Multi-marginal optimal transport and probabilistic graphical models

Isabel Haasler, Rahul Singh, Qinsheng Zhang, Johan Karlsson, and Yongxin Chen

***Abstract*—We study multi-marginal optimal transport problems from a probabilistic graphical model perspective. We point out an elegant connection between the two when the underlying cost for optimal transport allows a graph structure. In particular, an entropy regularized multi-marginal optimal transport is equivalent to a Bayesian marginal inference problem for probabilistic graphical models with the additional requirement that some of the marginal distributions are specified. This relation on the one hand extends the optimal transport as well as the probabilistic graphical model theories, and on the other hand leads to fast algorithms for multi-marginal optimal transport by leveraging the well-developed algorithms in Bayesian inference. Several numerical examples are provided to highlight the results.**

***Index Terms*—Optimal transport, Probabilistic graphical models, Belief Propagation, Norm-product, Iterative Scaling, Bayesian inference.**

I. Introduction

OPTIMAL transport (OT) theory [1], [2] is a powerful tool in the study of probability distributions. The subject dates back to 1781, when the civil engineer Monge aimed to find an optimal strategy to move soil to road construction sites. Over 200 years of development have brought OT far beyond a civil engineering problem to a compelling mathematical framework which has found applications in economics, signal and image processing, systems and control, statistics, and machine learning [3], [4], [5], [6], [7], [8], [9]. The inherent properties of OT make it especially suitable for handling high-dimensional data with low-dimensional structure, which is the case in most machine learning settings. Thanks to the discoveries of several efficient algorithms such as iterative scaling, also called Sinkhorn iterations [10], OT has become a powerful framework for a range of machine learning problems. In the 30s this algorithm has also been studied in the statistics community under the name contingence table [11].

The aim of standard OT problems is to find a joint distribution of two given marginals that minimizes the total transportation cost between them. In some applications, such as incompressible fluid flow modeling, video prediction, tomography, and information fusion problems, more than two marginal distributions are given. To tackle these problems, a multi-marginal generalization of OT has been developed, known as multi-marginal optimal transport (MOT) [12], [13], [14], [15], [16]. MOT was first proposed in [14] as a theoretical extension to OT. Since then, the problem has been studied from a theoretical viewpoint [15] as well as computational perspective [16]. It has found applications in signal processing [17], fluid dynamics [18], density functional theory [19], [20], and estimation and control [21], [22]. Many results for the standard OT problem have been extended to the multi-marginal setting. In particular, the iterative scaling method [10] has been generalized to MOT [23]. However, for the multi-marginal setting the computational complexity remains high, especially when the number of marginal distributions is large [16], [24].

On a seemingly different topic, probabilistic graphical models (PGMs) [25], [26], [27] provide a framework for multi-dimensional random variables. They have been used for a large variety of applications including speech recognition, computer vision, communications, and bioinformatics [28], [29], [25], [26]. PGMs capture the dependencies of a set of random variables compactly as a graph, and are an efficient and robust tool to study the relationship of several probabilistic quantities. Moreover, prior knowledge can be easily incorporated in the model. During the last decades, many efficient algorithms have been developed for inference and learning of PGMs. These algorithms leverage the underlying graph structure, making it possible to solve many otherwise extremely difficult problems. Well-known algorithms for the inference problem include, e.g., belief propagation, and the junction tree algorithm [30], [31], [32], [33], [34].

The purpose of this paper is to point out an elegant connection between MOT and PGMs. More precisely, the main contribution of this work is to establish an equivalence between regularized MOT problems, where the cost function is structured according to a graph, and the inference problem for a PGM on the same graph, where some marginal distributions are fixed. This connection leads to a novel interpretation for both MOT and PGMs. On the one hand, MOT can be viewed as an inference problem of a PGM with constraints on some of the marginals, that is, constrained inference problems, and on the other hand, the inference problem of a PGM is a MOT problem, where the available marginals are Dirac distributions.

From a numerical point of view, this connection allows for adapting existing PGM algorithms to this class of MOT problems with graphical structured cost. In this work, we focus on belief propagation (BP) [35], [30] and an extension of it known as norm-product algorithm [36]. These belong to the so called message passing algorithms which exploit

This work was supported by the Swedish Research Council (VR), grant 2014-5870, SJTU-KTH cooperation grant and the NSF under grant 1901599 and 1942523.

I. Haasler and J. Karlsson are with the Division of Optimization and Systems Theory, Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden. haasler@kth.se, johan.karlsson@math.kth.se

R. Singh, Q. Zhang and Y. Chen are with the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA. {qzhang419,rasingh,yongchen}@gatech.edu

I. Haasler, R. Singh and Q. Zhang contribute equally to this paper.

the underlying graphical structure, and rely on exchanging information between the nodes. Thus, only local updates are needed, which greatly reduces the computational complexity of the inference problem. If the graphical model is a tree, these algorithms converge globally to the exact conditional marginal distributions in a finite number of iterations. For general PGMs with cycles, there is no convergence guarantee, but both methods usually work well in practice and provide relatively accurate approximations of the marginals. In this work, we develop algorithms for solving entropy regularized MOT problems, or equivalently constrained inference problems, by combining these message passing algorithms with the iterative scaling method. Moreover, we build on earlier results and establish global convergence of our algorithms for tree graphs. Similar constrained inference problems have previously been studied in [37]. Interestingly, the algorithm presented therein is comparable to our proposed extension of the BP algorithm. With the connections to MOT, we provide a new motivation for studying this problem, which also leads to a more complete picture of the algorithms.

A promising application of our framework are inference problems for collective dynamics, for instance, the estimation of the behavior of large groups from only aggregate measurements. Such types of filtering methods are crucial for collective dynamics since it is usually impossible to track the trajectories of each single agent in a large population, due to exploding computational complexity, lack of sensor data or for privacy considerations. Related problems have been studied under the name collective graphical models (CGMs) [38], [39], [40]. These works consider a large collection of identical graphical models, which are observed simultaneously, and aim to infer the aggregate distribution over the nodes. Several heuristic algorithms [40] have been proposed to solve the resulting inference problems. Our MOT framework suggests a different observation model [41], [42], which is reasonable in many scenarios. More importantly, our algorithms, which enjoy global convergence guarantee, provide a reliable machine to estimate collective dynamics in these models.

The rest of this paper is structured as follows. In Section II we review some background knowledge in optimal transport and probabilistic graphical models. In Section III we provide the main theoretical result in this paper, which is the equivalence between entropy regularized MOT and the inference problem for PGMs. We also modify the belief propagation algorithm to solve MOT problems. Another algorithm based on the norm-product algorithm is introduced in Section IV. We test and verify our results through several numerical examples in Section V. This is followed by a brief concluding remark in Section VI.

Notation: The notation used throughout is mostly standard. However, with $\exp(\cdot)$, $\ln(\cdot)$, $\odot$, and $./$ we denote the elementwise exponential, logarithm, multiplication, and division of vectors, matrices, and tensors, respectively. Moreover, $\otimes$ denotes the outer product. By $\mathbf{1}$ we denote a vector of ones, the size of which will be clear from the context. Throughout, we use bold symbols to represent vectors, e.g., $\mathbf{b}_j, \boldsymbol{\mu}_j$, and regular symbol for the corresponding entries, e.g., $b_j(x_j), \mu_j(x_j)$.

II. Preliminaries

In this section, we provide a quick overview of optimal transport theory, probabilistic graphical models and belief propagation algorithm. We only cover material that is most relevant to this work. The reader is referred to [2], [25], [26] for more details.

A. Optimal transport

In optimal transport (OT) problems, one seeks an optimal plan that transports mass from a source distribution to a target distribution with minimum cost. In its original formulation [43], OT was studied over Euclidean space. However, in general, OT problems can be formulated in both continuous space and discrete space. In this work, we focus on optimal transport over discrete space.

Let $\boldsymbol{\mu}_1 \in \mathbb{R}_+^{d_1}, \boldsymbol{\mu}_2 \in \mathbb{R}_+^{d_2}$ be two discrete distributions, viz., nonnegative vectors, with equal mass, that is $\sum_{i_1} \mu_1(i_1) = \sum_{i_2} \mu_2(i_2)$. Here $\mu_1(i)$ denotes the amount of mass in the source distribution at location i and $\mu_2(i)$ denotes the amount of mass in the target distribution at location i. Without loss of generality, we assume that both $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are probability vectors, that is, the total mass is $\sum_{i_1} \mu_1(i_1) = \sum_{i_2} \mu_2(i_2) = 1$. The transport cost of moving a unit mass from point i_1 to i_2 is denoted by $C(i_1, i_2)$, and collected in the matrix $\mathbf{C} = [C(i_1, i_2)] \in \mathbb{R}^{d_1 \times d_2}$. In the Kantorovich formulation [44] of OT, the goal is to find a transport plan between the two marginal distributions $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ that minimizes the total transport cost. A transport plan is encoded in a joint probability matrix $\mathbf{B} = [B(i_1, i_2)] \in \mathbb{R}_+^{d_1 \times d_2}$ of $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$. Then the total transport cost is $\sum_{i_1, i_2} C(i_1, i_2) B(i_1, i_2) = \mathrm{trace}(\mathbf{C}^T \mathbf{B})$ and therefore the OT problem reads

$$\begin{aligned} \min_{\mathbf{B} \in \mathbb{R}_+^{d_1 \times d_2}} \quad & \mathrm{trace}(\mathbf{C}^T \mathbf{B}) \\ \text{subject to } & \mathbf{B}\mathbf{1} = \boldsymbol{\mu}_1 \\ & \mathbf{B}^T \mathbf{1} = \boldsymbol{\mu}_2, \end{aligned} \tag{1}$$

where $\mathbf{1}$ denotes a vector of ones of proper dimension. The constraints are to enforce that $\mathbf{B}$ is a joint distribution between $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$.

Even though the above OT problem (1) is a linear program, in many practical applications it is too difficult to be solved directly using standard solvers due to the large number of variables [23], [17], especially in the case where the marginal distributions $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ come from discretizations of continuous measures. Recently, a regularization of OT [10] was proposed that greatly reduces the computational complexity of (approximately) solving OT problems over discrete space. In this method, an entropy term

$$\mathcal{H}(\mathbf{B}) = -\sum_{i_1, i_2} B(i_1, i_2) \ln B(i_1, i_2) \tag{2}$$

is added to regularize the problem, leading to

$$\begin{aligned} \min_{\mathbf{B} \in \mathbb{R}_+^{d_1 \times d_2}} \quad & \mathrm{trace}(\mathbf{C}^T \mathbf{B}) - \epsilon \mathcal{H}(\mathbf{B}) \\ \text{subject to } & \mathbf{B}\mathbf{1} = \boldsymbol{\mu}_1 \\ & \mathbf{B}^T \mathbf{1} = \boldsymbol{\mu}_2, \end{aligned} \tag{3}$$

where $\epsilon > 0$ is a regularization parameter. The entropy regularized OT problem (3) is strictly convex and thus the solution is unique. More importantly, it can be solved efficiently via the Sinkhorn algorithm [45], [10], also known as iterative scaling [11], [46]. Let $\mathbf{K} = [K(i_1, i_2)] \in \mathbb{R}^{d_1 \times d_2}$ be defined as $K(i_1, i_2) = \exp(-C(i_1, i_2)/\epsilon)$, then the iterative scaling updates alternate between the two steps

$$\mathbf{u}_1 \leftarrow \boldsymbol{\mu}_1./\mathbf{K}\mathbf{u}_2, \quad \mathbf{u}_2 \leftarrow \boldsymbol{\mu}_2./\mathbf{K}^T\mathbf{u}_1, \tag{4}$$

where ./ denotes element-wise division. The algorithm converges linearly to a unique pair of vectors $\mathbf{u}_1 \in \mathbb{R}^{d_1}, \mathbf{u}_2 \in \mathbb{R}^{d_2}$ up to a normalization [46]. Given the limit point of the iteration, the solution to (3) has the form

$$\mathbf{B} = \operatorname{diag}(\mathbf{u}_1)\mathbf{K}\operatorname{diag}(\mathbf{u}_2), \tag{5}$$

that is, $B(i_1, i_2) = K(i_1, i_2)u_1(i_1)u_2(i_2)$ for all $1 \leq i_1 \leq d_1, 1 \leq i_2 \leq d_2$.

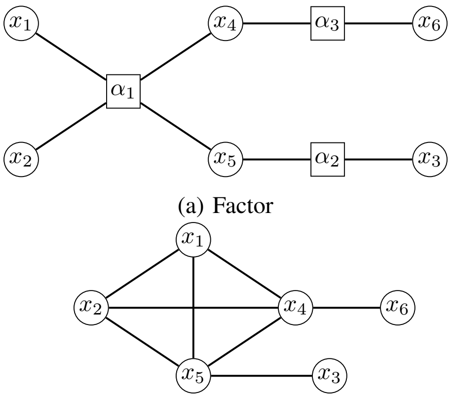
B. Probabilistic graphical models

A probabilistic graphical model (PGM) is a graph-based representation of a collection of random vectors that captures the conditional dependencies between them. It provides a compact representation of their joint distributions through factorization: a graphical model consists of a collection of distributions that factorize according to an underlying graph structure. In this work we focus on undirected graphs, which represent Markov random fields [25]. Note that directed graphs represent Bayesian networks, and can always be transformed into a Markov random field [26].

Among the many representations for Markov random fields, the factor graph representation has been widely used due to its elegance and flexibility [25], and is also used in this paper. Consider a graphical model with underlying factor graph $G = (V, F, E)$ where V denotes the set of variable nodes, F denotes the set and factor nodes, and E stands for the edges connecting them (see Figure 1 for an example). In such a factor graph G, the neighbors of a node $j \in V$ consist of factor nodes, $N(j) \subset F$, and the neighbors of a factor node $\alpha \in F$ are variable nodes $N(\alpha) \subset V$. Therefore, G is a bipartite graph [47]. Each variable node $j \in V$ is associated with a random variable x_j which can be either discrete or continuous. Here we consider only the discrete cases and assume that the random variable x_j can take d_j possible values. Each factor node $\alpha \in F$ corresponds to the dependence between the variable nodes connected to α, which are compactly denoted by $\mathbf{x}_\alpha := \{x_j \ ; \ j \in N(\alpha)\}$. In Markov random fields with underlying factor graph G, the joint probability is assumed to be of the form

$$p(\mathbf{x}) := p(x_1, x_2, \ldots, x_J) = \frac{1}{Z} \prod_{j \in V} \phi_j(x_j) \prod_{\alpha \in F} \psi_\alpha(\mathbf{x}_\alpha) \tag{6}$$

where $\boldsymbol{\phi}_j$ is the node/local potential corresponding to node j, $\boldsymbol{\psi}_\alpha$ is the factor node potential corresponding to factor node α, and Z is a normalization constant. A factor node potential ψ_α describes the dependence between random variables in $\{x_j \ ; \ j \in N(\alpha)\}$. The node potentials normally come from two sources: prior belief and evidence from measurements. In the latter, $\phi_j(x_j)$ is short for $\phi_j(x_j, y_j)$ [33] with y_j being the measurement. In cases where all the local potentials are induced by evidence, a more precise formula for the model is $p(\mathbf{x}) = \frac{1}{Z} \prod_{j \in V} \phi_j(x_j, y_j) \prod_{\alpha \in F} \psi_\alpha(\mathbf{x}_\alpha)$. Since the measurement is usually specified in inference problems, y_j is often neglected to simplify the notation. In principle, the node potentials can be fully absorbed into the factor potentials, that is, the joint distribution becomes $p(\mathbf{x}) \propto \prod_{\alpha \in F} \psi_\alpha(\mathbf{x}_\alpha)$. For the ease of presentation, we adopt the formulation (6).



(a) Factor

(b) Standard

Fig. 1: Factor graph representation vs standard graph representation.

Apart from factor graphs, another popular representation of PGMs is the standard graph where the nodes are all variables. In the standard graph representation, the dependence between the variables are fully captured by the edges of the graphs. The two representations are equivalent and one can transform one to another easily as the following example illustrates.

Example 1. *The factor graph in Figure 1a models the joint distribution*

$$p(x_1, \ldots, x_6) = \frac{1}{Z} \psi_{\alpha_1}(\mathbf{x}_{\alpha_1}) \psi_{\alpha_2}(\mathbf{x}_{\alpha_2}) \psi_{\alpha_3}(\mathbf{x}_{\alpha_3}) \prod_{j=1}^{6} \phi_j(x_j)$$

with $\mathbf{x}_{\alpha_1} = \{x_1, x_2, x_4, x_5\}$, $\mathbf{x}_{\alpha_2} = \{x_3, x_5\}$, $\mathbf{x}_{\alpha_3} = \{x_4, x_6\}$. *To convert this into a standard graph representation, the dependence among variables induced by the three factors have to be translated to edges. This is straightforward for factors* $\boldsymbol{\psi}_{\alpha_2}$ *and* $\boldsymbol{\psi}_{\alpha_3}$. *The factor* $\boldsymbol{\psi}_{\alpha_1}$ *involves* 4 *variables and is more complicated. Without further assumptions on the structure of this factor, it may induce dependence among all of the* 4 *variables, and thus a complete graph connecting them is required (see Figure 1b).*

The two fundamental problems in PGMs are inference and learning. Learning refers to estimating the underlying graphical models (often the parametrized factors) using available data sampled from the models. In inference problems, the parameters of the models are given. Instead, the goal is to infer the statistics of the node variables. The two main approaches to inference problems are maximum a posteriori estimation and Bayesian marginal inference [25]. Given a set of measurements $\{y_1, y_2, \ldots, y_J\}$, the aim of maximum a posteriori estimation is to find the most likely variable value $\{x_1, x_2, \dots, x_J\}$ given the model and measurements. Instead, Bayesian inference calculates the posterior marginal distributions of each node. The focus of this paper is most relevant to Bayesian/marginal inference.

Formally, given a graphical model (6), the objective of Bayesian inference is to calculate marginal distributions $p(x_j)$ for $j \in V$. In cases where the nodes variables are discrete, the marginal $p(x_j)$ is defined to be

$$\begin{aligned} p(x_j) &= \sum_{x_1,\dots,x_{j-1},x_{j+1},\dots,x_J} p(\mathbf{x}) \\ &= \frac{1}{Z} \sum_{x_1,\dots,x_{j-1},x_{j+1},\dots,x_J} \prod_{i\in V} \phi_i(x_i) \prod_{\alpha\in F} \psi_\alpha(\mathbf{x}_\alpha). \end{aligned} \tag{7}$$

The Bayesian inference problem can be reformulated as the optimization problem of minimizing

$$\min_{\mathbf{b}} \mathcal{F}(\mathbf{b}) := \mathcal{U}(\mathbf{b}) - \mathcal{H}(\mathbf{b}), \tag{8}$$

with

$$\mathcal{U}(\mathbf{b}) = -\sum_{\mathbf{x}} b(\mathbf{x}) \left(\sum_{j\in V} \ln \phi_j(x_j) + \sum_{\alpha\in F} \ln \psi_\alpha(\mathbf{x}_\alpha) \right) \tag{9}$$

and

$$\mathcal{H}(\mathbf{b}) = -\sum b(\mathbf{x}) \ \ln \ b(\mathbf{x}) \tag{10}$$

over the space of all the probability distributions on $\mathbf{x}$. By convention, $\mathcal{F}, \mathcal{U}, \mathcal{H}$ are known as free energy, average energy and entropy respectively due to their similarity to thermodynamics [48]. To see this, we note that the Kullback-Leibler (KL) divergence [49] between $b(\mathbf{x})$ and $p(\mathbf{x})$ is

$$\begin{aligned} \mathrm{KL}(\mathbf{b} \mid \mathbf{p}) &:= \sum_{\mathbf{x}} b(\mathbf{x}) \ln \frac{b(\mathbf{x})}{p(\mathbf{x})} = \mathcal{F}(\mathbf{b}) + \ln Z \\ &= \sum_{\mathbf{x}} b(\mathbf{x}) \ln \frac{b(\mathbf{x})}{1/Z \prod_{j\in V} \phi_j(x_j) \prod_{\alpha\in F} \psi_\alpha(\mathbf{x}_\alpha)}. \end{aligned}$$

Since the KL divergence is nonnegative and equals 0 only if $\mathbf{b} = \mathbf{p}$, the unique minimizer of $\mathcal{F}(\mathbf{b})$ is $\mathbf{p}$ with the associated minimum being $-\ln Z$.

The optimization formulation (8) of Bayesian inference is the basis for variational inference [25], one of the most popular approximate inference techniques. In the variational inference approach, the approximate distribution $\mathbf{b}$ is usually assumed to have some simple structure to ease the optimization, e.g., the mean field approximation $b(\mathbf{x}) = b_1(x_1) \cdots b_J(x_J)$ [50]. This work is not concerned with variational inference; (8) simply serves as a link to connect Bayesian inference with optimization. In the PGM literature [51], [36], it is common to introduce a temperature coefficient $\epsilon > 0$ into (8), which leads to a slightly more general optimization problem

$$\min_{\mathbf{b}} \ \mathcal{F}(\mathbf{b}) = \mathcal{U}(\mathbf{b}) - \epsilon\mathcal{H}(\mathbf{b}). \tag{11}$$

It corresponds to the Bayesian inference for the model

$$p_\epsilon(\mathbf{x}) = \frac{1}{Z_\epsilon} \prod_{j\in V} \phi_j(x_j)^\epsilon \prod_{\alpha\in F} \psi_\alpha(\mathbf{x}_\alpha)^\epsilon.$$

From an optimization point of view, (11) is a regularized version of the linear program

$$\min_{\mathbf{b}} \ \mathcal{U}(\mathbf{b}).$$

Interestingly, this linear program in fact corresponds to the maximum a posteriori problem [36] for the model (6).

C. Belief Propagation

In principle, Bayesian inference is achievable through the definition (7) by calculating the marginals using brute force summation. The complexity of this summation however scales exponentially as the number of variable nodes J goes up [33]. Also the normalization factor Z is extremely difficult to calculate when J is large due to the same reason.

During the last two decades, many methods have been developed to solve or approximately solve Bayesian marginal inference problems. One of the most widely used methods is a message-passing algorithm called Belief Propagation [35]. It updates the marginal distribution of each node through communications of beliefs/messages between them. In the factor graph representation, it reads

$$m_{\alpha\to j}(x_j) \ \propto \sum_{\mathbf{x}_\alpha \setminus x_j} \psi_\alpha(\mathbf{x}_\alpha) \prod_{i\in N(\alpha)\setminus j} n_{i\to\alpha}(x_i) \tag{12a}$$

$$n_{j\to\alpha}(x_j) \ \propto \ \phi_j(x_j) \prod_{\beta\in N(j)\setminus\alpha} m_{\beta\to j}(x_j), \tag{12b}$$

where $m_{\alpha\to j}(x_j)$ denotes the message from factor node α to variable node j, and $n_{j\to\alpha}(x_j)$ represents the message from variable node j to factor node α. The symbol $\propto$ means "proportional to" and indicates that often a normalization is applied in the Belief Propagation algorithm. The messages in (12) are updated iteratively over the factor graph.

The Belief Propagation algorithm was first invented to solve Bayesian inference problems over trees, in which case global convergence is guaranteed [35], [33]. This method was later generalized to deal with inference problems involving general graphs under the name Loopy Belief Propagation [30]. Even though there is no convergence proof and the algorithm does diverge in some occasions, it works well in practice and is widely adopted. When the algorithm converges, one can calculate the beliefs on the variables and factors by

$$b_j(x_j) \ \propto \ \phi_j(x_j) \prod_{\alpha\in N(j)} m_{\alpha\to j}(x_j) \tag{13a}$$

$$b_\alpha(\mathbf{x}_\alpha) \ \propto \ \psi_\alpha(\mathbf{x}_\alpha) \prod_{j\in N(\alpha)} n_{j\to\alpha}(x_j). \tag{13b}$$

In cases where the factor graph has no cycles (i.e., it is a tree), the beliefs in (13) coincide with the true posterior marginals, that is,

$$p(x_j) = b_j(x_j), \ \ \forall j \in V, \ \forall x_j \tag{14a}$$

$$p(\mathbf{x}_\alpha) = b_\alpha(\mathbf{x}_\alpha), \ \ \forall \alpha \in F, \ \forall \mathbf{x}_\alpha. \tag{14b}$$

For general graphs with cycles, convergence is not guaranteed and even if it does converge, the beliefs in (13) are only approximations of the true marginals $p(x_j)$, $p(\mathbf{x}_\alpha)$. A remarkable discovery [30], [31] related to (Loopy) Belief

Propagation is that if the updates (12) converge, then the beliefs in (13) form a fixed point of the Bethe free energy [30], [31]

$$\mathcal{F}_{\text{Bethe}}(\mathbf{b}) = \mathcal{U}_{\text{Bethe}}(\mathbf{b}) - \mathcal{H}_{\text{Bethe}}(\mathbf{b}), \tag{15}$$

where $\mathcal{U}_{\text{Bethe}}(\mathbf{b})$ is the Bethe average energy

$$-\sum_{\alpha\in F}\sum_{\mathbf{x}_\alpha} b_\alpha(\mathbf{x}_\alpha)\ln\psi_\alpha(\mathbf{x}_\alpha) - \sum_{j\in V}\sum_{x_j} b_j(x_j)\ln\phi_j(x_j) \tag{16}$$

and $\mathcal{H}_{\text{Bethe}}(\mathbf{b})$ is the Bethe entropy

$$-\sum_{\alpha\in F}\sum_{\mathbf{x}_\alpha} b_\alpha(\mathbf{x}_\alpha)\ln b_\alpha(\mathbf{x}_\alpha) + \sum_{j\in V}(N_j - 1)\sum_{x_j} b_j(x_j)\ln b_j(x_j) \tag{17}$$

with N_j denoting the degree of the variable node j, i.e., $N_j = |N(j)|$. In (15), we define $\mathbf{b} = \{\mathbf{b}_j, \mathbf{b}_\alpha \ : \ j \in V, \alpha \in F\}$. This is different to $\mathbf{b}$ in (8), which is a J-mode tensor. For the sake of conciseness, by abuse of notation, we use $\mathbf{b}$ in both settings. For a factor tree, the two are connected through the relation $b(\mathbf{x}) \sim (\prod_{\alpha\in F} b_\alpha(\mathbf{x}_\alpha))(\prod_{j\in V} b_j(x_j)^{1-N_j})$ [25]. In terms of Bethe free energy, the Bayesian inference problem reads

$$\min_{\mathbf{b}} \quad \mathcal{F}_{\text{Bethe}}(\mathbf{b}) \tag{18a}$$

$$\text{subject to} \sum_{\mathbf{x}_\alpha\setminus x_j} b_\alpha(\mathbf{x}_\alpha) = b_j(x_j), \forall j \in V, \alpha\in N(j) \tag{18b}$$

$$\sum_{\mathbf{x}_\alpha} b_\alpha(\mathbf{x}_\alpha) = 1, \ \forall \alpha \in F. \tag{18c}$$

The constraint (18b) is to ensure that $\mathbf{b}_\alpha, \mathbf{b}_j$ are compatible and (18c) is to guarantee that they are in the probability simplex. It is easy to check that when the factor graph has no cycles, the Bethe free energy (15) is strictly convex in the feasible set defined by the constraints (18b)-(18c), and is equal to the free energy (8), i.e., $\mathcal{F}_{\text{Bethe}} = \mathcal{F}$. Thus, (18) is again a convex optimization problem. For general graphs, the Bethe free energy serves as a good approximation of the free energy [31], but is no longer convex.

III. Multimarginal optimal transport as Bayesian inference

Multimarginal optimal transport (MOT) extends the OT framework (1) to the setting involving multiple distributions. In particular, in MOT, one aims to find a transport plan among a set of marginals $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_J$ with $J \geq 2$. In this setting, the transport cost is encoded in a tensor $\mathbf{C} = [C(i_1, i_2, \dots, i_J)] \in \mathbb{R}^{d_1\times d_2\times\cdots\times d_J}$ where $C(i_1, i_2, \dots, i_J)$ denotes the unit transporting cost corresponding to the locations $i_1, i_2, \dots, i_J$, and the transport plan is in the same way described by a J-mode tensor $\mathbf{B} \in \mathbb{R}_+^{d_1\times d_2\times\cdots\times d_J}$.

For a given transport plan $\mathbf{B}$, the total cost of transportation is

$$\langle \mathbf{C}, \mathbf{B}\rangle := \sum_{i_1, i_2, \dots, i_J} C(i_1, i_2, \dots, i_J) B(i_1, i_2, \dots, i_J).$$

Thus, similar to (1), MOT has a linear programming formulation

$$\begin{aligned} \min_{\mathbf{B}\in\mathbb{R}_+^{d_1\times\cdots\times d_J}} & \langle \mathbf{C}, \mathbf{B}\rangle \\ \text{subject to } & P_j(\mathbf{B}) = \boldsymbol{\mu}_j, \ \text{ for } j \in \Gamma, \end{aligned} \tag{19}$$

where $\Gamma \subset \{1, 2, \dots, J\}$ is an index set specifying which marginal distributions are given, and the projection on the j-th marginal of $\mathbf{B}$ is computed as

$$P_j(\mathbf{B})(i_j) = \sum_{i_1,\dots,i_{j-1},i_{j+1},i_J} B(i_1, \dots, i_{j-1}, i_j, i_{j+1}, \dots, i_J). \tag{20}$$

Note that the standard bi-marginal OT problem (1) is a special case of the MOT problem (19) with $J = 2$ and $\Gamma = \{1, 2\}$.

In the original MOT formulation [12], [15], constraints are given on all the marginal distributions, viz., the index set $\Gamma = \{1, 2, \dots, J\}$. However, in many applications [15], [16], [52], only a subset of marginal distributions are explicitly given. For instance, the Barycenter problem [53] is a MOT where the target distribution is not given. In this work we consider the setting where constraints are only imposed on a subset of marginals, i.e., $\Gamma \subset \{1, 2, \dots, J\}$.

A. Entropy regularized MOT

Although MOT (19) is a standard linear program, its complexity grows exponentially as J increases. This computational burden can be partly alleviated in an analogous manner as for the classical bi-marginal problem (1), which again yields an iterative scaling algorithm. In particular, similarly to (3), one can add an entropy term

$$\mathcal{H}(\mathbf{B}) = -\sum_{i_1,\dots,i_J} B(i_1, \dots, i_J)\ln\ B(i_1, \dots, i_J) \tag{21}$$

to (19) to regularize the problem, resulting in the strictly convex optimization problem

$$\begin{aligned} \min_{\mathbf{B}\in\mathbb{R}^{d_1\times\cdots\times d_J}} & \langle \mathbf{C}, \mathbf{B}\rangle - \epsilon\mathcal{H}(\mathbf{B}) \\ \text{subject to } & P_j(\mathbf{B}) = \boldsymbol{\mu}_j, \ \text{ for } j \in \Gamma \end{aligned} \tag{22}$$

with $\epsilon > 0$ being a regularization parameter.

For the bi-marginal case, (22) reduces to problem (3). The iterative scaling algorithm (4) can be generalized to the multi-marginal setting [46] in order to solve (22). From an optimization perspective, the iterative scaling algorithm amounts to a coordinate ascent method [54] in the dual problem of (22). The introduction of the entropy term in (22) allows for closed-form expressions for the updates of the dual variables [17]. Utilizing Lagrangian duality theory, one can show that the optimal solution to (22) is of the form

$$\mathbf{B} = \mathbf{K} \odot \mathbf{U}, \tag{23}$$

where $\odot$ denotes element-wise multiplication and the tensors are given by

$$\mathbf{K} = \exp(-\mathbf{C}/\epsilon) \tag{24}$$

and

$$\mathbf{U} = \mathbf{u}_1 \otimes \mathbf{u}_2 \otimes \cdots \otimes \mathbf{u}_J, \tag{25}$$

where the vectors $\mathbf{u}_j \in \mathbb{R}^{d_j}$ are given by

$$\mathbf{u}_j = \begin{cases} \exp\left(-\frac{1}{J} - \frac{\boldsymbol{\lambda}_j}{\epsilon}\right), & \text{if } j \in \Gamma \\ \exp\left(-\frac{1}{J}\right)\mathbf{1}, & \text{otherwise,} \end{cases} \tag{26}$$

and $\boldsymbol{\lambda}_j \in \mathbb{R}^{d_j}$ is the dual variable corresponding to the constraint $P_j(\mathbf{B}) = \boldsymbol{\mu}_j$ on the j-th marginal. Moreover, the dual of (22) is

$$\max_{\{\boldsymbol{\lambda}_j, j\in\Gamma\}} -\epsilon\langle \mathbf{K}, \mathbf{U}\rangle - \sum_{j\in\Gamma} \boldsymbol{\lambda}_j^T \boldsymbol{\mu}_j. \tag{27}$$

We emphasis that in (27), $\mathbf{U}$ is a function of the multipliers $\{\boldsymbol{\lambda}_j, j \in \Gamma\}$ as defined in (25)-(26).

The iterative scaling algorithm iteratively updates the vectors $\mathbf{u}_j$, for $j \in \Gamma$, in (26) according to

$$\mathbf{u}_j \leftarrow \mathbf{u}_j \odot \boldsymbol{\mu}_j./P_j(\mathbf{K} \odot \mathbf{U}). \tag{28}$$

For future reference, we summarize the steps in Algorithm 1. The Iterative Scaling algorithm (Algorithm 1) is a special case of the iterative Bregman projection algorithm [55], [23], which itself is a special case of a dual block coordinate ascent method [56], [57], and thus enjoys a global convergence guarantee [55], [57].

Note that the standard Sinkhorn iterations (4) for the two-marginal case (3) is a special case of Algorithm 1 when $J = 2$ and $\Gamma = \{1, 2\}$. Indeed, in this case, Algorithm 1 boils down to iterating

$$\mathbf{u}_1 \leftarrow \mathbf{u}_1 \odot \boldsymbol{\mu}_1./P_1(\mathbf{K} \odot \mathbf{U}), \qquad \mathbf{u}_2 \leftarrow \mathbf{u}_2 \odot \boldsymbol{\mu}_2./P_2(\mathbf{K} \odot \mathbf{U}).$$

With

$$\begin{aligned} P_1(\mathbf{K} \odot \mathbf{U}) &= \operatorname{diag}(\mathbf{u}_1)\mathbf{K}\operatorname{diag}(\mathbf{u}_2)\mathbf{1} \\ &= \operatorname{diag}(\mathbf{u}_1)(\mathbf{K}\mathbf{u}_2) = \mathbf{u}_1 \odot (\mathbf{K}\mathbf{u}_2) \end{aligned}$$

and similarly $P_2(\mathbf{K} \odot \mathbf{U}) = \mathbf{u}_2 \odot (\mathbf{K}^T\mathbf{u}_1)$, it follows

$$\mathbf{u}_1 \leftarrow \boldsymbol{\mu}_1./\mathbf{K}\mathbf{u}_2, \qquad \mathbf{u}_2 \leftarrow \boldsymbol{\mu}_2./\mathbf{K}^T\mathbf{u}_1, \tag{29}$$

which coincide with (4).

Although Algorithm 1 is easy to implement and considerably faster than general linear programming solvers, its complexity still scales exponentially as J grows since the number of elements in $\mathbf{B}$ are $d_1 d_2 \dots d_J$. The computational bottleneck of it lies in the calculation of the projections $P_j(\mathbf{B})$, for $j \in \Gamma$, in (20). Generally, this computational burden is inevitable. However, in some cases it is possible to utilize graph structures in the cost tensor $\mathbf{C}$ to compute the projections efficiently [23], [17], [52], [58]. In Section III-B we consider MOT problems with cost tensors that can be decomposed according to a graph. This graphical structure allows us to leverage the Bayesian inference tools [26] in PGMs to compute the projections efficiently. Other than providing a workhorse for solving MOT problems with graphical structured cost, this connection between MOT and PGMs also presents new elements and perspective to Bayesian inference in PGMs, which is discussed in details in Section III-C.

B. MOT with graphical structures

Consider the cases where the cost tensor $\mathbf{C}$ can be decomposed according to a factor graph. More specifically, the cost tensor $\mathbf{C}$ has the form

$$C(\mathbf{x}) = \sum_{\alpha\in F} C_\alpha(\mathbf{x}_\alpha), \tag{30}$$

where F denotes the set of factors of a graph. Here, to be consistent with the notations in PGMs, we write the cost of associating $i_1, i_2, \dots, i_J$ by $C(\mathbf{x}) = C(x_1, x_2, \dots, x_J)$ instead of $C(i_1, i_2, \dots, i_J)$, but the two have exactly the same meaning; both x_j and i_j take values in a set with d_j elements. Thus, by abuse of notation, we use $C(\mathbf{x})$ and $C(i_1, i_2, \dots, i_J)$ interchangeably.

A graph structured cost tensor (30) occurs in various applications of the OT framework [23], [17]. For instance, in Barycenter problems [53], the cost $\mathbf{C}$ can be decomposed into the sum of pairwise costs between the target distribution and each given marginal distribution. For general cost functions, it might be possible to approximate them using the structured cost (30). Thus the framework we establish can potentially be used to approximate the solutions to many MOT problems.

Remark 2. *The idea of leveraging the structure in the cost* $\mathbf{C}$ *to accelerate OT algorithms has been explored before. In [59], [60], [61], [62], low-rank approximations of the cost matrix and the corresponding kernel* (24) *have been utilized to solve* **bi-marginal OT** *problems more efficiently. A key idea used in these works is that the matrix-vector multiplications* $\mathbf{K}\mathbf{u}, \mathbf{K}^T\mathbf{u}$ *in* (29) *can be accelerated if* $\mathbf{K}$ *is a low-rank matrix. This line of research is fundamentally different to ours which aims to leverage the existing graphical structures of cost tensor in* **MOT** *problems. In fact, they are complementary to each other. Our algorithm can be further improved by applying the results in [61], [62], assuming* $\mathbf{C}_\alpha$ *in* (30) *are (approximately) low-rank.*

Denote the factor graph associated with the cost (30) by $G = (V, F, E)$. Then the j-th mode of $\mathbf{C}$ corresponds to node $j \in V$ and the marginal distribution of the j-th mode is the same as the marginal distribution of x_j at node j. In this paper, we only consider the cases where the factor graph G is connected but does not have any loop, that is, G is a factor tree. We associate the cost $\mathbf{C}$ with a probabilistic graphical model

$$p(\mathbf{x}) = \frac{1}{Z}\prod_{\alpha\in F} K_\alpha(\mathbf{x}_\alpha)$$

where

$$K_\alpha(\mathbf{x}_\alpha) = \exp(-C_\alpha(\mathbf{x}_\alpha)/\epsilon). \tag{31}$$

Clearly, $\mathbf{K}$ in (24) has the form

$$\mathbf{K} = [K(i_1, i_2, \dots, i_J)] = [K(\mathbf{x})] = \left[\prod_{\alpha\in F} K_\alpha(\mathbf{x}_\alpha)\right],$$

and

$$\mathbf{K}\odot\mathbf{U} = [K(\mathbf{x})U(\mathbf{x})] = \left[\left(\prod_{\alpha\in F} K_\alpha(\mathbf{x}_\alpha)\right)\left(\prod_{j\in V} u_j(x_j)\right)\right]. \tag{32}$$

From a PGM point of view, the (transformed) Lagrangian multipliers $\mathbf{u}_j$, for $j \in \Gamma$, introduced by the Iterative Scaling algorithm are local potentials of the modified graphical model $K(\mathbf{x})U(\mathbf{x})$. The Lagrangian approach of solving the constrained optimization problem (22) seeks multipliers $\mathbf{u}_j$, for $j \in \Gamma$, such that the tensor $\mathbf{B} = \mathbf{K} \odot \mathbf{U}$ satisfies all the constraints $P_j(\mathbf{B}) = \boldsymbol{\mu}_j$, for $j \in \Gamma$. Thus, in the language of

Algorithm 1 Iterative Scaling Algorithm for MOT

```
Compute $\mathbf{K} = \exp(-\mathbf{C}/\epsilon)$
Initialize $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_J$ to $\exp(-\frac{1}{J})\mathbf{1}$
while not converged do
  for $j \in \Gamma$ do
    Compute $\mathbf{U} = \mathbf{u}_1 \otimes \mathbf{u}_2 \otimes \cdots \otimes \mathbf{u}_J$
    Update $\mathbf{u}_j$ as $\mathbf{u}_j \leftarrow \mathbf{u}_j \odot \boldsymbol{\mu}_j ./ P_j(\mathbf{K} \odot \mathbf{U})$
  end for
end while
```

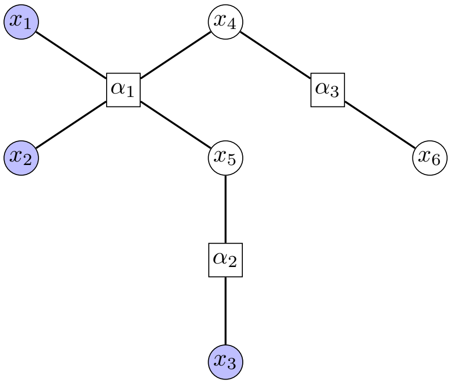


Fig. 2: Factor graph with some marginal distribution constraints on nodes $\{x_1, x_2, x_3\}$.

PGMs, to solve the MOT problem (22), one can search for a proper set of artificial local potentials $\mathbf{u}_j$, for $j \in \Gamma$, such that the modified graphical model $K(\mathbf{x})U(\mathbf{x})$ in (32) has the specified marginal distribution $\boldsymbol{\mu}_j$ on the j-th variable node for each $j \in \Gamma$. Note that $\mathbf{u}_j = \exp(-1/J)\mathbf{1}$ is a uniform potential for all $j \notin \Gamma$ and thus does not affect the graphical model $\mathbf{K} \odot \mathbf{U}$.

For fixed multipliers $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_J$, calculating (with proper normalization) the projection $P_j(\mathbf{K} \odot \mathbf{U})$ is exactly a Bayesian inference problem of inferring the j-th variable node over the modified graphical model $K(\mathbf{x})U(\mathbf{x})$. When G does not have any loops, a condition we assume throughout, Bayesian inference can be achieved efficiently using the Belief Propagation algorithm. Generally, the marginal constraints $P_j(\mathbf{B}) = \boldsymbol{\mu}_j$ can be imposed on any variable node $j \in V$. However, a marginal constraint on a non-leaf node will decompose the MOT problem (22) into several independent MOT problems with constraints only on leaf nodes, see [52]. Thus, without loss of generality, we assume marginal constraints on leaf nodes only, that is, $\Gamma \subset L$ where $L \subset V$ denotes the set of leaf nodes of G.

Example 3. *Figure 2 depicts a factor graph with leaf nodes $L = \{1, 2, 3, 6\}$. The shaded nodes in the figure represent the fixed distribution variables, thus, in this example, $\Gamma = \{1, 2, 3\} \subset L$.*

Leveraging the graphical structure (30) of the cost, based on the discussions above, we obtain a simple strategy to solve the MOT problem (22): We apply the Iterative Scaling algorithm and utilize the Belief Propagation algorithm to carry out the computation of $P_j(\mathbf{K} \odot \mathbf{U})$ with the current multiplier $\mathbf{U}$. The acceleration is tremendous for MOT problems with a large number of marginals; the Belief Propagation algorithm scales well for large problem while the complexity of the brute force projection using definition (20) grows exponentially as the dimension increases. It turns out that some more tricks can be adopted to further improve the speed of the projection step $P_j(\mathbf{K} \odot \mathbf{U})$. The full algorithm, which we call Iterative Scaling Belief Propagation (ISBP) algorithm will be presented and discussed in details in Section III-D.

C. MOT and Bayesian inference

In the previous section, we have seen that in cases where the cost tensor $\mathbf{C}$ in MOT problem (22) has a graphical structure, one can take advantage of PGM methods, in particular the Belief Propagation algorithm, to accelerate the Iterative Scaling algorithm. In this section, we establish further connections between MOT and PGMs. These links add novel components to both the MOT theory and PGM theory. These connections also bring new insight and interpretation of Iterative Scaling Belief Propagation.

Clearly, the objective function of the entropy regularized MOT problem (22) is exactly the free energy $\mathcal{F}$ in (11) with

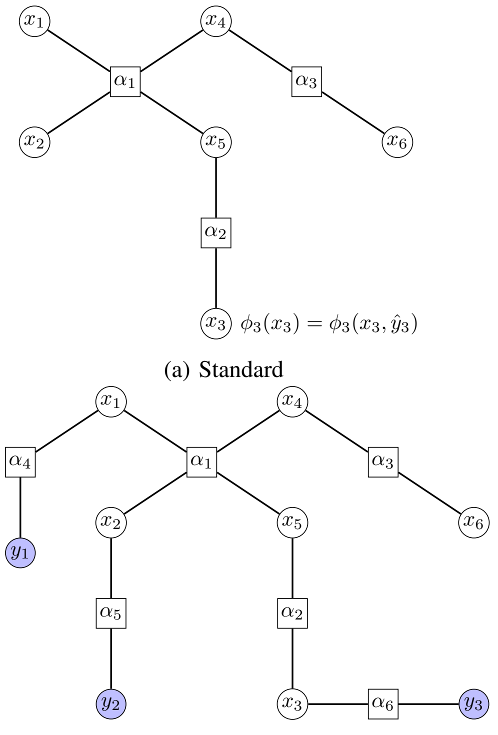
$$\begin{aligned} &\psi_\alpha(\mathbf{x}_\alpha) = \exp(-C_\alpha(\mathbf{x}_\alpha)),\ \forall \alpha \in F, \forall \mathbf{x}_\alpha \\ &\phi_j(x_j) = 1,\ \forall j \in V, \forall x_j. \end{aligned} \tag{33}$$

Thus, the MOT problem (22) can be written as

$$\begin{aligned} \min_{\mathbf{B} \in \mathbb{R}_+^{d_1 \times \cdots \times d_J}} \ & \mathcal{F}(\mathbf{B}) \\ \text{subject to } & P_j(\mathbf{B}) = \boldsymbol{\mu}_j, \quad \forall j \in \Gamma. \end{aligned} \tag{34}$$

Therefore, the entropic regularized MOT problem (22) with cost function that decouples according to a graph structure as in (30) is equivalent to a Bayesian inference problem in a PGM with additional constraints on the marginal distributions of a set of variable nodes. In other words, (34) is a constrained version of a Bayesian inference problem.

On the other hand, any Bayesian inference problem in a PGM can be rewritten in the constrained form (34). More specifically, consider the problem of inferring the posterior distribution of $\frac{1}{Z}\prod_{j\in\Gamma}\phi_j(x_j, y_j)\prod_{\alpha\in F}\psi_\alpha(\mathbf{x}_\alpha)$, where y_j is the observation associated with the variable node potential ϕ_j. When the observations are fixed, say $y_j = \hat{y}_j$, then the standard Bayesian inference method replaces $\phi_j(x_j, y_j)$ by a local potential $\phi_j(x_j)$ and infers the marginal distributions of the resulting graphical model with only the nodes $\{x_j,\ \mathbf{x}_\alpha\}$.



(a) Standard

(b) Constrained

Fig. 3: Equivalence between standard Bayesian inference and constrained Bayesian inference: (a) The local potentials of variables $x_1,\ x_2,\ x_3$ are induced by measurements $y_1 = \hat{y}_1,\ y_2 = \hat{y}_2,\ y_3 = \hat{y}_3$ respectively, namely, $\phi_1(x_1) = \phi_1(x_1, \hat{y}_1),\ \phi_2(x_2) = \phi_2(x_2, \hat{y}_2),\ \phi_3(x_3) = \phi_3(x_3, \hat{y}_3)$; (b) The graph is augmented by the nodes $y_1,\ y_2,\ y_3$ and factors $\psi_{\alpha_4} = \phi_1(x_1, y_1),\ \psi_{\alpha_5} = \phi_2(x_2, y_2)$, and $\psi_{\alpha_6} = \phi_3(x_3, y_3)$. The measurements become marginal constraints $p(y_1) = \delta(y_1 - \hat{y}_1),\ p(y_2) = \delta(y_2 - \hat{y}_2),\ p(y_3) = \delta(y_3 - \hat{y}_3)$.

Alternatively, the measurement $y_j = \hat{y}_j$ can be viewed as constraints on the node y_j of an augmented graphical model which includes also the observation variable node y_j. In particular, the constraint is of the form $p(y_j) = \delta(y_j - \hat{y}_j)$, where δ denotes Dirac distribution. Thus, the posterior distribution can also be obtained by solving the constrained Bayesian inference problem (34) over the augmented graphical models under the constraints that $p(y_j) = \delta(y_j - \hat{y}_j)$, for $j \in \Gamma$. This equivalence is illustrated in Figure 3. Therefore, from this point of view, the constrained Bayesian inference problem (34) can also be viewed as a generalization of standard Bayesian inference.

When the underlying factor graph associated with the cost function (30) is in fact a factor tree, the free energy $\mathcal{F}(\mathbf{B})$ is equal to the Bethe free energy (see Section II-C)

$$\begin{aligned} \mathcal{F}_{\mathrm{Bethe}}(\mathbf{B}) &= -\sum_{\alpha \in F} \sum_{\mathbf{x}_\alpha} B_\alpha(\mathbf{x}_\alpha) \ln \psi_\alpha(\mathbf{x}_\alpha) \qquad (35) \\ &+ \epsilon \sum_{\alpha \in F} \sum_{\mathbf{x}_\alpha} B_\alpha(\mathbf{x}_\alpha) \ln B_\alpha(\mathbf{x}_\alpha) \\ &- \epsilon \sum_{j \in V} (N_j - 1) \sum_{x_j} B_j(x_j) \ln B_j(x_j), \end{aligned}$$

where $\mathbf{B}_\alpha$ is the marginal distribution on factor node α and $\mathbf{B}_j$ denotes the marginal on variable node j, namely, $B_\alpha(\mathbf{x}_\alpha) = \sum_{\mathbf{x} \setminus \mathbf{x}_\alpha} B(\mathbf{x})$, and $B_j(x_j) = \sum_{\mathbf{x} \setminus x_j} B(\mathbf{x})$. In (35), $\mathbf{B}$ is the collection of marginal distributions, that is, $\mathbf{B} = \{\mathbf{B}_j, \mathbf{B}_\alpha \ :\ j \in V, \alpha \in F\}$. This is different from the J-mode tensor $\mathbf{B}$ in (34). However, with slight abuse of notation, we use the same symbol $\mathbf{B}$ in both contexts. Again, due to tree structure, the two have a one-to-one correspondence with each other.

The marginals $\mathbf{B}_\alpha$ and $\mathbf{B}_j$ capture only local information around a factor variable or node variable and they have to satisfy certain conditions in order to be feasible marginal distributions of some joint distribution. In particular, they have to be compatible in the sense

$$\sum_{\mathbf{x}_\alpha \setminus x_j} B_\alpha(\mathbf{x}_\alpha) = B_j(x_j), \quad \forall j \in V, \forall x_j.$$

In terms of local marginals, the constraints in (34) read

$$B_j(x_j) = \mu_j(x_j), \quad \forall j \in \Gamma, \forall x_j.$$

Therefore, the MOT problem (34) can be reformulated as

$$\begin{aligned} \min_{\mathbf{B}} \quad & \mathcal{F}_{\mathrm{Bethe}}(\mathbf{B}) && (36a) \\ \text{subject to} \quad & B_j(x_j) = \mu_j(x_j), \quad \forall j \in \Gamma,\ \forall x_j && (36b) \\ & \sum_{\mathbf{x}_\alpha \setminus x_j} B_\alpha(\mathbf{x}_\alpha) = B_j(x_j), \forall j \in V, \alpha \in N(j), \forall x_j && (36c) \\ & \sum_{\mathbf{x}_\alpha} B_\alpha(\mathbf{x}_\alpha) = 1, \forall \alpha \in F, && (36d) \end{aligned}$$

where the last constraint (36d) is to ensure that the optimization variables $\{\mathbf{B}_j, \mathbf{B}_\alpha \ :\ j \in V, \alpha \in F\}$ are in the probability simplex. Since the Bethe free energy is convex for factor trees, and the constraints are linear, Problem (36) is a convex optimization problem. One advantage of (36) over (34) is that the size of optimization variables in (36) is considerably smaller than that in (34). More specifically, the optimization variables of (36) are local marginals which are either vectors $\mathbf{B}_j$ or low-dimensional tensors $\mathbf{B}_\alpha$, which is in contrast to the high-dimensional J-mode tensor $\mathbf{B}$ in (34).

D. Iterative Scaling Belief Propagation algorithm

In this section, we present the full Iterative Scaling Belief Propagation algorithm for the entropy regularized MOT problem (22) (or equivalently (34) and (36)). We start with a characterization of the solution to (36).

Theorem 4. *The solution to the MOT problem* (36) *is given by*

$$\begin{aligned} B_\alpha(\mathbf{x}_\alpha) &\propto K_\alpha(\mathbf{x}_\alpha) \prod_{j \in N(\alpha)} n_{j \to \alpha}(x_j),\ \forall \alpha \in F && (37a) \\ B_j(x_j) &\propto \prod_{\alpha \in N(j)} m_{\alpha \to j}(x_j),\ \forall j \notin \Gamma && (37b) \\ B_j(x_j) &= \mu_j(x_j),\ \forall j \in \Gamma && (37c) \end{aligned}$$

where $m_{\alpha\to j}$, $n_{j\to\alpha}$ are fixed points of the following iterations

$$m_{\alpha\to j}(x_j) \propto \sum_{\mathbf{x}_\alpha\backslash x_j} K_\alpha(\mathbf{x}_\alpha) \prod_{i\in N(\alpha)\backslash j} n_{i\to\alpha}(x_i); \forall \alpha\in N(j), \tag{38a}$$

$$n_{j\to\alpha}(x_j) \propto \prod_{\beta\in N(j)\backslash\alpha} m_{\beta\to j}(x_j); \forall j\notin\Gamma, \forall\alpha\in N(j), \tag{38b}$$

$$n_{j\to\alpha}(x_j) \propto \mu_j(x_j)(m_{\alpha\to j}(x_j))^{-1}; \forall j\in\Gamma, \alpha\in N(j). \tag{38c}$$

Here $\propto$ indicates that a normalization step is needed.

Proof. In order to solve the constrained optimization problem (36), we introduce Lagrange multipliers η_α for the simplex constraints (36d), $\boldsymbol{\lambda}_{j,\alpha}$ for the marginalization compatibility constraints (36c), and $\boldsymbol{\nu}_j$ for the fixed-marginal constraints (36b), yielding the Lagrangian

$$\begin{aligned}\mathcal{L} &= \frac{1}{\epsilon}\mathcal{F}_{\text{Bethe}}(\mathbf{B}) + \sum_\alpha \eta_\alpha \left(\sum_{\mathbf{x}_\alpha} B_\alpha(\mathbf{x}_\alpha) - 1\right) \\ &+ \sum_{j,x_j}\sum_{\alpha\in N(j)} \lambda_{j,\alpha}(x_j)\left(\sum_{\mathbf{x}_\alpha\backslash x_j} B_\alpha(\mathbf{x}_\alpha) - B_j(x_j)\right) \\ &+ \sum_{j\in\Gamma}\sum_{x_j} \nu_j(x_j)\left(B_j(x_j) - \mu_j(x_j)\right).\end{aligned} \tag{39}$$

Note that we have used a scaled version $\frac{1}{\epsilon}\mathcal{F}_{\text{Bethe}}$ of the objective function. In view of (33), (35) and (31),

$$\begin{aligned}&\frac{1}{\epsilon}\mathcal{F}_{\text{Bethe}}(\mathbf{B}) = -\sum_{j\in V}(N_j-1)\sum_{x_j} B_j(x_j)\ln B_j(x_j) \\ &+ \sum_{\alpha\in F}\sum_{\mathbf{x}_\alpha} B_\alpha(\mathbf{x}_\alpha)\ln B_\alpha(\mathbf{x}_\alpha) - \sum_{\alpha\in F}\sum_{\mathbf{x}_\alpha} B_\alpha(\mathbf{x}_\alpha)\ln K_\alpha(\mathbf{x}_\alpha).\end{aligned}$$

Setting the derivatives of the Lagrangian with respect to the local marginals $\mathbf{B}_\alpha$ and $\mathbf{B}_j$ to zero, we get that the minimizer satisfies

$$B_\alpha(\mathbf{x}_\alpha) = K_\alpha(\mathbf{x}_\alpha)\exp\left(-1 - \sum_{j\in N(\alpha)} \lambda_{j,\alpha}(x_j) - \eta_\alpha\right) \tag{40a}$$

$$B_j(x_j) = \exp\left(-1 - \frac{1}{N_j-1}\sum_{\alpha\in N(j)} \lambda_{j,\alpha}(x_j)\right) \text{if } N_j > 1 \tag{40b}$$

$$0 = \sum_{\alpha\in N(j)} \lambda_{j,\alpha}(x_j), \text{if } N_j = 1, j\notin\Gamma \tag{40c}$$

$$0 = \sum_{\alpha\in N(j)} \lambda_{j,\alpha}(x_j) - \nu_j(x_j) \quad \text{if } j\in\Gamma \tag{40d}$$

Denote

$$n_{j\to\alpha}(x_j) := \exp(-\lambda_{j,\alpha}(x_j)) \tag{41a}$$

$$m_{\alpha\to j}(x_j) := \sum_{\mathbf{x}_\alpha\backslash x_j} K_\alpha(\mathbf{x}_\alpha)\prod_{i\in N(\alpha)\backslash j} n_{i\to\alpha}(x_i). \tag{41b}$$

The relation (37a) follows immediately from (40a) and (41a). This together with the constraint (36c) and (41b) leads to

$$B_j(x_j) = \sum_{\mathbf{x}_\alpha\backslash x_j} B_\alpha(\mathbf{x}_\alpha) \propto n_{j\to\alpha}(x_j)m_{\alpha\to j}(x_j). \tag{42}$$

By (40c), we obtain

$$n_{j\to\alpha}(x_j) = 1 \text{ if } N_j = 1, j\notin\Gamma.$$

It follows that

$$B_j(x_j) \propto m_{\alpha\to j}(x_j) \text{ if } N_j = 1, j\notin\Gamma,$$

which is (37b) for leaf nodes. We next show (37b) when $N_j > 1$. To this end, we plug (40a) and (40b) into the constraint (36c) and arrive at

$$\begin{aligned} n_{j\to\gamma}(x_j)m_{\gamma\to j}(x_j) &\propto \sum_{\mathbf{x}_\gamma\backslash x_j} B_\gamma(\mathbf{x}_\gamma) \\ = \quad & B_j(x_j) \propto \exp\left(-1 - \frac{1}{N_j-1}\sum_{\beta\in N(j)}\lambda_{j,\beta}(x_j)\right) \\ \propto \quad & \prod_{\beta\in N(j)} n_{j\to\beta}(x_j)^{\frac{1}{N_j-1}}.\end{aligned}$$

For fixed j, the above holds for all $\gamma\in N(j)$. Multiplying the above equation for all $\gamma\in N(j)\backslash\alpha$ yields

$$\prod_{\gamma\in N(j)\backslash\alpha}(n_{j\to\gamma}(x_j)m_{\gamma\to j}(x_j)) \propto \prod_{\beta\in N(j)} n_{j\to\beta}(x_j),$$

which is (38b) after canceling out equal terms. Thus, in view of (42), if $N_j > 1$,

$$B_j(x_j) \propto n_{j\to\alpha}(x_j)m_{\alpha\to j}(x_j) \propto \prod_{\alpha\in N(j)} m_{\alpha\to j}(x_j).$$

Finally, (37c) is clearly true due to constraints. This together with (42) leads to (38c), which completes the proof. □

The updates in (38) resemble the standard Belief Propagation algorithm (12). In particular, the updates (38a) and (38b) are exactly the same as (12). The update (38c) is new and is due to the constraints (36b) on the marginal distributions. Pictorially the message $\mathbf{m}_{\alpha\to j}$ sent to a constrained node j from node α bounces back to α, in form of $\mathbf{n}_{j\to\alpha}$. This is illustrated in Figure 4. The update (38c) in fact corresponds to the scaling step (28) of the Iterative Scaling algorithm (Algorithm 1). In particular, the multipliers $\{\mathbf{u}_j : j\in\Gamma\}$ in (26) relate to the messages as $\mathbf{u}_j = \mathbf{n}_{j\to\alpha}$, for $j\in\Gamma$. To see this, we note that the projection $P_j(\mathbf{K}\odot\mathbf{U})$ requires solving a Bayesian inference problem with respect to the modified graphical model

$$p(\mathbf{x}) = \frac{1}{Z}\prod_{\alpha\in F} K_\alpha(\mathbf{x}_\alpha)\prod_{j\in\Gamma} u_j(x_j). \tag{43}$$

Upon convergence of the Belief Propagation algorithm (12), it holds $P_j(\mathbf{K}\odot\mathbf{U}) = \mathbf{u}_j\mathbf{m}_{\alpha\to j}$, where α is the only factor node in $N(j)$ since $j\in\Gamma$ is a leaf node. Thus, the projection step (28) reads

$$\mathbf{u}_j\odot\boldsymbol{\mu}_j./P_j(\mathbf{K}\odot\mathbf{U}) = \boldsymbol{\mu}_j./\mathbf{m}_{\alpha\to j} = \mathbf{n}_{j\to\alpha}. \tag{44}$$

Therefore, the updates (38) contain all the components of our ISBP algorithm with (38a)-(38b) being the Belief Propagation part and (38c) being the Iterative Scaling part. ISBP is a scheduling of these updates in a certain order. As discussed in Section III-B, the key idea of ISBP is to

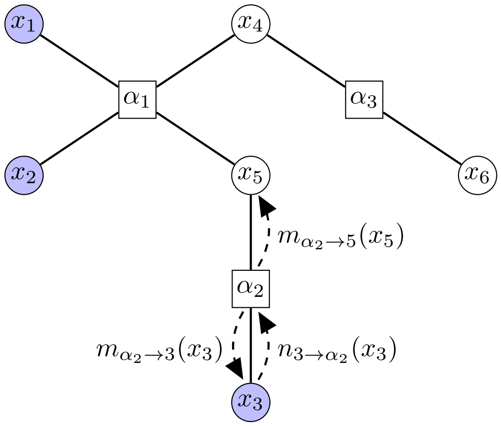


Fig. 4: Messages in ISBP

implement the projection $P_j(\mathbf{K}\odot\mathbf{U})$ in the iterative scaling step (28) using Belief Propagation. In the contexts of the updates (38), it is equivalent to run (38a)-(38b) sufficiently many iterations to obtain the precise projection $P_j(\mathbf{K}\odot\mathbf{U})$ and then run (38c), which is essentially (28). How many iterations of (38a)-(38b) are enough? One option is to run Belief Propagation over the whole graph G with the most recent modified model $\mathbf{K}\odot\mathbf{U}$ to compute $P_j(\mathbf{K}\odot\mathbf{U})$ for all $j\in V$. This is clearly sufficient but it is not necessary. Let $j_1, j_2, \ldots$ be a sequence taking values in Γ in some specific order and suppose the Iterative Scaling algorithm is carried out in this order. Then after the k-th step, $\mathbf{u}_{j_k}$ is updated, and the only projection required in the next step is $P_{j_{k+1}}(\mathbf{K}\odot\mathbf{U})$. It turns out that to evaluate $P_{j_{k+1}}(\mathbf{K}\odot\mathbf{U})$, it suffices to update all the messages on the path from j_k to j_{k+1}. Compared to the naive Belief Propagation over the whole graph, this local updating strategy is considerably faster. The steps of the ISBP algorithm are summarized in Algorithm 2. When $\{j_1, j_2, \ldots\}$ is a cyclic order of Γ, the ISBP corresponds to the standard iterative scaling algorithm (Algorithm 1). The order also can be chosen more aggressively according to the Gaussian-Southwell Rule [63] which updates the coordinates that leads to maximum improvement. Either way, the ISBP algorithm enjoys global linear convergence as described below.

Theorem 5. *Let $\{\mathbf{n}_{j\to\alpha}^{(k)}\}_{j\in\Gamma}$ be the set of messages*[1] *from the constraint set Γ after the k-th iteration in Algorithm 2. Then the sequence $\{\mathbf{n}_{j\to\alpha}^{(k)}\}_{j\in\Gamma}$ converges at least linearly as $k\to\infty$. Upon convergence, the solution to Problem* (36) *can be obtained through* (37)*. The belief tensor $\mathbf{B}$ (the solution to* (22)*) can also be obtained through $\mathbf{B}=\mathbf{K}\odot\mathbf{U}$ with $\mathbf{U}=\mathbf{u}_1\otimes\mathbf{u}_2\otimes\ldots\otimes\mathbf{u}_J$, where $\mathbf{u}_j=\mathbf{n}_{j\to\alpha}$ for $j\in\Gamma$, and $\mathbf{u}_j=\exp(-\frac{1}{J})\mathbf{1}$ otherwise.*

Proof. We identify the messages $\mathbf{n}_{j\to\alpha}$ with the updates in the multi-marginal iterative scaling algorithm as in (44). According to [52, Theorem 3.5] (see also [63]) the sequence of these updates $\{\mathbf{n}_{j\to\alpha}^{(k)}\}_{j\in\Gamma}$ converges linearly as $k\to\infty$. The limit point of Algorithm 2 coincides with the limit point of (38), and thus in the limit the solution to Problem (36) is given by (37). Finally, since we identify the free energy (11) with the entropy regularized MOT problem (22) as in (33), the optimal $\mathbf{B}$ is of the form $\mathbf{B}=\mathbf{K}\odot\mathbf{U}$, where $\mathbf{U}=\mathbf{u}_1\otimes\mathbf{u}_2\otimes\ldots\otimes\mathbf{u}_J$ as in (25). The components of $\mathbf{U}$ are given by $\mathbf{u}_j=\exp(-\frac{1}{J})\mathbf{1}$ for $j\in V\setminus\Gamma$, and from (44) it follows that in the limit point it holds $\mathbf{u}_j=\mathbf{n}_{j\to\alpha}$ for $j\in\Gamma$. □

Remark 6. *The ISBP algorithm is based on the iterative scaling/Sinkhorn algorithm and the belief propagation algorithm. Over years, many accelerated versions of the Sinkhorn algorithm have been developed, including the Greenkhorn algorithm [64], [65] and accelerated primal-dual gradient descent algorithm [66]. Combining these accelerations with belief propagation can be done in a similar way as in ISBP by realizing the marginal projections through belief propagation, and will lead to accelerated versions of ISBP.*

Remark 7. *ISBP can be used to find approximate solutions to the unregularized MOT problem* (19) *by choosing a sufficiently small ϵ as in [24]. An extra rounding step is needed in the end to recover a solution that is compatible with the marginals. Let $\mathbf{M}^*$ be an optimal solution to* (19)*, then a feasible solution $\mathbf{M}$ is called a δ-approximate solution for* (19)*, if $\langle\mathbf{C},\mathbf{M}\rangle\le\langle\mathbf{C},\mathbf{M}^*\rangle+\delta$. It is shown in [24, Algorithm 1] that it takes $\tilde{\mathcal{O}}\left(\delta^{-2}J^3\right)$ iterative scaling iterations (cf. [24, Appendix A.2]) to find a δ-approximate solution for unregularized MOT problems. Note that in order to decide the next greedy scaling step, the projections on all marginals need to be recomputed, which in general requires $\tilde{\mathcal{O}}(d^J)$ operations with $d=d_1=d_2=\cdots=d_J$. In contrast, utilizing tree-structures in the cost tensor, we can compute all projections in $\tilde{\mathcal{O}}(Jd^2)$. Thus, the computational complexity of using ISBP to solve the unregularized MOT with tree-structured cost to accuracy δ is $\tilde{\mathcal{O}}\left(\delta^{-2}J^4d^2\right)$. In the special case of Wasserstein barycenter problems, the complexity of ISBP is worse than existing state-of-the-art bounds $\tilde{\mathcal{O}}\left(\delta^{-4/3}Jd^{7/3}\right)$ in [67] and $\tilde{\mathcal{O}}\left(\delta^{-2}Jd^2\right)$ in [68]. However, $\tilde{\mathcal{O}}\left(\delta^{-2}J^4d^2\right)$ is a conservative bound which is a direct application of the results in [24]. We believe a tighter bound can be established by further leveraging the tree-structure of the cost and adopting accelerated Sinkhorn algorithms [64], [65], [66].*

IV. Constrained Norm-product algorithm

One potential drawback of the ISBP algorithm lies in the fact that it is a two-loop algorithm with the outer loop being iterative scaling and inner loop being belief propagation. Such a two-loop structure might slow down the convergence rate, especially when the underlying graph is large. Moreover, the two loops have to coordinate closely to guarantee convergence. Such coordination is even more difficult, or impossible if a distributed implementation is needed. Thus, we seek to develop a single loop algorithm for the entropy regularized MOT problems. A natural question to ask is whether we can borrow ideas from the Bayesian inference literature. After all, the Belief Propagation algorithm is not the only algorithm for Bayesian inference.

The answer is affirmative. In this section, we examine the Norm-product algorithm [36], another powerful Bayesian

[1] Note that since Γ is a subset of the leaf nodes, each element in Γ connects to a single factor node, that is, $\alpha\in N(j)$ is unique for each $j\in\Gamma$.

Algorithm 2 Iterative Scaling Belief Propagation (ISBP) Algorithm for MOT

```
Initialize the messages $m_{\alpha\to j}(x_j)$ and $n_{j\to\alpha}(x_j)$
Update $m_{\alpha\to j}(x_j)$ and $n_{j\to\alpha}(x_j)$ using (38a)-(38b) until convergence
while not converged do
    Update $n_{j_k\to\alpha}(x_{j_k})$ using (38c)
    Update all the messages on the path from node $j_k$ to node $j_{k+1}$ according to (38a) and (38b)
end while
```

inference method, and extend it to a single loop algorithm for our MOT problems. Below we first review the Norm-product algorithm for standard Bayesian inference problems in Section IV-A. The extensions to entropy regularized MOT, or equivalently, constrained Bayesian inference problems are presented in Section IV-B.

A. The Norm-product algorithm

Consider the Bayesian inference problem (8). The Norm-product algorithm [36] for Bayesian inference is based on the so called fractional entropy approximation

$$\mathcal{H}_{\rm frac}(\mathbf{b}) = \sum_{\alpha\in F}\bar{c}_\alpha\mathcal{H}(\mathbf{b}_\alpha) + \sum_{j\in V}\bar{c}_j\mathcal{H}(\mathbf{b}_j), \tag{45}$$

of the entropy $\mathcal{H}(\mathbf{b})$ in (10). The coefficients $\bar{c}_\alpha$, for $\alpha\in F$, and $\bar{c}_j$, for $j\in V$ are defined as

$$\begin{aligned}\bar{c}_\alpha &= c_\alpha + \sum_{j\in N(\alpha)} c_{j\alpha},\\ \bar{c}_j &= c_j - \sum_{\alpha\in N(j)} c_{j\alpha},\end{aligned} \tag{46}$$

for a set of real numbers c_α, c_j, and $c_{j\alpha}$, for $j\in V$ and $\alpha\in F$, which are known as counting numbers [69]. Clearly, an equivalent formulation of the fractional entropy (45) is

$$\begin{aligned}\mathcal{H}_{\rm frac}(\mathbf{b}) \;=\; & \sum_{\alpha\in F} c_\alpha\mathcal{H}(\mathbf{b}_\alpha) + \sum_{j\in V} c_j\mathcal{H}(\mathbf{b}_j) \\ & + \sum_{j\in V}\sum_{\alpha\in N(j)} c_{j\alpha}(\mathcal{H}(\mathbf{b}_\alpha)-\mathcal{H}(\mathbf{b}_j)).\end{aligned} \tag{47}$$

The fractional entropy resembles the Bethe entropy (17). In fact, for the choice of counting numbers $c_j = 1-N_j$, $c_{j\alpha}=0$, and $c_\alpha = 1$, the fractional entropy (47) reduces to the Bethe entropy. Moreover, just like the Bethe entropy, the fractional entropy approximation can be made exact when the underlying graph is a tree (see Section IV-D).

With the fractional entropy representation (47), the total free energy (11) is modified to the fractional free energy

$$\mathcal{F}_{\rm frac}(\mathbf{b}) = \mathcal{U}_{\rm frac}(\mathbf{b}) - \epsilon\,\mathcal{H}_{\rm frac}(\mathbf{b}), \tag{48}$$

where

$$\mathcal{U}_{\rm frac}(\mathbf{b}) = -\sum_{\alpha\in F}\sum_{\mathbf{x}_\alpha} b_\alpha(\mathbf{x}_\alpha)\ln\psi_\alpha(\mathbf{x}_\alpha) - \sum_{j\in V}\sum_{x_j} b_j(x_j)\ln\phi_j(x_j)$$

is the average energy defined as in (9). Thus, in terms of fractional free energy, the Bayesian inference problem (8) reads

$$\begin{aligned}\min_{\mathbf{b}} \quad & \mathcal{F}_{\rm frac}(\mathbf{b}) && \text{(49a)}\\ \text{subject to} \quad & \sum_{\mathbf{x}_\alpha\setminus x_j} b_\alpha(\mathbf{x}_\alpha) = b_j(x_j), \forall j\in V, \alpha\in N(j) && \text{(49b)}\\ & \sum_{\mathbf{x}_\alpha} b_\alpha(\mathbf{x}_\alpha) = 1, \forall\alpha\in F. && \text{(49c)}\end{aligned}$$

The constraints (49b)-(49c) are to ensure that $\mathbf{b}_\alpha, \mathbf{b}_j$ are indeed marginal distributions of some certain joint distribution. For a given graphical model, there are infinitely many different fractional free energy approximations determined by the counting numbers c_j, $c_\alpha, c_{j\alpha}$; some of them are convex and some of them are not. A sufficient condition for the convexity of the fractional free energy is as follows.

Lemma 8 ([36]). *When the counting numbers satisfy $c_j\geq 0$, $c_{j\alpha}\geq 0$, and $c_\alpha > 0$, for $j\in V$ and $\alpha\in F$, then the fractional free energy $\mathcal{F}_{\rm frac}$ is strictly convex over the set defined by the constraints* (49b)-(49c).

Denote the two sets corresponding to the constraints (49b) and (49c) by

$$\mathcal{M} = \Big\{\mathbf{b} : \sum_{\mathbf{x}_\alpha\setminus x_j} b_\alpha(\mathbf{x}_\alpha) = b_j(x_j), \forall j\in V, \alpha\in N(j)\Big\} \tag{50}$$

and

$$\mathcal{P} = \Big\{\mathbf{b} : \sum_{\mathbf{x}_\alpha} b_\alpha(\mathbf{x}_\alpha) = 1, \quad \forall\alpha\in F\Big\}, \tag{51}$$

respectively, and define

$$\hat{f}(\mathbf{b}) = -\sum_{\alpha\in F}\sum_{\mathbf{x}_\alpha} b_\alpha(\mathbf{x}_\alpha)\ln\psi_\alpha(\mathbf{x}_\alpha) - \sum_{\alpha\in F}\epsilon\, c_\alpha\mathcal{H}(\mathbf{b}_\alpha) \tag{52}$$

and

$$\begin{aligned}\hat{h}_j(\mathbf{b}) \;=\; & -\sum_{x_j} b_j(x_j)\ln\phi_j(x_j) - \epsilon\, c_j\mathcal{H}(\mathbf{b}_j) \\ & - \sum_{\alpha\in N(j)}\epsilon\, c_{j\alpha}(\mathcal{H}(\mathbf{b}_\alpha)-\mathcal{H}(\mathbf{b}_j)).\end{aligned} \tag{53}$$

Further denote $f(\mathbf{b}) = \hat{f}(\mathbf{b}) + \delta_{\mathcal{P}}(\mathbf{b})$ and $h_j(\mathbf{b}) = \hat{h}_j(\mathbf{b}) + \delta_{\mathcal{M}}(\mathbf{b})$ where δ is the indicator function. Then the Bayesian inference problem (49) can be reformulated as

$$\min_{\mathbf{b}} f(\mathbf{b}) + \sum_{j=1}^{J} h_j(\mathbf{b}). \tag{54}$$

In cases where $c_j\geq 0$, $c_{j\alpha}\geq 0$, and $c_\alpha > 0$, for $j\in V$ and $\alpha\in F$, by Lemma (8), $\hat{f}$ is strictly convex and $\hat{h}_j$ is convex for each $1 \le j \le J$ for $\mathbf{b} \in \mathcal{M} \cap \mathcal{P}$. The Norm-product algorithm relies on the reformulation (54). In particular, it leverages a powerful primal-dual ascent algorithm (stated below in Lemma 9) that is well studied in the convex optimization community to solve problem with the special structure of (54). The primal-dual ascent in Lemma 9 is derived from a more general algorithm known as dual block ascent [57] and thus inherits the nice convergence property of the latter. We refer the reader to [36] for more details on these algorithms.

Lemma 9 ([36]). *Consider the convex optimization problem* $\min f + \sum_{j=1}^{J} h_j$ *with* $f(\mathbf{b}) = \hat{f}(\mathbf{b}) + \delta_{\mathcal{B}}(\mathbf{b})$ *where* $\mathcal{B} = \{\mathbf{b} : A\mathbf{b} = \mathbf{c}\}$. *The* ***primal-dual ascent*** *algorithm initializes* $\boldsymbol{\lambda}_1 = 0, \ldots, \boldsymbol{\lambda}_J = 0$ *and repeatedly iterates the following steps for* $j = 1, \ldots, J$ *until convergence:*

$$\boldsymbol{\nu}_j \leftarrow \sum_{i \neq j} \boldsymbol{\lambda}_i \tag{55a}$$

$$\mathbf{b}^* \leftarrow \mathrm{argmin}_{\mathbf{b} \in dom(f) \cap dom(h_j)} \{ f(\mathbf{b}) + h_j(\mathbf{b}) + \mathbf{b}^T \boldsymbol{\nu}_j \} \tag{55b}$$

$$\boldsymbol{\lambda}_j \leftarrow -\boldsymbol{\nu}_j - \nabla \hat{f}(\mathbf{b}^*) + A^T \boldsymbol{\sigma} \text{ where } \boldsymbol{\sigma} \text{ is a vector.} \tag{55c}$$

Suppose $\hat{f}$ *is strictly convex and smooth, and* h_j, $j = 1, \ldots, J$ *are convex and continuous over their domains, then* $\mathbf{b}^*$ *in the above iteration converges to the unique global minimizer of* $f + \sum_{j=1}^{J} h_j$.

The Norm-product algorithm [36] (Algorithm 3) is a direct application of the primal-dual ascent algorithm to the formulation (54) of the Bayesian inference problem (49). It can be seen as a message-passing type algorithm for problem (49), where the dual variables $\boldsymbol{\lambda}_j$ in the primal-dual ascent algorithm work as "messages" between neighboring nodes. To see this, note that $h_j(\mathbf{b})$ depends only on $\mathbf{b}_\alpha$, where $\alpha \in N(j)$, and thus the corresponding dual variable $\boldsymbol{\lambda}_j$ depends only on $\mathbf{x}_\alpha$, where $\alpha \in N(j)$. This sparsity is encoded by the representation $\boldsymbol{\lambda}_j = \{\lambda_{j,\alpha}(\mathbf{x}_\alpha)\}$. The relation between the dual variables $\boldsymbol{\lambda}_j$ in Lemma 9 and messages in Algorithm 3 is given by $n_{j \to \alpha}(\mathbf{x}_\alpha) = \exp(-\lambda_{j,\alpha}(\mathbf{x}_\alpha))$.

For more details on the derivation of the Norm-product algorithm as a primal-dual ascent method, see [36]. Moreover, our development of the constrained Norm-product algorithm (Algorithm 4) is similar to this, and is provided in the appendix. Upon convergence of Algorithm 3, the solution to (49) has the form

$$b_j(x_j) \quad \propto \quad \left(\phi_j(x_j) \prod_{\alpha \in N(j)} m_{\alpha \to j}(x_j) \right)^{1/\epsilon \hat{c}_j}, \tag{56a}$$

$$b_\alpha(\mathbf{x}_\alpha) \quad \propto \quad \left(\psi_\alpha(\mathbf{x}_\alpha) \prod_{j \in N(\alpha)} n_{j \to \alpha}(\mathbf{x}_\alpha) \right)^{1/\epsilon c_\alpha}, \tag{56b}$$

where again $\propto$ indicates that a normalization step might be needed. Finally, note that for the special choice of counting numbers $c_j = 1 - N_j$, $c_{j\alpha} = 0$, and $c_\alpha = 1$, the Norm-product algorithm reduces to the Belief Propagation algorithm (12). However, note that this choice of counting number does not satisfy the conditions in Lemma 8 for convergence of the algorithm, although it is well known that the Belief Propagation algorithm is guaranteed to converge for trees [50]. Therefore, even though formally the Norm-product algorithm can be viewed as a unifying framework for many message-passing algorithms, its convergence proof is restricted in some sense due to the strong requirement on the counting numbers.

B. Constrained Norm-product algorithm

In this section, we develop a Norm-product type algorithm for the entropy regularized MOT problem (22), or equivalently the constrained Bayesian inference problem (34). Consider a modification of problem (49) with constrained marginal distributions, which reads

$$\min_{\mathbf{b}} \quad \mathcal{F}_{\text{frac}}(\mathbf{b}) \tag{57a}$$

$$\text{subject to} \quad b_j(x_j) = \mu_j(x_j), \quad \forall j \in \Gamma, \tag{57b}$$

$$\sum_{\mathbf{x}_\alpha \setminus x_j} b_\alpha(\mathbf{x}_\alpha) = b_j(x_j), \forall j \in V, \alpha \in N(j) \tag{57c}$$

$$\sum_{\mathbf{x}_\alpha} b_\alpha(\mathbf{x}_\alpha) = 1 \quad \forall \alpha \in F. \tag{57d}$$

Problem (57) can be seen in the light of the entropy regularized MOT problem formulated as a free energy minimization problem (34). In particular, if the free energy $\mathcal{F}$ is approximated by the fractional free energy $\mathcal{F}_{\text{frac}}$, then (34) becomes (57). Recall that in the MOT problem the factor and node potentials are $\boldsymbol{\psi}_\alpha = \exp(-\mathbf{C}_\alpha)$ and $\boldsymbol{\phi}_j \equiv 1$ (cf. (33)). However, the constrained Norm-product algorithm, which we develop in the following solves the Bayesian inference problem (57) for any potentials $\boldsymbol{\psi}_\alpha$ and $\boldsymbol{\phi}_j$.

Note that compared to (49), the modified problem (57) is only augmented by one linear constraint (57b). Thus, problem (57) can be formulated as in (54) by changing the set $\mathcal{M}$ in $h_j(\mathbf{b}) = \hat{h}_j(\mathbf{b}) + \delta_{\mathcal{M}}(\mathbf{b})$ to

$$\begin{aligned} \mathcal{M} = \{\mathbf{b} : & \sum_{\mathbf{x}_\alpha \setminus x_j} b_\alpha(\mathbf{x}_\alpha) = b_j(x_j), \forall j \in V, \alpha \in N(j), \\ & b_j(x_j) = \mu_j(x_j), \, \forall j \in \Gamma\}, \end{aligned}$$

instead of (50), and defining all other components of (54) as in (51)-(53). The primal-dual ascent algorithm in Lemma 9 can then be applied to (57). The resulting Constrained Norm-product (CNP) algorithm is presented in Algorithm 4. For a detailed derivation of the method see Appendix A. Upon convergence of Algorithm 4, the solution to (57) is of the form (56), as in the standard Norm-product algorithm. Moreover, the optimal marginal calculated through (56) satisfies the constraint $\mathbf{b}_j = \boldsymbol{\mu}_j$ for all $j \in \Gamma$. Algorithm 4 is presented for general constrained Bayesian inference problems (57). Recall that the entropy regularized MOT problem (34) is recovered as the special case, where the potentials are given by $\boldsymbol{\psi}_\alpha = \exp(-\mathbf{C}_\alpha)$ and $\boldsymbol{\phi}_j \equiv 1$, as in (33).

Compared with the standard Norm-product algorithm, the messages from variable nodes with marginal constraint to the neighboring factor nodes, i.e., $\mathbf{n}_{j \to \alpha}$, for $j \in \Gamma$, $\alpha \in N(j)$, depend not only on the incoming messages to j and α, but also the given marginal $\boldsymbol{\mu}_j$. Moreover, in the case when the marginal constraint (57b) is absent, namely, $\Gamma = \emptyset$,

Algorithm 3 The Norm-product Algorithm

```
Initialize $n_{j\to\alpha}(\mathbf{x}_\alpha) = 1$ for all $j = 1, \cdots, J$, $\alpha \in N(j)$ and $\mathbf{x}_\alpha$
while not converged do
  for $j = 1, 2, \ldots, J$ do
    $$m_{\alpha\to j}(x_j) = \left(\sum_{\mathbf{x}_\alpha \setminus x_j} \left(\psi_\alpha(\mathbf{x}_\alpha) \prod_{i\in N(\alpha)\setminus j} n_{i\to\alpha}(\mathbf{x}_\alpha)\right)^{1/\epsilon\hat{c}_{j\alpha}}\right)^{\epsilon\hat{c}_{j\alpha}} \quad, \ \forall\alpha \in N(j), \forall x_j$$
    $$n_{j\to\alpha}(\mathbf{x}_\alpha) \propto \left(\frac{\phi_j^{1/\hat{c}_j}(x_j)\prod_{\beta\in N(j)} m_{\beta\to j}^{1/\hat{c}_j}(x_j)}{m_{\alpha\to j}^{1/\hat{c}_{j\alpha}}(x_j)}\right)^{c_\alpha} \left(\psi_\alpha(\mathbf{x}_\alpha) \prod_{i\in N(\alpha)\setminus j} n_{i\to\alpha}(\mathbf{x}_\alpha)\right)^{-c_{j\alpha}/\hat{c}_{j\alpha}} \quad, \ \forall\alpha \in N(j), \forall \mathbf{x}_\alpha$$
  end for
end while
```

Algorithm 4 Constrained Norm-product (CNP) algorithm

```
Set $n_{j\to\alpha}(\mathbf{x}_\alpha) = 1$ for all $j = 1, \cdots, J$, $\alpha \in N(j)$ and $\mathbf{x}_\alpha$
while not converged do
  for $j = 1, 2, \ldots, J$ do
    $$m_{\alpha\to j}(x_j) = \left(\sum_{\mathbf{x}_\alpha \setminus x_j} \left(\psi_\alpha(\mathbf{x}_\alpha) \prod_{i\in N(\alpha)\setminus j} n_{i\to\alpha}(\mathbf{x}_\alpha)\right)^{1/\epsilon\hat{c}_{j\alpha}}\right)^{\epsilon\hat{c}_{j\alpha}} \quad, \ \forall\alpha \in N(j), \forall x_j$$
    if $j \notin \Gamma$ then
      $$n_{j\to\alpha}(\mathbf{x}_\alpha) \propto \left(\frac{\phi_j^{1/\hat{c}_j}(x_j)\prod_{\beta\in N(j)} m_{\beta\to j}^{1/\hat{c}_j}(x_j)}{m_{\alpha\to j}^{1/\hat{c}_{j\alpha}}(x_j)}\right)^{c_\alpha} \left(\psi_\alpha(\mathbf{x}_\alpha) \prod_{i\in N(\alpha)\setminus j} n_{i\to\alpha}(\mathbf{x}_\alpha)\right)^{-c_{j\alpha}/\hat{c}_{j\alpha}} \quad, \ \forall\alpha \in N(j), \forall \mathbf{x}_\alpha$$
    else if $j \in \Gamma$ then
      $$n_{j\to\alpha}(\mathbf{x}_\alpha) \propto \left(\frac{\mu_j(x_j)}{m_{\alpha\to j}^{1/\epsilon\hat{c}_{j\alpha}}(x_j)}\right)^{\epsilon c_\alpha} \left(\psi_\alpha(\mathbf{x}_\alpha) \prod_{i\in N(\alpha)\setminus j} n_{i\to\alpha}(\mathbf{x}_\alpha)\right)^{-c_{j\alpha}/\hat{c}_{j\alpha}} \quad, \ \forall\alpha \in N(j), \forall \mathbf{x}_\alpha$$
    end if
  end for
end while
```

Algorithm 4 reduces to the standard Norm-product belief algorithm 3.

Remark 10. *The message updates in Algorithm 4 can be problematic when the denominators become zero. This scenario can occur when either the factor or node potentials $\psi_\alpha(\mathbf{x}_\alpha)$ or $\phi_j(x_j)$ contain zero elements. Note that zero entries in the potential let the average energy* (9) *be unbounded if $b_j(x_j)$ or $b_\alpha(\mathbf{x}_\alpha)$ are nonzero on the corresponding entries. In implementations, this can be avoided by ignoring the updates involving zero denominators. See [36, Appendix F] for a more detailed discussions of this issue.*

C. Relations to Iterative Scaling Belief Propagation algorithm

Compared to the ISBP algorithm, the CNP algorithm is a single loop algorithm. Each iteration of Algorithm 4 requires visiting every variable node only once. In contrast, since Algorithm 2 has a double-loop structure and each inner-loop iteration requires updating throughout an entire path between two leaf nodes, the messages associated with most variable nodes will be updated multiple times in one iteration of the algorithm. Thus, the iteration complexity of the ISBP algorithm is higher than that of the CNP algorithm. This difference becomes more significant as the diameter/size of the underlying graph increases; for larger graphs, the inner-loop iteration of ISBP algorithm takes more updates. Apart from the iteration complexity, another potential advantage of the CNP algorithm is that its single loop structure allows for more flexible scheduling of the message passing/updating. In particular, it does not require any communication between inner and outer loop updates. Thus, it is easier to parallelize the Constrained Norm-product algorithm or develop a distributed version of it.

Recall from Section IV-A that the standard Norm-product method with counting numbers chosen as $c_\alpha = 1$, $c_j = 1 - N_j$ and $c_{j\alpha} = 0$ reduces to the standard Belief propagation method as given in (12). It turns out that similar results can be established to relate the Iterative Scaling Belief Propagation algorithm and the Constrained Norm-product algorithm. In particular, with this set of counting numbers, the constrained Norm-Product algorithm reads

$$m_{\alpha\to j}(x_j) = \sum_{\mathbf{x}_\alpha\setminus x_j}\left(\psi_\alpha(\mathbf{x}_\alpha)\prod_{i\in N(\alpha)\setminus j} n_{i\to\alpha}(x_j)\right), \forall\alpha\in N(j) \tag{58a}$$

$$n_{j\to\alpha}(x_j) \propto \left(\phi_j(x_j)\prod_{\beta\in N(i)\setminus\alpha} m_{\beta\to j}(x_j)\right), \forall j\notin\Gamma, \forall\alpha\in N(j) \tag{58b}$$

$$n_{j\to\alpha}(x_j) \propto \mu_j(x_j)(m_{\alpha\to j}(x_j))^{-1}, \forall j\in\Gamma, \forall\alpha\in N(j). \tag{58c}$$

Note that in general the messages $\mathbf{n}_{j\to\alpha}$ in the Constrained Norm-product algorithm depend on $\mathbf{x}_\alpha$, but for this special choice of counting numbers, they depend only on x_j. The messages (58) are exactly the same as the messages (38) in the ISBP algorithm. If the messages in (58) are scheduled in a specific way, then this becomes the Iterative Scaling Belief Propagation Algorithm 2. In particular, this is achieved by cycling through the nodes in Γ, where for two successive nodes $j_1, j_2 \in \Gamma$, one schedules the messages (58a) and (58b) on the path from j_1 to j_2, and finally the message $\mathbf{n}_{j_2\to\alpha}$ as in (58c). In this light, Algorithm 2 may not only be understood as Iterative scaling Belief propagation, but also as constrained Belief propagation, i.e., an extension of the standard Belief propagation method, where the marginals on some nodes are fixed.

What if we update the messages (58) following the scheduling of Algorithm 4? In fact, this is a single-loop version of the ISBP algorithm, and we have empirically observed good convergence properties of it. However, the choice of counting numbers $c_\alpha = 1$, $c_j = 1 - N_j$ and $c_{j\alpha} = 0$ does not yield a strictly convex objective function decomposition in the associated fractional variational inference problem (57) as discussed in Lemma 8. Thus, the convergence result for Algorithm 4 does not apply to this setting, and a global convergence proof remains an open problem.

D. Counting numbers of fractional entropy

One way to guarantee the convergence of the Constrained Norm-product algorithm is to choose the counting numbers for the fractional entropy $\mathcal{H}_{\text{frac}}$ such that they satisfy the convexity conditions in Lemma 8. Thus, a crucial question is whether, given a graphical model, such a choice of counting numbers exists, and how to find them. This question has been discussed in [36, Appendix E] where several optimization based methods have been proposed. In this section, we present a structured method to construct a feasible set of counting numbers that satisfy the assumptions in Lemma 8, viz., $c_j \geq 0, c_{j\alpha} \geq 0, c_\alpha > 0$, for factor graphs, which are trees. In particular, we provide a closed form expression for the choice of counting numbers, which makes parameter tuning for the Constrained-norm product algorithm simple and intuitive.

The fractional entropy decomposition requires the fractional entropy $\mathcal{H}_{\text{frac}}(\mathbf{b})$ to be equal to the entropy $\mathcal{H}(\mathbf{b})$, that is

$$\begin{aligned}\mathcal{H}(\mathbf{b}) = \mathcal{H}_{\text{frac}}(\mathbf{b}) &= \sum_{\alpha\in F} c_\alpha \mathcal{H}(\mathbf{b}_\alpha) + \sum_{j\in V} c_j \mathcal{H}(\mathbf{b}_j) \\ &+ \sum_{j\in V,\alpha\in N(j)} c_{j\alpha}(\mathcal{H}(\mathbf{b}_\alpha) - \mathcal{H}(\mathbf{b}_j)).\end{aligned}$$

On the other hand, for a factor tree, the entropy equals the Bethe entropy, namely,

$$\mathcal{H}(\mathbf{b}) = \mathcal{H}_{\text{Bethe}}(\mathbf{b}) = \sum_{\alpha\in F}\mathcal{H}(\mathbf{b}_\alpha) - \sum_{j\in V}(N_j - 1)\mathcal{H}(\mathbf{b}_j).$$

It follows that

$$\begin{aligned}&\sum_{j\in V}(1-N_j)\mathcal{H}(\mathbf{b}_j) + \sum_{\alpha\in F}\mathcal{H}(\mathbf{b}_\alpha) = \sum_{j\in V}(c_j - \sum_{\alpha\in N(j)} c_{i\alpha})\mathcal{H}(\mathbf{b}_j) \\ &+ \sum_{\alpha\in F}(c_\alpha + \sum_{i\in N(j)} c_{j\alpha})\mathcal{H}(\mathbf{b}_\alpha).\end{aligned}$$

Hence, by identifying the coefficients, we see that finding a set of feasible convex counting numbers is achieved by finding $c_\alpha > 0, c_j \geq 0, c_{j\alpha} \geq 0$ that satisfy the following equations

$$c_j - \sum_{\alpha\in N(j)} c_{j\alpha} = 1 - N_j, \tag{59a}$$

$$c_\alpha + \sum_{j\in N(\alpha)} c_{j\alpha} = 1. \tag{59b}$$

A direct consequence of (59) is

$$\sum_{j\in V} c_j + \sum_{\alpha\in F} c_\alpha = 1. \tag{60}$$

To see this, sum up (59a) and (59b) over all variable nodes and factor nodes. The left hand side becomes $\sum_{j\in V} c_j + \sum_{\alpha\in F} c_\alpha$ as all the terms $c_{j\alpha}$ get canceled. The right hand side becomes

$$\begin{aligned}\sum_{j\in V}(1-N_j) + \sum_{\alpha\in F} 1 &= -\sum_{j\in V} N_j + \sum_{j\in V} 1 + \sum_{\alpha\in F} 1 \\ &= -|E| + |V| = 1.\end{aligned}$$

The last equality is due to the fact that the factor graph is a tree (acyclic).

The property (60) can be generalized to subgraphs of G. Let (j_*, α_*) be any edge of G. If we cut this edge, then the tree G is split into two trees G_1 and G_2, where G_1 contains the variable node j_* and G_2 contains the factor node α_*. This is illustrated in Figure 5. Let $G_1 = (V_1, F_1, E_1)$, then

$$c_{j_*\alpha_*} = \sum_{j\in V_1} c_j + \sum_{\alpha\in F_1} c_\alpha. \tag{61}$$

This relation (61) can be established similarly to (60). It determines values for $c_{j\alpha}$ given $c_j, j\in V$ and $c_\alpha, \alpha\in F$. Moreover, it guarantees that $c_{j\alpha}$ is non-negative, as long as c_j and c_α are non-negative. Hence, based on (60) and (61), we obtain a remarkably simple strategy to get a set of convex counting numbers $c_\alpha > 0, c_j \geq 0, c_{j\alpha} \geq 0$.

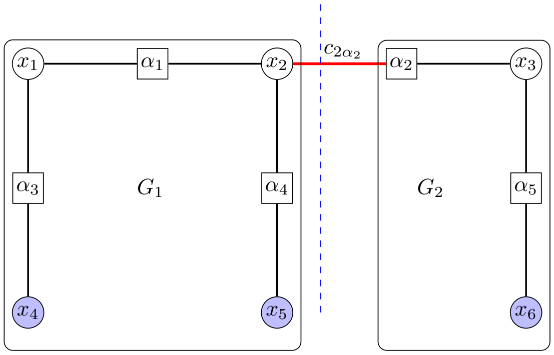


Fig. 5: Subgraphs G_1 and G_2 of G by cutting edge $(2, \alpha_2)$.

Proposition 1. *The following procedures lead to a feasible set of counting numbers $c_\alpha > 0, c_j \geq 0, c_{j\alpha} \geq 0$ that solves* (59)*:*

i) *Choose $c_\alpha > 0, c_j \geq 0$ for $j \in V, \alpha \in F$ such that* (60) *is satisfied;*
ii) *Iterate over each edge in the graph, split the graph along the edge and calculate the corresponding $c_{j\alpha}$ through* (61).

Proof. Obviously, by construction, $c_\alpha > 0, c_j \geq 0, c_{j\alpha} \geq 0$ for all $j \in V, \alpha \in F$. We next show that they satisfy (59). To this end, denote the two subgraphs G_1 and G_2 discussed earlier by cutting edge (j, α) by $G_{1,j\alpha} = (V_{1,j\alpha}, F_{1,j\alpha}, E_{1,j\alpha})$ and $G_{2,j\alpha} = (V_{2,j\alpha}, F_{2,j\alpha}, E_{2,j\alpha})$, respectively. It follows, for any $j \in V$,

$$\begin{aligned} c_j - \sum_{\alpha\in N(j)} c_{j\alpha} &= c_j - \sum_{\alpha\in N(j)} \Big(\sum_{i\in V_{1,j\alpha}} c_i + \sum_{\beta\in F_{1,j\alpha}} c_\beta\Big) \\ &= c_j - \sum_{\alpha\in N(j)} \Big(1 - \sum_{i\in V_{2,j\alpha}} c_i - \sum_{\beta\in F_{2,j\alpha}} c_\beta\Big) \\ &= \sum_{i\in V} c_i + \sum_{\beta\in F} c_\beta - N_j \times 1 = 1 - N_j, \end{aligned}$$

where the second last equality is due to the fact that $V = \{c_j\} \cup (\cup_{\alpha\in N(j)} V_{2,j\alpha})$ and $F = \cup_{\alpha\in N(j)} F_{2,j\alpha}$. This establishes (59a). The proof of (59b) is similar. □

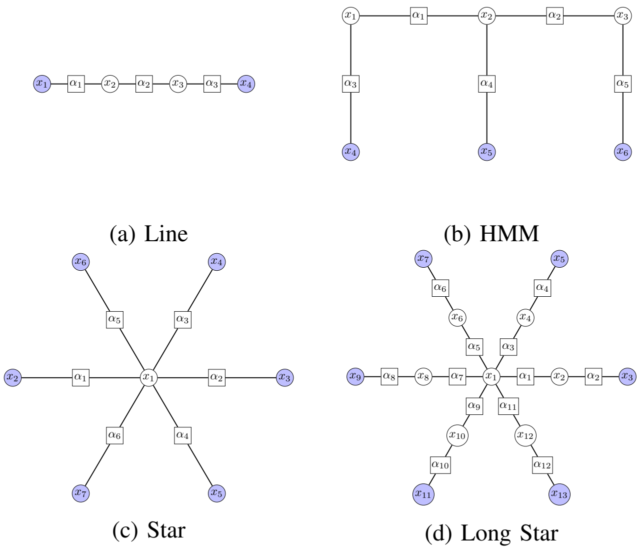
Proposition 1 makes constructing a feasible set of counting numbers that induces convex fractional free energy (see Lemma 8) as easy as finding $c_\alpha > 0, c_j \geq 0, j \in V, \alpha \in F$ that satisfy (60). One choice we found effective is

$$c_j = c_\alpha = \frac{1}{|V| + |F|}$$

for all $j \in V, \alpha \in F$. For the specific example in Figure 5, this choice leads to $c_j = c_\alpha = \frac{1}{11}$. The value of $c_{2\alpha_2}$ is $\frac{7}{11}$ by (61).

V. Numerical examples

In this section, we present three sets of experiments to highlight our framework. The first set of experiments is to validate the correctness of the Constrained Norm-Product and the Iterative Scaling Belief Propagation algorithms, and compare their performance. The second experiment illustrates potential applications of our framework in nonlinear filtering problems for collective dynamics with aggregate observations. Finally in the third experiment, we present an application of our algorithm for the task of color transfer across several images via color palette averaging.



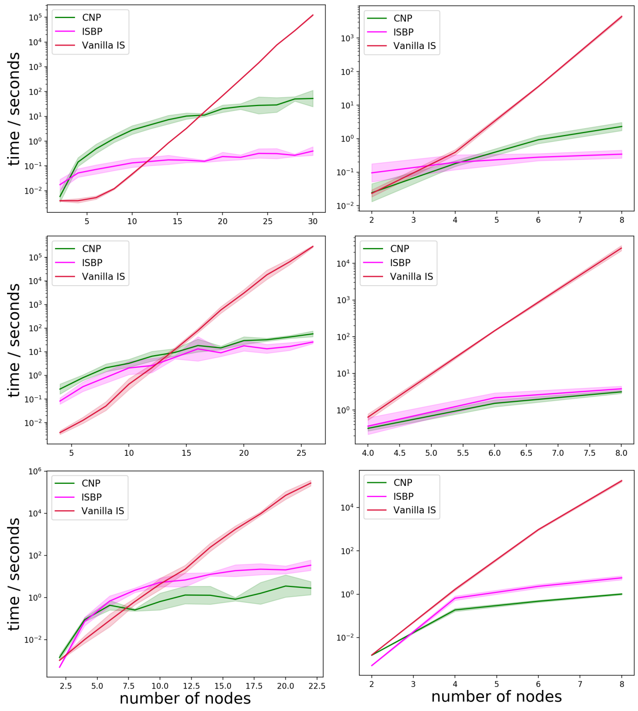
(a) Line
(b) HMM
(c) Star
(d) Long Star

Fig. 6: Testing graphical models

A. Performance evaluation

We implement three algorithms: Constrained Norm-Product (CNP) (Algorithm 4), Iterative Scaling Belief Propagation (ISBP) (Algorithm 2) and Vanilla Iterative Scaling (Vanilla IS) (Algorithm 1) on four different type of graphs: line graphs, hidden Markov models (HMMs), and two star shape graphs (see Figure 6). For the line graph (Figure 6a), the constraints of marginal distribution are on the head and tail nodes. This corresponds to a standard OT problem with two marginals. HMMs (Figure 6b) are widely used in many real-world applications. In the standard HMM framework, the measurements are deterministic values, which can be equivalently viewed as Dirac distributions on the observation/measurement nodes. In our MOT framework, these observation nodes are associated with marginal distribution constraints, which can be viewed as a relaxation of the standard HMM where the deterministic measurements are replaced with "soft" stochastic measurements. The star graph (Figure 6c) structure has marginal constraints on all the leaf nodes. This corresponds to the Barycenter problem over the Wasserstein space [53], which has found applications in information fusion [17].

We test the algorithms with several different configurations. In particular, we vary the number of discrete states at each variable node $d_1 = d_2 = \cdots = d_J = d$ as well as the number of nodes J in the tests. Throughout, we let $\epsilon = 1$. The factor potentials and the counting numbers are set consistently for all the experiments. In particular, the factor potentials are chosen in a way such that the variable nodes connecting to a common factor node are strongly correlated. In our examples, all the factors are connected to only two variable nodes. This choice



(a) $d = 2$ (b) $d = 10$

Fig. 7: Comparison among CNP, ISBP and Vanilla IS. The three rows, from top to bottom, correspond to examples with line graph, HMM, and star graph, respectively. The subplots in the same column have the same value of d.

of factor potentials amounts to taking diagonally dominant matrices as potentials. The counting numbers are selected using the strategy in Proposition 1 by setting $c_j = 0$, $\forall j \in V$ and $c_\alpha = c_\beta$, $\forall \alpha, \beta \in F$. In all our experiments, we observe that the three algorithms converge to the same solutions. To fairly compare the computation complexity of the three algorithms, we use a unified stopping criteria; the algorithms stop when the relative error with respect to the "ground truth" $\mathbf{b}^*$ in terms of the 1-norm is less than 10^{-4}. The "ground truth" $\mathbf{b}^*$ is obtained by running one of the algorithms (e.g., Vanilla IS) for sufficiently many iterations so that the duality gap is less than 10^{-8}. Figure 7 depicts the evaluation results of the three algorithms under different configurations. The y-axis shows the total time consumption of the algorithms before they stop. The x-axis represents the size of the graphs, more specifically, the number of nodes J of the graphs being used. Thus, each subplot showcases the relation between computational complexity and the number of nodes of the graphs. The dependence of the computational complexity on the number d of discrete states at each node can be understood by comparing the subplots along each rows. Each row of subplots corresponds to a type of graph. Thus, the effect of the graph topology on the computational complexity can be captured by comparing the subplots in the same column. From the results it can be seen that, for all types of graphs, and all values of d, the complexity of the Vanilla Iterative Scaling grows exponentially as the number of nodes J increases. In contrast, both CNP and ISBP scale much better than Vanilla IS when J increases. Moreover, CNP and ISBP seem to be less sensitive to the number of discrete states d at each node, compared with Vanilla IS.

To comprehensively compare the performances of ISBP and CNP, we conduct several more experiments on graphs of larger sizes where the Vanilla Iterative Scaling is no longer applicable. Besides the three graphs used in the previous experiment, we study an additional star shape graph with more nodes on each branch (Figure 6d). The stopping criteria is the same as before; the algorithms stop when the relative error with respect to a "ground truth" $\mathbf{b}^*$ in terms of 1-norm is less than 10^{-4}. Since Vanilla IS is computationally forbidden for large J, we run the ISBP algorithm for sufficiently many iterations to obtain $\mathbf{b}^*$. The experiments results are summarized in Figure 8. The presentation of the results in Figure 8 is similar to that in Figure 7, so that we can understand the dependence of the computational complexity over the number of discrete states d, number of nodes J and graph topology. From the figures we can see that the two algorithms CNP and ISBP, have comparable performances. Both of them scale well when J and d increase. ISBP behaves better on line graphs and HMMs, while CNP is faster on star shaped graphs.

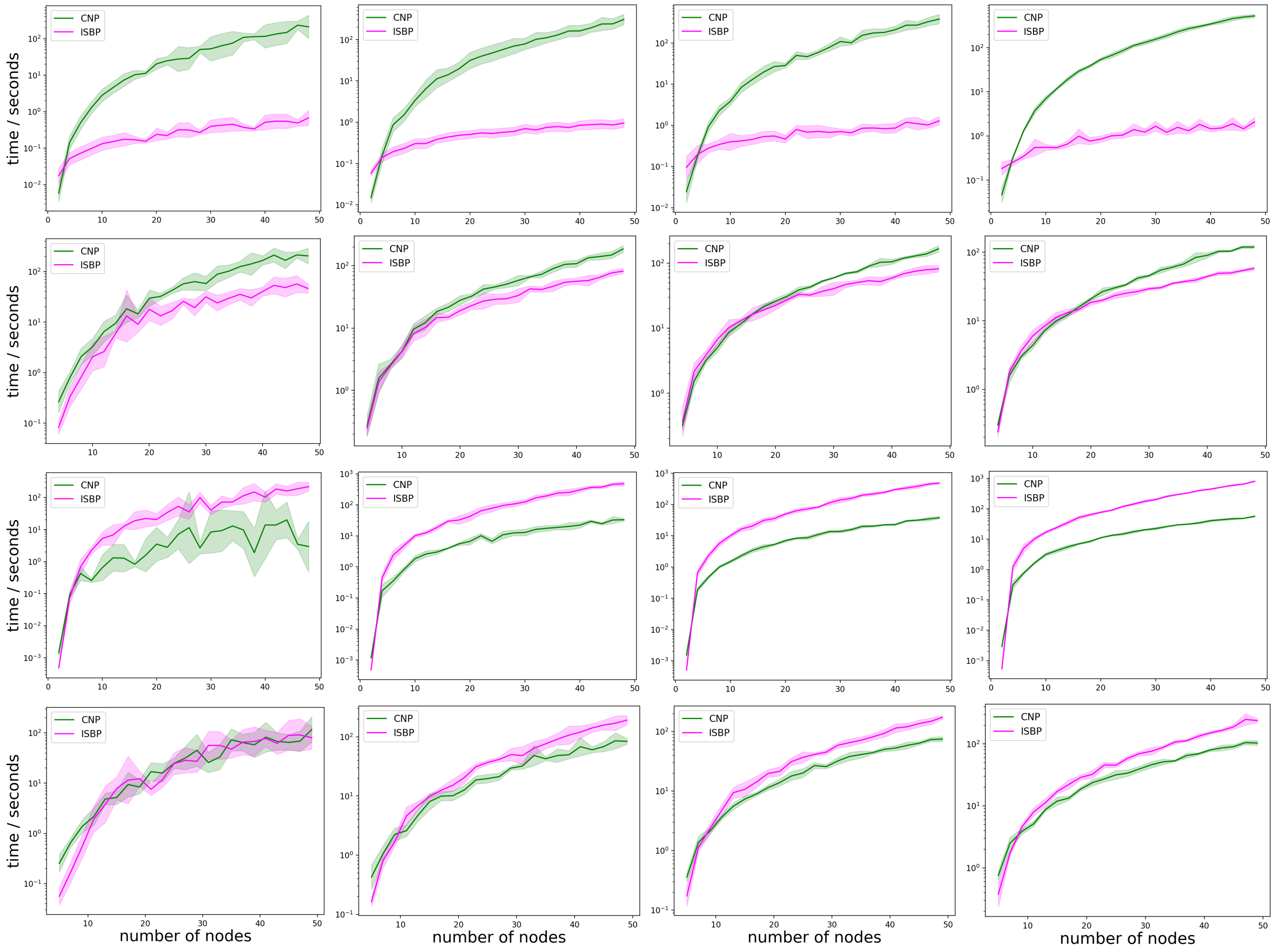
B. Filtering for collective dynamics

In this section, we present an example that is applicable to various applications such as human mobility analysis and bird migration analysis that involve aggregate inference tasks with population level observations. The geographical area is divided into a grid such that each individual can be in one of the grid points. The individuals move along trajectories that are not directly observed, instead, population-level data is recorded using sensors such as cell phone stations or Wi-Fi hotspots that count the number of individuals connected to them at different times. This observation model can be depicted by an HMM (Figure 6(b)) with aggregate counts as observations as illustrated in Figure 9. Our goal is to estimate the movement of the whole population using this limited sensor information.

For illustration purposes, consider a particle ensemble with 10000 agents moving over a 20×20 grid, aiming from bottom-left corner to top-right corner as shown in Figure 10. The dynamics of the agents follow a log-linear distribution governed by four factors: the distance between two locations, the angle between the movements direction and an external force (e.g., wind), the angle between the direction of movement and the direction to the goal, and the preference to stay in the original cell. More specifically, denote the four factors by a single vector

$$F(a,b) = \left(\|b-a\|, \cos^{-1}\left(\frac{\langle b-a, v\rangle}{\|b-a\|\|v\|}\right), \right.$$
$$\left. \cos^{-1}\left(\frac{\langle b-a, x_{\text{goal}}-a\rangle}{\|b-a\|\|x_{\text{goal}}-a\|}\right), -\mathbb{I}[a=b] \right),$$

where a, b represent any two locations, v represents the external force, x_{goal} is the goal position, and $\mathbb{I}[\cdot]$ denotes the indicator function, then the transition probability from a to b is proportional to $\exp(-\langle w, F(a,b)\rangle)$. In our experiments,



(a) $d = 2$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$

Fig. 8: Performance evaluation of CNP and ISBP. The four rows, from top to bottom, are associated with line graph, HMM, star graph and long star graph respectively. The subplots in the same column share the same value of d.

the weights w for the log-linear model associated with these four factors are set to be $(5, 3, 3, 1)$. This model has been used to model the migration of birds [40], [42]. There are 16 sensors placed over the grid as shown in Figure 10a. These sensors can not measure the exact locations of the agents. Instead, the measurement of each sensor is a count of agents it currently observes. An agent can only be captured by one sensor at every time instance. The probability of an observation decreases exponentially as the distance between the sensor and the agent increases.

This filtering problem for collective dynamics can be modeled as a MOT problem, or equivalently a constrained marginal inference problem in our framework. In particular, the agents form a HMM and the sensor measurements correspond to constraints on marginal distributions over the observation nodes. The number of discrete states at each node is $d = 20 \times 20 = 400$ and the number of nodes depends on the number of time steps. We simulate the model for 15 time steps and run both ISBP and CNP to infer the marginal distributions of the free nodes in order to estimate the group behavior of the 10000 agents. The agents start in two clusters: one in the bottom-left corner and one in the bottom-middle area; both aim to reach the top-right corner of the grid in 15 time steps. The results are depicted in Figure 10b. Both CNP and ISBP give the same estimation result and thus we only display one of them in the figures. As can be seen from the plots, even though the sensor data (center column) is hard to interpret visually, our constrained marginal inference framework can still infer the population movements to a satisfying accuracy.

C. Image Pixel Style Transfer via Color Palette Averaging

Finally, we present an experiment on color transfer among images via color palette averaging. An image with N number of pixels is stored as a matrix of size $N \times 3$ with R, G, and B color channels. The color palette of such image is represented by a distribution over the 3-dimensional RGB space. Given multiple color images, we average the color palettes with our proposed algorithm with underlying graph being a star graph; it is a Wasserstein barycenter problem.

A graphical representation of the problem with three images is depicted in Figure 11, wherein each image corresponds to

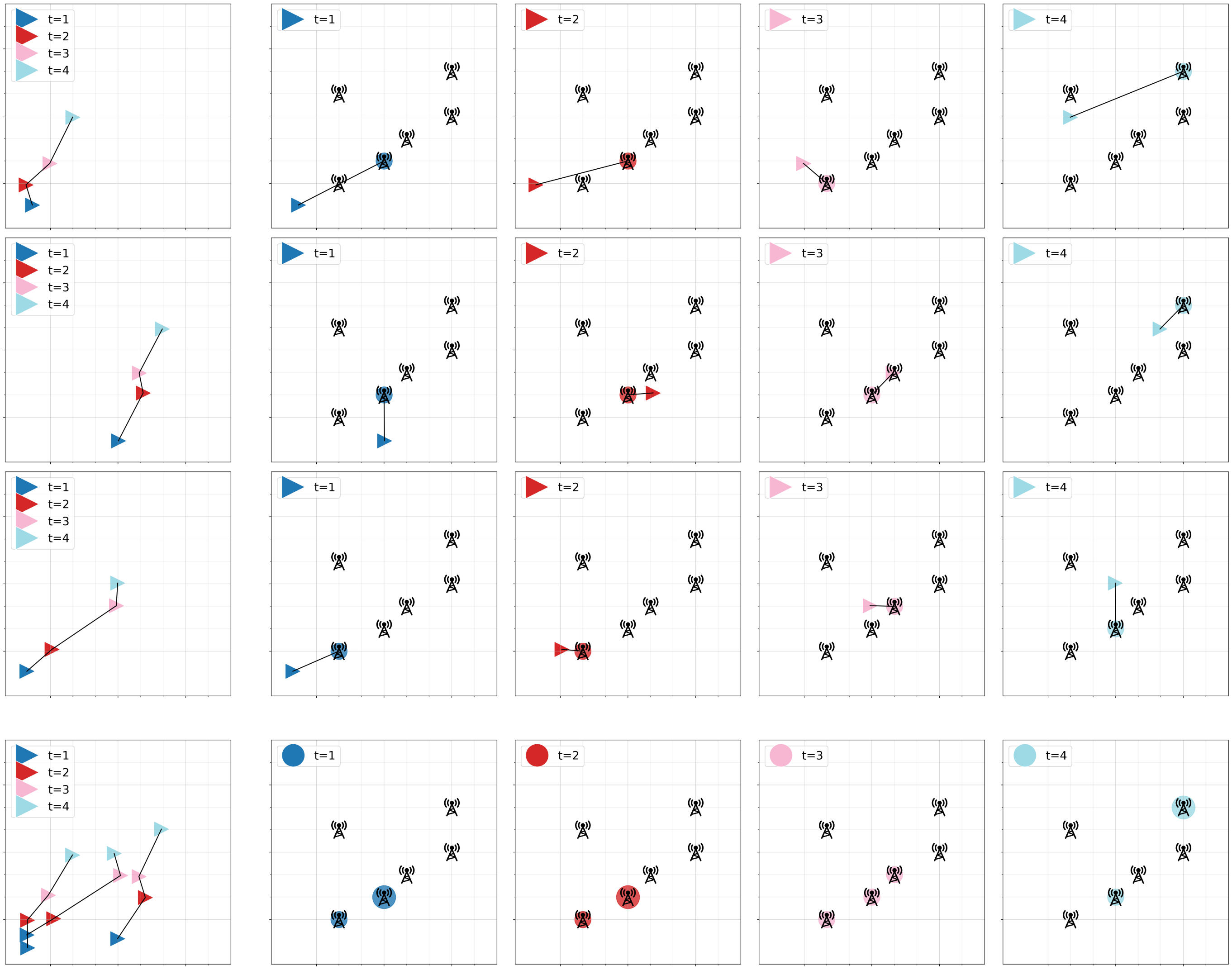


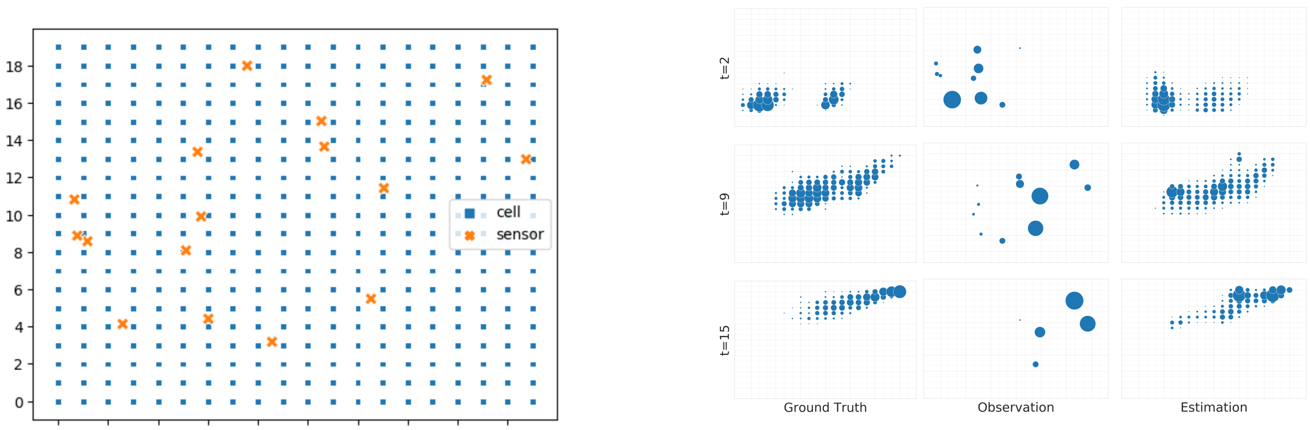
Fig. 9: Illustration of aggregate observations and collective filtering of three individuals. The first three rows depict the true trajectories of the three individuals, their locations, and the corresponding individual (noisy) observations at different time steps. The bottom row shows the aggregate observations in terms of individual counts recorded by each sensor at different time steps. For time steps $t = 1, \ldots, 4$, the observation at each time step is an aggregation of three individuals. The goal of aggregate inference (collective filtering) is to estimate the individual trajectories, which is shown in the first column, from aggregate observations. The locations and observations at different time steps are marked in different colors. The black lines connecting marks and sensors represent the corresponding individual observations.

a leaf-node and is represented by its RGB distribution. We discretize each of the three color channels $[0, 255]$ into 50 different states such that the fixed nodes of the underlying star graph represent fixed color palette distributions $\boldsymbol{\mu}_i \in \mathbb{R}^{50\times50\times50}, i = 1, 2, 3$. The factor potentials of the underlying star graph are constructed based on the (square) Euclidean distance over the 3-dimensional RGB space.

We deploy the ISBP algorithm to estimate the averaged color palette as well as the joint distributions between the center node and the leaf nodes. The joint distribution between the center node and a leaf node forms a mapping between pixel distributions which is then utilized to color transform the corresponding fixed image. We present color transfer results in Figure 12, wherein the top row shows the original images and bottom row depicts the corresponding color transferred images.

VI. Conclusion

We studied multi-marginal optimal transport problems and pointed out an unexpected connection to probabilistic graphical models. This relation between MOT on graphs and constrained PGMs provides a completely new perspective of both MOT and PGMs, which may have far-reaching impact in the future development of both subjects, in both theory and applications. This connection also enables us to adapt the rich class of algorithms in PGMs to tackle difficult MOT problems. In this work, to highlight the key ideas of this line of research, we focused on MOT on trees with discrete states. The next step is to generalize the results to more general graphs with cycles as well as continuous state spaces.



(a) Sensor locations

(b) Simulation results

Fig. 10: Movement estimation of 10000 agents over a 20×20 grid for 15 time steps: (a) displays the grid and the locations of the sensors; (b) shows the estimation results. The three columns, from left to right, represents the simulated movement of agents at three time steps $t = 2, 9, 15$, the sensor measurements, and the estimated agent distributions respectively. The size of the blue dots is proportional to the number of agents at that location.

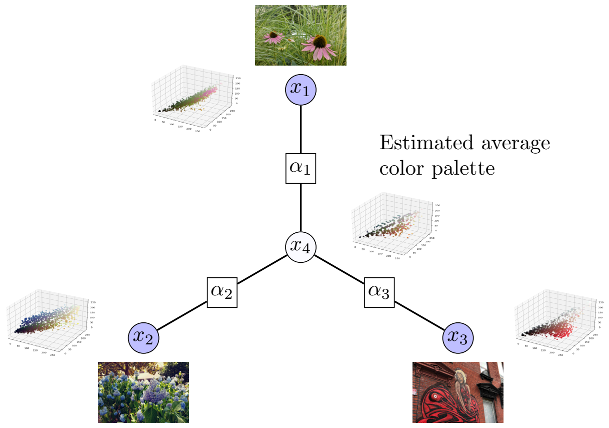


Fig. 11: Color palette averaging of three images. The underlying graph is a star graph with three given images and their corresponding color palettes at the leaf nodes.

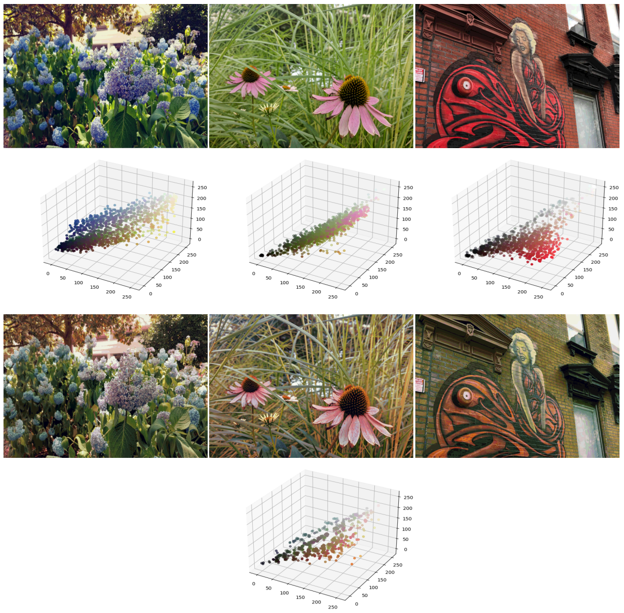


Fig. 12: Image style transformation and color palette averaging. The first two rows show the original images and its corresponding color palettes. The last two rows display the transferred images and the averaged palette (distribution of x_4 in Figure 11).

These are more challenging problems in PGMs, for which exact Bayesian inference is usually too expensive, and one needs to turn to approximate inference such as variational inference or sampling based methods [26]. Other interesting research directions include developing approximation schemes for general cost tensors based on graphical structures, as well as the acceleration of the proposed algorithms.

Appendix A
Derivation of Constrained Norm-product algorithm (Algorithm 4)

We follow the primal-dual ascent algorithm stated in Lemma 9. Denote $\boldsymbol{\lambda}_j = \{\lambda_{j,\alpha}(\mathbf{x}_\alpha)\}$ and $\boldsymbol{\nu}_j = \{\nu_{j,\alpha}(\mathbf{x}_\alpha)\}$. For the sake of convenience, we introduce the following notation

$$\begin{aligned}\hat{\psi}_{j,\alpha}(\mathbf{x}_\alpha) &:= \psi_\alpha(\mathbf{x}_\alpha)\exp(-\nu_{j,\alpha}(\mathbf{x}_\alpha)), \\ \hat{c}_{j\alpha} &:= c_\alpha + c_{j\alpha}, \\ \hat{c}_j &:= c_j + \sum_{\alpha \in N(j)} c_\alpha.\end{aligned} \tag{62}$$

For a fixed $1 \leq j \leq J$, the step (55b) in the primal-dual ascent algorithm requires solving

$$\min_{\mathbf{b}\in dom(f)\cap dom(h_j)} f(\mathbf{b}) + h_j(\mathbf{b}) + \mathbf{b}^T\boldsymbol{\nu}_j.$$

Recall that $f(\mathbf{b}) = \hat{f}(\mathbf{b}) + \delta_{\mathcal{P}}(\mathbf{b})$, $h_j(\mathbf{b}) = \hat{h}_j(\mathbf{b}) + \delta_{\mathcal{M}}(\mathbf{b})$ where

$$\begin{aligned}
\mathcal{M} &= \Big\{\mathbf{b} : \sum_{\mathbf{x}_\alpha\backslash x_j} b_\alpha(\mathbf{x}_\alpha) = b_j(x_j), \forall j\in V, \alpha\in N(j),\\
&\qquad b_j(x_j) = \mu_j(x_j),\ \forall j\in\Gamma\Big\},\\
\mathcal{P} &= \Big\{\mathbf{b} : \sum_{\mathbf{x}_\alpha} b_\alpha(\mathbf{x}_\alpha) = 1, \quad \forall\alpha\in F\Big\},\\
\hat{f}(\mathbf{b}) &= -\sum_{\mathbf{x}_\alpha,\alpha\in F} b_\alpha(\mathbf{x}_\alpha)\ln\psi_\alpha(\mathbf{x}_\alpha) - \sum_{\alpha\in F}\epsilon\, c_\alpha\mathcal{H}(\mathbf{b}_\alpha),\\
\hat{h}_j(\mathbf{b}) &= -\sum_{x_j} b_j(x_j)\ln\phi_j(x_j) - \epsilon\, c_j\mathcal{H}(\mathbf{b}_j)\\
&\quad -\sum_{\alpha\in N(j)}\epsilon\, c_{j\alpha}(\mathcal{H}(\mathbf{b}_\alpha) - \mathcal{H}(\mathbf{b}_j)).
\end{aligned}$$

Thus, for any fixed $1\le j\le J$, step (55b) of the primal-dual ascent algorithm can be reformulated as

$$\begin{aligned}
\min_{\mathbf{b}_j,\mathbf{b}_\alpha,\alpha\in N(j)} \Big\{&-\sum_{x_j} b_j(x_j)\ln\phi_j(x_j) - \sum_\alpha\sum_{\mathbf{x}_\alpha} b_\alpha(\mathbf{x}_\alpha)\ln\hat{\psi}_{j,\alpha}(\mathbf{x}_\alpha)\\
&-\epsilon\hat{c}_j\mathcal{H}(\mathbf{b}_j) - \sum_{\alpha\in N(j)}\epsilon\hat{c}_{j\alpha}(\mathcal{H}(\mathbf{b}_\alpha) - \mathcal{H}(\mathbf{b}_j))\Big\} \qquad (63a)\\
\text{subject to } &\sum_{\mathbf{x}_\alpha} b_\alpha(\mathbf{x}_\alpha) = 1, \sum_{\mathbf{x}_\alpha\backslash x_j} b_\alpha(\mathbf{x}_\alpha) = b_j(x_j), \alpha\in N(j) \qquad (63b)\\
&b_j(x_j) = \mu_j(x_j), \quad \forall x_j,\ \text{if } j\in\Gamma. \qquad (63c)
\end{aligned}$$

Note that when $j\in\Gamma$, the above problem has an extra constraint (63c) compared to the cases where $j\notin\Gamma$.

We next derive a closed form solution $\mathbf{b}_j^*, \mathbf{b}_\alpha^*$ to (63). The constraint (63b) implies that $\mathbf{b}_j$ is a marginal distribution of $\mathbf{b}_\alpha$, thus, $\mathbf{b}_\alpha$ can be rewritten in terms of conditional distribution $\mathbf{b}_{\alpha|j}$ as

$$b_\alpha(\mathbf{x}_\alpha) = b_j(x_j) b_{\alpha|j}(\mathbf{x}_\alpha \mid x_j). \qquad (64)$$

The entropy $\mathcal{H}(\mathbf{b}_\alpha)$ can be rewritten as [70]

$$\mathcal{H}(\mathbf{b}_\alpha) = \mathcal{H}(\mathbf{b}_j) + \sum_{x_j} b_j(x_j)\mathcal{H}(\mathbf{b}_{\alpha|j})$$

where

$$\mathcal{H}(\mathbf{b}_{\alpha|j}) = -\sum_{\mathbf{x}_\alpha\backslash x_j} b_{\alpha|j}(\mathbf{x}_\alpha \mid x_j)\ln b_{\alpha|j}(\mathbf{x}_\alpha \mid x_j).$$

Thus, in terms of new variables $\mathbf{b}_j, \mathbf{b}_{\alpha|j}, \alpha\in N(j)$, the optimization problem (63) reads

$$\begin{aligned}
&\min_{\mathbf{b}_j}\{-\sum_{x_j} b_j(x_j)\ln\phi_j(x_j) - \epsilon\hat{c}_j\mathcal{H}(\mathbf{b}_j) + \sum_{x_j} b_j(x_j) \qquad (65)\\
&\sum_{\alpha\in N(j)}\epsilon\hat{c}_{j\alpha}\underbrace{[\min_{\mathbf{b}_{\alpha|j}} -\sum_{\mathbf{x}_\alpha\backslash x_j} b_{\alpha|j}(\mathbf{x}_\alpha \mid x_j)\ln\hat{\psi}_{j,\alpha}^{1/(\epsilon\hat{c}_{j\alpha})}(\mathbf{x}_\alpha) - \mathcal{H}(\mathbf{b}_{\alpha|j})]}_{\star}\}
\end{aligned}$$

together with the extra constraint $\mathbf{b}_j = \boldsymbol{\mu}_j$ if $j\in\Gamma$. One advantage of this reformulation is that the problem now can be optimized over $\mathbf{b}_{\alpha|j}$ first and then over $\mathbf{b}_j$.

Minimizing (65) over $\mathbf{b}_{\alpha|j}$ is a standard exercise, and the minimizer is

$$b^*_{\alpha|j}(\mathbf{x}_\alpha \mid x_j) = \hat{\psi}_{j,\alpha}(\mathbf{x}_\alpha)^{1/\epsilon\hat{c}_{j\alpha}} / \sum_{\mathbf{x}_\alpha\backslash x_j}\hat{\psi}_{j,\alpha}(\mathbf{x}_\alpha)^{1/\epsilon\hat{c}_{j\alpha}}.$$

Thus, the value for block $(\star)$ is

$$(\star) = -\ln\sum_{\mathbf{x}_\alpha\backslash x_j}\hat{\psi}_{j,\alpha}(\mathbf{x}_\alpha)^{1/\epsilon\hat{c}_{j\alpha}}.$$

Denote

$$m_{\alpha\to j}(x_j) = \left(\sum_{\mathbf{x}_\alpha\backslash x_j}\hat{\psi}_{j,\alpha}(\mathbf{x}_\alpha)^{1/\epsilon\hat{c}_{j\alpha}}\right)^{\epsilon\hat{c}_{j\alpha}}, \qquad (66)$$

then (65) with optimal $\mathbf{b}_{\alpha|j}$ can be simplified as

$$\min_{\mathbf{b}_j}\left[-\mathcal{H}(\mathbf{b}_j) - \sum_{x_j} b_j(x_j)\ln\phi_j^{1/\epsilon\hat{c}_j}(x_j)\prod_{\alpha\in N(j)} m_{\alpha\to j}^{1/\epsilon\hat{c}_j}(x_j)\right]. \qquad (67)$$

When $j\in\Gamma$, $\mathbf{b}_j = \boldsymbol{\mu}_j$ is the only feasible point; $\mathbf{b}_j^* = \boldsymbol{\mu}_j$. When $j\notin\Gamma$, (67) is again a standard exercise with the unique minimizer being

$$b_j^*(x_j) \propto \left(\phi_j(x_j)\prod_{\alpha\in N(j)} m_{\alpha\to j}(x_j)\right)^{1/\epsilon\hat{c}_j}. \qquad (68)$$

Combining $\mathbf{b}_j^*$ and $\mathbf{b}^*_{\alpha|j}$, we obtain

$$b_\alpha^*(\mathbf{x}_\alpha) = b_j^*(x_j) b^*_{\alpha|j}(\mathbf{x}_\alpha|x_j) = \frac{b_j^*(x_j)}{m_{\alpha\to j}^{1/\epsilon\hat{c}_{j\alpha}}(x_j)}\hat{\psi}_{j,\alpha}(\mathbf{x}_\alpha)^{1/\epsilon\hat{c}_{j\alpha}}. \qquad (69)$$

Next, we move to step (55c) of the primal-dual ascent algorithm (Lemma 9), which reads

$$\boldsymbol{\lambda}_j \leftarrow -\boldsymbol{\nu}_j - \nabla\hat{f}(\mathbf{b}^*) + A^T\boldsymbol{\sigma}$$

with $\boldsymbol{\sigma}$ being an arbitrary vector. The vector $A^T\boldsymbol{\sigma}$ spans the orthogonal space to the domain $\mathcal{B}$ (see Lemma 9) of f. In our problem, $\mathcal{B} = \mathcal{P}$ is the probability simplex, thus $A^T\boldsymbol{\sigma}$ is in alignment with $\mathbf{1}$, the vector with all 1 entries. It follows that

$$\lambda_{j,\alpha}(\mathbf{x}_\alpha) = -\nu_{j,\alpha}(\mathbf{x}_\alpha) - \nabla\hat{f}(b_\alpha^*(\mathbf{x}_\alpha)) + \sigma_\alpha\mathbf{1}. \qquad (70)$$

The value of $\nabla\hat{f}(b_\alpha^*(\mathbf{x}_\alpha))$ is

$$\nabla\hat{f}(b_\alpha^*(\mathbf{x}_\alpha)) = -\ln\psi_\alpha(\mathbf{x}_\alpha) + \epsilon c_\alpha\left(\ln b_\alpha^*(\mathbf{x}_\alpha) + 1\right).$$

Define $n_{j\to\alpha}(\mathbf{x}_\alpha)$ as

$$n_{j\to\alpha}(\mathbf{x}_\alpha) := \exp(-\lambda_{j,\alpha}(\mathbf{x}_\alpha)), \qquad (71)$$

then, in view of (69),

$$\begin{aligned} n_{j\to\alpha}(\mathbf{x}_\alpha) &\propto \exp(\nu_{j,\alpha}(\mathbf{x}_\alpha) - \ln\psi_\alpha(\mathbf{x}_\alpha))(b^*_\alpha(\mathbf{x}_\alpha))^{\epsilon c_\alpha} \\ &= \hat{\psi}^{-1}_{j,\alpha}(\mathbf{x}_\alpha)\left(\frac{b^*_j(x_j)}{m^{1/\epsilon\hat{c}_{j\alpha}}_{\alpha\to j}(x_j)}\right)^{\epsilon c_\alpha}\hat{\psi}^{c_\alpha/\hat{c}_{j\alpha}}_{j,\alpha}(\mathbf{x}_\alpha) \\ &= \left(\frac{b^*_j(x_j)}{m^{1/\epsilon\hat{c}_{j\alpha}}_{\alpha\to j}(x_j)}\right)^{\epsilon c_\alpha}\hat{\psi}^{c_\alpha/\hat{c}_{j\alpha}-1}_{j,\alpha}(\mathbf{x}_\alpha) \\ &= \left(\frac{b^*_j(x_j)}{m^{1/\epsilon\hat{c}_{j\alpha}}_{\alpha\to j}(x_j)}\right)^{\epsilon c_\alpha}\hat{\psi}^{-c_{j\alpha}/\hat{c}_{j\alpha}}_{j,\alpha}(\mathbf{x}_\alpha), \end{aligned} \tag{72}$$

where the last equation is due to $\hat{c}_{j\alpha} = c_\alpha + c_{j\alpha}$. By step (55a) of the primal-dual ascent algorithm, $\nu_{j,\alpha} = \sum_{i\in N(\alpha)\backslash j}\lambda_{i,\alpha}(\mathbf{x}_\alpha)$. Combining it with (62) and (71) yields

$$\begin{aligned} \hat{\psi}_{j,\alpha}(\mathbf{x}_\alpha) &= \psi_\alpha(\mathbf{x}_\alpha)\prod_{i\in N(\alpha)\backslash j}\exp(-\lambda_{i,\alpha}(\mathbf{x}_\alpha)) \\ &= \psi_\alpha(\mathbf{x}_\alpha)\prod_{i\in N(\alpha)\backslash j} n_{i\to\alpha}(\mathbf{x}_\alpha). \end{aligned} \tag{73}$$

Finally, plugging (73) into (66) leads to

$$m_{\alpha\to j}(x_j) = \left(\sum_{\mathbf{x}_\alpha\backslash x_j}\left(\psi_\alpha(\mathbf{x}_\alpha)\prod_{i\in N(\alpha)\backslash j} n_{i\to\alpha}(\mathbf{x}_\alpha)\right)^{1/\epsilon\hat{c}_{j\alpha}}\right)^{\epsilon\hat{c}_{j\alpha}}$$

Plugging (73) into (72), in view of the different forms of $\mathbf{b}^*_j$ for $j\in\Gamma$ and $j\notin\Gamma$, we obtain

$$\begin{aligned} n_{j\to\alpha}(\mathbf{x}_\alpha) \propto &\left(\frac{\phi^{1/\hat{c}_j}_j(x_j)\prod_{\beta\in N(j)} m^{1/\hat{c}_j}_{\beta\to j}(x_j)}{m^{1/\hat{c}_{j\alpha}}_{\alpha\to j}(x_j)}\right)^{c_\alpha} \\ &\left(\psi_\alpha(\mathbf{x}_\alpha)\prod_{i\in N(\alpha)\backslash j} n_{i\to\alpha}(\mathbf{x}_\alpha)\right)^{-c_{j\alpha}/\hat{c}_{j\alpha}} \end{aligned}$$

for $j\notin\Gamma$, and

$$n_{j\to\alpha}(\mathbf{x}_\alpha) \propto \left(\frac{\mu_j(x_j)}{m^{1/\epsilon\hat{c}_{j\alpha}}_{\alpha\to j}(\mathbf{x}_\alpha)}\right)^{\epsilon c_\alpha}\left(\psi_\alpha(\mathbf{x}_\alpha)\prod_{i\in N(\alpha)\backslash j} n_{i\to\alpha}(\mathbf{x}_\alpha)\right)^{-c_{j\alpha}/\hat{c}_{j\alpha}}$$

for $j\in\Gamma$. This concludes the derivation.

REFERENCES

[1] L. C. Evans and W. Gangbo, Differential equations methods for the Monge-Kantorovich mass transfer problem. American Mathematical Soc., 1999, vol. 653.

[2] C. Villani, Topics in optimal transportation. American Mathematical Soc., 2003, no. 58.

[3] S. Haker, L. Zhu, A. Tannenbaum, and S. Angenent, "Optimal mass transport for registration and warping," International Journal of Computer Vision, vol. 60, no. 3, pp. 225–240, 2004.

[4] M. Mueller, P. Karasev, I. Kolesov, and A. Tannenbaum, "Optical flow estimation for flame detection in videos," IEEE Transactions on image processing, vol. 22, no. 7, pp. 2786–2797, 2013.

[5] Y. Chen, T. T. Georgiou, and M. Pavon, "On the relation between optimal transport and Schrödinger bridges: A stochastic control viewpoint," Journal of Optimization Theory and Applications, vol. 169, no. 2, pp. 671–691, 2016.

[6] ——, "Optimal transport over a linear dynamical system," IEEE Transactions on Automatic Control, vol. 62, no. 5, pp. 2137–2152, 2017.

[7] A. Galichon, Optimal Transport Methods in Economics. Princeton University Press, 2016.

[8] Y. Chen, "Modeling and control of collective dynamics: From Schrödinger bridges to optimal mass transport," Ph.D. dissertation, University of Minnesota, 2016.

[9] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in International conference on machine learning, 2017, pp. 214–223.

[10] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in Advances in neural information processing systems, 2013, pp. 2292–2300.

[11] W. E. Deming and F. F. Stephan, "On a least squares adjustment of a sampled frequency table when the expected marginal totals are known," The Annals of Mathematical Statistics, vol. 11, no. 4, pp. 427–444, 1940.

[12] W. Gangbo and A. Świech, "Optimal maps for the multidimensional Monge-Kantorovich problem," Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences, vol. 51, no. 1, pp. 23–45, 1998.

[13] G. Carlier, "On a class of multidimensional optimal transportation problems," Journal of convex analysis, vol. 10, no. 2, pp. 517–530, 2003.

[14] B. Pass, "On the local structure of optimal measures in the multi-marginal optimal transportation problem," Calculus of Variations and Partial Differential Equations, vol. 43, no. 3-4, pp. 529–536, 2012.

[15] ——, "Multi-marginal optimal transport: theory and applications," ESAIM: Mathematical Modelling and Numerical Analysis, vol. 49, no. 6, pp. 1771–1790, 2015.

[16] L. Nenna, "Numerical methods for multi-marginal optimal transportation," Ph.D. dissertation, 2016.

[17] F. Elvander, I. Haasler, A. Jakobsson, and J. Karlsson, "Multi-marginal optimal transport using partial information with applications in robust localization and sensor fusion," Signal Processing, p. 107474, 2020.

[18] J.-D. Benamou, G. Carlier, and L. Nenna, "Generalized incompressible flows, multi-marginal transport and Sinkhorn algorithm," Numerische Mathematik, vol. 142, no. 1, pp. 33–54, 2019.

[19] G. Buttazzo, L. De Pascale, and P. Gori-Giorgi, "Optimal-transport formulation of electronic density-functional theory," Physical Review A, vol. 85, no. 6, p. 062502, 2012.

[20] Y. Khoo, L. Lin, M. Lindsey, and L. Ying, "Semidefinite relaxation of multimarginal optimal transport for strictly correlated electrons in second quantization," SIAM Journal on Scientific Computing, vol. 42, no. 6, pp. B1462–B1489, 2020.

[21] Y. Chen, G. Conforti, and T. T. Georgiou, "Measure-valued spline curves: An optimal transport viewpoint," SIAM Journal on Mathematical Analysis, vol. 50, no. 6, pp. 5947–5968, 2018.

[22] Y. Chen and J. Karlsson, "State tracking of linear ensembles via optimal mass transport," IEEE Control Systems Letters, vol. 2, no. 2, pp. 260–265, 2018.

[23] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, "Iterative Bregman projections for regularized transportation problems," SIAM Journal on Scientific Computing, vol. 37, no. 2, pp. A1111–A1138, 2015.

[24] T. Lin, N. Ho, M. Cuturi, and M. I. Jordan, "On the complexity of approximating multimarginal optimal transport," arXiv preprint arXiv:1910.00152, 2020.

[25] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," Foundations and Trends® in Machine Learning, vol. 1, no. 1–2, pp. 1–305, 2008.

[26] D. Koller and N. Friedman, Probabilistic graphical models: principles and techniques. MIT press, 2009.

[27] H. Attias, "A variational Baysian framework for graphical models," in Advances in neural information processing systems, 2000, pp. 209–215.

[28] J. Bilmes and G. Zweig, "The graphical models toolkit: An open source software system for speech and time-series processing," in 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 4. IEEE, 2002, pp. IV–3916.

[29] P. Larranaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armananzas, G. Santafé, A. Pérez et al., "Machine learning in bioinformatics," Briefings in bioinformatics, vol. 7, no. 1, pp. 86–112, 2006.

[30] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in Advances in neural information processing systems, 2001, pp. 689–695.

[31] ——, "Constructing free-energy approximations and generalized belief propagation algorithms," IEEE Transactions on information theory, vol. 51, no. 7, pp. 2282–2312, 2005.

[32] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in Proceedings of the

Fifteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 1999, pp. 467–475.

[33] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," Exploring artificial intelligence in the new millennium, vol. 8, pp. 236–239, 2003.

[34] S. M. Aji and R. J. McEliece, "The generalized distributive law," IEEE transactions on Information Theory, vol. 46, no. 2, pp. 325–343, 2000.

[35] J. Pearl, "Probabilistic reasoning in intelligent systems: Networks of plausible inference," Morgan Kaufmann Publishers Inc, 1988.

[36] T. Hazan and A. Shashua, "Norm-product belief propagation: Primal-dual message-passing for approximate inference," IEEE Transactions on Information Theory, vol. 56, no. 12, pp. 6294–6316, 2010.

[37] Y. W. Teh and M. Welling, "The unified propagation and scaling algorithm," in Advances in neural information processing systems, 2002, pp. 953–960.

[38] D. R. Sheldon and T. G. Dietterich, "Collective graphical models," in Advances in Neural Information Processing Systems, 2011, pp. 1161–1169.

[39] D. Sheldon, T. Sun, A. Kumar, and T. Dietterich, "Approximate inference in collective graphical models," in International Conference on Machine Learning, 2013, pp. 1004–1012.

[40] T. Sun, D. Sheldon, and A. Kumar, "Message passing for collective graphical models," in International Conference on Machine Learning, 2015, pp. 853–861.

[41] I. Haasler, A. Ringh, Y. Chen, and J. Karlsson, "Estimating ensemble flows on a Hidden Markov Chain," IEEE 58th Conference on Decision and Control, 2019.

[42] R. Singh, I. Haasler, Q. Zhang, J. Karlsson, and Y. Chen, "Inference with aggregate data: An optimal transport approach," arXiv preprint arXiv:2003.13933, 2020.

[43] G. Monge, Mémoire sur la théorie des déblais et des remblais. De l'Imprimerie Royale, 1781.

[44] L. V. Kantorovich, "On the transfer of masses," in Dokl. Akad. Nauk. SSSR, vol. 37, no. 7-8, 1942, pp. 227–229.

[45] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," The annals of mathematical statistics, vol. 35, no. 2, pp. 876–879, 1964.

[46] J. Franklin and J. Lorenz, "On the scaling of multidimensional matrices," Linear Algebra and its applications, vol. 114, pp. 717–735, 1989.

[47] A. S. Asratian, T. M. Denley, and R. Häggkvist, Bipartite graphs and their applications. Cambridge university press, 1998, vol. 131.

[48] P. Atkins, The laws of thermodynamics: A very short introduction. OUP Oxford, 2010.

[49] S. Kullback and R. A. Leibler, "On information and sufficiency," The annals of mathematical statistics, vol. 22, no. 1, pp. 79–86, 1951.

[50] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," Machine learning, vol. 37, no. 2, pp. 183–233, 1999.

[51] Y. Weiss, C. Yanover, and T. Meltzer, "MAP estimation, linear programming and belief propagation with convex free energies," in Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, 2007, pp. 416–425.

[52] I. Haasler, A. Ringh, Y. Chen, and J. Karlsson, "Multi-marginal optimal transport with a tree-structured cost and the schrödinger bridge problem," arXiv preprint arXiv:2004.06909, 2020.

[53] M. Agueh and G. Carlier, "Barycenters in the Wasserstein space," SIAM Journal on Mathematical Analysis, vol. 43, no. 2, pp. 904–924, 2011.

[54] S. J. Wright, "Coordinate descent algorithms," Mathematical Programming, vol. 151, no. 1, pp. 3–34, 2015.

[55] H. H. Bauschke and A. S. Lewis, "Dykstras algorithm with Bregman projections: A convergence proof," Optimization, vol. 48, no. 4, pp. 409–427, 2000.

[56] P. Tseng, "Dual ascent methods for problems with strictly convex costs and linear constraints: A unified approach," SIAM Journal on Control and Optimization, vol. 28, no. 1, pp. 214–242, 1990.

[57] Z.-Q. Luo and P. Tseng, "On the convergence rate of dual ascent methods for linearly constrained convex minimization," Mathematics of Operations Research, vol. 18, no. 4, pp. 846–867, 1993.

[58] J. M. Altschuler and E. Boix-Adsera, "Polynomial-time algorithms for multimarginal optimal transport problems with structure," arXiv preprint arXiv:2008.03006, 2020.

[59] J. Altschuler, F. Bach, A. Rudi, and J. Weed, "Approximating the quadratic transportation metric in near-linear time," arXiv preprint arXiv:1810.10046, 2018.

[60] E. Tenetov, G. Wolansky, and R. Kimmel, "Fast entropic regularized optimal transport using semidiscrete cost approximation," SIAM Journal on Scientific Computing, vol. 40, no. 5, pp. A3400–A3422, 2018.

[61] J. Altschuler, F. Bach, A. Rudi, and J. Niles-Weed, "Massively scalable Sinkhorn distances via the Nyström method," in Advances in Neural Information Processing Systems, 2019, pp. 4427–4437.

[62] M. Scetbon and M. Cuturi, "Linear time Sinkhorn divergences using positive features," arXiv preprint arXiv:2006.07057, 2020.

[63] Z.-Q. Luo and P. Tseng, "On the convergence of the coordinate descent method for convex differentiable minimization," Journal of Optimization Theory and Applications, vol. 72, no. 1, pp. 7–35, 1992.

[64] J. Altschuler, J. Niles-Weed, and P. Rigollet, "Near-linear time approximation algorithms for optimal transport via sinkhorn iteration," Advances in neural information processing systems, vol. 30, pp. 1964–1974, 2017.

[65] T. Lin, N. Ho, and M. Jordan, "On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms," in International Conference on Machine Learning, 2019, pp. 3982–3991.

[66] P. Dvurechensky, A. Gasnikov, and A. Kroshnin, "Computational optimal transport: Complexity by accelerated gradient descent is better than by sinkhorn's algorithm," in 35th International Conference on Machine Learning, ICML 2018, 2018, pp. 2196–2220.

[67] T. Lin, N. Ho, X. Chen, M. Cuturi, and M. I. Jordan, "Revisiting fixed support Wasserstein barycenter: Computational hardness and efficient algorithms," arXiv preprint arXiv:2002.04783, 2020.

[68] A. Kroshnin, N. Tupitsa, D. Dvinskikh, P. Dvurechensky, A. Gasnikov, and C. Uribe, "On the complexity of approximating Wasserstein barycenters," in International conference on machine learning. PMLR, 2019, pp. 3530–3540.

[69] O. Meshi, A. Jaimovich, A. Globerson, and N. Friedman, "Convexifying the Bethe free energy," in Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. AUAI Press, 2009, pp. 402–410.

[70] T. M. Cover and J. A. Thomas, Elements of information theory. John Wiley & Sons, 2012.