Week 5

Naho Yokoyama

Overview

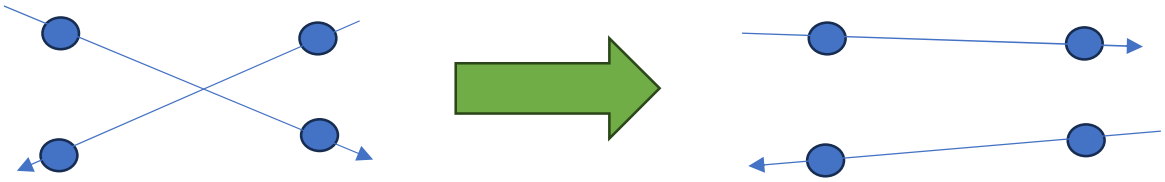
This program finds the shortest path that starts from point 0, goes through all points, and returns to the original point 0. It uses three main techniques: the greedy method, 2-opt, and simulated annealing.

Greedy Method

The greedy method is an approach where, each time the program chooses the next point to visit, it selects the unvisited point closest to the current location. This method will be used as the initial solution for other methods (like a baseline).

2-opt

2-opt is a method where, each time the program finds two intersecting paths, it rearranges the four points so that the paths no longer intersect. This reduces the total length.



Simulated Annealing

Simulated annealing is a method for reconnecting the pathways. In each trial, I randomly choose two points on the pathway and swap their positions on the tour. If this reduces the total length, I keep the new tour. If it increases the total length, I accept the result only with a certain probability, which decreases as I repeat the trials. I repeated the trials for 60 seconds.

This method should work, but in my implementation, the result only improves when I reject all trials that increase the total length. There may be some mistakes in my implementation that I cannot find right now.

The result of combining techniques

Table 1. Experiment result for comparing the three techniques for input_6.csv

2-opt	annealing	annealing2	result
			49892.05
☑			46576.49
	☑		826260.80
		☑	48627.02
☑		☑	45982.10

Table 1 shows the results of a comparison of the three techniques using input_6.csv. From the sample data, I know that the greedy method works better than random trials. Therefore, I compared combinations of the other two techniques. I found that my annealing method did not improve the total score. Consequently, I modified the program so that my annealing method would not accept any modification that did not improve the total distance. After this change, I noticed a slight improvement in the score. Finally, I combined the two methods, and this yielded my shortest path.