

# Transport optimal et distances de Wasserstein

## Table des matières

<b>I Introduction</b>	<b>2</b>
<b>II Formulation du problème</b>	<b>2</b>
1 Modélisation du problème par Monge . . . . .	2
a Problème dans le cas discret . . . . .	2
b Transport optimal pour un coût optimal . . . . .	3
2 Assouplissement des contraintes, problème de Monge-Kantorovich . . . . .	5
a Une approche probabiliste du transport optimal . . . . .	5
b Noyaux et couplages . . . . .	6
<b>III Dualité de Kantorovich et distance de Wasserstein</b>	<b>12</b>
1 Existence d'un couplage optimal pour une fonction de coût sci. . . . .	12
2 Dualité de Kantorovich . . . . .	14
3 Distance de Wasserstein . . . . .	15
<b>IV Algorithme Hongrois</b>	<b>17</b>
1 Problème . . . . .	17
a Problème discret . . . . .	17
b Linear Sum Assignment Problem (LSAP) . . . . .	17
2 Présentation de l'algorithme Hongrois . . . . .	18
a Vue d'ensemble du fonctionnement de l'algorithme. . . . .	18
b Exemple . . . . .	18
3 Application à un problème dans le plan . . . . .	20
a Problème . . . . .	20
b Méthode de Résolution : Approche par graphe . . . . .	22
c Méthode de Résolution : Autre version de l'algorithme Hongrois . . . . .	24
<b>V Applications</b>	<b>27</b>
1 Généralisation du problème de production et de stockage dans le plan . . . . .	27
2 Transfert de couleurs entre deux images . . . . .	30

## I Introduction

Le transport optimal est un problème mathématique qui consiste à trouver le coût minimal pour transporter une mesure de probabilité source vers une autre mesure de probabilité cible. Cette notion a été introduite pour la première fois par le mathématicien français Gaspard Monge en 1781, dans son ouvrage "Mémoire sur la théorie des déblais et des remblais". Celui-ci cherchait à déplacer de manière optimale des tas de sable vers des fossés pour des fins militaires.

Comme nous pourrons le voir, la modélisation du problème de transport optimal par Monge est particulièrement contraignante. Elle ne trouve pas toujours de réponses et l'unicité de celle-ci n'est pas garantie. L'approche du problème fût assouplie par les travaux de Leonid Kantorovich et Tjalling Koopmans en 1975. Ils ont tous deux reçu le prix Nobel d'économie pour leur travail sur cette question.

Le transport optimal est un sujet de recherche utilisé dans des domaines comme l'économie, l'informatique, la physique, la biologie et bien d'autres. Il sert à résoudre des problèmes de planification de production dans l'industrie, il aide en biologie pour le suivi de la circulation sanguine dans le corps humain. Par exemple, il peut être utilisé dans la grande distribution pour faciliter les transports de marchandises. Connaissant les stocks dans les entrepôts et les demandes des clients, il est possible d'optimiser l'envoi de marchandises en utilisant le transport optimal.

Dans ce sujet de recherche encadré par *Fabien Panloup*, nous allons explorer les origines du transport optimal ainsi que sa modélisation mathématique à travers les époques. Nous introduirons le théorème de dualité de Kantorovich et la distance de Wasserstein qui découlent de ce problème mathématique. Cette distance nous permettra de comparer des densités de probabilités en cherchant leurs similarités. Nous proposerons finalement une approche algorithmique avec la résolution de problèmes de transport optimal sur des exemples.

### Notations mathématiques :

On note

- Soit  $E$  un ensemble et  $x_0 \in E$ ,  $\delta_{\{x_0\}}$  est la mesure de Dirac en  $x_0$ .
- Pour  $E$  un ensemble fini,  $\mathcal{P}(E)$  est l'ensemble des parties de  $E$ .
- La notation  $\|\cdot\|$  est associée à la norme euclidienne s'il n'y a pas de précision donnée.
- Soit  $I$  un intervalle réel, on note  $\mathcal{B}(I)$  l'ensemble des Boréliens de  $I$ .
- Un espace topologique est séparable s'il contient un sous-ensemble dense et au plus dénombrable.
- Un espace métrique  $(E, d)$  est polonais s'il est séparable et complet.

## II Formulation du problème

### 1 Modélisation du problème par Monge

#### a Problème dans le cas discret

Le mathématicien Gaspard Monge posait la question suivante : quelle est la manière la plus économique de remplir un trou avec un tas de sable ?

Soit  $(n, m) \in \mathbb{N}^2$ . Pour essayer de répondre à cette question, considérons  $E := \{x_1, \dots, x_n\}$  un espace discret muni d'une mesure de probabilité source  $\mu = \sum_{i=1}^n \alpha_i \delta_{\{x_i\}}$ . L'espace de départ est associé aux tas de sables avec

les densités respectives de sable pour chacun des tas. Ainsi on a  $\sum_{i=1}^n \alpha_i = 1$  avec  $\alpha_i \geq 0$  pour tout  $i \in \llbracket 1, n \rrbracket$ . De même, considérons  $F := \{y_1, \dots, y_m\}$  un espace discret muni d'une mesure de probabilité cible  $\nu = \sum_{i=1}^m \beta_i \delta_{\{y_i\}}$ . L'espace de départ est associé aux trous avec les densités respectives de vide pour chacun. Ainsi on a  $\sum_{i=1}^m \beta_i = 1$  avec  $\beta_i \geq 0$  pour tout  $i \in \llbracket 1, m \rrbracket$ .

La modélisation du problème de Monge revient à construire une application mesurable  $T : E \longrightarrow F$  transportant  $\mu$  sur  $\nu$  à coût minimal (*la notion de coût sera introduite par la suite*). La contrainte est la suivante :

$$\forall C \in \mathcal{P}(F), \quad \mu(\{x : T(x) \in C\}) = \nu(C)$$

Nous introduisons la notation  $T_{\#}\mu$  pour désigner la *mesure image* de  $\mu$  par  $T$ . Ainsi

$$\forall C \in \mathcal{P}(F), \quad T_{\#}\mu(C) = \mu(T^{-1}(C)) = \mu(x \in E : T(x) \in C)$$

**Exemple 1.** Plaçons-nous dans le plan. Soit  $E := \{(0,0), (1,1)\}$  muni de la mesure  $\mu = \frac{1}{2}\delta_{\{(0,0)\}} + \frac{1}{2}\delta_{\{(1,1)\}}$  et  $F := \{(0,1), (1,0)\}$  muni de la mesure  $\nu = \frac{1}{2}\delta_{\{(0,1)\}} + \frac{1}{2}\delta_{\{(1,0)\}}$ . On trouve deux applications de transport  $T_1, T_2$  sont possibles :

$$T_1((0,0)) = (0,1); \quad T_1((1,1)) = (1,0)$$

$$T_2((0,0)) = (1,0); \quad T_2((1,1)) = (0,1)$$

## b Transport optimal pour un coût optimal

Monge avait introduit le coût d'un transport dans le cadre de son problème, les espaces de départ et d'arrivée étant les mêmes ( $E = F = \mathbb{R}^2$ ). Il avait défini le coût pour transporter une unité de masse en  $x$  vers  $y = T(x)$  comme la distance euclidienne entre  $x$  et  $y$ . Pour une fonction de coût donnée, les applications effectuant le transport n'ont pas nécessairement le même coût. Ainsi, nous chercherons à minimiser le coût de transport en optimisant sur l'ensemble des fonctions de transport sous la contrainte  $\nu = T_{\#}\mu$ .

**Exemple 2.** Le coût engendré par une application de transport donnée  $T$  dans le problème de Monge vaut

$$C_1(T) = \int_{\mathbb{R}^2} \|x - T(x)\| \mu(dx).$$

Pour une fonction de coût  $C$  donnée, on cherche à déterminer  $C(\mu, \nu) = \inf\{C(T); T : A \rightarrow B, T_{\#}\mu = \nu\}$ . Si cela est possible, on note  $T^*$  comme une application optimale réalisant cet infimum.

$$T^* = \operatorname{arginf}\{C(T); T : E \rightarrow F, T_{\#}\mu = \nu\}$$

**Remarque 1.** *Sous les contraintes du problème de Monge, certaines questions peuvent être posées :*

- *Pour deux mesures de probabilités  $\mu$  et  $\nu$ , existe-t-il toujours une application  $T^*$  qui réalise l'infimum précédent ?*
- *S'il existe une application  $T^*$  qui réalise l'infimum, est-elle unique ?*

**Propriété 1.** Soit  $\mathbb{R}^2$  muni de deux mesures discrètes  $\mu, \nu$ . Considérons le problème de transport optimal associé à la fonction de coût  $C_1$  qui est la distance euclidienne. Si  $T^*$  est une application de transport minimisant  $C(\mu, \nu)$ , pour  $x_1, x_2 \in \mathbb{R}^2$  et  $y_1, y_2 \in \mathbb{R}^2$  tels que  $T^*(x_1) = y_1$  et  $T^*(x_2) = y_2$  on a :

- ou bien tous les points sont alignés
- ou bien les segments  $[x_1, y_1]$  et  $[x_2, y_2]$  ne se coupent pas sauf si  $x_1 = x_2$  ou  $y_1 = y_2$ .

*Démonstration.*

Par l'absurde supposons que les points ne sont pas alignés et  $x_1 \neq x_2$  et  $y_1 \neq y_2$  mais que les segments se coupent en un point  $c$ . Posons  $T'$  une autre application de transport vérifiant la contrainte  $T'_{\#}\mu = \nu$  telle que  $T'(x_1) = y_2$ ,  $T'(x_2) = y_1$  et  $T' = T^*$  sur les autres points de  $E$ . La norme euclidienne vérifie l'inégalité triangulaire avec égalité si et seulement si les vecteurs sont colinéaires. On obtient :

$$\begin{aligned} \|x_1 - T'(x_1)\| + \|x_2 - T'(x_2)\| &= \|x_1 - y_2\| + \|x_2 - y_1\| \\ &< \|x_1 - c\| + \|y_2 - c\| + \|x_2 - c\| + \|y_1 - c\| \\ &= \|x_1 - c\| + \|c - y_1\| + \|x_2 - c\| + \|c - y_2\| \\ &= \|x_1 - y_1\| + \|x_2 - y_2\| \\ &= \|x_1 - T^*(x_1)\| + \|x_2 - T^*(x_2)\| \end{aligned}$$

En sommant sur  $E$  on obtient que  $C_1(T') < C_1(T^*)$  ce qui est absurde car cela contredit la minimalité de l'application de transport optimal  $T^*$ .  $\square$

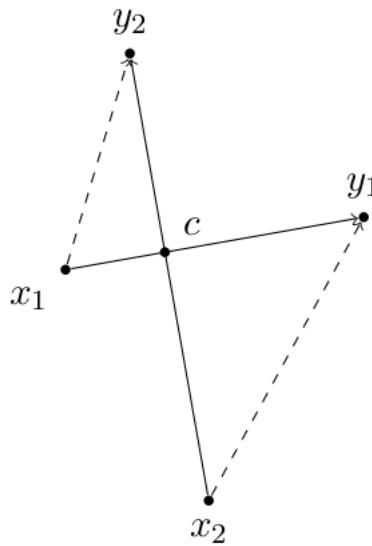


FIGURE 1 – Illustration du raisonnement précédent

**Exemple 3.** "Un transport optimal qui n'est pas unique"

Soit  $A$  le carré de diagonale  $(0,0) -- (1,1)$  et  $B$  le carré de diagonale  $(0,0) -- (-1,1)$ . On associe les mesures de probabilité  $\mu$  et  $\nu$  sur  $\mathbb{R}^2$  définies pour tout  $x \in \mathbb{R}^2$  par

$$d\mu(x) = \frac{\mathbb{1}_A(x)}{|A|} dx = \mathbb{1}_A(x) dx \quad \text{et} \quad d\nu(x) = \frac{\mathbb{1}_B(x)}{|B|} dx = \mathbb{1}_B(x) dx$$

Proposons deux applications distinctes de transport optimal  $T_1$  et  $T_2$  sous la contrainte du problème de Monge associées au coût de la distance euclidienne.

$$T_1((x, y)) = (-x, y) \quad \text{et} \quad T_2((x, y)) = (x - 1, y).$$

En effet on a :

- $C_1(T_1) = \int_{\mathbb{R}^2} \|(x, y) - T_1((x, y))\| d\mu(x) = \int_{[0,1]^2} \sqrt{(2x)^2 + (y - y)^2} dx dy = 2 \int_{[0,1]^2} |x| dx dy = 1$
- $C_1(T_2) = \int_{\mathbb{R}^2} \|(x, y) - T_2((x, y))\| d\mu(x) = \int_{[0,1]^2} \sqrt{(x - (x - 1))^2 + (y - y)^2} dx dy = \int_{[0,1]^2} 1 dx dy = 1$

Ainsi  $C_1(T_1) = C_1(T_2)$ , il n'y a pas toujours unicité dans le problème de transport optimal.

#### **Exemple 4. "Application de transport à sens unique"**

$A$  est le carré unité et  $B$  un carré infinitésimal en  $(0, 0)$ . On associe les mesures de probabilité  $\mu$  et  $\nu$  sur  $\mathbb{R}^2$  définies par

$$d\mu(x) = \frac{\mathbb{1}_A(x)}{|A|} dx = \mathbb{1}_A(x) dx \quad \text{et} \quad d\nu(x) = \delta_{\{(0,0)\}}(x).$$

Pour transporter la mesure  $\mu$  sur la mesure  $\nu$ , il n'existe qu'une seule application de transport  $T^*$  telle que pour tout  $x \in A$ ,  $T^*(x) = (0, 0)$ . En revanche, il est impossible de transporter la mesure  $\nu$  sur la mesure  $\mu$  sous les contraintes du problème de Monge. Toute la masse est concentrée en un point  $(0, 0)$ , pour toute application de transport  $T$  on remarque qu'il existe un unique  $x_0 \in A$  tel que  $T((0, 0)) = x_0$  (ceci par unicité de l'image d'une application). Or  $\mu(\{x_0\}) = 0 \neq \nu(T^{-1}(\{x_0\})) = \nu((0, 0)) = 1$ .

**Remarque 2.** Dans l'exemple précédent, pour le transport retour on souhaiterait pouvoir répartir la masse présente en  $(0, 0)$ , sans l'envoyer uniquement au point  $T((0, 0))$ . Pour pallier à ce problème, nous allons remplacer l'application de transport  $T$  par une notion mathématique plus générale.

## 2 Assouplissement des contraintes, problème de Monge-Kantorovich

Nous allons introduire une version relâchée du problème de transport optimal, cette approche est celle proposée par le mathématicien russe Leonid Kantorovich pour laquelle il a obtenu le prix Nobel d'économie en 1975.

### a Une approche probabiliste du transport optimal

#### Définition 1.

Soit  $(E, \mathcal{A})$  un ensemble mesurable et  $\mathcal{P}(E)$  l'ensemble des mesures de probabilité sur  $E$ . Un *noyau* sur  $E$  est une application  $p : E \rightarrow \mathcal{P}(E)$  telle que pour tout  $x \in E$  on note  $p_x(\cdot) := p(x)(\cdot)$  la mesure de probabilité associée et pour toute fonction mesurable bornée de  $E$  vers  $\mathbb{R}$ , l'application  $x \mapsto \int_E f(y)p_x(dy)$  est mesurable.

Ainsi pour  $\mu$  une mesure de probabilité source et  $\nu$  une mesure de probabilité cible sur un même espace  $(E, \mathcal{A})$ , nous souhaitons modéliser le problème de transport optimal dans un cadre non déterministe. Nous pourrions essayer de trouver un noyau  $p$  sur  $E$  sous la contrainte :

$$\forall A \in \mathcal{A}, \nu(A) = \int_E \int_A p_x(dy) \mu(dx)$$

Ainsi nous retrouvons l'idée de mesure pré-image vue à la section précédente mais avec la possibilité de déplacer la masse présente en  $x$  selon une certaine densité de probabilité  $p_x$ .

### Exemple 5.

On se place dans  $\mathbb{R}^2$  muni de la fonction de coût  $C_p(x, y) = |x - y|^p$ ,  $x, y \in \mathbb{R}^2$ ,  $p \in \mathbb{N}^*$  et on considère  $A = \{0\} \times [0, 1]$ ,  $B = \{-1\} \times [0, 1]$  et  $C = \{1\} \times [0, 1]$ . Pour  $m$  une mesure uniforme sur  $[0, 1]$ , on considère ensuite les mesures de probabilité.

$$\mu = \delta_0 \otimes m \quad \text{et} \quad \nu = \frac{1}{2}\delta_{-1} \otimes m + \frac{1}{2}\delta_1 \otimes m.$$

$\mu$  est la loi de  $(0, X)$  où  $X$  suit la loi uniforme sur  $[0, 1]$ , et  $\nu$  est la loi de  $(\varepsilon, X)$  où  $\varepsilon$  est une variable aléatoire indépendante de  $X$ , de loi de Rademacher  $\mathbb{P}[\varepsilon = 1] = \mathbb{P}[\varepsilon = -1] = \frac{1}{2}$ .

- on tire  $(0, X)$  suivant la loi  $\mu$ .
- conditionnellement au tirage de  $(0, X) = (0, x_2) = x$ , on tire  $(\varepsilon, X)$  suivant le noyau  $p_x$  défini par :

$$p_x(\cdot) = \frac{1}{2}\delta_{(-1, x_2)} \otimes \delta_{\{-1; 1\} \times \{x_2\}}(\cdot) + \frac{1}{2}\delta_{(1, x_2)} \otimes \delta_{\{-1; 1\} \times \{x_2\}}(\cdot) = \frac{1}{2}\delta_{(-1, x_2)}(\cdot) + \frac{1}{2}\delta_{(1, x_2)}(\cdot)$$

Soit  $a \in \{-1; 1\}$  et  $A \in \mathcal{B}([0, 1])$  on a  $\nu(\{a\} \times A) = \frac{1}{2} \times |A|$  où  $|A|$  est la mesure de Lebesgue pour l'ensemble  $A$ . Or on remarque que

$$\begin{aligned} \iint_{\mathbb{R}^2 \times \mathbb{R}^2} \mathbb{1}_{\{a\} \times A}(y) p_x(dy) \mu(dx) &= \int_{[0, 1]} \left( \int_{\mathbb{R}^2} \mathbb{1}_{\{a\} \times A}(y) \frac{1}{2}\delta_{(-1, x_2)}(dy) + \frac{1}{2}\delta_{(1, x_2)}(dy) \right) dx_2 \\ &= \int_{[0, 1]} \left( \int_{\mathbb{R}^2} \mathbb{1}_{\{a\} \times A}(y) \frac{1}{2}\delta_{(a, x_2)}(dy) \right) dx_2 \\ &= \int_{[0, 1]} \left( \int_{\mathbb{R}^2} \mathbb{1}_{\{a\} \times A}(y) \frac{1}{2}\delta_{(a, x_2)}(dy) \right) dx_2 \\ &= \int_A \frac{1}{2} dx_2 = \frac{1}{2} \times |A| = \nu(\{a\} \times A) \end{aligned}$$

Nous observons que le noyau  $p_x$  défini précédemment respecte la contrainte de transport de la mesure  $\mu$  sur la mesure  $\nu$ . Étudions le coût de ce transport :

$$\begin{aligned} \iint_{\mathbb{R}^2 \times \mathbb{R}^2} |x - y|^p p_x(dy) \mu(dx) &= \int_{[0, 1]} \left( \int_{\mathbb{R}^2} |(0, x_2) - y|^p \frac{1}{2}\delta_{(-1, x_2)}(dy) + \frac{1}{2}\delta_{(1, x_2)}(dy) \right) dx_2 \\ &= \int_{[0, 1]} \left( \frac{1}{2} \times |(0, x_2) - (-1, x_2)|^p + \frac{1}{2} \times |(0, x_2) - (1, x_2)|^p \right) dx_2 \\ &= \int_{[0, 1]} \left( \frac{1}{2} + \frac{1}{2} \right) dx_2 = 1. \end{aligned}$$

## b Noyaux et couplages

### Définition 2.

Soit  $(E, \mathcal{A})$  un ensemble mesurable et  $\mu, \nu \in \mathcal{P}(E)$ . Un *couplage* entre  $\mu$  et  $\nu$  est une probabilité  $\pi$  sur  $E \times E$  vérifiant pour tout  $A, B \in \mathcal{A}$

$$\mu(A) = \iint_E \mathbb{1}_A(x) \pi(dxdy) \quad \text{et} \quad \nu(B) = \iint_E \mathbb{1}_B(y) \pi(dxdy)$$

On note  $\Pi(\mu, \nu)$  l'ensemble de tous les couplages entre  $\mu$  et  $\nu$ .

**Remarque 3.** Un couplage entre  $\mu$  et  $\nu$  est une probabilité  $\pi$  telle que la première marginale est  $\mu$  et la seconde est  $\nu$ .

Faisons le lien entre un noyau muni d'une mesure source et un couplage pour pouvoir utiliser la notion de couplage dans le problème de transport optimal.

**Propriété 2.** Soit  $(E, \mathcal{A})$  un ensemble mesurable.

1. Pour toute mesure  $\mu \in \mathcal{P}(E)$  et tout noyau  $p$  sur  $E$  on peut associer une mesure  $\pi$  sur  $E \times E$  qui couple la mesure  $\mu$  et la mesure  $\nu$  défini par :

$$\forall B \in \mathcal{A}, \quad \nu(B) = \int_E \int_B p_x(dy) \mu(dx)$$

2. Soit  $\pi$  une mesure de couplage entre deux mesures de probabilité  $\mu$  et  $\nu$  sur  $E$ , on peut associer  $\pi$  à un noyau  $p$  défini presque partout sur  $E$  et à la mesure  $\mu$ .

Démonstration.

1. Soit  $A, B \in \mathcal{A}$  on note  $\pi(A \times B) = \int_A \int_B p_x(dy) \mu(dx)$ .

On a  $\pi$  qui est une mesure de probabilité sur  $E \times E$ , montrons qu'elle couple  $\mu$  avec  $\nu$ .

On obtient les mesures marginales de la manière suivante :

$$\pi(E \times B) = \int_E \int_B p_x(dy) \mu(dx) = \nu(B)$$

$$\pi(A \times E) = \int_A \int_E p_x(dy) \mu(dx) = \int_A \mu(dx) \int_E p_x(dy) = \int_A \mu(dx) \times 1 = \mu(A)$$

2. Soit  $\pi$  une mesure de couplage entre deux mesures de probabilité  $\mu$  et  $\nu$  sur  $E$  et  $F$  (ici  $F = E$ ). Notons  $X, Y$  deux variables aléatoires définies sur un même espace probabilisé  $(\Omega, \mathcal{B}, \mathbb{P})$  à valeurs dans  $E$  de loi respective  $\mu$  et  $\nu$ . Soit  $\Phi : E \times E \rightarrow \mathbb{R}$ .

$$\begin{aligned} \int_{E \times F} \Phi(x, y) d\pi(x, y) &= \mathbb{E}(\Phi(X, Y)) \\ &= \mathbb{E}(\mathbb{E}(\Phi(X, Y)|X)) \\ &\stackrel{(*)}{=} \int_E \mathbb{E}(\Phi(X, Y)|X = x) d\mu(x) \\ &= \int_E \left( \int_F \Phi(x, y) dp_x(y) \right) d\mu(x) \\ &= \int_{E \times F} \Phi(x, y) dp_x(y) d\mu(x) \end{aligned}$$

(\*) Nous remarquons que pour tout  $x \in E$ , l'objet  $\mathbb{E}(\Phi(X, Y)|X = x)$  n'est pas toujours défini correctement. Cet objet est défini presque partout sur  $E$  au sens de la mesure  $\mu$  par

$$E(\Phi(X, Y)|X = x) = \lim_{\varepsilon \rightarrow 0} E(\Phi(X, Y)|X \in [x - \varepsilon, x + \varepsilon])$$

□

**Exemple 6.** La mesure  $\pi$  sur  $E \times E$  peut être associée au couple de variables aléatoires  $(X, Y)$  définies sur un même espace de probabilité  $(\Omega, \mathcal{B}, \mathbb{P})$  telles que  $X$  est distribuée selon  $\mu$  et  $Y$  selon  $\nu$ . Donnons sous certaines conditions la forme du noyau  $p_x$  :

— Si  $X$  est à valeurs discrètes, alors pour tout  $x \in X(\Omega)$  tel que  $\mathbb{P}(\{X = x\}) > 0$ .

$$p_x(A) = \mathbb{P}(\{Y \in A\} | \{X = x\}) = \frac{\mathbb{P}(\{Y \in A\} \cap \{X = x\})}{\mathbb{P}(\{X = x\})}$$

— Dans le cas où  $\pi$  est une probabilité sur  $\mathbb{R}^n \times \mathbb{R}^n$  admettant une densité par rapport à Lebesgue :

$$\pi(dxdy) = h(x, y)dxdy.$$

$$\forall A \in \mathcal{B}(\mathbb{R}^n), \quad p_x(A) = \int_A \frac{h(x, y)}{\int_E h(x, z)dz} dy$$

**Remarque 4.** Par ce qui précède, nous observons que la contrainte posée sur le transport de la mesure  $\mu$  vers la mesure  $\nu$  peut être associée de manière équivalente à l'aide d'un noyau ou bien d'une mesure de couplage.

**Définition 3.** Soient  $c$  une fonction de coût mesurable sur  $E \times E$  et  $\mu, \nu \in \mathcal{P}(E)$ ; le coût de transport associé à un couplage  $\pi \in \Pi(\mu, \nu)$  est défini par

$$I_c(\pi) = \iint_{E \times E} c(x, y)\pi(dxdy)$$

Le coût de transport optimal entre  $\mu$  et  $\nu$  est donné par :  $\mathcal{T}_c(\mu, \nu) = \inf \{I_c(\pi) : \pi \in \Pi(\mu, \nu)\}$

**Remarque 5.** La modélisation du problème de Monge-Kantorovich s'apparente à un problème d'optimisation sur l'espace des couplages possibles entre deux mesures de probabilité. Si le minimum est réalisé,  $I_c(\pi^*) = T_c(\mu, \nu)$ , on dit que  $\pi^*$  est un couplage optimal ou encore un plan de transport optimal.

**Exemple 7.** "Transport de deux Bernoulli"

Soient  $x_1, x_2, y_1, y_2 \in \mathbb{R}^2$  et  $p_1, p_2 \in [0, 1]$ , on pose

$$\mu = p_1\delta_{x_1}(\cdot) + (1 - p_1)\delta_{x_2}(\cdot) \quad \text{et} \quad \nu = p_2\delta_{y_1}(\cdot) + (1 - p_2)\delta_{y_2}(\cdot)$$

deux mesures de probabilité modélisant deux lois de Bernoulli. Soit  $X, Y$  deux variables aléatoires définies sur le même espace de lois respectives  $\mu$  et  $\nu$ , le couplage  $\pi$  modélisant la loi du vecteur aléatoire  $(X, Y)$ .

Tout couplage  $\pi \in \Pi(\mu, \nu)$  s'apparente à une matrice de  $\mathcal{M}_2([0, 1])$  telle que :

$$\pi = \begin{pmatrix} x & y \\ z & w \end{pmatrix} = \begin{pmatrix} \mathbb{P}(\{X = x_1\} \cap \{Y = y_1\}) & \mathbb{P}(\{X = x_2\} \cap \{Y = y_1\}) \\ \mathbb{P}(\{X = x_1\} \cap \{Y = y_2\}) & \mathbb{P}(\{X = x_2\} \cap \{Y = y_2\}) \end{pmatrix} \quad (1)$$

$$\begin{cases} x + y = p_2 \\ z + w = 1 - p_2 \\ x + z = p_1 \\ y + w = 1 - p_1 \end{cases} \quad (2)$$

Ce problème est équivalent à la résolution du système

$$AX = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} X = Y_0 \quad \text{avec} \quad X \in \mathbb{R}^4, \quad Y_0 = \begin{pmatrix} p_2 \\ 1 - p_2 \\ p_1 \\ 1 - p_1 \end{pmatrix}$$

Une solution particulière de ce système est le cas d'indépendance en les variables  $X$  et  $Y$ . On trouve donc

$$X_0 = \begin{pmatrix} p_2 p_1 \\ p_2(1-p_1) \\ p_1(1-p_2) \\ (1-p_1)(1-p_2) \end{pmatrix} \text{ pour le couplage } \pi_0 = \begin{pmatrix} p_2 p_1 & p_2(1-p_1) \\ p_1(1-p_2) & (1-p_1)(1-p_2) \end{pmatrix}$$

On résout le système linéaire :

$$AX = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} X = 0_{\mathbb{R}^4}$$

Ainsi l'ensemble solution est :

$$S := \left\{ \begin{pmatrix} p_2 p_1 \\ p_2(1-p_1) \\ p_1(1-p_2) \\ (1-p_1)(1-p_2) \end{pmatrix} + \lambda \cdot \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix} : \lambda \in \mathbb{R} \right\}$$

On en déduit que  $\Pi(\mu, \nu)$  est associé à l'ensemble des matrices :

$$\Pi(\mu, \nu) \cong \left\{ \begin{pmatrix} p_2 p_1 + \lambda & p_2(1-p_1) - \lambda \\ p_1(1-p_2) - \lambda & (1-p_1)(1-p_2) + \lambda \end{pmatrix} \right\} \cap \mathcal{M}_2([0, 1])$$

La contrainte d'être dans l'ensemble  $\mathcal{M}_2([0, 1])$  implique de choisir  $\lambda$  dans un intervalle  $J$  restreint de  $\mathbb{R}$ .

Posons

$$m = \max(-p_1 p_2 ; -(1-p_1)(1-p_2) ; p_2(1-p_1) - 1 ; p_1(1-p_2) - 1)$$

$$M = \min(1 - p_1 p_2 ; 1 - (1-p_1)(1-p_2) ; p_2(1-p_1) ; p_1(1-p_2))$$

$$J = [m, M]$$

Considérons la fonction de coût  $c$  sur  $\mathbb{R}^2$  associée à un problème de transport optimal (la distance euclidienne par exemple  $\|\cdot\|$ ), pour tout  $\lambda \in J$  soit  $\pi_\lambda \in \Pi(\mu, \nu)$  on a :

$$I_c(\pi_\lambda) = c(x_1, y_1) \times (p_2 p_1 + \lambda) + c(x_2, y_1) \times (p_2(1-p_1) - \lambda) + c(x_1, y_2) \times (p_1(1-p_2) - \lambda) + c(x_2, y_2) \times ((1-p_1)(1-p_2) + \lambda)$$

Or le coût de transport optimal entre  $\mu$  et  $\nu$  est donné par :  $\mathcal{T}_c(\mu, \nu) = \inf \{I_c(\pi) : \pi \in \Pi(\mu, \nu)\}$

Nous remarquons que

$$I'_c(\pi_\lambda) = c(x_1, y_1) - c(x_2, y_1) - c(x_1, y_2) + c(x_2, y_2)$$

La fonction  $I_c$  est affine, pour connaître son argument minimum sur  $J$  il suffit de connaître le signe de  $I'_c$ . On en déduit la propriété suivante :

1. Si  $I'_c \geq 0$  le couplage optimal  $\pi^*$  est :

$$\pi^* = \begin{pmatrix} p_2 p_1 + m & p_2(1-p_1) - m \\ p_1(1-p_2) - m & (1-p_1)(1-p_2) + m \end{pmatrix}$$

2. Si  $I'_c < 0$  le couplage optimal  $\pi^*$  est :

$$\pi^* = \begin{pmatrix} p_2 p_1 + M & p_2(1-p_1) - M \\ p_1(1-p_2) - M & (1-p_1)(1-p_2) + M \end{pmatrix}$$

## Productions et stockages dans le plan

Deux sites de production  $X1$  et  $X2$  ainsi que deux espaces de stockage  $Y1$  et  $Y2$  sont fixés dans le pavé  $[-15, 15]^2$  avec des capacités de production et de stockage différentes respectivement  $p1 = 0.4$ ,  $1 - p1 = 0.6$  et  $p2 = 0.7$ ,  $1 - p2 = 0.3$ . Nous cherchons à connaître la manière la plus optimale de répartir les marchandises entre les deux entrepôts.

```
import numpy as np
import numpy.linalg as npl
import matplotlib.pyplot as plt

p1 = 0.4
p2 = 0.7

m = max(-p1*p2, -(1-p1)*(1-p2), (1-p1)*p2-1, (1-p2)*p1-1)
M = min(1-p1*p2, 1-(1-p1)*(1-p2), (1-p1)*p2, (1-p2)*p1)

def c2(x,y):
    return npl.norm(x-y,2)

def Ic(l,c):
    a = c(x1,y1)*(p2*p1 + l) + c(x2,y1)*((p2*(1-p1) - l))
    b= c(x1,y2)*((p1*(1-p2) - l))+ c(x2,y2)*((1-p1)*(1-p2)+l)
    return a+b

x1 = np.array([8,8])
x2 = np.array([1,2])
y1 = np.array([5,7])
y2 = np.array([10,0])

k = c2(x1,y1)- c2(x2,y1)- c2(x1,y2) + c2(x2,y2)

if k>=0:
    l0 = m
else:
    l0= M

print(f"K={k}, l0={l0}")

print(f"le coût optimal de ce transport est de : {Ic(l0,c2)}")

k=-2.2675133712069027 l0=0.12000000000000002
le coût optimal de ce transport est de : 5.9517116724850725
```

FIGURE 2 – Modélisation du problème sous Jupyter-notebook

```
L = np.linspace(m,M,1000)
Y = Ic(L,c2)
plt.plot(L,Y)

[<matplotlib.lines.Line2D at 0x7f441bc36f70>]
```

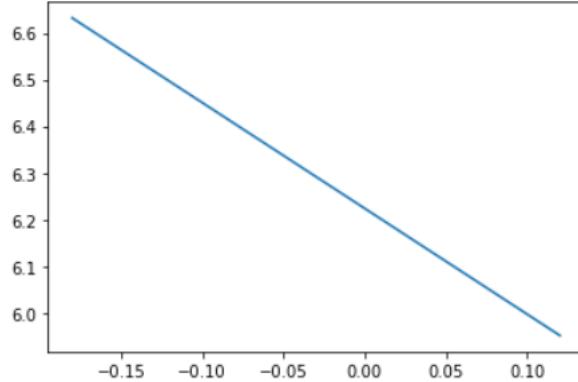


FIGURE 3 – Représentation du coût de transport pour les différentes matrices de couplage

```
pistar = np.array([[p2*p1 + l0 ,p2*(1-p1) -l0],
                  [p1*(1-p2)-l0,(1-p1)*(1-p2)+l0]])
pistar
array([[0.4, 0.3],
       [0., 0.3]])
```

FIGURE 4 – Matrice de couplage optimale

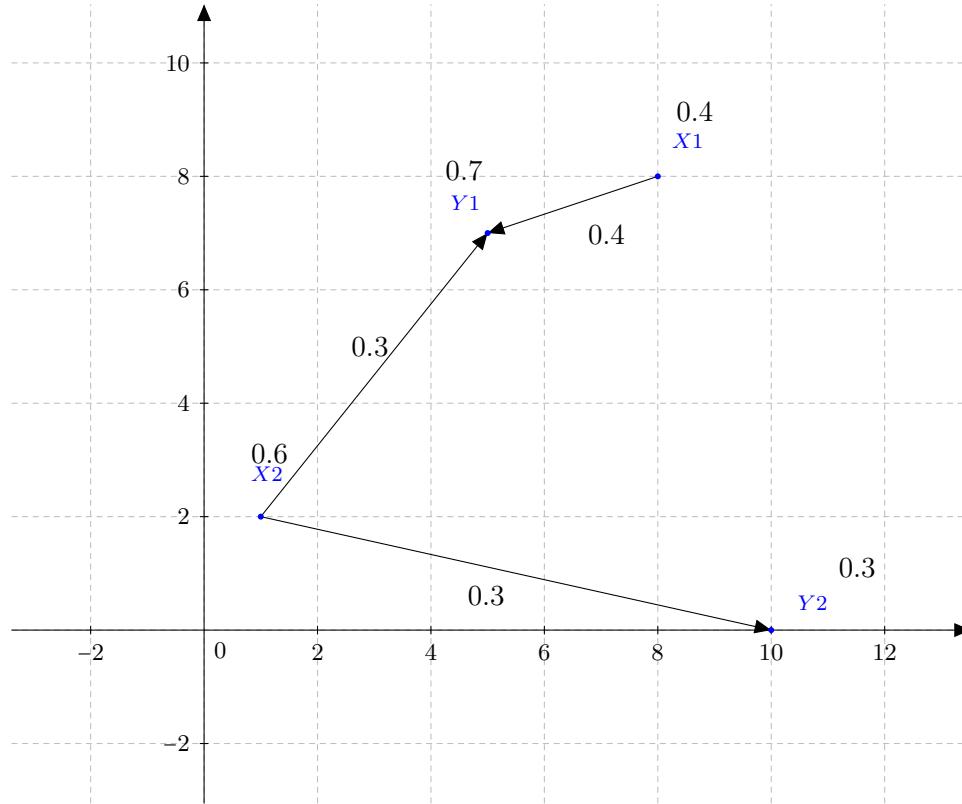


FIGURE 5 – Représentation dans le plan de la solution optimale

**Remarque 6.** Le problème de transport optimal au sens de Monge se retrouve comme un cas particulier de la formulation du problème de transport optimal selon l'approche de Kantorovich. En effet si  $T$  envoie  $\mu$  sur  $\nu$ , le couplage  $\pi_T$  de  $\mu$  et  $\nu$  est donné par le noyau  $p: x \mapsto \delta_{T(x)}(\cdot)$  et la mesure  $\mu$ .

$$I_c(\pi_T) = \iint_{E \times E} c(x, y) \delta_{T(x)}(y) \mu(dx) = \int_E c(x, T(x)) \mu(dx) = C(T) \geq T_c(\mu, \nu)$$

Si le problème de Kantorovich admet un couplage optimal  $\pi^*$  et que ce couplage est déterministe alors l'application  $T$  sera une solution du problème de Monge. Ainsi pour tout  $x \in E$ , il existe un unique  $y \in E$  tel que  $p_{\pi^*}(x) = \delta_y(\cdot)$  avec  $p_{\pi^*}$  un noyau et  $\mu$  la mesure source qui sont tous deux associés au couplage  $\pi^*$ .

### III Dualité de Kantorovich et distance de Wassertein

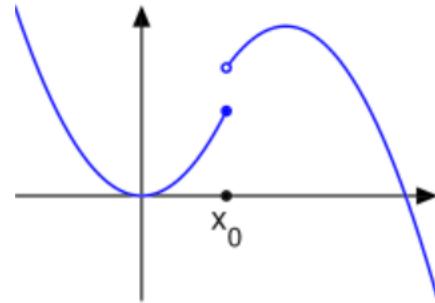
Dans cette partie de notre rapport, nous allons apporter quelques éléments théoriques associés au transport optimal. Nous ne démontrerons pas tous les résultats et nous les supposerons parfois admis par la suite.

#### 1 Existence d'un couplage optimal pour une fonction de coût sci.

**Définition 4.**

Une fonction  $f : E \rightarrow \mathbb{R} \cup \{+\infty\}$  est dite semi-continue inférieurement en  $x \in E$  si pour toute suite  $(x_n)_{n \in \mathbb{N}}$  convergeant vers  $x$ , on a

$$\liminf_{n \rightarrow +\infty} f(x_n) \geq f(x)$$



**Remarque 7.** Une fonction est dite semi-continue inférieurement sur  $E$  si elle est semi-continue inférieurement en tout point de  $E$ . On notera « semi-continue inférieurement » par sci.

**Remarque 8.**  $f$  est dite sci en  $x_0$  si, lorsque  $x$  est proche de  $x_0$ ,  $f(x)$  est soit proche de  $f(x_0)$ , soit supérieur à  $f(x_0)$ .

**Propriété 3.** Une fonction  $f : E \rightarrow \mathbb{R} \cup \{+\infty\}$  continue sur  $E$  est sci sur  $E$ .

*Démonstration.* Pour tout  $x \in E$  et toute suite  $(x_n)_{n \in \mathbb{N}}$  convergeant vers  $x$ ,

$$\liminf_{n \rightarrow +\infty} f(x_n) = \lim_{n \rightarrow +\infty} f(x_n) = f(x) \geq f(x)$$

□

**Propriété 4. (Admis.)**

Une fonction sci sur un compact est minorée et atteint sa borne inférieure.

**Propriété 5. "Sup de fonctions sci, régularisation" (Admis.)**

1. Si  $(f_i)_{i \in I}$  est une famille quelconque de fonctions sci, alors  $f = \sup_{i \in I} f_i$  est sci.
2. Si  $f$  est une application sci minorée, alors il existe une suite croissante  $(f_n)_{n \in \mathbb{N}}$  de fonctions continues bornées telles que  $f = \sup_{n \in \mathbb{N}} f_n$ . On peut même supposer que pour tout  $n \in \mathbb{N}$ , la fonction  $f_n$  est  $n$ -Lipschitz sur  $X$ .

Soit  $(E, d)$  est un espace polonais, on va munir l'espace  $\mathcal{P}(E)$  des mesures de probabilité de Borel sur  $E$  de la topologie faible ou encore appelée topologie de la convergence étroite. On notera  $\mathcal{C}_b(E)$  l'ensemble des fonctions continues bornées sur  $E$ .

**Définition 5.**

La topologie faible sur  $\mathcal{P}(E)$  est la topologie la moins fine rendant continues les applications

$$\mu \in \mathcal{P}(E) \longmapsto \int_E \Phi(x) d\mu(x)$$

telle que  $\Phi \in C_b(E)$ .

**Définition 6.**

Une suite  $(\mu_n)_{n \in \mathbb{N}}$  de mesures de probabilité converge vers  $\mu$  pour la topologie de la convergence étroite si

$$\int_E \Phi(x) d\mu_n(x) \xrightarrow{n \rightarrow +\infty} \int_E \Phi(x) d\mu(x)$$

pour toute  $\Phi \in C_b(E)$ .

**Propriété 6. (Admis.)**

Pour toutes mesures de probabilité  $\mu, \nu \in \mathcal{P}(E)$ , l'espace des couplages entre  $\mu$  et  $\nu$  noté  $\Pi(\mu, \nu)$  est un compact de  $\mathcal{P}(E \times E)$  muni de la topologie faible.

**Théorème 1. "Existence d'un couplage optimal"**

Soit  $c : E \times E \rightarrow [0, +\infty]$  une fonction de coût sci. Pour toutes mesures de probabilité  $\mu, \nu \in \mathcal{P}(E)$ , il existe  $\pi^* \in \Pi(\mu, \nu)$  tel que

$$I_c(\pi^*) = \iint_{E \times E} c(x, y) \pi^*(dx dy) = T_c(\mu, \nu) = \inf \{I_c(\pi) : \pi \in \Pi(\mu, \nu)\}$$

*Démonstration.*

- Montrons que la fonction  $I_c$  est sci sur  $\mathcal{P}(E \times E)$  :

La fonction  $c$  est positive donc minorée et sci. Ainsi d'après le point 2 de la Proposition 5, il existe une suite de fonctions  $(c_n)$  continues positives et bornées telle que  $c = \sup_{n \in \mathbb{N}} c_n$ . Par le théorème de convergence monotone, pour tout  $\pi \in \Pi(\mu, \nu)$

$$I_c(\pi) = \int_{E \times E} c(x, y) \pi(dx dy) = \int_{E \times E} \sup c_n(x, y) \pi(dx dy) = \sup \int_{E \times E} c_n(x, y) \pi(dx dy) = \sup I_{c_n}(\pi)$$

Comme  $c_n$  est continue bornée, par définition de la topologie de la convergence étroite sur  $\mathcal{P}(E \times E)$ , la fonction  $I_{c_n} : \pi \mapsto I_{c_n}(\pi)$  est continue sur  $\mathcal{P}(E \times E)$ . Donc elle est sci sur  $\mathcal{P}(E \times E)$  par la propriété 3. Enfin la fonction  $I_c$  est elle-même sci comme sup de fonctions sci par le point 1 de la propriété 5.

- Montrons qu'il existe  $\pi^* \in \Pi(\mu, \nu)$  tel que  $I_c(\pi^*) = T_c(\mu, \nu)$ .

Par la propriété 6,  $\Pi(\mu, \nu)$  est un compact de  $\mathcal{P}(E \times E)$ . Or l'application  $I_c$  est sci sur  $\mathcal{P}(E \times E)$ , par la propriété 4, la fonction  $I_c$  est minorée et atteint sa borne inférieure.

□

## 2 Dualité de Kantorovich

**Théorème 2.** "Dualité de Kantorovich"

Soit  $c : E \times E \rightarrow [0, +\infty[$  une fonction de coût sci. Pour toutes mesures de probabilité  $\mu, \nu \in \mathcal{P}(E)$  telles que  $T_c(\mu, \nu) < +\infty$ . On a alors

$$T_c(\mu, \nu) = \sup_{(\varphi, \psi) \in \Phi_b^2} \left\{ \int_E \varphi(x) d\mu(x) + \int_E \psi(y) d\nu(y) \right\}$$

où  $\Phi_b$  est l'ensemble des fonctions continues bornées telles que

$$\psi(x) + \varphi(y) \leq c(x, y), \quad \forall x, y \in E.$$

*Démonstration.* • Montrons l'inégalité suivante :

$$\sup_{(\varphi, \psi) \in \Phi_b^2} \left\{ \int_E \varphi(x) d\mu(x) + \int_E \psi(y) d\nu(y) \right\} \leq T_c(\mu, \nu)$$

Soit  $\pi \in \Pi(\mu, \nu)$  et  $\varphi, \psi \in \Phi_b^2$  on a par hypothèse pour tout  $(x, y) \in E^2$

$$\psi(x) + \varphi(y) \leq c(x, y)$$

Ainsi par croissance de l'intégrale on a

$$\begin{aligned} \int_{E \times E} c(x, y) \pi(dx dy) &\geq \int_{E \times E} (\psi(x) + \varphi(y)) \pi(dx dy) \\ &= \int_{E \times E} \psi(x) \pi(dx dy) + \int_{E \times E} \varphi(y) \pi(dx dy) \\ &= \int_E \psi(x) \mu(dx) + \int_E \varphi(y) \nu(dy) \end{aligned}$$

Par passage à l'infimum sur  $\Pi(\mu, \nu)$  on obtient

$$T_c(\mu, \nu) \geq \int_E \psi(x) \mu(dx) + \int_E \varphi(y) \nu(dy)$$

Enfin, par passage au sup sur  $\Pi(\mu, \nu)^2$  on obtient le résultat voulu

$$\sup_{(\varphi, \psi) \in \Phi_b^2} \left\{ \int_E \varphi(x) d\mu(x) + \int_E \psi(y) d\nu(y) \right\} \leq T_c(\mu, \nu)$$

• L'inégalité suivante sera admise dans la suite de ce rapport :

$$\sup_{(\varphi, \psi) \in \Phi_b^2} \left\{ \int_E \varphi(x) d\mu(x) + \int_E \psi(y) d\nu(y) \right\} \geq T_c(\mu, \nu)$$

□

**Remarque 9.** La dualité de Kantorovitch nous permet de faire la connexion entre la formulation duale et la formulation primaire dans des problèmes de transport optimal. En particulier, elle nous permet pour le LSAP (Linear Sum Assignment Problem, vu ultérieurement) d'appliquer l'algorithme hongrois pour obtenir la solution optimale.

### 3 Distance de Wasserstein

Soit  $(E, d)$  un espace métrique polonais avec une fonction de coût  $c$  supposée *sci* sur  $E \times E$ .  $T_c(\mu, \nu)$  est le coût de transport dit optimal nous permettant d'envoyer la mesure de probabilité  $\mu$  sur la mesure de probabilité  $\nu$ . Nous aimerions pouvoir comparer des mesures de probabilité et ainsi dire si elles sont proches ou non dans l'espace  $\mathcal{P}(E)$ . Pour cela, il serait judicieux d'avoir une notion de distance sur l'espace  $\mathcal{P}(E)$ .

Pour  $p \in [1, +\infty[$ , notons

$$T_p(\mu, \nu) = \inf \left\{ \int_{E \times E} d(x, y)^p d\pi(x, y) : \pi \in \Pi(\mu, \nu) \right\}$$

**Définition 7.** Pour  $p \in [1, +\infty[$  on pose  $\mathcal{P}_p(E)$  l'ensemble des mesures de probabilité qui admettent un moment d'ordre  $p$ .

$$\mathcal{P}_p(E) = \left\{ \mu \in \mathcal{P}(E) : \exists x_0 \in E, \int_E d(x, x_0)^p d\mu(x) < +\infty \right\}$$

**Remarque 10.** Le choix du point  $x_0$  n'a pas d'importance dans le caractère fini de l'intégrale précédente, il s'agit de remarquer pour  $x_1 \in E$  que par l'inégalité triangulaire on a :  $d(x_0, x) \leq d(x_0, x_1) + d(x_1, x)$ . Ainsi par croissance de  $(x \mapsto x^p)$  et croissance de l'intégrale on a :

$$\int_E d(x, x_1)^p d\mu(x) \leq \int_E (d(x_0, x_1) + d(x_0, x))^p d\mu(x) \stackrel{(*)}{<} +\infty$$

(\*) On utilise l'inégalité de Minkowski pour la mesure de probabilité  $\mu$  pour la majoration et le fait que  $\mu$  est de masse finie (les constantes sont intégrables).

**Propriété-définition 1.** "Distance de Wasserstein"

Pour tout couple  $(\mu, \nu) \in \mathcal{P}_p(E)^2$ , la quantité

$$W_p(\mu, \nu) = T_p(\mu, \nu)^{\frac{1}{p}}$$

est appelée la distance de Wasserstein suivant la valeur  $p$ . L'application

$$\begin{aligned} W_p : \quad \mathcal{P}_p(E)^2 &\longrightarrow \mathbb{R}^+ \\ (\mu, \nu) &\longmapsto T_p(\mu, \nu)^{\frac{1}{p}} \end{aligned}$$

est une distance sur  $\mathcal{P}_p(E)$ .

*Démonstration.*

Soit  $\mu, \nu, \gamma \in \mathcal{P}_p(E)$

- Montrons que  $W_p$  vérifie la symétrie :

$$W_p(\mu, \nu) = T_p(\mu, \nu)^{\frac{1}{p}} = T_p(\nu, \mu)^{\frac{1}{p}} = W_p(\nu, \mu)$$

On utilise seulement la symétrie de la distance  $d$ .

- Montrons que  $W_p$  vérifie la séparation :

- Si  $\mu = \nu$  alors  $X = Y$   $\mathbb{P}$ -p.s. ainsi pour tout couplage  $\pi \in \Pi(\mu, \nu)$  on a  $\int_{E \times E} d(x, y)^p d\pi(x, y) = 0$ .  
Donc  $T_p(\mu, \nu)^{\frac{1}{p}} = 0 = W_p(\mu, \nu)$
- Si  $W_p(\mu, \nu) = 0$ , pour  $\pi_1$  le couplage optimal associé à  $T_p(\mu, \nu)$  on sait que  $d(X, Y)^p = 0$   $\pi_1$ -p.s. et par le caractère séparable de  $d$  on sait que  $X = Y$   $\pi_1$ -p.s. Pour toute fonction continue bornée  $\Phi : E \rightarrow \mathbb{R}$  on a

$$\mathbb{E}(\Phi(X)) = \int_E \Phi(x) d\mu(x) = \int_{E \times E} \Phi(x) d\pi_1(x, y) = \int_{E \times E} \Phi(y) d\pi_1(x, y) = \int_E \Phi(y) d\nu(y) = \mathbb{E}(\Phi(Y))$$

Ainsi  $X$  et  $Y$  ont la même loi donc  $\mu = \nu$ .

- Montrons que  $W_p$  vérifie l'inégalité triangulaire :

- Notons  $X, Y, Z$  trois variables aléatoires définies sur un même espace probabilisé  $(\Omega, \mathcal{B}, \mathbb{P})$  à valeurs dans  $E$  de loi respective  $\mu, \nu, \gamma$ .
- Supposons que  $(X, Z)$  et  $(Y, Z)$  sont les couples optimaux en lien avec le problème de transport optimal.  
On peut leur associer respectivement les couplages  $\pi_1$  et  $\pi_2$ . Soit  $\pi \in \mathcal{P}(E \times E \times E)$  la loi du triplet  $(X, Y, Z)$ . On a ainsi

$$\mathbb{E}_\pi(d(X, Z)^p) = \mathbb{E}_{\pi_1}(d(X, Z)^p) = T_p(\mu, \gamma) \quad \text{et} \quad \mathbb{E}_\pi(d(Y, Z)^p) = \mathbb{E}_{\pi_2}(d(Y, Z)^p) = T_p(\nu, \gamma)$$

Par inégalité triangulaire dans  $(E, d)$ , pour tout  $\omega \in \Omega$

$$0 \leq d(X(\omega), Y(\omega))^p \leq (d(X(\omega), Z(\omega)) + d(Y(\omega), Z(\omega)))^p$$

Soit  $U = d(X, Y), V = d(X, Z), W = d(Y, Z)$  des variables aléatoires sur  $(\Omega, \mathcal{B}, \mathbb{P})$  à valeurs dans  $\mathbb{R}^+$ .

$$\begin{aligned} W_p(\mu, \nu) &= \mathbb{E}_\pi(U^p)^{\frac{1}{p}} \leq \mathbb{E}_\pi((V + W)^p)^{\frac{1}{p}} \\ &= \|V + W\|_{L^p(\pi)} \\ &\stackrel{(*)}{\leq} \|V\|_{L^p(\pi)} + \|W\|_{L^p(\pi)} \\ &= \mathbb{E}_\pi(V^p)^{\frac{1}{p}} + \mathbb{E}_\pi(W^p)^{\frac{1}{p}} = W_p(\mu, \gamma) + W_p(\gamma, \nu) \end{aligned}$$

(\*) Nous utilisons l'inégalité triangulaire pour l'espace normé  $(L^p(\pi), \|\cdot\|_{L^p(\pi)})$

□

## IV Algorithme Hongrois

### 1 Problème

On rappelle le problème de transport optimal à résoudre.

$$\left\{ \begin{array}{l} \text{Minimiser : } \int c(x, y) d\pi(x, y) \\ \text{Étant donné : les mesures } \mu, \nu \text{ et le coût } c \\ \text{Sous la contrainte : } \pi \text{ a pour marginales } \mu \text{ et } \nu \end{array} \right.$$

Pour résoudre le problème, il existe des algorithmes permettant de construire des solutions au problème simplifié (par discréétisation). On considère alors le problème où  $\mu$  et  $\nu$  sont des sommes finies de masses de Dirac.

#### a Problème discret

Le problème discret se formule donc de la même façon avec  $\mu$  et  $\nu$  des mesures discrètes.

$$\mu = \sum_{i=1}^n p_i \delta_{a_i} \text{ et } \nu = \sum_{i=1}^m q_i \delta_{b_i}$$

Et tout couplage  $\pi \in \Pi(\mu, \nu)$  est donc discret. Un plan de transport correspond à une matrice de taille  $m \times n$ . On le notera  $x = (x_{i,j})_{i \in \llbracket 1; n \rrbracket, j \in \llbracket 1; m \rrbracket}$ .

$$\left\{ \begin{array}{l} \text{Minimiser : } \sum_{ij} c_{i,j} \cdot x_{ij} \\ \text{Sous la contrainte : } \forall i, \sum_j x_{ij} = p_i ; \quad \forall j, \sum_i x_{ij} = q_j \\ \text{et } \forall i, j \quad x_{ij} \geq 0 \end{array} \right.$$

**Remarque 11.** *X* est dit faisable s'il respecte les contraintes du problème (marginales).

**Remarque 12.** *X* est dit optimal s'il minimise le coût total  $\sum_{ij} c_{i,j} \cdot x_{ij}$ , ce minimum est appelé valeur du problème.

#### b Linear Sum Assignment Problem (LSAP)

Ce problème est un cas particulier du problème discret. Les mesures  $\mu$  et  $\nu$  ont le même nombre  $N$  de points, et les  $p_i$  et  $q_i$  sont égaux à  $\frac{1}{N}$ . De plus, la solution a une contrainte supplémentaire : les  $x_{ij}$  sont égaux à 0 ou 1.

**Remarque 13.** Dans le Problème de LSAP, il existe une approche "duale" vue dans théorème 2, elle permet de faciliter la recherche de solutions.

## 2 Présentation de l'algorithme Hongrois

L'algorithme Hongrois permet de construire une assignation optimale pour le problème de LSAP relaché, et construit un couple solution du problème dual  $(\phi, \psi)$ . Tout au long de l'algorithme on disposera :

d'un couple  $(\phi, \psi)$  faisable pour le problème dual, c'est-à-dire vérifiant  $\phi(i) + \psi(j) \leq c_{ij}$

d'une assignation partielle  $x_{ij} \in \{0, 1\}$  vérifiant  $\sum_i x_{ij} \leq 1$ ,  $\sum_j x_{ij} \leq 1$ , et la lâcheté complémentaire : si  $x_{ij}$  vaut 1 alors l'arête (i,j) est saturée. Autrement dit : seules les arêtes saturées peuvent être assignées.

Le nombre de sites assignés est croissant pendant l'algorithme ; celui-ci finit quand tous les sites sont assignés.

### a Vue d'ensemble du fonctionnement de l'algorithme.

#### Initialisation

On part de l'assignation vide  $x = 0$  et de  $(\phi, \psi)$  identiquement nulle. On « augmente » d'abord  $\phi$  et  $\psi$  en deux étapes.

1. Pour i de 1 à N on pose  $\phi(i) = \min_j(c_{ij})$ . C'est le  $\phi$  le plus grand tel que le couple  $(\phi, \psi)$  reste faisable
2. Puis, pour j de 1 à N on pose  $\psi(j) = \min_i(c_{ij} - \phi(i))$ . À  $\phi$  fixé, c'est le  $\psi$  le plus grand tel que  $(\phi, \psi)$  reste faisable

Enfin, pour i entre 1 et N :

1. S'il existe j non-assigné tel que l'arête (i, j) est saturée, on assigne i au premier j trouvé (on change  $x_{ij}$  en 1) ;
2. Sinon i reste non-assigné.

#### Boucle principale

Pour chaque sommet non assigné :

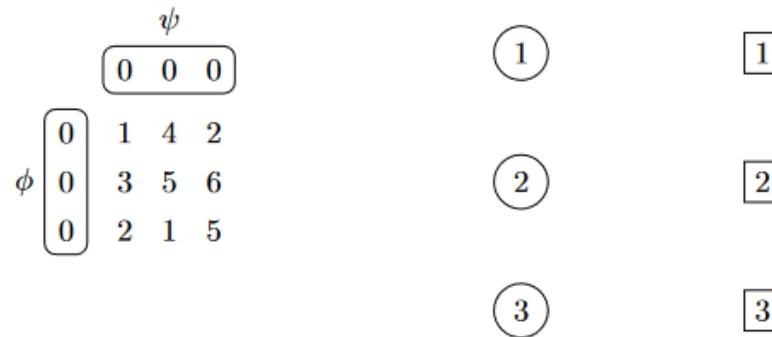
1. Chercher dans le graphe des arêtes saturées un chemin augmentant.
2. S'il y en a un, l'inverser.
3. S'il n'y en a pas, créer des arêtes saturées en modifiant  $(\phi, \psi)$ , et revenir au 1. C'est la procédure d'ouverture d'arêtes saturées. Cette procédure peut ouvrir et fermer des arêtes saturées. (cela fait évoluer le graphe et on peut avancer dans l'algorithme)

### b Exemple

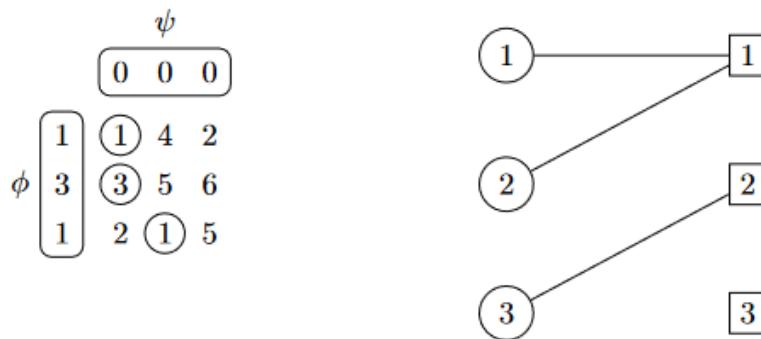
Dans l'exemple, la matrice de coûts est

$$c = \begin{pmatrix} 1 & 4 & 2 \\ 3 & 5 & 6 \\ 2 & 1 & 5 \end{pmatrix}$$

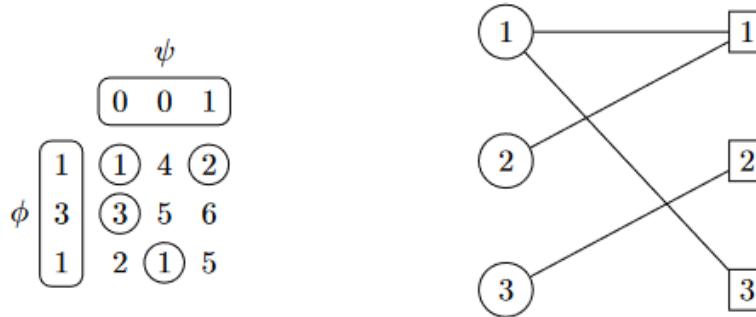
**Initialisation.** Au départ de l'algorithme on a donc :



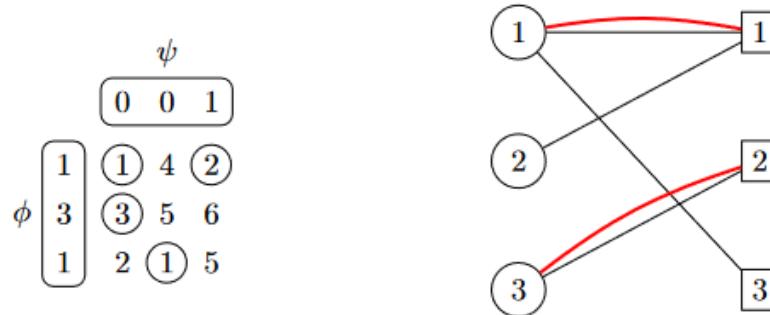
On met à jour  $\phi$ , ce qui sature trois arêtes :



On met à jour  $\psi$ . Seul  $\psi(3)$  peut être augmenté jusqu'à 1, ce qui donne :



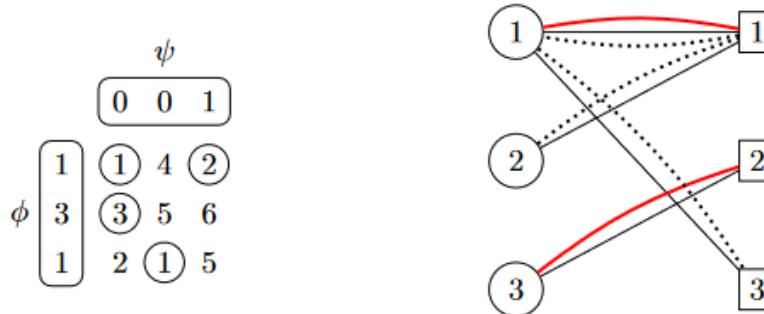
Enfin on essaie d'assigner les sommets de gauche : (1) est assigné à [1], (2) ne peut pas être assigné puisque [1] est déjà pris, et (3) est assigné à [2].



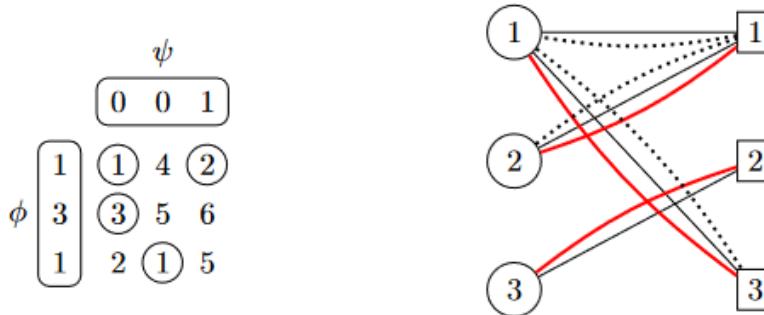
**Recherche d'un chemin augmentant.** Partant d'un sommet de gauche  $u_1$  non-assigné, on cherche les chemins  $(u_1, v_1, u_2, v_2, \dots)$  dans le graphe des arêtes saturées tels que :

1. les arêtes « aller »  $(u_1, v_1), (u_2, v_2), \dots$  sont non-assignées ;
2. les arêtes « retour »  $(v_1, u_2), (v_2, u_3), \dots$  sont assignées.

On construit ainsi un *arbre* dont les branches sont des chemins *alternés* entre arêtes assignées et non-assignées. Si un des chemins finit en un sommet de droite  $v_k$  non-assigné, on dit que le chemin est *augmentant*. Un chemin augmentant contient donc  $k$  arêtes non-assignées et  $k - 1$  arêtes assignées. Dans notre exemple le chemin  $(2) \rightarrow (1) \rightarrow (1) \rightarrow (3)$  est augmentant.



**Inversion** On modifie l'assignation des arêtes le long du chemin augmentant : les arêtes non-assignées deviennent assignées et réciproquement. Comme le chemin augmentant n'utilise que des arêtes saturées, la condition de lâcheté reste vérifiée, mais le nombre d'arêtes assignées augmente de 1 (on désassigne  $k - 1$  arêtes et on en assigne  $k$ ). On arrive finalement à



On remarque que l'assignation est complète à ce stade, et vérifie la propriété de lâcheté complémentaire : on a donc trouvé un plan de transport optimal. Le coût de transport vaut  $6 = \sum \phi(i) + \sum \psi(j) = c_{13} + c_{21} + c_{32}$ .

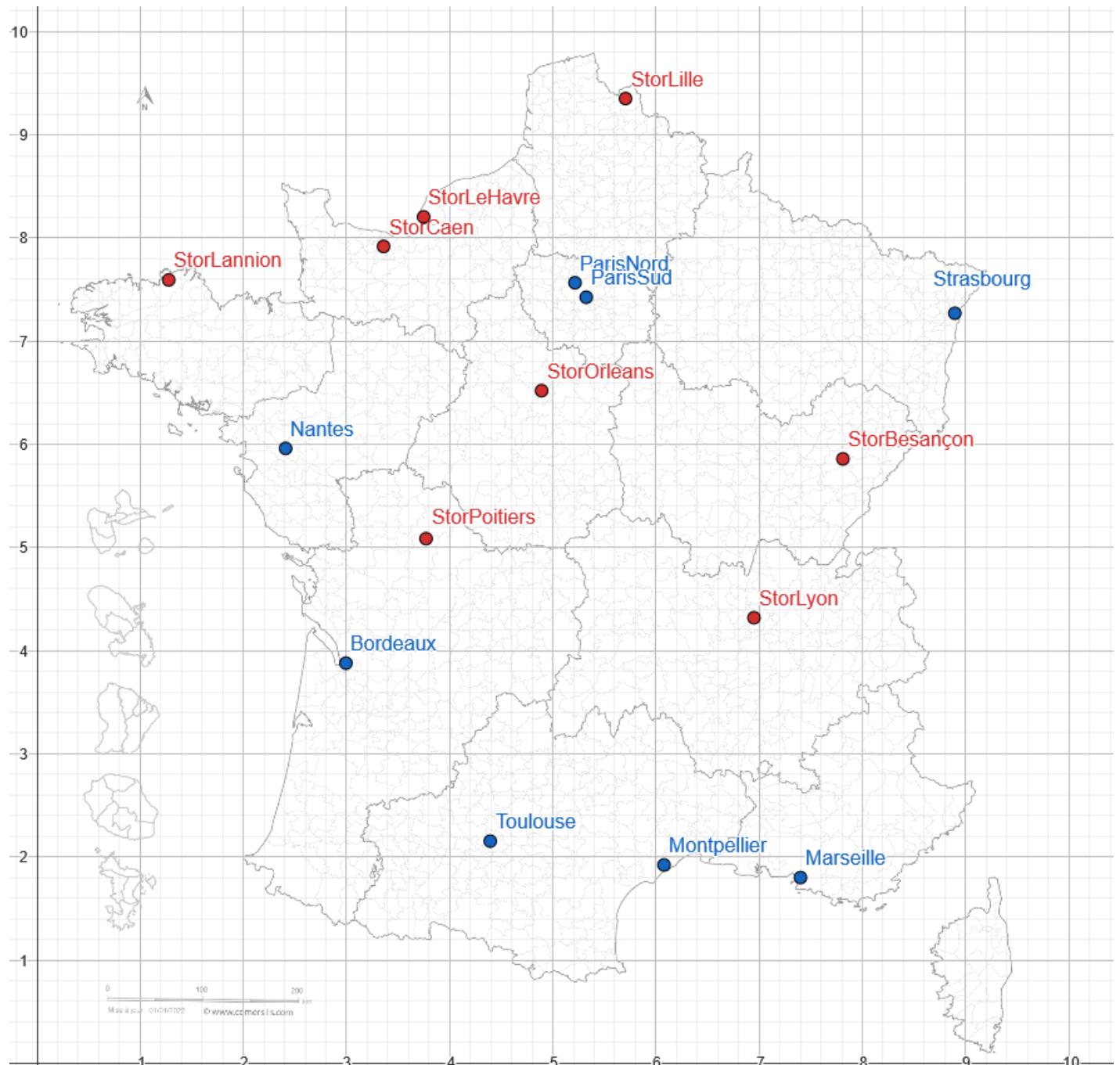
### 3 Application à un problème dans le plan

#### a Problème

Nous avons choisi une première application pour illustrer l'application de l'algorithme Hongrois au problème du Linear Sum Assignment Problem.

Le contexte est le suivant : nous avons 8 entrepôts/usines/fournisseurs situés dans différentes villes, qui doivent alimenter 8 magasins dans d'autres villes. Nous souhaitons assigner à chaque magasin un fournisseur de manière à ce que les coûts de transports soient les plus limités possibles. Dans notre problème, les coûts de transports entre un fournisseur et le client sont proportionnels au temps de trajet par voie terrestre pour les transporteurs.

Le problème est similaire au problème des tas de sable de Monge, ainsi qu'à d'autres problèmes d'assignation au moindre coût. On place les villes du problème sur une carte, et les temps de trajets dans une matrice des coûts.



Sous python, nous avons entré les villes et la matrice des coûts pour paramétriser le problème.

```

1 # Les donnees du probleme.
2 import numpy as np
3 stor={0:'Lannion',1:'Le Havre',2:'Caen',3:'Lille',4:'Orleans', 5:'Besancon',
       ,6:'Lyon',7:'Poitiers'}
4 dest={0:'Paris Nord',1:'Paris Sud',2:'Strasbourg',3:'Nantes',4:'Bordeaux',
      5:'Toulouse',6:'Montpellier',7:'Marseille'}
5 # matrice des couts
6 c=np.array([[4.75,4.5,8.5,2.68,5.5,7.5,9.25,10.33],
7 [2.125,2.125,5.99,3.65,5.9,7.6,8.16,8.42],
8 [2.16,2.13,6.17,2.71,5.2,7.1,7.85,8.5],
9 [1.9,2.13,4.68,5.28,7.33,7.96,8.4,8.46],
10 [1.6,1.31,5.18,3.05,4.03,4.83,5.41,6.58],
11 [3.68,3.42,2.38,6.28,6.63,6.16,4.83,4.86],
12 [4.11,3.83,4.28,6.11,4.85,4.7,2.78,2.82],
13 [3.35,3.08,7.13,2.,2.28,4.18,6.38,7.78]])
```

## b Méthode de Résolution : Approche par graphe

Nous avons utilisé 2 méthodes différentes pour appliquer l'algorithme Hongrois dans Python. La première est celle que nous détaillons dans cette partie. Nous utilisons le module networkx de python. Cette méthode est intéressante pour représenter le problème de LSAP avec des graphes. Nous utilisons la méthode minimum\_weight\_full\_matching du module networkx, pour obtenir une solution au problème LSAP.

```

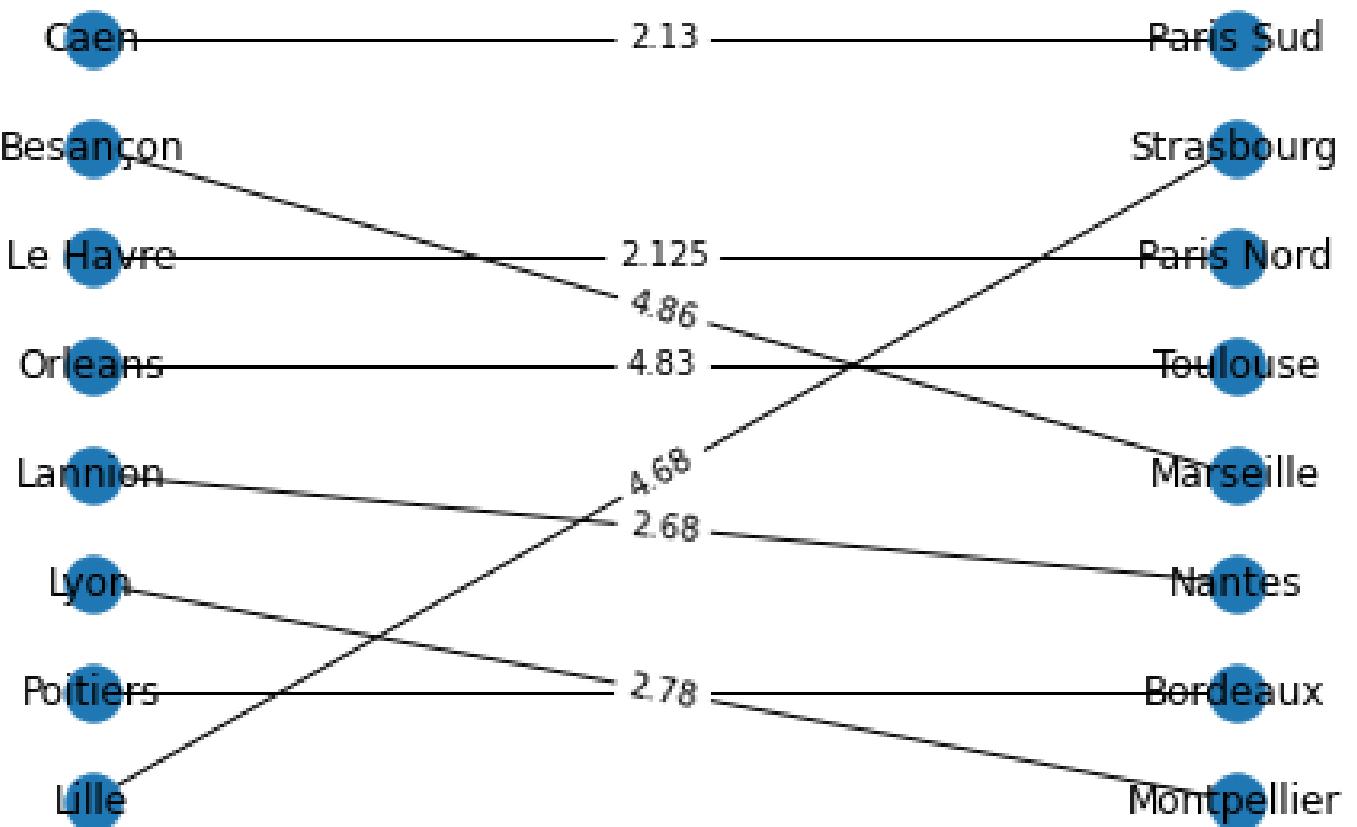
1 import networkx as nx
2 import matplotlib.pyplot as plt
3 %matplotlib inline
4
5 # On cree un graphe
6 G = nx.Graph()
7
8 ls=list(stor.values())
9 ld=list(dest.values())
10
11 # On cree les noeuds du Graphe
12 G.add_nodes_from(ls, bipartite=0)
13 G.add_nodes_from(ld, bipartite=1)
14
15 costs=[(stor[i],dest[j],c[i,j]) for j in range(c.shape[0]) for i in range(c.
   shape[0])]
16
17 # On cree les arretes du graphe (avec les poids egaux au cout)
18 G.add_weighted_edges_from(costs)
```

```

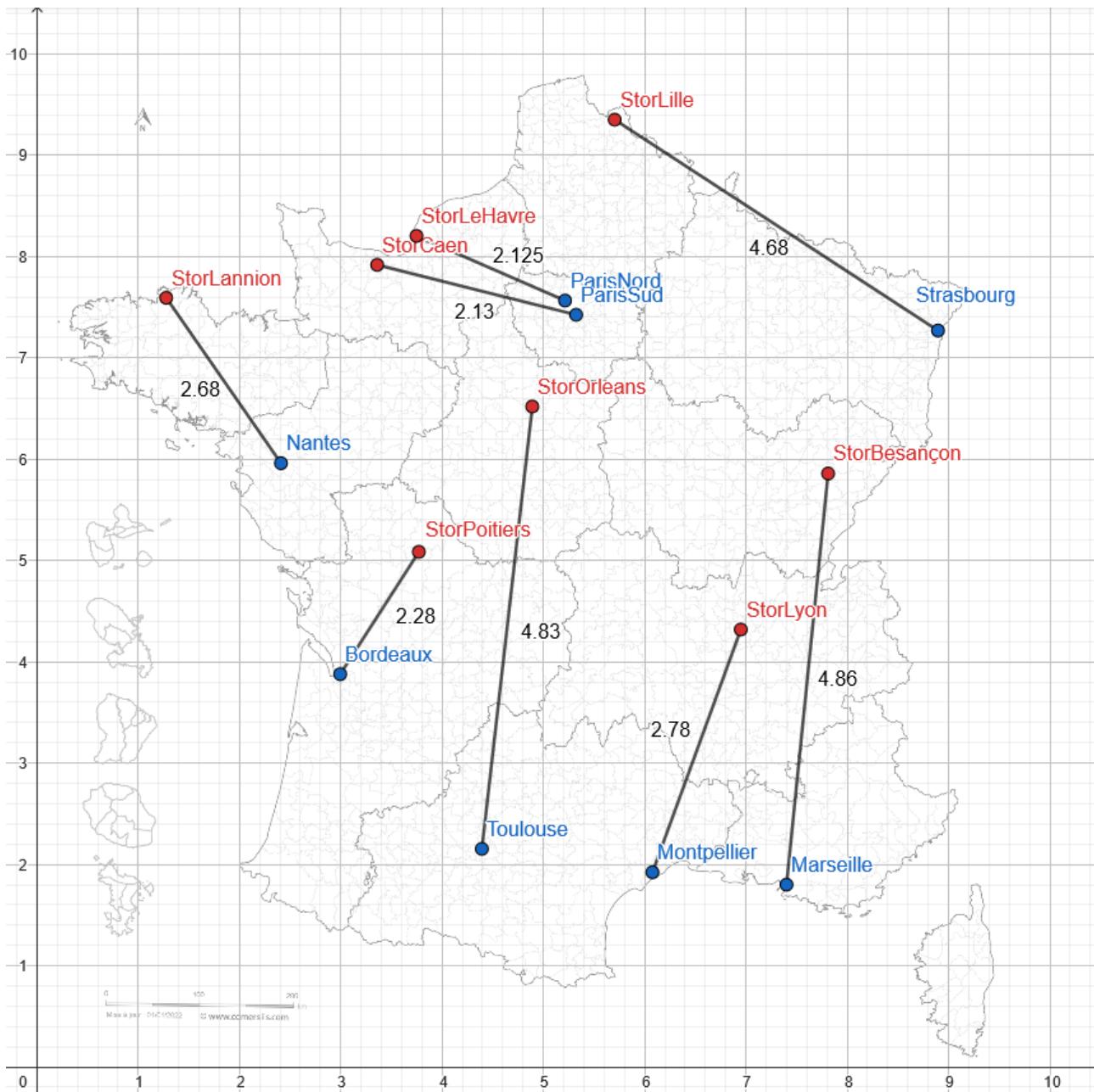
19
20 # On obtient la solution au probleme LSAP par l'algorithme hongrois.
21 matching = nx.bipartite.minimum_weight_full_matching(G, top_nodes=ls)
22
23 # On cree un nouveau graphe d'assignation pour afficher la solution
24 assignment_graph = nx.Graph()
25 for a, b in matching.items():
26     assignment_graph.add_edge(a, b, weight=G[a][b]['weight'])
27
28 # Affichage du resultat
29 pos = nx.bipartite_layout(assignment_graph, ls)
30 nx.draw(assignment_graph, pos=pos, with_labels=True)
31 edge_labels = nx.get_edge_attributes(assignment_graph, 'weight')
32 nx.draw_networkx_edge_labels(assignment_graph, pos=pos, edge_labels=
    edge_labels)
33 plt.show()
34
35 # On affiche le cout total
36 sum(edge_labels.values())

```

On obtient le graphe solution suivant, pour un coût total de 26.365 (Heures des trajets combinées) :



Nous avons représenté la solution au problème sur la carte.



### c Méthode de Résolution : Autre version de l'algorithme Hongrois

La deuxième méthode pour appliquer l'algorithme hongrois que nous avons testé, utilise le module Munkres. Ce module contient des méthodes utilisant l'algorithme Hongrois.

**Remarque 14.** Notons que Munkres fait référence à James Raymond Munkres, mathématicien américain ayant contribué (entre autres choses) au développement de l'algorithme Hongrois. D'ailleurs, l'algorithme Hongrois est aussi appelé algorithme de Kuhn-Munkres.

Nous utilisons la méthode compute du module Munkres, pour obtenir une solution au problème LSAP.

```

1 import copy
2 import networkx as nx
3 import matplotlib.pyplot as plt
4 from munkres import Munkres
5 %matplotlib inline
6
7 # On recree la matrice des couts adaptee au module
8 c1=[[4.75,4.5,8.5,2.68,5.5,7.5,9.25,10.33],
9 [2.125,2.125,5.99,3.65,5.9,7.6,8.16,8.42],
10 [2.16,2.13,6.17,2.71,5.2,7.1,7.85,8.5],
11 [1.9,2.13,4.68,5.28,7.33,7.96,8.4,8.46],
12 [1.6,1.31,5.18,3.05,4.03,4.83,5.41,6.58],
13 [3.68,3.42,2.38,6.28,6.63,6.16,4.83,4.86],
14 [4.11,3.83,4.28,6.11,4.85,4.7,2.78,2.82],
15 [3.35,3.08,7.13,2.,2.28,4.18,6.38,7.78]]
16
17 # Fonction de resolution utilisant Munkres, avec une matrice de couts en
18 # entree
19 def solve_assignment(matrix):
20     m = Munkres()
21
22     # resort une liste de tuples (indice_ligne, indice_colonne) correspondants a
23     # l'assignation (solution du LSAP) issue #de l'algorithme hongrois
24     indexes = m.compute(matrix)
25
26     # calcul du cout total
27     total_cost = sum(matrix[row][col] for row, col in indexes)
28
29     # on met en forme le resultat dans une matrice d'assignation
30     assignment_matrix = [[0] * len(matrix) for _ in range(len(matrix[0]))]
31     for row, col in indexes:
32         assignment_matrix[row][col] = 1
33
34     return indexes, assignment_matrix, total_cost
35
36
37 # application a notre probleme
38 index, assignment, cost = solve_assignment(c1)
39
40 print(index)
41 print("Assignment Matrix:")
42 for row in assignment:
43     print(row)
44 print("Total Cost:", cost)

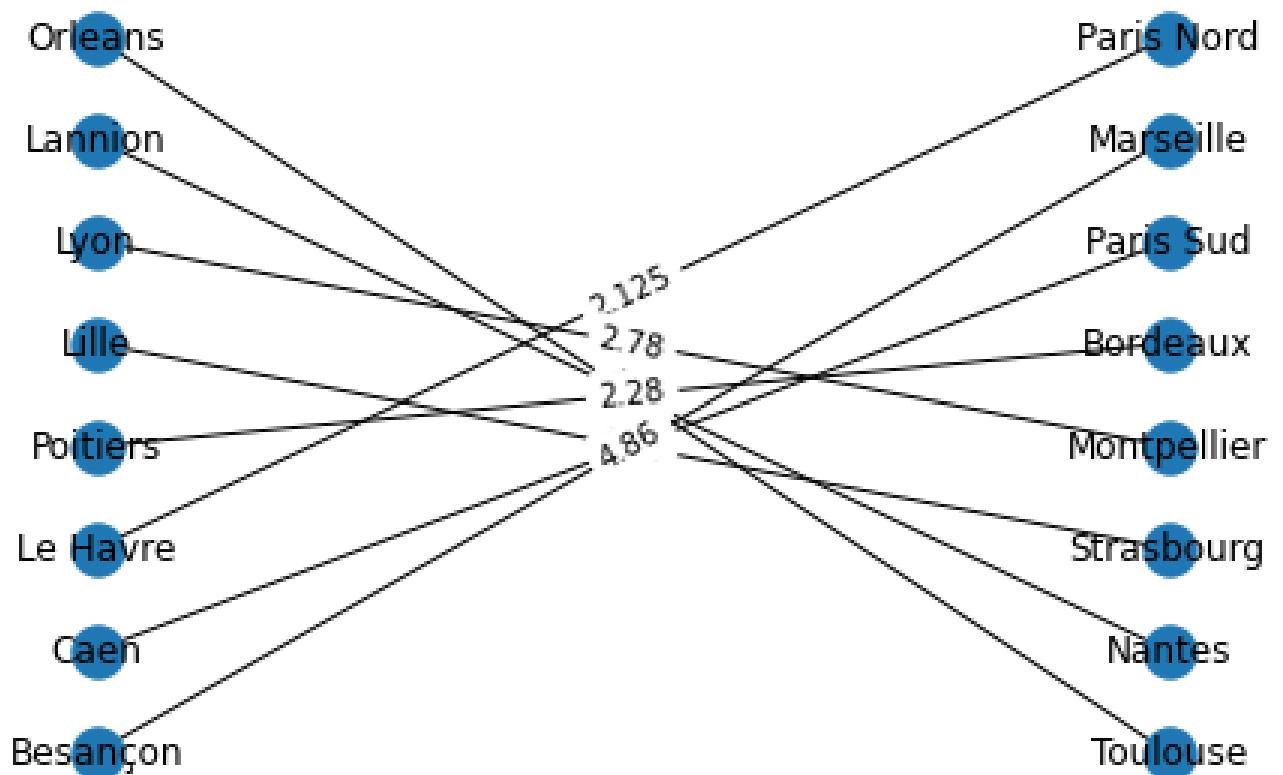
```

```

42
43 # [(0, 3), (1, 0), (2, 1), (3, 2), (4, 5), (5, 7), (6, 6), (7, 4)]
44 # Assignment Matrix:[
45 #     [0, 0, 0, 1, 0, 0, 0, 0]
46 #     [1, 0, 0, 0, 0, 0, 0, 0]
47 #     [0, 1, 0, 0, 0, 0, 0, 0]
48 #     [0, 0, 1, 0, 0, 0, 0, 0]
49 #     [0, 0, 0, 0, 0, 1, 0, 0]
50 #     [0, 0, 0, 0, 0, 0, 0, 1]
51 #     [0, 0, 0, 0, 0, 0, 1, 0]
52 #     [0, 0, 0, 0, 1, 0, 0, 0]]
53 #Total Cost: 26.365000000000002

```

En affichant le résultat dans un graphe, on remarque que c'est bien la même solution (ordre d'affichage différent) pour le même coût :



## V Applications

Dans les deux premières parties de cette section, nous allons présenter quelques applications de transport optimal à l'aide du module python 'POT' dont nous donnons les informations ci-dessous.

```

1 @article{flamary2021pot,
2   author = {Rémi Flamary and Nicolas Courty and Alexandre Gramfort and
3             Mokhtar Z. Alaya and Aurélie Boisbunon and Stanislas Chambon and Laetitia
4             Chapel and Adrien Corenflos and Kilian Fatras and Nemo Fournier and Léo
5             Gautheron and Nathalie T.H. Gayraud and Hicham Janati and Alain
6             Rakotomamonjy and Ievgen Redko and Antoine Rolet and Antony Schutz and
7             Vivien Seguy and Danica J. Sutherland and Romain Tavenard and Alexander
8             Tong and Titouan Vayer},
9   title = {POT: Python Optimal Transport},
10  journal = {Journal of Machine Learning Research},
11  year = {2021},
12  volume = {22},
13  number = {78},
14  pages = {1-8},
15  url = {http://jmlr.org/papers/v22/20-451.html}
16 }
```

### 1 Généralisation du problème de production et de stockage dans le plan

Nous avons vu précédemment un exemple de transport optimal entre deux mesures de Bernoulli. Nous allons généraliser cette approche avec un nombre  $n$  de points au départ comme à l'arrivée. Les poids seront répartis selon une loi uniforme.

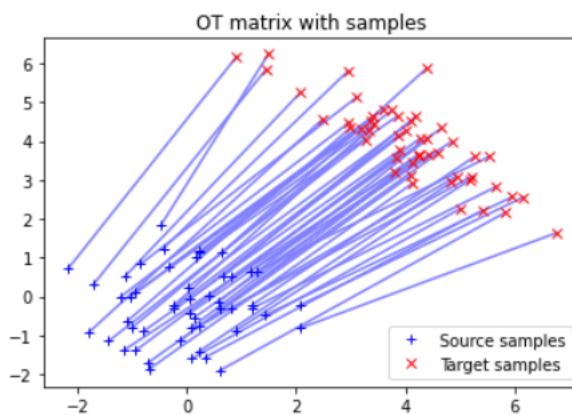


FIGURE 6 – Exemple de couplage pour  $n=50$

Afin de mieux comprendre les étapes de programmation à l'aide du module 'POT', nous allons choisir de travailler avec  $n = 5$ . Dans un premier temps, nous devons placer de manière aléatoire 5 points sources et 5 points cibles dans le plan. Pour cela, il suffit de considérer deux suites de vecteurs Gaussiens indépendants  $(X_n)_n$  et  $(Y_n)_n$  tels que :

$$\forall i \in \mathbb{N}, X_i \sim \mathcal{N}\left(\begin{pmatrix} 4 \\ 4 \end{pmatrix}, Id_2\right) \quad \text{et} \quad Y_i \sim \mathcal{N}\left(\begin{pmatrix} 4 \\ 4 \end{pmatrix}, \begin{pmatrix} 1 & -0.8 \\ -0.8 & 1 \end{pmatrix}\right)$$

**Remarque 15.** *Le choix des valeurs est arbitraire.*

Considérons les vecteurs  $a$  et  $b$  associés aux densités discrètes uniformes. À l'aide de  $Xs$  et de  $Xt$  nous pouvons placer les points dans le plan mais aussi, nous pouvons construire la matrice de coût  $M$ .

```

n = 5 # nb samples

mu_s = np.array([4, 4]) #vecteur espérance source
cov_s = np.array([[1, 0], [0, 1]]) #matrice de covariance source

mu_t = np.array([4, 4]) #vecteur espérance target
cov_t = np.array([[1, -.8], [-.8, 1]]) #matrice de covariance target

xs = ot.datasets.make_2D_samples_gauss(n, mu_s, cov_s) #échantillon de n samples source
xt = ot.datasets.make_2D_samples_gauss(n, mu_t, cov_t) #échantillon de m samples target

a, b = np.ones((n,)) / n, np.ones((n,)) / n # uniform distribution on samples

# loss matrix
M = ot.dist(xs, xt)

print(f'xs=\n{xs}\n')
print(f'tx=\n{xt}')


xs=
[[5.31074803 3.33254731]
 [3.96648218 4.52037479]
 [6.16194062 4.13834407]
 [3.29599119 3.00107665]
 [4.37077978 4.25019839]]


xt=
[[3.22939693 5.45099664]
 [4.7070785 2.98077016]
 [5.1332745 3.73269129]
 [4.62703605 3.56570642]
 [5.59741039 2.75368829]]
```

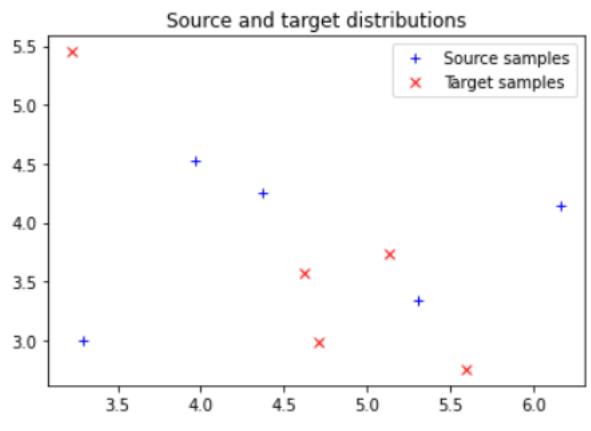


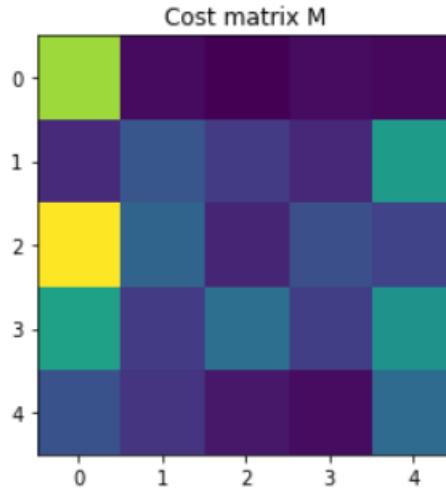
FIGURE 7 – Préparation du problème de transport optimal

```

pl.figure(1)
pl.imshow(M, interpolation='nearest')
pl.title('Cost matrix M')
#Couleurs foncées impliquent petit coût,
#couleurs claires impliquent fort coût

```

Text(0.5, 1.0, 'Cost matrix M')



M # matrice des coûts pour la norme euclidienne au carré

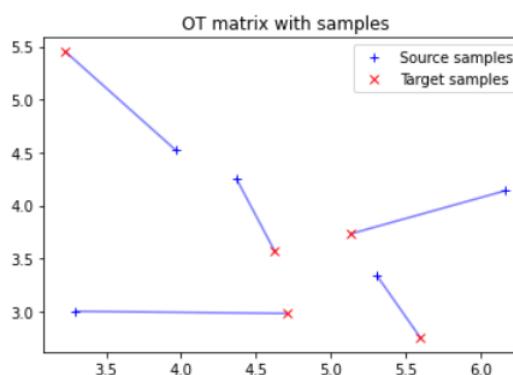
```

array([[ 8.81984999,  0.48816406,  0.19161206,  0.52182524,  0.41725307],
       [ 1.40935169,  2.91886534,  1.98184962,  1.34772313,  5.78110803],
       [10.32286926,  3.45660114,  1.22270816,  2.68384592,  2.23596602],
       [ 6.00654272,  1.99157976,  3.91086996,  2.09048719,  5.35773134],
       [ 2.74467124,  1.72454486,  0.8492118 ,  0.53419654,  3.74416514]])

```

FIGURE 8 – Matrice de coût et visualisation

Nous utilisons la fonction '*emd*' du module 'POT' pour obtenir la matrice de couplage. Cette matrice est solution du problème de transport optimal pour les objets  $a, b$  et  $M$  fixés au préalable. L'explication de l'algorithme permettant de trouver cette matrice  $G_0$  sera pas approfondi dans la suite du rapport.



```

G0 = ot.emd(a, b, M)
#Solves the Earth Movers distance problem and returns the OT matrix
G0 #The OT_matrix qui correspond à la matrice de couplage

```

```

array([[0. , 0. , 0. , 0.2, 0. ],
       [0. , 0.2, 0. , 0. , 0. ],
       [0. , 0. , 0. , 0. , 0.2],
       [0.2, 0. , 0. , 0. , 0. ],
       [0. , 0. , 0.2, 0. , 0. ]])

```

FIGURE 9 – Couplage et associations dans le plan sous les contraintes du transport optimal

**Remarque 16.** Certains axes supplémentaires sont développés en (Annexe 2) :

- Nombre de points cibles et sources différent.
- Densité de probabilité discrète mais non uniforme.

## 2 Transfert de couleurs entre deux images

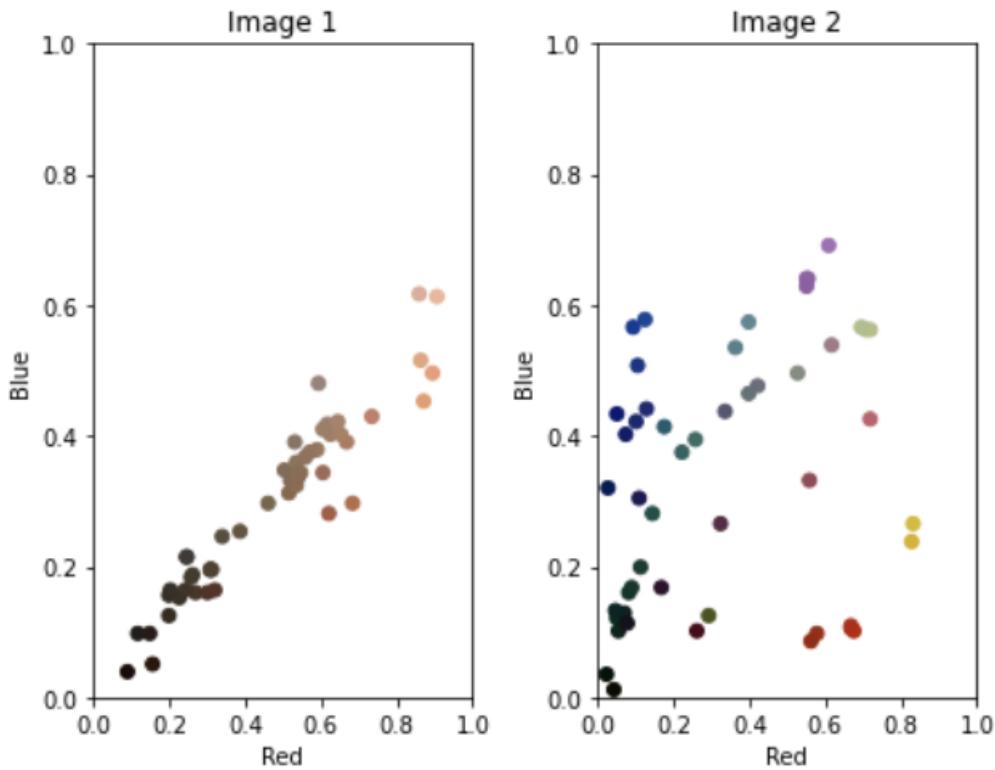
Le transport optimal est né d'un problème d'ingénierie de Monge, il trouve des applications dans le monde réel comme nous allons le voir. Il peut être utilisé pour le traitement d'images, on peut modifier une image source pour que ses couleurs ressemblent à celles d'une image cible. Imaginons un artiste (peintre/plasticien) qui chercherait à reproduire un paysage, un portrait ou une scène de vie à l'aide de ses couleurs fétiches.

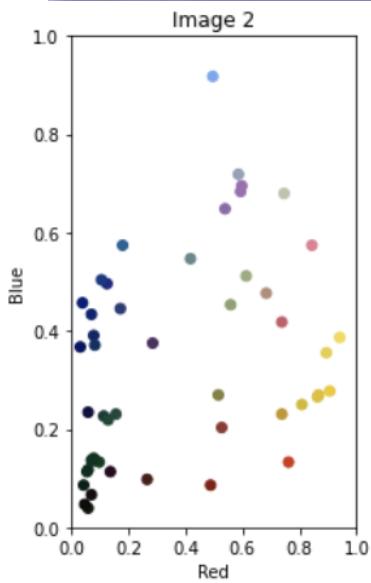
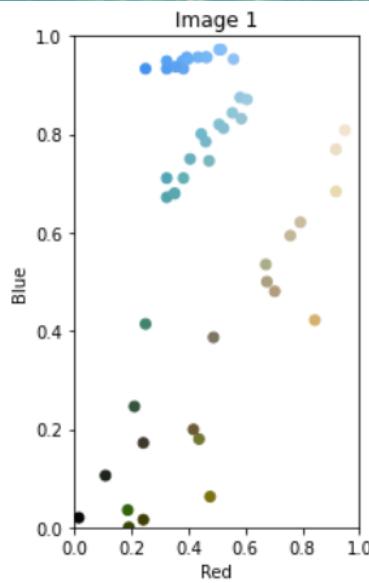


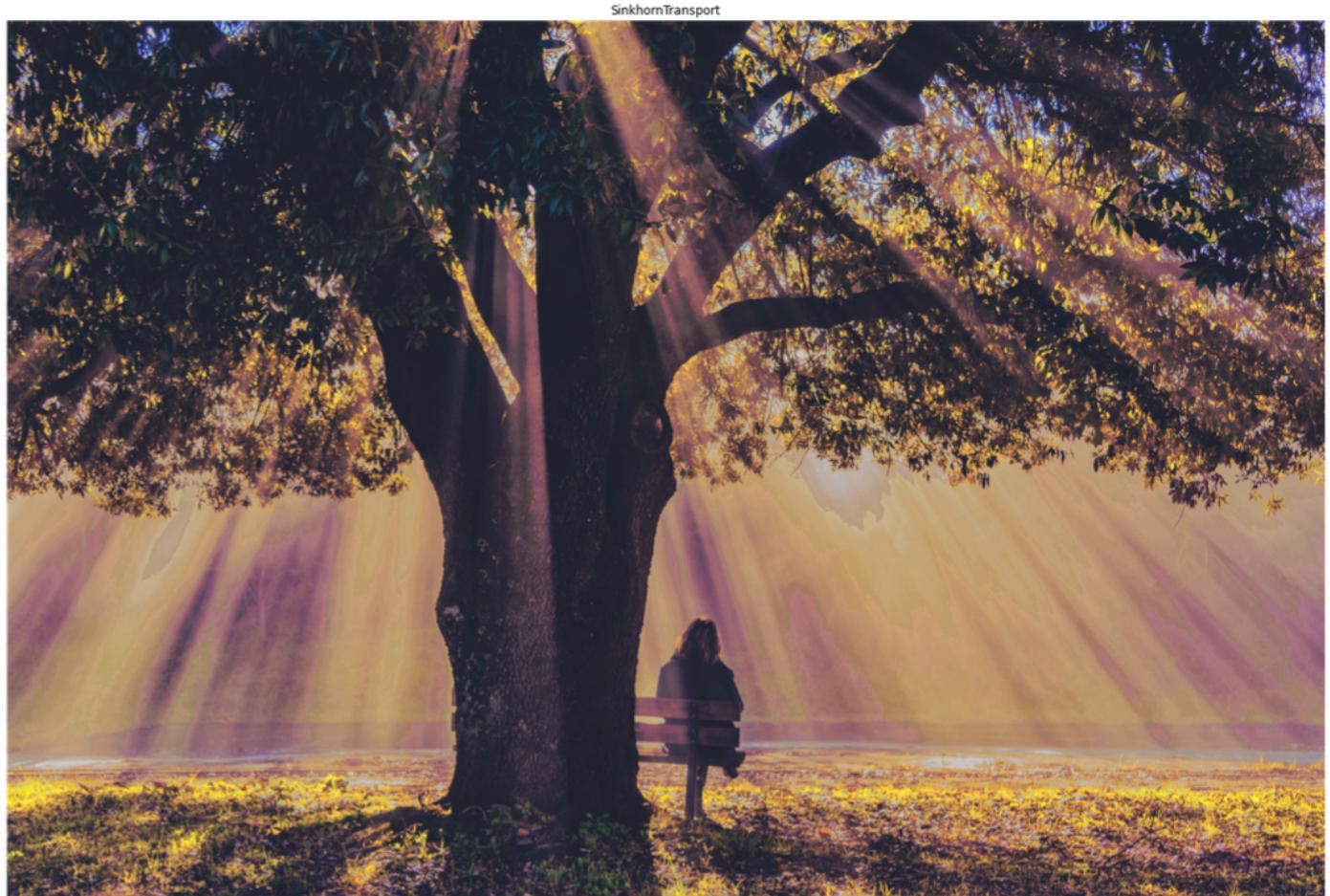
FIGURE 10 – Couleurs fétiches

**Exemple 8.** Un portrait





**Exemple 9.** *Un paysage de Corse*

**Exemple 10.** *Une scène de vie*

## Bibliographie :

- Nathael Gozlan, Paul-Marie Samson, Pierre-André Zitt. Cours de Transport Optimal.
- Gabriel Peyré. Article d'introduction, Images des maths.
- Computational Optimal Transport
- Module POT: Python Optimal Transport
- Wikipédia : semi-continuité
- Openai pour des questions techniques sur python et en L<sup>A</sup>T<sub>E</sub>X