MA8.401: Topics in Applied Optimization			Marks obtained ↓
Date: 7.11.2024	Total questions: 4	Total points: 30	
Roll No:		Time: 60 Minutes	

Don't submit the question paper. No questions will be entertained during exam. If you have a doubt, please write the doubt in answer sheet and answer the best as you can.

6 1. Short Answer Questions (half page answers!).

1. Write dynamic optimization (DP) algorithm for deterministic finite horizon problems.
2. Write dynamic optimization (DP) algorithm for stochastic finite horizon problems.
3. Describe policy iteration for infinite horizon.
4. Describe multiagent problem and multiagent rollout.
5. Describe how DP algorithm is modified to handle time delays, i.e., when the current state x_k depends not only on the preceding state x_k and control u_k, but also on earlier states and control.
6. Describe how dynamic optimization (DP) algorithm is modified to handle forecasts.

10 2. Consider a modified version of the four-city traveling salesman problem (done in class), where there is a fifth city E. The intercity travel costs are shown in Fig. 1. The cities are denoted by A, B, C, D, E. The tour starts from city A. For example, the first row in matrix in Fig. 1 indicates the cost to go from city A to other cities B, C, D, E.

	A	B	C	D	E
A	■	5	1	20	10
B	20	■	1	4	10
C	1	20	■	3	10
D	18	4	3	■	10
E	30	10	0	10	■

Figure 1: Intercity cost

1. Use exact dynamic optimization (DP) with starting city A to verify that the optimal tour is $ABDECA$ with cost 20.
2. Verify that the nearest neighbor heuristic starting with city A generates the tour $ACDBEA$ with cost 48.
3. Apply rollout with one-step lookahead minimization, using as base heuristic the nearest neighbor heuristic. Show that it generates the tour $AECDBA$ with cost 37.

7 3. Describe an approach (any of neural network or least squares approach) for training an approximation architecture $\tilde{J}_k(x_k, r_k)$ for finite horizon DP problem. When training such approximation architectures, what are the labels in training data? How can an existing policy improve if the training architectures for cost function approximation use existing labels? For example, if the labels (that is cost function estimate for a given state) are obtained from existing best player of Tetris for those states, and these labels are used to build the approximation architecture for cost prediction, then how can one improve on the performance of existing best player?

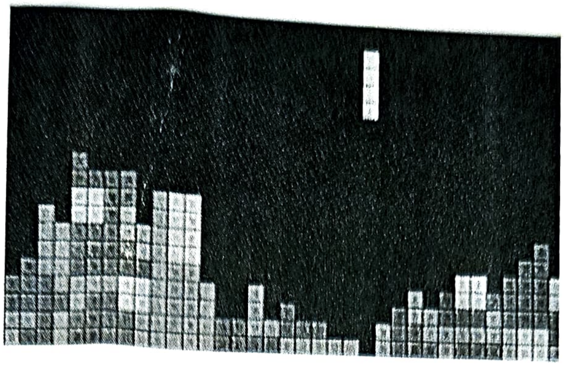

Figure 1.6.3 Illustration of a tetris board.

Figure 2: Tetris Game.

$\boxed{7}$ 4. Tetris is a popular video game played on a two-dimensional grid. Each square in the grid can be full or empty, making up a "wall of bricks" with "holes" and a "jagged top" (see Fig. 2). The squares fill up as blocks of different shapes fall from the top of the grid and are added to the top of the wall. As a given block falls, the player can move horizontally and rotate the block in all possible ways, subject to the constraints imposed by the sides of the grid and the top of the wall. The falling blocks are generated independently according to some probability distribution, defined over a finite set of standard shapes. The game starts with an empty grid and ends when a square in the top row becomes full and the top of the wall reaches the top of the grid. When a row of full squares is created, this row is removed, the bricks lying above this row move one row downward, and the player scores a point. The player's objective is to maximize the score attained (total number of rows removed) up to termination of the game, whichever occurs first. Formulate this problem as dynamic optimization (DP), describing clearly each terms. What is the state space dimension for 20×10 tetris board? What are important features that can be extracted for states?