SIAM J. APPL. MATH.
Vol. 80, No. 1, pp. 599–619

INVERSE OPTIMAL TRANSPORT*

ANDREW M. STUART† AND MARIE-THERESE WOLFRAM‡

Abstract. Discrete optimal transportation problems arise in various contexts in engineering, the sciences, and the social sciences. Often the underlying cost criterion is unknown, or only partly known, and the observed optimal solutions are corrupted by noise. In this paper we propose a systematic approach to infer unknown costs from noisy observations of optimal transportation plans. The algorithm requires only the ability to solve the forward optimal transport problem, which is a linear program, and to generate random numbers. It has a Bayesian interpretation and may also be viewed as a form of stochastic optimization. We illustrate the developed methodologies using the example of international migration flows. Reported migration flow data captures (noisily) the number of individuals moving from one country to another in a given period of time. It can be interpreted as a noisy observation of an optimal transportation map, with costs related to the geographical position of countries. We use a graph-based formulation of the problem, with countries at the nodes of graphs and nonzero weighted adjacencies only on edges between countries which share a border. We use the proposed algorithm to estimate the weights, which represent cost of transition, and to quantify uncertainty in these weights.

Key words. optimal transport, international migration flows, linear program, parameter estimation, Bayesian inversion

AMS subject classifications. 90C08, 62F15, 65K10

DOI. 10.1137/19M1261122

1. Introduction.

1.1. Background. There are many problems in engineering, the sciences, and the social sciences, in which an input is transformed into output in an optimal way according to a cost criterion. We are interested in problems where the transformation from input to output is known, and the objective is to infer the cost criterion which drives this transformation. Our primary motivation is optimal transport (OT) problems in which the transport plan is known but the cost is not. More generally linear programs in which the solution is known, but the cost function and constraints are to be determined, fall into the category of problems to which the methodology introduced in this paper applies. We illustrate the type of problem of interest by means of an example.

Example: International migration. Quantifying migration flows between countries is essential to understand contemporary migration flow patterns. Typically two types of migration statistics are collected—flow and stock data. Migration stock data states the number of foreign born individuals present in a country at a given time and is usually based on population censuses. Stock data is available for almost all countries in the world. Migration flow data captures the number of migrants entering and leaving (inflow and outflow, respectively) a country over the course of a specific period, such as one year; see [1]. It is collected by most developed countries, but no

*Received by the editors May 10, 2019; accepted for publication (in revised form) December 4, 2019; published electronically February 25, 2020.
https://doi.org/10.1137/19M1261122
Funding: The work of the first author was supported by U.S. National Science Foundation (NSF) under grant DMS 1818977 and by AFOSR grant FA9550-17-1-0185. The work of the second author was partially supported by the Royal Society International Exchanges grant IE 161662.
†California Institute of Technology, Pasadena, CA 91125 (astuart@caltech.edu).
‡University of Warwick, Coventry CV4 7AL, UK, and RICAM, Austrian Academy of Sciences, 4040 Linz, Australia (m.wolfram@warwick.ac.uk).

TABLE 1.1
Harmonized migration flow statistics for the period 2002–2007*; see* [9].

From		**To** CZ	DE	DK	LU	NL	PL
CZ	R	0	9,218	262	4	511	45
	S	0	560	24	3	81	583
DE	R	1,362	0	4,001	454	9,182	2,876
	S	8,104	0	3,095	1,686	9,293	100,827
DK	R	46	2,687	0	11	475	34
	S	179	2,612	0	1,387	602	833
LU	R	2	2,282	162	0	161	5
	S	13	911	99	0	97	23
NL	R	255	13,681	864	27	0	163
	S	298	10,493	533	191	0	1,020
PL	R	1,608	136,927	2,436	19	5,744	0
	S	63	14,417	111	23	577	0
Tot:	S	3,273	164,795	7,725	515	16,073	3,123
	R	8,657	28,993	3,862	2,041	10,650	103,286

international standards are defined. For example, the time of residence after which a person counts as an international migrant varies from country to country. Because of the different definitions and data collection methods, these statistics can be hard to compare. International agencies, such as the United Nations Statistics Division or the Statistical Office of the European Union (Eurostat), publish annual migration flow estimates. These estimates are often based on Poisson or Bayesian linear regression. For more information about the estimation of migration flows using flow or stock statistics we refer to [2, 4, 20, 21]. For the purposes of this paper the main issue to appreciate is that migration data is available but should be viewed as noisy.

Flow data is typically presented in an origin-destination matrix, in which the (i, j)th off-diagonal entry contains the number of people moving from country i to country j in a given period of time. This origin-destination data can be reported by both the sending (S) and the receiving (R) countries. Hence two migration flow tables are available, often desegregated by sex and age groups. Table 1.1 shows harmonized data, which was preprocessed to improve comparability, reported by 6 European countries for the period 2002–2007. The numbers of the sending and receiving countries vary significantly. For example, Germany reported that $136,927$ people immigrated from Poland, while Poland reported $14,417$ individuals who left for Germany. These very different numbers naturally raise the question of the true migration flows. In many settings it is natural to place greater weight on receiving data rather than departure data. But even this data is not subject to uniform standards, and therefore providing reliable estimates and quantifying uncertainty is of great interest.

We interpret the reported origin-destination data maps (when appropriately normalized) as a noisy estimate of a transport plan arising from an OT problem with unknown cost. It is then natural to try and infer the transportation cost, as it carries information about the migration process. □

The preceding example serves as motivation, and we will come back to it throughout this paper. However, we reemphasize that the proposed identification methodologies that we introduce in this paper can be used for general inverse OT and linear programming problems; further examples will serve to illustrate this fact.

1.2. Literature review. OT originates with the French mathematician Gaspard Monge who, in 1781, investigated the problem of finding the most cost-effective

way to move a pile of sand to fill a hole of the same volume. Kantorovich introduced the modern (relaxed) formulation of the problem, in which mass can be split, in 1942. In more mathematical terms Kantorovich considered the following setup: given two positive measures (of equal mass) and a cost function, find the transportation map that moves one measure to the other minimizing the transport cost. The corresponding infimum induces a distance between these two measures—the so-called Wasserstein distance. The Wasserstein distance plays an important role in probability theory, partial differential equations, and many other fields in applied mathematics [27, 30]. Furthermore the techniques and methodologies developed in OT have found application in a variety of scientific disciplines including data science, economics, imaging, and meteorology [13].

With the spread and application of OT into different scientific disciplines the interest in computational methodologies has increased. Commonly used numerical methods broadly speaking fall into two categories: linear programming [8] and methods specific to the structure of OT. Linear programs are classic problems which have been extensively studied in the field of optimization and operations research. Many computational methodologies have been developed, such as the famous simplex algorithm (and its many variants), the Hungarian algorithm, and the auction algorithm. All these methods work well for small to medium sized problems but are too slow in modern applications such as imaging or supply chain management. Recently a significant speed up, of linear programming, was achieved by considering a regularized OT problem, leading to the Sinkhorn algorithm (or variants thereof) in which an additional entropic regularization term is added to the objective function; this allows efficient computation of the corresponding minimizer and induces a trade-off between fidelity to the original problem and computational speed. This family of efficient algorithms resulted in the rapid advancement of computational OT in recent years, especially in the context of imaging and data science; see [7, 19, 22].

Inverse problems for linear programming received considerable interest in the engineering literature. The paper [3], building on earlier work in [32], studies the problem by seeking a cost function nearest to a given one in ℓ^p for which the given solution is an optimal linear program; this problem is itself a linear program in the case $p = 1$. The formulation of an inverse problem for linear programming in [10] took a slightly more general perspective, as it does not assume that the given data necessarily arises as the solution of a linear program, and rather seeks to minimize the distance to the solution set of a linear program. Recent application of the inverse problem for linear programming may be found in [26], for example. The most closely related work to this paper is the recent publication by Li et al. (see [16]) in which the authors minimize the log likelihood function to estimate the underlying cost. These papers on inverse linear programming are foundational and have opened up a great deal of subsequent research. However, the methods used in them do not account in a systematic way for noise in the data provided and for the incorporation of prior information. We address these issues by adopting a Bayesian formulation of the inverse problem for linear programming, concentrating on OT in particular; the ideas are readily generalized to inverse linear programming in general. The Bayesian approach not only allows for the quantification of uncertainty but also leads to algorithms which may be viewed as stochastic methods for exploring the space of solutions, constrained by the observed data. An overview of the computational state of the art for Bayesian inversion may be found in [15]. The specific methods that we introduce have the desirable feature that they require only solution of the forward OT problem and the ability to generate random numbers.

1.3. Our contribution. Our contributions to the subject of inverse problems within linear programming are as follows.

- We formulate inverse OT problems in a Bayesian framework.
- We provide a computational framework for solving inverse OT problems in an efficient fashion.
- We give a systematic discussion of identifiability issues arising for finite dimensional inverse OT.
- We introduce graph-based cost functions for OT, using graph-shortest paths in an integral way.
- Graph-based OT has considerable potential for application, and we introduce a new way of studying migration flow data using inverse OT in the graph-based setting.

We emphasize that, while the graph-based formulation of cost corresponds to a rather specific way of designing cost functions for discrete linear programs, the framework and algorithms developed in this paper apply quite generally to inverse linear programming and hence to OT in general. We develop the methodology in general, using graph-based migration flow as a primary illustrative example. In section 2 we define OT as a linear program, describe the cost criteria considered, and formulate inverse OT in a Bayesian setting; in this section we discuss the identifiability issue for finite dimensional inverse OT. Section 3 presents algorithms for the forward and inverse OT problem, and section 4 contains numerical results.

We will use the following notation throughout this manuscript. Let $|\cdot|$ and $\langle\cdot,\cdot\rangle$ denote the Euclidean norm and inner-product on $\mathbb{R}^n$ and the Frobenius norm and inner-product on $\mathbb{R}^{n\times n}$. The spaces of probability matrices, probability vectors, and probability matrices with specified marginals are defined as

$$\mathcal{P}_{n\times n} = \Big\{B \in \mathbb{R}^{n\times n} : B_{ij} \geq 0, \sum_{i,j=1}^{n} B_{ij} = 1\Big\}, \quad \mathcal{P}_n = \Big\{u \in \mathbb{R}^n : u_j \geq 0, \sum_{j=1}^{n} u_j = 1\Big\},$$

$$\mathcal{S}_{\mathfrak{p},\mathfrak{q}} = \Big\{B \in \mathcal{P}_{n\times n} : B\mathbf{1} = \mathfrak{p}, B^T\mathbf{1} = \mathfrak{q} \text{ for } \mathfrak{p},\mathfrak{q} \in \mathcal{P}_n\Big\}, \text{ where } \mathbf{1} = (1,\dots,1)^T \in \mathbb{R}^n.$$

2. Inverse OT. In this section we introduce the forward OT problem and discuss specific cost criteria, before formulating the respective inverse OT problem in the Bayesian framework.

2.1. Forward problem. We consider two discrete probability vectors $\mathfrak{q} \in \mathcal{P}_n$ and $\mathfrak{p} \in \mathcal{P}_n$ and a given cost $C \in \mathcal{P}_{n\times n}$. Then the OT problem corresponds to finding a map transporting $\mathfrak{p}$ to $\mathfrak{q}$ at minimal cost. Note that in OT the cost matrix has nonnegative entries, which can be normalized to be an element of $\mathcal{P}_{n\times n}$ without loss of generality. The respective forward OT problem is to find

$$T^* \in \operatorname{argmin}_{T\in\mathcal{S}_{\mathfrak{p},\mathfrak{q}}} \langle C, T\rangle. \tag{2.1}$$

Problem (2.1) falls into the more general class of linear programs. Linear programs (and their many variants) arise in various specific settings—such as the earth mover's distance [25] or cost network flows [5]—in different scientific communities. The problem (2.1) has, by virtue of being a specific class of linear programs, at least one solution; this solution lies on the boundary of the feasible set of solutions (defined by the equality constraints). If the solution is unique, then we define mapping $\mathcal{F} : \mathcal{P}_n \times \mathcal{P}_n \times \mathcal{P}_{n\times n} \to \mathcal{P}_{n\times n}$ by

$$T^* = \mathcal{F}(\mathfrak{p},\mathfrak{q},C). \tag{2.2}$$

In the nonunique setting we define $\mathcal{F}(\mathfrak{p}, \mathfrak{q}, C)$ to be a unique element determined by running a specific nonrandom algorithm for the linear program to termination, started at a specific initial guess.

We now consider (2.1) regularized by the addition of the discrete entropy, an approach popularized in [7, 19] and which has led to considerable analytical and computational developments. Besides the advantageous analytical and computational aspects, the proposed regularization term can be interpreted as an inherent uncertainty in the cost due to the heterogeneity of agents. Galichon and Salanie [13] propose the same regularization in the context of marriage market and matching problems. The resulting problem is

$$H(T) = -\langle T, \log(T)\rangle + \mathrm{Tr}(T) = -\sum_{i,j=1}^{n} T_{i,j}(\log T_{i,j} - 1), \tag{2.3}$$

where the matrix logarithm operation is applied elementwise. Then

$$T_\epsilon^* = \mathrm{argmin}_{T \in \mathcal{S}_{\mathfrak{p},\mathfrak{q}}} \Big(\langle C, T\rangle + \epsilon H(T) \Big). \tag{2.4}$$

This problem has a unique minimizer T_ϵ^*, since $H(T)$ is strongly convex. Following our previous notation we define the corresponding mapping by $\mathcal{F}_\epsilon : \mathcal{P}_n \times \mathcal{P}_n \times \mathcal{P}_n \to \mathcal{P}_{n\times n}$

$$T_\epsilon^* = \mathcal{F}_\epsilon(\mathfrak{p}, \mathfrak{q}, C). \tag{2.5}$$

It is, in contrast to the optimal solution of (2.1), not sparse. It is known that solutions to (2.4) converge to minimizers of (2.1) as $\epsilon \to 0$. Determining the rate of convergence is still an open problem. The special structure of this regularized problem can be used to construct efficient splitting algorithms. These methods are based on the equivalent formulation of finding the projection of the joint coupling with respect to the Kullback–Leibler divergence

$$D_{KL}(T\|K) := \langle T, \log(T/K)\rangle - \mathrm{Tr}(T) + \mathrm{Tr}(K) = \sum_{i,j=1}^{n} T_{i,j} \log \frac{T_{i,j}}{K_{i,j}} - T_{i,j} + K_{i,j},$$

where the matrix logarithm and division operations are applied elementwise and K is the Gibbs kernel

$$K_{i,j} = \exp^{-\frac{C_{i,j}}{\epsilon}}. \tag{2.6}$$

In particular

$$T_\epsilon^* = \mathrm{argmin}_{T \in \mathcal{S}_{\mathfrak{p},\mathfrak{q}}} D_{KL}(T\|K). \tag{2.7}$$

The Kullback–Leibler divergence can be computed extremely efficiently using proximal methods, yielding, for example, the celebrated Sinkhorn algorithm. We will briefly outline the underlying ideas in section 3.1.

2.2. Cost criteria. Problems (2.1) and (2.4) are formulated for general cost matrices C—the specific structure of C depends on the application considered. We will investigate the behavior of the proposed methodologies for C being

(i) Toeplitz
(ii) nonsymmetric
(iii) determined by an underlying graph structure.

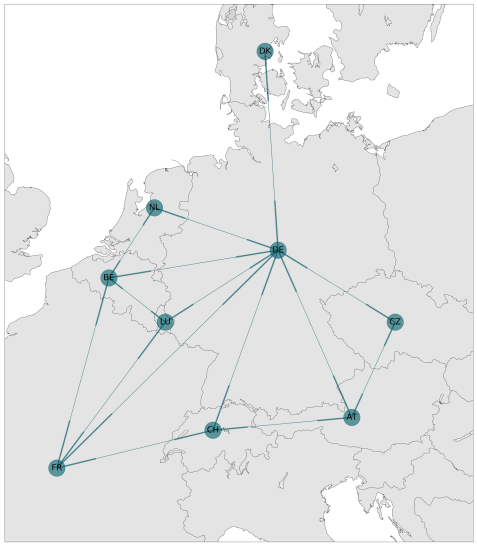


FIG. 2.1. *Network defined by the European countries used in our example.*

We assume that all individuals move; hence $T_{ii} = 0$ for all $i = 1, \dots n$ in all three cases. Therefore staying is penalized by setting

$$C_{ii} = \bar{C} \gg 1 \quad \text{for all } i = 1, \dots n. \tag{2.8}$$

If C is Toeplitz the cost depends on the difference between indices and C has $2n-3$ degrees of freedom. Case (ii) corresponds to general nonsymmetric transportation cost, which in the context of migration flows could include factors such as sharing the same language, the ratio of the gross national income per capita, or their European Union membership. In case (iii) we assume that costs are related to an underlying discrete structure. In the context of migration flows the geographical position of countries defines an underlying graph with edges only between countries which share a border; see Figure 2.1. We assume that the total transportation cost corresponds to the sum of the individual costs of moving from one country to another along edges of the graph. In defining cost this way we are implicitly assuming that, between the European countries studied here, migration is primarily via land. This resulting discrete underlying structure, which relates the cost matrix to a directed graph representing the migration network between countries, is detailed in the following.

Let (V, E) be a directed graph with $n = |V|$ vertices and a (possibly nonsymmetric) weighted adjacency matrix $A \in \mathbb{R}^{n\times n}$. We can then define a cost matrix $W \in \mathbb{R}^{n\times n}$ whose (i,j)th entry $W_{i,j}$ is the shortest path cost of moving from vertex i to j according to the weighted adjacency matrix A. Let m be the number of nonzero entries of A and $f \in \mathbb{R}^m$ the vector defining the nonzero entries. Then we may define a mapping $\mathcal{E}$ such that $W = \mathcal{E}(f)$. This $W \in \mathbb{R}^{n\times n}$ can then be normalized to give a $C \in \mathcal{P}_{n\times n}$, and we may define the solution of the resulting OT problem via (2.2). For this graph-based cost the solution of the OT problem may be viewed as a function of $\mathfrak{p}, \mathfrak{q}$, and f. The minimal cost of moving between vertices of a graph can be computed using Dijkstra's algorithm, recalled in section 3.1 below.

We define a similar mapping in the case of Toeplitz cost. Here the respective cost matrix C has $2n-2$ free entries, before normalization to a probability vector and

recalling that we fix the diagonal to penalize not moving, and so we define a mapping $\mathcal{E} : f \in \mathbb{R}^{2n-2} \to \mathbb{R}_+^{n\times n}$; normalization then gives $C = \mathcal{M}_{n\times n}(\mathcal{E}(f))$.

2.3. Inverse problem and identifiability. The inverse OT problem is to find $\mathfrak{p}, \mathfrak{q}$, and C from the solution T to the OT problem (2.1) or its regularized counterpart (2.4). We tackle this problem by introducing a space of componentwise positive and real-valued latent variables u, v, W or u, v, f which map to the unknowns $\mathfrak{p}, \mathfrak{q} \in \mathcal{P}_n$ and $C \in \mathcal{P}_{n\times n}$. It is easier, and more natural, to specify priors in terms of these real-valued latent variables. To this end we introduce mappings from $\mathbb{R}_+^n$ into $\mathcal{P}_n$ and from $\mathbb{R}_+^{n\times n}$ into $\mathcal{P}_{n\times n}$ as follows: $\mathcal{M}_n : \mathbb{R}_+^n \mapsto \mathcal{P}_n$ is defined by

$$\mathcal{M}_n(u)_j = u_j / \left(\sum_{\ell=1}^{n} u_\ell \right),$$

and $\mathcal{M}_{n\times n} : \mathbb{R}_+^{n\times n} \mapsto \mathcal{P}_{n\times n}$ is defined by

$$\mathcal{M}_{n\times n}(W)_{i,j} = W_{i,j} / \left(\sum_{k,\ell=1}^{n} W_{k,\ell} \right).$$

Note that $\mathcal{M}_n(\lambda u) = \mathcal{M}_n(u)$ for all $\lambda \in \mathbb{R}$; the same holds for $\mathcal{M}_{n\times n}$. Then the forward problem (2.2) can be written as

$$T^* = \mathcal{G}(u, v, W) := \mathcal{F}\big(\mathcal{M}_n(u), \mathcal{M}_n(v), \mathcal{M}_{n\times n}(W)\big), \tag{2.9}$$

or, in the case of graph-based cost or Toepliz cost, we have

$$T^* = \mathcal{G}(u, v, f) := \mathcal{F}\big(\mathcal{M}_n(u), \mathcal{M}_n(v), \mathcal{M}_{n\times n}(\mathcal{E}(f))\big). \tag{2.10}$$

This is readily generalized to the use of regularized OT as the forward model, simply replacing $\mathcal{F}$ by $\mathcal{F}_\epsilon$.

We wish to invert the map $\mathcal{G}$, given noisy observations of T^*. Such problems are in general ill-posed; hence suitably regularized versions have to be considered. Different approaches can be found in the literature—we focus on the Bayesian framework, which allows us to estimate the posterior distribution of u, v, and W (or f).

Depending on the structure of the cost matrix the inverse problem related to (2.9) or (2.10) can be over- or underdetermined. We recall that in case of Toeplitz cost the matrix C has $2n-3$ degrees of freedom. Then we have n^2-1 equations for $4n-5$ unknowns (taking into account the normalization of u, v and W). Hence the inverse problem is overdetermined for $n > 2$. If C is a general cost matrix with a set penalty on the diagonal, that is, case (ii), the cost matrix has $n^2 - n$ degrees of freedom. In total we have $n^2 + n - 3$ unknowns, and therefore the problem is underdetermined for $n > 2$. For graph-based cost (case (iii)) the matrix C has m degrees of freedom, and therefore the problem is only underdetermined if

$$2n + m - 3 > n^2 - 1. \tag{2.11}$$

Recall that n denotes the dimension of the space on which the marginals live. In summary the situation in which the entire cost is unknown, and no structure is placed on it, is generically not-identifiable if $n > 2$. Cost structures which impose a linear number of unknowns are generically identifiable when n is large enough; indeed the graph-based example is generically identifiable when n is large enough, provided that the number of edges grows sublinearly with n.

2.4. Bayesian formulation of inverse problem. We define a Bayesian formulation of the inverse problem, working in the case where u, v, W are the unknowns; the extension to u, v, f as unknowns is similar.

2.4.1. Prior. Let $\mathbb{P}(u, v, W)$ denote the prior information concerning u, v and W. To be concrete we assume throughout this paper that u, v, and W have independently and identically distributed (i.i.d.) entries uniformly distributed in $[0, 1]$, and denote the set of vectors and matrices which satisfy this componentwise constraint by U. In view of the scale invariance of $\mathcal{M}_n(\cdot)$ the choice of unit interval $[0, 1]$ is immaterial; any bounded interval $[0, \lambda]$ would deliver an identical posterior on u, v, W. We emphasize that we have also experimented with different priors, such as Gaussians, obtaining qualitatively similar results.[1]

2.4.2. Likelihood. We assume that the observed transport maps T are corrupted by noise:

$$T = \mathcal{G}(u, v, W) + \eta, \tag{2.12}$$

where η is a mean zero noise. To be concrete we assume throughout this paper that η is a Gaussian random matrix with i.i.d. entries of variance σ^2; other noise models are readily accommodated into the methodology proposed here—they simply result in different functions Φ. Since the algorithms used here require only evaluation of Φ they are extended to different noise models very easily.

The conditional probability distribution of T, *given* (u, v, W), that is, the variable $T \mid (u, v, W)$, defines the likelihood. This is given by

$$\mathbb{P}(T|u, v, W) \propto \exp\big(-\Phi(u, v, W; T)\big), \tag{2.13}$$

where the misfit Φ is defined, under our assumptions on η, by

$$\Phi(u, v, W; T) = \frac{1}{2\sigma^2}|T - \mathcal{G}(u, v, W))|^2. \tag{2.14}$$

2.4.3. Posterior. Using Bayes' formula

$$\mathbb{P}(u, v, W \mid T) = \frac{1}{\mathbb{P}(T)}\mathbb{P}(T \mid u, v, W)\mathbb{P}(u, v, W) \tag{2.15}$$

and the preceding prior and likelihood constructions, the posterior distribution of u, v, and W given the noisy observed transport map T is defined by

$$\mathbb{P}(u, v, W|T) = \frac{1}{Z}\exp\Big(-\frac{1}{2\sigma^2}|T - \mathcal{G}(u, v, W)|^2\Big)\mathbf{1}_{\mathsf{U}}(u, v, W) \tag{2.16}$$

with a normalization constant

$$Z = \int_{\mathsf{U}} \exp\Big(-\frac{1}{2\sigma^2}|T - \mathcal{G}(u, v, W)|^2\Big)\,du\,dv\,dW\,.$$

We can either sample from the posterior (2.16) (which corresponds to the full Bayesian approach) or maximize the posterior probability (2.16), which leads to the optimization problem of minimizing $\Phi(u, v, W; T)$ over U. The first approach allows us to quantify uncertainty in the estimates of u, v and W, the latter gives a single estimate. We discuss how to sample from the posterior, using a random walk Metropolis (RwM) method, in section 3.2. This method may also be viewed as a form of stochastic exploration of the solution space, constrained by observed data.

[1] In this case we used the preconditioned Crank–Nicolson method [6] rather than the random walk Metropolis method used in the experiments reported here.

3. Algorithms for inversion. In the following we present the numerical methods used in the computational experiments in section 4. Since the proposed Bayesian framework requires the solution of an OT problem (2.1) (or its regularized version (2.4)) in every iteration of the sampling algorithm, computational efficiency is essential. We start by presenting the solvers for the forward OT problem followed by the Markov chain Monte Carlo methods used to sample from the posterior.

3.1. Computational OT. Numerical methods for linear programming go back to the seminal works of Dantzig on the simplex method; see [8]. Solutions to the linear program (2.1) lie on the boundary of the feasible polytope, which is defined by the constraints. The simplex method iterates over the vertices of this polytope to find the optimal solution; see [18]. The method works well in practice; however, examples in which the performance scales exponentially with the dimension of the problem can be constructed. Different approaches to speed up computations have been proposed: for example, network simplex algorithms are based on the fact that specific linear programs can be formulated as minimization problems on graphs. The particular structure of the underlying graph can be used to speed up the simplex method significantly. Further information on computational methods for linear programming can be found in [9].

More recently computational techniques, which are based on the regularized OT problem (2.4), have been proposed in the literature. These methods are extremely efficient, since they are based on the formulation of the OT problem in terms of the Kullback–Leibler divergence (2.7). Its minimizer is given by

$$T_{i,j} = a_i K_{i,j} b_j.$$

Here K is the Gibbs kernel (2.6), and the vectors a and b satisfy the mass constraint

$$\operatorname{diag}(a) K \mathbf{1} = \mathfrak{p} \text{ and } \operatorname{diag}(b) K^T \mathbf{1} = \mathfrak{q}. \tag{3.1}$$

This mass constraint can be enforced iteratively via

$$a^{(l+1)} = \frac{\mathfrak{p}}{K b^{(l)}} \text{ and } b^{(l+1)} = \frac{\mathfrak{q}}{K a^{(l+1)}}. \tag{3.2}$$

This splitting, known as Sinkhorn's algorithm, is very efficient as it involves matrix vector multiplications only. Since the entropic regularization term (2.3) introduces blurring in the otherwise sparse solution, one is interested in keeping ϵ as small as possible. Since the convergence of Sinkhorn's algorithm (3.2) deteriorates as $\epsilon \to 0$, it is important to keep a balance between regularization and computational stability. In practice small values of ϵ lead to diverging scaling factors in (3.2) and subsequent numerical instabilities. These problems can often be remedied using suitable scalings; see [28].

If the transportation costs depend on an underlying discrete structure, such as for our graph-based migration problem, then the computational burden of computing this cost must be taken into consideration. For our example the total transportation cost corresponds to the sum of edge weights when between vertices traversed on the shortest path. Note that the transportation costs are not necessarily the same in both directions since we consider directed graphs. We use Dijkstra's algorithm to compute the shortest path from one node to all others in the graph; see [11]. Dijkstra's algorithm is based on continuous updates of the shortest distance to a starting point and excludes longer distances in updates. It is the graph-based methodology that underpins the fast marching method to solve the eikonal equation [29].

3.2. Markov chain Monte Carlo and optimization. We propose the use of Markov chain Monte Carlo (MCMC) methods to sample from the posterior distribution (2.16). For the user interested simply in optimization the algorithm we propose may be viewed as a stochastic optimization method to reduce the model-data misfit. MCMC methods originated with the seminal paper [17] in which what is now termed the RwM algorithm was introduced for a specific high dimensional integral required in statistical physics. In our context the key desirable feature of the method is that it requires only solution of the forward OT (or regularized OT) problem, together with the generation of random numbers. Given a current (approximate) sample from the posterior distribution, a new sample is proposed by adding a mean zero Gaussian to the current one. This is rejected if the resulting new state leaves U, and otherwise accepted with a probability designed to preserve detailed balance with respect to the posterior. The covariance of the Gaussian is an important tuning parameter: intuitively it should be chosen such that the acceptance rate is close to neither 0 nor 1, as either of these limits leads to successive iterates which are highly correlated. The optimal scaling of RwM algorithms for different target densities has been investigated in [23, 24]; although the theory developed there applies in rather restricted scenarios, widespread experience and a variety of theories demonstrate that the work leads to useful rule-of-thumb for tuning acceptance probabilities within the RwM algorithm [31], arguably because it leads to average acceptance probabilities that stay away from 0 or 1.

In 1970 Hastings introduced a wide class of MCMC methods, now known as Metropolis–Hastings algorithms [14], and in principle this provides a wide range of variants on RwM that may be used for our Bayesian formulation of inverse OT. A popular variation of MCMC that we have found useful in the inverse OT setting is Gibbs sampling. In high dimensional spaces it can be hard to design proposals which are accepted with a reasonable acceptance probability, and the idea of fixing subsets of the variables, and proposing moves in the remainder, is natural. The Gibbs sampler allows this to be achieved in a statistically consistent fashion. At each iteration one (or several) components of the unknown parameter is updated by sampling from its full conditional probability distribution and cycling through all the variables. The method may be relaxed to allow a RwM step from the conditional probability distribution, rather than a full sample. The corresponding RwM-within-Gibbs method is outlined in Algorithm 3.1. In this algorithm we consecutively update u, v, and W (or f). We generate proposals for each variable, which we accept or reject. Note that in general, for all the methods described here, any proposal which decreases the value of Φ and remains in U is accepted with probability one. Thus Algorithm 3.1 may be viewed as an optimization method which induces a stochastic gradient; the numerics will demonstrate that this acts to minimize the misfit.

4. Numerical results. In this section we demonstrate the behavior of MCMC methods for inverse OT, and Algorithm 3.1 in particular. We start by presenting results for the migration flow example introduced at the beginning and use it as a "proof-of-concept" for the proposed framework. We then continue with systematic numerical investigation to study the identifiability of the cost matrix in a variety of scenarios, as well as discussing the behavior of the proposed methodology. We focus on the three cost criteria discussed in section 2.2: Toeplitz cost (i), nonsymmetric cost (ii), and graph-based cost (iii). We use the following functions implemented in the Python Optimal Transport library [12] to solve the linear program (2.1) as well as its regularized version (2.4):

Algorithm 3.1 Random walk Metropolis within Gibbs.

Initialize (u^0, v^0, W^0) **for** $k \geq 0$ **do**

Generate $\xi_u \sim \mathcal{N}(0, \delta_u^2)$ and **propose** new value $x = u^k + \xi_u,\ y = v^k,\ Z = W^k$
if $(x, y, Z) \notin \mathsf{U}$ **then** $(u^{k+1}, v^{k+1}, W^{k+1}) = (u^k, v^k, W^k)$
else

$$(u^{k+1}, v^{k+1}, W^{k+1}) = \begin{cases} (x, y, Z) & \text{with probability } a((u^k, v^k, W^k), (x, y, Z)) \\ (u^k, v^k, W^k) & \text{otherwise} \end{cases}$$

Generate $\xi \sim \mathcal{N}(0, \delta_v^2)$ and **propose** new value $y = v^k + \xi_v,\ x = u^k,\ Z = W^k$
if $(x, y, Z) \notin \mathsf{U}$ **then** $(u^{k+1}, v^{k+1}, W^{k+1}) = (u^k, v^k, W^k)$
else

$$(u^{k+1}, v^{k+1}, W^{k+1}) = \begin{cases} (x, y, Z) & \text{with probability } a((u^k, v^k, W^k), (x, y, Z)) \\ (u^k, v^k, W^k) & \text{otherwise} \end{cases}$$

Generate $\xi_W \sim \mathcal{N}(0, \delta_W^2)$ and **propose** new value $Z = W^k + \xi_W,\ x^k = u^k,\ y = v^k$ **if** $(x, y, Z) \notin \mathsf{U}$ **then** $(u^{k+1}, v^{k+1}, W^{k+1}) = (u^k, v^k, W^k)$
else

$$(u^{k+1}, v^{k+1}, W^{k+1}) = \begin{cases} (x, y, Z) & \text{with probability } a((u^k, v^k, W^k), (x, y, Z)) \\ (u^k, v^k, W^k) & \text{otherwise} \end{cases}$$

Here

$$a((u, v, W),(x, y, Z)) = \min\Big\{1, \exp\big(\frac{1}{2\sigma^2}|T - \mathcal{G}(u, v, W)|^2 - \frac{1}{2\sigma^2}|T - \mathcal{G}(x, y, Z)|^2\big)\Big\}.$$

- *emd*–this solver for linear programs is based on the respective network OT flow formulation of the problem and was introduced in [5].
- *sinkhorn*—implements the Sinkhorn-Knopp scaling algorithm to solve the regularized OT problem (2.4) as proposed in [7].

We test the proposed methodologies using simulated data as well as real migration data. In making simulated data we compute the OT maps T for a given set of vectors $\mathfrak{p}$, $\mathfrak{q}$, and f and add i.i.d. Gaussian noise with mean 0 and variance σ^2; see (2.12). Note that the resulting perturbed map T^* may have negative entries and is not an element of $\mathcal{P}_{n\times n}$. Therefore we set all negative entries to zero and normalize it to ensure that it is an admissible solution.

We illustrate the performance of the methodologies with plots of the running means and the respective posterior distributions. All posterior distributions are calculated after 500,000 RwM iterations with a burn-in of 300,000. The performed numerical experiments indicate that this number is sufficient for the convergence of MCMC. Note that we always plot the scaled vectors and matrices (unless stated otherwise). The penalty $\bar{C}$ in (2.8) is set to 10. Numerical simulations show that its absolute magnitude does not influence the posterior distributions significantly once above a certain level. On the other hand numerical stability favors not choosing the penalty too large. The resulting compromise led us to the value chosen.

4.1. European migration flows. We start by presenting estimates for the European network shown in Figure 2.1. We recall that vertices represent countries and

that edges connect countries sharing a border. The weights of these edges correspond to the cost of moving from one country to another. The network shown in Figure 2.1 consists of $n = 9$ countries, which are connected by $m = 30$ edges. We use the estimated transportation map reported in [21] and assume that the noise level is 4%. The variance for the proposals is set to $\delta_u^2 = \delta_v^2 = \delta_W^2 = 0.04$. We perform two runs of the RwM-within-Gibbs algorithm, using the exact solver in the first and Sinkhorn's algorithm with $\epsilon = 0.04$ in the second. The acceptance rate of the exact solver is 50.8% ((53.8%, 53.7%, 44.9%) for the different components u, v, and f); for Sinkhorn we have 82.9% (84.7%, 85.5%, 78.6%). The running averages of three components of u, v, and f are shown in Figure 4.1 and the corresponding posterior distributions in Figure 4.2. We observe that both runs give comparable results; however, the misfit for Sinkhorn is smaller; see Figure 4.3. This difference might be explained by the fact that we underestimate the noise level σ or that the actual transportation maps look more like solutions of regularized OT problems than the OT problem itself.

4.2. Graph-based cost. Next we investigate the behavior of the proposed methodologies for graph-based cost more thoroughly. We will see the following:

- The identification of u, v, and f is robust with respect to the sampling variances; see Figure 4.4.
- The posterior estimates are consistent using different solvers; see Figure 4.5.

These results are obtained using noisy transportation maps T^* for a graph connecting $n = 5$ nodes with $m = 12$ edges. In doing so we solve problem (2.1) for given vectors $\mathfrak{p}$, $\mathfrak{q}$, and f and add 4% noise. Note that this inverse problem is overdetermined since $2 \cdot 5 + 12 - 3 < 5^2 - 1$.

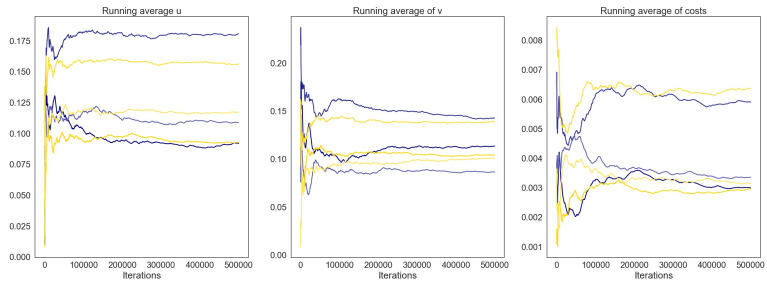
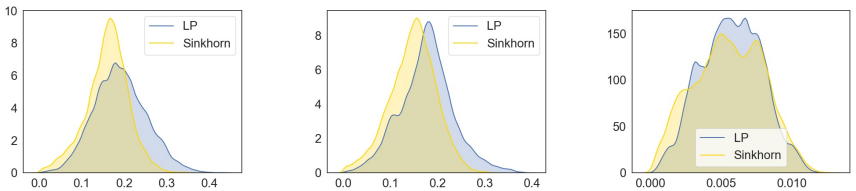


FIG. 4.1. *European network: Each plot shows the running average of three components of u, v, and f. The colors refer to the different combinations the exact linear programming (LP) solver (blue) and Sinkhorn (gold).*



(a) Posterior DE (b) Posterior FR (c) Posterior DE→FR

FIG. 4.2. *European network: Posterior distributions of components of u, v, and f using the exact LP solver and Sinkhorn.*

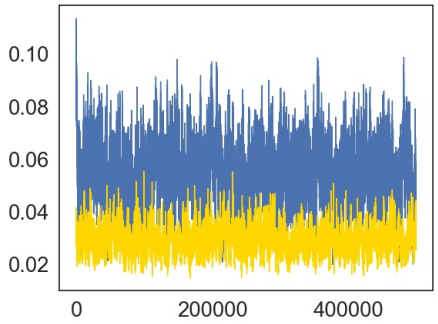
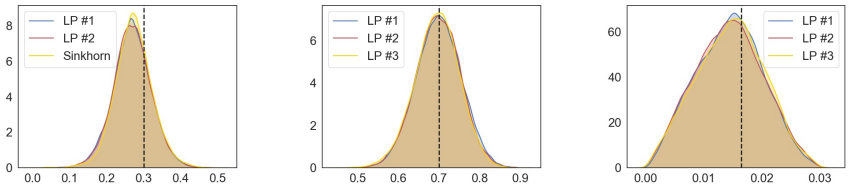
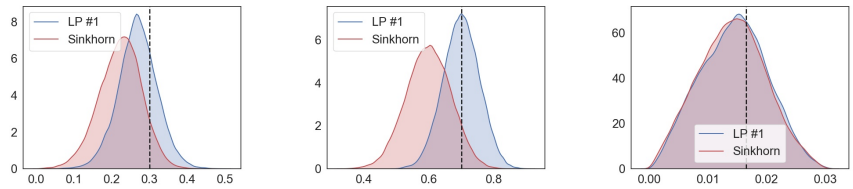


FIG. 4.3. *European network: misfit function for the exact LP solver (blue) and Sinkhorn (gold).*



(a) Posterior u_2 (b) Posterior v_2 (c) Posterior $C_{1,2}$

FIG. 4.4. *Graph based cost: Posterior distributions of components of u, v, and f using different combinations of δ.*



(a) Posterior u_2 (b) Posterior v_2 (c) Posterior $C_{1,2}$

FIG. 4.5. *Graph based cost: Posterior distributions of components of u, v, and f when using the exact LP solver in the RwM with Gibbs or regularizing the LP to employ the Sinkhorn algorithm.*

Influence of the sampling variance δ^2. We start by investigating the impact of the sampling variance δ^2. We perform MCMC runs for different combinations of δ_u, δ_v, and δ_f (listed in Table 4.1) and compute the running average and posterior distributions of some components. Note that these parameters affect the rate of convergence of the algorithm, but not the posterior distribution itself. The variance of the samples determines how much new samples differ from the previous iterates—the larger the variance the more adventurous the search. This corresponds to a lower acceptance rate and leads to more correlated samples. On the other hand smaller variance has a higher probability of accepting but is not adventurous and hence leads to highly correlated samples. It is thus desirable to have an acceptance rate that is

TABLE 4.1
Graph based cost: acceptance rates in % for different combinations of δ_u, δ_v, and δ_f.

δ_u^2	δ_v^2	δ_f^2	a	a_u	a_v	a_f
0.02	0.02	0.04	65	80	52	62
0.04	0.04	0.04	51	65	26	62
0.04	0.02	0.04	60	65	52	62

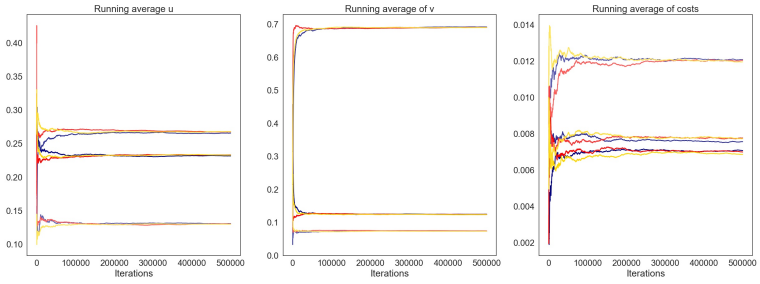


FIG. 4.6. *Graph based cost: Each plot shows the running average of three components of u, v, and f. The colors refer to the different combinations of δ used—blue $(\delta_u^2, \delta_v^2, \delta_f^2) = (0.02, 0.02, 0.04)$, red $(\delta_u^2, \delta_v^2, \delta_f^2) = (0.04, 0.04, 0.04)$, and gold $(\delta_u^2, \delta_v^2, \delta_f^2) = (0.04, 0.02, 0.04)$.*

close to neither 0 or 1. The running averages of three components of u, v, and W are shown in Figure 4.6, the respective posteriors in Figure 4.4. We see that the results are consistent for all combinations of δ's. However, the respective convergence rates vary; see Table 4.1. We observe a generally higher acceptance rate when sampling from the marginal distribution of $\mathfrak{p}$ and a decreased rate when increasing the sampling variance.

Exact vs. Sinkhorn. Next we investigate the sensitivity of the results with respect to the forward solver used in Algorithm 3.1. We run two RwM simulations—the first one using the exact solver and the second one using the Sinkhorn algorithm. We observe that both runs give similar posterior distributions if we choose the regularization parameter ϵ in a sensible way; see Figure 4.5. Generally speaking it seems advisable to choose it similar to the noise level (as in the shown results). We will investigate the impact of the regularization parameter in the next subsection in more detail.

4.3. Toeplitz cost. In the following we present more detailed numerical experiments if C is Toeplitz. The findings of the numerical experiments performed in this subsection can be summarized as follows:

- The posterior distributions of u, v, and f are consistent for varying ranges of proposal variances δ; see Figures 4.7 and 4.8.
- The exact solver and Sinkhorn's algorithm converge to similar posteriors if the entropic regularization parameter ϵ is chosen sensibly; see Figures 4.9, 4.10, and 4.11.
- The variance of the posteriors increases with the noise level in the data, as shown, for example, in Figures 4.12 and 4.13.
- Sinkhorn's algorithm gives a higher acceptance rate and a more monotone decrease of the data-misfit function; see Figure 4.14.

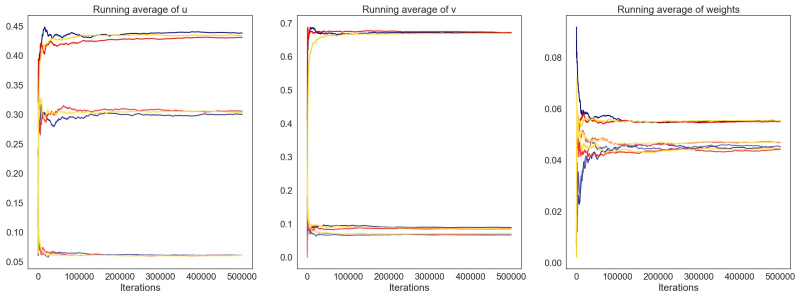
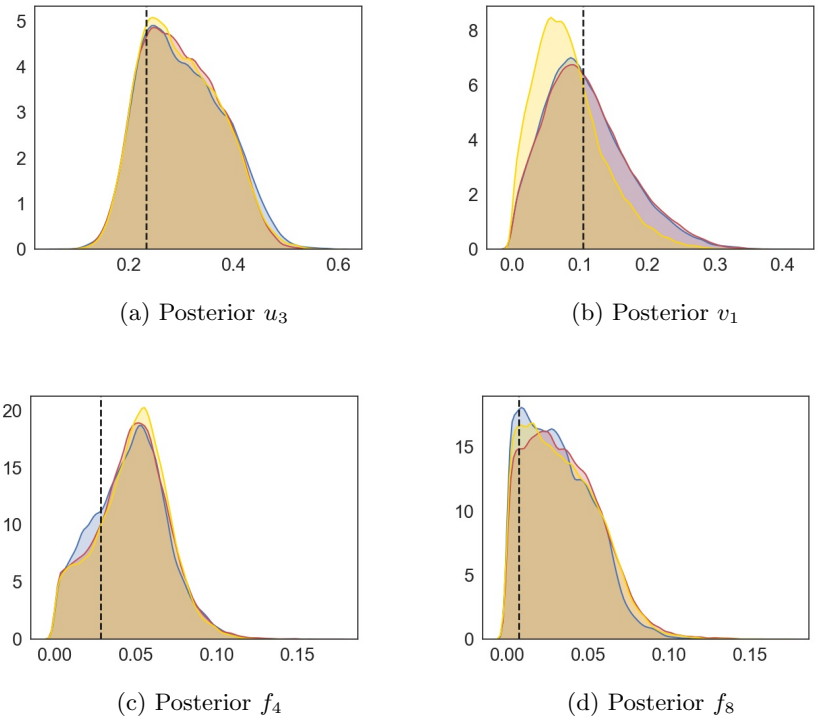


FIG. 4.7. *Toeplitz cost: Running averages of three components of u, v, and f. The colors refer to different variances of the proposals. The blue plots correspond to $\delta_u^2 = \delta_v^2 = \delta_f^2 = \delta^2 = 0.02$, the red ones to $\delta_u^2 = \delta_v^2 = 0.02$ and $\delta_f^2 = 0.04$, and the golden ones to $\delta_u^2 = \delta_v^2 = \delta_f^2 = 0.04$.*



(a) Posterior u_3 (b) Posterior v_1

(c) Posterior f_4 (d) Posterior f_8

FIG. 4.8. *Toeplitz cost: Posterior distributions of four components of u, v, and f. The colors refer to different variances of the proposals. The blue plots correspond to $\delta_u^2 = \delta_v^2 = \delta_f^2 = \delta^2 = 0.02$, the red ones to $\delta_u^2 = \delta_v^2 = 0.02$ and $\delta_f^2 = 0.04$, and the golden ones to $\delta_u^2 = \delta_v^2 = \delta_f^2 = 0.04$.*

We underpin these statements with numerical simulations using generated noisy transportation maps. We recall that C has $2n-3$ degrees of freedom in case of Toeplitz cost (i). This defines, as in the case of graph-based cost (iii), a mapping from the vector $f \in \mathbb{R}^{2n-3}$ to the cost matrix C, that is, $\mathcal{E} : \mathbb{R}^{2n-3} \to \mathcal{P}_{n\times n}$ with $C = \mathcal{E}(f)$. Hence we generate proposals for the vector f, which define the entries of C.

We set $n = 5$ and generate a noisy realization T^* for a given set of vectors $\mathfrak{p}, \mathfrak{q} \in \mathcal{P}_5$ and $f \in \mathbb{R}^7$ (which is mapped to the respective Toeplitz cost matrix C

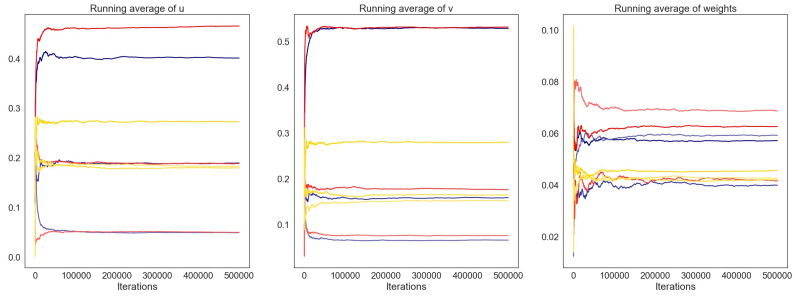
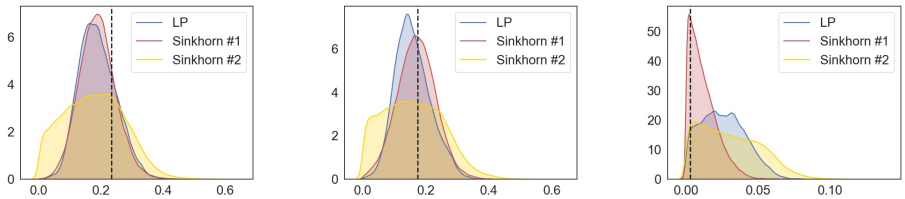


FIG. 4.9. *Toeplitz cost: Running averages of three components of u, v, and f. The colors refer to the used solver for the forward OT problem. Red corresponds to the exact LP solver, blue and gold to the Sinkhorn algorithm with regularization parameter $\epsilon = 0.04$ and $\epsilon = 0.1$, respectively.*



(a) Posterior u_3 (b) Posterior v_3 (c) Posterior $C_{1,2}$

FIG. 4.10. *Toeplitz cost: Posterior distributions of three components of u, v, and f. The colors refer to the solver used for the forward OT problem. Red corresponds to the exact LP solver, blue and gold to the Sinkhorn algorithm with regularization parameter $\epsilon = 0.04$ and $\epsilon = 0.1$, respectively.*

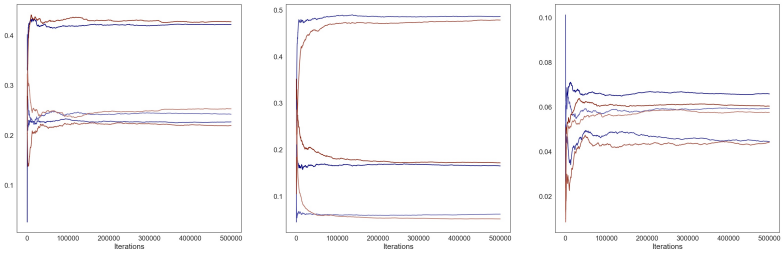
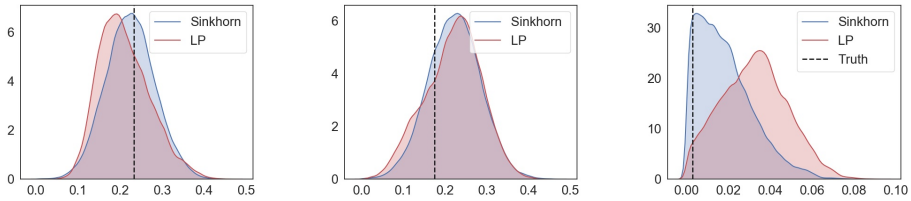


FIG. 4.11. *Toeplitz cost: Running averages of three components of u, v, and f for data generated by the Sinkhorn algorithm with $\epsilon = 0.04$. The blue plots are the running averages using Sinkhorn in the RwM, the red ones the exact LP solver.*

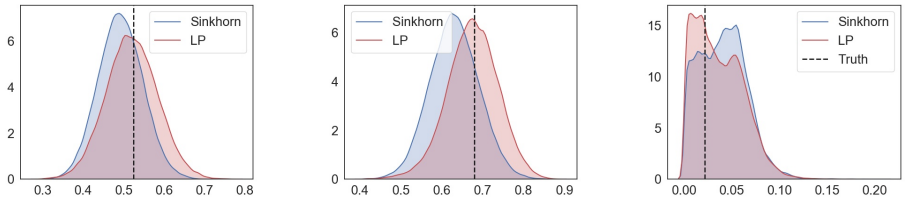
$\mathcal{P}_{5\times5}$). Then T^* is obtained by adding noise η with variance $\sigma^2 = 0.04$ (unless stated otherwise) and subsequent normalization of the distorted map. Note that this problem is overdetermined, since C is Toeplitz and $n > 2$.

Influence of the sample variance δ^2. As in the case of graph-based cost, we investigate the performance of the RwM-within-Gibbs algorithm for different combinations of δ_u, δ_v, and δ_W. Table 4.2 lists the considered δ-combinations together with the acceptance rates. The running average of the 3 or 4 different components of the posteriors are shown in Figures 4.7 and 4.8. The figures show, as expected, that



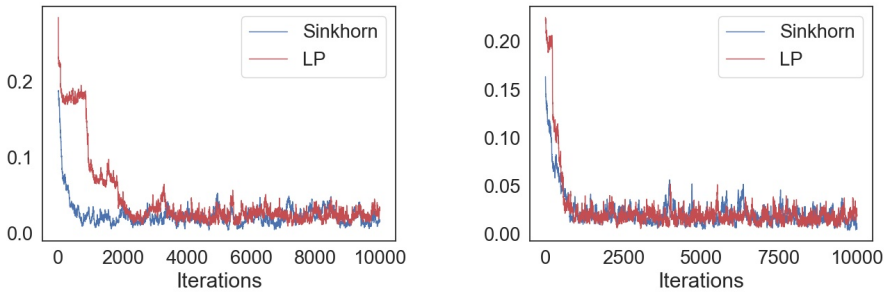
(a) Posterior u_3 (b) Posterior v_3 (c) Posterior $C_{1,2}$

FIG. 4.12. *Toeplitz cost: Posteriors of u_3, v_3, and $C_{1,2}$ using data generated by the Sinkhorn algorithm with $\epsilon = 0.04$.*



(a) Posterior u_1 (b) Posterior v_2 (c) Posterior $C_{5,1}$

FIG. 4.13. *Toeplitz cost: Posteriors of u_1, v_2, and $C_{5,1}$ using data generated by the Sinkhorn algorithm with $\epsilon = 0.1$.*



(a) Data generated with $\epsilon = 0.04$. (b) Data generated with $\epsilon = 0.1$.

FIG. 4.14. *Toeplitz cost: First* 10,000 *iterations of the misfit function* (2.14).

the posterior distributions are independent of the choice of the δ parameters. They also show that, in the ranges chosen, the rate of convergence does not vary in any considerable way—the method is fairly robust.

Exact vs. Sinkhorn Next we take a closer look how results change if we use Sinkhorn's algorithm instead of the exact solver. In particular we investigate how the size of the regularization parameter ϵ as well as the way we generate data affects the performance and results of the RwM algorithm.

We start by generating a noisy transportation map using the exact solver for (2.1). Then we compare the posterior distributions using the exact solver for the

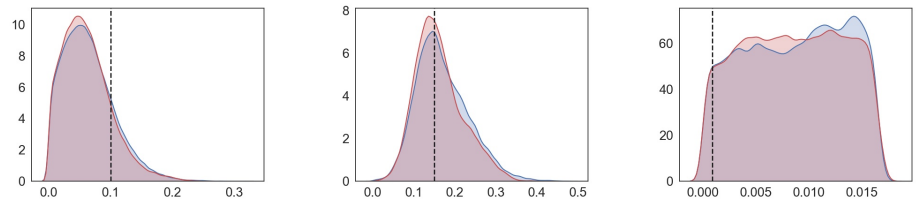
TABLE 4.2
Toeplitz cost: Acceptance rates for different combinations of δ_u, δ_v, and δ_f.

δ_u^2	δ_v^2	δ_f^2	a	a_u	a_v	a_f
0.02	0.02	0.02	66.1	67.9	55.4	75
0.02	0.02	0.04	60.3	68.3	55.3	52.7
0.04	0.04	0.04	44.1	45.5	30.0	57

reconstruction in the first run and the Sinkhorn algorithm with $\epsilon = 0.04$ and $\epsilon = 0.1$ in the next two test runs. Figure 4.9 shows the running average of three components of the vectors $\mathfrak{p}$, $\mathfrak{q}$, and f (left to right). The color coding relates to the solver used—red corresponds to the exact forward solver, blue and yellow when the Sinkhorn algorithm was used. Figure 4.10 shows the posterior distribution of the second component of $\mathfrak{p}$ and $\mathfrak{q}$ as well as the fifth entry of the vector f. We observe that we obtain similar posteriors when using the exact solver (LP) and Sinkhorn with $\epsilon = 0.04$. If the regularization parameter ϵ is chosen larger, which results in blurred (and therefore less sparse) transportation maps, the posterior distributions are less pronounced and close to uniform on the respective scaled intervals (due to the normalization constraint).

Next we generate the noisy transportation map using the Sinkhorn algorithm. We set the regularization parameter $\epsilon = 0.04$, and we distort the computed map with 4% and 10% noise. In each case we perform two different RwM runs, first using the Sinkhorn algorithm and then the exact solver. The respective posterior distributions are shown in Figures 4.12 and 4.13. We observe no significant difference in the quality of the posteriors. Figure 4.14 illustrates an interesting difference in the convergence behavior of the RwM algorithm. The data misfit term (2.14) shows multiple drops when using the exact solver. Such jumps have not been observed when using the Sinkhorn algorithm. We recall that the Sinkhorn algorithm solves the respective regularized (convex) optimization problem, which has a unique minimum. We believe that the nonuniqueness of the exact forward problem leads to several local minima in the inverse problem, in which the RwM algorithm gets stuck.

4.4. General cost. So far we have investigated overdetermined problems only. Hence we conclude by considering general nonsymmetric costs, that is, case (ii), for $n = 5$. This identification problem is underdetermined, and we expect poorer identifiability and quality of posteriors. This presumption is confirmed by our numerical experiments; see, for example, Figure 4.15.



(a) Posterior u_2 (b) Posterior v_2 (c) Posterior $C_{1,2}$

FIG. 4.15. *General cost: posterior distributions of different components of u, v and W. The two colors refer to the different combinations of δ - red to $\delta_u^2 = \delta_v^2 = \delta_W^2 = \delta = 0.02$ and ones to $\delta_u^2 = \delta_v^2 = 0.02$ and $\delta_W^2 = 0.04$.*

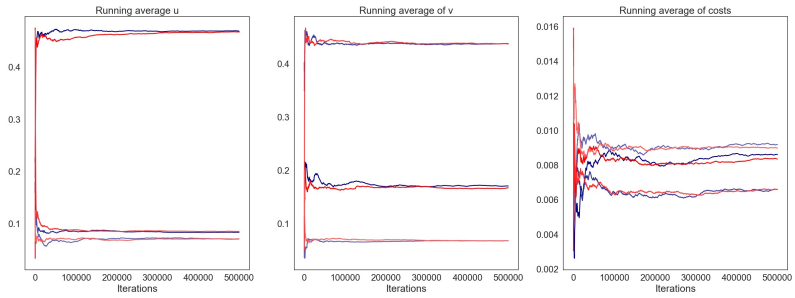


FIG. 4.16. *General cost: Each plot shows the running average of three components of u, v, and W. The colors correspond to different combinations of δ—red plots to $\delta_u^2 = \delta_v^2 = \delta_W^2 = \delta^2 = 0.02$ and blue ones to $\delta_u^2 = \delta_v^2 = 0.02$ and $\delta_W^2 = 0.04$.*

We investigate the identification from generated data in case of 4% noise. We perform two RwM test runs using the exact solver to calculate the posterior distributions of u, v, and W. In the first run we set the sample variance to $\delta_u^2 = \delta_v^2 = \delta_W^2 = \delta = 0.02$ and in the second to $\delta_u^2 = \delta_v^2 = 0.02$ and $\delta_W^2 = 0.04$. Figure 4.16 shows the running averages for 3 different components of u, v, and W for both runs. We observe that the components of u and v converge much faster than the ones of W and that the convergence is consistent for both sets of δ's. The posterior distributions of u and v give reasonable results, while the posteriors of the cost matrix are close to uniform on the respective scaled intervals (due to the normalization constraint). This indicates that the components of the cost matrix W are difficult to identify. We expect that the identifiability gets worse as the dimension n increases.

5. Conclusions. This paper introduces a systematic approach to infer unknown costs from noisy observations of OT plans. It is based on the Bayesian framework for inverse problems and allows us to quantify uncertainty in the obtained estimates; however, the methodology may also be viewed as a stochastic optimization procedure in its own right, tuning the unknowns so that the OT plan better fits the data. The performance of the developed methodologies is investigated using the example of international migration flows. In this context reported annual migration flow statistics can be interpreted as noisy observations of OT plans with cost related to the geographical position of countries. We formulate the graph-based problem, estimate the weights, which represent the costs of moving between neighboring countries, and quantify uncertainty in the weights. Our numerical investigations show that the proposed methodologies are robust and consistent for different cost functions and parametrizations. We observed that the distributions as well as the costs can be accurately determined for a variety of settings if the problem is overdetermined. The identifiability declines as the dimensionality increases or if the problem becomes underdetermined.

The proposed framework provides the basis for a multitude of future research directions in applied mathematics and other scientific disciplines. The next steps will focus on several questions related to the use of the Sinkhorn algorithm in the context of inverse OT, such as the convergence rate of the regularized problem (2.4) as $\epsilon \to 0$ or the optimal choice of ϵ with respect to the noise level σ; furthermore, hierarchical algorithms which learn parameters such as these from the data would also be of interest. In the context of migration flows, different modeling aspects, such as the coupling to age structured population models or the formulation of the OT

problem on the continuous level, will be investigated. Furthermore the application of the developed methodologies for general linear programs, which play an important role in transportation research, manufacturing, economics and demography, will be of interest.

Acknowledgment. The authors are grateful to Venkat Chandrasekaran for helpful discussions about the literature in inverse linear programming.

REFERENCES

[1] *Handbook on Measuring International Migration Through Population Censuses*, UN, New York, NY, 2017.
[2] G. J. Abel and N. Sander, *Quantifying global international migration flows*, Science, 343 (2014), pp. 1520–1522.
[3] R. K. Ahuja and J. B. Orlin, *Inverse optimization*, Oper. Res., 49 (2001), pp. 771–783.
[4] J. J. Azose and A. E. Raftery, *Estimation of emigration, return migration, and transit migration between all pairs of countries*, Proc. Natl. Acad. Sci. USA, 116 (2019), pp. 116–122, https://doi.org/10.1073/pnas.1722334116.
[5] N. Bonneel, M. van de Panne, S. Paris, and W. Heidrich, *Displacement interpolation using Lagrangian mass transport*, ACM Trans. Graph., 30 (2011), pp. 158:1–158:12, https://doi.org/10.1145/2070781.2024192.
[6] S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White, *MCMC methods for functions: Modifying old algorithms to make them faster*, Statist. Sci., 28 (2013), pp. 424–446, https://doi.org/10.1214/13-STS421.
[7] M. Cuturi, *Sinkhorn distances: Lightspeed computation of optimal transport*, in Advances in Neural Information Processing Systems 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds., 2013, pp. 2292–2300.
[8] G. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 2016.
[9] J. de Beer, J. Raymer, R. van der Erf, and L. van Wissen, *Overcoming the problems of inconsistent international migration data: A new method applied to flows in Europe*, Eur. J. Population, 26 (2010), pp. 459–481, https://doi.org/10.1007/s10680-010-9220-z.
[10] S. Dempe and S. Lohse, *Inverse linear programming*, in Recent Advances in Optimization, A. Seeger, ed., Springer, Berlin, 2006, pp. 19–28.
[11] E. W. Dijkstra, *A note on two problems in connexion with graphs*, Numer. Math., 1 (1959), pp. 269–271, https://doi.org/10.1007/BF01386390.
[12] R. Flamary and N. Courty, *Python Optimal Transport Library*, 2017, https://github.com/rflamary/POT.
[13] A. Galichon and B. Salanie, *Cupid's invisible hand: Social surplus and identification in matching models*, Rev. Econ. Stud., submitted.
[14] W. K. K. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, 57 (1970), pp. 97–109.
[15] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Springer Science & Business Media, New York, NY, 2006.
[16] R. Li, X. Ye, H. Zhou, and H. Zha, *Learning to match via optimal transport*, J. Mach. Learn. Res., 20 (2019), pp. 1–37.
[17] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *Equation of state calculations by fast computing machines*, J. Chem. Phys., 21 (1953), pp. 1087–1092.
[18] J. Nocedal and S. Wright, *Numerical Optimization*, Springer Science & Business Media, New York, NY, 2006.
[19] G. Peyré and M. Cuturi, *Computational optimal transport*, Found. Trends Mach. Learn., 11 (2019), pp. 355–607.
[20] J. Raymer, J. de Beer, and R. van der Erf, *Putting the pieces of the puzzle together: Age and sex-specific estimates of migration amongst countries in the EU/EFTA*, 2002–2007, Eur. J. Population, 27 (2011), pp. 185–215, https://doi.org/10.1007/s10680-011-9230-5.
[21] J. Raymer, A. Wiśniowski, J. J. Forster, P. Smith, and J. Bijak, *Integrated modeling of European migration*, J. Amer. Statist. Assoc., 108 (2013), pp. 801–819, https://doi.org/10.1080/01621459.2013.789435.
[22] S. Reich, *Data assimilation: The Schrödinger perspective*, Acta Numer., 28 (2019), pp. 635–711.

[23] G. O. Roberts, A. Gelman, and W. R. Gilks, *Weak convergence and optimal scaling of random walk Metropolis algorithms*, Ann. Appl. Probab., 7 (1997), pp. 110–120.
[24] G. O. Roberts and J. S. Rosenthal, *Optimal scaling for various Metropolis-Hastings algorithms*, Statist. Sci., 16 (2001), pp. 351–367, https://doi.org/10.1214/ss/1015346320.
[25] Y. Rubner, C. Tomasi, and L. J. Guibas, *The earth mover's distance as a metric for image retrieval*, Int. J. Comput. Vis., 40 (2000), pp. 99–121, https://doi.org/10.1023/A:1026543900054.
[26] J. Saez-Gallego and J. M. Morales, *Short-term forecasting of price-responsive loads using inverse optimization*, IEEE Trans. Smart Grid, 9 (2018), pp. 4805–4814.
[27] F. Santambrogio, *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*, Birkhaüser-Verlag, Basel.
[28] B. Schmitzer, *Stabilized sparse scaling algorithms for entropy regularized transport problems*, SIAM J. Sci. Comput., 41 (2019), pp. A1443–A1481.
[29] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, Cambridge, UK, 1999.
[30] C. Villani, *Topics in Optimal Transportation*, American Mathematical Society, Providence, RI, 2003.
[31] J. Yang, G. O. Roberts, and J. S. Rosenthal, *Optimal Scaling of Metropolis Algorithms on General Target Distributions*, arXiv preprint, arXiv:1904.12157, 2019.
[32] J. Zhang and Z. Liu, *Calculating some inverse linear programming problems*, J. Comput. Appl. Math., 72 (1996), pp. 261–273.