T, U, Y**
Explanation: *When the tree is expanded all the way to the left, E is evaluated 3 (MAX), and since its father (B) is playing MIN, it is not gonna take any decision that results* ***more*** *than 3, therefore the range of the values for B would be [-inf,3]. Next, when F is being evaluated, its first child (N) returns 8 that limits F's range to [8,+inf]. Now, recall that F is playing min, and no matter what other children of F return, B is not gonna take them, as it is already guaranteed with a MIN value of 3. This is why* ***O*** *is pruned and is not visited. The same intuition can be used for the other nodes as well.*

d) Is there a different ordering for the children of the root for which more pruning would result by α-β? If so, state the order. If not, say why not. **Yes, when the children are ordered D, B, C, which is best to worst ordering of A's children.**
Explanation: *with this order of expanding the children more pruning will be done by the algorithm. The reason is that the nodes will sooner achieve their final value (minimum/maximum) and this results in more prunings.*

5. **[Written Exercise - 5 Marks]** Solve the following crossword puzzle



Across

4. a form of exploration to cause the search to consider new areas. Examines unvisited regions, generates different solutions.
7. a feasible solution that provides a superior objective function value, but not necessarily the best.
8. a problem constraint that is desirable to satisfy.
9. algorithm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding the global optimum.
11. a graph search algorithm for traversing or searching tree or graph data structures. One starts at the root (selecting some arbitrary node as the root in the case of a graph) and explores as far as possible along each branch before backtracking.

Down

1. a problem that does not define an explicit objective function. Instead, the objective is to find a solution which satisfies a set of constraints.
2. a guessing game for two players where not all state information is available to all players and outcome of actions is deterministic.
3. a solution strategy or rules by trial-and-error to produce acceptable (optimal or near-optimal) solutions to a complex problem in a reasonably practical time.
5. a problem that specifies more than one sub-objective which need to be

12. a problem that can be stated in terms of numerical variables. Its goals can be specified in terms of a well-defined objective function and there exists an algorithmic solution for it.
14. a graph search algorithm that solves the single-source shortest path problem for a fully connected graph with nonnegative edge path costs, producing a shortest path tree.
16. a procedure which can prune large parts of the search tree and allows search to go deeper.
18. a part of computer science devoted to solution of non-algoritmizable problems.
19. any algorithm able to adjust to new or different situations or to improve behaviour or evolves and learns from instructor, example or by discovery.
20. an NP-hard problem.

simultaneously optimized.

6. is a game where the players' preferences are opposed. The winner's reward or payoff is exactly equal to the defeated player's loss.
10. an optimization technique in which time is not so important and a user is willing to wait maybe even days if he/she can get an optimal or close-to-optimal result.
13. an ordered combination.
15. an informed breadth-first algorithm.
17. Asymptotic notation used to describe asymptotically tight bounds of a computer algorithm.

Hint: Spaces and dashes MUST be used if the answer consists of two or more words.