In this work, we develop a new framework for dynamic network flow problems based on optimal transport theory. We show that the dynamic multi-commodity minimum-cost network flow problem can be formulated as a multi-marginal optimal transport problem, where the cost function and the constraints on the marginals are associated with a graph structure. By exploiting these structures and building on recent advances in optimal transport theory, we develop an efficient method for such entropy-regularized optimal transport problems. In particular, the graph structure is utilized to efficiently compute the projections needed in the corresponding Sinkhorn iterations, and we arrive at a scheme that is both highly computationally efficient and easy to implement. To illustrate the performance of our algorithm, we compare it with a state-of-the-art Linear programming (LP) solver. We achieve good approximations to the solution at least one order of magnitude faster than the LP solver. Finally, we showcase the methodology on a traffic routing problem with a large number of commodities.

Key words: Multi-marginal optimal transport; Dynamic network flow; Multi-commodity network flow; Sinkhorn’s method; Computational methods; Traffic routing
MSC2000 subject classification: 49Q22, 90-08, 90C35, 90C08, 49M29
OR/MS subject classification: Networks/graphs: Flow algorithms; Networks/graphs: Multicommodity; Programming: Linear: Large scale systems

1. Introduction. Many phenomena in today’s society can be modelled as large scale transportation or flow problems, and new technological advances create the need for solving larger and larger problems. An example is the introduction of self driving-cars to the road network, which will create both new opportunities and new challenges [42, 48]. Increasing automation and communication between vehicles will result in very large systems where all vehicles need to be routed simultaneously taking into account destinations, vehicle properties and urgency [13]. Another challenge is to direct large crowds in, e.g., transit areas in airports, subways, or event venues [59, 33, 2], which is particularly critical for evacuation scenarios in the case of emergencies, but also essential for every-day use.

Many of these problems can be modelled as large scale dynamic network flow problems [10, 39, 2]. The most common strategy for handling such problems is to convert the dynamic flow problem to a static flow problem on a time-expanded network, and this strategy goes back to the classical work [24]. In addition to this, there are typically several classes of groups of agents with heterogeneous properties and objectives in the system. For instance, each agent in a traffic network drives a vehicle with certain properties, and the objective is typically to reach a certain destination with a certain degree of urgency. Similar problems appear in air traffic planning, railroad traffic scheduling, communication and logistics, and are often treated as multi-commodity flow problems.
over networks [33, 10, 39, 2]. Although such problems are usually formulated as linear programming (LP) problems, for real applications the corresponding optimization problems are often too large to be handled by standard methods. Specialized methods exploit the structure of multi-commodity flow problems, using, e.g., column generation methods. These include price-directive decomposition [36], resource-directive decomposition [38, 46], and basis partitioning methods [23]. However, it has been reported that these methods typically decrease the solution time of standard (LP) solvers by at most one order of magnitude [4, 52].

During the last few decades there has been considerable development in the field of optimal transport theory. Traditionally the optimal transport problem addresses a static scenario where one given distribution is transported to another, and this problem has been extensively used in areas such as economics and logistics [57]. There has recently been a rapid advancement of theory and applications for optimal transport, in particular towards applications in imaging, statistics and machine learning (see [51] and references therein), and systems and control [6, 15], which has led to a mature framework with computationally efficient algorithms [51] that can be used to address a wide range of problems. The optimal transport problem is a linear program, but the number of variables often makes it intractable to solve with general-purpose optimization methods for large size problems. However, a recent computational breakthrough in this area builds on introducing an entropic barrier term in the objective function. The resulting optimization problem can then be solved efficiently using the so called Sinkhorn iterations [20]. This allows for computing an approximate solution of large transportation problems and has opened up the field for new applications where no computationally feasible method previously existed.

The optimal transport framework has in some cases been used for modelling several kinds of interacting classes, e.g., for transport of multiple species [19, 3] or flows with several phases [7]. In this paper we will build on some of these results and we propose to use a generalization of the optimal transport problem with several marginals to address multi-commodity flow problems. This multi-marginal optimal transport problem [27, 49, 53, 54] is computationally challenging since the number of variables grows exponentially in the number of marginals. Even though entropy regularization methods have been derived for the multi-marginal optimal transport problem [8], the cost for each iteration still grows exponentially in the number of marginals (see [44] for computational complexity bounds). However, in many cases the cost function has a structure that can be utilized for efficient computations, as for example in barycenter, information fusion, and tracking problems [8, 22, 31].

In this paper we show that the dynamic flow problem can be formulated as a structured multi-marginal optimal transport problem. This structure can be visualized in a graph where the set of nodes corresponds to the marginals, and where there is an edge between two nodes if there is a cost term or a constraint that depends jointly on the two nodes. For the single commodity case, this structure is a path graph with one node for each time point that represents the flow in the network at that time. For the dynamic multi-commodity network flow problem, there is one additional node in the graph that represents the distribution over the different commodity classes. The solution to this optimal transport problem then describes a joint distribution, which consists of the optimal flow for all commodities in the dynamic network problem.

We consider the corresponding entropy-regularized approximation of this problem, and by utilizing the structure in the cost function we derive methods for solving this problem. Many of the classical methods for dynamic flow problems consider standard network flow methods on the time-expanded network. By instead formulating this problem as a multi-marginal optimal transport problem, we can more efficiently utilize the sequential structure without explicitly setting up the time-expanded network. This results in an elegant and easily implementable method. We illustrate experimentally that this method is computationally competitive with state-of-the-art methods, and then apply it to a traffic routing problem.
The rest of the paper is structured as follows. Section 2 summarizes background material on dynamic multi-commodity network flows and multi-marginal optimal transport. In Section 3 we explain how to formulate network flow problems as structured multi-marginal optimal transport problems. Based on this, we develop numerical schemes to solve the problems in Section 4. Finally, in Section 5 we compare the performance of our methods to a commercial LP solver, and showcase it in a traffic routing application.

2. Background. In this section we review background on the two central topics of this paper: dynamic multi-commodity network flows and multi-marginal optimal transport. We also use this Section to set up notation. In particular, bold-faced letters are used throughout to denote tensors, and $\otimes$ denotes the tensor (outer) product, e.g., for vectors $v_1 \in \mathbb{R}^{n_1}$ and $v_2 \in \mathbb{R}^{n_2}$ we have that $v_1 \otimes v_2 \in \mathbb{R}^{n_1 \times n_2}$ and $(v_1 \otimes v_2)_{ij} = (v_1)_j (v_2)_i$. Moreover, by 1 we denote a column vector of ones of appropriate size, by $\mathbb{R}_+$ we denote the nonnegative real numbers, and we use $\overline{\mathbb{R}}_+ = \mathbb{R}_+ \cup \{\infty\}$ and $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty\} \cup \{-\infty\}$ to denote the extended nonnegative real line and extended real line, respectively. Throughout we will adopt the convention that $0 \cdot \infty = 0$. Finally, by exp$(\cdot)$, log$(\cdot)$, $\odot$, $\mathbin{/}$, and $\min(\cdot, \cdot)$ we denote elementwise exponential, logarithm, product, division, and minimum respectively.

2.1. Minimum-cost network flow problems. A minimum-cost network flow problem is to determine a flow from sources to sinks with minimum cost \cite{26, 9}. More specifically, the flow is defined on a network $\mathcal{N} = (\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V}$ and directed edges $\mathcal{E}$, and the sources and sinks are sets of edges\(^1\) $\mathcal{S}^+ \subset \mathcal{E}$ and $\mathcal{S}^- \subset \mathcal{E}$. Let each source $e \in \mathcal{S}^+$ be equipped with a supply $r_e^+ \in \mathbb{R}_+$, and each sink $e \in \mathcal{S}^-$ with a demand $r_e^- \in \mathbb{R}_+$, and we assume that the total supply matches the total demand, i.e., that $\sum_{e \in \mathcal{S}^+} r_e^+ - \sum_{e \in \mathcal{S}^-} r_e^- = 0$. In addition, let each edge $e \in \mathcal{E}$ be assigned a cost $c_e \in \mathbb{R}_+$ of transporting a unit of flow on that edge. The goal of minimum cost-flow problems is to transport the flow from the sources to the sinks with minimal total transporting cost. We also include capacity constraints, which require that the total flow on an edge is limited by the edge capacity $d_e \in \mathbb{R}_+$ on $e \in \mathcal{E}$.

There are two standard formulations for the network flow problem. One is the arc-chain formulation, where one optimizes over a set of flow paths (arc-chains) from sources to sinks \cite{26, 55}. This is the main formulation considered in this work and is described in detail below. Another common formulation is the node-edge formulation, where one seeks the optimal amount of flow over each edge while maintaining flow balance in each node. For more details on this formulation, and a comparison of both formulations we refer the reader to \cite{26, 55}.

2.1.1. The arc-chain formulation. Given a network $\mathcal{N} = (\mathcal{V}, \mathcal{E})$, a path is a sequence of edges that joins two vertices such that all edges and all visited vertices are distinct, i.e., they occur at most once in the sequence \cite[p. 6]{21}. A path is thus a subgraph, which we denote by $p$, and is defined by a list of edges $(p_1, p_2, \ldots, p_N)$, where $p_j \in \mathcal{E}$ denotes the $j$-th element of the path for $j = 1, \ldots, N$. Here, $N$ is called the length of the path $p$. Moreover, since $p$ is a path the edge $p_j$ ends in the initial node of $p_{j+1}$ for $j = 1, \ldots, N-1$.

In the arc-chain formulation, we consider the paths, or arc-chains, which start in a source and end in a sink. Let $\mathcal{P}$ denote the set of all such paths, where the first element lies in $\mathcal{S}^+$, and its last element lies in $\mathcal{S}^-$. Moreover, let $\mathcal{P}^+_e$ denote the paths starting from the edge $e \in \mathcal{S}^+$, and let $\mathcal{P}^-_e$ denote the paths ending in the edge $e \in \mathcal{S}^-$. The cost of a path $p \in \mathcal{P}$ is the sum of the costs of its

\(^1\) Often the sources and sinks are defined on the nodes $\mathcal{V}$ not the edges $\mathcal{E}$. In this work we consider the latter case, however the framework introduced herein can easily be modified to define the sources and sinks on the nodes $\mathcal{V}$ instead.
edges $c_p = \sum_{e \in p} c_e$. Next, let $x_p$ denote the amount of flow associated with path $p \in \mathcal{P}$. Then, the arc-chain formulation of the minimum-cost network flow problem reads

$$\begin{align*}
\min_{x_p \in \mathbb{R}_+, \forall p \in \mathcal{P}} & \sum_{p \in \mathcal{P}} c_p x_p \\
\text{subject to} & \sum_{p \in \mathcal{P}_e^+} x_p = r_e^+, \quad \forall e \in \mathcal{S}^+, \\
& \sum_{p \in \mathcal{P}_e^-} x_p = r_e^-, \quad \forall e \in \mathcal{S}^-, \\
& \sum_{p \in \mathcal{P}} \delta_{e \in p} x_p \leq d_e, \quad \forall e \in \mathcal{E},
\end{align*}$$

where $\delta_{e \in p} = 1$ if the edge $e$ is part of path $p$, and $\delta_{e \in p} = 0$ otherwise. Here, the objective function corresponds to the total cost of the flow. The first two sets of constraints guarantee that the supply and demand for all sources and sinks are satisfied, and the last set of constraints enforces that the flow on each edge does not exceed the given capacity.

2.1.2. Multi-commodity network flow. The extension to multi-commodity network flow problems deals with the case where there are multiple commodities present in the network [34, 58, 55, 25, 39]. Here we let $L$ denote the number of commodities, and let $c_{\ell}^e$ denote the cost of a unit flow on edge $e \in \mathcal{E}$ of commodity $\ell$, for $\ell = 1, \ldots, L$. The supply and demand typically depend on the commodity, thus each commodity $\ell$ has specified sources $\mathcal{S}^{\ell,+} \in \mathcal{E}$ with supplies $r_\ell^e$ for $e \in \mathcal{S}^{\ell,+}$, and sinks $\mathcal{S}^{\ell,-} \in \mathcal{E}$ with demands $r_\ell^e$ for $e \in \mathcal{S}^{\ell,-}$. Moreover, for each commodity $\ell = 1, \ldots, L$, let $\mathcal{P}_\ell^e$ denote the sets of paths from the sources to the sinks, and let $\mathcal{P}_\ell^{e,+}$ denote the paths starting in $e \in \mathcal{S}^{\ell,+}$, and let $\mathcal{P}_\ell^{e,-}$ denote the paths ending in $e \in \mathcal{S}^{\ell,-}$. The cost of a unit flow of commodity $\ell$ on a path $p \in \mathcal{P}$ is the sum of the corresponding costs of the edges in the path $c_p^\ell = \sum_{e \in p} c_{\ell}^e$. Next, by letting $x_p^\ell$ denote the amount of flow of commodity $\ell$ on path $p$, the minimum cost multi-commodity network flow problem in arc-chain formulation reads

$$\begin{align*}
\min_{x_p^\ell \in \mathbb{R}_+, \forall p \in \mathcal{P}^\ell, \forall \ell = 1, \ldots, L} & \sum_{\ell = 1}^L \sum_{p \in \mathcal{P}^\ell} c_p^\ell x_p^\ell \\
\text{subject to} & \sum_{p \in \mathcal{P}_\ell^{e,+}} x_p = r_\ell^e, \quad \forall e \in \mathcal{S}^{\ell,+}, \quad \ell = 1, \ldots, L, \\
& \sum_{p \in \mathcal{P}_\ell^{e,-}} x_p = r_\ell^e, \quad \forall e \in \mathcal{S}^{\ell,-}, \quad \ell = 1, \ldots, L, \\
& \sum_{\ell = 1}^L \sum_{p \in \mathcal{P}^\ell} \delta_{e \in p} x_p^\ell \leq d_e, \quad \forall e \in \mathcal{E}.
\end{align*}$$

Here, the first two sets of constraints guarantee that the demand and supply for all commodities are satisfied. The third set of constraints enforces that the flow on each edge does not exceed the given capacity. In particular, note that the multi-commodity problem (2) with only one commodity, i.e., $L = 1$, boils down to the single-commodity problem (1).

2.1.3. Dynamic network flow. In this work we consider dynamic flows, also called flows over time, where the time that it takes for the flow to travel in the network is taken into account [24, 2, 34]. In this work we develop efficient methods that exploit the temporal structure. For this to work we need to assume synchronous travelling times for all edges, but on the other hand the efficient methods allows for handling problems with large networks and fine time discretization.
More precisely, we consider a flow problem on the network \( N = (V, E) \) over the time interval \( 0 \) to \( T \). The problem is to transport a given flow at time 0 through the network to a final flow at time \( T \) with minimal cost, while satisfying capacity constraints at all time points. We consider the discretized problem on the time steps \( 0, 1, \ldots, T \). Dynamic flow problems are typically solved as a static problem on the time-expanded network \([24]\). The time-expanded network \( N_{\text{exp}} \) is constructed by considering \( T + 1 \) copies of the vertices \( V \), denoted by \( V_0, \ldots, V_T \). Here the copy \( V_t \) is associated with time instance \( t \) in the time expanded network, and we denote these nodes by \((t, v)\) where \( v \in V \) in the original network.

The edges of \( N_{\text{exp}} \) connect nodes corresponding to consecutive time instances according to the edges \( E \) in the original network, that is, \( E_{\text{exp}} = \bigcup_{t=1}^{T} E_t \) where \( E_t \) consists of the directed edges \(((t-1, v_{t-1}), (t, v_t))\) where \((v_{t-1}, v_t) \in E\), for \( t = 1, \ldots, T \). The capacities and costs on these added edges are defined to be the same as the corresponding edges in the original network \( N \). The time-expanded network is illustrated for a simple example in Figure 1.

To express the dynamic flow problem in arc-chain formulation similarly to \((1)\) and \((2)\), a path \( p \) is as before a tuple of edges \((p_1, \ldots, p_T)\). Its element \( p_t \in E_t \) denotes the edge, which the path takes in the time interval \([t-1, t]\). In the setting of one commodity, let \( P \) denote the set of feasible paths in the time-expanded network \( N_{\text{exp}} \), i.e., \( p \in P \) if \( p \) is a path that starts in a source, \( p_1 \in S^+ \), and ends in a sink, \( p_T \in S^- \). The corresponding cost of unit flow on the path \( p \in P \) is then \( c_p = \sum_{t=1}^{T} c_{p_t} \). The dynamic minimum-cost network flow problem can then be written as

\[
\begin{align*}
\text{minimize} & \quad \sum_{p \in P} c_p x_p \\
\text{subject to} & \quad \sum_{p \in P} \delta_{e=p_1} x_p = r_{e^+}, \quad \text{for } e \in S^+, \\
& \quad \sum_{p \in P} \delta_{e=p_T} x_p = r_{e^-}, \quad \text{for } e \in S^-, \\
& \quad \sum_{p \in P} \delta_{e=p_t} x_p \leq d_e, \quad \text{for } e \in E, \quad t = 2, \ldots, T-1.
\end{align*}
\]

Note that the network flow problem \((1)\) on the time-expanded network \( N_{\text{exp}} \), corresponds to \((3)\) line by line.

To formulate the multi-commodity counterpart of the dynamic flow problem \((3)\), let \( P' \) denote the set of feasible paths in the time-expanded network \( N_{\text{exp}} \) for commodity \( \ell = 1, \ldots, L \). The

---

\[2\] Note that there is a canonical bijection \((v_1, v_2) \leftrightarrow ((t-1, v_1), (t, v_2))\) between the edges \( E \) and the edges \( E_t \).
corresponding cost of unit flow on the path for commodity \( \ell \) is then \( c_{p}^{\ell} = \sum_{t=1}^{T} c_{t}^{\ell} \) for a path \( p \in \mathcal{P}^{\ell} \), and the dynamic minimum-cost multi-commodity network flow problem reads

\[
\begin{align*}
\text{minimize} \quad & \sum_{\ell=1}^{L} \sum_{p \in \mathcal{P}^{\ell}} c_{p}^{\ell} x_{p}^{\ell} \\
\text{subject to} \quad & \sum_{p \in \mathcal{P}^{\ell}} \delta_{e=p^{\ell}} x_{p}^{\ell} = r_{e}^{\ell, +}, \quad \text{for } e \in \mathcal{S}^{\ell, +}, \quad \ell = 1, \ldots, L, \\
& \sum_{p \in \mathcal{P}^{\ell}} \delta_{e=p^{\ell}} x_{p}^{\ell} = r_{e}^{\ell, -}, \quad \text{for } e \in \mathcal{S}^{\ell, -}, \quad \ell = 1, \ldots, L, \\
& \sum_{\ell=1}^{L} \sum_{p \in \mathcal{P}^{\ell}} \delta_{e=p^{\ell}} x_{p}^{\ell} \leq d_{e}, \quad \text{for } e \in \mathcal{E}, \quad t = 2, \ldots, T - 1;
\end{align*}
\]

see \cite{40} for a similar problem formulation.

A problem with the arc-chain formulations is that the number of variables, corresponding to possible paths, grows exponentially with \( T \). Thus, standard linear programming methods are not applicable when \( T \) is large. A way to circumvent this issue is to use specialized solvers building on, e.g., column generation, or to instead consider the corresponding node-edge formulations of the problem (cf. \cite{26, 55}). In this work we take a different approach that builds on formulating the problem as an optimal transport problem that utilize the structure in the arc-chain formulation.

2.2. Optimal transport. The optimal transport problem is to find a mapping that moves the mass from one distribution to another with minimal cost, based on an underlying metric \cite{57}. In this paper we consider the discrete setting, where the two distributions are represented by two non-negative vectors \( \mu_{1} \in \mathbb{R}_{+}^{n_{1}}, \mu_{2} \in \mathbb{R}_{+}^{n_{2}} \) with equal mass. In this setting the transport cost is defined in terms of a underlying non-negative cost matrix \( C \in \mathbb{R}_{+}^{n_{1} \times n_{2}} \), where \( C_{ij} \) denotes the cost\(^3\) of moving a unit mass from position \( i \) to \( j \). Analogously, a transport plan \( M \in \mathbb{R}_{+}^{n_{1} \times n_{2}} \) is a non-negative matrix, where \( M_{ij} \) represents the amount of mass moved from \( i \) to \( j \). The optimal transport plan from \( \mu_{1} \) to \( \mu_{2} \) is then a minimizing solution of

\[
\begin{align*}
\text{minimize} \quad & \text{trace}(C^{T} M) \\
\text{subject to} \quad & M1 = \mu_{1}, \\
& MT1 = \mu_{2}.
\end{align*}
\]

Multi-marginal optimal transport extends the concept of the classical optimal transport problem (5) to the setting with a set of marginals \( \mu_{t} \in \mathbb{R}_{+}^{n_{t}}, \) for \( t = 1, \ldots, T \), where \( T \geq 2 \) \cite{49, 8, 22, 31}. In this setting, the transport cost and transport plan are described by tensors \( C \in \mathbb{R}_{+}^{n_{1} \times n_{2} \cdots \times n_{T}} \) and \( M \in \mathbb{R}_{+}^{n_{1} \times n_{2} \cdots \times n_{T}} \). Here, \( C_{i_{1}\ldots i_{T}} \) denotes the unit cost associated with the tuple \((i_{1}, \ldots, i_{T})\), and \( M_{i_{1}\ldots i_{T}} \) denotes the amount of mass associated with this tuple. Then the total transportation cost for a given transport plan \( M \) is

\[
(C, M) = \sum_{i_{1}, \ldots, i_{T}} C_{i_{1}\ldots i_{T}} M_{i_{1}\ldots i_{T}}.
\]

Moreover, \( M \) is a transport plan between the desired marginals if its projections on the marginals satisfy \( P_{t}(M) = \mu_{t} \), for \( t = 1, \ldots, T \), where the projection on the \( t \)-th marginal is defined by

\[
(P_{t}(M))_{i_{t}} := \sum_{i_{1}, \ldots, i_{t-1}, i_{t+1}, \ldots i_{T}} M_{i_{1}\ldots i_{t-1} i_{t} i_{t+1}\ldots i_{T}}.
\]

\(^3\) If transport of mass is not allowed from position \( i \) to position \( j \), then we let \( C_{ij} = \infty \).
The discrete multi-marginal optimal transport problem thus reads

\[
\begin{align*}
\text{minimize} & \quad \langle C, M \rangle \\
\text{subject to} & \quad P_t(M) = \mu_t, \text{ for } t \in \Gamma.
\end{align*}
\]

(7)

Here \(\Gamma\) is an index set that describes the set of constrained marginals. In the original multi-marginal optimal transport formulation, constraints are typically given on all marginals, i.e., for the index set \(\Gamma = \{1, 2, \ldots, T\}\). However, in this work we typically consider the case where constraints are only imposed on a subset of marginals, i.e., \(\Gamma \subset \{1, 2, \ldots, T\}\), or when some of the constraints are inequality constraints.

Note that the standard bi-marginal optimal transport problem (5) is a special case of the multi-marginal optimal transport problem (7), where \(T = 2\) and \(\Gamma = \{1, 2\}\). It is also worth noting that the bi-marginal optimal transport problem can be interpreted as a minimum-cost network flow problem. However, this interpretation does in general not extend to the multi-marginal case [43]. In this work we show how to formulate any dynamic network flow problem as a multi-marginal optimal transport problem with a structured cost tensor.

2.2.1. Sinkhorn iterations. Although linear, the number of variables in the multi-marginal optimal transport problem (7) is often too large to be solved directly. A popular approach for the bi-marginal setting to bypass the size of the problem has been to add a regularizing entropy term to the objective [20]. In principle, the same approach can be used also for the multi-marginal case. With the entropy term

\[
D(M) = \sum_{i_1, \ldots, i_T} (M_{i_1 \ldots i_T} \log(M_{i_1 \ldots i_T}) + M_{i_1 \ldots i_T} - 1),
\]

(8)

the entropy regularized multi-marginal optimal transport problem is defined as

\[
\begin{align*}
\text{minimize} & \quad \langle C, M \rangle + \epsilon D(M) \\
\text{subject to} & \quad P_t(M) = \mu_t, \text{ for } t \in \Gamma,
\end{align*}
\]

(9)

where \(\epsilon > 0\) is a small regularization parameter. The introduction of the entropy term in problem (9) allows for expressing the optimal solution \(M\) in terms of Lagrange dual variables, which may be computed by Sinkhorn iterations [8, 47]. In particular, it can be shown that the optimal solution of (9) is of the form [22]

\[
M = K \odot U,
\]

(10)

where \(K = \exp(-C/\epsilon)\) and where \(U\) can be decomposed as

\[
U = u_1 \otimes u_2 \otimes \cdots \otimes u_T.
\]

(11)

Here, the vectors \(u_t \in \mathbb{R}^{n_t}_+\), for \(t = 1, 2, \ldots, T\), are given by

\[
\begin{align*}
u_t &= \begin{cases} \exp(\lambda_t/\epsilon), & \text{if } t \in \Gamma \\ 1, & \text{else}, \end{cases}
\end{align*}
\]

(12)

where \(\lambda_t \in \mathbb{R}^{n_t}\) for \(t \in \Gamma\) are optimal dual variables in the dual problem of (9). This dual problem takes the form

\[
\begin{align*}
\maximize_{\lambda_t \in \mathbb{R}^{n_t}, t \in \Gamma} -\epsilon \langle K, U \rangle + \sum_{t \in \Gamma} \lambda_t^T \mu_t,
\end{align*}
\]

(13)
where $U$ depends on $\{\lambda_t\}_{t \in \Gamma}$ as specified in (11) and (12). For details the reader is referred to, e.g., [22, 8].

The Sinkhorn scheme for finding $U$ in (11) is to iteratively update $u_t$ according to

$$u_t \leftarrow u_t \odot \mu_t / P_t(K \odot U),$$

for all $t \in \Gamma$. This scheme may for instance be derived as Bregman projections [8] or a block coordinate ascend in the dual (13), [37, 22, 56]. As a result, global convergence of the Sinkhorn scheme (14) is guaranteed [5, 56, 45]. The computational bottleneck of the Sinkhorn iterations (14) is computing the projections $P_t(K \odot U)$, for $t \in \Gamma$, which in general scales exponentially in $T$. In fact, even storing the tensor $M$ is a challenge as it consists of $\prod_{t=1}^{T} n_t$ elements. However, in many cases of interest, structures in the cost tensors can be exploited to perform the sum operations in (6) in an appropriate order, which makes the computation of the projections feasible [22, 8, 31, 29, 22]. More precisely, in many applications the tensor $K \odot U$ factorizes such that it can be described by a graph $G = (V, E)$, where the vertices $V$ correspond to the tensor marginals and its dependencies are described by the set of edges $E$. The projections (6) can then be computed efficiently by first eliminating the variables, i.e., performing the sum operations, for the vertices that have few dependencies. For instance, when the tensor $K \odot U$ factorizes according to a tree structure, the projections (6) can be computed by first eliminating the variables corresponding to the trees leafs and successively moving down the branches. Computing the projections requires then only matrix-vector multiplications, where the matrices are at most of size $\max_i(n_i)$ [31, 22]. In the case of more complex graphs a similar approach can be utilized, but computations become more expensive. For instance, in case the graph is a cycle the complexity is increased by a factor of $\max_i(n_i)$ as compared to the tree setting [8, 29].

3. Network flow problems via optimal transport. In this section we introduce a reformulation of the dynamic minimum-cost flow problem as a multi-marginal optimal transport problem (7). In the single-commodity case this optimal transport problem has a path-structure. The multi-commodity case can be expressed as several single-commodity problems, which are coupled through the capacity constraints. Alternatively, this can be set up as one multi-marginal optimal transport problem, where the cost function decouples as a graph that contains cycles.

3.1. The dynamic minimum-cost flow problem. Let $N_{\text{exp}}$ be the time-expansion of the network $N$ for the time steps $t = 0, \ldots, T$, and let $\mathcal{P}$ denote the set of feasible paths in $N_{\text{exp}}$. In order to solve an arc-chain formulation of a flow-problem on this network, one has to identify all paths in this set. Clearly, the set of feasible paths $\mathcal{P}$ is a subset of the set $\bar{\mathcal{P}} = \{(i_1, \ldots, i_T) : i_t \in \mathcal{E}_t \text{ for } t = 1, \ldots, T\}$, which contains all combinations of $T$ edges in $\mathcal{E}$. In fact, the set $\bar{\mathcal{P}}$ is generally much larger than $\mathcal{P}$, since it lifts the set of feasible paths to the set of all “paths” possible from purely combinatorial considerations (ignoring the graph structure).

However, using this representation, the network flow can be described by a tensor $M \in \mathbb{R}^{n_T}_+$, where $n = |\mathcal{E}|$, and where the element $M_{i_1, \ldots, i_T}$ denotes the amount of flow on the path $(i_1, \ldots, i_T)$. The vector $P_t(M) \in \mathbb{R}^{n_i}_+$, where the projection operator is defined as in (6), then describes the flow distribution over the edges between time $t - 1$ and $t$, as illustrated in Figure 2. That is, its element $P_t(M)_i$ denotes the amount of flow over edge $i \in \mathcal{E}_t$.

Similarly, the evolution of flow between time intervals $(t_1 - 1, t_1)$ and $(t_2 - 1, t_2)$ is described by the bi-marginal projections $P_{t_1, t_2}(M) \in \mathbb{R}^{n \times n}_+$, which are defined as

$$P_{t_1, t_2}(M)_{i_1, i_2} = \sum_{i_1, \ldots, i_T \setminus \{i_1, i_2\}} M_{i_1, \ldots, i_T}.$$  

(15)
That is, the element \((P_{t_1,t_2}(M))_{ij}\) describes the amount of flow that is in edge \(i\) at time \(t_1\) and that is in edge \(j\) at time \(t_2\). Let \(c \in \mathbb{R}_+^n\), where \(c_i\) denotes the cost of a unit flow on edge \(i \in \mathcal{E}\), and let \(C \in \mathbb{R}_{+}^{n \times n}\) encode the network topology, i.e., \(C_{ij} = 0\) if edge \(i\) leads to\(^4\) edge \(j\), and \(C_{ij} = \infty\) otherwise. Then we define the cost of a transport plan \(M\) as

\[
\sum_{t=1}^{T} c^T P_t(M) + \sum_{t=1}^{T-1} \text{trace}(C^T P_{t,t+1}(M)) = \langle C, M \rangle,
\]

where the tensor \(C \in \mathbb{R}_{+}^{n \times n \times \ldots \times n}\) is defined as

\[
C_{i_1 \ldots i_T} = \sum_{t=1}^{T} c_{i_t} + \sum_{t=1}^{T-1} C_{i_t i_{t+1}}.
\]

Note that \(\langle C, M \rangle = \infty\) here means that the transport plan contains paths that are not consistent with the network structure, i.e., that for some \(t \in \{1, \ldots, T-1\}\) and some \((i, j) \notin \mathcal{E}\), \((P_{t,t+1}(M))_{ij} > 0\). The structure of the cost function (16) can be illustrated by the path-graph in Figure 3.

Let \(\mu_1\) and \(\mu_T\) be the supply and demand distributions, respectively. That is, \((\mu_1)_i = r^+_i\), for \(i \in S^+\), and 0 otherwise, and \((\mu_T)_i = r^-_i\) for \(i \in S^-\), and 0 otherwise. Moreover, let \(d \in \mathbb{R}_+^n\) encode the capacity constraints of the network, that is \(d_i\) is the flow capacity on edge \(i \in \mathcal{E}\). These supply, demand, and capacity constraints can be encoded as equality and inequality constraints on the flow distributions over the edges \(P_t(M)\). Based on this, we formulate the linear program

\[
\begin{align*}
\min_{M \in \mathbb{R}^{nT}_+} & \quad \langle C, M \rangle \\
\text{subject to} & \quad P_1(M) = \mu_1.
\end{align*}
\]

\(^4\)That is, the second vertex of edge \(i\) is the first vertex of edge \(j\) in the network \(\mathcal{N}\).
This problem is equivalent to the dynamic minimum-cost network flow problem (3) in the sense described in the following theorem.

**Theorem 1.** The dynamic minimum-cost network flow problem (3) and problem (18) correspond to each other in the following sense.

1. Assume that (18) has a feasible solution with finite objective value. Then it has a finite optimal value, and (3) has the same optimal value. Moreover, if \( M \) is an optimal solution of (18), then there is an optimal solution \( \{x_p : p \in P\} \) of (3) such that

\[
M_{i_1...i_T} = \begin{cases} 
  x_p & \text{for } (i_1, \ldots, i_T) \in P, \text{ where } p = (i_1, \ldots, i_T) \\
  0 & \text{for } (i_1, \ldots, i_T) \in \hat{P} \setminus P. 
\end{cases} 
\]  

(19)

2. Assume that there is a feasible solution to (3). Then it has a finite optimal value, and problem (18) has the same optimal value. Moreover, if \( \{x_p : p \in P\} \) is an optimal solution of (3), then there is an optimal solution \( M \) of (18) such that (19) holds.

**Proof:** First, note that the amount of flow on edge \( e \in E \) between time \( t - 1 \) and \( t \) is given in the optimal transport formulation (18) by

\[
\sum_{i \in P, i_{t-1} = e} M_{i_1...i_T} = P_t(M)_e. 
\]  

(20)

Thus the flow distribution over \( E_t \) is exactly the projection \( P_t(M) \) as defined in (6). Then, with

\[
(\mu_1)_i = \begin{cases} 
  r^+_i & i \in S^+ \\
  0 & \text{otherwise}, 
\end{cases} \quad
(\mu_T)_i = \begin{cases} 
  r^-_i & i \in S^- \\
  0 & \text{otherwise}, 
\end{cases} 
\]

the set of constraints (3a)-(3b) and (18b)-(18c) both restrict the respective problems to paths that satisfy the supply and demand constraints. In the formulation (3) the total flow on edge \( e \in E_t \) is given by

\[
\sum_{p \in P} \delta_{e=p_t} x_p, 
\]  

(21)

and thus the inequality constraints (3c) and (18d) restrict the flows in the respective problems to the same capacity constraints. Moreover, note that \( (P_{t+1}(M))_{ij} \) describes the amount of flow moving from edge \( i \in E_t \) to edge \( j \in E_{t+1} \). Therefore, the objective (18a) is finite if and only if \( M_{i_1...i_T} = 0 \) for all \( (i_1, \ldots, i_T) \in \hat{P} \setminus P \). Now, by associating the amount of flow on edge \( i \in E_t \) with (20) and (21), respectively, the cost of a feasible flow plan, i.e., a plan that satisfies \( M_{i_1...i_T} = 0 \) for all \( (i_1, \ldots, i_T) \in \hat{P} \setminus P \), can be written in the two formulations as

\[
\sum_{p \in P} c_p x_p = \sum_{p \in P} \left( \sum_{t=1}^T \sum_{e \in E} \delta_{e=p_t} c_e \right) x_p = \sum_{t=1}^T \sum_{e \in E} \left( \sum_{p \in P} \delta_{e=p_t} x_p \right) c_e = \sum_{t=1}^T \sum_{e \in E} P_t(M)_e c_e = \sum_{t=1}^T c^T P_t(M). 
\]

This completes the proof. \( \square \)

Comparing problem (18) to problem (3), we have expanded the set of optimization variables by adding a large number of infeasible paths. However, the novel formulation (18) is structured as a multi-marginal optimal transport problem as in (7), which opens up for efficiently computing an approximate solution. In particular, the structure of problem (18) can be described by the path graph in Figure 3. Although problem (18) lifts the set of optimization variables in (3) from the set of feasible paths to the set of all combinatorially possible paths in the network, the infinite values in the tensor (17) restrict the problem to the set of feasible paths as in (3).
Remark 1. The second term in (16) is needed only to restrict the solution of problem (18) to the set of feasible paths $P$. Naturally this could instead be imposed as a set of hard constraints $P_{t,t+1}(M) \leq E$, for $t = 1, \ldots, T - 1$, where $E_{ij} = \infty$ if edge $i$ leads to edge $j$, and $E_{ij} = 0$ otherwise. Instead, we choose to use the penalty terms in (16) for computational reasons. In Section 4 we develop a scheme, which is based on the methods introduced in Section 2.2, i.e., solving the dual of a regularization of problem (18). Note that adding more hard constraints to (18) leads to a larger number of dual variables, which makes it more expensive to solve the regularized dual problem. We thus impose the network structure through the penalty terms in (16), which yields a dual problem with considerably fewer variables. Moreover, infinite values in $C$ induce sparsity to the tensor $K = \exp(-C/\epsilon)$ in (10), which can be exploited when computing the projections (6) needed for the Sinkhorn scheme.

3.2. The dynamic multi-commodity minimum-cost flow problem. In this section we extend the optimal transport formulation of the dynamic minimum-cost network flow problem from Section 3.1 to the multi-commodity setting.

Assume that there are $L$ different commodities present in the network $\mathcal{N}$, and each of these is assigned an initial distribution $\mu_1^T$, and a final distribution $\mu_T^T$, for $\ell = 1, \ldots, L$. For each commodity we define a cost vector $c_\ell \in \mathbb{R}^n$, where $(c_\ell)_i$ denotes the cost of a unit flow of commodity $\ell$ on edge $i \in \mathcal{E}$. As in the single-commodity case in Section 3.1, the network structure is imposed by a matrix $C \in \mathbb{R}_{+}^{n \times n}$, and the total flow capacity is bounded on all edges, and described by a vector $d \in \mathbb{R}^n$. One way to formulate an optimal transport problem for the multi-commodity flow is to describe each commodities flow by a mass transport tensor $M^\ell$, for $\ell = 1, \ldots, L$. Then each of these transport tensors has to satisfy the respective supply and demand constraints (18b)-(18c), and its cost is given by $\langle C^\ell, M^\ell \rangle$ as defined in (16). The capacity constraints in the network need to hold for the sum of all commodity flows, i.e., the sum of the projections $P_t(M^\ell)$ over all commodities $\ell = 1, \ldots, L$. The dynamic multi-commodity minimum-cost flow problem (4) can therefore be written as

\[
\begin{align*}
\text{minimize} & \quad \sum_{\ell=1}^L \langle C^\ell, M^\ell \rangle \\
\text{subject to} & \quad P_t(M^\ell) = \mu_1^\ell, \quad \text{for } \ell = 1, \ldots, L, \\
& \quad P_T(M^\ell) = \mu_T^\ell, \quad \text{for } \ell = 1, \ldots, L, \\
& \quad \sum_{\ell=1}^L P_t(M^\ell) \leq d, \quad \text{for } t = 2, \ldots, T - 1.
\end{align*}
\]

(22a)

(22b)

(22c)

(22d)

Note here that the $L$ optimal transport problems are each of the form in (18), and are coupled only through the capacity constraint (22d).

We will now bring problem (22) on a form similar to a multi-marginal optimal transport problem (7), i.e., a formulation containing only one mass transport tensor. This is done by combining all information from the $T$-mode transport plans $M^\ell \in \mathbb{R}^{n \times \cdots \times n}$, for $\ell = 1, \ldots, L$, to a new mass transport tensor $M \in \mathbb{R}^{L \times n \times \cdots \times n}$ with $T + 1$ modes. That is, we let its element $M_{i_1 \ldots i_T}$ describe the amount of flow of commodity $\ell$ over the path $i_1, \ldots, i_T$. Accordingly, for the added mode in the tensor we introduce a marginal $\mu_0 \in \mathbb{R}_+^L$, where $(\mu_0)_\ell = 1^T \mu_1^\ell = 1^T \mu_T^\ell$ denotes the total supply and demand of commodity $\ell \in L$. The initial and final distributions for the commodities can then be summarized in two matrices $R^{(0,1)} \in \mathbb{R}^{L \times n}$, defined as $R^{(0,1)} = (\mu_1^1, \mu_1^2, \ldots, \mu_1^L)^T$ and $R^{(0,T)} = (\mu_T^1, \mu_T^2, \ldots, \mu_T^L)^T$. In particular, with this construction it holds that $R^{(0,1)} 1 = R^{(0,T)} 1 = \mu_0$. Moreover, define a matrix $C_L \in \mathbb{R}^{L \times n}$ as $C_L = (c_1, c_2, \ldots, c_L)^T$, that is $(C_L)_{i,\ell}$ denotes the cost for commodity $\ell \in L$ to be on edge $i \in \mathcal{E}$. This setup is illustrated in Figure 4.
Note that the objective function (22a) can be written as
\[
\sum_{t=2}^{T-1} \operatorname{trace}(C_L^T P_{0,t}(M)) + \sum_{t=1}^{T-1} \operatorname{trace}(C^T P_{t,t+1}(M)) = \langle C, M \rangle,
\]
where the cost tensor \( C \in \mathbb{R}_+^{L \times n_T} \) is given by
\[
C_{i_0...i_T} = \sum_{t=2}^{T-1} (C_L)_{i_0i_t} + \sum_{t=1}^{T-1} C_{i_ti_{t+1}}.
\]
Thus, the dynamic multi-commodity minimum-cost network flow problem (22) can be expressed as
\[
\begin{align*}
\text{minimize} & \quad \langle C, M \rangle \\
\text{subject to} & \quad P_{0,1}(M) = R_{0,1} \\
& \quad P_{0,T}(M) = R_{0,T} \\
& \quad P_t(M) \leq d_t, \quad \text{for} \quad t = 2, \ldots, T-1
\end{align*}
\]
(24)

Utilizing the result in Theorem 1 we have now proved that the solution to (24) and the dynamic multi-commodity minimum-cost network flow problem (4) are equivalent, as summarized in the following Theorem.

**Theorem 2.** The dynamic minimum-cost network flow problem (4) and problem (24) correspond to each other in the following sense.

1. Assume that (24) has a feasible solution with finite objective value. Then (24) has a finite optimal value, and (4) has the same optimal value. Moreover, if \( M \) is an optimal solution of (24), then there is an optimal solution \( \{x_p^\ell : p \in \mathcal{P}^\ell, \ell = 1, \ldots, L\} \) of (4) such that
\[
M_{\ell,i_1...i_T} = \begin{cases} 
  x_p^\ell & \text{for } (i_1, \ldots, i_T) \in \mathcal{P}^\ell, \text{ where } p = (i_1, \ldots, i_T) \\
  0 & \text{for } (i_1, \ldots, i_T) \in \bar{\mathcal{P}} \setminus \mathcal{P}^\ell.
\end{cases}
\]
(25)

2. Assume that there is a finite feasible solution to (4). Then it has a finite optimal value, and problem (24) has the same optimal value. Moreover, if \( \{x_p^\ell : p \in \mathcal{P}^\ell, \ell = 1, \ldots, L\} \) is an optimal solution of (4), then there is an optimal solution \( M \) of (24) such that (25) holds.

**3.3. Generalizations.** In this section we have introduced novel formulations for dynamic minimum-cost network flow problems based on the optimal transport framework. We will now discuss a few modifications and generalizations of the proposed problems (18) and (24), and show that the proposed formulations in fact provide a highly flexible framework for dynamic network flow problems.
An advantage of our framework is that, due to the fact that the network structure is imposed by
the cost matrix $C$, a time-varying network can be modelled in a straightforward way. Namely, the
matrix $C$ can simply be replaced by a set of time-dependent matrices $C_t$, for $t = 1, \ldots, T - 1$, where
$C_t$ encodes the network topology in the interval $(t, t + 1)$. Moreover, based on the formulation (22),
where each commodity is described by a separate transport tensor, one can extend the problem
to the setting, where different commodities enter and leave the network at different times. In fact,
the computational methods derived in this work can easily be modified to this setting, as we will
argue in Remark 5.

In some applications, for instance in traffic flow problems, where edges and nodes describe streets
and junctions, respectively, it is natural to allow for intermediate storage on the edges. This can
be easily incorporated in our framework by letting $C_{ii}$ denote the cost for staying on edge $i \in \mathcal{E}$. It
should be noted that in this case the cost $c_{ii}$ denotes the cost for commodity $\ell$ to use edge $i \in \mathcal{E}$, and
not the cost for traveling between the two vertices. That is, the cost accumulates if flow remains
on an edge for several time intervals, which is useful, e.g., in traffic routing problems, where the
cost model should take the travel time of agents into account. However, we can achieve a cost that
does not accumulate in the case where all commodities are described by the same cost $c_i = c_i^\ell$ for
all $\ell = 1, \ldots, L$, by defining a negative cost $C_{ii} = -c_i$ for staying on the edge $i \in \mathcal{E}$.

A more classical setting in network flow problems is to allow for storage in the vertices. One
way to include this in the presented framework is to augment the support of the modes of the
mass transport tensor by the set of vertices, i.e., by letting $n = |\mathcal{E}| + |\mathcal{V}|$. In particular, in the
multi-commodity problem (24) the mass transport tensor is then of the size $M \in \mathbb{R}^{|\mathcal{E}| \times (|\mathcal{E}| + |\mathcal{V}|)}$, and
the distributions are of the size $\mu_t \in \mathbb{R}^{(|\mathcal{E}| + |\mathcal{V}|)}$, for $t = 1, \ldots, T$. Analogously to before, the network
structure is imposed by the cost matrices $C \in \mathbb{R}^{(|\mathcal{E}| + |\mathcal{V}|) \times (|\mathcal{E}| + |\mathcal{V}|)}$, i.e., we define $C_{ij} = 0$ if $i \in \{\mathcal{E} \cup \mathcal{V}\}$
is adjacent\footnote{A vertex is adjacent to all edges it connects to, and to itself.} to $j \in \{\mathcal{E} \cup \mathcal{V}\}$, and $C_{ij} = \infty$ otherwise. Similarly, the definition of the cost $C_\ell$ and
the capacities $d$ can be extended to the vertices. It is worth noting that this extension of the state
space also allows for defining the set of sinks and sources on the vertices instead of the edges.

Another extension of the formulation, of particular interest for traffic routing problems, is the
setting where the sinks and sources are defined on nodes, but intermediate storage is only allowed
in the sinks and sources, and agents are not permitted to enter sources, or leave sinks. In this case,
we let $n = |\mathcal{E}| + |\mathcal{S}^+| + |\mathcal{S}^-|$, and define the network structure through the cost matrix as follows

$$C_{ij} = \begin{cases}
0, & \text{if } i \in \{\mathcal{E} \cup \mathcal{S}^+\} \text{ is adjacent to } j \in \{\mathcal{E} \cup \mathcal{S}^+ \cup \mathcal{S}^-\} \\
0, & \text{if } i \in \{\mathcal{E} \cup \mathcal{S}^+ \cup \mathcal{S}^-\} \text{ is adjacent to } j \in \{\mathcal{E} \cup \mathcal{S}^-\}
\infty, & \text{otherwise.}
\end{cases} \quad (26)$$

A final extension worth mentioning is the possibility of introducing commodity-dependent capac-
ity constraints [39, 28]. This may be done by introducing the set of constraints $P_{0,t}(M) \leq D_{0,t}$
for $t = 2, \ldots, T - 1$, with capacity matrices $D_{0,t} \in \mathbb{R}^{L \times n}$, where $D_{0,t}^\ell_{ii}$ denotes the capacity of
commodity $\ell$ on edge $i \in \mathcal{E}$.

4. The graph-structured multi-marginal optimal transport problem. In this section we define the general graph structured optimal transport problem and develop methods to solve
the corresponding entropy regularized problem, see also [31, 32, 8, 1]. We will also consider the
dynamic flow problems in Section 3 in detail, and show how to exploit the graph-structures in order to derive efficient methods.

We have noted that the network flow problems (18) and (24) can be seen as multi-marginal
optimal transport problems with the underlying graph-structures in Figure 3 and Figure 4.
particular, we let each mode of the transport tensor $M$ be associated with a vertex, and let interaction terms be described by edges. This defines a graph $G = (V, E)$ with vertices $V$ and edges $E$. The interaction terms defining the edges are given by bi-marginal cost terms in the cost tensor, i.e., $C \in \mathbb{R}^{n_1 \times \cdots \times n_T}$ with

$$C_{i_1 \ldots i_T} = \sum_{(t_1, t_2) \in E} C^{(t_1, t_2)}_{i_1 i_2}.$$  

We denote the set of marginals that are constrained by equality and inequality constraints by $\tilde{V}_- \subset V$ and $\tilde{V}_- \subset V$, respectively. Moreover, the set of tuples that are associated with a bi-marginal constraint is denoted by $\tilde{E}$. Thus, the dynamic network flow problems (18) and (24) are special cases of the graph-structured optimal transport problem

$$\begin{align*}
\text{minimize} & \quad \langle C, M \rangle \\
\text{subject to} & \quad P_t(M) = \mu_t, \quad \text{for } t \in \tilde{V}_- , \\
& \quad P_t(M) \leq d_t, \quad \text{for } t \in \tilde{V}_- , \\
& \quad P_{t_1, t_2}(M) = R^{(t_1, t_2)}, \quad \text{for } (t_1, t_2) \in \tilde{E},
\end{align*}$$

(27)

where $\mu_t, d_t \in \mathbb{R}^{n_t}$, and $R^{(t_1, t_2)} \in \mathbb{R}^{n_1 \times n_2}$. Following the approach presented in Section 2.2.1 we develop a scheme for approximately solving optimal transport problems of this form. It is worth noting that the results in Theorem 3, Theorem 4 and Proposition 1 are only based on the structure of the constraints in (27), and thus hold for arbitrary cost tensors $C$. However, to derive the efficient schemes presented in Section 4.2 and Section 4.2 the graph-structures in the objective function have to be exploited.

4.1. Sinkhorn’s method In order to apply the approach in Section 2.2.1 we regularize (27) with an entropy term (8), which yields the regularized problem

$$\begin{align*}
\text{minimize} & \quad \langle C, M \rangle + \epsilon D(M) \\
\text{subject to} & \quad P_t(M) = \mu_t, \quad \text{for } t \in \tilde{V}_- , \\
& \quad P_t(M) \leq d_t, \quad \text{for } t \in \tilde{V}_- , \\
& \quad P_{t_1, t_2}(M) = R^{(t_1, t_2)}, \quad \text{for } (t_1, t_2) \in \tilde{E}.
\end{align*}$$

(28)

Similarly to the standard multi-marginal optimal transport problem, the solution to (28) can be expressed in terms of its optimal dual variables, as the following theorem describes.

**Theorem 3.** Assume $C$ is finite, and the prescribed marginals $\mu_t$ for $t \in \tilde{V}_-$, $d_t$ for $t \in \tilde{V}_-$, and $R^{(t_1, t_2)}$ for $(t_1, t_2) \in \tilde{E}$ are strictly positive. Moreover, assume that (28) has a feasible solution. Let $\tilde{V} = \tilde{V}_- \cup \tilde{V}_-$. Then the optimal solution to (28) has the structure $M = K \odot U$ where $K = \exp(-C/\epsilon)$ and

$$U_{t_1 \ldots t_T} = \left( \prod_{t \in \tilde{V}} (u_t)_{i_t} \right) \left( \prod_{(t_1, t_2) \in \tilde{E}} U^{(t_1, t_2)}_{(t_1, t_2)} \right),$$

(29)

where $u_t \in \mathbb{R}^{n_t}$, for $t \in \tilde{V}$, and $U^{(t_1, t_2)} \in \mathbb{R}^{n_1 \times n_2}$, for $(t_1, t_2) \in \tilde{E}$.

In particular, $u_t = \exp(-\lambda_t/\epsilon)$ and $U^{(t_1, t_2)} = \exp(-\Lambda^{(t_1, t_2)}/\epsilon)$, where $\lambda_t \in \mathbb{R}^{n_t}$ and $\Lambda^{(t_1, t_2)} \in \mathbb{R}^{n_1 \times n_2}$, for $t \in \tilde{V}$ and $(t_1, t_2) \in \tilde{E}$, respectively, are optimal variables for the dual problem of (28), which is given by

$$\begin{align*}
\text{maximize} & \quad -\epsilon \langle K, U \rangle - \sum_{(t_1, t_2) \in \tilde{E}} \langle \Lambda^{(t_1, t_2)}, R^{(t_1, t_2)} \rangle - \sum_{t \in \tilde{V}} \langle \lambda_t, \mu_t \rangle.
\end{align*}$$

(30)
Proof: Define Lagrange multipliers \( \Lambda^{(t_1, t_2)} \in \mathbb{R}^{n_{t_1} \times n_{t_2}} \) for \((t_1, t_2) \in \tilde{E} \), and \( \lambda_t \in \mathbb{R}^{n_t} \) for \( t \in \tilde{V} \).
Moreover, let \( \lambda := (\lambda_t)_{t \in \tilde{V}} \) and \( \Lambda := (\Lambda^{(t_1, t_2)})_{(t_1, t_2) \in \tilde{E}} \). With these, a Lagrangian of (28) is

\[
\mathcal{L}(M, \lambda, \Lambda) := \langle C, M \rangle + \varepsilon D(M) + \sum_{(t_1, t_2) \in \tilde{E}} \langle \Lambda^{(t_1, t_2)}, P_{t_1, t_2}(M) - R^{(t_1, t_2)} \rangle + \sum_{t \in \tilde{V}} (\lambda_t, P_t(M) - \mu_t) .
\]

(31)

The minimum of (31) with respect to \( M_{i_1...i_T} \) is achieved when its derivative vanishes, i.e., when

\[
C_{i_1...i_T} + \varepsilon \log(M_{i_1...i_T}) + \sum_{(t_1, t_2) \in \tilde{E}} \Lambda^{(t_1, t_2)}_{t_1, t_2} + \sum_{t \in \tilde{V}} (\lambda_t)_{i_t} = 0.
\]

Thus, the optimal transport tensor is of the form \( M = K \odot U \) with \( K \) and \( U \) as defined in the theorem. Note that the entropy term \( D(K \odot U) \) reads

\[
\mathcal{L}(M, \lambda, \Lambda) = -\varepsilon \langle K \odot U, C \rangle - \varepsilon \sum_{(t_1, t_2) \in \tilde{E}} \langle \Lambda^{(t_1, t_2)}, P_{t_1, t_2}(K \odot U) \rangle - \varepsilon \sum_{t \in \tilde{V}} \langle \lambda_t, P_t(K \odot U) \rangle - \langle K, U \rangle + \prod_{t=1}^T n_t .
\]

Thus, plugging \( M = K \odot U \) into \( \mathcal{L}(M, \lambda, \Lambda) \) in (31), and removing constants, yields

\[
-\varepsilon \langle K, U \rangle - \sum_{(t_1, t_2) \in \tilde{E}} \langle \Lambda^{(t_1, t_2)}, R^{(t_1, t_2)} \rangle - \sum_{t \in \tilde{V}} \langle \lambda_t, \mu_t \rangle .
\]

(32)

The dual to (28) is to maximize (32) with respect to \( \Lambda^{(t_1, t_2)} \) for \((t_1, t_2) \in \tilde{E} \), and \( \lambda_t \) for \( t \in \tilde{V} \). Finally, given the assumptions, strong duality holds between the primal and the dual problem, see, e.g., [11, p. 226].

The assumptions in Theorem 3 are typically not satisfied for the network flow problems (18) and (24). If the underlying network is not a complete graph, the cost tensor has infinite entries. Moreover, in most flow problems, the sources and sinks are a strict subset of the set of edges, which is modeled by zero entries in the prescribed marginals \( \mu_t \), or \( R^{(t_1, t_2)} \). The following theorem extends Theorem 3 to these cases.

**Theorem 4.** Let \( C \in \mathbb{R}^{n_1 \times \cdots \times n_T} \) and assume that there is a feasible solution \( M \) of (28) for which \( M_{i_1...i_T} > 0 \) if and only if \( C_{i_1...i_T} < \infty \), \( \mu_t > 0 \), \( d_t > 0 \), and \( R^{(t_1, t_2)}_{i_1, i_2} > 0 \). Then the optimal solution to (28) has the structure \( M = K \odot U \) where \( K = \exp(-C/\varepsilon) \) and \( U \) factorizes as in (29).

**Proof:** Define the set of tuples

\[
I = \{(i_1, \ldots, i_T)|i_i \in \{1, \ldots, n\}, C_{i_1...i_T} < \infty, (\mu_t)_{i_t} > 0, (d_t)_{i_t} > 0, R^{(t_1, t_2)}_{i_1, i_2} > 0\}.
\]

For \((i_1, \ldots, i_T) \in I\) we define \( \hat{C}_{i_1...i_T} = C_{i_1...i_T}, (\hat{\mu}_t)_{i_t} = (\mu_t)_{i_t}, (\hat{d}_t)_{i_t} = (d_t)_{i_t} \), and \( \hat{R}^{(t_1, t_2)}_{i_1, i_2} = R^{(t_1, t_2)}_{i_1, i_2} \).

Consider the problem

\[
\begin{array}{ll}
\text{minimize} & \sum_{(i_1, \ldots, i_T) \in I} \hat{C}_{i_1...i_T} M_{i_1...i_T} + \varepsilon D(\hat{M}) \\
\text{subject to} & P_t(\hat{M}) = \hat{\mu}_t, \quad \text{for } t \in \tilde{V}_1, \\
& P_t(\hat{M}) \leq \hat{d}_t, \quad \text{for } t \in \tilde{V}_2, \\
& P_{t_1, t_2}(\hat{M}) = \hat{R}^{(t_1, t_2)}_{i_1, i_2}, \quad \text{for } (t_1, t_2) \in \tilde{E},
\end{array}
\]

(33)
where the definition of $D(M)$, $P_i(M)$, and $P_{t_1,t_2}(M)$ is relaxed to the case where the argument
is not a tensor. The proof of Theorem 3 can be mirrored for the case where the variable is not
a tensor. Thus, the optimal solution to (33) can be written as $M_{i_1\ldots i_T} = K_{i_1\ldots i_T} U_{i_1\ldots i_T}$, where
$K_{i_1\ldots i_T} = \exp(-C_{i_1\ldots i_T}/\epsilon)$, and

$$
\hat{U}_{i_1\ldots i_T} = \left( \prod_{t \in \hat{V}} (\hat{u}_t)_{i_t} \right) \left( \prod_{(t_1,t_2) \in \hat{E}} \hat{U}^{(t_1,t_2)}_{i_{t_1} i_{t_2}} \right),
$$

where $(i_1,\ldots, i_T) \in I$. Now, define the tensors $K = \exp(-C/\epsilon)$ and $U \in \mathbb{R}^{n_1 \times \cdots \times n_T}$, which is con-
structed as in (29), where

$$(u_t)_{i_t} = \begin{cases} (\hat{u}_t)_{i_t}, & \text{if it is defined,} \\ 0, & \text{otherwise.} \end{cases} \quad U^{(t_1,t_2)}_{i_{t_1} i_{t_2}} = \begin{cases} \hat{U}^{(t_1,t_2)}_{i_{t_1} i_{t_2}}, & \text{if it is defined,} \\ 0, & \text{otherwise.} \end{cases} \quad (34)$$

Then by construction $M = K \otimes U$ is an optimal solution to (28).

The Sinkhorn iterations for problem (28) can be derived as a block-coordinate ascend method
in the dual problem (30), as summarized in the following proposition.

**Proposition 1.** Assume (28) has a feasible solution as in the assumptions of Theorem 4. Let
$K = \exp(-C/\epsilon)$ and $U$ as defined in (29). Then, the iterative scheme

$$
U^{(t_1,t_2)} \leftarrow U^{(t_1,t_2)} \odot R^{(t_1,t_2)} \odot P_{t_1,t_2}(K \otimes U), \quad \text{for } (t_1,t_2) \in \hat{E} \quad (35a)
$$

$$
u_t \leftarrow u_t \odot \mu_t, \quad \text{for } t \in \hat{V} \quad (35b)
$$

$$
u_t \leftarrow \min(u_t \odot d_t, \nu_t), \quad \text{for } t \in \hat{V} \quad (35c)
$$

converges linearly, and in the limit point the optimal solution of (28) is given by $M = K \otimes U$.

**Proof:** We first assume that the stronger assumptions from Theorem 3 hold. The scheme is
derived as a block coordinate ascent method in the dual (30). This is to maximize the objective
with respect to one set of dual variables while keeping the other dual variables fixed, i.e., to perform
the updates

$$
\Lambda^{(t_1,t_2)} \leftarrow \arg \max_{\Lambda^{(t_1,t_2)} \in \mathbb{R}^{n_1 \times n_2}} -\epsilon(K, U) - \langle \Lambda^{(t_1,t_2)}, R^{(t_1,t_2)} \rangle, \quad \text{for } (t_1,t_2) \in \hat{E} \quad (36a)
$$

$$
\lambda_t \leftarrow \arg \max_{\lambda_t \in \mathbb{R}^{n_T}} -\epsilon(K, U) - \langle \lambda_t, \mu_t \rangle, \quad \text{for } t \in \hat{V} \quad (36b)
$$

$$
\lambda_t \leftarrow \arg \max_{\lambda_t \in \mathbb{R}^{n_T}} -\epsilon(K, U) - \langle \lambda_t, d_t \rangle, \quad \text{for } t \in \hat{V} \quad (36c)
$$

The objectives of the unconstrained problems (36a) and (36b) are strictly concave, and thus a
necessary and sufficient condition for optimality is that the respective gradient vanishes. Note that
for each $(t_1,t_2) \in E$ the gradient of (36a) with respect to $\Lambda^{(t_1,t_2)}$ is

$$
\exp(-\Lambda^{(t_1,t_2)}/\epsilon) \odot \left( \sum_{i_0\ldots,i_T \setminus \{i_t,i_{t_2}\}} K_{i_0\ldots,i_T} \left( \prod_{t \in \hat{V}} (u_t)_{i_t} \right) \left( \prod_{(t_1,t_2) \in \hat{E} \setminus (i_t,i_{t_2})} U_{i_{t_1} i_{t_2}}^{(t_1,t_2)} \right) \right) - R^{(t_1,t_2)},
$$

and setting it to zero gives (35a). Similarly, for $t \in V_-$ the gradient of (36b) with respect to $\lambda_t$ is

$$
\exp(-\lambda_t/\epsilon) \odot \left( \sum_{i_0\ldots,i_{t-1},i_{t+1},\ldots,i_T} K_{i_0\ldots,i_T} \left( \prod_{\tau \in \hat{V} \setminus \{t\}} (u_\tau)_{i_\tau} \right) \left( \prod_{(t_1,t_2) \in \hat{E}} U_{i_{t_1} i_{t_2}}^{(t_1,t_2)} \right) \right) - \mu_t.
$$
which yields (35b). Finally, note that the objective in (36c) can be written as

\[
\sum_{t} \left( -\epsilon e^{-\left(\lambda_{t}\right)_{it}} \left( \sum_{i_{0},\ldots,i_{t-1},i_{t+1},\ldots,i_{T}} K_{i_{0},\ldots,i_{T}} \left( \prod_{\tau \in \tilde{V} \setminus \{t\}} (u_{t})_{i_{\tau}} \right) \left( \prod_{(t_{1},t_{2}) \in E} (U_{i_{t_{1}},i_{t_{2}}})^{(t_{1},t_{2})} \right) \right) -(\lambda_{t})_{it} d_{it} \right).
\]

Thus, the maximization in (36c) can be performed in each element of \( \lambda_{t} \) individually. If the derivative of the objective in (36c) with respect to \((\lambda_{t})_{it}\) vanishes for a feasible, i.e., non-negative, point, then this is the global maximizer. Otherwise, the maximizer is the projection on the feasible set, i.e., \((\lambda_{t})_{it} = 0\). This yields (35c). The linear convergence of the scheme follows from [45].

In case the assumptions in Theorem 4 are satisfied, we perform a block coordinate ascent in the dual of (33). The dual variables can then be augmented by zero-entries as in (34) to arrive at the scheme in the Proposition.

Computing the projections of \( K \odot U \) in Proposition 1 is in general still expensive, since computing the sums in (6) and (15) requires \( \mathcal{O}(n^T) \) operations. However, in the dynamic minimum-cost flow problems, there are additional structures in the cost tensor \( C \), and thus in the tensor \( K \). Namely, these tensors decouple according to the graphs in Figures 3 and 4. The next subsections describe how these structures can be utilized in order to efficiently compute the projections needed to apply the scheme in Proposition 1.

4.2. Sinkhorn’s method for the dynamic minimum-cost flow problem. Recall that the dynamic minimum-cost flow problem (18) is a multi-marginal optimal transport problem. In particular, it can be written on the form (27), where \( \tilde{V} = \{1, T\}, \tilde{V}_c = \{2, \ldots, T - 1\} \) and \( E = \emptyset \). Adding the entropy term (8) yields then an entropy regularized problem (28), which in this case explicitly reads

\[
\min_{M \in \mathbb{R}_{+}^{n \times T}} \langle C, M \rangle + \epsilon D(M)
\]

subject to

\[
P_{t}(M) \leq d_{t}, \quad \text{for} \quad t = 2, \ldots, T - 1
\]

\[
P_{1}(M) = \mu_{1},
\]

\[
P_{T}(M) = \mu_{t},
\]

where \( C \) is defined by

\[
C_{i_{1},\ldots,i_{T}} = \sum_{t=1}^{T} c_{it} + \sum_{t=1}^{T-1} C_{i_{t}i_{t+1}}.
\]

REMARK 2. Without the inequality constraints \( P_{t}(M) \leq d_{t} \), and with zero cost on the edges, \( c = 0 \), the entropy-regularized problem (37) is a discrete Schrödinger bridge problem [50, 30, 31]. The Schrödinger bridge problem is tightly connected to optimal transport [14, 41]. It is a popular tool in ensemble control applications, as it provides a framework for steering a given distribution, i.e., an ensemble of agents, to a target one [15, 12]. In particular, network flow problems of this form have previously been considered in [16, 17, 18]. This connection to the Schrödinger bridge problem gives another motivation for adding the regularizing entropy term to the objective of (18). Namely, the Schrödinger bridge problem on a network can be interpreted as an ensemble of agents, which are each evolving according to a Markov chain [30, 50]. The entropy term thus induces a stochastic component to the problem, which yields a more smoothed out solution. Therefore, the solutions to the regularized problem (37) can be understood as robust transport plans [16, 17, 18].

According to Theorem 4, the solution to the regularized problem (37) is of the form \( M = K \odot U \), where

\[
K_{i_{1},\ldots,i_{T}} = \left( \prod_{t=1}^{T} k_{it} \right) \left( \prod_{t=1}^{T-1} K_{i_{t}i_{t+1}} \right),
\]
The Sinkhorn algorithm (efficiently \[\text{Proposition 2}\] in particular, the solution is found by iterating

\[
\begin{align*}
\text{Initialize } & \ u_1, \ldots, u_T, \ t = 1, \ \hat{\varphi}_1 = 1, \ \varphi_T = 1 \\
\text{while } & \text{Not converged do} \\
\text{for } & \ t = T - 1, \ldots, 1 \text{ do} \\
\text{Update } & \varphi_t \leftarrow K(u_{t+1} \odot k_{t+1} \odot \varphi_{t+1}) \\
\text{end for} \\
\text{Update } & \ u_1 \leftarrow \mu_1 / \varphi_1 \\
\text{for } & \ t = 2, \ldots, T - 1 \text{ do} \\
\text{Update } & \hat{\varphi}_t \leftarrow K^T(u_{t-1} \odot k_{t-1} \odot \hat{\varphi}_{t-1}) \\
\text{Update } & \ u_t \leftarrow \min (d_t / (\varphi_t \odot \hat{\varphi}_t \odot k_t), 1) \\
\text{end for} \\
\text{Update } & \ u_T \leftarrow \mu_T / \varphi_T \\
\text{end while} \\
\text{return } & \ u_1, \ldots, u_T
\end{align*}
\]

with \( k = \exp(-c/\epsilon) \) and \( K = \exp(-C/\epsilon) \), and \( U = u_1 \odot \cdots \odot u_T \). The components of the tensor \( U \) can be found utilizing Proposition 1. In particular, the solution is found by iterating

\[
\begin{align*}
\text{Initialize } & \ u_t \leftarrow u_t \odot \mu_t / P_t(K \odot U), \quad \text{for } t = 1, T, \\
\text{Initialize } & \ u_t \leftarrow \min (u_t \odot d_t / (P_t(K \odot U))_1), \quad \text{for } t = 2, \ldots, T - 1. \\
\end{align*}
\]

In this case, where the cost decouples according to a path graph, the projections can be computed efficiently \[\text{Proposition 2}\]. Namely, the projections for this problem are of the form

\[
P_t(K \odot U) = u_t \odot k_t \odot \hat{\varphi}_t \odot \varphi_t,
\]

for \( t = 1, \ldots, T \), where

\[
\begin{align*}
\hat{\varphi}_t &= K^T \text{diag}(u_{t-1} \odot k_{t-1}) K^T \cdots \text{diag}(u_2 \odot k_2) K^T (u_1 \odot k_1), \\
\varphi_t &= K \text{diag}(u_{t+1} \odot k_{t+1}) K \cdots \text{diag}(u_{T-1} \odot k_{T-1}) K (u_T \odot k_T).
\end{align*}
\]

The Sinkhorn algorithm (38) is summarized in Algorithm 1.

Note that intermediate results of (40a) and (40b) are stored, and the updates in (38) are scheduled such that for each update only one matrix-vector multiplication needs to be performed. Thus, in the case of a dense matrix \( K \), one iteration sweep, i.e., once updating all vectors \( u_t \), for \( t = 1, \ldots, T \), is of complexity \( O(Tn^2) \). However, for sparse networks the matrix \( K \) is also sparse, and thus the matrix multiplications required to compute the projections (39) via (40) become even more efficient, as discussed in the following remark.

Remark 3. Note that \( K_{ij} = 0 \) if \( (i, j) \notin \mathcal{E} \), and \( K_{ij} > 0 \) if \( (i, j) \in \mathcal{E} \). Thus, multiplication with a vector \( v \in \mathbb{R}^n \) can be performed as

\[
(Kv)_i = \sum_{j \in N(i)} K_{ij} v_j.
\]

This multiplication is of order \( O(\Delta(N) \cdot n) \), where \( \Delta(N) \) is the maximum degree of \( N \), i.e., the highest number of neighboring nodes among the nodes \( V \). The complexity of one iteration sweep in Algorithm 1 is thus \( O(Tn\Delta(N)) \).
4.3. Sinkhorn’s method for the dynamic multi-commodity minimum-cost flow problem. Similarly to the previous section, the multi-commodity problem (24) is also a multi-marginal optimal transport problem of the form (27). In particular, here the constraint sets are \(V_n = \emptyset, \ V_s = \{2, \ldots, T - 1\} \) and \(E = \{(0, 1), (0, T)\}\). Regularizing the problem with an entropy term, it is of the form (28), which in this case reads

\[
\begin{align*}
\text{minimize} & \quad \langle C, M \rangle + \epsilon D(M) \\
\text{subject to} & \quad P_{0,1}(M) = R^{(0,1)} , \\
& \quad P_{0,T}(M) = R^{(0,T)} , \\
& \quad P_t(M) \leq d_t, \quad \text{for } t = 2, \ldots, T - 1,
\end{align*}
\]

where \(C\) is defined by

\[
C_{i_0 \ldots i_T} = \sum_{t=2}^{T-1} (C_L)_{i_0 i_t} + \sum_{t=1}^{T-1} C_{i_{t+1} i_t}.
\]

The solution to (41) can again be expressed in terms of its dual variables, as described in Theorem 4. In particular, the optimal mass transport plan is of the form \(M = K \otimes U\), where \(K\) factorizes as

\[
K_{i_0 \ldots i_T} = \left(\prod_{t=2}^{T-1} (K_L)_{i_0 i_t}\right) \left(\prod_{t=1}^{T-1} K_{i_{t+1} i_t}\right),
\]

where \(K_L = \exp(-C_L/\epsilon)\) and \(K = \exp(-C/\epsilon)\). Moreover, the tensor \(U\) is of the form

\[
U_{i_0 \ldots i_T} = U^{(0,1)}_{i_0 i_1} U^{(0,T)}_{i_T i_1} \prod_{t=2}^{T-1} (u_t)_{i_t},
\]

and its components can be found according to Propositions 1 by iteratively updating

\[
U_t \leftarrow U_t^{(0,t)} \odot R^{(0,t)} / P_{0,t}(K \odot U), \quad \text{for } t = 1, T,
\]

\[
u_t \leftarrow \min \left(\nu_t \odot d_t / P_t(K \odot U, 1)\right), \quad \text{for } t = 2, \ldots, T - 1.
\]

Again, the tensor \(K \odot U\) has a graph structure, which is illustrated in Figure 4. This graph contains cycles, and thus the results from [31] cannot be utilized. Nevertheless, the projections can be computed relatively efficiently, as demonstrated by the next theorem.

THEOREM 5. Consider the tensors \(K = \exp(-C/\epsilon)\), with \(C\) defined as in (23) and \(\epsilon > 0\), and \(U\) in (43). With the matrices \(K_L = \exp(-C_L/\epsilon)\) and \(K = \exp(-C/\epsilon)\), define

\[
\hat{\Psi}_t = \begin{cases} 
U^{(0,1)}_{i_0 i_1} K, & t = 2, \\
(\hat{\Psi}_{i-1} \odot K_L) \text{diag}(u_{i-1}) K, & t = 3, \ldots, T,
\end{cases}
\]

and

\[
\Psi_t = \begin{cases} 
U^{0,T} K^T, & t = T - 1, \\
(\Psi_{i+1} \odot K_L) \text{diag}(u_{i+1}) K^T, & t = 1, \ldots, T - 2.
\end{cases}
\]

Then, the bi-marginal projections of the tensor \(K \odot U\) are

\[
\begin{align*}
P_{0,1}(K \odot U) &= U^{(0,1)} \odot \Psi_1 \\
P_{0,T}(K \odot U) &= U^{(0,T)} \odot \Psi_T \\
P_{0,t}(K \odot U) &= \left(\hat{\Psi}_t \odot \Psi_t \odot K_L\right) \text{diag}(u_t), \quad \text{for } t = 2, \ldots, T - 1.
\end{align*}
\]
Proof: Note that the tensor $K = \exp(- C/\varepsilon)$ is element-wise defined as in (42), thus the bi-marginal projections of the tensor $K \odot U$ on the marginals $0$ and $t$, where $t \in \{2, \ldots, T-1\}$, are given by

$$
P_{0,t}(K \odot U) = \sum_{i_1, \ldots, i_{t-1}, i_{t+1}, \ldots, i_T} \left( \prod_{s=1}^{T-1} K_{i_{si_{s+1}}} \right) \left( \prod_{s=2}^{T-1} (K_L)_{i_{0si}} \right) U_{i_{0i_1}i_1}^{(0,1)} \prod_{s=2}^{T-2} (u_{is})_{is}
$$

where

$$
\hat{\Psi}_t = \sum_{i_1, \ldots, i_{t-1}} U_{i_{0i_1}}^{(0,1)} K_{i_{1i_2}} \left( \prod_{s=2}^{t-1} (K_L \text{diag}(u_{is}))_{i_{0is}} K_{is_{s+1}} \right),
$$

and

$$
\Psi_t = \sum_{i_{t+1}, \ldots, i_T} U_{i_{ti_T}}^{(0,T)} K_{i_{t-1}i_T} \left( \prod_{s=t+1}^{T-1} (K_L \text{diag}(u_{is}))_{i_{0is}} K_{is_{s+1}} \right).
$$

These terms lead to the recursive definitions of $\hat{\Psi}_t$ and $\Psi_t$ in (45) and (46). The projections $P_{0,1}(K \odot U)$ and $P_{0,T}(K \odot U)$ are derived similarly. □

The projections on one marginal can then be found by projecting the bi-marginal projections in (47) on one of the marginals, which yields the following.

**Corollary 1.** The marginals of the tensor $K \odot U$ in Theorem 5 are given by

$$
P_t(K \odot U) = u_t \odot \left( \hat{\Psi}_t \odot \Psi_t \odot K_L \right)^T \mathbf{1}, \quad \text{for } t = 2, \ldots, T-1,
$$

$$
P_0(K \odot U) = \left( \hat{\Psi}_t \odot \Psi_t \odot K_L \right) u_t.
$$

Theorem 5 and Corollary 1 describe an efficient way to compute the projections required for the Sinkhorn scheme (44), and the resulting computational method is summarized in Algorithm 2. Similarly to the algorithm for the single-commodity setting, intermediate results can be stored and utilized.

**Remark 4.** The computational bottleneck of the Sinkhorn iterations lies in updating the projections. One iteration sweep of the Sinkhorn iterations requires updating each of the matrices in (45) and (46) once. For dense matrices $K$ each of these updates is of complexity $O(Ln^2)$, and thus one full iteration sweep can be done in $O(TLn^2)$. However, as noted in Remark 3, the matrix $K$ inherits the sparsity of the network, and this can be exploited to perform the matrix multiplications in (45) and (46) more efficiently. Thus, the complexity of the matrix-matrix multiplication is decreased to $O(\Delta(N) \cdot Ln)$, and one full iteration sweep can be done in $O(T \Delta(N) \cdot Ln)$.

**Remark 5.** In Section 3.2 we have formulated the multi-tensor problem (22) as the one-tensor problem (24) in order to bring it on the form of a graph-structured optimal transport problem (27) and then solve it. Alternatively, we could have regularized each of the $L$ optimal transport problems in (22) separately, yielding the regularized problem

$$
\text{minimize } \sum_{i=1}^{L} \left( (C, M^\ell) + \varepsilon D(M^\ell) \right)
$$

subject to

$$
P_1(M^\ell) = \mu_1^\ell, \quad \ell = 1, \ldots, L,
$$

$$
P_T(M^\ell) = \mu_T^\ell, \quad \ell = 1, \ldots, L,
$$

$$
\sum_{\ell=1}^{L} P_t(M^\ell) \leq d \quad \text{for } t = 2, \ldots, T-1,
$$

where $d$ is the demand.
Algorithm 2 Scheme for solving the dual of the regularized dynamic multi-commodity flow problem (41)

Initialize $u_2, \ldots, u_{T-1}, U^{(0,1)}, U^{(0,T)}$
Compute $\Psi_t$, for $t = 1, \ldots, T$

while Not converged do
  Update $U^{(0,1)} \leftarrow R^{(0,1)}/\Psi_1$
  Update $\hat{\Psi}_2 \leftarrow U^{(0,1)}K$
  for $t = 2, \ldots, T-1$ do
    Update $u_t \leftarrow \min \left( d./((\hat{\Psi}_t \circ \Psi_t \circ K)^T 1), 1 \right)$
    Update $\hat{\Psi}_{t+1} \leftarrow (\hat{\Psi}_t \circ K_L)\text{diag}(u_t)K$
  end for
  Update $U^{(0,T)} \leftarrow R^{(0,T)}/\hat{\Psi}_T$
  Update $\Psi_{T-1} \leftarrow U^{(0,T)}K^T$
  for $t = T-1, \ldots, 2$ do
    Update $\Psi_{t-1} \leftarrow (\Psi_t \circ K_L)\text{diag}(u_t)K^T$
  end for
end while

return $u_2, \ldots, u_{T-1}, U^{(0,1)}, U^{(0,T)}$

where $C$ is defined as in (16). In fact, this problem is equivalent to the regularized problem (41). Moreover, in this representation the Sinkhorn iterations are given by

$$u_t \leftarrow u_t \circ P_t(K^t \circ U^t), \quad \text{for } t = 1, \ldots, L$$

$$u_t \leftarrow \min \left( u_t \circ d./\left( \sum_{t \in L} P_t(K^t \circ U^t) \right), 1 \right), \quad \text{for } t = 2, \ldots, T - 1$$

$$u_{T-1} \leftarrow u_{T-1} \circ \mu_T \circ (K^T \circ U^t), \quad \text{for } \ell = 1, \ldots, L$$

and these are equivalent to the Sinkhorn iterations derived above (cf. (44)). Recall from Section 3.3 that one convenient feature of formulation (22) is that it can be easily extended to allow for commodities that enter and leave the network at different times. Therefore, as can be seen here, such problems can also be solved efficiently.

5. Simulations. In this Section we illustrate the computational efficiency of our proposed framework. First, we compare its performance with a Simplex solver on two different types of networks. Finally, we illustrate it in a traffic routing problem with a large number of commodities.

5.1. Performance study on a sparse grid network. We first consider a dynamic multi-commodity minimum-cost network flow problem on a sparse network. To this end, let $\mathcal{N}$ be a grid of $10 \times 10$ nodes, and let the source $\mathcal{S}^+$ for all commodities be an incoming edge to one corner of the square, and let the sink $\mathcal{S}^-$ be an outgoing edge from the opposite corner. Thus, the total number of directed edges is $n = 84$. Moreover, in this set-up the sink and source can be understood as the two corner vertices. We consider the case of $L = 50$ commodities, and let the total flow of each commodity be 1, that is $\mu_0 = 1$. Moreover, the capacity vector $d \in \mathbb{R}^n_+$ is defined as $d_i = L$ for $i \in \{\mathcal{S}^+ \cup \mathcal{S}^-\}$, and $d_i = 1$ otherwise. Here we do not allow for intermediate storage on the vertices or the edges, except in the sink and source. This problem is solved for a time horizon of $T = 80$ utilizing Algorithm 2. We also solve the problem in node-edge formulation (cf. [26, 55]) in the time-expanded network using the solver CPLEX [35]. In our experiments we observed that CPLEX performs best when using the dual simplex algorithm, and we thus assign this algorithm when
calling CPLEX to decrease its start-up time. We run this problem for 10 different experiments, where in each trial the cost for a unit flow of each commodity on each edge is randomly assigned from a uniform distribution on $[0, 1]$, that is we let $c^\ell_e \sim \text{Unif}([0, 1])$, for $\ell = 1, \ldots, L$, and $e \in \mathcal{E}$.

Figure 5 shows some measures of error as a function of computation time. The error in the objective value compares the objective value of the current solution to the optimal objective value. Note that our proposed algorithm is based on the regularized problem (41), and therefore cannot achieve the true objective value. However, the smaller the regularization parameter $\epsilon$, the closer we get to the true optimum. Since CPLEX utilizes a dual simplex method, its solution becomes meaningful only after the last iteration, in the sense that not all constraints are fulfilled for the intermediate iterates. In particular, this is the case for the flow balance constraint. In contrast, when solving (41) using Algorithm 2, the intermediate iterates by construction satisfy the flow balance constraints in the nodes, and they also satisfy the mismatch in the sinks and sources to machine precision. Moreover, Algorithm 2 converges linearly to the optimal solution of (41).

With the smallest tested regularization parameter, $\epsilon = 0.01$, the capacity constraints are satisfied in about 0.1 seconds. On the other hand CPLEX takes more than 10 seconds to find a solution, which is two orders of magnitude longer than Algorithm 2. Note that state-of-the-art methods for multi-commodity flows can typically not be expected to improve the run time by more than an order of magnitude as compared to standard LP solvers [4, 52, 40]. Our proposed algorithm is thus competitive with specialized state-of-the-art methods for network flow problems.

5.2. Performance study on a dense random network. Next, we study the performance of Algorithm 2 in a less favourable setting. Here we consider a dense random network with 40 nodes. Between each (ordered) pair of nodes we create a directed edge with probability $1/2$. The expected value of the number of edges in the network is thus $\binom{40}{2} = 780$. Moreover, we allow for intermediate storage in the nodes, but not in the edges. Therefore, we augment the state space by the set of nodes as described in Section 3.3, and the expected size of the distributions support is
Figure 6. Performance study on a larger dense random network. The plots show the $L_1$-norm of the objective value error and constraint mismatches of the current solution over time. Circles denote each experiment's outcome, and the full lines are plotted between their means. The machine precision is $10^{-16}$.

thus $E[n] = 820$. We equip each of the $L = 100$ commodities with a random source and sink on the set of nodes. The total flow of each commodity is set to 1, i.e., $\mu_0 = 1$, and the capacity vector $d \in \mathbb{R}_+^n$ is defined as $d_i = 1$, if $i \in \mathcal{E}$, and $d_i = L$, if $i \in \mathcal{V}$. As in the previous example, the cost for each commodity and each edge is assigned from a uniform distribution on $[0, 1]$. Moreover, the cost for intermediate storage on the nodes is 0. We solve the problem for $T = 100$ time intervals using Algorithm 2 and solve its node-edge formulation in the time-expanded network with the dual simplex algorithm in CPLEX. Here, the time expanded network has 4000 nodes and in the mean 81180 edges.

The performance for 10 trials of the described setup is illustrated in Figure 6. Qualitatively, we see a similar behavior as in the performance study for the sparse network in Section 5.1. In particular, as in the previous example, in contrast to the intermediate iterates produced by our method, the intermediate iterates produced by CPLEX do not correspond to flows since the flow balance constraint is in general not fulfilled. Moreover, our method converges linearly to an optimal solution of (41), and with the smallest tested regularization parameter $\epsilon = 0.0025$ Algorithm 2 gives a good approximation to the optimal solution in about one second, whereas the CPLEX solver requires about 15 seconds to find a solution. Even in the less favourable setting of a dense network with intermediate storage on the nodes we thus still get an approximate solution in less than an order of magnitude of CPLEX’s run-time.

5.3. Traffic routing problem with a large number of commodities. We apply our framework to a traffic routing problem in the street network illustrated in Figure 7, which consists of 57 nodes and 150 directed edges. Let every node be both a sink and a source, and as described in Section 3.3 we thus let the state space be of size $n = 150 + 2 \cdot 57 = 264$. Assume that there is an equal amount of 10 agents travelling between every pair of nodes. This can be modelled by associating each node with one commodity, and imposing that every commodity is initially uniformly distributed on the set of sources, and finally concentrated in the associated sink node. In
Figure 7. Map of a street network. Every edge represents two directed edges, one in each direction. Broader edges represent highways. The three stars represent three different commodities' sinks.

In particular, this means that the number of commodities is $L = 57$, and the two matrix constraints in (24) are defined by the matrices $R^{(0,1)}, R^{(0,T)} \in \mathbb{R}_{+}^{L \times n}$ with entries

$$R^{(0,1)}_{\ell,i} = \begin{cases} 10, & \text{if } i \in S^{+}_\ell = \mathcal{V}, \\ 0, & \text{otherwise}, \end{cases} \quad R^{(0,T)}_{\ell,i} = \begin{cases} 570, & \text{if } i \in S^{-}_\ell, \\ 0, & \text{otherwise}. \end{cases}$$

We consider the scenario with intermediate storage in the edges, but without storage on the nodes. However, agents are permitted to stay in their respective sink and source, but once they leave their source they may not return to it, and once they reach their sink they may not leave it. This structure is imposed by the cost matrix $C$ as defined in (26). The wider streets in Figure 7 describe highways, and we denote the set of highways as $\mathcal{H}$. Since our framework assumes uniform travel time on all edges, the fact that the roads in $\mathcal{H}$ are longer than the other roads models that agents can drive faster on the highway. Let $l_i$ denote the Euclidean length of road $i \in \mathcal{E}$. We define the capacities for each state as

$$d_i = \begin{cases} 100l_i, & \text{if } i \in \mathcal{H}, \\ 20l_i, & \text{if } i \in \mathcal{E} \setminus \mathcal{H}, \\ 100L, & \text{if } i \in \mathcal{S}. \end{cases}$$

The cost for an agent to be in any of the states is defined in the matrix $C_L$. The costs are assumed equal for all agents and defined for all commodities $\ell = 1, \ldots, L$ as

$$(C_L)_{\ell i} = \begin{cases} 0.01, & \text{if } i \in S^{+}, \\ 0.1, & \text{if } i \in \mathcal{E}, \\ 0, & \text{if } i \in S^{-}. \end{cases}$$

Thus, the central controller aims to minimize the time agents spend inside the network, and makes them reach the sink early rather than wait in the source. We consider the problem with final time $T = 30$. The problem is solved using Algorithm 2 with regularization parameter $\epsilon = 0.01$. For the three commodities associated with the sinks highlighted in Figure 7, the optimal flows are visualized in Figure 8. One can see that traffic is sent at all places in the network, and finally concentrates towards the three sinks. For the three commodities the amount of agents in the sources, roads, and sinks, respectively, is plotted over time in Figure 9(a). The total flows distribution over source, roads, and sink over time can be seen in the blue lines in Figure 9(b). At the first time instance
Figure 8. The optimal traffic flow over time for three of the commodities.

(a) Number of agents in sources, roads, and sinks, over time for the three commodities in Figure 8.

(b) Blue curves correspond to all agents in the scenario in Figure 8. Green and red curves describe the scenario, where the cost for staying in a source is equal to the cost on the roads (0.1) and to the cost for staying in a sink (0), respectively.

Figure 9. Agents status over time.

many agents are sent from the sources into the network. Towards the end of the time interval less and less agents are on the roads.

We also vary the cost for agents to stay in the source, see Figure 9(b). Clearly, if the cost for being in a source is increased to \((C_L)_{\ell i} = 0.1\), for \(i \in \mathcal{S}^+\) and \(\ell = 1, \ldots, L\), more agents are sent into the network early on. If the cost for being in a source is equal to being in a sink, i.e., \((C_L)_{\ell i} = 0\), for \(i \in \mathcal{S}^+\) and \(\ell = 1, \ldots, L\), the amount of flow on the roads over time looks very symmetric.

Finally, we consider a scenario where a second type of commodity is present in the network. Therefore, the total amount of commodities is increased to \(2L = 114\). We interpret the first set of \(L\) commodities as cars and denote them as \(\mathcal{L}_C\). The second set of \(L\) commodities are interpreted as trucks and denoted by \(\mathcal{L}_T\). For each set of commodities, the initial and final distributions are defined as before, but the number of agents in each commodity is halved in order to get the same
total number of agents. That is, we define the new constraint matrices \( \hat{R}^{(0,1)}, \hat{R}^{(0,T)} \in \mathbb{R}_+^{2L \times n} \) as
\[
\hat{R}^{(0,1)} = \frac{1}{2} \begin{bmatrix} R^{(0,1)}_{(0,1)} \end{bmatrix}, \quad \hat{R}^{(0,T)} = \frac{1}{2} \begin{bmatrix} R^{(0,T)}_{(0,1)} \end{bmatrix}.
\]

For the agents in \( L_C \), the costs to be on an edge, sink or source is defined as before, i.e., for \( \ell \in L_C \), it is given by (48). Trucks are incentivized to use highways as much as possible by an increased cost for agents in \( L_T \) to be on small roads. Thus, we define the modified cost matrix \( \hat{C}_L \in \mathbb{R}_+^{2L \times n} \) by
\[
(C_L)_{\ell i} = \begin{cases} 
(C_L)_{\ell i}, & \text{if } \ell \in L_C \\
0.01, & \text{if } \ell \in L_T, i \in S^+ \\
0.1, & \text{if } \ell \in L_T, i \in H \\
0.7, & \text{if } \ell \in L_T, i \in E \setminus H \\
0, & \text{if } \ell \in L_T, i \in S^-.
\end{cases}
\]

The rest of the problem is set up as before, and we solve it with Algorithm 2 and regularization parameter \( \epsilon = 0.01 \). For each of the three sinks highlighted in Figure 7, we consider the two associated commodities, and show the number of agents on the small roads and highways over time in Figure 10. As expected, the trucks avoid the small roads and mainly use the highways. In order to not exceed the capacity constraints on the highways, the cars are thus forced to the small roads.

6. Conclusion. We have developed a novel framework for dynamic network flow problems, which is based on formulating the problem as a structured multi-marginal optimal transport problem. Regularizing the problem with an entropy term opens up for efficiently finding an approximate solution. By taking advantage of the graph-structure in the optimal transport formulations, we derived a scheme that is computationally highly efficient, as well as easy to implement. Its competitiveness with state-of-the-art methods for network flow problems is experimentally illustrated in performance studies and on a traffic routing problem with a huge number of commodities.

References
[1] Altschuler JM, Boix-Adsera E (2020) Polynomial-time algorithms for multimarginal optimal transport problems with structure. Preprint. arXiv:2008.03006 38 pages.
[2] Aronson J (1989) A survey of dynamic network flows. Annals of Operations Research 20(1):1–66.
[3] Bacon X (2020) Multi-species optimal transportation. Journal of Optimization Theory and Applications 184(2):315–337.
[4] Barnhart C, Krishnan N, Vance PH, Floudas C, Pardalos P (2009) Multicommodity flow problems. *Encyclopedia of Optimization* 14:2354–2362.

[5] Bauschke H, Lewis A (2000) Dykstras algorithm with Bregman projections: A convergence proof. *Optimization* 48(4):409–427.

[6] Benamou JD, Brenier Y (2000) A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik* 84(3):375–393.

[7] Benamou JD, Brenier Y, Guittet K (2004) Numerical analysis of a multi-phasic mass transport problem. *Contemporary Mathematics* 353:1–18.

[8] Benamou JD, Carlier G, Cuturi M, Nenna L, Peyré G (2015) Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing* 37(2):A1111–A1138.

[9] Bertsekas DP, Tseng P (1988) Relaxation methods for minimum cost ordinary and generalized network flow problems. *Operations Research* 36(1):93–114.

[10] Bertsimas D, Patterson SS (2000) The traffic flow management rerouting problem in air traffic control: A dynamic network flow approach. *Transportation Science* 34(3):239–255.

[11] Boyd S, Vandenberghe L (2004) *Convex optimization* (Cambridge university press).

[12] Brockwell RW (2012) Notes on the control of the Liouville equation. *Control of partial differential equations*, 101–129 (Springer).

[13] Carlino D, Depinet M, Khandelwal P, Stone P (2012) Approximately orchestrated routing and transportation analyzer: Large-scale traffic simulation for autonomous vehicles. *2012 15th International IEEE Conference on Intelligent Transportation Systems*, 334–339 (IEEE).

[14] Chen Y, Georgiou TT, Pavon M (2016) On the relation between optimal transport and Schrödinger bridges: A stochastic control viewpoint. *Journal of Optimization Theory and Applications* 169(2):671–691.

[15] Chen Y, Georgiou TT, Pavon M (2016) Optimal Steering of a Linear Stochastic System to a Final Probability Distribution, Part I. *IEEE Transactions on Automatic Control* 61(5):1158–1169.

[16] Chen Y, Georgiou TT, Pavon M, Tannenbaum A (2016) Robust transport over networks. *IEEE Transactions on Automatic Control* 62(9):4675–4682.

[17] Chen Y, Georgiou TT, Pavon M, Tannenbaum A (2017) Efficient robust routing for single commodity network flows. *IEEE Transactions on Automatic Control* 63(7):2287–2294.

[18] Chen Y, Georgiou TT, Pavon M, Tannenbaum A (2019) Relaxed Schrödinger bridges and robust network routing. *IEEE Transactions on Control of Network Systems* 7(2):923–931.

[19] Chen Y, Georgiou TT, Tannenbaum A (2018) Vector-valued optimal mass transport. *SIAM Journal on Applied Mathematics* 78(3):1682–1696.

[20] Cutler MI (2013) Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems (NIPS)*, 2292–2300.

[21] Diestel R (2017) *Graph Theory* (Berlin, Heidelberg: Springer).

[22] Elvander F, Haasler I, Jakobson A, Karlsson J (2020) Multi-marginal optimal mass transport using partial information with applications in robust localization and sensor fusion. *Signal Processing* 171:107474.

[23] Farvolden JM, Powell WB, Lustig IJ (1993) A primal partitioning solution for the arc-chain formulation of a multicommodity network flow problem. *Operations Research* 41(4):669–693.

[24] Ford LR, Fulkerson DR (1958) Constructing maximal dynamic flows from static flows. *Operations research* 6(3):419–433.

[25] Ford LR, Fulkerson DR (1958) A suggested computation for maximal multi-commodity network flows. *Management Science* 5(1):97–101.

[26] Ford LR, Fulkerson DR (1962) *Flows in networks* (Princeton university press).

[27] Gangbo W, Świech A (1998) Optimal maps for the multidimensional Monge-Kantorovich problem. *Comm. on Pure and Appl. Math.: Courant Inst. of Math. Sci.* 51(1):23–45.
[28] Gendron B, Crainic TG, Frangioni A (1999) Multicommodity capacitated network design. Telecommunications network planning, 1–19 (Springer).

[29] Haasler I, Chen Y, Karlsson J (2020) Optimal steering of ensembles with origin-destination constraints. IEEE Control Systems Letters 5(3):881–886.

[30] Haasler I, Ringh A, Chen Y, Karlsson J (2019) Estimating ensemble flows on a hidden Markov chain. 2019 IEEE 58th Conference on Decision and Control (CDC), 1331–1338 (IEEE).

[31] Haasler I, Ringh A, Chen Y, Karlsson J (2020) Multi-marginal Optimal Transport with a Tree-structured cost and the Schrödinger Bridge Problem. Preprint. arXiv:2004.06909 29 pages.

[32] Haasler I, Singh R, Zhang Q, Karlsson J, Chen Y (2021) Multi-marginal optimal transport and probabilistic graphical models. IEEE Transactions on Information Theory In press. Preprint: arXiv:2006.14113.

[33] Haghani A, Oh SC (1996) Formulation and solution of a multi-commodity, multi-modal network flow model for disaster relief operations. Transportation Research Part A: Policy and Practice 30(3):231–250.

[34] Hall A, Hippler S, Skutella M (2007) Multicommodity flows over time: Efficient algorithms and complexity. Theoretical Computer Science 379(3):387–404.

[35] IBM: ILOG CPLEX (2019) Optimization Studio 12.10.0: CP Optimizer Online Documentation. URL https://www.ibm.com/docs/en/icos/12.10.0.

[36] Jones K, Lustig I, Farvolden J, Powell W (1993) Multicommodity network flows: The impact of formulation on decomposition. Mathematical Programming 62(1-3):95–117.

[37] Karlsson J, Ringh A (2017) Generalized Sinkhorn iterations for regularizing inverse problems using optimal mass transport. SIAM Journal on Imaging Sciences 10(4):1935–1962.

[38] Kennington J, Shalaby M (1977) An effective subgradient procedure for minimal cost multicommodity flow problems. Management Science 23(9):994–1004.

[39] Kennington JL (1978) A survey of linear cost multicommodity network flows. Operations Research 26(2):209–236.

[40] Khodayifar S (2019) Minimum cost multicommodity network flow problem in time-varying networks: by decomposition principle. Optimization Letters 1–18.

[41] Léonard C (2014) A survey of the Schrödinger problem and some of its connections with optimal transport. Discrete & Continuous Dynamical Systems - A 34(4):1533–1574.

[42] Levinson J, Askeland J, Becker J, Dolson J, Held D, Kammel S, Kolter J, Langer D, Pink O, Pratt V, et al. (2011) Towards fully autonomous driving: Systems and algorithms. 2011 IEEE Intelligent Vehicles Symposium (IV), 163–168 (IEEE).

[43] Lin T, Ho N, Chen X, Cuturi M, Jordan M (2020) Fixed-support Wasserstein barycenters: Computational hardness and fast algorithm. Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, eds., Advances in Neural Information Processing Systems, volume 33, 5368–5380 (Curran Associates, Inc.).

[44] Lin T, Ho N, Cuturi M, Jordan M (2019) On the complexity of approximating multimarginal optimal transport. Preprint. arXiv:1910.00152. 39 pages.

[45] Luo ZQ, Tseng P (1992) On the convergence of the coordinate descent method for convex differentiable minimization. Journal of Optimization Theory and Applications 72(1):7–35.

[46] McBride R (1998) Progress made in solving the multicommodity flow problem. SIAM Journal on Optimization 8(4):947–955.

[47] Nenna L (2016) Numerical methods for multi-marginal optimal transportation. Ph.D. thesis, PSL.

[48] Pasquale C, Sacone S, Siri S, Ferrara A (2019) Traffic control for freeway networks with sustainability-related objectives: Review and future challenges. Annual Reviews in Control 48:312–324.

[49] Pass B (2015) Multi-marginal optimal transport: theory and applications. ESAIM: Mathematical Modelling and Numerical Analysis 49(6):1771–1790.

[50] Pavon M, Ticozzi F (2010) Discrete-time classical and quantum Markovian evolutions: Maximum entropy problems on path space. Journal of Mathematical Physics 51(4):042104.
[51] Peyré G, Cuturi M (2019) Computational optimal transport. Foundations and Trends® in Machine Learning 11(5-6):355–607.

[52] Retvdri G, Biró J, Cinkler T (2004) A novel lagrangian-relaxation to the minimum cost multicommodity flow problem and its application to ospf traffic engineering. Proceedings. ISCC 2004. Ninth International Symposium on Computers And Communications (IEEE Cat. No. 04TH8769), volume 2, 957–962 (IEEE).

[53] Rüschendorf L (1995) Optimal solutions of multivariate coupling problems. Applicationes Mathematicae 23(3):325–338.

[54] Rüschendorf L, Uckelmann L (2002) On the n-coupling problem. Journal of Multivariate Analysis 81(2):242–258.

[55] Tomlin JA (1966) Minimum-cost multicommodity network flows. Operations Research 14(1):45–51.

[56] Tseng P (1990) Dual ascent methods for problems with strictly convex costs and linear constraints: A unified approach. SIAM Journal on Control and Optimization 28(1):214–242.

[57] Villani C (2008) Optimal transport: Old and new (Berlin Heidelberg: Springer).

[58] Wang IL (2018) Multicommodity network flows: A survey, Part I: Applications and Formulations. International Journal of Operations Research 15(4):145–153.

[59] Yamada T (1996) A network flow approach to a city emergency evacuation planning. International Journal of Systems Science 27(10):931–936.