Heterogeneous Multi-Commodity Network Flows Over Time

Yifen Li[1,4], Xiaohui Bei[2], Youming Qiao[3], Dacheng Tao[4], and Zhiya Chen[1]

[1] School of Traffic and Transportation Engineering, Central South University, Changsha, China. `{yifen.li,chzy}@csu.edu.cn`.

[2] School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. `xhbei@ntu.edu.sg`

[3] Centre for Quantum Software and Information, University of Technology Sydney, Australia. `jimmyqiao86@gmail.com`

[4] UBTECH Sydney AI Centre, School of Computer Science, Faculty of Engineering, The University of Sydney, Darlington, NSW 2008, Australia. `dacheng.tao@sydney.edu.au`

Abstract. In the 1950's, Ford and Fulkerson introduced dynamic flows by incorporating the notion of time into the network flow model (*Oper. Res.*, 1958). In this paper, motivated by real-world applications including route planning and evacuations, we extend the framework of multi-commodity dynamic flows to the heterogeneous commodity setting by allowing different transit times for different commodities along the same edge.

We first show how to construct the time-expanded networks, a classical technique in dynamic flows, in the heterogeneous setting. Based on this construction, we give a pseudopolynomial-time algorithm for the quickest flow problem when there are two heterogeneous commodities. We then present a fully polynomial-time approximation scheme when the nodes have storage for any number of heterogeneous commodities. The algorithm is based on the condensed time-expanded network technique introduced by Fleischer and Skutella (*SIAM J. Comput.*, 2007).

Keywords: Multi-Commodity Network Flow · Dynamic Flow.

1 Introduction

Network flows form a well-studied and hugely successful area in optimisation, with many deep theorems and efficient algorithms. Still, in some real-world applications, it is natural to augment the basic network flow model further. One such example, described in [13], is to consider scheduling cars in a traffic network. In this setting, it is clear that *time* is an essential factor to take into consideration. Therefore, it is natural to associate each edge in the network a *transit time*, and consider flows that can vary with time. It is not hard to come up with other examples in production systems, communication networks, and financial flows, where time plays a key role in the corresponding network flow problems.

A formal study of network flows where flows vary with time was initiated in the 1950's by Ford and Fulkerson in their seminal works [7, 8]. It has now become a rather mature area with many basic questions answered. We refer the reader to [15] for an excellent introduction. There are also several nice surveys [1, 2, 13, 14], and the thesis of Hoppe [11], where the reader can find an abundance of information. In the literature, several names have been used for network flows with flow transit times, including flows over time [5], dynamic flows [13], and time-dependent flows [3]. In this paper we adopt flows over time, following works of e.g. Fleischer and Skutella [5, 15].

In this paper, we consider a further augmentation to the multi-commodity network flows over time model. *The key assumption is that various commodities can have different speeds when traveling along the same edge.* We shall call such commodities as *heterogeneous*, as opposed to the *homogeneous* commodities in previous models where commodities have the same speed along a fixed edge.

To see the motivation of doing so, consider the following setting. Suppose a factory needs two types of raw materials, material A and material B, for production. Each of the materials A and B needs to be transported by special trucks in a common road network. When traveling along the same road, these two types of trucks can have different speed limits. To make things more interesting, it is possible that different roads have different speed limits for even the same truck. It is also not hard to come up with other situations in say emergency evacuation, where different commodities or agents have different speeds when traveling along the same edge.

1.1 An overview of the heterogeneous model and our results

In this subsection, we outline our model, review some results from literature, and briefly introduce our results.

The heterogeneous model. The above discussions motivate us to study the *multi-commodity network flows over time* problem with heterogeneous commodities. Recall that, compared to the homogeneous commodities that have been studied previously, heterogeneous commodities may have different speeds when traveling along the same edge. This amounts to setting different transit times for different commodities for each edge.

A key new feature of the heterogeneous model is that, because of the differences in speeds, it is possible for a faster commodity to catch up with a slower one in the middle of an edge, therefore causing a violation of the capacity constraint. We shall refer to such event as a *collision*, where flows of multiple commodities sent at different times meet at the same point within an edge. A concrete example is shown in Figure 1.

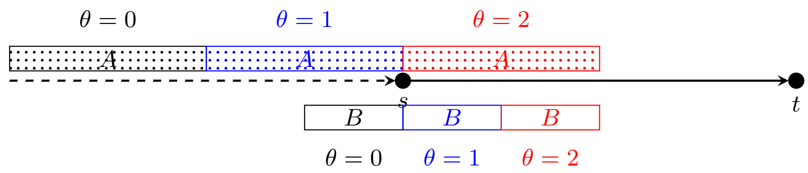


Fig. 1: An example of an edge $e = (s, t)$ of capacity 1, and two heterogeneous commodities A and B, where commodity A flows twice as fast as commodity B. Suppose one unit of B flows into edge e in the time interval $[0, 1)$, and one unit of B flows into e in the time interval $[1, 2)$. The two flow packets never overlap at point s. However, because commodity A flows faster than B, its packet would catch up and collide with the packet of B, hence causing a capacity violation in the middle of edge (s, t).

How to handle collisions will be the key technical problem in this model. To the best of our knowledge, no studies on such collision issues have yet addressed these issues adequately.

For network flows over time, the time can be either discrete or continuous, and the nodes may have storage or cannot – if nodes have storage, then the flow can be held at this node and only released later if needed. In this paper, we work in the *continuous-time* model, and most of our results *require node storage*. For a detailed description of the model, see Section 3.

In the *Heterogeneous Multi-commodity Flows over Time* problem (HeteroMFTfor short), we are given a network with capacities and transit times, a time horizon, and for each commodity a source node, a sink node, and a demand. The goal is to obtain a multi-commodity flow over time within the time horizon satisfying the demands, if there exists one. For a formal definition of the problem, see Definition 2. Clearly, HeteroMFT is closely related to the *quickest* version of the heterogeneous multi-commodity flow over time problem, which asks to find a multi-commodity flow that satisfies the demands of all commodities from their sources to their respective destinations within the minimal time horizon.

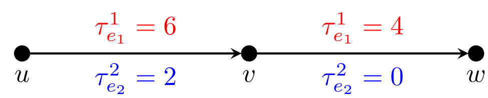
Review of some works in the homogeneous setting. Since the heterogeneous model is a generalisation of the homogeneous model, it is necessary to review some results from the homogeneous multi-commodity dynamic network flow problem (HomoMFT).

Using the classical technique of time-expanded networks [7], HomoMFT can be solved in pseudo-polynomial time, i.e. polynomial in the time horizon. (A true polynomial-time algorithm needs to run in time polynomial in the logarithm of the time horizon.) From the perspective of approximation algorithms for HomoMFT, a breakthrough result is a fully polynomial-time approximation scheme (FPTAS) in the setting of bounded costs and with storage [5]. In [9], the HomoMFT problem is shown to be weakly NP-hard for two or more commodities, and it is also NP-hard to design fully polynomial-time approximation scheme (FPTAS) for the quickest HomoMFT with simple paths and without storage.

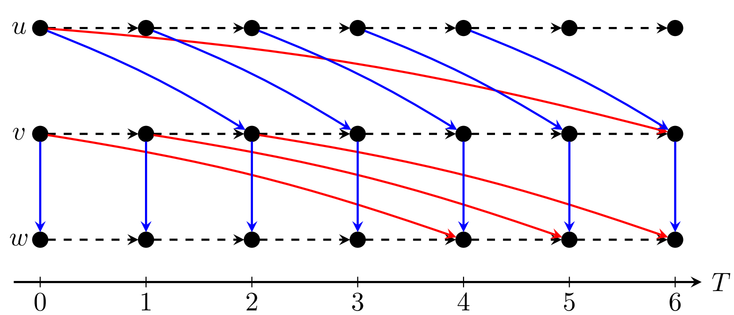
Since the HeteroMFT problem is a generalisation of the HomoMFT problem, the NP-hardness results for HomoMFT apply to HeteroMFT as well. Therefore, we focus on designing approximation algorithms for the HeteroMFT problem.

Our first result: time-expanded networks in the heterogeneous setting. We first examine the classical time-expanded network technique. Briefly speaking, in the homogeneous setting, given a network $G = (V, E)$, a time-expanded network $\tilde{G}$ with time horizon T is built by replicating T copies of V, with each copy called a layer. Then for each edge in the original graph with transit time τ, connect the corresponding nodes in ith layer and the $(i + \tau)$th layer.

In the heterogeneous setting, the construction of time-expanded networks is trickier. Suppose the number of commodities is k. As the transit times are different for different commodities on a fixed arc in the heterogeneous setting, it is natural to split an arc in the original network into k arcs in the time-expanded network, one for each commodity. Figure 2 gives such an example.



(a) dynamic network G



(b) time-expanded network G^T

Fig. 2: A dynamic network with two commodities and its corresponding time-expanded network with time horizon $T = 6$.

Here, we come across the first difficulty caused by the heterogeneity regarding the capacity constraints. First, note that the capacities of these k arcs starting from the same layer need to be considered altogether as not to exceed the capacity. However, this just suggests that the flows on this arc in the original network do not exceed the capacity at the tail of the arc. In order to avoid *collison*, we also need to examine the capacities of several arcs starting from different layers. At first sight, this seems to involve enumerating every point in e, leading to possibly exponentially many constraints. Fortunately, Proposition 1 indicates that only a polynomial number of additional capacity constraints need to be considered.

While the above gives a proper definition of time-expanded networks in the heterogeneous setting, there is a more serious problem which prevents it from yielding a pseudo-polynomial time algorithm for the HeteroMFT problem. This is because, the key observation for using time-expanded networks in the homogeneous setting is the following (see e.g. [15, Lemma 4.4]): A feasible flow over time in G with time horizon T yields a feasible static flow in $\tilde{G}$ (by averaging according to each time interval), and the inverse direction is also true (by a straightforward construction). However, in the heterogeneous setting, while we can still construct a feasible flow over time from a feasible static flow, the inverse direction does not necessarily hold, as the averaging technique no longer works due to the collision issue.

Despite the above difficulty, we show that by incorporating a further observation, the averaging technique still works for *two* commodities (Proposition 2), giving a pseudopolynomial-time algorithm for this case.

Theorem 1. *There exists a pseudopolynomial-time algorithm for the HeteroMFT problem when the number of commodities is* 2*.*

The proof for Theorem 1 does not apply to more than two commodities; see Remark 1 for some discussions. We leave designing a pseudo-polynomial time algorithm for more than two commodities as an intriguing open problem.

Our second result: an FPTAS for HeteroMFT. Given the problems encountered in the time-expanded network construction, it is perhaps rather surprising that a fully polynomial-time approximation scheme (FPTAS) can still be achieved for HeteroMFT, when the nodes are allowed to have storage. In Theorem 2, we present such an FPTAS.

Theorem 2. *For any $\epsilon > 0$, a $(1+\epsilon)$-approximate solution to the HeteroMFT problem can be found in time polynomial in the input size and $\frac{1}{\epsilon}$.*

Note that by [9], unless P=NP, there is no FPTAS for the quickest multi-commodity flow problem when node storage is prohibited and flows are only sent on simple paths, even in the homogeneous setting. So allowing node storage is unavoidable.

An FPTAS for the homogeneous version of the problem was given in [5], and our algorithm builds upon and generalises that algorithm. More specifically, we utilize the *condensed time-expanded network* technique introduced there, which are time-expanded networks with longer time intervals. Figure 3 gives an illustration of this idea, following the example in Figure 2.

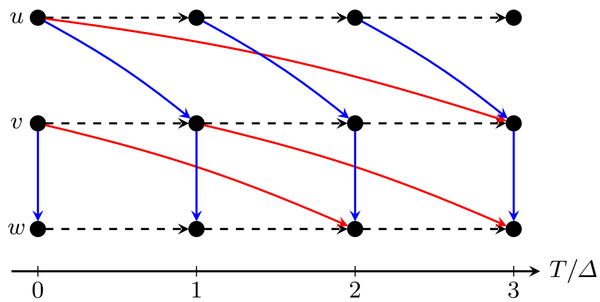


Fig. 3: Δ-condensed time-expanded network G^T/Δ with $\Delta = 2$

Overcoming the difficulties brought by the heterogeneity of the commodities requires some non-trivial technical works. Indeed, in the construction of time expanded networks, we need to adjust existing constraints and introduce new constraints on the static flow network to ensure that the solution corresponds to a feasible flow over time, and that the static network problem is of polynomial size. This means that the analysis in [5] cannot be directly applied to prove the correctness of this algorithm. More specifically, the averaging step there no longer yields a feasible flow in the time expanded network in the heterogeneous setting. Our main technical contribution is to show that the feasibility is still approximately preserved thanks to a previous flow smoothing step , which is a key step for the FPTAS.

1.2 Structure of the paper

In the following, we give a detailed account of our results. In Section 2, to prepare for introducing our model, we review the static and homogeneous dynamic network flow models. In Section 3, we formally define the heterogeneous multi-commodity flow over time model. In Section 4, we describe the time-expanded network construction in the heterogeneous setting. In Section 5, we present the FPTAS for the HeteroMFT problem. Due to space constraints, some proofs are omitted and can be found in the full version of this paper.

2 Review of static and homogeneous flows over time

In this section we review the classic static multi-commodity network flow model, and the homogeneous multi-commodity network flow over time model.

Static flows in networks. In a static network flow problem, we are given a network (directed graph) $G = (V, E)$ with $|V| = n$ nodes and $|E| = m$ edges. Each edge $e \in E$ has a capacity $u_e : E \to \mathbb{R}_{\geq 0}$, which bounds the total amount of flow allowed to go through this edge at any time. For each edge $e = (v, w)$, we denote

$\mathsf{tail}(e) = v$ and $\mathsf{head}(e) = w$. We also let δ_v^+ (resp. δ_v^-) to denote the set of edges $e \in E$ going out of v, i.e. $\mathsf{tail}(e) = v$ (resp. going into v, i.e. $\mathsf{head}(e) = v$).

Our goal is to transport k types of commodities through the same network G by sharing edges. More specifically, assume that each commodity $i \in [k] := \{1, 2, \ldots, k\}$ has a source node $s_i \in N$, a sink node $t_i \in N$, and a demand d_i that represents the amount of commodity i that needs to be transported from s_i to t_i.

A *static flow* x in network G allocates a flow value $x_e^i : E \to \mathbb{R}_{\geq 0}$ to each edge $e \in E$ and each commodity $i \in [k] := \{1, \ldots, k\}$. A static flow is called *feasible* if it satisfies the following constraints.

$$\forall e \in E, \quad 0 \leq \sum_{i \in [k]} x_e^i \leq u_e \qquad \text{(capacity constraints)} \tag{1}$$

$$\forall i \in [k], v \in V \backslash \{s_i, t_i\}, \quad \sum_{e \in \delta_v^-} x_e^i - \sum_{e \in \delta_v^+} x_e^i = 0 \qquad \text{(flow conservation constraints)} \tag{2}$$

$$\forall i \in [k], \quad \sum_{e \in \delta_{t_i}^-} x_e^i - \sum_{e \in \delta_{t_i}^+} x_e^i = d_i \qquad \text{(demand constraints)} \tag{3}$$

Network flows over time. When taking *time* into consideration, we arrive at the network flow over time problem. There are two main approaches to model time. The first one is the discrete-time model, first studied by Ford and Fulkersen [7,8]. The second one is the continuous model. Fleischer and Tardos showed strong connections between these two models [6]. In this paper, we focus on a continuous-time model, mostly following notations in [5, 15].

Another feature of the flow over time model is that storing flows in the intermediate nodes becomes possible. That is, we may assume that intermediate nodes have storage that could hold inventory of (any amount of) flow before sending it forward. Allowing storage is a common assumption in most previous works on flows over time. On the other hand, to fit certain applications such as telecommunications, one could also consider a model in which storage is limited, or even no storage is allowed, at any intermediate node. Then the flow conservation constraints simply change the inequality condition to equality. Flows over time with no storage or restricted storage were studied in several works [4, 10–12]. In this paper we will adopt the model *with* intermediate storage.

For network flows over time, apart from capacity u_e, every edge $e \in E$ also has a transit time $\tau_e \geq 0$ that specifies the time it takes for a unit of flow of a commodity to travel from $\mathsf{tail}(e)$ to $\mathsf{head}(e)$. That is, the flow for a commodity sent at time θ from $\mathsf{tail}(e)$ will reach $\mathsf{head}(e)$ at time $\theta + \tau_e$.

The following definition of dynamic flows (named *flows over time* in this paper) is standard (cf. e.g. [15, Definition 2.1]).

Definition 1. *A* flow over time f *with time horizon* $T \geq 0$ *is a Lebesgue-integrable function* $f_e^i : \mathbb{R} \to \mathbb{R}^{\geq 0}$ *for each edge* $e \in E$ *and* $i \in [k]$.

We will only consider flows that can arrive at its destination by time T*, that is, we require that* $f_e^i(\theta) = 0$ *for all* $\theta \geq T - \tau_e$ *or* $\theta < 0$.

That is, for any $\theta \in [0, T)$ and any commodity i, $f_e^i(\theta)$ denotes the flow *rate* of commodity i going into the tail of edge e at time θ.

We now discuss on feasibility constraints for a multi-commodity flow over time $f = (f_e^i)$ with time horizon T. Let $G = (V, E)$ be the underlying directed graph. Suppose that there are k types of commodities, and each commodity has a source node s_i and a sink node t_i.

– The *capacity constraint* is

$$\forall e \in E, \theta \in [0, T), 0 \leq \sum_{i=1}^{k} f_e^i(\theta) \leq u_e. \tag{4}$$

– *Flow conservation constraints.* Recall that storing flows in the intermediate nodes may be allowed. This leads to the following weak flow conservation constraints.

$$\forall v \in V \backslash \{s_i, t_i\}, i \in [k], \theta \in [0, T], \sum_{e \in \delta_v^-} \int_0^{\theta - \tau_e} f_e^i(\xi) d\xi - \sum_{e \in \delta_v^+} \int_0^{\theta} f_e^i(\xi) d\xi \geq 0 \tag{5}$$

In the above equation, when $\geq$ is replaced by $=$, the conservation constraint is called strict.

– *Demand constraints.* Finally, demand constraints state that for each commodity i, the net flow that has reached sink t_i by time T should equal its demand d_i.

$$\forall i \in [k], \sum_{e \in \delta_{t_i}^-} \int_0^{T - \tau_e} f_e^i(\xi) d\xi - \sum_{e \in \delta_{t_i}^+} \int_0^{T} f_e^i(\xi) d\xi = d_i \tag{6}$$

3 Our model: heterogeneous multi-commodity dynamic flow

In this paper, we generalise the above multi-commodity network flow over time model by introducing speed heterogeneity among different commodities. In the previous network flow over time model, on each edge all commodities are assumed to have the same speed, i.e., the flows are *homogeneous*. We propose a model in which commodities may have different speeds on the same edge, and we refer to this as the *heterogeneous multi-commodity flow over time* model. This is achieved by allowing an edge e to have k transit times $\{\tau_e^1, \tau_e^2, \ldots, \tau_e^k\}$, for which τ_e^i is the time needed for one unit of flow of commodity i to go through edge e. In this paper, we also assume that the transit times τ_e^i are *integral*, which is a realistic assumption for most potential applications.

Flow feasibility. We now define the feasibility constraints of a heterogeneous multi-commodity flow over time.

The flow conservation constraint 5 and the demand constraint 6 are defined with regard to each commodity. So they carry over from homogeneous to heterogeneous setting, after changing τ_e to τ_e^i, namely:

$$\forall v \in V \setminus \{s_i, t_i\}, i \in [k], \theta \in [0, T], \quad \sum_{e \in \delta_v^-} \int_0^{\theta - \tau_e^i} f_e^i(\xi) d\xi - \sum_{e \in \delta_v^+} \int_0^{\theta} f_e^i(\xi) d\xi \geq 0. \tag{7}$$

$$\forall i \in [k], \quad \sum_{e \in \delta_{t_i}^-} \int_0^{T - \tau_e^i} f_e^i(\xi) d\xi - \sum_{e \in \delta_{t_i}^+} \int_0^{T} f_e^i(\xi) d\xi = d_i, \tag{8}$$

We need to pay special attention to the capacity constraints. Because of the heterogeneity of commodity speeds, it no longer suffices to only require capacity constraints only at the entrance of each edge. That is, when k commodities move in a common pipeline at various speeds, some fast commodity may catch up with the slower commodities to create congestion in the middle of the edge. Figure 1 above shows a concrete example.

Therefore, we need to require that at any moment, *at any point* of any edge, the sum of the rates of all flows together must not exceed the capacity of that edge. The capacity constraint is then the following.

$$\forall e \in E, \theta \in [0, T), \alpha \in [0, 1] : \quad \sum_{i=1}^{k} f_e^i(\theta - \alpha \cdot \tau_e^i) \leq u_e. \tag{9}$$

The heterogeneous multi-commodity flow over time problem. In this paper, we focus on the problem of transporting each commodity to their respective destinations within a pre-set time horizon.

Definition 2. *An instance of the* heterogeneous multi-commodity flow over time *problem, denoted by* HeteroMFT, *consists of the following.*

Input: *A network $G = (V, E)$. For every edge $e \in E$, a capacity $u_e \geq 0$. There are k commodities. For every commodity $i \in [k]$, a source node $s_i \in V$, a sink node $t_i \in V$, and a demand $d_i \geq 0$. For every $e \in E$ and $i \in [k]$, the transit time of commodity i on e is $\tau_e^i \geq 0$. A time horizon $T \geq 0$.*

Output: *A feasible multi-commodity flow over time with time horizon T satisfying the given demands, if there exists one.*

Correspondingly, the homogeneous multi-commodity flow over time Problem, studied in e.g. [5, 9], is denoted by HomoMFT.

Approximations. In Section 5, we will focus on FPTAS for HeteroMFT. Here, a $(1 + \epsilon)$-approximate solution to HeteroMFTmeans that the output is a feasible multi-commodity flow over time with time horizon $(1 + \epsilon)T$ satisfying the given demands.

4 Time-expanded networks in the heterogeneous setting

4.1 Time-expanded networks: from homogeneous to heterogeneous

As already mentioned in Section 1.1, a classical technique for tackling flows over time problem is to construct time-expanded networks. The underlying principle for using time-expanded networks lies in the conversions between feasible static flows in the time-expanded network and feasible flows over time in the original network. To convert feasible flows over time in the original network to feasible static flows in the time-expanded network, we need the averaging technique. That is, by averaging the flows in each time unit, we can start from any feasible flow over time, to obtain a "stair-case" like flow which is also feasible and completes within the same time horizon.

A surprising feature of the heterogeneous setting is that the averaging technique does not result in a feasible flow, when the number of commodities is larger than 2. Even for the case of two commodities, some subtle argument is needed. In the following, we first define time-expanded networks in the heterogeneous setting. As the readers will see, the time-expanded networks need to be adjusted to accommodate different speeds, and the feasibility condition is also more complicated due to the need to avoid collisions in the middle of the edges.

4.2 Heterogeneous time-expanded networks

We now present a construction of time-expanded static networks in the heterogeneous setting. Let $G = (V, E)$ be a network with capacities u_e^i, transit times τ_e^i, and time horizon T. We assume T and τ_e^i are integers.

First, we construct a static network $G^T = (N^T, E^T)$ as follows.

- The set of nodes N^T consists of $T+1$ copies of the set of vertices N, labeled from N_0 to N_T. For any $v \in N$ and $\zeta = 0, 1, \ldots, T$, v_ζ in G^T is the ζth copy of node v.
- For every commodity i and every edge $e = (v, w) \in E$ and $\zeta = 0, 1, \ldots, T - \tau_e^i$, there is an edge $e_\zeta^i = (v_\zeta, w_{\zeta+\tau_e^i})$ in E^T.
- For each $v \in N$ and $\zeta = 0, 1, \ldots, T-1$, there is an edge $(v_\zeta, v_{\zeta+1})$. It is used to model storage at the node.
- Finally, for each commodity i, its source node is s_0^i, and its sink node is t_T^i.

The following table summarizes the correspondences of the network structures.

	Dynamic network	Time-expanded network
Network	$G = (N, E)$	$G^T = (N^T, E^T)$ $N^T = N_0 \cup \cdots \cup N_\zeta \cup \cdots \cup N_{T-1}$ $E^T = \bigcup_{e \in E} \{e_0^i, \ldots, e_\zeta^i, \ldots, e_{T-\tau_e^i}^i \mid i = 1, 2, \ldots, k\}$ $\bigcup_{v \in N} \{(v_\zeta, v_{\zeta+1}) \mid \zeta = 0, 1, \ldots, T\}$
Nodes	$v \in N$	$v_i \in N_i$
Edges	$e = (v, w) \in E$	$e_\zeta^i = (v_\zeta, w_{\zeta+\tau_e^i})$

See also Figure 2 (a) and (b) for an instance of a dynamic network and its corresponding time-expanded network.

Next, let us define flow feasibility in this situation. The constraints for feasibility will be closely related to the construction of flows over time, so let us first illustrate the desired static-to-dynamic conversion here.

Definition 3 (Static to flow over time conversion). *Let x be a static flow in G^T in which $x^i(e)$ denotes the flow of commodity i on edge e. For $e^i_\zeta = (v_\zeta, w_{\zeta+\tau^i_e})$, we interpret the value of $x^i(e^i_\zeta)$ as the flow rate of commodity i entering edge (v, w) in the time interval $[\zeta, \zeta + 1)$. This gives a flow f_x in the original dynamic network.*

Our goal now is to introduce appropriate constraints on the static flow in the time-expanded network, so that the above procedure can convert it into a feasible flow over time.

1. (Flow conservation.) This constraint is the same as for static network flows; see Equation 2.
2. (Each edge is exclusive to a specific commodity.) Note that for problem HeteroMFT, for $v_\zeta \in N^T$, every edge $e = (v, w)$ in G is converted into k different edges $(v_\zeta, v_{\zeta+\tau^i_e})$ in E^T, where k is the number of commodities. Since each edge e^i_ζ is G^T is now catered only for the specific commodity i, we need to add further constraints to the feasible static flow conditions to forbid other commodities to travel along this edge.
We therefore add the following constraints
$$x^i(e^j_\zeta) = 0 \qquad \forall \zeta \in \{0, 1, \ldots, T-1\}, i \neq j. \tag{10}$$
3. (Capacity constraints at edge tails.) Again, since we split an edge e at time ζ into k edges e^i_ζ, we need to impose
$$\sum_{i\in[k]} x^i(e^i_\zeta) \leq u_e.$$
4. (Capacity constraints along the edges.) The above capacity constraints at edge tails, when interpreted in the context of the flow over time f_x as defined in Definition 3, only impose the flows not to exceed the edge capacity at the entrance of each edge. However, the capacity constraints for a feasible flow over time in G, as shown in Equation 9, are defined not only at the entrance of each edge, but also at *every point* along the edge; see Figure 1 for an example where collision happens in the middle of an edge.
To take care of that issue, let us first focus on the set of edges $\{e^i_\zeta : i \in [k], \zeta \in \{0, 1, \ldots, T-1\}\}$ which are derived from the edge $e \in E$. To analyze collisions happening in the middle of edges, we need to identify those $(\zeta_1, \ldots, \zeta_k)$, $\zeta_i \in \{0, 1, \ldots, T\}$, such that there exist $\chi_i \in [\zeta_i, \zeta_i + 1)$, and the flows sending commodity i at time χ_i arrive at the same point along the edge e at the same time. For such $(\zeta_1, \ldots, \zeta_k)$, we need to ensure that the sum of static flows along $e^i_{\zeta_i}$ is within the capacity u_e.

At first sight, this seems to involve enumerating every point in e, leading to infinitely many constraints. Fortunately, this can be reduced to finitely many points in e. Thanks to the following Propostion 1, we can first compute, for each pair of commodities $j, j' \in [k]$, every two time points $\zeta_j, \zeta_{j'} \in \{0, \dots, T-1\}$, and every edge e, whether flow of commodity j at time ζ_j meets with flow of commodity j' at time $\zeta_{j'}$, or catches up with commodity j' at time $\zeta_{j'}+1$, at some point $\alpha \in [0,1]$ along e at time t. If so, for every $i \neq j, j'$, compute ζ_i such that $[\zeta_i, \zeta_i + 1)$ covers α at time t. After computing such ζ_i's, $i \in [k]$, we set up the constraint that

$$\sum_{i \in [k]} x^i(e^i_{\zeta_i}) \leq u_e. \tag{11}$$

Proposition 1. *Suppose flows only change rate at integral time steps. Given $\alpha \in [0,1]$ and time θ. For every commodity $i \in [k]$, let χ_i be the point in time such that $\chi_i + \alpha\tau^i_e = \theta$, and let $\zeta_i = \lfloor \chi_i \rfloor$.*

Then there exists another set of times $\{\chi'_i \in [\zeta_i, \zeta_i + 1]\}$, $\alpha' \in [0,1]$, and time θ', such that we have $\chi'_i + \alpha'\tau^i_e = \theta'$ for each $i \in [k]$, and one of the following cases hold:

1. *$\alpha' = 0$, and $\forall i, j \in [k]$, $\zeta_i = \zeta_j$;*
2. *there exist $\ell \neq j \in [k]$, such that $\chi'_\ell = \zeta_\ell$, $\chi'_j = \zeta_j$;*
3. *there exist $\ell \neq j \in [k]$, such that $\chi'_\ell = \zeta_\ell$, $\chi'_j = \zeta_j + 1 - \epsilon$ for any small $\epsilon > 0$.*

Proof. If $\alpha = 0$, then we have $\chi_i = \chi_j$ for any $i, j \in [k]$. Since $\chi_i \in [\zeta_i, \zeta_i + 1)$, we have $\zeta_i = \lfloor \chi_i \rfloor = \lfloor \chi_j \rfloor = \zeta_j$.

If $\alpha \neq 0$, then take $\ell \in [k]$ such that $\chi_\ell - \zeta_\ell$ is the minimum among $\chi_j - \zeta_j$, $j \in [k]$. For any $j \in [k]$, let $\chi'_j = \chi_j - (\chi_\ell - \zeta_\ell)$, so $\chi'_\ell = \zeta_\ell$. Note that for any $i, j \in [k]$, we still have $\chi'_i + \alpha\tau^i_e = \chi'_j + \alpha\tau^j_e$.

If there exists $j \in [k]$ and $j \neq \ell$, such that $\chi'_j = \zeta_j$, then case (b) holds. Otherwise, we have $\chi'_j > \zeta_j$ for any $j \neq \ell$.

For any $\delta > 0$ and $j \in [k]$, we have

$$\zeta_\ell + \alpha\tau^\ell_e - \delta = \chi'_j + \alpha\tau^j_e - \delta.$$

Note that the left-hand side is $\zeta_\ell + (\alpha - \delta/\tau^\ell_e)\tau^\ell_e$, and the right-hand side is $\chi'_j + \frac{\tau^j_e - \tau^\ell_e}{\tau^\ell_e} \cdot \delta + (\alpha - \delta/\tau^\ell_e)\tau^j_e$, and we need to ensure that (1) $\alpha - \delta/\tau^\ell_e > 0$, (2) for any $j \neq \ell$, $\zeta_j < \chi'_j + \frac{\tau^j_e - \tau^\ell_e}{\tau^\ell_e} \cdot \delta$, and (3) for any $j \neq \ell$, $\chi'_j + \frac{\tau^j_e - \tau^\ell_e}{\tau^\ell_e} \cdot \delta < \zeta_j + 1 - \epsilon$ for arbitrarily small $\epsilon > 0$. We now increase δ, and take $\chi'_j + \frac{\tau^j_e - \tau^\ell_e}{\tau^\ell_e} \cdot \delta$ to be the new χ'_j for each j, until one of (1), (2), and (3) is violated.

- If (1) is violated, then we are back to the $\alpha = 0$ setting.
- If (2) is violated, then we are in case (b). This means $\tau^j_e < \tau^\ell_e$, i.e. ℓ is slower than j.
- If (3) is violated, then we are in case (c). This means $\tau^j_e > \tau^\ell_e$, i.e. ℓ is faster than j.

This concludes the proof.

After the above adjustments, we can ensure that any static flow in the time-expanded network satisfying all the above constraints corresponds to a feasible flow over time in the original network.

4.3 A pseudo-polynomial-time algorithm for two commodities

In the homogeneous setting, a well-known application of time-expanded networks is a pseudo-polynomial-time algorithm to decide whether a feasible flow exists with time horizon T (cf. e.g. [15, Theorem 4.2]). The key is to realise that a flow over time can be converted to a static flow using the averaging technique, and a static flow can be converted to a flow over time using the obvious transformation in Definition 3.

However, it is not clear that the averaging technique can be applied to the heterogeneous setting in general, due to the possible violations of the capacity constraint due to heterogeneity. Interestingly, when the number of commodities is 2, a simple argument ensures that the averaging technique still works. This will also suggest why the averaging technique cannot work, at least not directly, for three or more commodities.

Given an instance of the HeteroMFT problem, suppose f^i, $i = 1, 2$, are feasible flows over time with time horizon T. We also assume that all transit times are integral. By averaging f^i in $[\zeta, \zeta + 1)$ for $\zeta \in \{0, 1, \ldots, T-1\}$, we obtain a static flow x^i on G^T:

$$\forall e \in E, x(e^i_\zeta) := \int_\zeta^{\zeta+1} f^i_e(\xi) d\xi.$$

Proposition 2. *Let f^i and x^i, $i = 1, 2$, be as above. Then x^i's form a feasible flow on G^T.*

Proof. Note that feasible flows on G^T correspond exactly to those feasible, staircase like, flows over time in the original network. So we need to show that the flow over time f_x corresponding to x as defined in Definition 3 form a feasible flow over time. The flow conservation constraint clearly holds. We then examine the capacity constraint. Fix an edge e, and suppose commodity 1 is faster than commodity 2, i.e. $\tau^1_e < \tau^2_e$. Consider two flow intervals, f^1_e in $[\zeta_1, \zeta_1 + 1)$ and f^2_e in $[\zeta_2, \zeta_2 + 1)$. Suppose the first interval catches up with the second. Then there exists a time $\theta \in [0, T]$, $0 \le \alpha \le 1$, and $\chi_2 \in [\zeta_2, \zeta_2 + 1)$, such that $\theta - \alpha \cdot \tau^1_e = \zeta_1$, and $\theta - \alpha \cdot \tau^2_e = \chi_2$. It follows that $\alpha = (\zeta_1 - \chi_2)/(\tau^2_e - \tau^1_e)$. Note that τ^i_e, $i = 1, 2$, and ζ_1 are integers. So if χ_2 is not an integer, α cannot be 1. We then also have that $(\zeta_1 - \zeta_2)/(\tau^2_e - \tau^1_e) \le 1$. Let $\alpha' = (\zeta_1 - \zeta_2)/(\tau^2_e - \tau^1_e)$. Because $\alpha' \le 1$, the flow sent by f^1_e at time ζ_1 also catches up with the flow sent by f^2_e at time ζ_2 at the α' fraction of e. We can then use the capacity constraint 9 at the α'-fraction of e to conclude that the averaged flows also satisfy the capacity constraint.

Proposition 2 immediately gives the following.

Proof (Proof of Theorem 1). By a binary search, we can determine the optimal time T^* for which there exists a solution to the given instance of HeteroMFT in poly(T^*) rounds. For each time T' guessed during this procedure, we construct the time expanded network, solve the corresponding static flow problem in time polynomial in the input size and T', and convert that static flow (if it is solvable) to a dynamic one using the procedure in Definition 3. Proposition 2 ensures that for T no less than the optimal value in the dynamic network, there exists a feasible flow in G^T. This concludes the proof.

Remark 1. When the number of commodities is more than 2, the argument to prove Proposition 2 does not work, at least directly, due to the following. Suppose the intervals of f_e^i sent at $[\zeta_i, \zeta_i + 1)$, $i = 1, 2, 3$, do overlap at some point. Then we cannot ensure that the flows sent at time ζ_i meet at some time, which causes difficulty as we then cannot use the dynamic capacity constraints.

5 An FPTAS for HeteroMFT

In [5], Fleischer and Skutella designed a fully polynomial-time approximation scheme (FPTAS) for the HomoMFT problem. The key idea in their algorithm is to convert the dynamic network into a static *Δ-condensed time-expanded network*, whose definition we will discuss in detail later, and find a static flow in that network to approximate the optimal flow over time.

In this section we design an FPTAS for the more general HeteroMFT problem and any number of commodities. The main ideas supporting our FPTAS for HeteroMFT are drawn from [5]. However, because of the heterogeneity of the commodity speeds and the possible failure of the averaging technique for more than 2 commodities (see Section 4.3), it is not clear that any feasible flow over time can be converted to a feasible static flow in the condensed time-expanded network, which does hold in the homogeneous setting [5, Lemma 4.1]. Therefore, though our techniques are mostly already in [5], new analyses are needed to show that the best static flow produced by our algorithm is indeed a good approximation to the optimal dynamic problem for HeteroMFT.

Below we first present some preliminaries that support our algorithm, and then explain our algorithm and its analysis.

5.1 Preliminaries

Δ-condensed time-expanded network. The size of the static time-expanded network is linear in the value of time horizon T. Therefore, even though one can find a static flow in that network in polynomial time, it will be polynomial in T and therefore pseudo-polynomial in the input-size. To overcome this issue, Fleischer and Skutella introduced in [5] the *Δ-condensed time-expanded network*. More specifically, when the transit time of each commodity on each edge is always a multiple of $\Delta > 0$, then the time-expanded network G^T can be rescaled to a Δ-condensed time expanded network, denoted by G^T/Δ, in which each unit time

interval now has length Δ. The new network contains only T/Δ copies of N, as depicted in Figure 3. All capacities are also multiplied by Δ because each edge in G^T corresponds to a time interval of length Δ in G^T/Δ.

Paths with delays. In the static flow setting, the well-known flow decomposition theorem states that a flow can be decomposed into a sum of path and cycle flows. In the flow over time setting, such a nice decomposition may not exist in general. Still, every infinitesimal unit of flow can be viewed as following a particular path. Since we allow node storage, we also need to record how long it stays at each node. This leads to the notion of paths with delays from [5, Sec. 4.6]. Let $P = (v_0, \ldots, v_\ell)$ be a path in G, and let $\iota = (\iota_1, \ldots, \iota_\ell)$, $\iota_i \in \mathbb{R}^{\geq 0}$, be a sequence of non-negative numbers. A *path with delays* P^ι is understood as indicating that a flow along P needs to stop at v_j for exactly ι_j time, $j \in \{1, \ldots, \ell\}$. The flow for commodity i along P^ι is then denoted by $f^i_{P^\iota}$.

Given this notion, a flow over time f with time horizon T can be decomposed into (possibly infinitely many) flows over time f_{P^ι} along paths with delays. In particular, the total flow of commodity i entering $e = (v_j, v_{j+1})$ at time θ is

$$f^i_e(\theta) = \sum_{P^\iota : e \in P} f^i_{P^\iota}(\theta - \tau(P^\iota, e)),$$

where

$$\tau(P^\iota, e) := \sum_{s=1}^{j} (\tau^i_{(v_{s-1}, v_s)} + \iota_s) \tag{12}$$

for $e = (v_j, v_{j+1})$. In addition, one can assume without loss of generality that all paths in the decomposition are simple. This is because the flow traveling along a cycle that visits some node v twice can also just wait at node v. Conversely, any flow over time given in the form of f_{P^ι} along paths with delays corresponds to a flow over time f defined on edges. Throughout the remainder of this section, we will discuss a flow over time both in its paths with delays representation f_{P^ι} and in its standard representation f.

5.2 Algorithm and proof outline

The idea of our approximation scheme is to first round up the transit times of each commodity on each edge to the nearest multiple of Δ, for some carefully selected Δ. Then we convert the dynamic network problem into a static Δ-condensed time-expanded network, and solve the quickest static flow problem on that network with the additional set of constraints 10 and 11. Finally we convert the static flow to a feasible flow over time.

We restate Theorem 2 as below here.

Theorem 2. For any $\epsilon > 0$, a $(1 + \epsilon)$-approximate solution to the HeteroMFT problem can be found in time polynomial in the input size and $\frac{1}{\epsilon}$.

The key to Theorem 2 is the following lemma.

Lemma 1. *For any constant* $\epsilon > 0$*, let* $\Delta = \frac{\epsilon^2}{2n}T$*. Let* f *be a heterogeneous multi-commodity flow over time* f *in a dynamic network* G *with demand* $D = (d_1, \ldots, d_k)$ *and time horizon* T*. Let* $T' = (1+\epsilon)^4 \cdot T$*.*

Then there exists another dynamic network G'*, obtained from* G *by modifying transit and delay times, satisfying the following property: in the time-expanded network* $G'^{T'}/\Delta$*, there exists a feasible static flow* x *such that the flow over time* f_x *constructed from* x *by Definition 3 is a feasible flow in* G *with time horizon at most* $(1+\epsilon)^4 \cdot T$*.*

Furthermore, the parameters of G' *can be computed in polynomial time.*

The remaining of this section is devoted to the proof of Lemma 1.

5.3 Proof of Lemma 1

Let $\epsilon > 0$, and set $\Delta = \frac{\epsilon^2}{2n}T$. Starting from a flow over time f with time horizon T in a network G, our goal is to devise a static flow x in the Δ-condensed time-expanded network with time horizon $(1+\epsilon) \cdot T$. The construction of x goes through the following four steps.

Step 1: flow smoothening. Briefly speaking, given any heterogeneous multi-commodity flow over time f in network G with demand D and time horizon T, we can get a smoothened heterogeneous multi-commodity flow over time $\hat{f}$ with the same demand D within time horizon $(1+\epsilon) \cdot T$ and still obeys the capacity constraints in G. Here, smooth means that the rate changes are not drastic. The flow smoothening procedure applies to flows along paths with delays.

Definition 4. *Let* f *be a flow over time in a network* G *with time horizon* T*, and* f_{P^ι} *be a path decomposition of* f*. Given* $\epsilon > 0$*, the* smoothed *flow* $\hat{f}$ *is defined by*

$$\hat{f}^i_{P^\iota}(\theta) = \frac{1}{\epsilon T} \int_{\theta - \epsilon T}^{\theta} f^i_{P^\iota}(\xi) d\xi \tag{13}$$

for all $\theta \in [0, (1+\epsilon)T]$ *and* P^ι *appears in the path decomposition of* f*.*

The smoothed flow $\hat{f}$ enjoys the following property.

Proposition 3. *Let* f *be a flow over time in a network* G *with time horizon* T*. For any* $\epsilon > 0$*, the smoothed flow* $\hat{f}$ *is also a feasible flow over time with time horizon* $(1+\epsilon)T$*, and for any* $\theta \in [0, (1+\epsilon)T]$*,* $\mu > 0$ *and commodity* i*,*

$$|\hat{f}^i_{P^\iota}(\theta) - \hat{f}^i_{P^\iota}(\theta - \mu)| \le \frac{\mu}{\epsilon T} u_P$$

where $u_P = \min_{e \in P} u_e$*.*

Step 2: rounding up the transit times. After smoothing the flow, the next step is to round up the transit time of each commodity on each edge to the nearest multiple of Δ, such that the dynamic network can be feasibly converted to a Δ-condensed time-expanded network. We need to ensure that this rounding procedure will not jeopardize the feasibility and optimality of the solution. Fortunately this can be guaranteed by the smoothness property of the flow over time.

Proposition 4. *Let $\epsilon > 0$ and $\Delta \leq \frac{\epsilon^2}{2n}T$. Let $\tilde{G}$ be the network where the transit time of each commodity on each edge is rounded up to the nearest multiple of Δ. We think of the smoothed flow $\hat{f}$ from Definition 4 as a flow over time in $\tilde{G}$ with the delay on each node also rounded up to the nearest multiple of Δ. Then this flow satisfies the demand of each commodity and finishes with the time horizon $\tilde{T} \leq (1+\epsilon)^2 \cdot T$, and the capacity constraint on each edge is violated by at most a factor of $(1+\epsilon)$.*

Step 3: construct a condensed time-expanded network. After Step 2, by averaging the flow over time $\tilde{f}$ over the time intervals $[i\Delta, (i+1)\Delta]$ in $\tilde{G}$, we get a corresponding static flow x in the Δ-condensed time-expanded network $\tilde{G}^{\tilde{T}}/\Delta$. Because $\tilde{f}$ is constructed from $\hat{f}$ which has been smoothened in Step 1, this static flow x achieves demand D within time $(1+\epsilon)^2 \cdot T$, and exceeds the capacity of edges by at most a factor of $(1+\epsilon)^2$.

Proposition 5. *The flow over time $\bar{f}$ constructed by Definition 3 from the static flow x in $\tilde{G}^{\tilde{T}}/\Delta$ achieves demand at least $D = (d_1, \ldots, d_k)$ with time horizon at most $(1+\epsilon)^2 \cdot T$, and the capacity constraint on each edge is violated by at most a factor of $(1+\epsilon)^2$.*

At this point, we have constructed a Δ-condensed time-expanded network $\tilde{G}^{\tilde{T}}/\Delta$ and have found a static flow x in it, such that the flow over time f_x constructed from x by Definition 3 achieves demand at least D with time horizon at most $(1+\epsilon)^2 T$, and the capacity constraint on each edge is violated by at most a factor of $(1+\epsilon)^2$.

Step 4: remove the capacity violations. The last step is to remove the capacity violations. To achieve this, we apply the following two procedures:

1. First if we keep the structure of the time-expanded network intact, but change the unit time interval length from Δ to $(1+\epsilon)^2\Delta$, it will correspond to a new dynamic network in which all transit times and the time horizon are increased by a factor of $(1+\epsilon)^2$. The static flow x still corresponds to a flow over time f_x in this new network with the supply and demand of each commodity also increased by a factor of $(1+\epsilon)^2$. That is, f_x has time horizon $(1+\epsilon)^4 T$ and achieves demand $(1+\epsilon)^2 D$. But the capacity constraint on each edge is still violated by at most a factor of $(1+\epsilon)^2$.

2. Next, we reduce the flow values of x on all edges by a factor of $(1+\epsilon)^2$. The resulting static flow x will achieve demand $\frac{(1+\epsilon)^2 D}{(1+\epsilon)^2} = D$, with all capacity constraints strictly satisfied. The timespan of f_x is still $(1+\epsilon)^4 T$.

The above four steps complete the proof of Lemma 1. □

With the help of Lemma 1, the proof of our main theorem becomes rather straightforward.

Proof (Proof of Theorem 2). This proof follows the same structure of the proof of Theorem 1. That is, we use binary search to find the smallest T, such that in the Δ-condensed time-expanded network $G'^{T'}/\Delta$ constructed from Lemma 1, there exists a feasible static flow, which in turn implies the desired flow over time in G.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B., Weihe, K.: Network flows: theory, algorithms and applications. ZOR-Methods and Models of Operations Research **41**(3), 252–254 (1995)
2. Aronson, J.E.: A survey of dynamic network flows. Annals of Operations Research **20**(1), 1–66 (1989)
3. Dhamala, T.N.: A survey on models and algorithms for discrete evacuation planning network problems. Journal of Industrial & Management Optimization **11**(1), 265–289 (2015)
4. Fleischer, L., Skutella, M.: Minimum cost flows over time without intermediate storage. In: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms. pp. 66–75. Society for Industrial and Applied Mathematics (2003)
5. Fleischer, L., Skutella, M.: Quickest flows over time. SIAM Journal on Computing **36**(6), 1600–1630 (2007)
6. Fleischer, L., Tardos, É.: Efficient continuous-time dynamic network flow algorithms. Operations Research Letters **23**(3-5), 71–80 (1998)
7. Ford, L.R., Fulkerson, D.R.: Constructing maximal dynamic flows from static flows. Operations Research **6**(3), 419–433 (1958)
8. Ford, L.R., Fulkerson, D.R.: Flows in networks. Princeton University Press (1962)
9. Hall, A., Hippler, S., Skutella, M.: Multicommodity flows over time: Efficient algorithms and complexity. Theoretical Computer Science **379**(3), 387–404 (2007)
10. Halpern, J.: A generalized dynamic flows problem. Networks **9**(2), 133–167 (1979)
11. Hoppe, B.: Efficient dynamic network flow algorithms. Tech. rep., Cornell University (1995)
12. Hoppe, B., Tardos, É.: The quickest transshipment problem. Mathematics of Operations Research **25**(1), 36–62 (2000)
13. Kotnyek, B.: An annotated overview of dynamic network flows. Ph.D. thesis, INRIA (2003)
14. Powell, W.B., Jaillet, P., Odoni, A.: Stochastic and dynamic networks and routing. Handbooks in operations research and management science **8**, 141–295 (1995)
15. Skutella, M.: An introduction to network flows over time. In: Research trends in combinatorial optimization, pp. 451–482. Springer (2009)