Beihang University
School of General Engineering

Beijing, June 1st, 2021
Lin Wei and Yida Ding

Data Management and Artificial Intelligence Lab Class 13

Task 1 ***Buffon's Needle****(30 minutes)*

In mathematics, Buffon's needle problem is a question first posed in the 18th century by Georges-Louis Leclerc, Comte de Buffon.

Suppose we have a floor made of parallel strips of wood, each the same width, and we drop a needle onto the floor. What is the probability that the needle will lie across a line between two strips?

The solution for the sought probability p, in the case where the needle length l is not greater than the width t of the strips, is

$$p = \frac{2 * l}{\pi * t} \tag{1}$$

In this task, you are going to use Monte Carlo method together with the above solution to estimate value π. The subtasks are shown as follows:

1. Create a class *Board* and initialize the class with the length and width of the board, the width of the strips t, the length of each needle l, and the random seed. Note that l should not be greater than t; for the solution under the case of $l > t$, please check https://en.wikipedia.org/wiki/Buffon if you are interested.

2. Randomly generate N needles within the board and record the position of each needle. (Think about how to represent the position of each needle)

3. Visualize the needles by the recorded positions, as well as the board, as shown in Figure 23.

4. Calculate the number of needle-strip crosses, and also evaluate their positions.

5. Estimate π using the number of needle-strip crosses and the solution formula.

6. Visualize all of the needle-strip crosses and the estimated π as title, as shown in Figure 23.

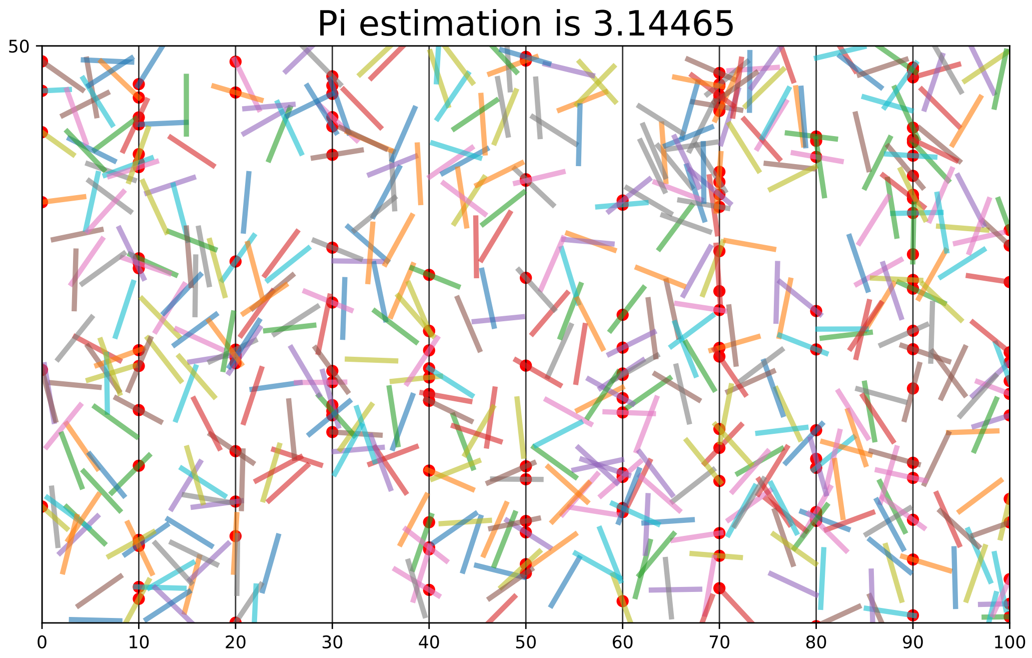

Figure 23: Task 1

Task 2 ***Monte-Carlo Tree Search for Traveling Salesman Problem*** *(60 minutes)*

In this exercise, we want to implement a solution algorithm based on Monte-Carlo Tree Search (MCTS) for the traveling salesmen problem (TSP). The problem of TSP asks for the shortest round-trip between a collection of points. For the last lab class, you implemented a solution based on GA, which took quite a while to converge. We want to see how MCTS performs on this problem. For the beginning, we focus on a small problem of TSP over points on a circle, where the solution is simply to visit all nodes (counter-)clock-wise. Please perform the following steps, in order to obtain a solution:

1. Implement a function which generates N evenly distributed points on a circle with radius 1. The function should return a list of tuples, where each tuple has a (x, y) shape.

2. Implement a function which computes a distance data structure to collect distances between all points. The function should receive the points as parameter and return a dictionary/list/array as a result (just as in the GA solution).

3. Implement a function which computes the length of a solution (=sequence of points). The function receives points, distances, and a specific order among points (=list of points) to be assessed.

4. Implement a function which visualizes a solution with matplotlib. Consecutive points should be connected and the title of the chart should state the length of the visualized solution.

5. Implement a class MCTNode to represent a node in a Monte-Carlo Tree. In this class, you should record the path from root node to this node, the total reward, number of simulation involving this node, the UCB value and the children of the current node.

6. Implement the a MCTS-based solution algorithm for TSP, following the general structure with four sub-steps in each iteration:

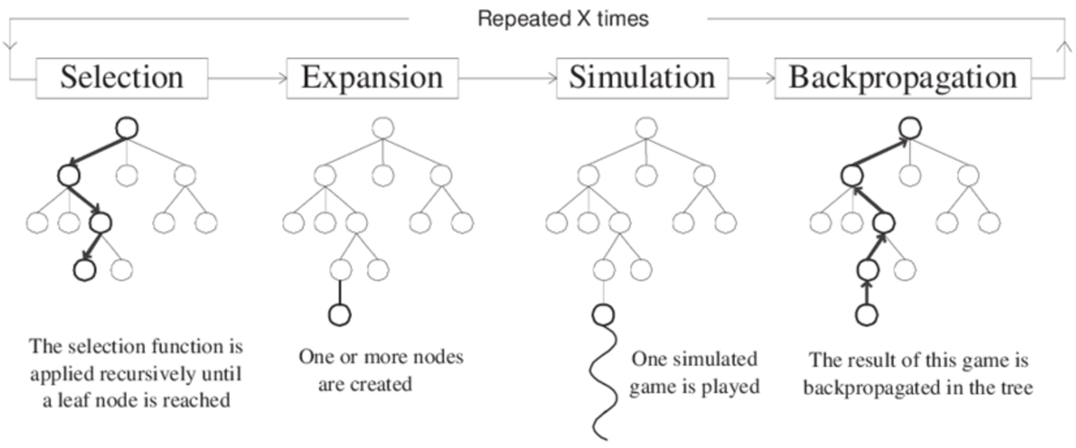
Repeated X times
Selection
Expansion
Simulation
Backpropagation
The selection function is applied recursively until a leaf node is reached
One or more nodes are created
One simulated game is played
The result of this game is backpropagated in the tree

Figure 24: MCTS for Task 2.