Karlsruhe Institute of Technology

Theoretical Analysis of Different Dynamic Flow Variations

Bachelor Thesis of

Jannik Westenfelder

At the Department of Informatics
Institute of Theoretical Informatics

Reviewers: Dr. Thomas Bläsius
PD Dr. Torsten Ueckerdt

Advisors: Adrian Feilhauer

Time Period: 5th May 2022 – 19th September 2022

Statement of Authorship

Ich versichere wahrheitsgemäß, die Arbeit selbstständig verfasst, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, September 19, 2022

Abstract

The dynamic max flow problem is based on the static max flow problem. It extends the base problem by adding a temporal component through a travel time for arcs. We provide an overview of the hardness of different variations of this max dynamic flow problem and the needed proofs. The dynamic max flow problem is weakly NP-hard if we allow the travel times or the capacities to change at integral points in time. These results show that, in contrast to dynamic max flow problems with arcs that have a constant travel time or capacity, allowing these changes causes the problem to become NP-hard.

Deutsche Zusammenfassung

Das Problem der Dynamischen Flussmaximierung beruht auf dem Problem der Statischen Flussmaximierung und kommt zustande, indem man das statische Problem um die zeitliche Dimension ergänzt. Dies geschieht dadurch, dass die Kanten zusätzlich zu der Kapazität auch eine Reisezeit haben. Diese beschreibt, wie lange ein Fluss braucht, um diese Kante zu passieren. In dieser Thesis bieten wir einen Überblick über Variationen von dem Problem der Dynamischen Flussmaximierung und eine Einordnung dieser in die Komplexitätsklassen P, schwach NP-schwer und stark NP-schwer. Dabei hat sich herausgestellt, dass die Dynamische Flussmaximierung schwach NP-schwer wird, wenn wir eine Änderung der Reisezeiten oder der Kapazitäten zu ganzzahligen Zeitpunkten zulassen.

Contents

1. Introduction

This thesis deals with dynamic flow variations, more precisely the max dynamic flow problem introduced by Ford and Fulkerson in [FF62]. Dynamic flows are also called flows over time. They extend the classical static flow model by adding a temporal component with a travel time for each arc, which means that flow needs a certain amount of time to travel along an arc. The dynamic flow problem also has a time horizon that is a point up to which we want to maximize the flow entering the sink. With this extension, the dynamic flow could be used in more applications like routing or traffic. We can generalize the problem to model more things for example a construction site in traffic by allowing the travel time or the capacity to change over time. These are two of the main variations that we analyse in this thesis.

The paper by Martin Skutella [Sku09] gives an introduction about dynamic flows and introduces time expanded networks that can be used to convert a dynamic flow problem into a static flow problem. Fleischer and Tardos [FT98] show a strong connection between the discrete and continuous time model.

This thesis is organized as follows. In Chapter 2 we introduce some basic notation and give some definitions that are used later on. In Chapter 3 we present a hardness analysis of the max dynamic flow with travel times or capacities that can change at integral times. In that chapter we also introduce overall capacity which is another concept of capacity and prove that it is NP-hard. In Chapter 4 we present some information about continuous networks and discrete networks. In Chapter 5 we conclude the thesis and mention possible future work.

2. Preliminaries

Before defining a maximum flow over time we need to define a network.

Network

A network consists of a directed Graph $G = (V, E)$ a source node $s \in V$, a sink node $t \in V$.

Each arc $e = (w, v) \in E$ has:

- A capacity $u_e \in \mathbb{R}_{\geq 0}$ which describes a limit of how much flow can be sent into the arc at a time.
- A travel time or transit time $\tau_e \in \mathbb{R}$ which describes the time needed for flow to travel from node w to node v while travelling across arc e.

In other settings we may have:

- Dynamic capacity: capacity of an arc can change on integral times θ therefore we write the function $u_e(\theta)$.
- Dynamic travel time: travel time of an arc can change on integral times θ therefore we write the function $\tau_e(\theta)$.
- Cost: each arc also has a cost $c_e \in \mathbb{R}$ which describes the cost for sending one unit of flow across arc e.

Now that the network is clear we can define a flow over time.

Flow Over Time

A continuous flow over time has a time horizon $T \in \mathbb{R}_{>0}$. The continuous flow is described by function $f : (E, [0, T)) \to \mathbb{R}_{\geq 0}$ describing the amount of flow entering the arc at the given time. We define f as zero for undefined values such as a negative time to simplify some definitions.
The continuous flow is feasible when:

- For all $e \in E$ and for all time points $\theta \in [0, T)$: f(e,θ) $\leq u_e(\theta)$.

- if we have weak flow conservation, which means that flow can be stored in a node, the following must be valid for all $a \in (V/\{s,t\})$ and for all points in time $\theta > 0$:

$$\int_0^{\theta} \sum_{e=(b,a)\in E} f(e, \theta - \tau_e) \geq \int_0^{\theta} \sum_{(a,v)\in E} f((a,v), \theta)$$

  for $\theta \geq$ T it has to be equal.

- if we have strong flow conservation, which means that flow can not be stored in a node, the following must be valid for all $a \in (V/\{s,t\})$ and for all points in time $\theta > 0$:

$$\int_0^{\theta} \sum_{e=(b,a)\in E} f(e, \theta - \tau_e) = \int_0^{\theta} \sum_{(a,v)\in E} f((a,v), \theta)$$

A discrete flow over time is discretized over time.

The *value* of a flow is the total flow send into node t which equals $\int_0^T \sum_{(a,t)\in E} f((a,t), \theta - \tau_{(a,t)}$.

A *maximum flow over time* is a feasible *st*-flow maximizing the value of a flow within time horizon.

For proving that a variation is NP-hard we first need to define NP-hard problems to reduce from later.

Definition 2.1 (2-Partition). *Given a multiset $M = \{m_0, m_1, ..., m_k\}$ of positive integers. Find a subset $S_1 \subset M$ with $\sum_{m\in S_0} m = \sum_{m\in M} m/2$.*

This problem is well known to be weakly NP-hard.

Definition 2.2 (3-Partition). *Given a multiset $M = \{m_0, m_1, ..., m_k\}$ with $k = 3n$ where the sum of all integers is $n * L$, Find n triplets $S_0, S_1, ..., S_{n-1}$ with the sum of L. These triplets can only contain elements of M and each element of M has to be in exactly one of these triplets.*
This problem is well known to be strongly NP-hard.

With the following lemma we can use multi-graphs later on instead of normal networks.

Lemma 2.3. *Multi-graphs can be transformed into networks in polynomial time.*

Proof. In a multi-graph $M = (V,E)$ transform a multi-arc $e = (x,y)$ with k arcs with $u_{e_1}, ...u_{e_k}$ and $\tau_{e_1}, ...\tau_{e_k}$ into $e_{1a} = (x, z_1)$with $u_{e_{1a}} = u_{e_{1b}} = u_{e_1}, \tau_{e_{1a}} = \tau_{e_1}$ and $e_{1b} = (z_1, y)$ with $c_{e_{1a}} = c_{e_1}, \tau_{e_{1a}} = 0$ with $z_1 \notin V$. Do this for all arcs in the multi-arc. Remove the multi-arc, add the new arcs and the new nodes. An example of this transformation can be seen in Figure 2.1. □

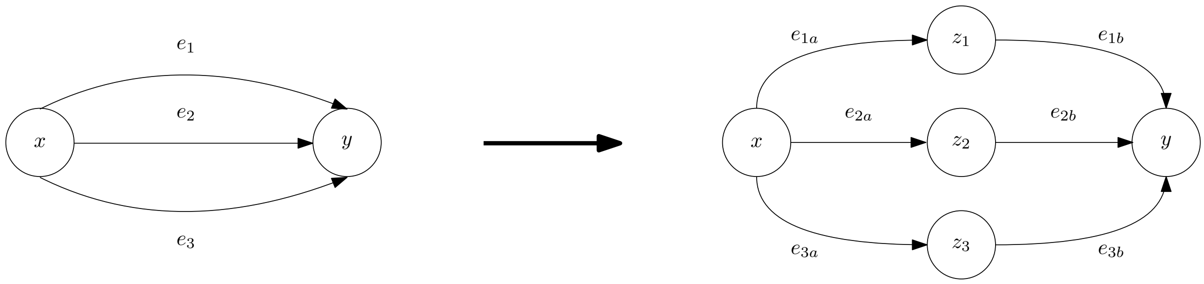

Figure 2.1: Example for converting a multi-graph to a network.

3. Different dynamic flow variations

Maximum flows over time have been analyzed mostly with static capacities or travel times. In this chapter we look at dynamic arcs and how these variations change the complexity of the problem.

3.1 Dynamic Arcs

Most max flow computations use static arcs, where each arc has a travel time and a capacity that stays constant over time. But in this section we consider dynamic arcs which means that the travel time or the capacity can change at any integral point in time.

First, we consider the max flow problem with dynamic capacities. To show that it is NP-hard, we reduce the 2-Partition problem defined in Definition 2.1 to the max flow problem with dynamic capacities. Let $M = \{m_1, m_2, ..., m_k\}$ be an instance of the 2-Partition problem. The goal is to find a subset $S \subset M$ such that the sum of S equals half the sum of M. To compute this, we construct a max-flow instance $N = (V, E = E_1 \cup E_2 \cup E_3)$ with T as time horizon and with dynamic arcs, such that it has a max flow of one if and only if M is solvable. First, we build a multi-graph, which can be seen in Figure 3.1, that has one vertex v_i for every $i \in \{0, ..., k\}$. For every $i \in \{1, ..., k\}$ we have two edges between v_{i-1} and v_i. One of these edges has travel time m_i (this set of arcs is called E_1) and the other has travel time zero (this set of arcs is called E_2). Both have capacity one. That way a flow has to decide whether to use the arc in E_1 or in E_2. This decision can be transferred to whether to take m_i into the solution. So if a flow uses the arc (v_4, v_5) in E_1 we include m_5 in the solution S of the 2-Partition instance M. With this construction we can ensure that S only contains elements that exist in M and the travel time of the flow matches the sum of S. To ensure S is a Solution we have to assure that the travel time is half the sum of M. To achieve that we add two guard arcs. The first guard arc (s, v_0) has capacity $u_{(s,v_0)}(\theta) = \begin{cases} 1 & \text{for } 0 \leq \theta < 1 \\ 0 & \text{else} \end{cases}$. The second guard arc is $u_{(v_k,t)}(\theta) = \begin{cases} 1 & \text{for } T-1 \leq t < T \\ 0 & \text{else} \end{cases}$. Both arcs belong to the arc set E_3 and have travel time zero. With the time horizon $T = 1 + \sum_{m \in M} m/2$ the travel time of a flow is exactly half the sum of M since the travel time of each arc is an integer and strong flow conservation.

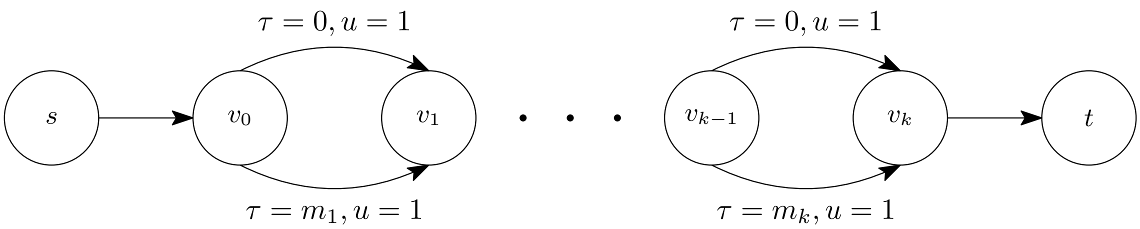

Figure 3.1: A graphic representation of the multi-graph that was built in the reduction from 2-Partition to max flow with dynamic capacities.

Theorem 3.1. *Max flow with dynamic capacity and with strong flow conservation is weakly NP-hard.*

Proof. This theorem is a direct consequence of the proof of the reduction above. So first we prove that if the 2-Partition instance is solvable we can find a flow of value one in the given time horizon. Then we prove that if the multi-graph above has a flow of value one within the time horizon the corresponding 2-Partition instance is solvable.

For the first part we take a solution to the 2-Partition instance and use it to create the desired flow. Let $S \subset M$ be a solution for the 2-Partition instance M. Then we can send one flow unit at the start along arc (s, g). If m_0 is not part of the solution S we take the arc with $\tau = 0$ else take the other arc. This flow will arrive at node m_k at time half the sum of M because the sum of S equals half the sum of M since S is a solution of the 2-Partition instance. Since $u_{(k,t)}(\sum_{m \in M} m/2) = 1$ we can send the flow unit to t and we have a flow with value one in the corresponding network.

Now we come to the second part. For this we take a flow in the multi-graph and show that this flow provides a solution to the corresponding 2-Partition instance. Assume there is a feasible flow of value 1. If the flow splits pick any path of a partial flow that does not split and is feasible. With this flow we build a solution S of the 2-Partition instance. The flow starts at s and reaches g strictly before $\theta = 1$ because at time $\theta = 1$ the edge closes. The flow takes either the arc with $\tau = 0$ or $\tau = m_0$ because the flow does not split. If the flow takes the arc with $\tau = m_0$, add m_0 to S. From m_0 the flow can either take the arc with $\tau = 0$ or $\tau = m_1$. If the flow takes the arc with $\tau = m_1$, add m_1 to S. Continue this until the flow reaches m_k. The flow is feasible and the arc (k, t) opens at $T - 1$ and is the only arc entering t. Therefore, the flow has to reach m_k when it is open. Since the flow does not split, the travel time of the flow equals $\sum_{s \in S} s$. Therefore the sum of S is at least $\sum_{m \in M} m/2$ and strictly less than T. Since all elements in S are integer it has to be $\sum_{m \in M} m/2$ and therefore is a solution to the 2-Partition instance. This means that the given 2-Partition instance is solvable. □

By switching the guard mechanism from reducing the capacity to zero to raising the travel time to sum of M in the multi-graph of Theorem 3.1 we can give the same proof and get the following corollary.

Corollary 3.2. *Max flow with dynamic travel times and without storage is weakly NP-hard.*

These also stay true if we use undirected graphs instead of directed graphs.

Corollary 3.3. *Theorem 3.1 and Corollary 3.2 are also true for undirected graphs.*

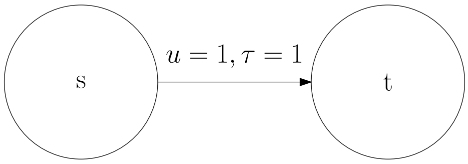

Figure 3.2: Example Network for the overall Capacity.

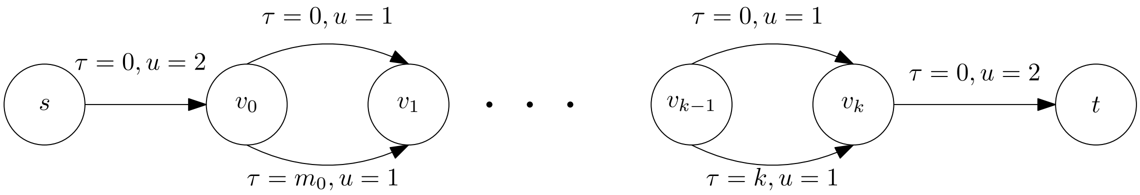

Figure 3.3: A graphic representation of the multi-graph that was build in the reduction from 2-Partition to max flow with overall capacity.

Proof. Take the multi-graph of Theorem 3.1 and add $\sum_{m \in M} m$ to the travel time of all arcs and extend T by $(|M| + 2) \cdot \sum_{m \in M} m$. This way a feasible flow can use at most $|M|$ of the static arcs. Since the flow needs to use at least $|M|$ of the static arcs to reach the sink it can only use arcs in the same direction, as in the directed graph. □

3.2 Overall Capacity

For the next variation we have a look at a different model of capacity which we call *overall capacity*.

Normally the capacity describes the amount of flow that can enter the arc at once. In contrast to that, overall capacity the capacity describes the amount of flow that can enter the arc overall.

For example the Network shown in Figure 3.2 has a max flow of $T - 1$ with normal capacity but with overall capacity it has a max flow of one (if $T \geq 2$). Since the flow has to be a real value the flow always needs time greater zero and therefore, sending the flow in a small time frame as possible does not provide a flow with greater value if all travel times as well as the time horizon are integers.

Now that we have defined the problem, we reduce the 2-Partition problem defined in Section 2.1 to the max flow problem with overall capacities. Let $M = m_1, m_2, ..., m_k$ be an instance of 2-Partition. The goal is to find a subset $S \subset M$ such that the sum of S equals half the sum of M. To compute this we construct a max flow instance $N = (V, E = E_1 \cup E_2 \cup E_3)$ with $T = 1 + \sum_{m \in M} m/2$ as time horizon and overall capacities, such that there exists a max flow of two if and only if M is solvable. First, we build a multi-graph that can be seen in Figure 3.3. We have the same V and the same E_1, E_2 as in the multi-graph of Theorem 3.1. The computation of S is also the same. But since we do not have dynamic arcs and we want a flow of value two, E_3 must differ. Therefore arcs in E_3 have overall capacity $u_{(k,t)} = u_{(s,g)} = 2$ and travel time $\tau_{(k,t)} = \tau_{(s,g)} = 0$. If we have a max flow of two each sub flow has travel time $T - 1$. Assume that one sub flow has a travel time less than $T - 1$. Then another one would have more than $T - 1$ which would lead to a travel time greater than T since all travel times are integer (more precise in the proof). So each sub flow can be used to get the 2-Partition solution the same way it is done in the proof of Theorem 3.1.

Theorem 3.4. *Max flow with overall capacity and without storage is weakly NP-hard.*

Proof. This theorem is a direct consequence of the proof of the reduction above. First, we prove that if the 2-Partition instance is solvable, then we can find a flow of value two in the given time horizon. Then we prove that if the multi-graph has a flow of value two within the time horizon, the corresponding 2-Partition instance is solvable.

For the first part, we take a solution to the 2-Partition instance and use it to create the desired flow. Let S_1, $S_2 \subset M$ be a solution for the 2-Partition instance M. If m_0 is not part of the solution S_1 we send flow of value 1 along the arc with $\tau = m_0$. Else, we take the other arc. This procedure continues until the flow reaches node v_k. This flow arrives at v_k at $\sum_{m \in M} m/2$ because $\sum_{l \in S} l = \sum_{m \in M} m/2$ since S_1 is a solution. Then we send the flow to the sink. We do the same for S_2. Since S_1 and S_2 are independent and every element of M is in one of the sets, we can combine those two flows into one. Now we have a discrete flow of value two and time horizon T in the corresponding network.

For the second part we take a flow of value two in the multi-graph and show that this flow provides a solution to the corresponding 2-Partition instance.

The given flow can be decomposed into at least two flows since every path from s to t contains at least one arc with overall capacity one and therefore it is not possible for the flow of value two to use only one path. Use one of these sub flows to build the set S_1. For each arc of with travel time greater than zero add the corresponding element of M to S_1, like in the proof of Theorem 3.1. Since the flow was feasible the sum of S_1 is less or equal to $T - 1$. The next step is to show that the sum of S_1 is also at least $T - 1$. To prove that, we also build the sets $S_2, ..., S_n$ of the other sub flows. Since the value of all flows combined is two and each flow visits each node and two nodes behind each other are only connected by a multi arc of capacity two, all arcs are used at their full overall capacity. Therefore $\sum_{S_i, 1 \leq i \leq n} f_i \sum_{u \in S_i} u = \sum_{m \in M} m$ with f_i being the flow value of the sub flow corresponding to S_i. If one set S_i would have a sum of less than $T - 1$ another set would have a sum higher than $T - 1$. But since every sub flow is feasible the sum of every S_i must be less or equal to $T - 1$. Therefore, the sum of S_1 is exactly $T - 1$ which means it is a solution to the 2-Partition instance. Now the reduction is verified so the theorem is proven. □

We can adapt this reduction, such that it also works works for undirected graphs.

Corollary 3.5. *Theorem 3.4 is also true for undirected graphs.*

Proof. Take the multi-graph and add $\sum_{m \in M} m$ to the travel time of all arcs and extend T by $(|M| + 2) \cdot \sum_{m \in M} m$. This way a feasible flow can use at most $|M| + 2$ arcs. Since the shortest path from the source to the sink needs $|M| + 2$ it can only use arcs in the same direction, as in the directed graph. □

This problem becomes even strongly NP-hard if we require integral flows. To show this we present a reduction from the 3-Partition problem defined in Definition 2.2 to the max flow problem with overall capacities and integral flows. Let $M = m_1, m_2, ..., m_k$ be an instance of 3-Partition with $k = 3n$ and the sum of all integers is $n \cdot L$. The goal is to find subsets $S_1, ..., S_n \subset M$ such that each set contains exactly three elements, with a sum of L and all sets are independent. To compute this we construct a max flow instance $N = (V, E = E_1 \cup E_2 \cup E_3)$ with $T = T = 1 + L + 3nL$ as time horizon and overall capacities, such that it has a max flow of n if and only if M is solvable. First, we build a multi-graph that can be seen in Figure 3.4. This multi graph has the same nodes and the arcs have the same orientation as in Theorem 3.4 but differs in the travel time and overall capacity. The overall capacity of the arcs in E_2 is $u = n - 1$ since we need a flow

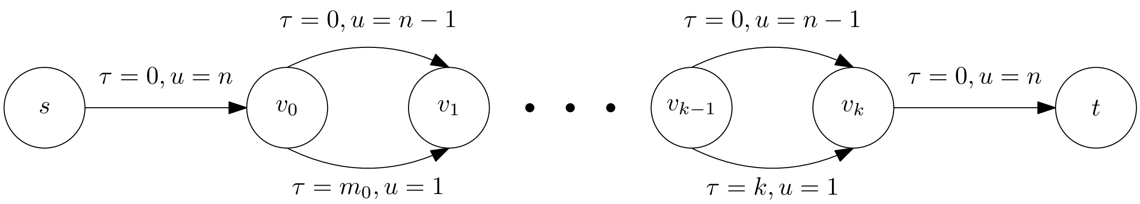

Figure 3.4: A graphic representation of the multi-graph that was built in the reduction from 3-PARTITION to max flow with overall capacity and integral flows.

of value n and one flow unit uses the other arc. The capacity of arcs in E_1 is $u = 1$ since each element can only be in one set. The travel time of arcs in E_2 is zero since they are used as bypass arcs.The travel time of arc (v_{i-1}, v_i) in E_1 is $\tau = m_i + n \cdot L$ to ensure that no sub flow leads to a set with more than three elements. Since we have a flow of value n and integral flows we have n sub flows of value one and each sub flow visits each node and two nodes behind each other are only connected by a multi arc of capacity n. Therefore, each arc is used with its full capacity and each sub flow provides an independent set with exactly three elements of M.

Theorem 3.6. *Max flow with overall capacity, integral flows and without storage is strongly NP-hard.*

Proof. This theorem is a direct consequence of the proof of the reduction above. First, we prove that if the 3-PARTITION instance is solvable, then we can find a flow of value n in the given time horizon. Second, we prove that if the multi-graph has a flow of value n, the corresponding 3-PARTITION instance is solvable.

For the first part, we take a solution to the 3-PARTITION instance and use it to create the desired flow. Let $S_1, S_2, ..., S_n \subset M$ be a solution of the 3-PARTITION instance M. With each set of the solution we create a flow of value one like we did it in the proof of Theorem 3.4. These flows can be combined since each element of M is in exactly one of the sets in the solution, each arc in E_1 is used by exactly one unit of flow, and therefore is feasible. The other $n-1$ flows use the bypass arc in E_2. Additionally, the flows are within capacity and also within time horizon T since the sum of the corresponding set is L and each set contains three elements, so the travel time of these flows is $L + 3 \cdot \sum_{m \in M} m$.

Now we come to the second part. For this, we take an integral flow of the multi-graph with value n. Since it is an integral flow, we can split the flow into n sub flows. For each of these we create a set $S_i \subset M$ the same way we did it in the second part of the proof of Theorem 3.4. These sets cannot have more than three elements, because then the corresponding flow would exceed the time horizon. If no set has more than three elements, all sets have exactly three elements, because all arcs in E_1 must be used in order to achieve a flow of value n. Therefore, all elements of M are distributed over the sets and $k = 3n$. In addition, the sum of each set is L because if one set had less than L another one would have more than L, which would correspond to a sub flow that violates the time horizon. □

We can also adapt this reduction, such that it works for undirected graphs.

Corollary 3.7. *Theorem 3.6 is also true for undirected graphs.*

Proof. Take the multi-graph and add $4 \cdot \sum_{m \in M} m$ to all arcs and extend T by $(|M| + 2) \cdot 4 \cdot \sum_{m \in M} m$. This way a feasible flow can use at most $|M| + 2$ arcs. Since the shortest path from the source to the sink needs $|M| + 2$ arcs, it can only use arcs in the same direction as in the directed graph. □

cost	dynamic capacity	dynamic travel time	Complexity strong flow conservation	Complexity weak flow conservation
-	-	-	P	P
-	-	X	weakly NP-Hard	?
-	X	-	weakly NP-Hard	?
-	X	X	weakly NP-Hard	?
X	-	-	weakly NP-Hard	weakly NP-Hard
X	-	X	weakly NP-Hard	weakly NP-Hard
X	X	-	weakly NP-Hard	weakly NP-Hard
X	X	X	weakly NP-Hard	weakly NP-Hard

Table 3.1: Overview of cost, dynamic travel time and dynamic capacity.

3.3 Limited Capacity

We have proven that allowing dynamic arcs or having an overall capacity instead of normal capacity makes the max flow problem NP-hard. In this small section we see, that it is still NP-hard, if we limit the network to 0/1 capacities. For Theorem 3.1 and Theorem 3.4 limiting the capacity does not change much since all arcs have a capacity that does not depend on the size of the instance. For Theorem 3.6 we need $|M| \cdot n - 2 + 2 \cdot n$ extra arcs so the reduction is still polynomial.

Corollary 3.8. *Theorem's 3.1,3.4 and 3.6 are also true, even if the capacity is limited to 0/1 capacities.*

3.4 Combination of Some Settings

In this section we give a short overview that describes the combination of different variations and their analysis.

Skutella [Sku09] has shown that the min cost flow over time problem is NP-hard with the foundation of Klinz and Woeginger [KW04]. We can compute a static min cost flow on the time-expanded network with linear programming and therefore have a pseudo polynomial algorithm for the min cost flow over time problem. This algorithm also works with dynamic capacities and dynamic travel times. Therefore, we get this Table 3.1 as an overview of the combination of cost, dynamic travel time and dynamic capacity. The complexity is the same for undirected graphs due to Corrollary 3.3,3.5 and 3.7.

In Section 3.3 we proved that allowing only 0/1 capacities does not make the problem easier. Therefore, Table 3.1 is also true for 0/1 capacities. However, if we have only 0/1 travel times none of the reductions are polynomially any more and therefore, the problem could be polynomially solvable.

The first attempt of an polynomial time algorithm for dynamic max flow with 0/1 travel times is naturally a divide and conquer approach. However, this approach does not work, due to the following lemma.

Lemma 3.9. *If the travel time of one arc changes in a time period of length one it may influence the usability of temporally repeated flows for arbitrary large periods of time, thus invalidating the use of the divide and conquer approach.*

Proof. We give an example with two settings, where in setting one a temporally repeated flow can be used for a max flow, but in setting two the temporally repeated flow cannot be used for an arbitrary large periods. A visualization of the upcoming example can be seen in Figure 3.5. We compare two settings that only differ in the travel time of one arc in a time period of length one. Let $T \in \mathbb{N}$.In setting one, the travel time of (s, a) is one for $\theta < 1$ and otherwise zero. In setting two, the travel time of (s, a) is one for $\theta < 1$ and some

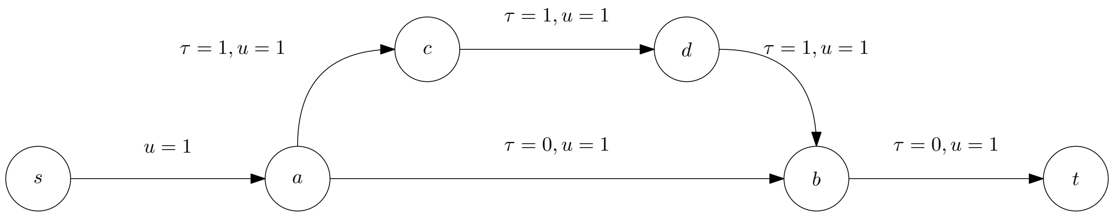

Figure 3.5: Example network the change of a single arc in a network that influences the max flow of the whole time horizon.

period $z \leq \theta < z + 1$ for some $z = 3c + 1$ with $c \in \mathbb{Z}$, such that $1 < z < T$ and otherwise zero so the same as in setting one. When computing a max flow on both settings, they have the same value but the flow is different. In setting one, the temporally repeated flow using arcs $(s, a), (a, b), (b, t)$ is a max flow. In setting two the flow crossing arc (s, a) is sent along the arcs $(a, c), (c, d), (d, b)$ instead of (a, b) for each time period $[3 \cdot x + 1, 3 \cdot x + 2)$ with $x \in \mathbb{Z}_{\geq 0}$ until z, otherwise the flow is the same as in setting one. The mentioned max flow for setting one cannot be used for a max flow in setting two in time period $[0, z)$. $\square$

4. Discretization

Fleischer and Tardos showed that any feasible discrete dynamic flow can be transformed into a feasible continuous dynamic flow "by defining the flow rate x on arc (i, j) in time interval $[\theta, \theta + 1)$, $\theta \in \mathbb{Z}$" [FT98] as $f((i, j), \theta)$. If the time horizon is integral, the transformed flow is still optimal. However, this is only for static travel time and static capacity. It is clear that the transformation produces feasible and optimal continuous flows if the arcs change at integral times and all travel times are integral. But if we allow the functions that describe the capacity or travel time to be piecewise constant and change at no integral points in time then we need some more adjustments.

We can stretch the time such that each change of an arc occurs on an integral point in time. To achieve this, we multiply each point in time with the inverse of the greatest common divider of all points in time where a change occurs. Note, that we can stretch the time by the inverse of $T \bmod 1$ to enable time horizons that are not integral. The same can be done for none integral travel times. Also note that the transformation takes exponential time if the representation of the numbers is not limited in accuracy. For example assume an arc changes its travel time at $1 \cdot 2^{-x}$ with $x \in \mathbb{N}$ which means that the binary representation uses at least $x + 1$ digits, but the transformation stretches T by at least 2^x which is exponential.

We can also transform a feasible continuous flow into a feasible discrete flow by defining $f_{ij}(\theta)$ to be equal to the total of flow sent into arc (i, j) in time interval $[\theta, \theta + 1)$. The produced flow is feasible since the capacity and travel time do not change in that time frame.

5. Conclusion

In this thesis we presented that the dynamic max flow problem with dynamic travel times or dynamic capacities is weakly NP-hard. That is interesting, since the dynamic max flow problem is solvable in polynomial time with static arcs. We also presented that no matter how we combine these Variations with costs the problem is still weakly NP-hard. We can even require 0/1 capacities without changing the hardness. We introduced a different concept of capacity called overall capacity and showed that the dynamic max flow problem with overall capacities is weakly NP-hard and even strongly NP-hard if we require integral flows.

Some Future work could be to research whether the max dynamic flow problem with dynamic travel times or dynamic capacities is still NP-hard if we require 0/1 travel times. It might be possible that the integral flows are not required for the reduction from 3-Partition to max dynamic flow with overall capacities, therefore it might be worth looking into. There could also be a discretization for non constant functions.

Bibliography

[FF62] Delbert Ray Fulkerson and Lester R Ford. *Flows in networks*. Princeton University Press Princeton, 1962.

[FT98] Lisa Fleischer and Éva Tardos. Efficient continuous-time dynamic network flow algorithms. *Operations Research Letters*, 23(3-5):71–80, 1998.

[KW04] Bettina Klinz and Gerhard J Woeginger. Minimum-cost dynamic flows: The series-parallel case. *Networks: An International Journal*, 43(3):153–162, 2004.

[Sku09] Martin Skutella. An introduction to network flows over time. In *Research trends in combinatorial optimization*, pages 451–482. Springer, 2009.