Computing Kantorovich-Wasserstein Distances on $d$-dimensional histograms using $(d + 1)$-partite graphs

Gennaro Auricchio, Stefano Gualandi, Marco Veneroni
Università degli Studi di Pavia, Dipartimento di Matematica “F. Casorati”
gennaro.auricchio01@universitadipavia.it, stefano.gualandi@unipv.it, marco.veneroni@unipv.it

Federico Bassetti
Politecnico di Milano, Dipartimento di Matematica
federico.bassetti@polimi.it

Abstract

This paper presents a novel method to compute the exact Kantorovich-Wasserstein distance between a pair of $d$-dimensional histograms having $n$ bins each. We prove that this problem is equivalent to an uncapacitated minimum cost flow problem on a $(d + 1)$-partite graph with $(d + 1)n$ nodes and $dn^{d+1}$ arcs, whenever the cost is separable along the principal $d$-dimensional directions. We show numerically the benefits of our approach by computing the Kantorovich-Wasserstein distance of order 2 among two sets of instances: gray scale images and $d$-dimensional biomedical histograms. On these types of instances, our approach is competitive with state-of-the-art optimal transport algorithms.

1 Introduction

The computation of a measure of similarity (or dissimilarity) between pairs of objects is a crucial subproblem in several applications in Computer Vision [24, 25, 22], Computational Statistic [17], Probability [6, 8], and Machine Learning [28, 12, 14, 5]. In mathematical terms, in order to compute the similarity between a pair of objects, we want to compute a distance. If the distance is equal to zero the two objects are considered to be equal; the more the two objects are different, the greater is their distance value. For instance, the Euclidean norm is the most used distance function to compare a pair of points in $\mathbb{R}^d$. Note that the Euclidean distance requires only $O(d)$ operations to be computed.

When computing the distance between complex discrete objects, such as for instance a pair of discrete measures, a pair of images, a pair of $d$-dimensional histograms, or a pair of clouds of points, the Kantorovich-Wasserstein distance [30, 29] has proved to be a relevant distance function [24], which has both nice mathematical properties and useful practical implications. Unfortunately, computing the Kantorovich-Wasserstein distance requires the solution of an optimization problem. Even if the optimization problem is polynomially solvable, the size of practical instances to be solved is very large, and hence the computation of Kantorovich-Wasserstein distances implies an important computational burden.

The optimization problem that yields the Kantorovich-Wasserstein distance can be solved with different methods. Nowadays, the most popular methods are based on (i) the Sinkhorn’s algorithm [11, 27, 3], which solves (approximately) a regularized version of the basic optimal transport problem, and (ii) Linear Programming-based algorithms [13, 15, 20], which exactly solve the basic optimal transport problem by formulating and solving an equivalent uncapacitated minimum cost flow problem. For a nice overview of both computational approaches, we refer the reader to Chapters 2 and 3 in [23], and the references therein contained.

Preprint. Work in progress.
In this paper, we propose a Linear Programming-based method to speed up the computation of Kantorovich-Wasserstein distances of order 2, which exploits the structure of the ground distance to formulate an uncapacitated minimum cost flow problem. The flow problem is then solved with a state-of-the-art implementation of the well-known Network Simplex algorithm \cite{16}.

Our approach is along the line of research initiated in \cite{19}, where the authors proposed a very efficient method to compute Kantorovich-Wasserstein distances of order 1 (i.e., the so-called Earth Mover Distance), whenever the ground distance between a pair of points is the $\ell_1$ norm. In \cite{19}, the structure of the $\ell_1$ ground distance and of regular $d$-dimensional histograms is exploited to define a very small flow network. More recently, this approach has been successfully generalized in \cite{7} to the case of $\ell_\infty$ and $\ell_2$ norms, providing both exact and approximations algorithms, which were able to compute distances between pairs of $512 \times 512$ gray scale images. The idea of speeding up the computation of Kantorovich-Wasserstein distances by defining a minimum cost flow on smaller structured flow networks is also used in \cite{22}, where a truncated distance is used as ground distance in place of a $\ell_p$ norm.

The outline of this paper is as follows. Section 2 reviews the basic notion of discrete optimal transport and fixes the notation. Section 3 contains our main contribution, that is, Theorem 1 and Corollary 2, and fixes the notation. Section 3 contains our main contribution, that is, Theorem 1 and Corollary 2, and fixes the notation. Section 4 presents numerical results of our approaches, compared with the Sinkhorn’s algorithm as implemented in \cite{11} and a standard Linear Programming formulation on a complete bipartite graph \cite{24}. Finally, Section 5 concludes the paper.

2 Discrete Optimal Transport: an Overview

Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$ be two discrete spaces. Given two probability vectors on $X$ and $Y$, say $\mu = (\mu(x_1), \ldots, \mu(x_n))$ and $\nu = (\nu(y_1), \ldots, \nu(y_m))$, and a cost $c : X \times Y \to \mathbb{R}_+$, the Kantorovich-Rubinshtein functional between $\mu$ and $\nu$ is defined as

$$W_c(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \sum_{(x,y) \in X \times Y} c(x, y) \pi(x, y)$$

(1)

where $\Pi(\mu, \nu)$ is the set of all the probability measures on $X \times Y$ with marginals $\mu$ and $\nu$, i.e. the probability measures $\pi$ such that $\sum_{y \in Y} \pi(x, y) = \mu(x)$ and $\sum_{x \in X} \pi(x, y) = \nu(y)$, for every $(x, y)$ in $X \times Y$. Such probability measures are sometimes called transportation plans or couplings for $\mu$ and $\nu$. An important special case is when $X = Y$ and the cost function $c$ is a distance on $X$. In this case $W_c$ is a distance on the simplex of probability vectors on $X$, also known as Kantorovich-Wasserstein distance of order 1.

We remark that the Kantorovich-Wasserstein distance of order $p$ can be defined, more in general, for arbitrary probability measures on a metric space $(X, \delta)$ by

$$W_p(\mu, \nu) := \left( \inf_{\pi \in \Pi(\mu, \nu)} \int_{X \times X} \delta^p(x, y) \pi(dx, dy) \right)^{\min(1/p, 1)}$$

(2)

where now $\Pi(\mu, \nu)$ is the set of all probability measures on the Borel sets of $X \times X$ that have marginals $\mu$ and $\nu$, see, e.g., \cite{4}. The infimum in (2) is attained, and any probability $\pi$ which realizes the minimum is called an optimal transport plan.

The Kantorovich-Rubinshtein transport problem in the discrete setting can be seen as a special case of the following Linear Programming problem, where we assume now that $\mu$ and $\nu$ are generic vectors of dimension $n$, with positive components,

\begin{align}
(P) \quad \min & \quad \sum_{x \in X} \sum_{y \in Y} c(x, y) \pi(x, y) \\
\text{s.t.} & \quad \sum_{y \in Y} \pi(x, y) \leq \mu(x) & \forall x \in X \\
& \quad \sum_{x \in X} \pi(x, y) \geq \nu(y) & \forall y \in Y \\
& \quad \pi(x, y) \geq 0.
\end{align}

(3)
If $\sum_x \mu(x) = \sum_y \nu(y)$ we have the so-called balanced transportation problem, otherwise the transportation problem is said to be unbalanced [18, 10]. For balanced optimal transport problems, constraints (4) and (5) must be satisfied with equality, and the problem reduces to the Kantorovich transport problem (up to normalization of the vectors $\mu$ and $\nu$).

Problem (P) is related to the so-called Earth Mover’s distance. In this case, $X,Y \subset \mathbb{R}^d$ and $x_i$ ($y_j$, respectively) is the center of the data cluster $i$ ($j$, respectively). Moreover, $\mu(x_i)$ ($\nu(y_j)$, respectively) is the number of points in the cluster $i$ ($j$, respectively) and, finally, $c(x_i, y_j)$ is some measure of dissimilarity between $x_i$ and $y_j$. Once the optimal transport $\pi^*$ is determined, the Earth Mover’s distance between the signatures $(x_i, \mu(x_i))]_{i}$ and $(y_j, \nu(y_j))]_{j}$ is defined as

$$EMD(\mu, \nu) = \frac{\sum_{x \in X} \sum_{y \in Y} c(x, y) \pi^*(x, y)}{\sum_{x \in X} \sum_{y \in Y} \pi^*(x, y)}.$$  

Problem (P) can be formulated as an uncapacitated minimum cost flow problem on a bipartite graph defined as follows [2]. The bipartite graph has two partitions of nodes: the first partition has a node for each point $x$ of $X$, and the second partition has a node for each point $y$ of $Y$. Each node $x$ of the first partition has a supply of mass equal to $\mu(x)$, each node of the second partition has a demand of $\nu(y)$ units of mass. The bipartite graph has an (uncapacitated) arc for each element in the Cartesian product $X \times Y$ having cost equal to $c(x, y)$. The minimum cost flow problem defined on this graph yields the optimal transport plan $\pi^*(x, y)$, which indeed is an optimal solution of problem (3)–(6).

For instance, in case of a regular 2D dimensional histogram of size $N \times N$, that is, having $N^2$ bins, we get a bipartite graph with $2N^2$ nodes and $N^4$ arcs (or $2n$ nodes and $n^2$ arcs). Figure 1(a) shows an example for a $3 \times 3$ histogram, and Figure 1(b) gives the corresponding complete bipartite graph.

In this paper, we focus on the case $p = 2$ in equation (2) and the ground distance function $\delta$ is the Euclidean norm $\ell_2$, that is the Kantorovich-Wasserstein distance of order 2, which is denoted by $W_2$.

We provide, in the next section, an equivalent formulation on a smaller $(d + 1)$-partite graph.

### 3 Formulation on $(d + 1)$-partite Graphs

For the sake of clarity, but without loss of generality, we present first our construction considering 2-dimensional histograms and the $\ell_2$ Euclidean ground distance. Then, we discuss how our construction can be generalized to any pair of $d$-dimensional histograms.
Let us consider the following flow problem: let \( \mu = \{ \mu_{i,j} \}_{i,j} \) and \( \nu = \{ \nu_{i,j} \}_{i,j} \) be two probability measures over a \( N \times N \) regular grid denoted by \( G \). In the following paragraphs, we use the notation sketched in Figure 2. In addition, we define the set \( U := \{ 1, \ldots, N \} \).

Since we are considering the \( \ell_2 \) norm as ground distance, we minimize the functional

\[
R : (F_1, F_2) \rightarrow \sum_{i,j=1}^{N} \left[ \sum_{a=1}^{N} (a - i)^2 f_{a,i,j}^{(1)} + \sum_{b=1}^{N} (j - b)^2 f_{i,j,b}^{(2)} \right]
\]

among all \( F_i = \{ f_{a,b,c}^{(i)} \} \), with \( a, b, c \in \{ 1, \ldots, N \} \) real numbers (i.e., flow variables) satisfying the following constraints

\[
\sum_{i=1}^{N} f_{a,i,j}^{(1)} = \mu_{a,j}, \quad \forall a, j \in U \times U \tag{8}
\]

\[
\sum_{j=1}^{N} f_{i,j,b}^{(2)} = \nu_{i,b}, \quad \forall i, b \in U \times U \tag{9}
\]

\[
\sum_{a} f_{a,i,j}^{(1)} = \sum_{b} f_{i,j,b}^{(2)}, \quad \forall i, j \in U \times U, a \in U, b \in U. \tag{10}
\]

Constraints \( 8 \) impose that the mass \( \mu_{a,j} \) at the point \( (a, j) \) is moved to the points \( (k, j) \) for \( k = 1, \ldots, N \) and \( (i, l) \) for \( l = i, \ldots, N \). Constraints \( 9 \) force the point \( (i, b) \) to receive from the points \( (i, l) \) a total mass of \( \nu_{i,b} \). Constraints \( 10 \) require that all the mass that goes from the points \( (a, j) \) to the point \( (i, j) \) is moved to the points \( (i, b) \) for \( b = 1, \ldots, N \). We call a pair \( (F_1, F_2) \) satisfying the constraints \( 8-10 \) a feasible flow between \( \mu \) and \( \nu \). We denote by \( F(\mu, \nu) \) the set of all feasible flows between \( \mu \) and \( \nu \).

Indeed, we can formulate the minimization problem defined by \( 7-10 \) as an uncapacitated minimum cost flow problem on a tripartite graph \( G = (V, A) \). The set of nodes of \( G \) is \( V := G^{(1)} \cup G^{(2)} \cup G^{(3)} \), where \( G^{(1)}, G^{(2)} \) and \( G^{(3)} \) are the nodes corresponding to three \( N \times N \) regular grids. We denote by \( (i, j)^{(l)} \) the node of coordinates \( (i, j) \) in the grid \( G^{(l)} \). We define the two disjoint set of arcs between the successive pairs of node partitions as

\[
A^{(1)} := \{ ((a, j)^{(1)}, (i, j)^{(2)}) | i, a, j \in U \}, \tag{11}
\]

\[
A^{(2)} := \{ ((i, j)^{(2)}, (b, i)^{(3)}) | i, b, j \in U \}, \tag{12}
\]

and, hence, the arcs of \( G \) are \( A := A^{(1)} \cup A^{(2)} \). Note that in this case the graph \( G \) has \( 3N^2 \) nodes and \( 2N^3 \) arcs. Whenever \( (F_1, F_2) \) is a feasible flow between \( \mu \) and \( \nu \), we can think of the values \( f_{a,i,j}^{(1)} \) as the quantity of mass that travels from \( (a, j) \) to \( (i, j) \) or, equivalently, that moves along the arc \( (a, j), (i, j) \) of the tripartite graph, while the values \( f_{i,j,b}^{(2)} \) are the mass moving along the arc \( (i, j), (i, b) \) (e.g., see Figures 1[c] and 2).

Now we can give an idea of the roles of the grids \( G^{(1)}, G^{(2)} \) and \( G^{(3)} \): \( G^{(1)} \) is the grid where is drawn the initial distribution \( \mu \), while on \( G^{(3)} \) it is drawn the final configuration of the mass \( \nu \). The grid \( G^{(2)} \) is an auxiliary grid that hosts a intermediate configuration between \( \mu \) and \( \nu \). From a strict mathematical point of view, this configuration drawn on \( G^{(2)} \) is a measure that lies on a geodetic connecting the measures \( \mu \) and \( \nu \).
We are now ready to state our main contribution.

**Theorem 1.** For each measure \( \pi \) on \( G \times G \) that transports \( \mu \) into \( \nu \), we can find a feasible flow \((F_1, F_2)\) such that

\[
R(F_1, F_2) = \sum_{((a, j), (i, b))} ((a - i)^2 + (b - j)^2)\pi((a, j), (i, b)).
\]

**Proof.** (Sketch). We will only show how to build a feasible flow starting from a transportation plan, the inverse building uses a more technical lemma (the so-called gluing lemma \([4, 30]\)) and can be found in the Additional Material. Let \( \pi \) be a transportation plan, if we write explicitly the ground distance \( \ell_2((a, j), (i, b)) \) we find that

\[
\sum_{((a, j), (i, b))} \ell_2((a, j), (i, b))\pi((a, j), (i, b)) = \sum_{((a, j), (i, b))} ((a - i)^2 + (j - b)^2)\pi((a, j), (i, b))
\]

so that

\[
\sum_{j,i} \left[ \sum_{a,b} (a - i)^2\pi((a, j), (i, b)) + \sum_{a,b} (j - b)^2\pi((a, j), (i, b)) \right].
\]

If we set \( f_{a,i,j}^{(1)} = \sum_b \pi((a, j), (i, b)) \) and \( f_{i,j,b}^{(2)} = \sum_a \pi((a, j), (i, b)) \) we find

\[
\sum_{((a, j), (i, b))} \ell_2((a, j), (i, b))\pi((a, j), (i, b)) = \sum_{i,j} \left[ \sum_{a} (a - i)^2 f_{a,i,j}^{(1)} + \sum_{b} (j - b)^2 f_{i,j,b}^{(2)} \right].
\]

In order to conclude we have to prove that those \( f_{a,i,j}^{(1)} \) and \( f_{i,j,b}^{(2)} \) satisfy the constraints \((8)–(10)\).

By definition we have

\[
\sum_{i} f_{a,i,j}^{(1)} = \sum_{i} \sum_{b} \pi((a, j), (i, b)) = \mu(a, j),
\]

thus proving \((8)\); similarly, it is possible to check constraint \((9)\). The constraint \((10)\) also follows easily since

\[
\sum_{a} f_{a,i,j}^{(1)} = \sum_{a} \sum_{b} \pi((a, j), (i, b)) = \sum_{b} f_{i,j,b}^{(2)}.
\]

\( \square \)

As a straightforward, yet fundamental, consequence we have the following result.

**Corollary 1.** If we set \( c((a, j), (i, b)) = (a - i)^2 + (j - b)^2 \) then, for each measures \( \mu \) and \( \nu \), we have that

\[
W^2_2(\mu, \nu) = \min_{\mathcal{J}(\mu, \nu)} R(F_1, F_2).
\]

Indeed, we can compute the Kantorovich-Wasserstein distance of order 2 between a pair of discrete measures \( \mu, \nu \), by solving an uncapacitated minimum cost flow problem on the given tripartite graph \( G := (G^{(1)} \cup G^{(2)} \cup G^{(3)}), A^{(1)} \cup A^{(2)}). \)

We remark that our approach is very general and it can be directly extended to deal with the following generalizations:

**More general cost functions.** The structure that we have exploited of the Euclidean distance \( \ell_2 \) is present in any cost function \( c : G \times G \rightarrow [0, \infty] \) that is separable, i.e., has the form

\[
c(x, y) = c^{(1)}(x_1, y_1) + c^{(2)}(x_2, y_2),
\]

where both \( c^{(1)} \) and \( c^{(2)} \) are positive real valued functions defined over \( G \). We remark that the whole class of costs \( c_p(x, y) = (x_1 - y_1)^p + (x_2 - y_2)^p \) is of that kind, so we can compute any of the Kantorovich-Wasserstein distances related to each \( c_p \).
Higher dimensional grids. Our approach can handle discrete measures in spaces of any dimension $d$, that is, for instance, any $d$-dimensional histogram. In dimension $d = 2$, we get a tripartite graph because we decomposed the transport along the two main directions. If we have a problem in dimension $d$, we need a $(d + 1)$-plet of grids connected by arcs oriented as the $d$ fundamental directions, yielding a $(d + 1)$-partite graph. As the dimension $d$ grows, our approach gets faster and more memory efficient than the standard formulation given on a bipartite graph.

In the Additional Material, we present a generalization of Theorem 1 to any dimension $d$ and to separable cost functions $c(x, y)$.

4 Computational Results

In this section, we report the results obtained on two different set of instances. The goal of our experiments is to show how our approach scales with the size of the histogram $N$ and with the dimension of the histogram $d$. As cost distance $c(x, y)$, with $x, y \in \mathbb{R}^d$, we use the squared $\ell_2$ norm. As problem instances, we use the gray scale images (i.e., 2-dimensional histograms) proposed by the DOTMark benchmark [26], and a set of $d$-dimensional histograms obtained by biomedical data measured by flow cytometer [9].

Implementation details. We run our experiments using the Network Simplex as implemented in the Lemon C++ graph library¹, since it provides the fastest implementation of the Network Simplex algorithm to solve uncapacitated minimum cost flow problems [16]. The tests are executed on a gaming laptop with Windows 10 (64 bit), equipped with an Intel i7-6700HQ CPU and 16 GB of Ram. The code was compiled with MS Visual Studio 2017, using the ANSI standard C++17. The code execution is single threaded. The Matlab implementation of the Sinkhorn’s algorithm [11] runs in parallel on the CPU cores, but we do not use any GPU in our test. Our C++ code is freely available at https://github.com/stegua/dotlib

Results for the DOTmark benchmark. The DOTmark benchmark contains 10 classes of gray scale images related to randomly generated images, classical images, and real data from microscopy images of mitochondria [26]. In each class there are 10 different images. Every image is given in the data set at the following pixel resolutions: $32 \times 32$, $64 \times 64$, $128 \times 128$, $256 \times 256$, and $512 \times 512$. The images in Figure 3 are respectively the ClassicImages, Microscopy, and Shapes images (one class for each row), shown at highest resolution, that are the three classes for which we report our results. For the lack of space, we do not report here the results for the other seven classes included in the DOTmark benchmark, but the results are essentially the same.

In our test, we first compared four approaches to compute the Kantorovich-Wasserstein distances on images of size $32 \times 32$:

1. EMD: The implementation of Transportation Simplex provided by [24], known in the literature as EMD code, that is an exact general method to solve optimal transport problem.

¹http://lemon.cs.elte.hu(last visited on May, 18th, 2018)
2. **Sinkhorn**: The Matlab implementation of the Sinkhorn’s algorithm\[11\], that is an approximate approach whose performance in terms of speed and numerical accuracy depends on a parameter $\lambda$: for smaller values of $\lambda$, the algorithm is faster but the solution value has a large gap with respect to the optimal value of the transportation problem; for larger values of $\lambda$, the algorithm is more accurate (i.e., smaller gap), but it becomes slower. Unfortunately, for very large value of $\lambda$ the method becomes numerically unstable. The best value of $\lambda$ is problem dependent, and for our test we used $\lambda = 1$ and $\lambda = 1.5$. The second one, is the largest value we found for which the algorithm computes the distances for all the instances considered without facing numerical issues.

3. **Bipartite**: The bipartite formulation presented in Figure 1–(b), which is the same as \[24\], but it is solved with the Network Simplex implemented in the Lemon Graph library \[16\].

4. **3-partite**: The 3-partite formulation proposed in this paper, which for 2-dimensional histograms is represented in 1–(c). Again, we use the Network Simplex of the Lemon Graph Library to solve the corresponding uncapacitated minimum cost flow problem.

Table 1 reports the averages of our results over the three classes of images shown in Table 3. Each class contains 10 instances, and we compute the distance between every possible pair of images within the same class: this corresponds to have 45 instances for each class. We report the means and the standard deviations (between brackets) of the runtime, measured in seconds. Table 1 shows in the second column the runtime for EMD \[24\]. The third and fourth columns gives the runtime and the optimality gap for the Sinkhorn’s algorithm with $\lambda = 1$; the 6-th and 7-th columns for $\lambda = 1.5$. The percentage gap is computed as $\text{Gap} = \frac{UB-\text{opt}}{\text{opt}} \cdot 100$, where $UB$ is the upper bound computed by Sinkhorn’s algorithm, and $\text{opt}$ is the optimal value computed by EMD. The last two columns report the runtime for the bipartite and 3-partite approaches.

As Table 1 shows, the 3-partite approach is clearly faster than any of the three alternatives considered here, despite being an exact method. In addition, we remark that, even on the bipartite formulation, the Network Simplex implementation of the Lemon Graph library is order of magnitude faster than EMD, and hence it should be the best choice in this particular type of instances.

Table 2 reports the results for the bipartite and 3-partite approaches for increasing size of the 2-dimensional histograms. The table report for each of the two approaches, the number of vertices $|V|$, and of arcs $|A|$, and the means and standard deviations of the runtime. As before, each row gives the averages over 45 instances. Table 2 shows that the 3-partite approach is clearly better in terms of memory, the 3-partite graph has a fraction of the number of arcs (which has to be stored in memory), and runtime, since it is at least an order of magnitude faster. Indeed, the 3-partite formulation is better essentially because it exploits the structure of the ground distance $c(x, y)$ used, that is, the squared $\ell_2$ norm.

**Flow Cytometry biomedical data.** Flow cytometry is a laser-based biophysical technology used to study human health disorders. Flow cytometry experiments produce huge set of data, which are very hard to analyze with standard statistics methods and algorithms \[9\]. Currently, such data is used to study the correlations of only two factors (e.g., biomarkers) at the time, by visualizing 2-dimensional histograms and by measuring the (dis-)similarity between pairs of histograms \[21\].

\[http://marcocuturi.net/SI.html\] (last visited on May, 18th, 2018)
Table 2: Comparison of the bipartite and the 3-partite approaches on 2-dimensional histograms.

| Size   | Image Class | Bipartite | 3-partite |
|--------|-------------|-----------|-----------|
|        |             | | | |
| 64 × 64| Classic     | 8 193     | 16 777 216 | 16.3 (3.6) | 12 288 | 524 288 | 2.2 (0.2) |
|        | Microscopy  | 11.7 (1.4) | 1.0 (0.2)  |
|        | Shape       | 13.0 (3.9) | 1.1 (0.3)  |
| 128 × 128| Classic   | 32 768    | 268 435 456 | 1368 (545) | 49 152 | 4 194 304 | 36.2 (5.4) |
|        | Microscopy  | 959 (181) | 23.0 (4.8) |
|        | Shape       | 983 (230) | 17.8 (5.2) |

Table 3: Comparison between the bipartite and the (d + 1)-partite approaches on Flow Cytometry data.

| N d n  | | V | A | Runtime | V | A | Runtime |
|--------|--------|--------|--------|--------|--------|--------|--------|
| 16 2 256 | 512 | 65 536 | 0.024 (0.01) | 768 | 8 192 | 0.003 (0.00) |
| 3 4 096 | 16777 216 | 38.2 (14.0) | 16 384 | 196 608 | 0.12 (0.02) |
| 4 65 536 | out-of-memory | 327 680 | 4 194 304 | 4.8 (0.84) |
| 32 2 1024 | 2 048 | 1 048 756 | 0.71 (0.14) | 3072 | 65 536 | 0.04 (0.01) |
| 3 32 768 | out-of-memory | 131 072 | 3 145 728 | 5.23 (0.69) |

However, during a flow cytometry experiment up to hundreds of factors (biomarkers) are measured and stored in digital format. Hence, we can use such data to build d-dimensional histograms that consider up to d biomarkers at the time, and then comparing the similarity among different individuals by measuring the distance between the corresponding histograms. In this work, we have used the flow cytometry data related to Acute Myeloid Leukemia (AML), available at [http://flowrepository.org/id/FR-FCM-ZZYA](http://flowrepository.org/id/FR-FCM-ZZYA), which contains cytometry data for 359 patients, classified as “normal” or affected by AML. This dataset has been used by the bioinformatics community to run clustering algorithms, which should predict whether a new patient is affected by AML [1].

Table 3 reports the results of computing the distance between pairs of d-dimensional histograms, with d ranging in the set {2, 3, 4}, obtained using the AML biomedical data. For simplicity, we considered regular histograms of size \( n = N^d \) (i.e., \( n \) is the total number of bins), using \( N = 16 \) and \( N = 32 \). Table 3 compares the results obtained by the bipartite and (d + 1)-partite approach, in terms of graph size and runtime. Again, the (d + 1)-partite approach, by exploiting the structure of the ground distance, outperforms the standard formulation of the optimal transport problem. We remark that for \( N = 32 \) and \( d = 3 \), we pass for going out-of-memory with the bipartite formulation, to compute the distance in around 5 seconds with the 4-partite formulation.

5 Conclusions and Future Works

In this paper, we have presented a new network flow formulation on (d + 1)-partite graphs that can speed up the optimal solution of transportation problems whenever the ground cost function \( c(x, y) \) (see objective function (3)) has a separable structure along the main \( d \) directions, such as, for instance, the squared \( \ell_2 \) norm used in the computation of the Kantorovich-Wasserstein distance of order 2.

Our computational results on two different datasets show how our approach scales with the size of the histograms \( N \) and with the dimension of the histograms \( d \). Indeed, by exploiting the cost structure, the proposed approach is better in term of memory consumption, since it has only \( dn^{d+1} \) arcs instead of \( n^2 \). In addition, it is much faster since it has to solve an uncapacitated minimum cost flow problem on a much smaller flow network.
References

[1] Nima Aghaeepour, Greg Finak, Holger Hoos, Tim R Mosmann, Ryan Brinkman, Raphael Gottardo, Richard H Scheuermann, FlowCAP Consortium, DREAM Consortium, et al. Critical assessment of automated flow cytometry data analysis techniques. *Nature methods*, 10(3):228, 2013.

[2] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. *Network flows: Theory, Algorithms, and Applications*. Cambridge, Mass.: Alfred P. Sloan School of Management, Massachusetts Institute of Technology, 1988.

[3] Jason Altschuler, Jonathan Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. In *Advances in Neural Information Processing Systems*, pages 1961–1971, 2017.

[4] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

[6] Federico Bassetti, Antonella Bodini, and Eugenio Regazzini. On minimum kantorovich distance estimators. *Statistics & probability letters*, 76(12):1298–1302, 2006.

[7] Federico Bassetti, Stefano Gualandi, and Marco Veneroni. On the computation of kantorovich-wasserstein distances between 2d-histograms by uncapacitated minimum cost flows. *arXiv preprint arXiv:1804.00445*, 2018.

[8] Federico Bassetti and Eugenio Regazzini. Asymptotic properties and robustness of minimum dissimilarity estimators of location-scale parameters. *Theory of Probability & Its Applications*, 50(2):171–186, 2006.

[9] Tytus Bernas, Elikplimi K Asem, J Paul Robinson, and Bartek Rajwa. Quadratic form: a robust metric for quantitative comparison of flow cytometric histograms. *Cytometry Part A*, 73(8):715–726, 2008.

[10] Haibin Ling and Kazunori Okada. An efficient earth mover’s distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):840–853, 2007.

[11] James B Orlin. A faster strongly polynomial minimum cost flow algorithm. *Operations research*, 41(2):338–350, 1993.

[12] Darya Y Orlova, Noah Zimmerman, Stephen Meehan, Connor Meehan, Jeffrey Waters, Eliver EB Ghosn, Alexander Filatenkov, Gleb A Kolyagin, Yael Gernez, Shanel Tsuda, et al. Earth mover’s distance (emd): a true metric for comparing biomarker expression levels in cell populations. *PLoS one*, 11(3):1–14, 2016.

[13] Ofir Pele and Michael Werman. Fast and robust earth mover’s distances. In *Computer vision, 2009 IEEE 12th international conference on*, pages 460–467. IEEE, 2009.
Then there exists a discrete probability measure

We can then define

\[ \sum_{(a_1, \ldots, a_d)} \pi_1(a_1, \ldots, a_d; b_1, \ldots, b_d) = \sum_{(b_1, \ldots, b_d)} \pi_2(b_1, \ldots, b_d; c_1, \ldots, c_d) \]

Then there exists a discrete probability measure \( \pi = (\pi(a_1, \ldots, a_d; b_1, \ldots, b_d; c_1, \ldots, c_d)) \) in \( \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \) such that

\[ \sum_{(c_1, \ldots, c_d)} \pi(a_1, \ldots, a_d; b_1, \ldots, b_d; c_1, \ldots, c_d) = \pi_1(a_1, \ldots, a_d; b_1, \ldots, b_d) \]

and

\[ \sum_{(a_1, \ldots, a_d)} \pi(a_1, \ldots, a_d; b_1, \ldots, b_d; c_1, \ldots, c_d) = \pi_2(b_1, \ldots, b_d; c_1, \ldots, c_d). \]

Let us take \( \mu = (\mu(a_1, \ldots, a_d)) \), \( \nu = (\nu(b_1, \ldots, b_d)) \) two probability measures and a ground distance of the form

\[ c((a_1, \ldots, a_d), (b_1, \ldots, b_d)) = \sum_{i=1}^d \Delta_i(a_i, b_i). \]  

We can then define

\[ R(F_1, \ldots, F_d) = \sum_{i}^d \left[ \sum_{b_{i-1}, a_{i-1}, a_i} \Delta_i(a_i, b_i) f_{b_1, \ldots, b_{i-1}, a_i, \ldots, a_d, b_i} \right], \]

where

\( F_i = \{ f_{b_1, \ldots, b_{i-1}, a_i, \ldots, a_d, b_i} \} \)

are a \( N^{d+1} \)-plet of real values satisfying the two congruence conditions

\[ \sum_{b_1} f_{a_1, \ldots, a_d, b_1} = \mu(a_1, \ldots, a_d), \]

\[ \sum_{a_N} f_{b_1, \ldots, b_{d-1}, a_d, b_d} = \nu(b_1, \ldots, b_d) \]

and the following \( d - 1 \) connection conditions

\[ \sum_{a_i} f_{b_1, \ldots, b_{i-1}, a_i, \ldots, a_d, b_i} = \sum_{b_{i+1}} f_{b_1, \ldots, b_i, a_{i+1}, \ldots, a_d, b_{i+1}} \]
for \( i = 1, \ldots, d - 1 \). We will call the \( d \)-plet of \((F_1, \ldots, F_d)\) a flow chart between \( \mu \) and \( \nu \).

The set of all possible flow charts between two measures \( \mu \) and \( \nu \) will be indicate with \( \mathcal{F}(\mu, \nu) \). We will then define

\[
\mathcal{R}(\mu, \nu) = \min_{F \in \mathcal{F}(\mu, \nu)} R(F_1, \ldots, F_d).
\]

**Theorem 3.** Let \( \mu = (\mu_{a_1}, \ldots, \mu_{a_d}) \) and \( \nu = (\nu_{b_1}, \ldots, \nu_{b_d}) \) be two probability measures over the grid \( G = \{1, \ldots, N\}^d \), \( c : G \times G \to [0, \infty] \) a separate ground distance, i.e. of the form \( |c| \). Then, for each \( \pi \) transportation plan between \( \mu \) and \( \nu \) there exists a flow chart \((F_1, \ldots, F_d)\) such that

\[
R(F_1, \ldots, F_d) = \sum_{G \times G} c(a, b) \pi(a, b).
\] (21)

In particular

\[
\mathcal{R}(\mu, \nu) = \mathcal{W}_c(\mu, \nu).
\] (22)

**Proof.** Let us consider \( \pi \) a transportation plan, then we can write

\[
\sum_{G \times G} c(a, b) \pi(a, b) = \sum_{a_1, \ldots, a_d, b_1, \ldots, b_d} \sum_{i=1}^d \Delta_i(a_i, b_i) \pi(a_1, \ldots, a_d, b_1, \ldots, b_d) = \sum_{i=1}^d \sum_{a_1, \ldots, a_d, b_1, \ldots, b_d} \Delta_i(a_i, b_i) \pi(a_1, \ldots, a_d, b_1, \ldots, b_d) = \sum_{i=1}^d \left[ \sum_{b_1, \ldots, b_{i-1}} \sum_{a_1, \ldots, a_d, b_i} \Delta_i(a_i, b_i) f_{b_1, \ldots, b_{i-1}, a_i, a_1, \ldots, a_d, b_i} \right],
\] (23)

where

\[
f_{b_1, \ldots, b_{i-1}, a_i, \ldots, a_d, b_i} = \sum_{a_1, \ldots, a_{i-1}, b_{i+1}} \pi(a_1, \ldots, a_d, b_1, \ldots, b_d).
\] (24)

To conclude, we have to prove that those \( f_{b_1, \ldots, b_{i-1}, a_i, \ldots, a_d, b_i} \) satisfy the conditions (17), (18) and (19).

All of those follow from the definition itself, in fact

\[
\sum_{b_{i-1}} f_{a_1, \ldots, a_d, b_{i-1}} = \sum_{b_1, \ldots, b_d} \pi(a_1, \ldots, a_d, b_1, \ldots, b_d) = \mu(a_1, \ldots, a_d),
\] (25)

\[
\sum_{a_d} f_{b_1, \ldots, b_{d-1}, a_d, b_d} = \sum_{a_1, \ldots, a_d} \pi(a_1, \ldots, a_d, b_1, \ldots, b_d) = \nu(b_1, \ldots, b_d)
\] (26)

and

\[
\sum_{a_i} f_{b_1, \ldots, b_{i-1}, a_i, \ldots, a_d, b_i} = \sum_{a_1, \ldots, a_{i-1}, a_i, b_{i+1}, \ldots, b_d} \pi(a_1, \ldots, a_d, b_1, \ldots, b_d) = \sum_{b_{i+1}} f_{b_1, \ldots, b_{i+1}, \ldots, b_N} = \sum_{b_{i+1}} f_{b_1, \ldots, b_{i+1}, \ldots, b_N}.
\] (27)

Let now \((F_1, \ldots, F_d)\) be a flow chart. We have that, for each \( i = 1, \ldots, d \), the \( F_i \) define a probability measure over \( \{1, \ldots, N\}^{d+1} \). For \( i = 1 \) we easily find that

\[
\sum_{a_1, \ldots, a_d, b_1} f_{a_1, \ldots, a_d, b_1} = \sum_{a_1, \ldots, a_d} \sum_{b_1} f_{a_1, \ldots, a_d, b_1} = \sum_{a_1, \ldots, a_d} \mu_{a_1, \ldots, a_d} = 1.
\]

If we assume that \( F_i \) is a probability measure, then, using the condition (19), we get that

\[
\sum_{b_1, \ldots, a_i, a_{i+1}, \ldots, b_{i+1}} f_{b_1, \ldots, b_{i+1}, a_i, a_{i+1}, \ldots, b_i} = \sum_{b_1, \ldots, a_i, a_{i+1}, \ldots, b_i} \sum_{a_{i+1}} f_{b_1, \ldots, b_{i+1}, a_{i+1}, a_i, \ldots, b_i} = \sum_{a_i} \sum_{b_1, \ldots, a_{i+1}, \ldots, b_i} f_{b_1, \ldots, b_i, a_{i+1}, a_i, \ldots, b_i} = 1.
\] (28)
Thus, by induction, we get that all the $F_i$ are actually probability measures.

Since we showed that $f^{(1)}_{a_1,\ldots,a_d,b_1}$ and $f^{(2)}_{b_1,a_2,\ldots,a_d,b_2}$ are both probability measures and holds the relation (19) we can apply the gluing lemma and find a probability measure $\pi^{(1)}_{a_1,\ldots,a_d,b_1,b_2}$ such that

$$\sum_{b_2} \pi^{(1)}_{a_1,\ldots,a_d,b_1,b_2} = f^{(1)}_{a_1,\ldots,a_d,b_1}$$

and

$$\sum_{b_2} \pi^{(1)}_{a_1,\ldots,a_d,b_1,b_2} = f^{(2)}_{b_1,a_2,\ldots,a_d,b_2}.$$  

Let us now consider $f^{(3)}_{b_1,b_2,a_3,\ldots,a_d,b_3}$ and $\pi^{(1)}_{a_1,\ldots,a_d,b_1,b_2}$, we have

$$\sum_{a_2} \sum_{a_1} \pi^{(1)}_{a_1,\ldots,a_d,b_1,b_2} = \sum_{a_2} f^{(2)}_{b_1,a_2,\ldots,a_d,b_2} = \sum_{b_3} f^{(3)}_{b_1,b_2,a_3,\ldots,a_d,b_3}.$$  

so we can apply once again the gluing lemma and find a probability measure $\pi^{(2)}_{a_1,\ldots,a_d,b_1,b_2,b_3}$ such that

$$\sum_{b_3} \pi^{(2)}_{a_1,\ldots,a_d,b_1,b_2,b_3} = \pi^{(1)}_{a_1,\ldots,a_d,b_1,b_2}$$

and

$$\sum_{a_1,a_2} \pi^{(2)}_{a_1,\ldots,a_d,b_1,b_2,b_3} = f^{(3)}_{b_1,b_2,a_3,\ldots,a_d,b_3}.$$  

We can iterate this process for $d - 1$ times and find a probability measure $\pi_{a_1,\ldots,a_d,b_1,\ldots,b_d}$ such that

$$\sum_{b_1,\ldots,b_d} \pi_{a_1,\ldots,a_d,b_1,\ldots,b_d} = \sum_{b_1,\ldots,b_{d-1}} \sum_{b_d} \pi_{a_1,\ldots,a_d,b_1,\ldots,b_d} = \sum_{b_1,\ldots,b_{N-1}} \pi_{a_1,\ldots,a_N,b_1,\ldots,b_{N-1}} = \ldots = \sum_{b_1} \sum_{b_2} \sum_{a_1,\ldots,a_d} \pi_{a_1,\ldots,a_d,b_1,b_2} = \sum_{b_1} f^{(1)}_{a_1,\ldots,a_d,b_1} = \mu_{a_1,\ldots,a_N}.$$  

Similarly, we have

$$\sum_{a_1,\ldots,a_d} \pi_{a_1,\ldots,a_d,b_1,\ldots,b_d} = \nu_{b_1,\ldots,b_d},$$

thus proving that $\pi$ transportes $\mu$ into $\nu$.

For such a $\pi$, we now prove that

$$R(F_1,\ldots,F_d) = \sum_{a_1,\ldots,a_d,b_1,\ldots,b_d} \sum_{i=1}^d \Delta_i(a_i,b_i)\pi_{a_1,\ldots,a_d,b_1,\ldots,b_d}. \tag{36}$$

Start with

$$\sum_{a_1,\ldots,a_d} \sum_{i=1}^d \Delta_i(a_i,b_i)\pi_{a_1,\ldots,a_d,b_1,\ldots,b_d} = \sum_{i=1}^d \sum_{a_1,\ldots,a_d,b_1,\ldots,b_d} \Delta_i(a_i,b_i)\pi_{a_1,\ldots,a_d,b_1,\ldots,b_d}. \tag{37}$$

Let us consider the term

$$\sum_{a_1,\ldots,a_d} \sum_{i=1}^d \Delta_i(a_i,b_i)\pi_{a_1,\ldots,a_d,b_1,\ldots,b_d} = \sum_{b_1,\ldots,b_{d-1}} \sum_{a_1,\ldots,a_d} \Delta_i(a_i,b_i) \sum_{a_1,\ldots,a_{d-1},b_1,\ldots,b_d} \pi_{a_1,\ldots,a_d,b_1,\ldots,b_d} \tag{38}$$

but, thanks to the Gluing Lemma, we have that

$$\sum_{a_1,\ldots,a_{d-1},b_1,\ldots,b_d} \pi_{a_1,\ldots,a_d,b_1,\ldots,b_d} = \sum_{a_1,\ldots,a_{d-1},b_1,\ldots,b_{d-1}} \pi_{a_1,\ldots,a_d,b_1,\ldots,b_{d-1}} = \ldots = \sum_{a_1,\ldots,a_1} \pi_{a_1,\ldots,a_N,b_1,\ldots,b_1} = f^{(i)}_{b_1,b_2,\ldots,a_N,b_1}. \tag{39}$$

So the proof is complete.