Flows in dynamic networks with aggregate arc capacities

Vardges Melkonian

Department of Mathematics, Ohio Universtiy, Athens, Ohio 45701, USA
vardges@math.ohiou.edu

Abstract

Dynamic networks are characterized by transit times on edges. Dynamic flow problems consider transshipment problems in dynamic networks. We introduce a new version of dynamic flow problems, called bridge problem. The bridge problem has practical importance and raises interesting theoretical issues. We show that the bridge problem is NP-complete. Traditional static flow techniques for solving dynamic flow problems do not extend to the new problem. We give a linear programming formulation for the bridge problem which is based on the time-expanded network of the original dynamic network.

Keywords: Networks flows; Computational complexity; NP-completeness; Linear programming

1. Introduction

In this paper we consider dynamic networks, a network flow model that includes transit times on edges. A *dynamic network* is defined by a directed graph $G = (V, E)$ with sources, sinks, non-negative capacities u_e, and integral transit times τ_e for every edge $e \in E$. In the dynamic flow problems considered before the capacities had the following interpretation. The flow initiated in arc e at each period of time cannot exceed u_e. Consider the example of figure 1.

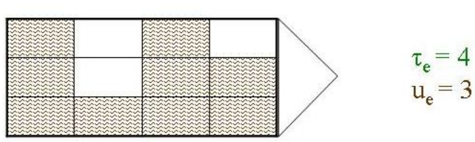

Figure 1: Highway arc

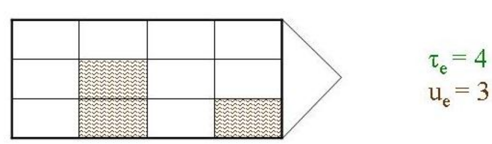

Figure 2: Bridge arc

Here the transit time is 4, so we have divided the arc into 4 time segments. Since the capacity is 3, each time segment can contain up to 3 units of flow. In the figure the capacity is fully used in time segments 1 and 3. This version of the network has obvious association with a pipeline or a highway flow. Thus, we will refer to it as *highway dynamic network* or just as *highway network*.

The highway dynamic transshipment problem (**HDT**) is defined by a highway network $\mathcal{N}$, a time horizon T, and demands (supplies) d_x for all sinks (sources) such that the total supply equals the total demand, *i.e.*, $\sum_{s \in sources} d_s = \sum_{t \in sinks} d_t$. The problem is to find a feasible dynamic flow f with time horizon T and $|f|_x = d_x$ for every terminal x, if such a flow exists.

The highway version of dynamic network flow problems were introduced by Ford and Fulkerson [3], who considered dynamic networks with a single source and a single sink. They showed that a natural variant of the maximum flow problem in single-source single-sink dynamic networks can be solved by one minimum-cost flow computation. Hoppe and Tardos [4] gave the first polynomial-time algorithms that solve dynamic network flow problems with multiple sources and/or sinks.

In this paper we consider a different version of dynamic networks. As before there are transit time τ_e and capacity u_e associated with each arc e. However, the meaning of capacity is different now: at any moment of time arc e can contain at most u_e units of flow. Let us see the difference on the example of the arc considered above (see figure 2). We can divide the arc into 4 segments again but the *total* flow in these 4 segments can be at most 3. In the figure, the capacity is fully used by having 2 unit in segment 2 and 1 unit in segment 4. A network with this type of arcs will be called *bridge network* because each arc e can be considered as a bridge with total capacity u_e.

The goal of the bridge dynamic transshipment problem (**BDT**) is the same as before: find a feasible dynamic flow in a bridge network if such a flow exists. We will refer to HDT and BDT as *highway problem* and *bridge problem*, respectively.

Dynamic network flow problems arise in many applications (*e.g.*, airline, truck, and railway scheduling, evacuation problems), see the surveys of Aronson [1] and Powell et al [7]. Minieka [6] considered the network flow problem for dynamic networks with varying capacities. Köhler and Skutella [5] considered the version of the problem where transit times are load-dependent.

The bridge problem has not been considered before. It was raised by the Dahlgren Lab of U.S. Navy in the joint project with Cornell University [2]. The main goal of the project was to give a practical implementation of an algorithm for solving the highway problem (Hoppe and Tardos [4]) and to test it on data set provided by U.S. Navy. Éva Tardos and the author of this paper were the main investigators of the project from Cornell University. However, the Navy representatives showed even more interest in solving the bridge problem. While this speaks about the practical importance of the problem, it also raises interesting and challenging theoretical issues which we address in this paper.

The paper is organized in the following way. In section 2 we discuss why the techniques for solving other dynamic flow problems do not extend to the bridge problem. A linear programming formulation for the bridge problem is given in section 3. In section 4 we prove that the bridge problem is NP-complete in the weak sense.

2. Difficulties of solving the bridge problem

Any dynamic network flow algorithm must somehow represent dynamic flow on an edge as that flow changes with time. The standard technique is to consider discrete steps of time and make a copy of the original network for every time step from time zero until the *time horizon* T, after which there is no flow left in the network. This process results in a *time-expanded network*. The time-expanded network contains a copy of the node set of the underlying original network for each discrete time step of the time horizon. Moreover, for every arc $i \rightarrow j$ of the original network with transit time τ_{ij} and for any time period $t \in 0, \ldots, T - \tau_{ij}$, there is a copy of the arc from node i of layer t to node j of layer $t + \tau_{ij}$ in the time-expanded network. For example, consider the instance of the highway problem given in figure 3.

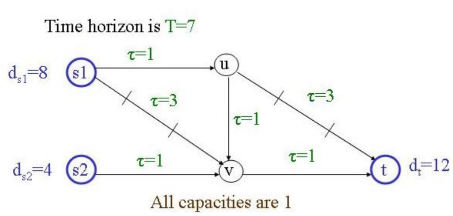

Figure 3: Dynamic network

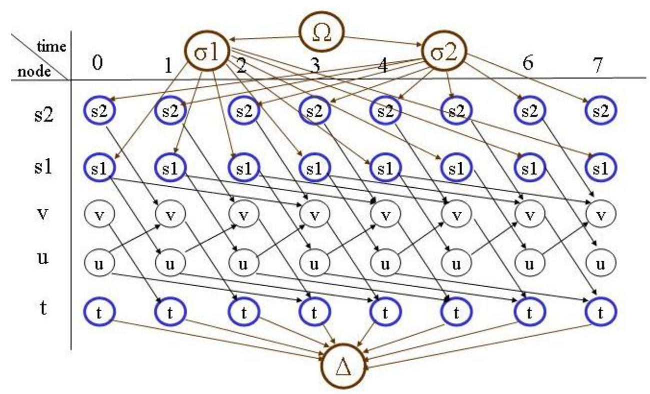

Figure 4: Time-expanded network

We create a copy of each node and arc for each time period as described above. The resulting time-expanded network is shown in figure 4. To solve the problem by maximum flow techniques, we also need supersources and supersinks. A supersource σ_i is created for every source s_i. There is an arc from σ_i to every copy of s_i with capacity equal to the supply d_i of that source. In addition, a supersupersource Σ is created. There is an arc from Σ to every supersource σ_i with capacity d_i. Similarly, supersinks δ_i's and a supersupersink Δ are created along with corresponding arcs. Note that if there is just one source (sink) in the dynamic network then there is no need for having supersources σ_i's (supersinks δ_i's); a supersupersource (supersupersink) is enough. That is the case with the sinks in our example. There are no transit times in the time-expanded network; the time-based layers of the network already take care of transit times. Finding the maximum flow from the supersupersource Σ to supersupersink Δ will also solve the original highway problem. The original highway problem is feasible if and only if the maximum flow value is equal to the total demand.

Thus, the highway problem can be solved by applying static maximum flow techniques to the time-expanded network. All algorithms based on time-expanded networks have running times depending polynomially on T; such algorithms are *pseudopolynomial*. Dynamic network flows can be used to model continuous-time problems in the real world. A more accurate model relies on finer granularity, implying more time steps before the dynamic flow is finished. The performance of a pseudopolynomial algorithm degrades at least linearly with the improvement of model granularity; this restricts the accuracy that can be achieved by the model.

Hoppe and Tardos [4] gave polynomial-time algorithms for solving the highway problem; they depend polynomially on $\log T$, not on T. This breakthrough is achieved by eliminating the time-expanded network. Rather than computing the flow on each edge at every individual time step, their algorithms produce solutions characterized by long time intervals for each edge during which its flow remains constant.

However, no efficient algorithm is known for the bridge problem. Unfortunately, the bridge problem does not have even the inefficient method of solving the problem by time-expanded networks. We will show that on the example of the network of figure 3. Suppose it is a bridge problem now. The time-expanded network for a bridge problem is created exactly the same way as we did it for a highway problem. Thus, figure 4 gives the time-expanded network of the bridge problem too.

Note that we cannot exclude any of the parallel arcs because we do not know beforehand which of them will be included in the solution by a maximum flow algorithm. But in that case the algorithm might choose to include two consecutive parallel arcs in the solution using the full capacity for each of them. For example, we might have 1 unit of flow on the copies of arc $s1 \to v$ which start at times 0 and 1. But this will not be a feasible solution to the bridge problem because at time period 2 we will have $\geq 1 + 1 = 2$ units of flow on arc $s1 \to v$; this is more than its allowed capacity of 1.

3. Solving the bridge problem by linear programming

Though the time-expanded network does not allow to solve the bridge problem by static maximum flow tecniques, it does give a linear program which can solve the problem.

We define a decision variable for each arc of the time-expanded network. Let

- x_{ijk} be the amount of flow initiated at the tail of arc $i \to j$ at time k;

• y_{ik} be the amount of flow from supersource σ_i to copy k of source s_i (from copy k of sink t_i to supersink δ_i);

• z_i be the amount of flow from supersupersource Σ to supersource σ_i (from supersink δ_i to super-supersink Δ).

By this definition we do not exclude any of the parallel copies of any arc. On the other hand, we can enforce the capacity constraints by requiring the flow on any τ_{ij} consequtive parallel arcs to be no more than u_{ij}:

$$\sum_{k=\alpha}^{\alpha+\tau_{ij}-1} x_{ijk} \leq u_{ij}, \text{ for any arc } i \to j \text{ and any time period } \alpha \tag{1}$$

The arc capacity constraints are based on our discussion from Section 2 about arc capacities of the time-expanded network:

$$x_{ijk} \leq u_{ij}, \qquad \text{for any arc } i \to j \text{ and for any time period } k \tag{2}$$

$$y_{ik} \leq d_i, \quad \text{for any source } s_i \text{ (or any sink } t_i\text{) and for any time period } k \tag{3}$$

$$z_i \leq d_i, \qquad \text{for any supersource } \sigma_i \text{ (or any supersink } \delta_i\text{)} \tag{4}$$

We need conservation of flow constraints for the nodes of the time-expanded network, that is, the outflow is equal to the inflow for each node except the ones corresponding to sources and sinks:

$$\sum_{i:i\to v} x_{i,v,k-\tau_{iv}} = \sum_{j:v\to j} x_{v,j,k}\ , \text{ for any non-terminal node } v \text{ and any time period } k \tag{5}$$

The conservation of flow constraints for the sources and sinks are the following:

$$y_{sk} = \sum_{j:s\to j} x_{s,j,k}\ , \quad \text{for any source } s \text{ and any time period } k \tag{6}$$

$$y_{tk} = \sum_{i:i\to t} x_{i,t,k-\tau_{it}}\ , \quad \text{for any sink } t \text{ and any time period } k \tag{7}$$

For supersources and supersinks we have the following constraints:

$$z_i = \sum_{k=0}^{T} y_{ik}, \text{ for any supersource } \sigma_i \text{ (or for any supersink } \delta_i\text{)} \tag{8}$$

Finally, we need constraints which provide that the total demand originates from the supersuper-source and ends in the supersupersink.

$$\sum_{s\in sources} z_s = \sum_{s\in sources} d_s \tag{9}$$

$$\sum_{t\in sinks} z_t = \sum_{t\in sinks} d_t \tag{10}$$

Of course, just one of these two constraints would be enough since we have conservation of flow constraints for all other nodes of the time-expanded network.

The objective function can be chosen arbitrarily because our goal is to verify whether there is a feasible solution or not. However, a secondary goal could be making the feasible flow (if it exists at all) to arrive at the sinks as early as possible. The following objective function aims to achieve that goal:

$$\min \quad \sum_{sinks\ t} \quad \sum_{i \to t,\ \alpha \in 0,\dots,T-\tau_{it}} (\alpha + \tau_{it}) \cdot x_{i,t,\alpha} \tag{11}$$

Note that $\alpha + \tau_{it}$ is the arrival time of flow $x_{i,t,\alpha}$ at sink t. By having coefficient $\alpha + \tau_{it}$ for the flow $x_{i,t,\alpha}$, the minimization of the objective function encourages larger size for those batches which have earlier arrival time.

The linear program for solving the bridge problem is defined by the objective function and the constraints given above. The bridge problem is feasible if and only if the linear program has a feasible solution.

4. Complexity of the bridge problem

The linear program of the previous section gives a pseudo-polynomial algorithm for solving the bridge problem. So the problem is not NP-complete in the strong sense. But in this section we show that the bridge problem is NP-complete in the weak sense even for the special case of a single source and a single sink. It is done by a reduction from the well-known NP-complete PARTITION problem.

PARTITION

Instance: A set of n items with sizes $a_1, ..., a_n \in Z^+$ such that $\sum_{i=1}^{n} a_i = 2L$ for some $L \in Z^+$.

Question: Is there a subset $I \subset \{1, ..., n\}$ with $\sum_{i \in I} a_i = L$?

Given an instance of PARTITION, we create an instance of the bridge problem as follows. We have just one source s and one sink t, a sequence of nodes $v_1, v_2, ..., v_{n+1}$ where $v_1 = s$ and $v_{n+1} = t$. For each $i = 1, ..., n$ there are two parallel arcs e_i and e'_i with tail v_i and head v_{i+1} (see figure 5). Let $E_n = \{e_1, e_2, ..., e_n\}$ and $E_{n+1} = \{e'_1, e'_2, ..., e'_n\}$. The capacities of all arcs are 1. Let $a_{max} = max_{i \in \{1,...,n\}} a_i$. The transit times of the arcs are $\tau_{e_i} = n \cdot a_{max} \cdot a_i$ for $e_i \in E_n$ and $\tau_{e'_i} = (n \cdot a_{max} + 1) \cdot a_i$ for $e'_i \in E_{n+1}$. The time horizon is $T = (2 \cdot a_{max} \cdot n + 1) \cdot L$. The problem is to send 2 units of flow from s to t in the given time horizon.

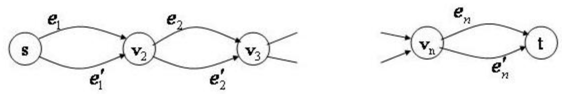

Figure 5: Reduction of PARTITION to the bridge problem

Theorem 1 *There exists a dynamic flow which sends* 2 *units of flow from* s *to* t *in time* T *if and only if the underlying instance of PARTITION is a 'yes'-instance.*

Proof: *If*: Let I be a subset of $\{1, ..., n\}$ such that $\sum_{i \in I} a_i = L$. Then the flow of 2 units can be sent in two packets, each containing one unit of flow. The packets use two arc-disjoint paths P_1 and P_2 of length n determined as follows. P_1 contains those arcs $e_i \in E_n$ such that $i \in I$ and those arcs $e'_i \in E_{n+1}$ such that $i \in \{1, ..., n\} - I$. P_2 contains the remaining n arcs of the network. Then the total transit time of path P_1 is equal to the sum of the transit times of its arcs:

$\tau(P_1) = \sum_{e_i \in E_n : i \in I} \tau_{e_i} + \sum_{e'_i \in E_{n+1} : i \in \{1,...,n\} - I} \tau_{e'_i} = n \cdot a_{max} \cdot L + (n \cdot a_{max} + 1) \cdot L = (2 \cdot a_{max} \cdot n + 1) \cdot L$
Similarly it can be shown that the total transit time of P_2 is $(2 \cdot a_{max} \cdot n + 1) \cdot L$.

Only if: Suppose we have a a dynamic flow which sends 2 units of flow from s to t in time T. Since the capacity of any arc is 1 then no arc can have more than one unit of flow at any given time. Moreover, we claim that no arc can have more than one unit of flow within the time horizon T. Assume the opposite: there is an arc that has more than one unit of flow. Let that arc be $e_k \in E_n$ (similar arguments can ba applied to the arcs from E_{n+1}). The earliest time the first unit of flow on e_k can reach the head of e_k is $\sum_{i=1}^{k} \tau_{e_i} = n \cdot a_{max} \cdot \sum_{i=1}^{k} a_i$. But this is the earliest time any extra flow can enter e_i at its tail. From that point on, the extra flow needs at least $\sum_{i=k}^{n} \tau_{e_i} = n \cdot a_{max} \cdot \sum_{i=k}^{n} a_i$ time to reach the sink t. Thus, the total time required is

$n \cdot a_{max} \cdot \sum_{i=1}^{k} a_i + n \cdot a_{max} \cdot \sum_{i=k}^{n} a_i = n \cdot a_{max} \cdot \sum_{i=1}^{n} a_i + n \cdot a_{max} \cdot a_k = 2 \cdot a_{max} \cdot n \cdot L + n \cdot a_{max} \cdot a_k \geq 2 \cdot a_{max} \cdot n \cdot L + 2 \cdot L \cdot a_k > T$

The contradiction proves that no arc can have more than one unit of flow within the time horizon T. Based on this argument, the 2 units of flow are sent through two arc-disjoint paths P_1 and P_2, each with one unit of flow. Then we claim that the underlying instance of PARTITION is a 'yes'-instance by choosing $I = \{i \in \{1, ..., n\} : e_i \in E_n \text{is on path} P_1\}$. We need to show that $\sum_{i \in I} a_i = L$. Assume the opposite: $\sum_{i \in I} a_i = L + \delta$ for some non-zero δ. Then the total transit time of the flow on path P_1 is

$\tau(P_1) = \sum_{e_i \in E_n : i \in I} \tau_{e_i} + \sum_{e'_i \in E_{n+1} : i \in \{1,...,n\} - I} \tau_{e'_i} = n \cdot a_{max} \cdot (L + \delta) + (n \cdot a_{max} + 1) \cdot (L - \delta) = (2 \cdot a_{max} \cdot n + 1) \cdot L - \delta = T - \delta$

$\tau(P_2) = \sum_{e_i \in E_n : i \in \{1,\ldots,n\} - I} \tau_{e_i} + \sum_{e'_i \in E_{n+1} : i \in I} \tau_{e'_i} = n \cdot a_{max} \cdot (L - \delta) + (n \cdot a_{max} + 1) \cdot (L + \delta) = (2 \cdot a_{max} \cdot n + 1) \cdot L + \delta = T + \delta$

That is, on one of the two paths the flow will not reach the sink t within the time horizon T. This contradiction proves that $\sum_{i \in I} a_i = L$. □

5. Concluding remarks

5.1 Reduction heuristic to highway problem

Recall that the main difference between the two problems is the different interpretation of the arc capacities. So when reducing a bridge problem to a highway problem, some modification should be done with capacities. We suggest the following modification. If e is an arc in the bridge network with capacity u_e and transit time τ_e then the corresponding arc in the highway network has transit time $\hat{\tau_e} = \tau_e$ and capacity $\hat{u_e} = u_e/\tau_e$. No other changes are made in the problem. A flow in the reduced highway problem will satisfy the capacity constraints of the original bridge problem because the flow on any arc e at any moment of time is at most $\hat{\tau_e} \cdot \hat{u_e} = \tau_e \cdot \frac{u_e}{\tau_e} = u_e$. If the reduced highway problem (which can be solved by the algorithm of Hoppe and Tardos [4]) is feasible then the original bridge problem is also feasible. However, the infeasibility of the highway problem does not yet imply that the bridge problem is infeasible.

5.2 Networks with mixed capacities

In practical settings it is most likely to have mixed networks where some arcs are highway arcs and the others are bridge arcs. Hybrids of the algorithms for highway and bridge problems could bring solution methods for the mixed problems. For example, in the reduction algorithm of subsection 5.1 we could change only the capacities of the bridge arcs while not changing anything about the highway arcs.

Yet another interesting variation of the problem is when we have two kind of capacity constraints on the same arc. We might have transit time τ_e and capacity u_e in the highway sense, for the amount of flow entering arc e at any given time. But at the same time there might be a capacity u'_e in the bridge sense: the total flow on arc e at any given time is no more than $u'_e \leq \tau_e \cdot u_e$. Note that the reduction algorithm of subsection 5.1 would work also for this variation of dynamic networks.

References

[1] Aronson, J.E., A survey of dynamic network flows, Annals of Operations Research 20 (1989) 1–66.

[2] Dahlgren Lab of U.S. Navy, Personal communications 2001.

[3] Ford, L.R. and Fulkerson, D.R., Flows in networks, Princeton University Press, 1962.

[4] Hoppe, B. and Tardos, É., The quickest transsipment problem, Mathematics of Operations Research 25 (2000) 36–62.

[5] Köhler, E. and Skutella, M., Flows over time with load-dependent transit times, Proceedings of SODA'02 (2002) 174–183.

[6] Minieka, E., Dynamic network flows with arc changes, Networks 4 (1974) 255–265.

[7] Powell, W.B., Jaillet, P. and Odoni, A., Stochastic and dynamic networks and routing, In: Network Routing, Vol. 8 (1995) of Handbooks in Operations Research and Management Science, Chapter 3 141–295.