

Mass Transportation Problems with Connectivity Constraints, with Applications to Energy Landscape Comparison

Frédéric Cazals, Dorian Mazauric

▶ **To cite this version:**

Frédéric Cazals, Dorian Mazauric. Mass Transportation Problems with Connectivity Constraints, with Applications to Energy Landscape Comparison. [Research Report] RR-8611, Inria Sophia Antipolis; INRIA. 2014. hal-01090705

HAL Id: hal-01090705
https://hal.science/hal-01090705

Submitted on 4 Dec 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Mass Transportation Problems with Connectivity Constraints, with Applications to Energy Landscape Comparison

Frédéric Cazals and Dorian Mazauric

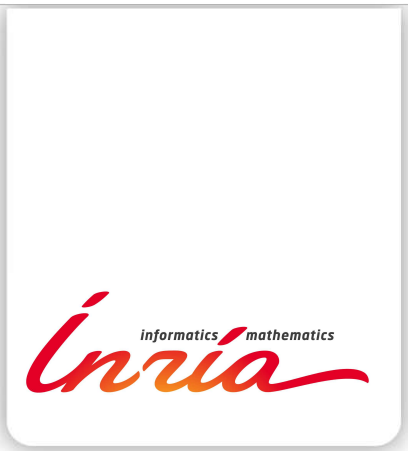
RESEARCH REPORT

N° 8611

October 2014

Project-Team Algorithms-Biology-Structure

ISSN 0249-6399 ISRN INRIA/RR--8611--FR+ENG



Mass Transportation Problems with Connectivity Constraints, with Applications to Energy Landscape Comparison

Frédéric Cazals and Dorian Mazauric

Project-Team Algorithms-Biology-Structure

Research Report n° 8611 — October 2014 — 24 pages

Abstract: Given two graphs, the *supply* and the *demand* graphs, we analyze the mass transportation problem between their vertices, under connectivity constraints. More precisely, for every subset of supply nodes inducing a connected component of the supply graph, we require that the set of demand nodes receiving non-zero flow from this subset induces a connected component of the demand graph. As opposed to the classical problem, a.k.a the *earth mover distance* (EMD), which is amenable to linear programming (LP), this new problem is very difficult to solve, and we make four contributions.
First, we formally introduce two optimal transportation problems, namely *minimum-cost flow under connectivity constraints problem* (EMD-CC) and *maximum-flow under cost and connectivity constraints problem* (EMD-CCC). Second, we prove that the decision version of EMD-CC is NP-complete even for very simple classes of instances. We deduce that the decision version of EMD-CCC is NP-complete, and also prove that EMD-CC is not in APX even for simple classes of instances. Third, we develop a greedy heuristic algorithm returning admissible solutions, of time complexity $O(n^3m^2)$ with n and m the numbers of vertices of the supply and demand graphs, respectively. Finally, on the experimental side, we compare the transport plans computed by our greedy method against those produced by the aforementioned LP. Using synthetic landscapes (Voronoi landscapes), we show that our greedy algorithm is effective for graphs involving up to 1000 nodes. We also show the relevance of our algorithms to compare energy landscapes of biophysical systems (protein models).

Key-words: Multi-commodity flow, optimal transportation, connectivity constraints, bipartite graphs, NP-hardness, not in APX, Polynomial Time Approximation Scheme (PTAS), bio-physics, energy landscapes, protein models, meta-stable states.

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Transport de masse avec contraintes de connectivité, et applications à la comparaison de paysages énergétiques

Résumé : Dans ce travail, nous analysons le problème de *multi-commodity flow* avec contraintes de connectivité. Ce problème peut être défini comme un problème classique de flot, avec contraintes de connectivité. Plus précisement, pour tout ensemble de noeuds du graphe source induisant un sous-graphe connexe, il est requis que les noeuds du graphe de demande recevant du flot induisent également un sous-graphe connexe. Cette contrainte rend le problème difficile, alors même qu'en l'absence de contrainte il se réduit à un programme linéaire. Par ailleurs, un contrainte sur le nombre d'arête est aussi posée.

Nous montrons que ce problème est est NP-complet et n'est pas dans APX, même pour des instances simples (e.g. quand le graphe induit par les noeuds de demande est complet). Nous développons également un algorithme de complexité polynomiale lorsque la contrainte sur le nombre d'arêtes est relaxée, ainsi qu'un algorithme glouton calculant une solution admissible pour le cas général.

Du point de vue applicatif, ce problème est motivé par la comparaison de paysages énergétiques en biophysique. Des tests sur un modèle simplifié de protéine et sur des donnees synthétiques montrent que malgré la complexité du problème, nos algorithmes sont effectifs jusqu'à quelques centaines de sommets.

Mots-clés : Multi-commodity flow, transport optimal, contraintes de connectivité graphes bipartie, NP-hardness, not in APX, Polynomial Time Approximation Scheme (PTAS), biophysique, paysages énergétiques, modèles de protéines, méta-stabilité.

Contents

1 Introduction **4**
1.1 Transportation Problems 4
1.2 Applications to Energy Landscapes in Physics 4
1.3 Contributions and Paper Overview 5

2 Problem Formulation and Models **5**
2.1 Minimum-cost Flow Problem 6
2.2 Minimum-cost Flow under Connectivity Constraints Problem 6
2.3 Maximum-flow under Cost and Connectivity Constraints Problem 8

3 Complexity Results **9**
3.1 The Problems are Difficult even for Simple Instances 9
3.2 PTAS when the Number of Active Edges is not Bounded 12

4 Algorithms **13**
4.1 Greedy Algorithm `Alg-EMD-CCC-G` 13
4.2 Iterative Algorithm `Alg-EMD-CCC-G-I` 15

5 Experiments **16**
5.1 Implementations 16
5.2 Potential Energy Landscapes of Simplified Protein Models 16
5.2.1 Specifications 16
5.2.2 Results 18
5.3 Voronoi Landscapes 19
5.3.1 Specifications 19
5.3.2 Results 20

6 Conclusion and Future Works **20**

7 Supplemental: Experiments **23**
7.1 BLN Models 23
7.2 Statistics 24

1 Introduction

1.1 Transportation Problems

Optimal transportation problems have a long standing history in mathematics and computer science, originating with the works of Monge on earth moving (≪ *la théorie des déblais et des remblais* ≫) [8]. Such problems were later rephrased in terms of Riemannian geometry and measure theory [13], one key concept being the distance between two distributions, namely is the minimal amount of work that must be performed to transform one distribution into the other by moving distribution mass around. Various applications were developed across all sciences, one of the early ones in computer science beingthe *earth mover distance* (EMD), used to compare two images using their color histograms [10].

From an algorithmic standpoint, the problem of computing the EMD is a special case of the transportation problem [4] that admits a polynomial algorithm using linear programming. We will formally define this problem later in terms of minimum-cost flow in graphs. In this paper, we investigate a similar problem adding connectivity constraints, motivated by a problem from bio-physics.

1.2 Applications to Energy Landscapes in Physics

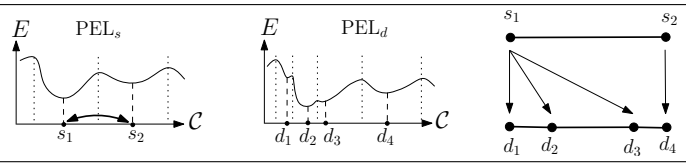
The terminology EMD is clearly evocative of the energy landscapes used to study molecular systems in biophysics [14]. To see why, consider a molecular system (a protein, a cluster of water molecules, a cluster of ions, etc) consisting of n atoms, so that its conformational space is $3n$ dimensional – each atom has three Cartesian coordinates. The potential energy landscape (PEL) of the system is defined as the function defined over this conformational space, which associates an energy to each conformation. Features of the PEL play a crucial role to understand the system's behavior: the local minima correspond to meta-stable states; the volume of the basin associated with a local minimum is a measure of the entropy associated with the meta-stable state; a *saddle* connecting two local minima defines a possible transition between the two corresponding meta-stable states. However, studying the PEL of systems involving from hundreds to tens of thousands of atoms is a major endeavor. Because an analytical expression of the PEL is in general unknown, PEL can only be explored via simulation methods, yielding questions in three directions [11, 14]: sampling, sketching, comparing.

Sampling consists of generating samples on the PEL based on molecular dynamics on Monte Carlo strategies [14]. This is a technical endeavor, since one needs to choose the type of model used (atomic or coarse grain model), the potential energy function for that model, and the sampling method. The multiplicity of these choices is clearly a hindrance to understand the very properties of the system, so that the problem of comparing sampled PEL obtained under different conditions arises.

Sketching consists of identifying samples which are near local minima and saddles. Upon sketching, one is left with a graph specified as follows: its nodes are the local minima, each node being endowed with the number of configurations associated with the meta-stable state; two nodes are linked by an edge provided by the corresponding local minima are linked by a saddle on the PEL.

Finally, comparing aims at checking whether two sets of samples uncovered the same regions of the landscape. As of now, this problem has only been addressed by comparing the local minima of the two sample sets. One could naturally resort to the aforementioned EMD (Fig. 1). However, we also wish to respect a connectivity constraint, which is absent from the original earth mover distance problem: prosaically, if a transportation plan aims at filling lakes of a mountain range from lakes of another mountain ranges, connected lakes of the former should be

Figure 1 Comparing energy landscapes. (Left) Each landscape is partitioned into the basins associated to its local minima. **(Right)** Comparing two landscapes is phrased as a mass transportation problem on the bipartite graph defined by the two sets of minima.



mapped to connected lakes of the latter. These new constraints make our problem, also called minimum-cost flow under connectivity constraints problem, much more difficult to solve. Indeed, this problem cannot be written as a linear program.

1.3 Contributions and Paper Overview

The simplest transportation problems reduce to linear programs. However, when the supply and demand nodes belong to graphs, these problems are oblivious to the connectivity of these graphs. In this context, we make the following contributions.

In Section 2, we define two optimal transportation problems for graphs so as to minimize a transport cost between their nodes, while respecting connectivity constraints.These problems are *minimum-cost flow under connectivity constraints problem* (`EMD-CC`) and *maximum-flow under cost and connectivity constraints problem* (`EMD-CCC`). In Section 3, we first prove that the decision version of `EMD-CC` is in NP, and obtain a simpler definition of the connectivity constraints. We then show that the decision version of `EMD-CC` is NP-complete even for very simple classes of instances. We deduce that the decision version of `EMD-CCC` is NP-complete, and also prove that the `EMD-CC` problem is not in APX even for simple classes of instances. We also prove a Polynomial Time Approximation Scheme for `EMD-CC` and `EMD-CCC` when transport plans involving a *large* number of edges are allowed. Since the flow problems are very hard to solve and even to approximate, a greedy heuristic algorithm is developed in Section 4. Finally, Section 5 presents an experimental assessment of our greedy algorithm on two types of landscapes (associated with protein models and synthetic), showing that it is effective for graphs of intermediate size.

2 Problem Formulation and Models

Consider two connected graphs: a *supply* graph $G = (V, E)$ and a *demand* graph $G' = (V', E')$. The set $V = \{v_1, \ldots, v_{|\mathcal{I}|}\}$ represents the supply nodes. We denote by $\mathcal{I}$ the set of indices of the supply nodes. The value $X_i \geq 0$ represents the volume of supply of node v_i for all $i \in \mathcal{I}$. The set $V' = \{v'_1, \ldots, v'_{|\mathcal{J}|}\}$ represents the demand nodes. We denote by $\mathcal{J}$ the set of indices of the demand nodes. The value $Y_j \geq 0$ represents the volume of demand of node v'_j for all $j \in \mathcal{J}$. Without loss of generality, we assume that $X_i > 0$ and $Y_j > 0$ for all $i \in \mathcal{I}, j \in \mathcal{J}$. Let $B = (V \cup V', V \times V')$ be the complete bipartite graph between the supply and the demand nodes. The real values $c_{i,j} \geq 0$ represent the linear cost of sending a unit of flow from node v_i to node v'_j for all $i \in \mathcal{I}, j \in \mathcal{J}$. The variable $F_{i,j}$ represents the volume of flow sent by node v_i to v'_j for all $i \in \mathcal{I}$, $j \in \mathcal{J}$. For all $i \in \mathcal{I}$, $j \in \mathcal{J}$, the cost of sending avolume of flow $F_{i,j}$ through edge $\{v_i, v'_j\} \in E(B)$ is $F_{i,j}c_{i,j}$. Given the flows $F_{i,j}$ for all edges of B,the total flow is $\sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} F_{i,j}$ and the total cost is $\sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} F_{i,j}c_{i,j}$.

2.1 Minimum-cost Flow Problem

The classical *minimum-cost flow problem* (EMD), or *transportation problem* [4], consists in determining a minimum cost flow satisfying the demands and respecting the supply constraints. This problem is polynomial since it reduces to solving the following linear program:

$$LP \begin{cases} \text{Min} & \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F_{i,j}\ c_{i,j} & \\ & \sum_{i\in\mathcal{I}} F_{i,j} \leq Y_j & \forall j \in \mathcal{J}, \\ & \sum_{j\in\mathcal{J}} F_{i,j} \leq X_i & \forall i \in \mathcal{I}, \\ & \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F_{i,j} = \min(\sum_{i\in\mathcal{I}} X_i, \sum_{j\in\mathcal{J}} Y_j) & \\ & F_{i,j} \geq 0 & \forall i \in \mathcal{I}, \forall j \in \mathcal{J}. \end{cases} \quad (1)$$

The first line is the objective function, the second line represents the demand constraints (the total volume of flow that arrives at node v'_j must be at most Y_j for all $j \in \mathcal{J}$), the third line describes the supply constraints (the total volume of flow sent by node v_i must be at most X_i for all $i \in \mathcal{I}$), the fourth line states that the total amount of flow equals the minimum between the total volume of supplies and the total volume of demands, and the last line guarantees that flows are positive. Note that if $\sum_{i\in\mathcal{I}} X_i \geq \sum_{j\in\mathcal{J}} Y_j$, then the fourth line can be removed and the inequality constraints of the second line become equality constraints.

Based on this LP, we introduce the *total number of edges*, the *total flow*, the *total cost*, and their ratio, known as the *earth mover distance* [10]:

$$M^{\text{EMD}} = \sum_{i,j|F_{i,j}>0} 1, F^{\text{EMD}} = \sum_{i,j} F_{i,j}, C^{\text{EMD}} = \sum_{i,j} F_{i,j}c_{i,j}, \text{ and } d_{\text{EMD}} = \frac{C^{\text{EMD}}}{F^{\text{EMD}}}. \quad (2)$$

2.2 Minimum-cost Flow under Connectivity Constraints Problem

Problem statement. In this paper, we investigate a more difficult problem consisting in adding *connectivity constraints*. The *minimum-cost flow under connectivity constraints problem* (EMD-CC) consists in computing a minimum cost flow (satisfying demand and supply constraints) such that for every subset of supply nodes $H \subseteq V$ that induces a connected subgraph, the graph induced by the set of nodes $H' = \{v'_j \mid F_{i,j} > 0, v_i \in H, i \in \mathcal{I}, j \in \mathcal{J}\}$ is connected. In other words, all the nodes of G' that receive non-zero flow from at least one node of H, induce a connected subgraph of G'. Let $\mathcal{H}$ be the set of all sets of nodes of G that induce a connected subgraph. Let $\mathcal{H}'$ be the set of all sets of nodes of G' that induce a connected subgraph.

Furthermore, we have a constraint on the number of edges of B that can support non-zero flow. From our application point of view, we aim at computing flow solutions with a linear (in the total number of nodes) number of edges that support non-zero flow, that is a number $\theta(|V|+|V'|)$ of such edges. But in practice, we do not know a priori this number of edges, and so we must addan upper-bound M for this number of edges. Furthermore, without this constraint, optimal solutions would not be interesting for our problem because of the quadratic number of edges supporting non-zero flow. Indeed, a class of near optimal solutions can be easily obtained as follows. We first add a volume of flow ε for all edges of the complete bipartite graph B, and then obtain an auxiliary instance (in which we update the supply and the demand volumes for all nodes). By construction, the connectivity constraints are satisfied, and so the EMD-CC is equivalent to EMD for this auxiliary instance, which gets solved by the LP of Eq. (1). Thus, we get a polynomial time approximation scheme for this problem choosing ε function of the desired approximation ratio (Theorem 3), and so optimal solutions may be not interesting. In summary, since we do not know the number of edges, then our implemented algorithms do not take this upper-bound in input, or equivalently $M = |E(B)|$, but we carefully analyze the number of edges

that support non-zero flow in our experimental results. Formally, given M, $0 \leq M \leq |E(B)|$, `EMD-CC` can be written as follows:

$$\begin{cases} \text{Min} & \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F_{i,j}\ c_{i,j} & \\ & \sum_{i\in\mathcal{I}} F_{i,j} \leq Y_j & \forall j \in \mathcal{J}, \\ & \sum_{j\in\mathcal{J}} F_{i,j} \leq X_i & \forall i \in \mathcal{I}, \\ & \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F_{i,j} = \min(\sum_{i\in\mathcal{I}} X_i, \sum_{j\in\mathcal{J}} Y_j) & \\ & F_{i,j} \geq 0 & \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \\ & \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}|F_{i,j}>0} 1 \leq M & \\ & H' = \{v'_j \mid F_{i,j} > 0, v_i \in H, i \in \mathcal{I}, j \in \mathcal{J}\} \in \mathcal{H}' & \forall H \in \mathcal{H}. \end{cases} \tag{3}$$

From which we define the *total number of edges*, the *total flow*, the *total cost*, and their ratio:

$$M^{\text{EMD-CC}} = \sum_{i,j|F_{i,j}>0} 1, F^{\text{EMD-CC}} = \sum_{i,j} F_{i,j}, C^{\text{EMD-CC}} = \sum_{i,j} F_{i,j}c_{i,j}, d_{\text{EMD-CC}} = \frac{C^{\text{EMD-CC}}}{F^{\text{EMD-CC}}}. \tag{4}$$

Note that the number of connectivity constraints may be exponential in the number of nodes of G. Observe also that if G' is a complete graph and $M = |E(B)|$, then `EMD-CC` is polynomial because the constraints of the two last lines are always satisfied.

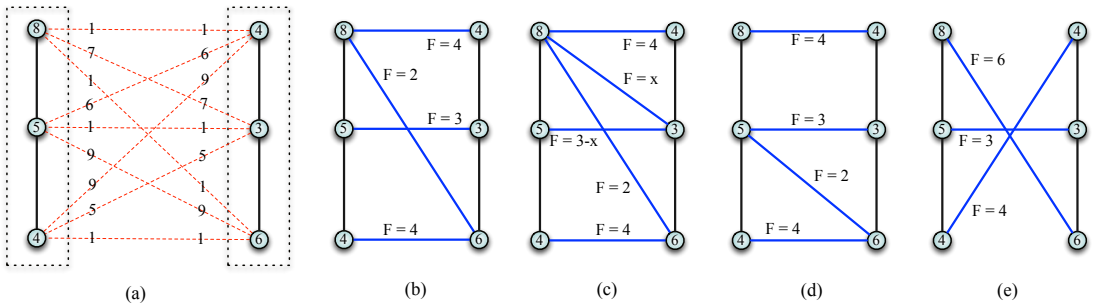
Example. Fig. 2 (a) describes a simple instance: three supply nodes with $X_1 = 8$, $X_2 = 5$, $X_3 = 4$ and three demand nodes with $Y_1 = 4$, $Y_2 = 3$, $Y_3 = 6$. Integers on nodes represent these supply and demand values. The graph $G = (V, E)$ is a path, where $V = \{v_1, v_2, v_3\}$ and $E = \{\{v_1, v_2\}, \{v_2, v_3\}\}$. The graph $G' = (V', E')$ is also a path, where $V' = \{v'_1, v'_2, v'_3\}$ and $E' = \{\{v'_1, v'_2\}, \{v'_2, v'_3\}\}$. Integers on edges of the complete bipartite graph B represents unitary costs $c_{i,j}$ for all $i, j \in \{1, 2, 3\}$. Theunit costs are $c_{1,1} = 1$, $c_{1,2} = 7$, $c_{1,3} = 1$, $c_{2,1} = 6$, $c_{2,2} = 1$, $c_{2,3} = 9$, $c_{3,1} = 9$, $c_{3,2} = 5$, and $c_{3,3} = 1$. Since G is a path, we get $\mathcal{H} = \{\{v_1\}, \{v_1, v_2\}, \{v_1, v_2, v_3\}, \{v_2\}, \{v_2, v_3\}, \{v_3\}\}$. Since G' is also a path, we get $\mathcal{H}' = \{\{v'_1\}, \{v'_1, v'_2\}, \{v'_1, v'_2, v'_3\}, \{v'_2\}, \{v'_2, v'_3\}, \{v'_3\}\}$.
`EMD` consists in solving:

$$\begin{cases} \text{Min} & F_{1,1} + 7F_{1,2} + F_{1,3} + 6F_{2,1} + F_{2,2} + 9F_{2,3} + 9F_{3,1} + 5F_{3,2} + F_{3,3}, \\ & F_{1,1} + F_{2,1} + F_{3,1} = 4, \\ & F_{1,2} + F_{2,2} + F_{3,2} = 3, \\ & F_{1,3} + F_{2,3} + F_{3,3} = 6, \\ & F_{1,1} + F_{1,2} + F_{1,3} \leq 8, \\ & F_{2,1} + F_{2,2} + F_{2,3} \leq 5, \\ & F_{3,1} + F_{3,2} + F_{3,3} \leq 4, \\ & F_{1,1}, F_{1,2}, F_{1,3}, F_{2,1}, F_{2,2}, F_{2,3}, F_{3,1}, F_{3,2}, F_{3,3} \geq 0. \end{cases}$$

The first three equations represent demand constraints, the following three inequations are supply constraints, and the last line guarantees non-negative flows. Fig. 2 (b) represents an optimal solutionfor `EMD`: $F_{1,1} = 4$, $F_{1,3} = 2$, $F_{2,2} = 3$, $F_{3,3} = 4$, and $F_{1,2} = F_{2,1} = F_{2,3} = F_{3,1} = F_{3,2} = 0$. Only links of cost 1 are used and so the cost of the solution is$C^{\text{EMD}} = \sum_{j\in\{1,2,3\}} Y_j = 13$. This solution is not admissible for `EMD-CC`. Indeed node $v_1 \in V$ sends non-zero flow only to demand nodes $v'_1 \in V'$ and $v'_3 \in V'$ (that is $F_{1,1} > 0$, $F_{1,2} = 0$, and $F_{1,3} > 0$), and the nodes v'_1 and v'_3 do not induce a connected subgraph because $\{v'_1, v'_3\} \notin E'$ (and so $\{v'_1, v'_3\} \notin \mathcal{H}'$). One can observe that there does not exist an admissible solution of cost 13 for `EMD-CC` even when $M = |E(B)| = 9$.

Fig. 2 (c) represents an admissible solution for `EMD-CC` for$5 \leq M \leq 9$ and for any real number $x \in]0, 3]$: $F_{1,1} = 4$, $F_{1,2} = x$, $F_{1,3} = 2$, $F_{2,2} = 3 - x$, $F_{3,3} = 4$, and $F_{2,1} = F_{2,3} = F_{3,1} = F_{3,2} = 0$.

Figure 2 (a) Example of supply and demand graphs. **(b)** Optimal solution for EMD of cost 13. **(c)** Optimal solution for EMD-CC of cost $6x+13$ for $5 \leq M \leq 9$. **(d)** Optimal solution for EMD-CC of cost 29 for $M = 4$. **(e)** Optimal solution for EMD-CC of cost 45 for $M = 3$.



The total cost is $\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F_{i,j}c_{i,j} = 6x + 13$. Thus, $\lim_{x\to 0}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F_{i,j}c_{i,j} = 13$ but we cannot obtain an admissible solution of cost 13 because $x > 0$. Fig. 2 (d) shows an optimal solution for EMD-CC for$M = 4$: $F_{1,1} = 4$, $F_{2,2} = 3$, $F_{2,3} = 2$, $F_{3,3} = 4$, and $F_{1,2} = F_{1,3} = F_{2,1} = F_{3,1} = F_{3,2} = 0$. The total cost is $\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F_{i,j}c_{i,j} = 29$. Fig. 2 (e) describes an optimal solution for EMD-CC for$M = 3$: $F_{1,3} = 6$, $F_{2,2} = 3$, $F_{3,1} = 4$, and $F_{1,1} = F_{1,2} = F_{2,1} = F_{2,3} = F_{3,2} = F_{3,3} = 0$. The total cost is $\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F_{i,j}c_{i,j} = 45$. One can observe that there does not exist an admissible solutionfor EMD-CC when $0 \leq M \leq 2$.

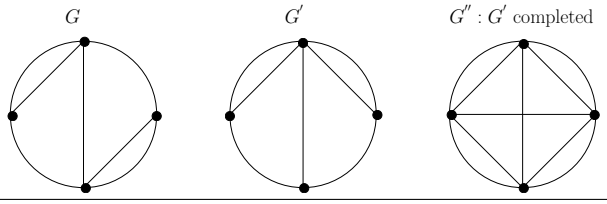
Solutions do not define a metric. Consider the case where the vertices of the two graphs live in a metric space $\mathcal{C}$, and let $d_{\mathcal{C}}$ be a metric for that space. Under these assumptions, solutions of the linear program specified by the system of Eq. (1) define a metric provided that the production satisfies the demand [10]. Such a property does not hold for solutions computed with connectivity constraints, since the symmetry and the triangle inequality do not hold in general (Fig. 3).

In this example, we consider three graphs G, G' and G'' as follows:G is a path, G' is a star graph (every node but one has degree one), G and G' have the same number of nodes, and G'' is G' (or G) completed to contain all edges. In the tree cases, we assume that the sum of masses associated to the vertices is one. We first observe that the triangle inequality fails for $d_{\text{EMD-CC}}$. To see why, first observe that the hypothesis made on the three graphs implies that

$$0 < C^{\text{EMD-CC}}(G, G') \not\leq C^{\text{EMD-CC}}(G, G'')+C^{\text{EMD-CC}}(G', G'') = C^{\text{EMD}}(G, G'')+C^{\text{EMD}}(G', G'') = 0.$$

Because the LP fully satisfies the demand, the previous inequality on total costs translates into an inequality on distances.

Figure 3 Solutions of EMD-CC may not satisfy the triangle inequality.



2.3 Maximum-flow under Cost and Connectivity Constraints Problem

Since EMD-CC does not necessarily admit a solution, we define a similar problem, called *maximum-flow under cost and connectivity constraints problem* (EMD-CCC), that aims at computing the

largest volume of flow that can be supported respecting the connectivity constraints and such that the total cost is less thana given bound C. We define the following upper bound for the maximum total cost of any admissible flow:

$$C_{max} = \sum_{j \in \mathcal{J}} Y_j \max_{i \in \mathcal{I}, j \in \mathcal{J}} c_{i,j}. \tag{5}$$

Formally, given C and M, $0 \leq C \leq C_{max}$, $0 \leq M \leq |E(B)|$, `EMD-CCC` is defined as follows:

$$\begin{cases} \text{Max} & \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} F_{i,j} & \\ & \sum_{i \in \mathcal{I}} F_{i,j} \leq Y_j & \forall j \in \mathcal{J}, \\ & \sum_{j \in \mathcal{J}} F_{i,j} \leq X_i & \forall i \in \mathcal{I}, \\ & F_{i,j} \geq 0 & \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \\ & \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} F_{i,j}\ c_{i,j} \leq C & \\ & \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J} | F_{i,j} > 0} 1 \leq M & \\ & H' = \{v'_j \mid F_{i,j} > 0, v_i \in H, i \in \mathcal{I}, j \in \mathcal{J}\} \in \mathcal{H}' & \forall H \in \mathcal{H}. \end{cases} \tag{6}$$

From which we define the *total number of edges*, the *total flow*, the *total cost*, and their ratio:

$$M^{\text{EMD-CCC}} = \sum_{i,j | F_{i,j} > 0} 1, F^{\text{EMD-CCC}} = \sum_{i,j} F_{i,j}, C^{\text{EMD-CCC}} = \sum_{i,j} F_{i,j} c_{i,j}, d_{\text{EMD-CCC}} = \frac{C^{\text{EMD-CCC}}}{F^{\text{EMD-CCC}}}. \tag{7}$$

Remark 1. *While developing algorithms (Section 4), we shall actually discard the constraint involving the upper-bound M in input, or equivalently $M = |E(B)|$. In doing so, we avoid speculating on a* good *value for M. On the other hand, while running experiments, we precisely assess the size of transport plans computed, by comparing the number of edges carrying flow against the size of the input graphs. As we shall see, in all our experiments, we end up with transport plans of linear size in the number of vertices,that is $M^{EMD\text{-}CCC} = \theta(|V| + |V'|)$.*

3 Complexity Results

3.1 The Problems are Difficult even for Simple Instances

We first show in Lemma 1 that the decision version of `EMD-CC` is in NP. We then prove that the problem is NP-complete and not in APX.

Lemma. 1. *The decision version of* `EMD-CC` *is in NP.*

Proof of Lemma 1. Consider any instance of `EMD-CC`.Without loss of generality, assume that $\sum_{i \in \mathcal{I}} X_i \geq \sum_{j \in \mathcal{J}} Y_j$. Let F be a solution. We can check in polynomial time if $\sum_{i \in \mathcal{I}} F_{i,j} = Y_j$, if $\sum_{j \in \mathcal{J}} F_{i,j} \leq X_i$, if $F_{i,j} \geq 0$, and if $\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J} | F_{i,j} > 0} 1 \leq M$ for all $i \in \mathcal{I}$, $j \in \mathcal{J}$.

We now prove that we can decide in polynomial time if $H' = \{v'_j \mid F_{i,j} > 0, v_i \in H, i \in \mathcal{I}, j \in \mathcal{J}\} \in \mathcal{H}'$ for all $H \in \mathcal{H}$. We prove the result by induction on the size $|H|$ of the subset of the supply nodes. First, we can verify in polynomial time if the constraints of connectivity are satisfied for all $H \in \mathcal{H}$, $|H| \leq 2$. Suppose now that $H' = \{v'_j \mid F_{i,j} > 0, v_i \in H, i \in \mathcal{I}, j \in \mathcal{J}\} \in \mathcal{H}'$ for all $H \in \mathcal{H}$, $|H| \leq t$,with $t \leq |V|-1$. We prove that $H' = \{v'_j \mid F_{i,j} > 0, v_i \in H, i \in \mathcal{I}, j \in \mathcal{J}\} \in \mathcal{H}'$ for all $H \in \mathcal{H}$, $|H| \leq t+1$. Let $H \in \mathcal{H}$ such that $|H| = t+1$. Since H induces a connected component, then there exists a node $u \in H$ such that $H \setminus \{u\}$ induces a connected component. The constraint of connectivity is satisfied for $H \setminus \{u\}$ because $|H \setminus \{u\}| = t$. Furthermore there

exists a node $v \in H \setminus \{u\}$ such that $\{u,v\} \in E$. The constraint of connectivity is satisfied for $\{u,v\}$ because $|\{u,v\}| = 2$. We deduce that the constraint of connectivity is satisfied for H because $(H \setminus \{u\}) \cap \{u,v\} \neq \emptyset$. Thus, $H' = \{v'_j \mid F_{i,j} > 0, v_i \in H, i \in \mathcal{I}, j \in \mathcal{J}\} \in \mathcal{H}'$ for all $H \in \mathcal{H}$, $|H| \leq t+1$.

Finally we can decide in polynomial time if $H' = \{v'_j \mid F_{i,j} > 0, v_i \in H, i \in \mathcal{I}, j \in \mathcal{J}\} \in \mathcal{H}'$ for all $H \in \mathcal{H}$ because it reduces to decide if $H' = \{v'_j \mid F_{i,j} > 0, v_i \in H, i \in \mathcal{I}, j \in \mathcal{J}\} \in \mathcal{H}'$ for all $H \in \mathcal{H}$, $|H| \leq 2$. □

Fromthe proof of Lemma 1, the constraints of connectivity can be written as follows:

$$\begin{cases} H'_i = \{v'_j \mid F_{i,j} > 0, j \in \mathcal{J}\} \in \mathcal{H}' & \forall i \in \mathcal{I}, \\ H'_{i_1,i_2} = \{v'_j \mid F_{i,j} > 0, i \in \{i_1,i_2\}, j \in \mathcal{J}\} \in \mathcal{H}' & \forall i_1, i_2 \in \mathcal{H}, \{v_{i_1}, v_{i_2}\} \in E. \end{cases}$$

In the following, we prove hardness results. In our reductions, we use the strongly NP-complete problem 3-Partition [6]. Let $m \geq 1$ be any integer.Given a set $S = \{n_1, n_2, \ldots, n_{3m}\}$ of $3m$ positive integers, 3-Partition problem consists in deciding if S can be partitioned into m subsets such that the sum of the numbers in each subset is equal.

Theorem. 1. *The decision version of* `EMD-CC` *is NP-complete even if:*

- *the demand graph G' is a complete graph;*
- *all the volumes of demands are equal, $Y_j = Y_{j'}$ for all $j, j' \in \mathcal{J}$;*
- *all the unitary costs are equal to one, $c_{i,j} = 1$ for all $i \in \mathcal{I}$, $j \in \mathcal{J}$;*
- *and the total volume of demands equals the total volume of supplies, $\sum_{i\in\mathcal{I}} X_i = \sum_{j\in\mathcal{J}} Y_j$.*

Proof of Theorem 1. Consider an instance of3-Partition problem. Let $m \geq 1$ be any integer and let $S = \{n_1, n_2, \ldots, n_{3m}\}$ be a set of $3m$ positive integers.

We construct the instance of `EMD-CC` as follows. Set $|\mathcal{I}| = 3m$ and $|\mathcal{J}| = m$. Set $X_i = n_i$ for all $i \in \mathcal{I}$. Let $Z = \sum_{i\in\mathcal{I}} X_i$. Without loss of generality, let $Y_j = Y$ with $Z = mY$. Set $c_{i,j} = 1$ for all $i \in \mathcal{I}, j \in \mathcal{J}$. Let $G = (V,E)$ be any connected graph and let $G' = (V', V' \times V')$. Let $M = 3m$. Since G' is a complete graph, the connectivity constraints are always satisfied and so `EMD-CC` can be written as follows:

$$\begin{cases} \text{Min} & \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} F_{i,j}\ c_{i,j} & \\ & \sum_{i\in\mathcal{I}} F_{i,j} = Y_j & \forall j \in \mathcal{J}, \\ & \sum_{j\in\mathcal{J}} F_{i,j} \leq X_i & \forall i \in \mathcal{I}, \\ & F_{i,j} \geq 0 & \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \\ & \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}|F_{i,j}>0} 1 \leq M. & \end{cases}$$

We prove that there is an admissible solution for `EMD-CC` ifand only if there is a solution for the instance of3-Partition problem.

($\Leftarrow$) Assume there is a solution for the instance of3-Partition problem, that is S can be partitioned into m subsets $S_1, S_2, \ldots, S_m$ such that the sum of the numbers in each subset is equal. We construct our solution for `EMD-CC` as follows. For all $i \in \mathcal{I}, j \in \mathcal{J}$, if $n_i \in S_j$, then set $F_{i,j} := n_i$, otherwise set $F_{i,j} := 0$. By construction,we have $0 \leq F_i \leq X_i$ for all $i \in \mathcal{I}$, $j \in \mathcal{J}$. Since $S_1, S_2, \ldots, S_m$ is a solution for the instance of3-Partition problem, then $\sum_{i\in\mathcal{I}} F_{i,j} = Y_j$ for all $j \in \mathcal{J}$. Finally we prove that the number of edges of B that support non-zero flow is (at most) $M = 3m$. By construction, $F_{i,j_1} F_{i,j_2} = 0$ for all $i \in \mathcal{I}$, $j_1, j_2 \in \mathcal{J}$. Thus, for all $i \in \mathcal{I}$, there is at most one edge adjacent to v_i that supports non-zero flow. Thus, the solution is admissible because $|\mathcal{I}| = 3m$.

($\Rightarrow$) Assume there is an admissible solution for EMD-CC. Since $\sum_{i\in\mathcal{I}} X_i = \sum_{j\in\mathcal{J}} Y_j$, then $\sum_{j\in\mathcal{J}} F_{i,j} > 0$ for all $i \in \mathcal{I}$. In other words, there is at least one edge adjacent to v_i that supports non-zero flow for all $i \in \mathcal{I}$. Furthermore there is at most one edge adjacent to v_i that supports non-zero flow for all $i \in \mathcal{I}$ because $M = 3m = |\mathcal{I}|$. Thus, for all $i \in \mathcal{I}$, there is exactly one edge adjacent to v_i that supports non-zero flow. We construct a solution for the instance of 3-Partition problem as follows. For all $i \in \mathcal{I}, j \in \mathcal{J}$, if $F_{i,j} > 0$, then $n_i \in S_j$, otherwise set $n_i \notin S_j$. By hypothesis (existence of an admissible flow), the sum of the numbers of S_j is Y because $\sum_{i\in\mathcal{I}} X_i = \sum_{j\in\mathcal{J}} Y$. Thus, there is a solution for the instance of3-Partition problem.

In conclusion, the decision version of EMD-CC is strongly NP-complete because it is in NP (Lemma 1) and because3-Partition problem is stronglyNP-complete [6]. □

Note that $M = \theta(\sqrt{|E(B)|})$ in the proof ofTheorem 1.

From Theorem 1, we deduce in Corollary 1 that the decision version of EMD-CCC is NP-complete. Indeed, given a maximum cost C and a maximum number of edges M, the problem of deciding if there exists a flow F satisfying the connectivity constraints and such that $\sum_{i\in\mathcal{I},j\in\mathcal{J}} F_{i,j} = \sum_{j\in\mathcal{J}} Y_j$, $\sum_{i\in\mathcal{I},j\in\mathcal{J}} F_{i,j}c_{i,j} \leq C$, and $\sum_{i\in\mathcal{I},j\in\mathcal{J}|F_{i,j}>0} 1 \leq M$, is equivalent to the problem of deciding if there is an admissible solution for EMD-CC with C as input.

Corollary. 1. *The decision version of* EMD-CCC *is NP-complete.*

We now prove in Theorem 2 that EMD-CC is hard to approximate even for simple classes of instances.

Theorem. 2. *The* EMD-CC *is not in APX even if:*

- *all the volumes of demands are equal, $Y_j = Y_{j'}$ for all $j, j' \in \mathcal{J}$;*
- *and there are only two possible unitary costs for edges of the bipartite graph B, that is $c_{i,j} \in \{1, K\}$ for all $i \in \mathcal{I}, j \in \mathcal{J}$, where $K > 1$.*

Proof of Theorem 2. Suppose there exists a constant $k > 1$ such that there is a polynomial time k-approximation algorithm for EMD-CC.

Consider an instance of3-Partition problem. Let $m \geq 1$ be any integer and let $S = \{n_1, n_2, \ldots, n_{3m}\}$ be a set of $3m$ positive integers. Set $|\mathcal{I}| = 3m + 1$ and $\mathcal{I}^- = \mathcal{I} \setminus \{3m + 1\}$. Set $|\mathcal{J}| = m + 1$ and $\mathcal{J}^- = \mathcal{J} \setminus \{m + 1\}$. Set $X_i = n_i$ for all $i \in \mathcal{I}^-$ Let $Z = \sum_{i\in\mathcal{I}^-} X_i$. Without loss of generality, let $Y_j = Y$ with $Z = mY$ for all $j \in \mathcal{J}^-$. Set $X_{3m+1} = Z$ and $Y_{m+1} = Y$. Set $c_{i,j} = 1$ for all $i \in \mathcal{I}^-, j \in \mathcal{J}^-$. Set $c_{3m+1,m+1} = 1$. Set $c_{3m+1,j} = K = k(Y + Z)$ for all $j \in \mathcal{J}^-$. Set $c_{i,m+1} = K = k(Y + Z)$ for all $i \in \mathcal{I}^-$. Let $G = (V, E)$ be any connected graph and let $G' = (V', V' \times V')$. The connectivity constraints are always satisfied because G' is a complete graph. Let $M = 3m + 1$.

There exists a solution for EMD-CC such that $\sum_{j\in\mathcal{J}^-} F_{3m+1,j} + \sum_{i\in\mathcal{I}^-} F_{i,m+1} = 0$if and only if thereis a solution for the instance of3-Partition problem (Theorem 1). The cost of this solution is $Y + \sum_{i\in\mathcal{I},j\in\mathcal{J}} F_{i,j}c_{i,j} = Y + Z$. We prove that if there does not exist a solution for the instance of3-Partition problem, then the cost of any admissible solution for EMD-CC is at least $Z - 1 + K$. Suppose that there does not exist a solution for the instance of3-Partition problem. Thus, we have $\sum_{j\in\mathcal{J}} F_{3m+1,j} > 0$. There are two cases.

- If $F_{3m+1,m+1} = 0$, then $\sum_{i\in\mathcal{I}^-} F_{i,3m+1} = Y$. Thus, we get $\sum_{i\in\mathcal{I}^-}\sum_{j\in\mathcal{J}^-} F_{i,j} \leq Z - Y$ and $\sum_{j\in\mathcal{J}^-} F_{3m+1,j} \geq Y$. Then, $\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F_{i,j}c_{i,j} \geq 2YK + Z - Y \geq Z - 1 + K$.
- If $F_{3m+1,m+1} > 0$, then $\sum_{i\in\mathcal{I}^-}\sum_{j\in\mathcal{J}^-} F_{i,j} \leq Z - 2$. Thus, we get $\sum_{j\in\mathcal{J}^-} F_{3m+1,j} \geq 2$. Then, $\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F_{i,j}c_{i,j} \geq 2K + Y + Z - 2 \geq Z - 1 + K$.

Since $K = k(Y + Z)$, we have $(Z - 1 + K)/Z > k$. As we have supposed that there exists a polynomial time k-approximation algorithm for EMD-CC, then if there is a solution of cost $Y + Z$, the k-approximation algorithm returns such a solution (otherwise the approximation ratio would be wrong); otherwise (solution of cost at least $Z-1+K$), the k-approximation ratio would return a solution with cost at least $Z - 1 + K$. Thus, the polynomial time (k-approximation) algorithm solves3-Partition problem which is a strongly NP-completeproblem [6]. A contradiction, unless P=NP. □

3.2 PTAS when the Number of Active Edges is not Bounded

We prove in Theorem 3 a Polynomial Time Approximation Scheme for EMD-CC when $M = |E(B)|$.

Theorem. 3. *Let $M = |E(B)|$. For any $\varepsilon > 0$, there is a polynomial time $(1+\varepsilon)$-approximation algorithm for* EMD-CC.

Proof of Theorem 3. Consider an instance of EMD-CC. We construct an auxiliary instance as follows. The graphs G, G', and B and the cost $c_{i,j}$ for all $i \in \mathcal{I}$, $j \in \mathcal{J}$ are those of the original instance. Let $\varepsilon' > 0$ be a real value such that $X_i - |\mathcal{J}|\varepsilon' > 0$ for all $i \in \mathcal{I}$ and such that $Y_j - |\mathcal{I}|\varepsilon' > 0$ for all $j \in \mathcal{I}$. We denote by X'_i the volume of supply for all $i \in \mathcal{I}$ in the auxiliary instance. Set $X'_i = X_i - |\mathcal{J}|\varepsilon'$ for all $i \in \mathcal{I}$. We denote by Y'_i the volume of demand for all $j \in \mathcal{J}$ in the auxiliary instance. Set $Y'_j = Y_j - |\mathcal{I}|\varepsilon'$ for all $j \in \mathcal{I}$. Let F' be an optimal solution for this auxiliary instance for EMD. Recall that this can be done in polynomial time since it reduces to solve a linear program. The cost of F' is $\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F'_{i,j}c_{i,j}$.

We now construct an admissible solution F for the original instance for EMD-CC as follows. Set $F_{i,j} = F'_{i,j} + \varepsilon'$ for all $i \in \mathcal{I}$, $j \in \mathcal{J}$. All the constraints of connectivity are satisfied because $F_{i,j} > 0$ for all $i \in \mathcal{I}$, $j \in \mathcal{J}$. Recall that $M = |E(B)|$. Thus, the solution is admissible. The cost of F is $\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F_{i,j}c_{i,j} = |\mathcal{I}||\mathcal{J}|\varepsilon' + \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F'_{i,j}c_{i,j}$.

Let F^* be an optimal solution for the original instance of EMD-CC. Observe that:

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F'_{i,j}c_{i,j} \leq \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F^*_{i,j}c_{i,j} \leq |\mathcal{I}||\mathcal{J}|\varepsilon' + \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F'_{i,j}c_{i,j} = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F_{i,j}c_{i,j}.$$

We finally choose $\varepsilon' > 0$ suchthat:

$$|\mathcal{I}||\mathcal{J}|\varepsilon' + \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F'_{i,j}c_{i,j} \leq (1+\varepsilon)\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F'_{i,j}c_{i,j}.$$

Thus, weget:

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F_{i,j}c_{i,j} = |\mathcal{I}||\mathcal{J}|\varepsilon' + \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F'_{i,j}c_{i,j} \leq (1+\varepsilon)\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F^*_{i,j}c_{i,j}.$$

We get a polynomial time $(1 + \varepsilon)$-approximation algorithm for EMD-CC because F' is obtained by solving a linear program and F is directly deduced from F'. □

Fromthe proof of Theorem 3, an interesting problem is to determine the minimum number of edges to add in an optimal solution for EMD in order to get an admissible solutionfor EMD-CC.

From Theorem 3, we deduce Corollary 2.

Corollary. 2. *Let $M = |E(B)|$ and let C be any real such that $0 \leq C \leq C_{max}$. For any $\varepsilon > 0$, there is a polynomial time $(1 + \varepsilon)$-approximation algorithm for* EMD-CCC.

4 Algorithms

In Section 4.1, we present a greedy algorithm, called `Alg-EMD-CCC-G`, to solve `EMD-CCC`. In Section 4.2, we design algorithm `Alg-EMD-CCC-G-I`, which uses `Alg-EMD-CCC-G` in an iterative fashion to deal withdifferent cost upper-bounds.

4.1 Greedy Algorithm `Alg-EMD-CCC-G`

Recall that we wish to solve `EMD-CCC`, while alleviating constraint involving the upper-bound M(remark of Section 1). Indeed, Algorithm 1 is used with $M = |E(B)|$. We do so with a greedy strategy, `Alg-EMD-CCC-G` (Algorithm 1). `Alg-EMD-CCC-G` returns the cost, the number of edges, the total flows, and the flows for every edge $\{v_i, v'_j\} \in E(B)$ of the solution found, namely $C^{\text{Alg-EMD-CCC-G}}$, $M^{\text{Alg-EMD-CCC-G}}$, $F^{\text{Alg-EMD-CCC-G}}$, and $F_{i,j}^{\text{Alg-EMD-CCC-G}}$ for every edge $\{v_i, v'_j\} \in E(B)$, respectively.

The algorithm greedily selects (Line 4 of Algorithm 1) edges of the bipartite graph that can support flow without violating the connectivity constraints and the cost upper-bound. After such a selection, Algorithm 2 updates the set of candidate edges for the next step of selection in respect with the connectivity constraints.

Before presenting in detail Algorithm 1, let us first define some notations. For any subset $S \subseteq V$, the open neighborhood $N_G(S)$ of S is the set of vertices in $V \setminus S$ having a neighbor in S and the closed neighborhood of S, denoted by $N_G[S]$, is defined as $N(S) \cup S$. If $S = \{v\}$, we use $N_G(v)$ and $N_G[v]$ instead of $N_G(\{v\})$ and $N_G[\{v\}]$,respectively. Similarly, for any subset $S' \subseteq V'$, the open neighborhood $N_{G'}(S')$ of S' is the set of vertices in $V' \setminus S'$ having a neighbor in S' and the closed neighborhood of S', denoted by $N_{G'}[S']$, is defined as $N(S') \cup S'$. If $S' = \{v'\}$, we use $N_{G'}(v')$ and $N_{G'}[v']$ instead of $N_{G'}(\{v'\})$ and $N_{G'}[\{v'\}]$,respectively. We denote by $\Delta(G)$ ($\Delta(G')$, respectively) the maximum degree of the graph G (G', respectively).

Algorithm 1. The inputs of Algorithm 1 are the supply graph $G = (V, E)$, the demand graph $G' = (V', E')$, acost upper-bound C, and a maximum number M of edges that can support non-zero flow.

We now describe the variables used in Algorithm 1. For all $i \in \mathcal{I}$, the value $x_i \geq 0$ represents the current volume of supply of node $v_i \in V$, and so $X_i - x_i$ is the current amount of flow sent by v_i. For all $j \in \mathcal{J}$, the value $y_j \geq 0$ represents the current volume of demand of node $v'_j \in V'$, and so $Y_j - y_j$ is the current amount of flow received by v'_j. For all $i \in \mathcal{I}, j \in \mathcal{J}$, the variable

Algorithm 1 Greedy algorithm `Alg-EMD-CCC-G` for `EMD-CCC`.

Require: $G = (V, E)$, $G' = (V', E')$, C, M.

1: **for** all $i \in \mathcal{I}, j \in \mathcal{J}$ **do**

2: $F_{i,j} := 0$; $C_F := 0$; $M_F := 0$; $x_i := X_i$; $y_j := Y_j$; $b_{i,j} := 1$

3: **while** $C_F < C$, $M_F \leq M - 1$, and $\exists (i, j)$ such that $b_{i,j} = 1$, $x_i > 0$, and $y_j > 0$ **do**

4: $(i_t, j_t) = \underset{i \in \mathcal{I}, x_i > 0, j \in \mathcal{J}, y_j > 0}{\arg\min} c_{i,j} b_{i,j}$; $f := \min((C - C_F)/c_{i_t,j_t}, \min(x_{i_t}, y_{j_t}))$

5: $F_{i_t,j_t} := F_{i_t,j_t} + f$; $C_F := C_F + f.c_{i_t,j_t}$; $M_F := M_F + 1$; $x_{i_t} := x_{i_t} - f$; $y_{j_t} := y_{j_t} - f$

6: Update of $b_{i,j}$ for all $i \in \mathcal{I}$, $j \in \mathcal{J}$ using Algorithm 2

7: $C^{\text{Alg-EMD-CCC-G}} := C_F$; $M^{\text{Alg-EMD-CCC-G}} := M_F$; $F^{\text{Alg-EMD-CCC-G}} = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} F_{i,j}$; $F_{i,j}^{\text{Alg-EMD-CCC-G}} = F_{i,j}$ for all $i \in \mathcal{I}$, $j \in \mathcal{J}$

8: **return** $C^{\text{Alg-EMD-CCC-G}}$, $M^{\text{Alg-EMD-CCC-G}}$, $F^{\text{Alg-EMD-CCC-G}}$, and $F_{i,j}^{\text{Alg-EMD-CCC-G}}$ for all $i \in \mathcal{I}$, $j \in \mathcal{J}$

Algorithm 2 Update of the boolean function b used in Algorithm 1.

Require: $G = (V, E)$, $G' = (V', E')$, (i_t, j_t), $b_{i,j}$, $F_{i,j}$, x_i, for all $i \in \mathcal{I}$, $j \in \mathcal{J}$.
Ensure: Binary values $b_{i,j}$ for all $i \in \mathcal{I}$, $j \in \mathcal{J}$.

```
1: if X_{i_t} − x_{i_t} − F_{i_t,j_t} = 0 then
2:    for all j such that v'_j ∈ N_{G'}(v'_{j_t}) do b_{i_t,j} := 1
3:    for all j such that v'_j ∉ N_{G'}(v'_{j_t}) do b_{i_t,j} := 0
4:    for all i such that v_i ∈ N_G(v_{i_t}) do
5:       if X_i − x_i = 0 then
6:          for all j such that v'_j ∉ N_{G'}[v'_{j_t}] do b_{i,j} := 0
7: else
8:    for all j such that v'_j ∈ N_{G'}(v'_{j_t}) do b_{i_t,j} := 1
9:    for all i such that v_i ∈ N_G(v_{i_t}) do
10:      if X_i − x_i = 0 then
11:         for all j such that v'_j ∈ N_{G'}[v'_{j_t}] do
12:            if b_{i,j} = 0 then
13:               b_{i,j} := 1
14:               for all k such that v_k ∈ N_G(v_i) do
15:                  if X_k − x_k > 0 and b_{k,j} = 0 do b_{i,j} := 0
16: return b_{i,j} for all i ∈ I, j ∈ J
```

$b_{i,j}$ is used to encode if the edge $\{v_i, v'_j\} \in E(B)$ can support non-zero flow in respect with the constraints. In other words, $b_{i,j} = 1$ if the edge $\{v_i, v'_j\}$ is an edge candidate ($b_{i,j} = 0$ otherwise). For all $i \in \mathcal{I}, j \in \mathcal{J}$, the variable $F_{i,j}$ represents the current flow sent from $v_i \in V$ to $v'_j \in V'$. The variable C_F represents the total cost of the current flow. The variable M_F is the current number of edges of B that support non-zero flow. Initially, $x_i = X_i$, $y_j = Y_j$, $F_{i,j} = 0$, $C_F = 0$, $M_F = 0$, and $b_{i,j} = 1$ for all $i \in \mathcal{I}, j \in \mathcal{J}$.

We are now ready to precisely explain the core of Algorithm 1. While the current cost C_F is less than the given cost upper-bound C, while the current number M_F of edges of B that support non-zero flow is strictly less than the given upper-bound M, and while there exists an edge candidate $\{v_i, v'_j\}$ such that $x_i, y_j > 0$ (that is such that a positive flow can be supported by $\{v_i, v'_j\} \in E(B)$), then an edge $\{v_{i_t}, v'_{j_t}\} \in E(B)$ is selected (Line 4). Then, the maximum amount of flow f that can be supported by the edge $\{v_{i_t}, v'_{j_t}\}$ is computed. Line 5 updates the values of F_{i_t,j_t}, C_F, M_F, x_{i_t}, and y_{j_t}. Line 6 updates the boolean function b using Algorithm 2 for all $i \in \mathcal{I}, j \in \mathcal{J}$.

Algorithm 1 finally returns $C^{\text{Alg-EMD-CCC-G}}$, $M^{\text{Alg-EMD-CCC-G}}$, $F^{\text{Alg-EMD-CCC-G}}$, and $F^{\text{Alg-EMD-CCC-G}}_{i,j}$ for every edge $\{v_i, v'_j\} \in E(B)$.

Algorithm 2. The inputs of Algorithm 2 are the supply graph $G = (V, E)$, the demand graph $G' = (V', E')$, the pair (i_t, j_t) representing the edge $\{v_{i_t}, v'_{j_t}\} \in E(B)$ selected for supporting flow, and for all $i \in \mathcal{I}, j \in \mathcal{J}$, the boolean variable $b_{i,j}$, the variable $F_{i,j}$, and the value x_i representing the current volume of supply of node $v_i \in V$.

We now prove in Lemma 2 that Algorithm 2 updates the set of candidate edges that can support flow in respect with the connectivity constraints.

Lemma. 2. *Algorithm 2 updates $b_{i,j}$ for all $i \in \mathcal{I}, j \in \mathcal{J}$.*

Proof of Theorem 2. Lines 1-6of Algorithm 2 update the boolean function b if $X_i - (x_{i_t} + F_{i_t,j_t}) = 0$, that is if the supply node v_{i_t} sends flow for the first time. In that case, all the neighbors of v'_{j_t} in G' can receive flow from v_{i_t} (Line 2) and all the other nodes of G' cannot receive flow from

Algorithm 3 Algorithm `Alg-EMD-CCC-G-I`.

Require: $G = (V, E)$, $G' = (V', E')$, C_{inf}, C_{sup}.
1: F_{inf} := total flow of `Alg-EMD-CCC-G` with G, G', $C := C_{inf}$, $M := |E(B)|$
2: F_{sup} := total flow of `Alg-EMD-CCC-G` with G, G', $C := C_{sup}$, $M := |E(B)|$
3: **if** $F_{inf} < F_{sup}$ **then**
4: `Alg-EMD-CCC-G-I` with G, G', C_{inf}, $C_{sup} := C$
5: `Alg-EMD-CCC-G-I` with G, G', $C_{inf} := C$, C_{sup}

v_{i_t} (Line 3). Furthermore, all the neighbors of v_{i_t} in G that do not have sent flow, cannot send flow to the nodes of G' that are not neighbors of v'_{j_t} in G' (Lines 4-6).

Lines 7-15 update the boolean function b if $X_i - (x_{i_t} + F_{i_t,j_t}) \neq 0$, that is if the supply node v_{i_t} has already sent flow before the current step. All the neighbors of v'_{j_t} in G' can receive flow from v_{i_t} (Line 8). Every neighbor v_i of v_{i_t} in G that does not have sent flow, can send flow to every neighbor v'_j of v'_{j_t} in G' if $b_{i,j} = 0$, that is if the edge $\{v_i, v'_j\}$ does not support flow (Lines 9-13). Furthermore, for every neighbor v_i of v_{i_t} in G that does not have sent flow, for every neighbor v'_j of v'_{j_t} in G' such that $b_{i,j} = 0$, and for every neighbor v_k of v_i in G that has already sent flow and such that v_k cannot send flow to v'_j, then we set that v_i cannot send flow to v'_j, that is $b_{i,j} = 0$ (Lines 9-15).

Algorithm 2 finally returns the variables $b_{i,j}$ for all $i \in \mathcal{I}, j \in \mathcal{J}$. □

We finally prove in Lemma 3 the time-complexity of Algorithm 1.

Lemma. 3. *The time-complexity of* `Alg-EMD-CCC-G` *is* $O(|V|^3|V'|^2)$.

Proof of Theorem 3. The time-complexity of Algorithm 2 is $O(|V'| + \Delta(G)^2\Delta(G'))$. Indeed, the time-complexity of the first part (Lines 1-6) is $O(|V'| + \Delta(G)\Delta(G'))$, and the time-complexity of the second part (Lines 7-15) is $O(\Delta(G)^2\Delta(G'))$. The time-complexity of Line 4 of Algorithm 1 to perform the edge selection is $O(|V||V'|)$. The number of steps (number of iterations of the while loop) of Algorithm 1 is at most $|V||V'|$. Thus, we deduce that the time-complexity of Algorithm 1 is $O(|V|^3|V'|^2)$ because $\Delta(G) = O(|V|)$ and $\Delta(G') = O(|V'|)$. □

4.2 Iterative Algorithm `Alg-EMD-CCC-G-I`

As previously explained, the maximum cost is an input of `Alg-EMD-CCC-G`. In order to compute different flow solutions for differentcost upper-bounds in the range $[0, C_{max}]$ (Eq. (5)), we design `Alg-EMD-CCC-G-I` (Algorithm 3). Thatis, `Alg-EMD-CCC-G-I` returns a collection of transport plans, from which one may select the one with the largest flow, or the one with the best ratio $d_{\text{EMD-CCC}}$ (Eq. 7).

The inputs of `Alg-EMD-CCC-G-I` are those of `Alg-EMD-CCC-G` and two additional inputs: C_{inf} and C_{sup}. `Alg-EMD-CCC-G-I` computes a flow solution of cost at most C_{inf} (Line 1) and a flow solution of cost at most C_{sup} (Line 2). Note that it is possible that these two flow solutions have been previously computed. If the total flows are different, then we refine our calculation by computing a flow solution of cost at most $(C_{inf} + C_{max})/2$. This is done in Lines 3-5 by the two recursive calls of `Alg-EMD-CCC-G-I`. To do that, we initially apply `Alg-EMD-CCC-G-I` with $C_{inf} := 0$ and $C_{sup} := C_{max}$.

5 Experiments

We present experiments on protein models and synthetic landscapes derived from Voronoi diagrams.

5.1 Implementations

Algorithms. We implemented our algorithms in generic C++, using the Boost Graph Library to represent the graphs. Each algorithm is templated by a traits class specifying two main parameters, namely (i) the information associated with a vertex (quantity of supply or demand, coordinates), and (ii) a distance functor used to compute the unit cost of flow between a source node and a demand node. This design makes it possible to instantiate the algorithms on various types of graphs, and to investigate the role of the cost by changing the distance functor.

In all experiments, the transport plan considered is the one returned by `Alg-EMD-CCC-G`, run with thecost upper bound C_{max} (Eq. (5)).

To make comparisons, we also implemented a procedure (`Alg-EMD-LP`) writing the linear program of Eq. (1) in the Mathematical Programming System format, so as to solve it with a LP solver. The corresponding cost is denoted C^{EMD}(Eq. (2)). Solutions of the LP are expected to violate connectivity constraints, so that we also implemented a *checker*. Consider the solution of the LP program. For a vertex of the source graph, let the *target vertices* be the vertices of the demand graph corresponding to edges along which strictly positive flow circulates. A vertex of the source graph violates the connectivity constraints if the subgraph induced by its target vertices is not connected. Likewise, for an edge of the input graph (or larger subgraphs), we check the connectedness of the subgraph induced by the union of the target vertices. Both checks merely require running a union-find algorithm [12].

Comparisons. To evaluate `Alg-EMD-LP`, we focus on the total cost and the associated distance, as specified by Eq. (2), and on the fraction of vertices and edges satisfying the connectivity constraints, as defined above, respectively denoted $r_V^{\text{c.c.}}$ and $r_E^{\text{c.c.}}$. To evaluate `Alg-EMD-CCC-G`, we resort to the total flow, the total cost and their ratio, as specified by Eq. (7). For `Alg-EMD-CCC-G`, we also assess the symmetry of costs and flows. For costs, given two graphs A and B, we compute the following ratio:

$$r_{\text{Cost}}^{\text{sym.}} = \frac{\min(C^{\text{Alg-EMD-CCC-G}}(A,B), C^{\text{Alg-EMD-CCC-G}}(B,A))}{\max(C^{\text{Alg-EMD-CCC-G}}(A,B), C^{\text{Alg-EMD-CCC-G}}(B,A))}. \tag{8}$$

Given a collection of pairs of graphs, we also compute the min and max of the ratio of Eq. (8) over all pairs. We proceed *mutatis mutandis* for the total flow. Finally, we also monitor running times, as well as the size of the transport plans computed(remark of Section 1).

5.2 Potential Energy Landscapes of Simplified Protein Models

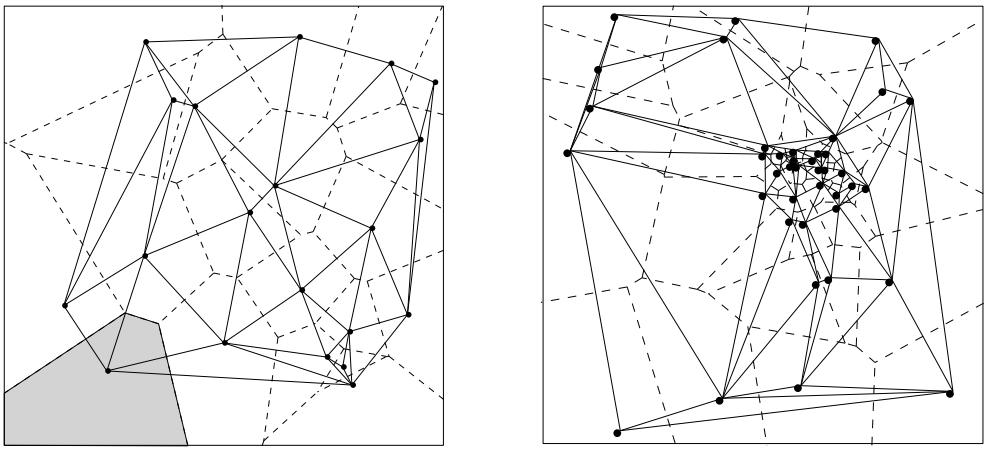
5.2.1 Specifications

Definition. A BLN model is a simplified protein model whose amino-acids have been replaced by three pseudo amino-acids modeled as beads, each with specific properties (hydrophobic (B), hydrophylic (L) and neutral (N)) [14]. A BLN model with k beads therefore consists of a linear chain containing $k-1$ covalent bonds linking every pair of consecutive beads, resulting in $3 \times k$ Cartesian coordinates. We use $k = 69$, so that the conformational space of the system has dimension $d = 207$. BLN models are known to fold into a structure with a hydrophobic core

favoring close interactions between hydrophobic beads, thus mimicking real proteins (Fig. 5). As a distance between conformations, we use the least Root Mean Square Deviation (lRMSD).

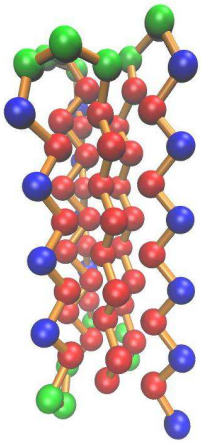
Generation. The *potential energy landscape* (PEL) of the BLN system is obtained by associating a potential energy to each conformation, and we use the expression provided in the supplemental Section 7.1. Exploring a PEL in general is a challenging endeavor [11], due to its high dimensionality and to its *ruggedness*, namely the presence of many shallow local minima. That of BLN69 has been thoroughly explored [9], with more than 450,000 local minima reported overall. Ten of them are of special interest since they are low in energy and interconnected by barriers of *small height* [9]. Our analysis focuses on this *top ten* in the sequel. We use the `T-RRT` algorithm [7] to generate samples at random in the vicinity of each minimum in the top ten. Algorithm `T-RRT` builds a random tree by adding nodes at its periphery, so as to favor the exploration of yet unexplored regions and discover low hanging local minima. Practically, we generate $N = 10^4$ samples for each minimum. We further process each such sampling, as explained below, to produce one graph to be used in the comparisons.

Figure 4 Voronoi landscapes. The graph of a Voronoi landscape is defined by the one-skeleton of the 2D Delaunay triangulation of points. The volume of supply or demand is the normalized area of the Voronoi tile (gray region on the left side). (**Left**) A landscape with $S = 20$ random points. (**Right**) A nested landscape obtained by merging samples within two nested squares.



with $S + s = 20 + 20$ points.

Figure 5 Example conformation of the BLN69 model The three types of beads are represented as follows: hydrophobic (B) in red, hydrophylic (L) in blue, and neutral (N) in green. Note the formation of a hydrophobic core clubbing the hydrophobic particles.



A given sampling may have visited several local minima in the vicinity of the local minimum it was generated from. To identify these, we perform a gradient descent of the potential energy at each sample $p_i, i = 1, \ldots, N$, an operation known as *quenching*. Quenching assigns a sample p_i to the local minimum $q(p_i)$ found at the bottom of the basin of the PEL sample p_i belongs to. To define a graph connecting these local minima $q(p_i)$ via index one saddles, we resort to a modified version of the `Tomato` algorithm [3]. Recall that the `Tomato` algorithm maps each sample to its local minimum by a discrete flow operator defined from a nearest neighbor graph (NNG) defined on all samples, and identifies transitions between basins as pairs of neighboring samples flowing to distinct local minima. We replace this discrete flow operator by the information provided by $q(p_i)$. (As argued in [2], in high dimensional spaces, information encoded in NNG falls short from providing accurate information for local minima in particular, and for critical points in general.) Furthermore, upon obtaining this graph, we simplify it using topological persistence, to retain the 50 most significant local minima.

Finally, we assign a mass to each node of this simplified graph, defined as the fraction of samples (out of $N = 10^4$) which get quenched to this local minimum. The process leaves us with one graph, called a *transition graph* in the sequel, for each local minimum in the top ten. We refer to this data set as TRRT-top10, and to a particular graph as TRRT-top10-i, $i = 1, \ldots, 10$.

5.2.2 Results

Our ten transition graphs yield 45 pairs, whence 45 instances for `Alg-EMD-LP` (which is symmetric), and 90 instances for `Alg-EMD-CCC-G` (which is not symmetric).

Algorithm `Alg-EMD-LP` and constraint satisfaction. Since algorithm `Alg-EMD-LP` is oblivious to critical point connectivity, we compute the fraction $r_V^{\text{c.c.}}$ and $r_E^{\text{c.c.}}$ of vertices and edges of the input graph inducing through the flow a connected subgraph of the demand graph. Out of the 45 instances of the dataset TRRT-top10, the min and max values for vertices and edges are (0.41, 1), and (0.24, 1), respectively. That is, transport plans obtained from solutions of the linear program do disrupt connectivity constraints.

Algorithm `Alg-EMD-CCC-G` and demand satisfaction. The connectivity constraints may prevent `Alg-EMD-CCC-G` to fully satisfy the demand. For each instance, we therefore monitor the total flow $F^{\text{Alg-EMD-CCC-G}}$ provided by the transport plan, the ideal value being one. On the 90 instances, a worst-case of 0.99 is observed, showing that the connectivity constraints are lenient on these instances. Further inspection shows that such performances owe to the distribution of weights in the basins. Indeed, for each transition graph TRRT-top10-i, it turns out that the local minimum from which the exploration was started takes most of the mass. Therefore, in comparing two such graphs, a transportation plan essentially reduces to moving the mass in-between the two prominent local minima.

Transport costs. To assess transport costs, we compute the linear correlation between three sets of 45 values, namely the transport costs of `Alg-EMD-LP` of the 45 instances, and those of `Alg-EMD-CCC-G` on the 45×2 pairs (recall that `Alg-EMD-CCC-G` is not symmetric). The three coefficients obtained are equal to 0.99, a property again owing to the structure of the basins, as just discussed.

These examples showthat `Alg-EMD-CCC-G` does find elementary transport plans when these exist. In the sequel, we therefore challenge it with cases involving a more uniform distribution of masses.

5.3 Voronoi Landscapes

The difficulty of transport problems associated with a biophysical system depends on the topography of its landscape, which impacts the number of local minima and the volume of their basins. We therefore challenge our algorithms with synthetic landscapes characterized by two features: a user-defined number of nodes, and a more uniform distribution of masses. These landscapes are based on 2D Voronoi diagrams.

5.3.1 Specifications

Definition. Consider a 2D point cloud $P = \{p_i\}_{i=1,\dots,S}$ in a square $Q = [0, a] \times [0, a]$. Define the following distance function $d_P(p) = \min_i d(p, p_i)$, with $d(p, p_i)$ the Euclidean distance between $p \in Q$ and $p_i \in P$.

A Voronoi landscape mimics a potential energy landscapes (PEL) in the following respect (Fig. 4):

- The conformational space of the molecule is replaced by the Euclidean space Q.
- The local minima of the PEL are replaced by the local minima of the distance function d_P, namely the samples in P.
- The volume of a basin of the PEL is replaced by the surface area of the Voronoi tile of a sample from P, restricted to Q. That is, these areas define the volume of supply or demand.
- A saddle between two basins of the PEL is replaced by a Delaunay edge connecting two samples from P.
- To compute the unit transport cost between two local minima on the PEL, the distance measure used between two molecular conformations is replaced by the Euclidean distance in $\mathbb{R}^2$.

Generation. We compute instances of Voronoi landscapes using the Computational Geometry Algorithms Library [1]. To go beyond the case where the samples in P are generated uniformly at random, we generate *nested landscapes*. More precisely, a nested landscape with parameters (S, s) results from merging S samples taken uniformly at random within Q with s samples located in a square of side $a/10$ randomly placed within Q. Equipped with these notations, a *random landscape generator* is specified by the pair (S, s), with S (resp. s) the number of samples in the big (resp. small) square. Practically, we use $S = 5s$, and use values of S in the range $[50, 1000]$. We observe in passing that 1000 is a comfortable upper bound for the number of meta-stable states of many bio-physical systems, including proteins [5, 14].

To remove random bias, we consider several instances of problems associated with a random landscape generator with parameters (S, s). That is, for each pair (S, s), we generate $I(= 10)$ pairs of graphs with parameters S and s.

5.3.2 Results

Algorithm `Alg-EMD-CCC-G` and running times. The maximum running times on our instances range from $t_s = .3s$ for instances of size 60 nodes ($S = 50, s = 10$), to $t_s = 1530s$ for graphs of 1200 nodes ($S = 1000, s = 200$). Thus, despite its complexity, algorithm `Alg-EMD-CCC-G` remains effective for graphs of intermediate size.

Algorithm `Alg-EMD-LP` and constraint satisfaction. On our instances, the fraction $r_V^{\text{c.c.}}$ is at least 0.89, while $r_E^{\text{c.c.}}$ is larger than 0.75 (Table 1). In this respect, these instances are easier than those defined by the BLN models.

Algorithm `Alg-EMD-CCC-G` and demand satisfaction. It is observed that the total flow $F^{\text{Alg-EMD-CCC-G}}$ is always larger than 0.62 (Table 2). Moreover, the symmetry ratio of Eq. (8), which is always larger than 0.79 (Table 3), shows that instances tend to be equally hard in both directions.

Transport costs. We observe that transport costs decrease with the number of nodes, which is expected since when the number of points increases, the distance between the endpoints of edges of the bipartite graph along which flow circulates decreases (Table 2).

In comparing the costs of `Alg-EMD-CCC-G` against those of `Alg-EMD-LP` (Table 2 vs Table 1), one sees that transport plans provided by `Alg-EMD-CCC-G` tend to be cheaper, which is expected since the demand is not fully satisfied. This trend remains upon re-normalizing the costs of `Alg-EMD-CCC-G` by the total flow, which is also in line with the greedy selection of edges. Indeed, choosing cheap edges leaves a pool or more expensive edges, which may not be used at a latter stage due to connectivity constraints. Note that `Alg-EMD-LP` does not stand the chance to skip these expensive edges, since it is forced to saturate the demand.

Algorithm `Alg-EMD-CCC-G` and size of solutions. We also computed the relative size RS of solutions, namely ratio between the number of edges carrying flow and the number of vertices of the bipartite graph (i.e. $2(S+s)$). With a value in the range $[0.55, 0.75]$, this statistic shows that solutions have a linear (in the number of vertices of the input graphs) number of edges. This observation legitimates the choice made in Section 2.3, which consists of relaxing the constraint on the number of edges M.

6 Conclusion and Future Works

In this paper, we developed several hardness results for the mass transportation problem with connectivity constraints, together with two polynomial time algorithms. Despite the problem

hardness, our experiments show that these algorithms are effective for graphs involving up to hundreds of nodes.

Despite these contributions, several research avenues remain open. The first one naturally concerns the approximation factors associated with our algorithms. The second one relates to the possibility of obtaining solutions respecting the connectivity constraint, by fixing the optimal transport plan computed by the linear program. Another one concerns large graphs (involving thousands of nodes), whose vertices can be embedded in a Euclidean space. In this case, the locality information provided by spatial partitions could be exploited to drive the flow. Yet another one would consist of generalizing our problems and algorithms so as to handle the more general case where the mass associated to a supply or demand node is not concentrated into a single point.

On the application side, two topics are of foremost importance. The first one is the comparison of energy landscapes of molecular systems, in two guises: landscapes coupled to atomic models, versus landscapes coupled to coarse grain models. Thelatter should indeed match the former, a topic which has barely been addressed while designing coarse grain models. The second one is the coupling of our algorithm to landscape exploration algorithms, so as to boost the exploration breadth.

Acknowledgments. This research has been partially supported by the European Research Council under Advanced Grant 339025 GUDHI (Algorithmic Foundations of Geometric Understanding in Higher Dimensions).

References

[1] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[2] F. Cazals, T. Dreyfus, D. Mazauric, A. Roth, and C. H. Robert. Conformational ensembles and sampled energy landscapes: Analysis and comparison. 2014.

[3] F. Chazal, L. Guibas, S. Oudot, and P. Skraba. Persistence-based clustering in riemannian manifolds. In *ACM SoCG*, pages 97–106. ACM, 2011.

[4] G. B. Dantzig. Application of the simplex method to a transportation problem. *Activity Analysis of Production and Allocation*, pages 359–373.

[5] A. Fersht. *Structure and Mechanism in Protein Science: A Guide to Enzyme Catalysis and Protein Folding*. Freeman, 1999.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[7] L. Jaillet, F.J. Corcho, J-J. Pérez, and J. Cortés. Randomized tree construction algorithm to explore energy landscapes. *Journal of computational chemistry*, 32(16):3464–3474, 2011.

[8] G. Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, pages 666–704, 1781.

[9] M. T. Oakley, D. J. Wales, and R. L Johnston. Energy landscape and global optimization for a frustrated model protein. *The Journal of Physical Chemistry B*, 115(39):11525–11529, 2011.

[10] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[11] C. Schön and M. Jansen. Prediction, determination and validation of phase diagrams via the global study of energy landscapes. *International Journal of Materials Research*, 100(2):135, 2009.

[12] R. E. Tarjan. *Data Structures and Network Algorithms*, volume 44 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1983.

[13] C. Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

[14] D.J. Wales. *Energy Landscapes*. Cambridge University Press, 2003.

7 Supplemental: Experiments

7.1 BLN Models

The potential energy V_{BLN} consists of two types of terms (Eq. 9 and [14]): first, terms describing the interaction between beads sharing a covalent bond (the first, second and third terms below are respectively bond lengths, valence angles, and dihedral angles); second, a term describing non covalent interactions between beads (the fourth term below is the so-called Lennard-Jones potential):

$$\begin{aligned} V_{BLN} = \frac{1}{2} \cdot K_r \cdot \sum_{i=1}^{N-1} (R_{i,i+1} - R_e)^2 + \frac{1}{2} K_0 \sum_{i=1}^{N-2} (\theta_i - \theta_e)^2 \\ + \epsilon \cdot \sum_{i=1}^{N-3} [A_i(1 + \cos\phi_i) + B_i(1 + 3\cos\phi_i)] \\ + 4\epsilon \cdot \sum_{i=1}^{N-2} \sum_{j=i+2}^{N} C_{ij}[(\frac{\sigma}{R_{i,j}})^{12} - D_{ij}(\frac{\sigma}{R_{i,j}})^6]. \end{aligned} \tag{9}$$

7.2 Statistics

Table 1 Algorithm `Alg-EMD-LP`: statistics for runs on the 45 instances defined by pairs of Voronoi landscapes. Columns 2-3, $r_V^{\text{c.c.}}$: fraction of vertices respecting the connectivity constraints; Columns 4-5, $r_E^{\text{c.c.}}$: fraction of edges respecting the connectivity constraints; Columns 6-7, C^{EMD}: total cost.

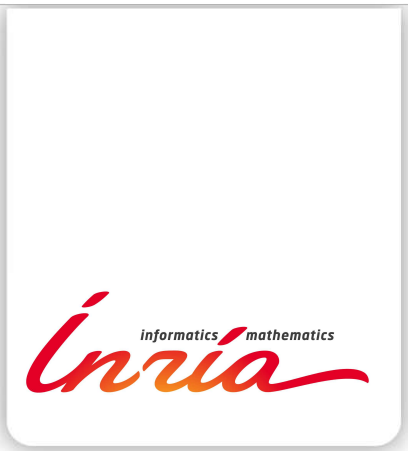
Batch spec.	$r_V^{\text{c.c.}}$		$r_E^{\text{c.c.}}$		C^{EMD}	
	min	max	min	max	min	max
VL-s10-S50	0.92	0.98	0.84	0.94	0.82	0.94
VL-s20-S100	0.93	0.98	0.85	0.93	0.59	0.67
VL-s30-S150	0.96	0.98	0.88	0.91	0.49	0.53
VL-s40-S200	0.95	0.97	0.87	0.92	0.44	0.47
VL-s100-S500	0.93	0.96	0.83	0.85	0.28	0.31
VL-s200-S1000	0.89	0.91	0.75	0.78	0.21	0.24

Table 2 Algorithm `Alg-EMD-CCC-G`: statistics for runs on the 90 instances defined by pairs of Voronoi landscapes. The columns read as follows: Column 2-3, t_s: running time in seconds; Columns 4-5, RS: relative size of solution (number of edges carrying flow / number of nodes of the bipartite graph); Columns 6-7, $F^{\text{Alg-EMD-CCC-G}}$: total flow (ideal value is one); Columns 8-9, $C^{\text{Alg-EMD-CCC-G}}$: transport cost; Columns 8-9, $d_{\text{Alg-EMD-CCC-G}} = C^{\text{Alg-EMD-CCC-G}}/F^{\text{Alg-EMD-CCC-G}}$

Batch spec.	t_s		RS		$F^{\text{Alg-EMD-CCC-G}}$		$C^{\text{Alg-EMD-CCC-G}}$		$d_{\text{Alg-EMD-CCC-G}}$	
	min	max	min	max	min	max	min	max	min	max
VL-s10-S50	0.23	0.29	0.57	0.75	0.62	0.83	0.48	0.77	0.75	0.92
VL-s20-S100	1.64	1.98	0.59	0.75	0.63	0.80	0.37	0.55	0.53	0.71
VL-s30-S150	5.13	5.97	0.59	0.72	0.68	0.77	0.29	0.38	0.42	0.51
VL-s40-S200	12.01	13.65	0.60	0.69	0.66	0.77	0.27	0.33	0.39	0.44
VL-s100-S500	185.25	207.17	0.59	0.62	0.67	0.72	0.17	0.20	0.25	0.28
VL-s200-S1000	1434.73	1538.4	0.55	0.59	0.67	0.71	0.12	0.14	0.18	0.20

Table 3 Algorithm `Alg-EMD-CCC-G`: symmetry assessment on the 45 instances defined by pairs of Voronoi landscapes. Columns 2-3 report the min and max values of Eq. (8), while Columns 4-5 report the equivalent statistic for the ratios of flows.

Batch spec.	Flow ratio		Cost ratio	
	min	max	min	max
VL-s10-S50	0.79	0.99	0.68	0.99
VL-s20-S100	0.85	0.97	0.78	0.96
VL-s30-S150	0.95	1.00	0.87	0.97
VL-s40-S200	0.85	0.99	0.81	0.99
VL-s100-S500	0.96	1.00	0.94	1.00
VL-s200-S1000	0.96	0.99	0.93	0.99



RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399