Simulating a Real World Problem

BY

KURISETI RAVI SRI TEJA - 2019CS10369
GATTU KARTHIK - 2019CS10348

Contents

1 Introduction **2**
1.1 Background Of The Problem 2
1.2 Creating a Mathematical Model 2

2 Notations Used **2**

3 Runtime and Hardness Analysis **3**

4 Solution to the Problem **3**

5 Some Possible Heuristics **5**

6 Pre-Processing the Graph for Heuristics **5**

7 Algorithms for the Heuristics **6**
7.1 Heuristic-1 6
7.1.1 Time-Complexity Analysis 7
7.2 Heuristic-2 7
7.2.1 Time-Complexity Analysis 8

8 Simulating the Problem Using a Maze **8**

1 Introduction

1.1 Background Of The Problem

In the recent days during the spread of COVID-19 second wave, we have seen that there was a shortage of medical grade oxygen at many hospitals due to which many patients had lost their lives. The main reason for the shortage of oxygen in the hospitals was due to the fact there was a lot of delay in the supply of oxygen from the production plants to the Hospitals.In this document we try to propose an algorithm to reduce the time taken for oxygen supply from M plants in a city to N hospitals using a single Oxygen-tanker i.e., propose an algorithm for the shortest path the tanker should take to ensure a faster delivery i.e., it can start at any of the plant and end its journey at some hospital after making a final delivery.

1.2 Creating a Mathematical Model

We assume that there are M production plants P_1,P_2,....P_M that produce oxygen.The oxygen produced is to be transported to N hospitals H_1,H_2,....H_N with oxygen demands of R_1,R_2,....R_N metric tonnes with H_j corresponding to the demand of the j^{th} hospital.The supply needs to be done using a vehicle that can carry oxygen and has capacity C metric tonnes.

We can model this problem using an undirected graph with M+N vertices i.e., M sources and N destinations with edges(weights are all clearly of same sign can be taken to be positive) as the road network connecting all the plants and hospitals subject to **minimizing the total distance** covered by the oxygen tanker transporting the oxygen from plants to destinations i.e., it is a **variant of the shortest problem** with M sources and N destinations and with **additional constraints** of the demand and supply at each source and destination.

2 Notations Used

We denote the production plants as P_1,P_2,....P_M ,hospitals as H_1,H_2,....H_N, initial requirements as R_1,R_2,....R_N, requirements of hospitals after some

time as $R_1{}'$,$R_2{}'$,....$R_M{}'$.

3 Runtime and Hardness Analysis

To solve this problem completely we need to dynamically generate all possible states from the given state and recursively solve all the generated subproblems which takes O($M(N+M)^b$) time where b is the maximum depth of recursion tree which is atleast N+M.Hence this problem is **NP-Complete** because the time complexity of this problem is Non-Polynomial and this problem can be solved by brute-force techniques (although it needs heavy computational power and large amounts of memory).

4 Solution to the Problem

Since the problem is NP-Complete,it takes large amount of time and computational power to solve the problem completely.Hence we try to obtain approximate solutions or use some other methods which try to provide an approximate solution using heuristics.Since we can generate a state-transition graph dynamically at each step that can generate all possible transitions at that state,we can perform an **A* Search** over the state-transition graph(tree in this case) to find an optimal solution by choosing a desired **heuristic function**.

Algorithm A^* Search

```
1: procedure TRACE_PATH(cameFrom,current)
2:     total_path=[current]                          ▷ Initializing the Path Array
3:     while current in cameFrom.keys() do
4:         current ←cameFrom[current]
5:         total_path.prepend(current)
6:     return total_path
7: end procedure
```

```
1: procedure A* SEARCH(start,goal,h)
2:     openSet={start}                               ▷ Initializing a Priority-Queue
3:     cameFrom={}                                   ▷ Initializing an Empty Map
4:     F={}                          ▷ Initializing an Empty Map F for f(n) values
5:     G={}                ▷ Initializing an Empty Map G for storing g(n) values
6:     F[start] ← h(start)
7:     G[start] ← 0
8:     while openSet is not empty do
9:         current ←openSet.getMin()
10:        if current = goal then
11:            return Trace_Path(cameFrom, current)
12:        openSet.deleteMin()
13:        for u reachable from current do    ▷ Finding next possible states
14:            tentative ← G[current] + d(current,u)
15:                                  ▷ Finding distance to next possible states
16:            if tentative < G[u] then
17:                cameFrom[u] ← current
18:                G[u] ← tentative
19:                F[u] = G[u] + h(u)
20:                if u not in openSet then
21:                    openSet.add(u)
22:    return PATH_NOT_FOUND
23: end procedure
```

5 Some Possible Heuristics

We try to propose two heuristic models to compute the value h(n) where n is any given state for the above problem.
In the first heuristic we are taking the following three assumptions.

1. Graph is bipartite with two partitions.The first partition contains all the source vertices and the second partition contains all the destination vertices.

2. Capacity of the tanker is infinite.

3. The route of the tankers consists of only single plant and all the plants are of infinite production-capacity.

In the second heuristic we are taking the size of the tanker to be finite(= C) i.e., reducing the number of assumptions from three to two.Hence the assumptions in this heuristic are as follows

1. Graph is bipartite with two partitions.The first partition contains all the source vertices and the second partition contains all the destination vertices.

2. The route of the tankers consists of only single plant and all the plants are of infinite production-capacity.

6 Pre-Processing the Graph for Heuristics

We initially process the original graph G to convert into the desired bipartite sub-graph G'.For this we iterate over all the edges of the graph which takes $O(|E|)$ time where $|E|$ is the total number of edges in the original graph i.e., pre-processing takes $O(|E|)$ time. After this we check for each destination the closest source in G' which is done in O(MN) time(Because for each of N destinations there are most M reachable sources).But we know that $O(|E|)= O(|V|^2) = O((M + N)^2)$.Hence total pre-processing time is of the order $O(M^2 + N^2 + MN)=O((M + N)^2)$.

7 Algorithms for the Heuristics

7.1 Heuristic-1

There are two possible sub-cases in this heuristic.

Tanker is at a Plant:- In this case,to reach the goal,we need to transport the oxygen to all the destinations where there is necessity of oxygen and the only way to do it is to travel to each hospital where oxygen is to be delivered and come back to plant and go to next destination and so on till requirement of all destinations is fulfilled i.e.,

$$h(n) = (2\sum_{j} d_{ij}) - \max_{\{j\}}(d_{ij}) \tag{1}$$

(Assuming that the tanker is at i^{th} plant and summation is over such j's where the hospital H_j hasn't received the oxygen equal to its demand)

Tanker is at a Hospital:- In this case we go to the closest plant(because hospitals are not connected from each other) and then compute the value of heuristic-function assuming that the tanker starts at that plant and add it to the distance travelled by tanker to reach the plant from the destination to compute the final value.

$$h_1(n) = \min_{\{i\}}(d_{ij}) \tag{2}$$

(Here tanker is at hospital H_j)

$$h_2(n) = (2\sum_{j} d_{i_0 j}) - \max_{\{j\}}(d_{i_0 j}) \tag{3}$$

$$h(n) = h_1(n) + h_2(n) \tag{4}$$

(Here i_0 is the value of i such that d_{ij} is minimum and the summation is over such j's where the hospital H_j hasn't received the oxygen equal to its demand)

7.1.1 Time-Complexity Analysis

Suppose if the tanker is at a plant, it takes $O(N)$ time to find out all the hospitals where there is still requirement and hence computation of H1 will be $O(N)$. Suppose if the tanker is at hospital,due to pre-processing it takes $O(1)$ to find distance to closest plant and $O(N)$ to reach to goal from that plant. Hence computation of H1 takes $O(N)$ time for any state.

7.2 Heuristic-2

There are two possible sub-cases in this heuristic.

Tanker is at a Plant:- In this case,to reach the goal,we need to transport the oxygen to all the destinations where there is necessity of oxygen and the only way to do it is to travel to each hospital where oxygen is to be delivered and come back to plant and go to next destination and so on till requirement of all destinations is fulfilled.In Heuristic-1 the capacity of the tanker is infinite,hence we can take all the required oxygen in single step to the destination.But here the size of the tanker is finite,hence we need to take multiple steps i.e., travel back and forth between plant and hospital.

$$h(n) = (2\sum_{j} d_{ij} \left\lceil \frac{R_j{}'}{C} \right\rceil) - \max_{\{j\}}(d_{ij} \left\lceil \frac{R_j{}'}{C} \right\rceil)$$

(Assuming that the tanker is at i^{th} plant and summation is over such j's where the hospital H_j hasn't received the oxygen equal to its demand)

Tanker is at a Hospital:- In this case we go to the closest plant(because hospitals are not connected from each other) and then compute the value of heuristic-function assuming that the tanker starts at that plant and add it to the distance travelled by tanker to reach the plant from the destination to compute the final value.

$$h_1(n) = \min_{\{i\}}(d_{ij}) \tag{5}$$

(Here tanker is at hospital H_j)

$$h_2(n) = (2\sum_{j} d_{i_0 j} \left\lceil \frac{R_j{}'}{C} \right\rceil) - \max_{\{j\}}(d_{i_0 j} \left\lceil \frac{R_j{}'}{C} \right\rceil)$$

$$h(n) = h_1(n) + h_2(n) \tag{6}$$

(Here i_0 is the value of i such that $d_{ij}\left\lceil \frac{R_j{}'}{C} \right\rceil$ is minimum and the summation is over such j's where the hospital H_j hasn't received the oxygen equal to its demand)

7.2.1 Time-Complexity Analysis

Suppose if the tanker is at a plant, it takes $O(N)$ time to find out all the hospitals where there is still requirement and hence computation of H1 will be $O(N)$Suppose if the tanker is at hospital,it takes $O(N)$ to find the required plant satisfying the constraint in equation-5 and $O(N)$ to reach to goal from that plant.Hence computation of H2 also takes $O(N)$ time for any state.

8 Simulating the Problem Using a Maze

We can generate a maze by using Maze-generating algorithms and place the M sources and N destinations at some random points by labelling them as S and D or using distinct Pacman Ghost Textures. We use a different texture to represent the tanker and it starts from some random source and visits all the possible destinations and retraces its path, checking for locally optimal path suggested by the heuristic and proceeds until it completes delivering to all the possible destinations.We colour the already visited path with green and then checking for the best positions with blue colour and then finally add the optimal path to the initial green path.