Solving Prize-Collecting Traveling Salesman Problem Using Reinforcement Learning

1. Background:

A traveling salesman problem (TSP) is defined as follows. Given a graph with edges and vertices/cities, where each edge has a weight, indicating the traveling cost on this edge. And given a traveling salesman in one of the nodes as its starting node/city, he looks to visit each of the node/city and comes back to its starting city with minimum total traveling cost.

TSP is NP-hard, which means there is no efficient and optimal algorithm to solve it.

- There is a famous minimum-spanning tree based 2-approximation algorithm

https://iq.opengenus.org/approximation-algorithm-for-travelling-salesman-problem/

https://www.geeksforgeeks.org/travelling-salesman-problem-set-2-approximate-using-mst/

- There is a simple greedy algorithm to solve TSP, which he visits the closest unvisited city, until all the cities are visited and comes back home (i.e., starting city)

IN **prize-collecting traveling salesman problem (PC-TSP)**, though, each city has a prize to collect, and there is a quota, which is the targeted amount of prizes he wants to collect by visiting the cities. For PC-TSP, the goal is to start from its starting city, visit a sequence of cities to collect the prizes at each visited city to collect the quota amount of prizes, and comes back.

When all the prizes are the same at different cities, it becomes the k-stroll problem, which asks to visit k distinct nodes and then comes back to the starting city with minimum cost. K-stroll is solved by a dynamic programming algorithm in below paper:

https://csc.csudh.edu/btang/papers/kstroll.pdf

2. Reinforcement learning

In this project, you will learn reinforcement learning and apply it to solve the classic traveling salesman problem, compare with the 2-approximation and greedy algorithm for TSP, and DP-algorithm in PC-TSP.

A starting point for this project is:

https://sagemaker-examples.readthedocs.io/en/latest/reinforcement_learning/rl_traveling_salesman_vehicle_routing_coach/rl_traveling_salesman_vehicle_routing_coach.html

Sagemaker is Amazon’s cloud machine learning platform, which enables developers to create, train, and deploy machine learning models in the AWS cloud.

At the most basic level, SageMaker provides **Jupyter notebooks**, a popular opern-source, web-based interactive computing platform for data science analysis and visualization. Using it, data scientists can create and share documents that integrate live code, equations, computational output, visualizations, and other multimedia resources,

For reinforcement learning, a few good links are at:

https://github.com/dennybritz/reinforcement-learning

https://neptune.ai/blog/best-reinforcement-learning-tutorials-examples-projects-and-courses

3. Deep Reinforcement learning

For large-scale TSP with thousands of nodes, RL becomes less efficient as the states and actions available becomes extremely large. Also, it is intractable to use any optimization algorithms such as dynamic programming to solve optimally. That’s how deep reinforcement learning comes into play:

https://aws.amazon.com/blogs/opensource/solving-the-traveling-salesperson-problem-with-deep-reinforcement-learning-on-amazon-sagemaker/