Greedy Algorithm Drasil Case Study

Don Chen

March 1, 2021

This is the proposal of greedy algorithms/heuristics to my drasil case study.

1 Problem Description

- Use greedy heuristics to find the "smallerst" number in a tree. The algorithm can be summarized as:
  - At each node, pick out the note contains the smallest result.
  - Repeat first step until reachs to any leaf.
- The Traveling Salesman Problem(TSP): Find the shortest distance tour passing through each node of the network exactly once.
  - Nearest Neighbor Alorithm
    - Start at any city
    - Visit the nearest node not yet visited
    - Return to the start node when all other nodes are visited
  - Kruskal's Algorithm
    - Select the miminum weight of edge if the edge doesn't form a circle
    - Stop when all node is connected
    - Kruskal's Algorithm Wiki

2 Goal Statements

- Find relatively a good solution with less costly computation. This maybe too general, how to define a good solution?
- Find the optimal solutiuon for The Traveling Salesman Problem.

3 Assumptions

Assume all type are number. Greedy heuristics work well in a quantified environment, but not all type can be quantified. Some type could be hard to be quantified, like boolean. In decision tree, some nodes will ask yes or no quesiton.

4 Theoretical Model

5 General Definition

G is a connected graph with n vertices, T is minimal connected spanning of G.

6 Supporting Data Definitions

7 Instance Models

- Tree
  - Input: Start at the top node
  - Output: Number
- Traveling Salesman Problem
  - Input: Start anywhere
  - Output: Number

8 Question

- (may not relevant now)Greedy algorithms/heuristics is more like a technique rather than an algorithm. Are we looking for two specific algorithm? The reason I aked this question is that it is eaiser to list a problem, then try to find a algorithm to solve the problem.
- greedy heuristics impact on wild animal. Animal cognitive, Migrate bird. Pigeon experiment

9 Reference

- Nearest Neighbor Alorithm
- Kruskal Alorithm in TSP