Home assignment report – Semenkina Olga

Network flow model

Let's assume that drivers can work 24/7, since in the request data the difference between the first and last request is almost a day. Let each node in the model, except for the depot, represent an event defined by location, time, and request id. Let N be a car number. For example, we have 4 requests (as on the plot):

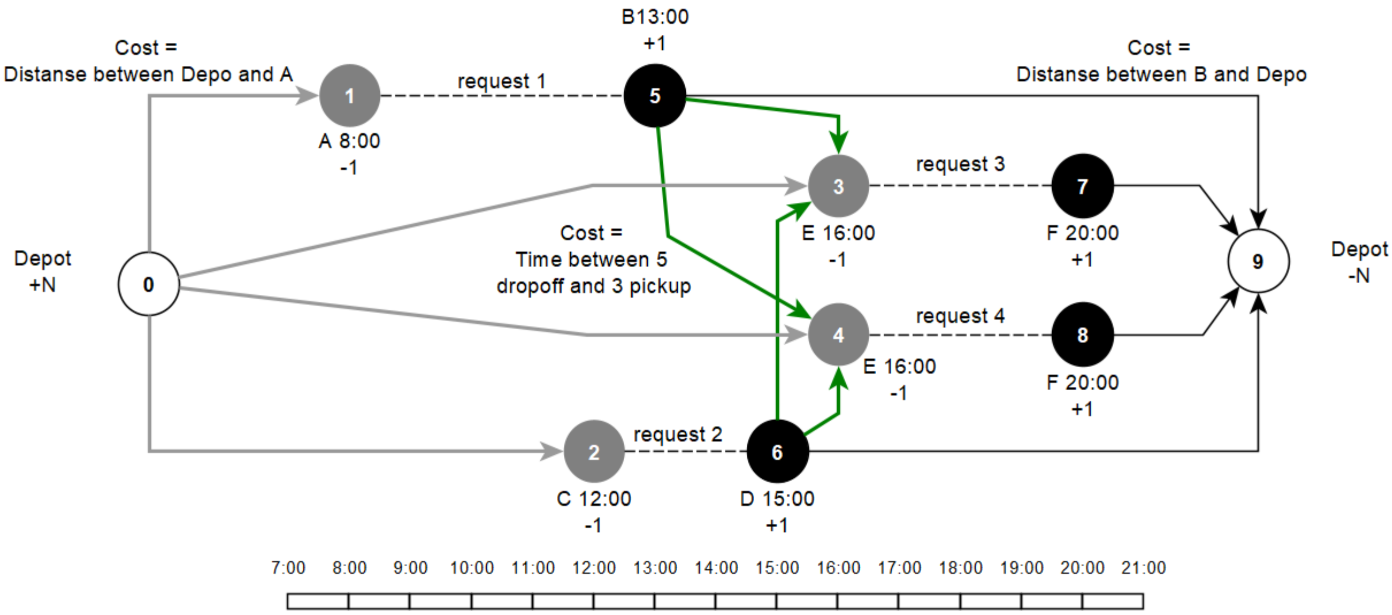
1. From point A at 8:00 to point B with a duration of 5 hours.
2. From point C at 12:00 to point D with a duration of 3 hours.
3. From point E at 16:00 to point F with a duration of 4 hours.
4. From point E at 16:00 to point F with a duration of 4 hours.

The two last requests are identical except for their id and that is why we have 4*2+2=10 different nodes. There are 4 types of nodes:

1. Depot from which cars leave. This is a **source (supply node)** with the number +N (+30).
2. Depot where cars return. This is a **sink (demand node)** with the number -N (-30).
3. Request pickup node. This is a **demand node** with the number -1.
4. Request dropoff node. This is a **supply node** with the number +1.

Because pickup and dropoff nodes are sources or sinks, all requests will be guaranteed to be served and included in one of the car routes. Each edge has a capacity equal to 1, since one and only one vehicle must be assigned per request. There are no edges between the pickup and dropoff nodes of the same request in the graph. So, there are 3 types of edges:

1. An edge between the depot and a request pickup node. The cost of this edge is equal to the duration of the ride between these 2 locations. On the plot, we can find an example of the edge between depot node 0 and node 1 that represents a pickup node for request 1.
2. An edge between a request dropoff node and depot. The cost of this edge is equal to the duration of the ride between these 2 locations. An example is the edge between node 5 and depot node 9.
3. An edge between one request (R1) dropoff node and another request (R2) pickup node. This node exists if and only if the duration of a ride between the location of R1 dropoff and the location of R2 pickup is less than the time between these two events. The cost of this edge is equal to the time between these events. For example, the edge between node 5 and node 4 exists because there are 3 hours difference and the duration of the ride between in my test case is less than 3 hours. If the ride time between them was more than 3 hours, then this edge would not exist in the graph. For the same reason, there is no edge between nodes 5 and 2 since it is impossible to finish request 1 and be in time for the beginning of request 2.



Bonus questions

1. Suppose you were only given the two datasets but not the number of vehicles in the service. Could you use this information to provide an estimate of the number of vehicles required?

I can suggest several approaches to solving this kind of problem.

> 1. Create a Gantt chart of all requests and estimate its maximum width, which will mean that no less than this number of machines will be required to service all orders.
>
> 2. Use a greedy strategy, that is, first build a plan for one machine, using for it a request closest in time, preventing long waiting time. Then the second car and so on until the orders run out.
>
> 3. Use a simple algorithm for selecting the parameter of the number of cars, for example, using the method of dividing the segment in half to investigate how many cars will be enough.
>
> 4. Use a different algorithm for solving the problem, that is, not just minimizing the cost of the flow, but searching for the maximum flow with the minimum cost.
>
> 5. Use a different model that I suggested at the interview, where each agent (ant) finds some way from the depot through requests. After each order is completed, evaluate the priority of each graph edge by increasing or decreasing the probability of its use, depending on how good the solution was found, which is determined by the total number of ants used.

2. In reality, durations between pickup/dropoff locations are not fixed, but vary throughout the day and are noisy. Can you think of a way to incorporate this into the algorithm?

If the data on the duration of trips is noisy, it is possible to research the distribution of a random variable. In this case, you can define the expected value, standard deviation, and dependency on the time of the day. These values can then be used to build the solution.

Also, if car ride time or car capacity on a flow network cannot be known exactly, it is possible to use methods of fuzzy logic. In this case, we can use fuzzy numbers to build a fuzzy graph, for example, the parameters can be represented as fuzzy triangular numbers. Then we come to the solution to the problem of finding the minimum cost flow in the flow network in fuzzy conditions.

Were to read more:

Sitara, M.; Akram, M.; Yousaf Bhatti, M. Fuzzy Graph Structures with Application. Mathematics 2019, 7, 63.

Sunitha, M S & Mathew, Sunil. (2013). Fuzzy Graph Theory, A survey. Ann Pure Appl Math. 4.