İTÜ

Homework 2

BLG435E

Optimization and Learning

The following questions are about optimization and policy learning. The first question is a Traveling Salesman Problem(TSP) and you will use Genetic Algorithms(GA). In the second question, you will use Q-Learning. In the last question, you train an agent to play a game using Deep-Q-Networks(DQN). Write necessary codes and provide the required analysis. Cite any outsourced material that you used. Write a report that includes your results and analysis. Give insights about your approach. Explain the results you obtained in detail.

You are free to use available implementations on the web for DQN if you can explain the code in the report and if you give proper citations.

Problem 1 - Traveling Salesman Problem - Optimization Algorithm Solutions (30 points)

The Traveling Salesman Problem(TSP) is a famous optimization problem. Given a graph with weighted edges, a solution is a path on the graph that visits each node exactly once and ends in the starting node. The cost of the path is the sum of the weights between nodes on the path. The problem is to the solution with the shortest path.

Greedy solutions that guarantee the shortest path have exponential time complexity making the problem NP-complete. Therefore some other optimization algorithms can be used such as Simulated Annealing(SA) or Genetic Algorithms(GA) etc.

As an example consider the graph in Figure 1, example solutions can be:

- C1-C3-C2-C4-C1, cost: 10
- C2-C4-C3-C1-C2, cost: 12

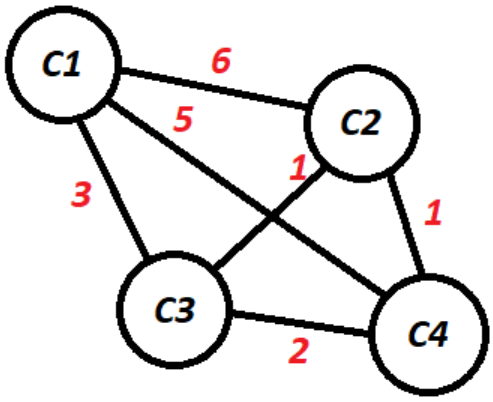


Figure 1: Example Graph for TSP

Homework 2

BLG435E

In this problem, you are given a file(cities12.txt) consisting of cities as nodes with their coordinates on a 2D map. Weights are the Euclidian distance between cities.

Choose one of the algorithms below for this optimization problem:

- Simulated Annealing (SA)
- Genetic Algorithm (GA)

Implement an SA-based or GA-based solver to solve the TSP problem.

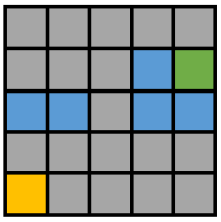
Provide the following results in your report:

- Explain your approach and justify the suitability of your algorithm(SA or GA) for this problem.
- Implement 3 different versions of your selected algorithm(different parameters such as temperature decrease rate in SA or different component such as elitism in GA). Name these algorithms as
  - SA-"NAMEX" or
  - GA-"NAMEX"

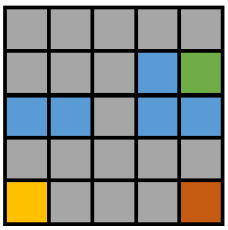
  You will use these versions to investigate the effect of the parameter or the component. Explain your approaches.
- Collect performance measures and create graphs showing the results (such as a cost-time graph).
- Show how your different versions of selected algorithms performed using the performance measures and graphs.
- Analyze and justify your results.

Problem 2 - Q-Learning (40 points)

In this problem, you are given a path planning problem in a grid world as illustrated in Figure 2a. In this problem, a robot starts its movement on the yellow tile with the goal of reaching the green tile. But passing through blue tiles(water) is hazardous for the robot, therefore the robot should learn to navigate in this environment safely.



(a) Vanilla Grid World Environment



(b) Jetpack Grid World Environment

Figure 2: Grid World Environments. Yellow: Start, Green: Goal, Blue: Water, Orange: Jetpack

a. Show and explain how Q-Learning Algorithm works by updating the Q-Values with the given trajectories below.

- U-U-U-U-R-R-R-R-D
- R-R-U-U-U-U-R-R-D

Plot a Q-Table with random initial values where a policy from it can not reach the goal. Then use the trajectories above to update the q table. After that, show how the q values changed and the new policy from it that presumably can reach the goal.

b. Propose a Q-Learning-based solution to obtain a safe robot policy reaching the goal without passing through water. Use e-greedy exploration. Show your approach(rewards, q-values, updates, etc.) in the report. Report all parameters that you set such as gamma. Sketch the resulting policy. Present a reward-step graph, analyze and justify your results.

c. Considering a jetpack in the orange tile as in Figure 2b, the robot can use the jetpack to move up to 2 tiles at a time(either 1 or 2 tiles without turning) and can not be affected by the water. Show how the search space changes with the new environment. Propose a Q-Learning-based solution to obtain a safe robot policy reaching the goal without passing through water for the new environment. Use e-greedy exploration. Show your approach(rewards, q-values, updates, etc.) in the report. Sketch the resulting policy. Present a reward-step graph, and analyze and justify your results. Compare performance metrics(memory usage, learning time) of the training in **a** and **b**.

Problem 3 - Deep Q Networks (30 points)

Implement a DQN agent using pytorch and prepare a reinforcement learning loop to train your agent. Use the environment(pygame-learning-environment) given to you through Ninova. Train a DQN agent for the Flappy Bird game using e-greedy exploration.

Use examples given in the examples folder to understand how to use the environment.

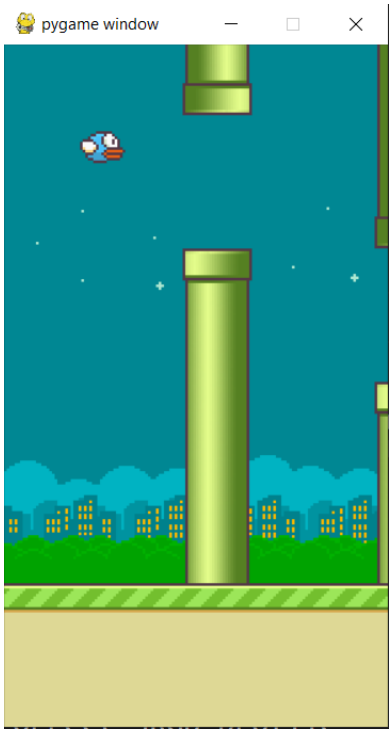


Figure 3: Flappy Bird environment

- Explain briefly the details of the DQN agent in your report.
- Include your training and evaluation results. These results should include a reward-episode graph.
- Investigate the effect of e-greedy exploration by three different versions of algorithms where e = 0.5, 0.1, 0.01.
- Investigate the effect of linearly decaying e-greedy exploration
- Use plots. To compare e-greedy versions of the DQN, plot their train results on one graph and their test on one. Analyze and justify your results.

Implementation and Submission Details

Submit your homework files through Ninova. Upload all your solutions as BLG435E_HW_2_STUDENTID.zip. You are going to submit:

1. Codes that include **your**
   - Optimization Algorithm Solution
   - Q-Learning Solutions
   - DQN Solutions
2. A detailed report explaining **your** approach, implementation, and findings(details are given in each question.)

Please pay attention to the deadline (date/time) of the homework. **Late submissions will not be accepted, without any exception**.

In case of any questions, feel free to send an e-mail to akab@itu.edu.tr.