Shortest Path Problem in Layered Graph with Arc Licensing Cost

Huy Duong, Thai Pham, Thuan Phan

April 2019

1 Literature

The Shortest Path in a Layered Graph Problem without Arc Licensing Cost is described in [2].

The Shortest Path in a Layered Graph Problem with Arc Licensing Cost is described in [1].

2 Introduction

3 Problem Statement

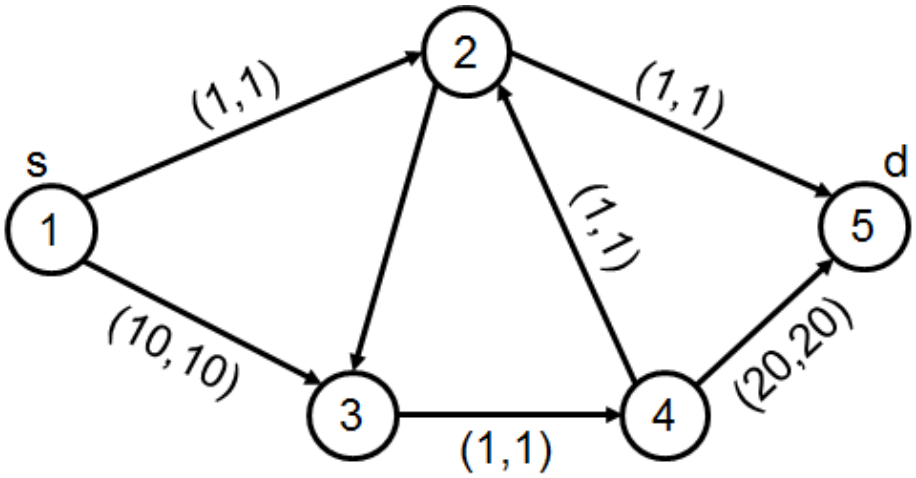
3.1 Ordered Sale Graph Problem

The initial problem can be described as a variant of the Travelling Salesman Problem. A salesman is travelling between a set of cities V connected by a set of directed arc A. He planed to start from the city s and end his travelling at the city d. He has also a set of items I from his company, in an advertisement campaign. Due to the policies, he has to sale the items exactly in the order given by his company. So we can assume the I is an ordered set. For an item i, if it is sold at a city v, the company has to pay him P_v^i. These profits can be either positive or negative values.

Each arc $a \in A$ is associated with two costs (W_a, L_a) which are transportation and license cost respectively. Each time he travels between two cities of arc $a \in A$, he has to pay the transportation cost W_a. The license is required to buy once, while the transportation cost must be paid every time. A graph, $G(V, A, I, P, W, L, s, d)$, is named an ordered sale graph.

Given an ordered sale graph, as the owner of the company, you must find a least cost path (traveling cost and sale cost) and give it to your salesman.

In the Figure 1, the least cost path is $(1, 2, 3, 4, 2, 5)$ with the value of 10.



P[i][v]

	1	2	3	4	5
1	10	10	0	0	10
2	10	10	0	0	10

Figure 1: Example of the problem

3.2 Layered Sale Graph Representation

In order to simplify the problem, it can be represented as a layered graph by which the payment to the salesman appears as transport arc cost. Since all the costs are now only related to arc cost (transport or license ones), they help to utilize classical shortest path problem algorithms. The formal representation of a layered graph from a sale graph is described below.

A layered sale graph $G^{\mathrm{L}}(V^{\mathrm{L}}, A^{\mathrm{L}})$ is comprised of $|I|+1$ layers where G_i^{L} is its ith layer. Each node in G^{L} is indexed by a pair (i, v) where i is the index of its layer and v is the original index of node in the graph G. Within a layer, nodes are connected by the same set of edges as the original graph G with the same transport cost. From a layer i to layer $i+1$, for each node v, there is an arc from (i, v) to $(i+1, v)$ with the transport cost equal to the payment that the company has to pay the salesman if he sold the item i at the city v. The layered graph from the example 1 is illustrated in the Figure 2.

4 Mathematical Model

Although the mathematical model is similar to the one presented in [1], it is represented according to the notations used in this paper.

Variables:

- $\forall a \in A : x_a = 1$ if the least cost path uses arc a, 0 otherwise.

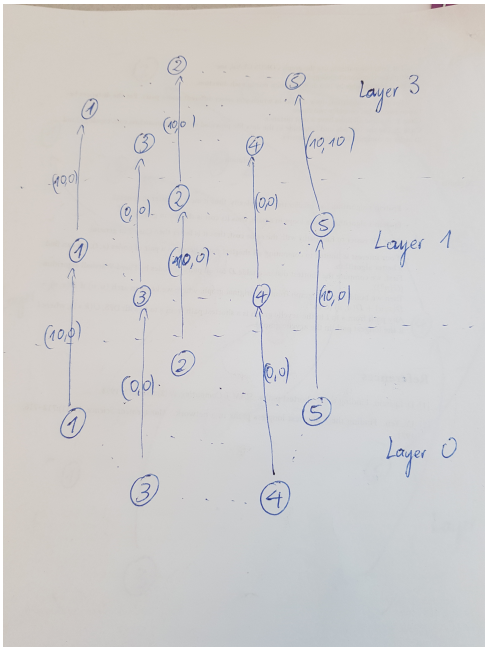


Figure 2: Layered Graph illustration

- $\forall a \in A : \delta_a$ = number of times the least cost path goes through the original arc a.
- $\forall a \in A, i \in \{0, .., |I|\} : \varphi_a^i = 1$ if, the least cost path uses arc a to go from the city where the ith item is sold to the city where the item $i+1$ is sold, 0 otherwise. Note that, when $i = 0$, φ_a^i represents the path from the start city to the item, when $i = |I|$, it is the path from the last item to the destination.
- $\forall u \in V, i \in \{0, .., |I| - 1\} : \beta_v^i = 1$ if the item I_i is sold at the city v_u, 0 otherwise.

Objective: [Thai: You may mean integer linear programming? I don't really understand your model] [Huy: I updated the formulations. Is it readable?]

$$\min \quad \sum_{a \in A} L_a x_a + \sum_{a \in A} W_a \delta_a + \sum_{v \in V} \sum_{i=0}^{|I|-1} \beta_v^i P_v^i. \tag{1}$$

Constraints:
Aggregation of arc usage:

$$\delta_\ell^\pi = \sum_{i=0}^{|I|} \varphi_a \qquad a \in A. \tag{2}$$

A arc of a layer is allowed to use only when the license is bought:

$$\varphi_a^i \leq x_a \qquad a \in A, i \in \{0, .., |I|\}. \tag{3}$$

Flow Conservation constraints

$$\sum_{\ell \in \omega^+(v)} \varphi_\ell^0 - \sum_{\ell \in \omega^- v} \varphi_\ell^0 + \beta_v^0 = \begin{cases} 1 & \text{if } v = s \\ 0 & \text{else} \end{cases} \qquad v \in V^{\text{P}} \tag{4}$$

$$\sum_{\ell \in \omega^+(v)} \varphi_\ell^{|I|} - \sum_{l \in w^- v} \varphi_\ell^{|I|} - \beta_v^{|I|-1} = \begin{cases} -1 & \text{if } v = d \\ 0 & \text{else} \end{cases} \qquad v \in V^{\text{P}} \tag{5}$$

$$\sum_{\ell \in \omega^+(v)} \varphi_\ell^i - \sum_{\ell \in \omega^-(v)} \varphi_\ell^i + \beta_v^i - \beta_v^{i-1} = 0 \qquad v \in V^{\text{P}}, 0 < i < |I|. \tag{6}$$

Constraints (4) ensure that the salesman departs at the starting city, then goes through a path to the location of the first item (unless first function is located at the source node). Similarly, constraints (5) make sure that the salesman reach the destination city after he sold the last item. From the city of item $i-1$ to the city of item i, constraints (6) define a path to connect them.

5 NP-Hard Property

[Huy: Now we have to prove that this problem is NP-Complete]

6 Heuristic Algorithm

6.1 First Heuristic: search algorithms

Use depth first search or breath first search to compute all possible paths. We can use few criterion to prune search trees.

Apply A* algorithms.

6.2 Second Heuristic: transport cost only

The first heuristic:

- Consider G^{TRAN} which is G with only transport costs, i.e., there is no license cost.
- Then, the problem on G^{TRAN} become a simple weighted shortest path problem in a DAG. Assume that the found shortest path is $P = \{a_1, a_2, \ldots, a_|P|\}$.
- Now, take into account the license cost, compute the license cost of path P in G and add it to the transport cost to make up the final cost.

6.3 Third Heuristic: license cost only

The first heuristic:

- Consider G^{LIC} which is G with only transport costs, i.e., there is no transport cost.
- Assume that the license costs are now the transport cost, i.e., every time the salemen travels on an arc, he has to buy the license again.
- Then, the problem on G^{LIC} become a simple weighted shortest path problem in a DAG. Assume that the found shortest path is $P = \{a_1, a_2, \ldots, a_{|}P|\}$.
- Now, take into account the original license cost condition and transport cost, compute the license cost for path P and add it to the transport cost to make up the final cost.

6.4 Forth Heuristic: Machine Learning techniques

[HUY: I am looking for Machine Leaning techniques for shortest path problems.]
Deep learning models for route planning in road networks [4].
Shortest Path Distance Approximation Using Deep Learning Techniques [3].

7 Conclusion

References

[1] Huy Duong and Brigitte Jaumard. A nested decomposition model for reliable NFV 5g network slicing. In *Proceedings of the 9th International Network Optimization Conference, INOC 2019, Avignon, France, June 12-14, 2019.*, pages 107–112, 2019.

[2] N. Huin, B. Jaumard, and F. Giroire. Optimization of network service chain provisioning. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2017.

[3] Fatemeh Salehi Rizi, Joerg Schloetterer, and Michael Granitzer. Shortest path distance approximation using deep learning techniques. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1007–1014. IEEE, 2018.

[4] Tianyu Zhou. Deep learning models for route planning in road networks, 2018.