



Transport Optimal : théorie générale et optimisation de la continuité

Mémoire de Stage
École Normale Supérieure de Cachan
2011-2012

Tristan DELEU, Romain DUCASSE, Ludovic SACCHELLI

Sous la direction de
Filippo SANTAMBROGIO
Loïc SIMON

Table des matières

1	D'un résultat théorique de transport à une approximation numérique	1
2	Résumé du cours de Filippo Santambrogio	2
2.1	Introduction	2
2.2	Problème primal, problème dual.	2
2.2.1	Problème de Kantorovich	2
2.2.2	Dualité	2
2.2.3	Cas où $c(x,y)=h(x-y)$, avec h strictement convexe.	3
2.2.4	c -cyclique monotonie	3
2.2.5	Le cas de la dimension 1	4
2.3	Le problème de Monge avec des distances	4
2.3.1	Reformulation de la dualité pour des coûts venants de distances	4
2.3.2	Problème variationnel secondaire pour L^1	4
2.3.3	Problème variationnel secondaire pour L^∞	4
2.4	Problème avec contraintes divergences	4
2.4.1	Points de vues Lagrangiens et Eulériens	4
2.4.2	Problème de Beckmann	5
2.5	Distances de Wasserstein et courbes dans les espaces de Wasserstein.	5
2.5.1	Distance de Wasserstein	5
2.5.2	Topologie induite par la distance de Wasserstein	5
2.5.3	Courbes de Lipschitz dans les espaces de Wasserstein et équation de continuité	5
2.5.4	Géodésiques à vitesse constante dans les espaces de Wasserstein	6
2.5.5	Formulation de Benamou-Brenier et applications numériques	6
3	Un problème de minimisation d'oscillation	7
3.1	Origine du problème et traduction	7
3.2	Minimisation d'un critère de type oscillation dans un plan de transport	8
3.3	Approximation numérique de ce transport optimal	9
4	Algorithmes génétiques	10
4.1	Une première approche	10
4.2	Évolution d'une population	11
4.2.1	Population initiale	11
4.2.2	Nouveaux éléments	11
4.2.3	Concurrence	12
4.2.4	Cas d'erreur	12
5	Approximation numérique d'un plan de transport et minimisation de l'oscillation	14
5.1	Description de l'algorithme	14
5.1.1	Notations et représentation	14
5.1.2	Initialisation	15
5.1.3	Calcul de l'oscillation	16
5.1.4	Reproduction	18
5.1.5	Mutation	18
5.2	Implémentation et exemples	20
5.2.1	Implémentation	20
5.2.2	Exemple	20

A	Détails de l'algorithme	27
A.1	Initialisation	27
A.2	Calcul de F_δ	27
A.3	Mutations infructueuses	29

Résumé

Notre stage d'initiation à la recherche s'est effectué sous la direction de Filippo SANTAMBROGIO et Loïc SIMON. Il a été l'occasion de découvrir la théorie du transport à travers un cours sur le sujet suivi à l'école doctorale d'Orsay. Notre objectif est de tirer partie des conclusions du stage des étudiants de l'année dernière. Les résultats théoriques établis par ces derniers nous servent de base à la constitution d'un algorithme d'approximation d'un transport optimal pour le coût oscillation à échelle δ . Pour se faire nous utilisons un algorithme génétique dont nous avons codé les étapes fondamentales pour être employées avec la librairie C++ *Evolving Objects*.

Chapitre 1

D'un résultat théorique de transport à une approximation numérique

Le stage d'initiation à la recherche que nous avons suivi s'est déroulé sous la direction de Filippo SANTAMBROGIO, spécialiste du transport optimal à l'université doctorale de Paris-Sud. Il s'est dans un premier temps attaché à nous faire découvrir le monde du transport optimal au travers de plusieurs séances de cours destinées aux étudiants en master et post-doctorat de Paris-Sud. le suivi du cours s'est vu accompagnée de la lecture d'ouvrages de référence en théorie de la mesure et d'analyse fonctionnelle.

Dans un deuxième temps, Filippo SANTAMBROGIO nous a fait prendre connaissance du sujet qu'il avait abordé lors du stage de l'année précédente. Ils s'étaient consacrés à la démonstration de résultats théoriques sur les transport à oscillation minimale, notamment en dimension 1. Toutefois, il était apparu l'année passée qu'obtenir de nouveau résultats théoriques semblait difficile (passage en dimension quelconque par exemple). Pour appréhender le problème, Filippo SANTAMBROGIO nous a proposé de chercher à approcher numériquement un tel transport en dimension 1, profitant de l'apport en la matière de Loïc SIMON. Notre objectif était d'écrire un programme capable de calculer un transport optimal solution du problème qui nous intéresse.

Après une phase de recherche pour savoir quelle méthode employer, nous nous sommes dirigé, sous les conseils de Loïc SIMON, vers les algorithmes génétiques. Cette phase nous a demandé de nous approprier à la fois les outils et la théorie de cette méthode. D'une part nous avons lu le livre de Sean LUKE sur la métaheuristique, puis nous nous sommes initiés au C++ afin de profiter de sa rapidité et de pouvoir employer la librairie *Evolving Objects*, nous donnant accès à un cadre adapté aux algorithmes génétiques.

Les semaines qui suivirent furent presque entièrement consacrées à l'écriture du programme, nous demandant de coder les différentes parties nécessaire au fonctionnement de notre algorithme. L'espace dans lequel évoluent les solutions étant particulier, ils nous a fallu parfois nous y prendre à maintes reprises pour réussir à obtenir des résultats satisfaisants, traversant des phases de simple recherche. Mais au final nous avons réussi à trouver des méthodes semblant fonctionner dans les cas les plus problématiques.

Nous avons parallèlement eu la chance d'assister en fin de stage à un colloque de transport optimal organisé à Orsay, Otto (Optimal transport (to) Orsay), et d'observer la pluralité des possibilité de recherche qu'offre le transport optimal aujourd'hui.

Chapitre 2

Résumé du cours de Filippo Santambrogio

2.1 Introduction

En 1871, Monge posa le problème suivant : pour deux densités de probabilité f et g , sur \mathbb{R}^d données, trouver une application $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ transformant la première densité en la seconde, i.e telle que, pour tout mesurable A , $\int_A g = \int_{T^{-1}(A)} f$, et qui minimise la quantité $\int |T(x) - x|f(x)dx$. Physiquement, on peut interpréter cela ainsi : on a des particules réparties selon la densité f , on veut les répartir selon la densité g , en minimisant le déplacement moyen. D'une manière plus générale, on peut écrire ce problème pour des mesures μ et ν , avec la contrainte $T_{\#}\mu = \nu$, et en minimisant la fonctionnelle $\int c(x, T(x))dx$, la fonction c étant une fonction coût.

2.2 Problème primal, problème dual.

2.2.1 Problème de Kantorovich

Kantorovich proposa le problème suivant, que nous noterons (PK) :

Définition 1 Soit μ et ν deux mesures, respectivement sur X et sur Y . Soit $c : X \times Y \rightarrow [0, +\infty]$ une fonction coût. Le problème de Kantorovich consiste à trouver la valeur suivante : $K(\gamma) = \inf\{\int cd\gamma, \gamma \in \Pi(\mu, \nu)\}$, où $\Pi(\mu, \nu)$ est l'ensemble des plans de transports, c'est à dire l'ensemble des probabilités sur $X \times Y$ dont la première marginale est μ , et la seconde est ν .

Ce problème est une généralisation du problème de Monge : en effet, plutôt que de dire qu'une particule en x va en $y = T(x)$, on dit que la particule peut se répartir en plusieurs points. Si c'est le cas, on aura pas de transport. On remarque enfin que si γ est de la forme $(id \times T)_{\#}\mu$, et que T est mesurable de X dans Y , alors T est un transport optimal.

On remarque que l'espace des plans de transports n'est jamais vide : il contient toujours le produit tensoriel des deux mesures.

L'avantage de ce problème est que l'on a de nombreux résultats. Par exemple :

Théorème 1 Si X et Y sont compacts et que c est à valeurs réelles semi-continue inférieurement, alors (PK) à une solution.

2.2.2 Dualité

Pour obtenir plus d'informations sur le problème de Kantorovich, nous allons trouver un problème dual.

Nous allons commencer par exprimer autrement la condition $\gamma \in \Pi(\mu, \nu)$. Remarquons que, si γ est une mesure positive sur $X \times Y$, alors :

$$\sup_{\phi, \psi} \left\{ \int \phi d\mu + \int \psi d\nu - \int (\phi \oplus \psi) d\gamma \right\} = \begin{cases} 0 & \text{si } \gamma \in \Pi(\mu, \nu) \\ +\infty & \text{sinon.} \end{cases}$$

Ainsi, la contrainte est transformée, et on peut écrire le problème ainsi :

$$\inf \left\{ \int c d\gamma + \sup_{\phi, \psi} \left\{ \int \phi d\mu + \int \psi d\nu - \int (\phi \oplus \psi) d\gamma \right\} \right\}$$

Un théorème de Rockafellar permet d'inverser l'inf et le sup. Ainsi, on obtient ainsi le problème dual suivant (de même valeur) :

$$\max \left\{ \int \phi d\mu + \int \psi d\nu \mid \phi \oplus \psi \leq c \right\} \quad (\text{PD})$$

Il est utile ici d'introduire la notion de c -transformée, une généralisation de la transformée de Legendre.

Définition 2 Si $\chi : X \rightarrow \mathbb{R}$, on définit sa c -transformée ainsi : $\chi^c(y) = \inf_x \{c(x, y) - \chi(x)\}$. On définit de façon symétrique la \bar{c} -transformée $\chi^{\bar{c}}(x) = \inf_y \{c(x, y) - \chi(y)\}$, pour une application $\chi : Y \rightarrow \mathbb{R}$. Une fonction est dite \bar{c} -concave si elle est la c -transformée d'une application. On note $\bar{c}C(Y)$ l'ensemble des fonctions \bar{c} -concave.

On définit de même les c -concaves. On peut alors prouver le résultat suivant :

Proposition 1 Si X et Y sont compacts et c est continue, il existe un couple solution à (PD) (ϕ, ψ) , où $\phi \in cC(X), \psi \in \bar{c}C(Y)$ et $\phi = \psi^c$. Ainsi, on a

$$\min(PK) = \max_{\phi \in cC(X)} \left\{ \int \phi d\mu + \int \phi^c d\nu \right\}$$

La fonction ϕ solution est appelée potentiel de Kantorovich.

2.2.3 Cas où $c(x, y) = h(x - y)$, avec h strictement convexe.

On se place dans le cas où $X = Y$ et $c(x, y) = h(x - y)$, avec h strictement convexe. Alors, on peut démontrer le résultat suivant :

Théorème 2 Si μ et ν sont deux mesures de probabilités sur un compact Ω de \mathbb{R} , avec μ absolument continue et $\partial\Omega$ négligeable. Alors, il existe un unique transport optimal, de la forme $(id \times T)_{\#}\mu$. De plus, il existe un unique potentiel de Kantorovich, il sera tel que :

$$T(x) = x - (\nabla h)^{-1}(\nabla \phi(x))$$

Pour le cas quadratique, on peut écrire facilement T comme le gradient d'une fonction convexe.

2.2.4 c -cyclique monotonie

Si on a une fonction $c : \Omega \times \Omega \rightarrow \bar{\mathbb{R}}$, un ensemble $\Gamma \subseteq \Omega \times \Omega$ est dit c -cycliquement monotone si pour tout $k \in \mathbb{N}$, pour tout permutation σ , et pour toute famille $(x_1, y_1), \dots, (x_k, y_k)$ on a :

$$\sum c(x_i, y_i) \leq \sum c(x_i, y_{\sigma(i)})$$

On peut alors prouver le résultat suivant :

Théorème 3 Si γ est un plan de transport optimal pour le coût c continu, alors le support de γ est c -cycliquement monotone.

On peut, en utilisant cette notion de c -cyclique monotonie, montrer le théorème suivant :

Théorème 4 Si $c : X \times Y \rightarrow \bar{\mathbb{R}}$ est continue, que X et Y sont compacts, le problème dual admet une solution et $(PD) = (PK)$.

2.2.5 Le cas de la dimension 1

La c -cyclique monotonie permet également d'établir des résultats intéressants en dimension 1. On peut commencer par remarquer que, pour deux probabilités sur \mathbb{R} , μ (prise non atomique) et ν , il existe une unique application croissante T telle que $T_{\#}\mu = \nu$. (On a $T = G^{-1}F$, où F est la fonction de répartition de μ et G est la fonction de répartition de ν .) Dans le cas quadratique, comme le transport est censé être le gradient d'une fonction convexe, le transport est croissant. Le T exhibé (que l'on notera T_{mon}) ici est donc le transport optimal dans le cas quadratique en dimension 1. L'utilisation de résultats sur la c -cyclique monotonie permet donc de montrer le théorème suivant :

Théorème 5 *Soit $h : \mathbb{R} \rightarrow \mathbb{R}$ strictement convexe, μ, ν des probabilités sur \mathbb{R} avec μ non atomique. Alors, le problème de transport optimal avec le coût $c(x,y) = h(x-y)$ a pour unique plan de transport optimal $(id \times T_{mon})_{\#}\mu$.*

2.3 Le problème de Monge avec des distances

2.3.1 Reformulation de la dualité pour des coûts venants de distances

On suppose ici que $c(x,y) = d(x,y)$, où d est une distance. La symétrie de la distance permet de ne pas distinguer la \bar{c} -transformation et la c -transformation. On obtient alors la caractérisation suivante des fonctions c -concaves :

Proposition 2 *Si $c : X \times Y \rightarrow \mathbb{R}$ est une distance, alors une fonction $u : X \rightarrow \mathbb{R}$ est c -concave si, et seulement si, elle est 1-Lipschitzienne, pour la distance c . On note Lip_1 , l'ensemble de ces fonctions. On a de plus, pour tout $u \in Lip_1$, $u^c = -u$.*

On peut donc reformuler le problème de Kantorovich ainsi (en utilisant la formulation duale) :

$$\min \left\{ \int d(x,y)d\gamma, \gamma \in \prod(\mu, \nu) \right\} = \max \left\{ \int ud(\mu - \nu), u \in Lip_1 \right\}$$

2.3.2 Problème variationnel secondaire pour L^1

Appellons $O(\mu, \nu)$ l'ensemble des plans de transports optimaux pour le coût $|x-y|$. On notera par la suite c_p la fonctionnelle qui, à γ associe la quantité $\int |x-y|^p d\gamma$. On notera m_p son minimum sur $\prod(\mu, \nu)$. Ainsi, on a $O(\mu, \nu) = \operatorname{argmin}_{\gamma \in \prod(\mu, \nu)} c_1(\gamma)$. Pour la topologie faible, $O(\mu, \nu)$ est compact.

Les minimiseurs de c_1 ne proviennent pas tous d'un transport. On va donc sélectionner un minimiseur particulier, en considérant le problème suivant : $\min\{c_2(\gamma), \gamma \in O(\mu, \nu)\}$. Des considérations de compacité topologiques nous donnent l'existence d'un plan de transport $\bar{\gamma}$. On peut montrer que ce plan de transport vient d'un transport, en utilisant des résultats de monotonie.

2.3.3 Problème variationnel secondaire pour L^∞

On définit ici $c_\infty(\gamma) = \|x-y\|_{L^\infty(\gamma)}$, et on veut le minimiser. (il s'agit du déplacement maximal). On peut montrer que $c_p(\gamma)^{1/p} \rightarrow c_\infty(\gamma)$. On en déduit que c_∞ admet un minimiseur sur $\prod(\mu, \nu)$. On considère donc le problème $\min\{c_\infty(\gamma), \gamma \in \prod(\mu, \nu)\}$, on voudrait montrer qu'un minimiseur provient d'un transport optimal. Comme on n'a pas unicité du minimiseur, on définit un problème variationnel secondaire comme avant : on pose $O_\infty(\mu, \nu) = \operatorname{argmin}_{\gamma \in \prod(\mu, \nu)} c_\infty(\gamma)$, et le problème variationnel secondaire est défini ainsi : $\min\{c_2(\gamma), \gamma \in O_\infty(\mu, \nu)\}$. Là encore, des considérations topologiques nous donnent l'existence d'un plan de transport optimal. On peut montrer que ce plan de transport vient d'un transport.

2.4 Problème avec contraintes divergences

2.4.1 Points de vues Lagrangiens et Eulériens

En mécanique des fluides, il existe deux façons de décrire un mouvement : le point de vue Lagrangien et le point de vue Eulérien. Le formalisme Lagrangien consiste à repérer chaque particule et à regarder, à chaque instant, sa position et/ou sa vitesse. Ainsi, on peut se donner des fonctions $y_x(t)$, qui donnent, pour une particule initialement en x sa position au temps t . Si on considère seulement le point de départ et le point final, le comportement d'une particule est donné par ses positions initiales et finales. Une

mesure sur des couples (position initiale, position finale) est un plan de transport. Ainsi, le problème de Kantorovich s'exprime naturellement en coordonnées Lagrangiennes. Le point de vue Eulérien consiste à décrire ce qui se passe au point x au moment t . Ainsi, dans le formalisme Eulérien, on se donne souvent deux variables (dans le cas d'un problème dynamique) : une densité $\rho(x, t)$ et une vitesse $v(x, t)$. On peut décrire le mouvements d'une population de particules se déplaçant selon ce champ de vitesses. Pour une densité ρ_0 donné, la position d'une particule initialement en x sera donnée par la résolution du système

$$\begin{cases} y'_x(t) = v(t, y_x(t)) \\ y_x(0) = x \end{cases}$$

On peut définir le transport $T_t(x) = y_x(t)$, et on pose $\rho_t = (T_t)_\# \rho_0$. Alors $\rho_t(x)$ et v_t forment un couple solution de l'équation de continuité $\partial_t \rho_t + \nabla \cdot (\rho_t v_t) = 0$.

On pourrait faire l'interprétation physique suivante : $\nabla \cdot (\rho_t v_t)$ représente un excès de masse injectée. Si on considère que les particules arrivent avec une certaine densité, et partent avec une autre densité, on peut écrire : $\nabla \cdot (\rho v) = \mu - \nu$ (dans le cas stationnaire).

2.4.2 Problème de Beckmann

On définit le problème de Beckmann

$$\min \left\{ \int |v(x)| dx, v : \Omega \rightarrow \mathbb{R}^n, \nabla \cdot v = \mu - \nu \right\} \quad (\text{PB})$$

Il s'agit de minimiser la quantité de mouvement en connaissant l'excès de masse. Comme avant, on peut transformer la condition pour avoir une formulation duale. Le problème dual obtenu est alors : $\sup_{\phi, |\nabla \cdot \phi| \leq 1} \int \phi d(\mu - \nu)$. On retrouve le problème dual de Kantorovich, en effet $|\nabla \phi| \leq 1 \Leftrightarrow \phi \in Lip_1$, on a donc équivalence entre le problème de Beckmann et celui de Kantorovich.

2.5 Distances de Wasserstein et courbes dans les espaces de Wasserstein.

2.5.1 Distance de Wasserstein

Pour $\Omega \subset \mathbb{R}^n$ et $p \geq 1$, on pose

$$P_p(\Omega) = \left\{ \mu \in P(\Omega), \int |x|^p d\mu < +\infty \right\}$$

On définira les distances dites "de Wasserstein" sur de tels ensembles. Si Ω est borné, $P_p(\Omega) = P(\Omega)$.

Si $\mu, \nu \in P_p(\Omega)$, on pose

$$W_p(\mu, \nu) = \inf \left\{ \int |x - y|^p d\gamma, \gamma \in \Pi(\mu, \nu) \right\}^{1/p}$$

Cette quantité définit une distance, la distance de Wasserstein.

2.5.2 Topologie induite par la distance de Wasserstein

Si on est sur un compact Ω , on peut faire le lien entre topologie faible et topologie induite par la distance de Wasserstein : en effet, on peut montrer que, quel que soit $p \geq 1$, $\mu_n \rightharpoonup \mu$ si, et seulement si, $W_p(\mu_n, \mu) \rightarrow 0$.

On peut raffiner ce résultat ainsi : quel que soit Ω , $p \geq 1$, $\mu_n \rightharpoonup \mu$ et $\int |x|^p d\mu_n \rightarrow \int |x|^p d\mu$ si, et seulement si, $W_p(\mu_n, \mu) \rightarrow 0$.

2.5.3 Courbes de Lipschitz dans les espaces de Wasserstein et équation de continuité

Pour une courbe de Lipschitz (i.e dont le paramétrage est donné par une application Lipschitzienne), on peut définir la dérivée métrique ainsi :

$$|\omega'(t)| = \lim_{h \rightarrow 0} \frac{d(\omega(t+h), \omega(t))}{|h|}$$

On peut faire le lien entre les courbes de Lipschitz dans $P_p(\Omega)$ muni de W_p et les solutions de l'équation de continuité $\partial_t \mu_t + \nabla \cdot (\mu_t v_t) = 0$. (Au sens des distributions.). On peut établir le résultat suivant :

Théorème 6 Soit $(\mu_t)_{t \in [0,1]}$ une courbe de Lipschitz pour la distance W_p ($p > 1$). Alors, pour presque tout $t \in [0, 1]$, il existe un champ de vecteurs $v_t \in L^p(\mu_t, \mathbb{R}^n)$ tel que $\partial_t \mu_t + \nabla \cdot (\mu_t v_t) = 0$ soit vérifiée et, pour presque tout t , on a $\|v_t\|_{L^p(\mu_t)} \leq |\mu'| (t)$.

2.5.4 Géodésiques à vitesse constante dans les espaces de Wasserstein

Une courbe ω est une géodésique à vitesse constante entre $\omega(0)$ et $\omega(1)$ si $d(\omega(t), \omega(s)) = |t - s|d(\omega(0), \omega(1))$ pour tout (t, s) . Une telle courbe est nécessairement une géodésique. Dans les espaces de Wasserstein, on peut faire le lien entre les géodésiques à vitesse constante et les plans de transport optimal via le théorème suivant :

Théorème 7 Soit μ_t une géodésique à vitesse constante entre μ et ν . Soit $p > 1$. Alors, il existe un plan de transport optimal $\gamma \in \Pi(\mu, \nu)$, tel que, pour tout t , on ait : $\mu_t = (\pi_t)_{\#} \gamma$, où on définit $\pi_t : \Omega \times \Omega \rightarrow \Omega$ ainsi : $\pi_t(x, y) = (1 - t)x + ty$.

On peut également montrer que réciproquement $\forall \gamma$ optimal, la courbe $\pi_t \# \gamma$ est une géodésique à vitesse constante.

2.5.5 Formulation de Benamou-Brenier et applications numériques

On a vu que rechercher un transport optimal pour $|x - y|^p$ revient à chercher une géodésique à vitesse constante dans W_p . On peut montrer que que les géodésiques à vitesse constante peuvent se trouver en minimisant $\int |\mu'| (t)^p dt$. On peut montrer que dans un espace de Wasserstein on a $|\mu'| (t)^p dt = \int |v_t|^p d\mu_t$.

Ainsi, résoudre $\min \int_{[0,1]} \int_{\Omega} |v_t|^p d\mu_t dt$, sous les conditions $\partial_t \mu_t + \nabla(\mu_t v_t) = 0$, $\mu_0 = \mu$, $\mu_1 = \nu$, permet de trouver le transport optimal entre deux mesures. Ce problème a des contraintes non-linéaires, mais on peut le transformer, et obtenir un problème convexe. Il suffit de faire le changement de variables $(\mu_t, v_t) \rightarrow (\mu_t, E_t = \mu_t v_t)$. Le problème devient : $\min \int_{[0,1]} \int_{\Omega} f_p(\mu_t, v_t) dt$, sous les conditions $\partial_t \mu_t + \nabla(E_t) = 0$, $\mu_0 = \mu$, $\mu_1 = \nu$. Les contraintes sont maintenant linéaires, et le problème est convexe, mais pas strictement convexe. Un algorithme de descente de gradient ne sera pas donc forcément efficace. Il existe un algorithme permettant de résoudre cela, cet algorithme est basé sur la réécriture des contraintes au sens des distributions.

Chapitre 3

Un problème de minimisation d'oscillation

3.1 Origine du problème et traduction

Certaines applications sur téléphone portable proposent de connaître la position de ses amis via la géolocalisation des smartphones. Le serveur gérant ces informations possède donc potentiellement la totalité des positions dans sa mémoire, ce qui peut poser des problèmes de sécurité. Une approche naturelle serait de donner des données faussées, mais permettant néanmoins au serveur de calculer les distances. Par exemple une rotation semblerait faire l'affaire, pourtant retrouver quelques grandes villes grâce à la forte densité de téléphones portables qui s'y trouvent permet de retrouver la position originale de la distribution.

L'idée serait donc de s'inspirer de la théorie du transport pour en quelque sorte crypter ces données. Le but est de transformer une distribution quelconque (celle des téléphones portables par exemple) en une distribution uniforme, masquant ainsi la position réelle des utilisateurs au serveur. Pour pouvoir continuer de retrouver ses amis, nous voudrions imposer la condition selon laquelle deux téléphones proche sur la distribution originale soient proches également dans la nouvelle distribution uniforme. Ceci garantirait que tout couple d'amis identifié comme loin dans la nouvelle distribution le soit vraiment. Pour cela, si le but est d'identifier les couples à distance inférieure ou égale à δ , il faut introduire la quantité $F_\delta(T)$ l'oscillation du transport T

$$F_\delta(T) = \sup_{\|x-y\| \leq \delta} \|T(x) - T(y)\|$$

De cette manière on sait que si la fausse distance est supérieure à F_δ , la distance réelle est supérieure à δ . Il est donc intéressant de minimiser F_δ , pour en augmenter la précision.

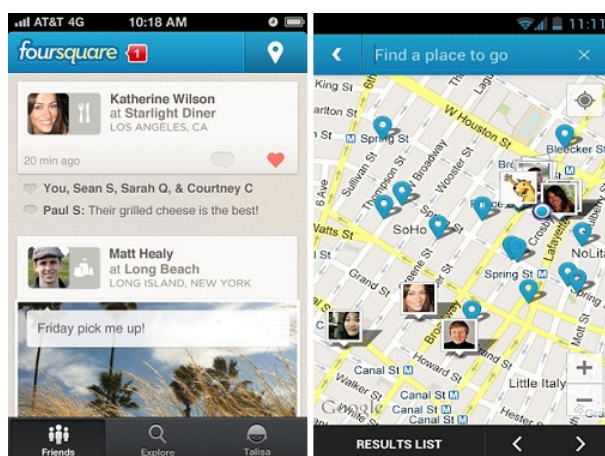


FIGURE 3.1 – Foursquare est un réseau social disponible sur mobile ayant la particularité de proposer aux utilisateurs de connaître la position de leurs amis

3.2 Minimisation d'un critère de type oscillation dans un plan de transport

Nous nous intéressons à minimiser parmi les transports la valeur de F_δ . Les étudiants de l'année dernière se sont attachés à démontrer des résultats théoriques sur le transport optimal, dont certaines propriétés qui nous seront utiles.

Leur cheminement est assez proche des méthodes vues en cours, la première étape est une relaxation du problème en passant au plan de transport.

On veut minimiser

$$F_\delta(T) = \sup_{\|x-y\| \leq \delta} \|T(x) - T(y)\|$$

On s'intéresse naturellement aux éléments du plan de transport

$$\Pi(\mu, \nu) = \{\gamma \in \mathbb{P}(\Omega \times \Omega) : \pi_1 \# \gamma = \mu, \pi_2 \# \gamma = \nu\}$$

Le critère devenant alors :

$$\vartheta_\delta : \gamma \in \Pi(\mu, \nu) \mapsto \sup_{\substack{\|x-x'\| \leq \delta \\ (x,y), (x',y') \in \text{Supp}(\gamma)}} \|y - y'\|$$

Théorème 8 *Si $\mu, \nu \in \mathbb{P}(\Omega)$, $\delta > 0$, il existe $\gamma \in \Pi(\mu, \nu)$ minimisant le coût ϑ_δ .*

Il est ensuite possible de trouver un transport dans le cas réel et compact. On cherche alors un transport optimal dans un fermé Γ contenant le support d'un γ optimal pour ϑ_δ , à la fois suffisamment grand pour que l'on puisse y trouver le graphe d'un tel transport, et assez proche pour que les éléments qui s'y trouvent conservent la minimalité du coût ϑ_δ . On délimite Γ par les fonctions

$$f : x \mapsto \inf \{y : (x, y) \in \Gamma\}$$

$$g : x \mapsto \sup \{y : (x, y) \in \Gamma\}$$

On parvient trouver un γ optimal contenu dans une bande délimitée par des f et g monotones par morceaux, avec la même monotonie sur chacun des intervalles, de sorte que Γ apparaisse comme une bande monotone par morceaux. C'est alors dans cette bande que l'on parvient à construire un transport également monotone par morceaux

Théorème 9 *Soient Ω un compact de \mathbb{R} , μ et ν des mesures de probabilités sur Ω . Si μ est sans atome, alors il existe un transport optimal $T \in T(\mu, \nu)$ pour le coût F_δ où*

$$F_\delta : T \mapsto \sup_{\|x-y\| \leq \delta} \|T(x) - T(y)\|$$

La preuve de ce théorème est fondée sur la construction d'un transport optimal, cela nous permet d'exhiber quelques propriétés utiles sur celui-ci. On apprend en effet qu'il est monotone par morceaux, et hormis l'un des intervalles de monotonie aux extrémités du support de μ , la longueur de ces intervalles est minorée par 2δ . On observe alors quelque chose de comparable à la figure 3.2

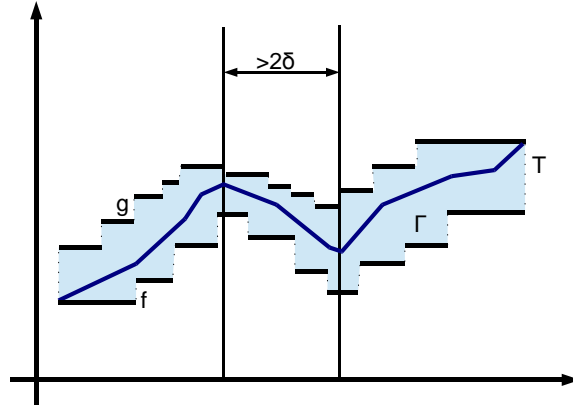


FIGURE 3.2 – Le transport est monotone par morceaux

3.3 Approximation numérique de ce transport optimal

Notre projet est d'approcher numériquement le transport optimal pour ce coût. Comme il s'agissait d'un problème de minimisation, une approche naturelle aurait été une descente de gradient dans $T(\mu, \nu)$, mais nous avons assez rapidement écarté cette piste. D'une part les contraintes de l'ensemble des transport $T(\mu, \nu)$ rend le calcul d'un gradient difficile et la forme que prend $F_\delta(T) = \sup_{\|x-y\| \leq \delta} \|T(x) - T(y)\|$ le rend simplement impossible puisqu'il s'agit d'un supremum. D'autre part nous ignorons la structure de l'espace $T(\mu, \nu)$, et une descente de gradient ne nous permet pas de trouver un minimum mais simplement un minimum local, ne répondant pas au problème fixé et se trouvant peut-être en grand nombre.

Sous les conseils de Loïc SIMON, nous nous sommes donc dirigés vers les algorithmes génétiques. Il s'agit d'algorithmes s'inspirant des dynamiques de population dans la nature pour trouver des solutions. À partir d'une population initiale et en s'assurant que certaines conditions soient respectées aux différentes étapes, ceux-ci nous permettent d'explorer l'ensemble des transport simplement sans être bloqué par des minima locaux, tout en s'assurant un contrôle sur la population afin que les solutions respectent les propriétés théoriques obtenues précédemment.

Chapitre 4

Algorithmes génétiques

4.1 Une première approche

L'algorithme que nous devons employer doit pouvoir optimiser une quantité dont on ne peut obtenir un gradient, comme notre F_δ , tout en s'adaptant aux contraintes du problème, ce qui peut rendre l'espace des solutions admissibles très complexe, comme $T(\mu, \nu)$. On peut alors chercher un algorithme fondé sur l'idée que l'on peut trouver des éléments de l'ensemble des solutions ($T(\mu, \nu)$), à l'aide des éléments déjà en notre possession. Lorsque nous avons un élément de notre ensemble, il faut savoir ce que nous souhaitons en faire pour trouver un élément qui semble meilleur (ayant un coût plus faible par exemple).

Sachant qu'il ne nous est pas possible de savoir quelle est la bonne direction selon laquelle se diriger dans notre espace, contrairement à une descente de gradient, il nous faut trouver un autre moyen de la connaître. Une solution est de chercher à améliorer le meilleur élément trouvé jusque là, mais en utilisant de l'aléatoire, en pratiquant une sorte de sondage dans le voisinage de notre solution. Il s'agit d'une méthode de type *Hill Climbing*, qui imite une descente de gradient. Le but est de perturber notre élément pour qu'il s'écarte de sa position, de manière aléatoire permettant d'explorer, pour éventuellement trouver une meilleure solution (cf figure 4.1).

Le défaut principal de cet algorithme est finalement le même que dans une descente de gradient, bien qu'il permette de trouver rapidement un extremum, il ne peut distinguer un extremum d'un extremum local, et une fois un tel point atteint, il n'est pas possible de s'en écarter. L'exploitation des résultats obtenus auparavant est trop forte et il est impossible d'explorer suffisamment pour se débloquent, puisque cela demanderait dans un premier temps de détériorer notre solution. Ce phénomène est illustré dans la deuxième sous figure de la figure 4.1.

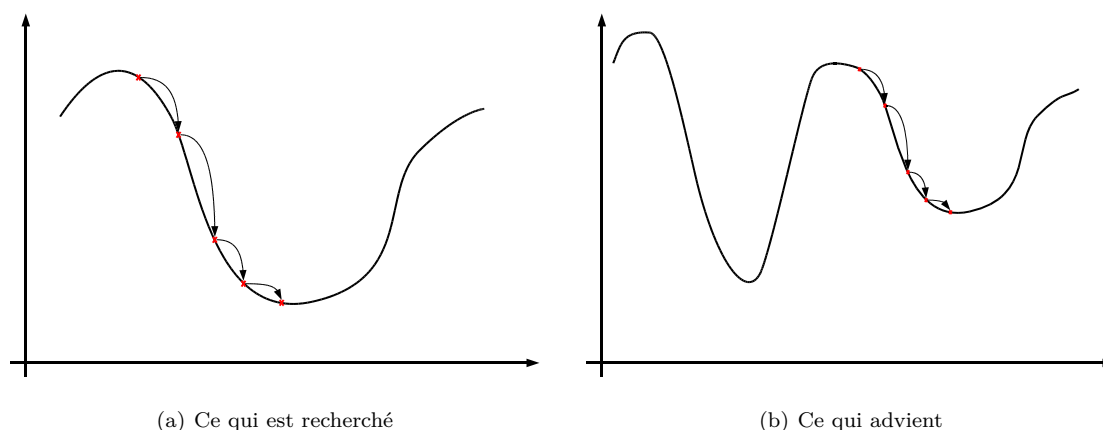


FIGURE 4.1 – Le *Hill Climbing*

4.2 Évolution d'une population

Il nous faut donc un algorithme permettant d'optimiser une quantité tout en évitant d'être piégé par des extrema locaux dont nous ignorons l'occurrence. Les algorithmes génétiques ont l'avantage de proposer à la fois exploration et exploitation, ce qui correspond mieux à nos besoins. Comme leur nom le suggère, ces algorithmes s'inspirent des dynamiques de population et notamment de la transmission d'allèles d'une génération à la suivante, les noms des différentes étapes sont d'ailleurs inspirés de la biologie, pour permettre de cerner rapidement leur utilité. Le principe est de gérer une population d'individus appartenant à l'ensemble des solutions admissibles, et de leur faire traverser différentes étapes de sélection et de création d'individus pour obtenir une nouvelle génération. En effet à partir d'une population initiale, une concurrence entre les différents éléments permet de sélectionner des géniteurs, qui vont servir à la reproduction. Les éléments obtenus seront mutés pour former la population des enfants. Ensuite une étape de survie permet de former la nouvelle génération qui servira de base à la nouvelle itération de l'algorithme. Les étapes de selections sont fondées sur la comparaison d'une quantité qualifiant chaque solution, appelée *fitness*, qui correspond ici à la quantité F_δ à minimiser. Ces étapes sont résumées dans le schéma explicatif 4.2, et observables dans la figure 4.3(a).

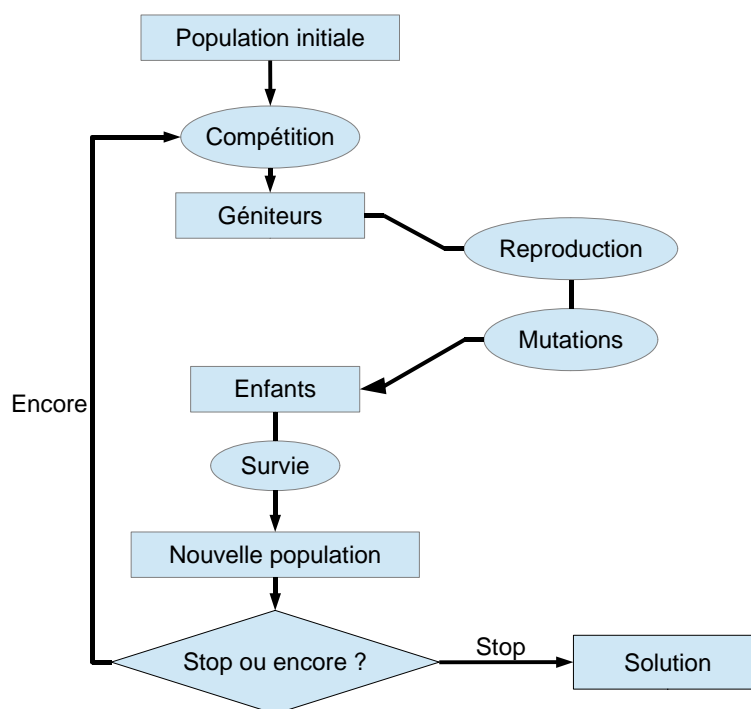


FIGURE 4.2 – Un schéma résumé des étapes de l'algorithme génétique

4.2.1 Population initiale

Les différentes étapes de l'algorithme visant à créer de nouveaux éléments pour notre population sont destinées à les former à partir de la génération précédente. Ces étapes vont conserver les caractéristiques faisant d'un élément une solution admissible (la fonction obtenue par reproduction de deux transports est encore un transport par exemple). Il nous faut cependant fabriquer de toute pièce une population qui sera prête à évoluer. Cette étape se fait donc en dehors de la boucle évolutive, et selon que le programme mette ou non l'accent sur l'exploration, il faudra chercher à la diversifier suffisamment.

4.2.2 Nouveaux éléments

Il s'agit d'une étape d'exploration, visant à former une population pour remplacer la précédente, en essayant d'améliorer les bons éléments tout en continuant d'explorer de nouvelles voies. La création de

nouveaux éléments ne demande que de former n enfants à partir de m parents, mais par analogie avec la nature, on se concentre sur deux méthodes, la reproduction et la mutation.

La reproduction consiste à prendre deux éléments pour en faire un troisième, on appelle également cette étape le mélange. L'exploration se fait donc dans le voisinage des deux éléments, de manière barycentrique par exemple. Elle permet de conserver des caractéristiques des deux parents, ce qui favorise une bonne qualité pour le fils si les parents sont bons. Si un extremum local est dans le voisinage de la population, on peut ainsi s'en rapprocher au fil des générations.

La mutation consiste quant à elle à perturber un élément pour le déplacer dans une direction aléatoire, on évite ainsi de se retrouver piégé par un extremum local, contrairement à la reproduction. Il permet donc de se diriger vers des parties encore inexplorées, ce qui n'est pas vraiment le cas avec la reproduction.

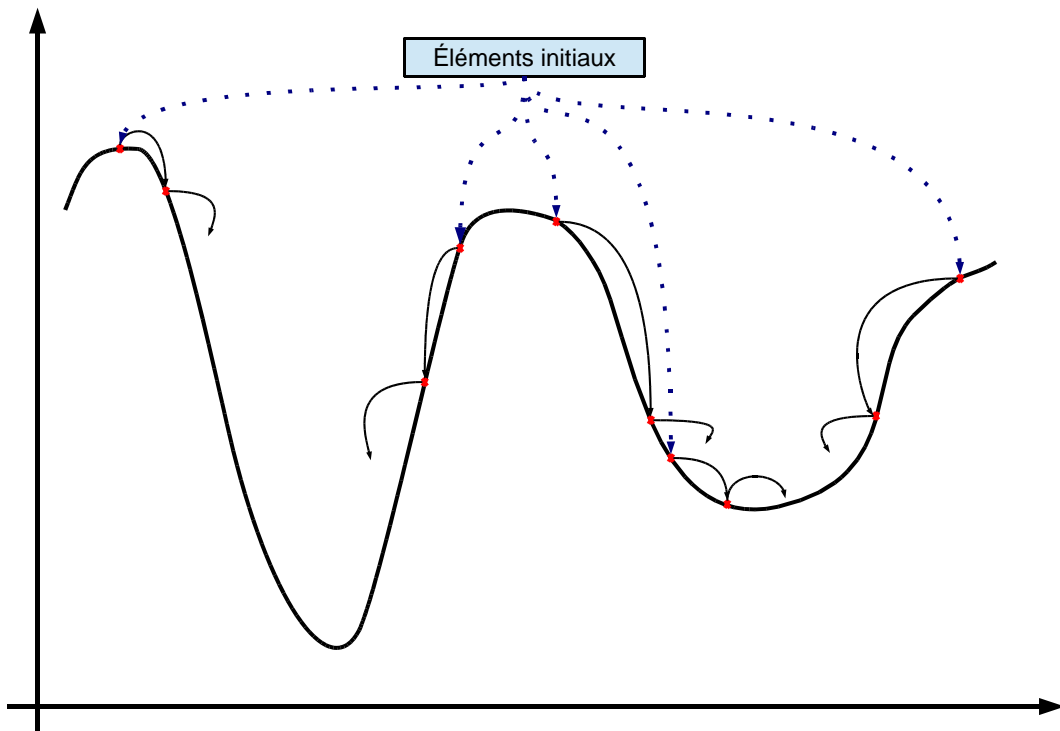
4.2.3 Concurrence

Par analogie avec la transmission d'allèle, les moins bons éléments du groupe font place à la génération suivante. Pour cela, il faut s'appuyer sur deux éléments, le calcul de la qualité (*fitness*) d'un élément, et la compétition entre les éléments d'une population.

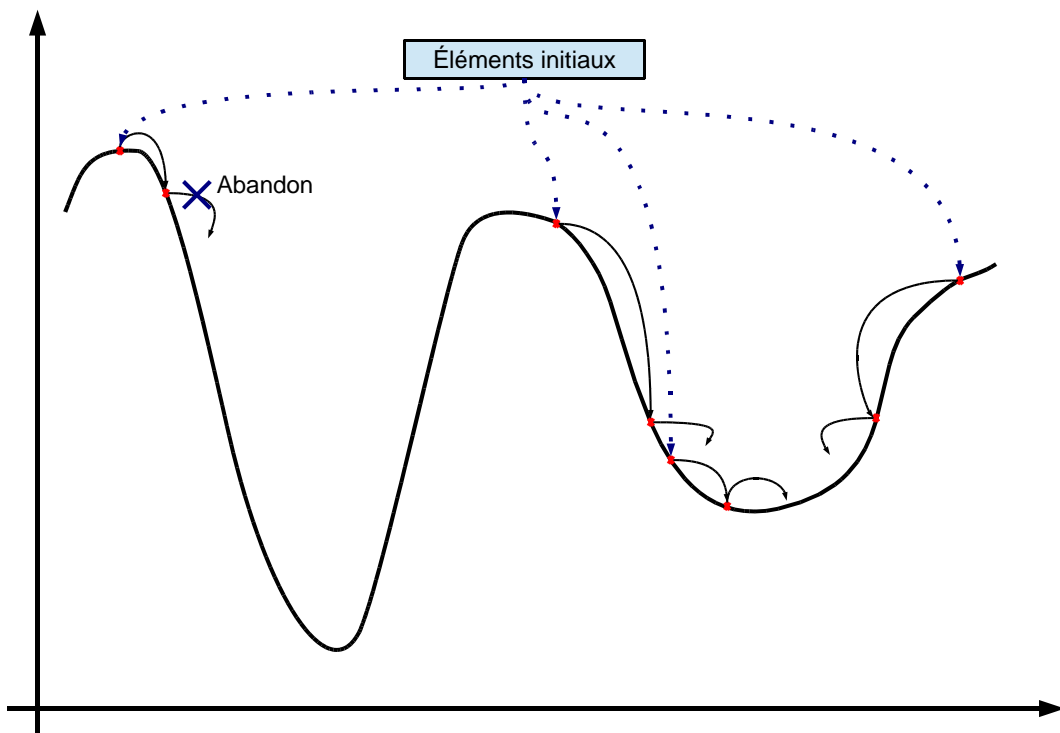
La qualité d'un élément revient à obtenir une grandeur permettant de comparer les éléments entre eux. Typiquement, lorsqu'il s'agit de minimiser une grandeur dans un problème d'optimisation, c'est cette grandeur que l'on choisit, ici F_δ . C'est ensuite en comparant les qualités de plusieurs éléments que l'on peut déterminer lesquels sont les plus adaptés. Il faut cependant réussir à calculer cette grandeur avec une complexité assez faible puisqu'elle est nécessaire pour chaque élément de la population. La compétition intervient dans deux étapes de l'algorithme, la compétition pour la reproduction et la survie. Parmi les μ éléments de la génération actuelle, seuls certains vont se reproduire, on choisit ces derniers grâce à une compétition de type tournoi : on forme des paires d'individus puis on les compare pour ne conserver que le meilleur des deux, il est choisi comme reproducteur. Une fois sélectionnés les géniteurs et formé λ enfants, il faut déterminer lesquels formeront la nouvelle population qui servira de base à la prochaine itération de l'algorithme. Plusieurs choix s'offrent à nous, nous pouvons par exemple choisir les μ meilleurs individus parmi les $\lambda + \mu$ éléments en notre possession (mode de sélection $\mu + \lambda$), ou si $\mu < \lambda$ on sélectionne μ éléments parmi les λ enfants au moyen d'une troncation déterministe (les μ meilleurs *fitness*) ou par des tournois (mode de sélection, il s'agit du mode de sélection (μ, λ)).

4.2.4 Cas d'erreur

Le bon fonctionnement de notre algorithme dépend de l'équilibre entre ces différentes étapes, en effet, rien n'empêche que la population converge vers un extremum local. Il peut même arriver qu'une branche avantageuse mais représentant une faible proportion d'individus soit suppléée par une autre partie de la population, convergeant massivement vers un extremum local. Ce cas peut être observé dans la figure 4.3(b).



(a) Des éléments sont répartis parmi l'ensemble des solutions, puis ils évoluent vers de minima



(b) Le risque est l'abandon d'une branche avantageuse parce que trop éloignée des autres

FIGURE 4.3 – L'algorithme génétique

Chapitre 5

Approximation numérique d'un plan de transport et minimisation de l'oscillation

5.1 Description de l'algorithme

5.1.1 Notations et représentation

Comme dans le problème classique de Monge décrit au chapitre 1, on note μ une mesure de départ sur un compact I de \mathbb{R} et ν une mesure d'arrivée sur un compact J de \mathbb{R} , toutes deux mesures de probabilité et absolument continues par rapport à la mesure de Lebesgue L . Pour que ces mesures soient exploitables numériquement, on se limite à des mesures dont la densité est constante par morceaux; ainsi, on note t_0, t_1, \dots, t_{n-1} une partition de I sur laquelle μ est constante par morceaux de densité f et y_0, y_1, \dots, y_{n-1} une partition de J sur laquelle ν est constante par morceaux de densité g . On appelle pas_μ (resp. pas_ν) le pas de discrétisation de μ (resp. ν), définie par :

$$\text{pas}_\mu = \frac{L(I)}{n-1} \quad \text{pas}_\nu = \frac{L(J)}{n-1}$$

Même si dans ce rapport on utilise souvent les densités de ces mesures, dans la pratique on les représente plutôt par leurs fonctions de répartition (qui sont donc affines par morceaux).

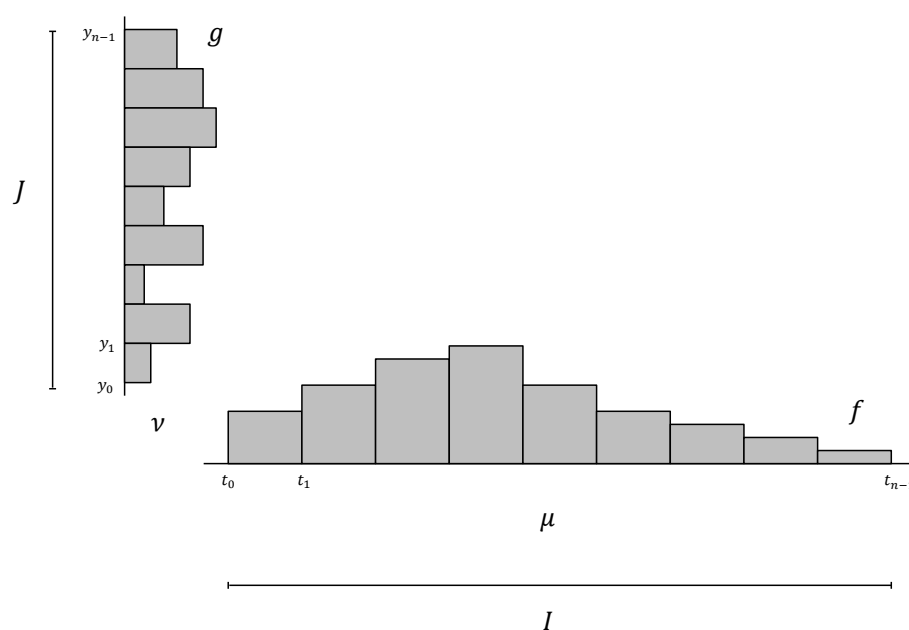


FIGURE 5.1 – Schéma récapitulatif des notations

Comme le suggère le théorème 9, le plan de transport optimal qui minimise F_δ est, dans le cas de la dimension 1, une fonction monotone par morceaux. Ces intervalles de monotonie correspondent à une décomposition de la mesure d'arrivée ν . Les individus de notre algorithme seront donc des m -uplets de mesures $\langle \nu_0, \nu_1, \dots, \nu_{m-1} \rangle$ vérifiant la relation suivante :

$$\nu_0 + \nu_1 + \dots + \nu_{m-1} = \nu$$

Notre algorithme permet de minimiser F_δ parmi les transports ayant m monotonies alternées et dont la première monotonie est fixée (dans la suite, on supposera que les transports parmi lesquels on minimise F_δ commencent par un premier morceau croissant). On notera g_j la densité de la mesure ν_j .

5.1.2 Initialisation

On souhaite créer des individus suffisamment diversifiés (pour ne pas perdre d'éventuelles bonnes solutions finales) qui constitueront notre population initiale. Ces individus sont donc créés de manière aléatoire. L'idée naturelle pour construire un individu de manière aléatoire serait de découper chaque partie constante par morceaux de la densité de la mesure ν selon une partition aléatoire à m éléments et d'affecter ensuite chaque sous partie à une mesure ν_j . Plus précisément, pour tout $k \in [0; n - 2]$, on crée une partition aléatoire $g^{(0)}(t_k), g^{(1)}(t_k), \dots, g^{(m-1)}(t_k)$ de $g(t_k)$ telle que :

$$g^{(0)}(t_k) + g^{(1)}(t_k) + \dots + g^{(m-1)}(t_k) = g(t_k)$$

On construit ainsi l'individu $\langle \nu_0, \nu_1, \dots, \nu_{m-1} \rangle$ de telle sorte que ν_i ait pour densité $g^{(i)}$ constante par morceaux. Malheureusement cette méthode n'est pas assez générale. En effet, n étant amené à être assez grand, si les partitions aléatoires sont toutes créées avec une loi uniforme, la masse totale distribuée pour chaque ν_j sera la même (égale à $\frac{1}{m}$) d'après la loi forte des grands nombres.

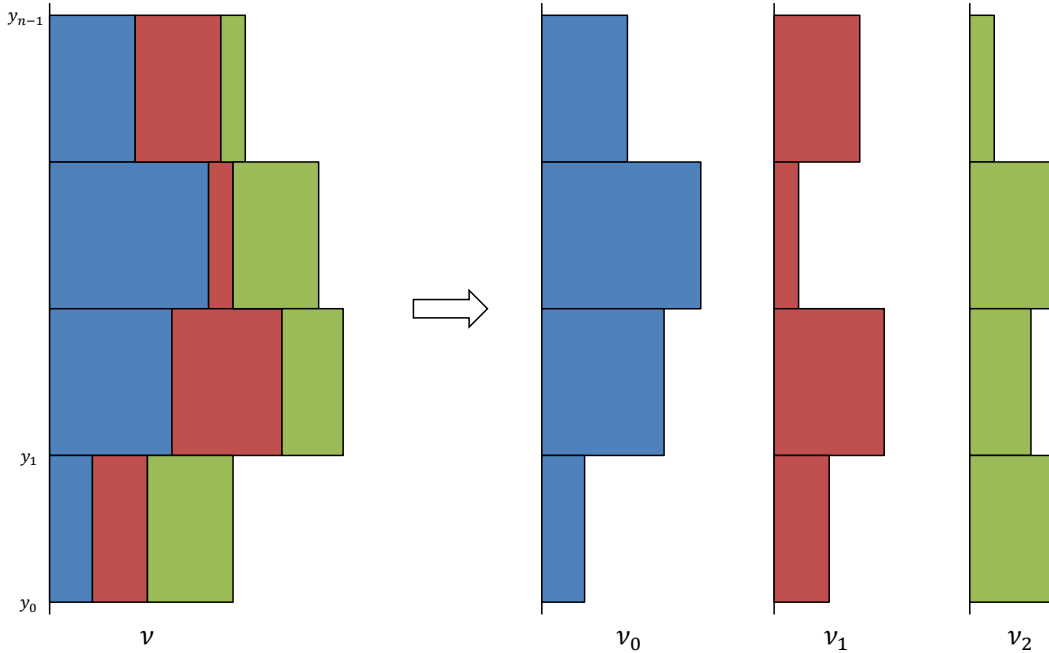


FIGURE 5.2 – Création des ν_i à partir de ν

Suivant les conseils de Filippo SANTAMBROGIO, nous avons donc utilisé une méthode qui fixerait d'abord la masse alouée pour chaque ν_j de l'individu de manière aléatoire. On choisit donc $\alpha_0, \alpha_1, \dots, \alpha_{m-1}$ aléatoirement tels que :

$$\alpha_0 + \alpha_1 + \dots + \alpha_{m-1} = 1 \quad \alpha_j \in]0, 1[$$

Ce α_j correspond à la masse totale (en moyenne) qui sera alouée à la mesure ν_j . On utilise alors une technique semblable à ce qui précède, à ceci près que dans ce cas, nous tirons aléatoirement les $g^{(j)}(t_k)$

non plus suivant une loi uniforme mais suivant une loi uniorne élevée à une certaine puissance p_j tel que :

$$\mathbb{E}(X^{p_j}) = \alpha_j \quad \iff \quad p_j = \frac{1}{\alpha_j} - 1$$

Dans la pratique, on crée les ν_j de manière itérative; cela maintient donc une corrélation entre les ν_j et ne nous donne pas des individus distribués uniformément, ce qui n'est pas gênant ici car notre but était seulement d'avoir des individus *suffisamment* diversifiés. Les détails de l'initialisation sont donnés en annexe sous la forme de pseudo-code.

5.1.3 Calcul de l'oscillation

L'un des enjeux ici était de calculer l'oscillation de manière efficace pour réduire le temps d'exécution de l'algorithme (en effet, c'est une opération que l'algorithme va potentiellement effectuer un très grand nombre de fois). Pour cela on procède en deux étapes.

Calcul des images réciproques des y_k La principale difficulté du calcul de $F_\delta(T)$ ici est que notre discrétisation (t_k) de l'espace de départ ne nous renseigne pas assez sur leur comportements dans l'espace d'arrivée. On crée donc une nouvelle discrétisation de l'espace de départ, totalement indépendante des (t_k), basée cette fois-ci sur la discrétisation de l'espace d'arrivée (y_k), que l'on note (x_k^j). Ils sont construits de la manière suivante :

$$\begin{aligned} \int_{t_0}^{x_k^j} d\mu &= \int_J d(\nu_0 + \dots + \nu_{j-1}) + \int_{y_0}^{y_k} d\nu_j & \text{si } j \bmod 2 = 0 \\ \int_{t_0}^{x_k^j} d\mu &= \int_J d(\nu_0 + \dots + \nu_{j-1}) + \int_{y_{n-1-k}}^{y_{n-1}} d\nu_j & \text{si } j \bmod 2 = 1 \end{aligned}$$

On appelle les (x_k^j) les *images réciproques* des (y_k) par μ . On peut remarquer qu'à j fixé, les (x_k^j) forment une partition de l'intervalle $[x_0^j, x_{n-1}^j]$ telle que le transport monotone (croissant si j est pair, décroissant sinon) de la mesure $\mu|_{[x_0^j, x_{n-1}^j]}$ vers ν_j soit affine par morceaux suivant cette partition. Ces m transports mis les uns à la suite des autres, on obtient alors le transport de μ vers ν , de monotonies alternées suivant la décomposition de ν induite par l'individu $\langle \nu_0, \nu_1, \dots, \nu_{m-1} \rangle$, que l'on note T . A ce stade, on sait que, à j fixé :

$$|T(x_k^j) - T(x_l^j)| = |y_k - y_l| = \text{pas}_\nu \times |k - l| \quad (5.1)$$

Cette discrétisation de l'espace sera donc très utile pour le calcul de F_δ .

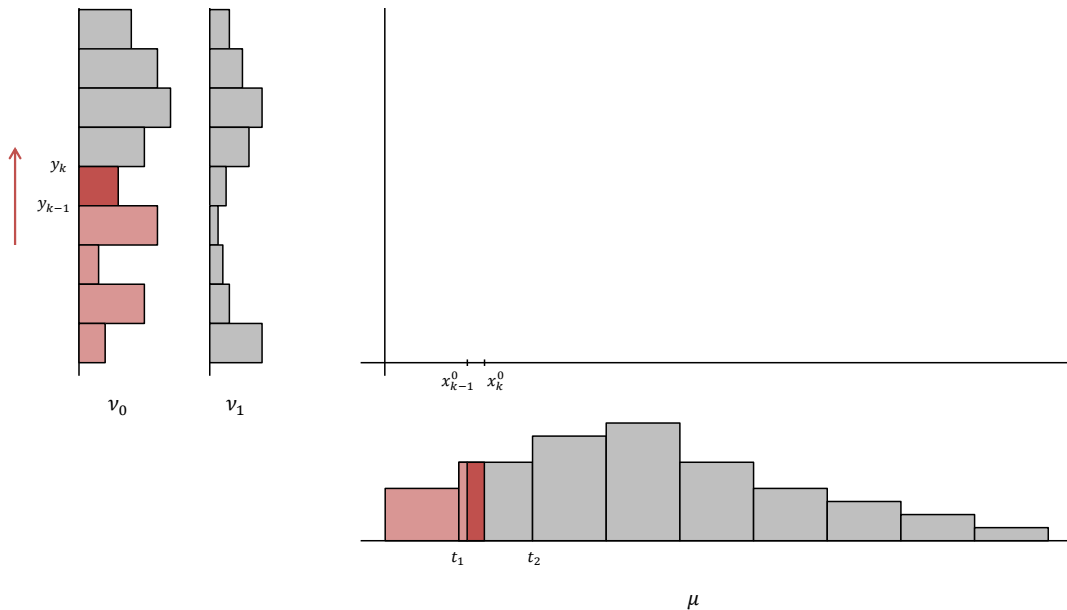


FIGURE 5.3 – Création des (x_k^j) lorsque j est pair

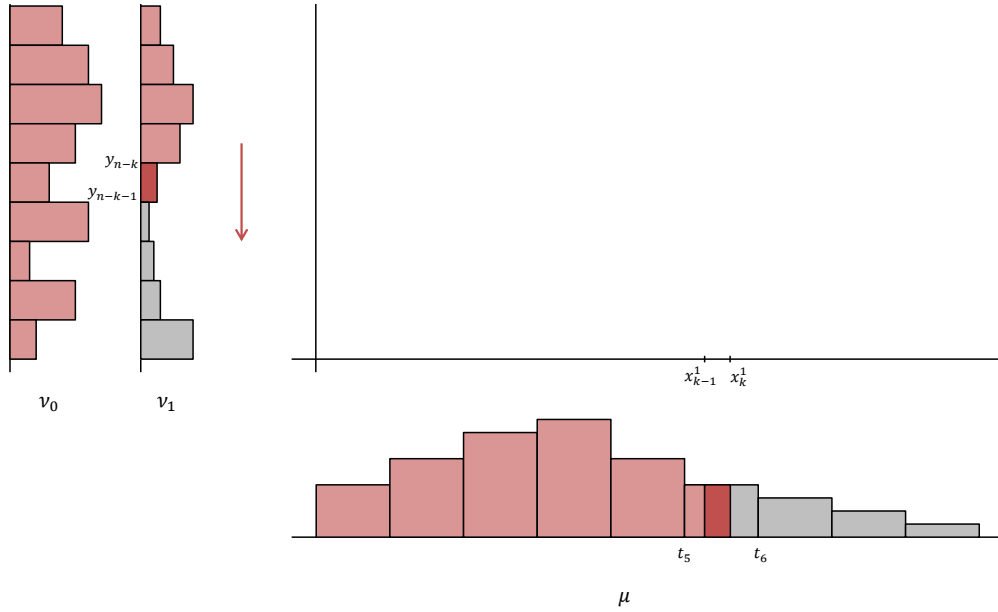


FIGURE 5.4 – Création des (x_k^j) lorsque j est impair

Calcul de F_δ On a construit T de telle sorte qu'il soit monotone par morceaux et continu. Il est clair que l'oscillation maximale de T ne pourra pas se trouver sur un intervalle $[x, y]$ de longueur au plus δ sur lequel T change de monotonie. Dans notre calcul de $F_\delta(T)$ on se limite donc juste à chaque intervalle $[x_0^j, x_{n-1}^j]$ sur lequel T est monotone.

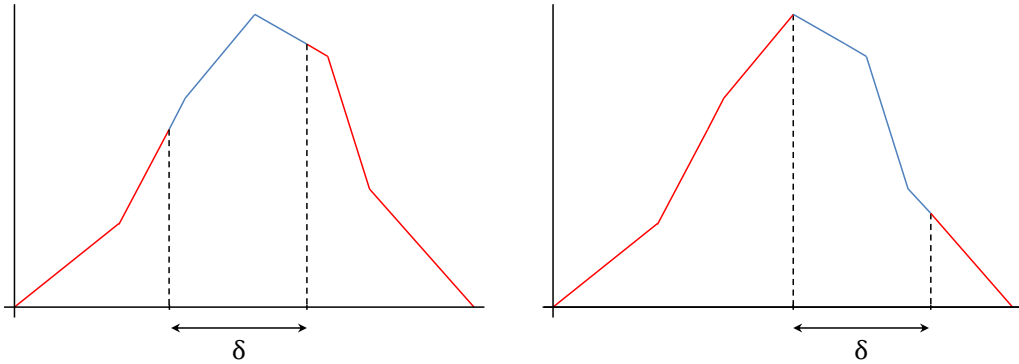


FIGURE 5.5 – Calcul de l'oscillation dans le cas non optimal (à gauche) où T change de monotonie sur l'intervalle de longueur δ et un cas meilleur (à droite) où T garde la même monotonie

Puisqu'on se limite à ces intervalles sur lequel T garde la même monotonie, on a alors, pour x et y dans un même intervalle $[x_0^j, x_{n-1}^j]$:

$$F_\delta(T) = \sup_{|x-y| \leq \delta} |T(x) - T(y)| = \sup_{|x-y| = \delta} |T(x) - T(y)| \quad (5.2)$$

On calcule alors $F_\delta(T)$ en utilisant la formule 5.1 et en se limitant uniquement au cas où

$$x_{k-1}^j - x_l^j < \delta \leq x_k^j - x_l^j \quad (k > l) \quad (5.3)$$

Cette opération peut se faire dynamiquement et ne requiert donc que $O(n \times m)$ opérations. Les détails du calcul de $F_\delta(T)$ sont donnés en annexe sous la forme de pseudo-code.

Erreur L'erreur sur le calcul $F_\delta(T)$ vient de deux approximations faites dans la procédure présentée :

- Tout d’abord le transport T qu’on obtient n’est pas *exactement* le transport de μ sur ν . On a supposé que la discrétisation (x_k^j) est indépendante de la discrétisation (t_k) . Pourtant, si il existe k' et k tel que

$$x_k^j < t_{k'} < x_{k+1}^j \quad \text{et} \quad f(t_{k'-1}) \neq f(t_{k'})$$

on constaterait au niveau du transport T une rupture de pente en $t_{k'}$ alors qu’elle n’apparaît pas dans notre construction (les ruptures de pente n’existent qu’au niveau des (x_k^j)) comme on le remarque dans la figure 5.6. Cette approximation est donc à l’origine d’une première erreur de *sous-évaluation* de $F_\delta(T)$.

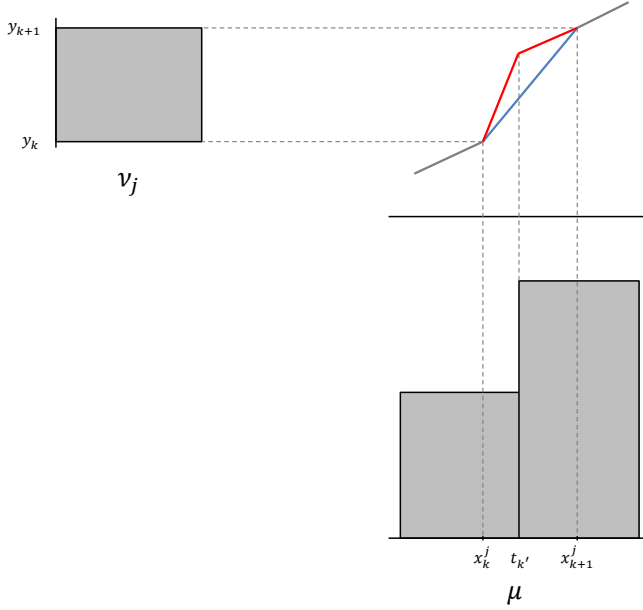


FIGURE 5.6 – Comparaison locale de T (en rouge) et du transport que l’on obtient (en bleu)

- Une seconde erreur, cette fois-ci de *sur-évaluation* de $F_\delta(T)$, vient du fait qu’on choisit, dans l’équation 5.3 k et l tel que la différence entre x_k^j et x_l^j soit éventuellement légèrement supérieure à δ là où l’équation 5.2 demande une stricte égalité.

Dans les deux cas, on montre que l’erreur commise est au plus de l’ordre de $\frac{1}{n}$.

5.1.4 Reproduction

Au niveau de la reproduction, nous sommes restés sur une idée plutôt simple qui ne dépend pas du problème en ne générant que des moyennes des éléments déjà présents. En d’autres termes, si $\text{parent}_1 = \langle \nu_0^{(1)}, \nu_1^{(1)}, \dots, \nu_{m-1}^{(1)} \rangle$ et $\text{parent}_2 = \langle \nu_0^{(2)}, \nu_1^{(2)}, \dots, \nu_{m-1}^{(2)} \rangle$ sont 2 individus parents, on génère alors :

$$\begin{aligned} \text{enfant}_1 &= \langle \frac{2}{3}\nu_0^{(1)} + \frac{1}{3}\nu_0^{(2)}, \frac{2}{3}\nu_1^{(1)} + \frac{1}{3}\nu_1^{(2)}, \dots, \frac{2}{3}\nu_{m-1}^{(1)} + \frac{1}{3}\nu_{m-1}^{(2)} \rangle \\ \text{enfant}_2 &= \langle \frac{1}{3}\nu_0^{(1)} + \frac{2}{3}\nu_0^{(2)}, \frac{1}{3}\nu_1^{(1)} + \frac{2}{3}\nu_1^{(2)}, \dots, \frac{1}{3}\nu_{m-1}^{(1)} + \frac{2}{3}\nu_{m-1}^{(2)} \rangle \end{aligned}$$

La reproduction préserve donc évidemment la relation 5.1.1 tout au long de l’algorithme.

5.1.5 Mutation

Si la reproduction est un peu naïve, la mutation elle doit être un peu plus sophistiquée pour éviter de ne rester bloqué dans la partie convexe délimitée par la population initiale. On a donc envisagé 3 méthodes différentes dont une seule donne des résultats suffisamment corrects. Elle est appelée la *mutation par compensation de masse*, et se base en fait sur 2 mutations aux effets radicalement différents. On décrit ici ces deux composants de la mutation par compensation de masse et laissons en annexe les 2 autres mutations infructueuses.

Mutation par diminution de pente Lors du calcul de $F_\delta(T)$, l'algorithme identifie non seulement la valeur de l'oscillation maximale mais également sur quel sous-intervalle de longueur δ elle apparaît (si il en existe plusieurs, il en choisit un arbitrairement). Cet intervalle est de la forme $[x_k^j, x_l^j]$. Ainsi, d'après la construction des (x_k^j) , cet intervalle correspond à l'arrivée à l'intervalle $[y_k, y_l]$ si j est pair et $[y_{n-1-l}, y_{n-1-k}]$ si j est impair. Pour se fixer les idées, supposons que j est pair (le cas impair est similaire).

La pente moyenne de T sur $[x_k^j, x_l^j]$ est la plus forte et est due à une masse trop faible localisée sur $[y_k, y_l]$. L'idée ici est de diminuer cette pente en augmentant la masse de ν_j sur $[y_k, y_l]$. En d'autres termes, on tire $m-1$ nombres aléatoires $(\alpha_{j'})_{j' \neq j}$ suivant une loi uniforme sur $[0, 1]$ et on effectue les changements suivants :

$$\begin{aligned} g_{j|[y_k, y_l]} &\leftarrow g_{j|[y_k, y_l]} + \sum_{j' \neq j} (1 - \alpha_{j'}) \times g_{j'|[y_k, y_l]} \\ g_{j'|[y_k, y_l]} &\leftarrow \alpha_{j'} \times g_{j'|[y_k, y_l]} \quad \forall j' \neq j \end{aligned}$$

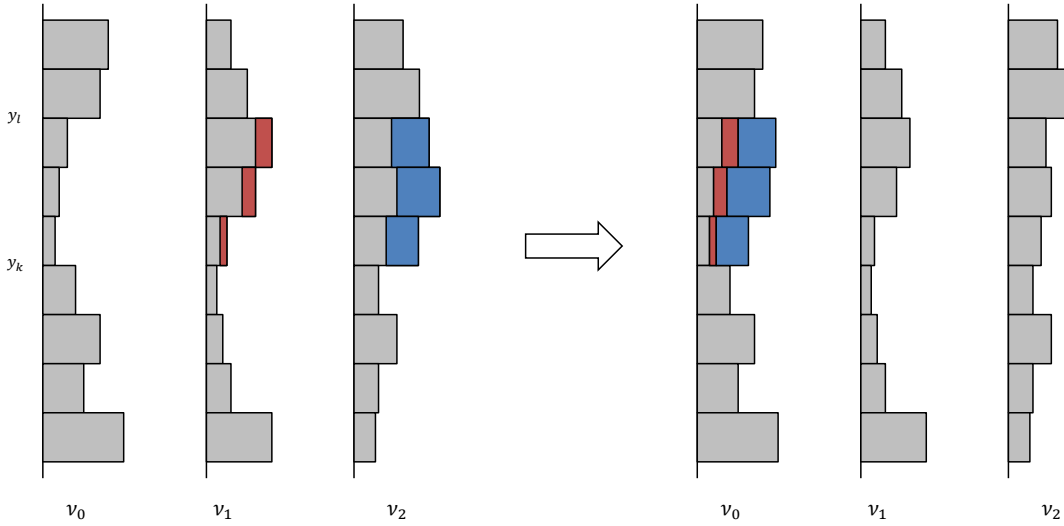


FIGURE 5.7 – Exemple de mutation par diminution de pente pour j pair

Mutation par augmentation de pente On peut également, de la même manière, identifier le sous-intervalle de longueur δ sur lequel l'oscillation est minimale (tout en veillant à avoir T monotone sur ce sous-intervalle). De même que précédemment, on note $[x_k^j, x_l^j]$ ce sous-intervalle et on se place dans le cas où j est pair pour se fixer les idées. Contrairement à ce qui précède, la pente moyenne de T sur ce sous-intervalle est trop faible et est due à une masse trop importante localisée sur $[y_k, y_l]$. On va donc augmenter cette pente en diminuant la masse de ν_j sur $[y_k, y_l]$. On tire alors α uniformément sur $[0, 1]$ et on effectue les changements suivants :

$$\begin{aligned} g_{j|[y_k, y_l]} &\leftarrow \alpha \times g_{j|[y_k, y_l]} \\ g_{j'|[y_k, y_l]} &\leftarrow g_{j'|[y_k, y_l]} + \frac{1 - \alpha}{m - 1} \times g_{j|[y_k, y_l]} \quad m \neq 1, \forall j' \neq j \end{aligned}$$

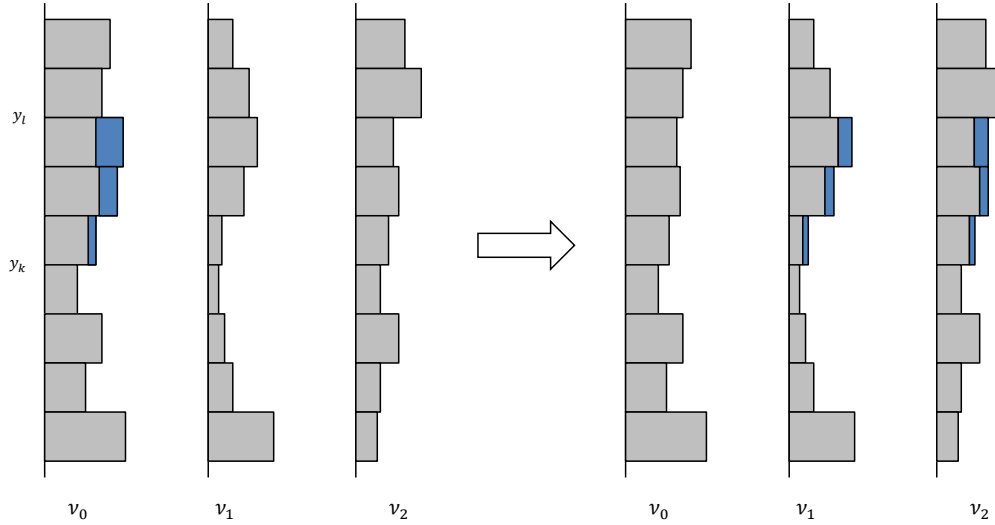


FIGURE 5.8 – Exemple de mutation par augmentation de pente pour j pair

Il faut cependant faire attention à ce genre de mutation qui vont à l’encontre de notre but (augmente la pente alors qu’on souhaite diminuer l’oscillation), mais c’est un mal nécessaire pour sortir des minima locaux de F_δ très attractifs. Dans la pratique, cette mutation n’est choisie que dans 15% des mutations (contre 85% pour la mutation par diminution de pente).

Enfin, on remarque que cette mutation par compensation de masse préserve bien la relation 5.1.1 tout au long de l’algorithme.

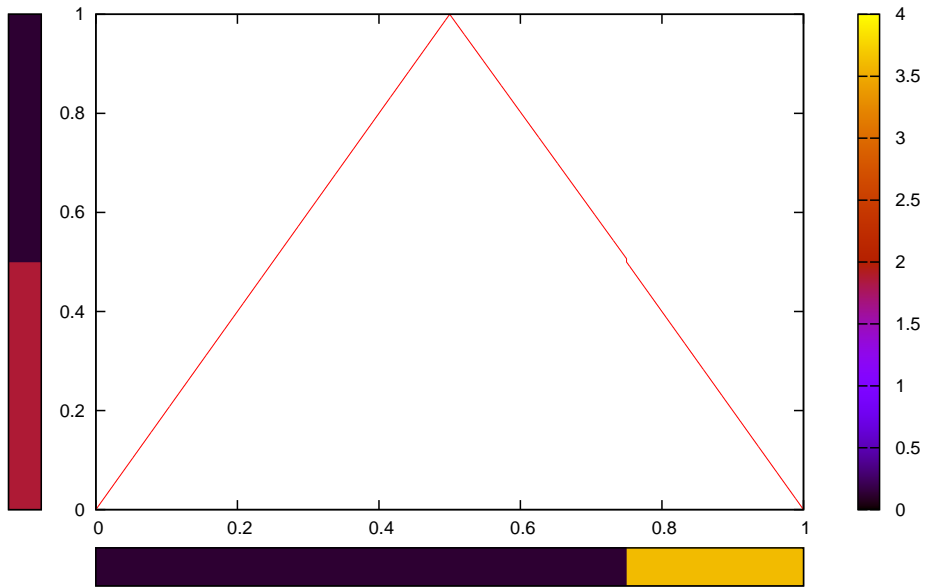
5.2 Implémentation et exemples

5.2.1 Implémentation

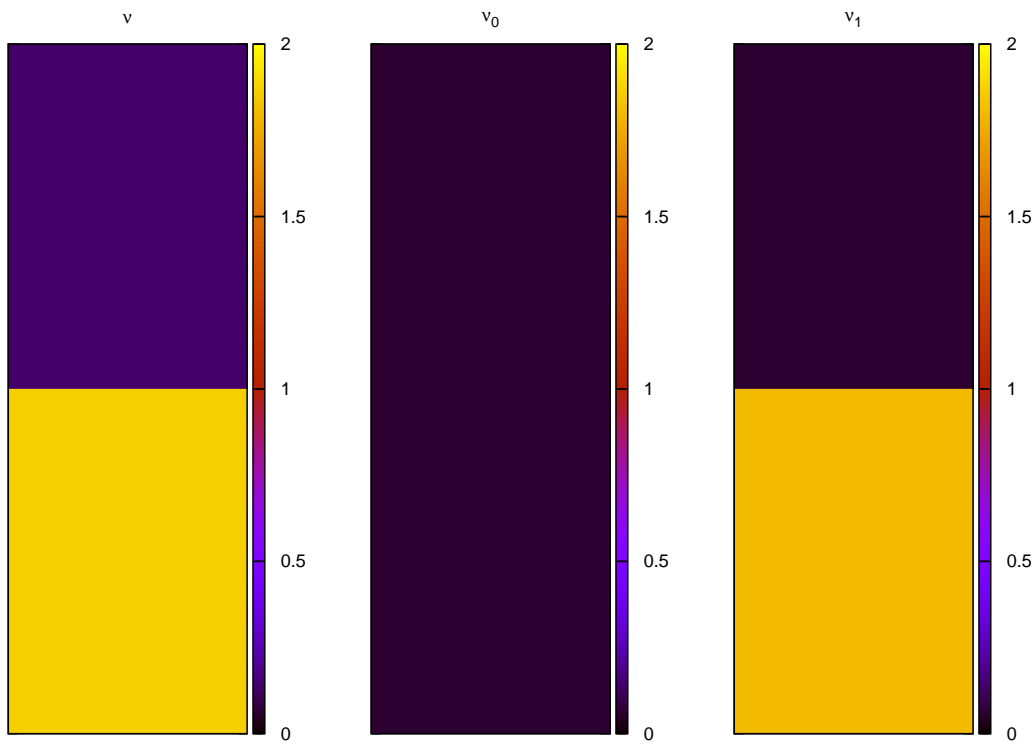
Suivant les conseils de Loïc SIMON, nous avons utilisé une librairie d’algorithmes génétiques pour C++ appelée Evolving Object. Cette librairie dispose déjà de tous les composants de bases pour s’adapter au problème considéré très rapidement. Ainsi, il suffit donc d’implémenter les composants décrits dans la partie 5.1 pour obtenir un programme fonctionnel. Malheureusement, la documentation de cette librairie était très obscure et la manière dont fonctionnaient les différents composants était peu intuitive pour des débutants en C++ comme nous l’étions. Nous avons malgré tout réussi à obtenir un programme fonctionnel et modulable que nous avons testé sur un très petit ensemble d’exemples suffisamment discriminants pour notre problème.

5.2.2 Exemple

Cas où l’optimum n’est pas le transport croissant Contrairement au théorème 5 qui assure que l’optimum est le transport croissant pour un coût particulier, le théorème 9, lui, ne fait que borner le nombre de monotomies alternées du transport. Il se peut dès lors que le transport croissant ne soit pas l’optimum pour notre problème. Nous donnons ici un exemple, construit spécialement pour obtenir un transport optimal en 2 morceaux. Nous avons utilisé une mesure μ , à support sur $I = [0, 1]$, qui possède une densité très faible sur $[0, \frac{3}{4}]$ et une densité très importante sur $[\frac{3}{4}, 1]$. Nous définissons ensuite ν comme étant la mesure image de μ par une fonction affine par morceaux en “triangle”. La décomposition de la mesure ν en $\langle \nu_0, \nu_1 \rangle$, les mesures μ et ν et la fonction en “triangle” T_{ideal} sont récapitulées dans la figure 5.9



(a) Fonction T_{ideal} avec μ (en abscisse) et ν (en ordonnée)



(b) Décomposition correspondant à T_{ideal}

FIGURE 5.9 – Construction de l'exemple

L'avantage d'un tel exemple est que nous pouvons calculer manuellement $F_\delta(T_{\text{ideal}})$; pour $\delta = 0.1$, nous avons donc $F_\delta(T_{\text{ideal}}) = 0.2$. Notre algorithme génétique nous a donné des résultats satisfaisants sur cet exemple puisque nous obtenons des transports (en 2 morceaux) très proches de T_{ideal} . Nous donnons en figure 5.10 quelques transports obtenu à partir de cet algorithme génétique après 1000 itérations et en figure 5.11 le meilleur résultat que nous avons obtenu, très proche du résultat idéal de la figure 5.9.

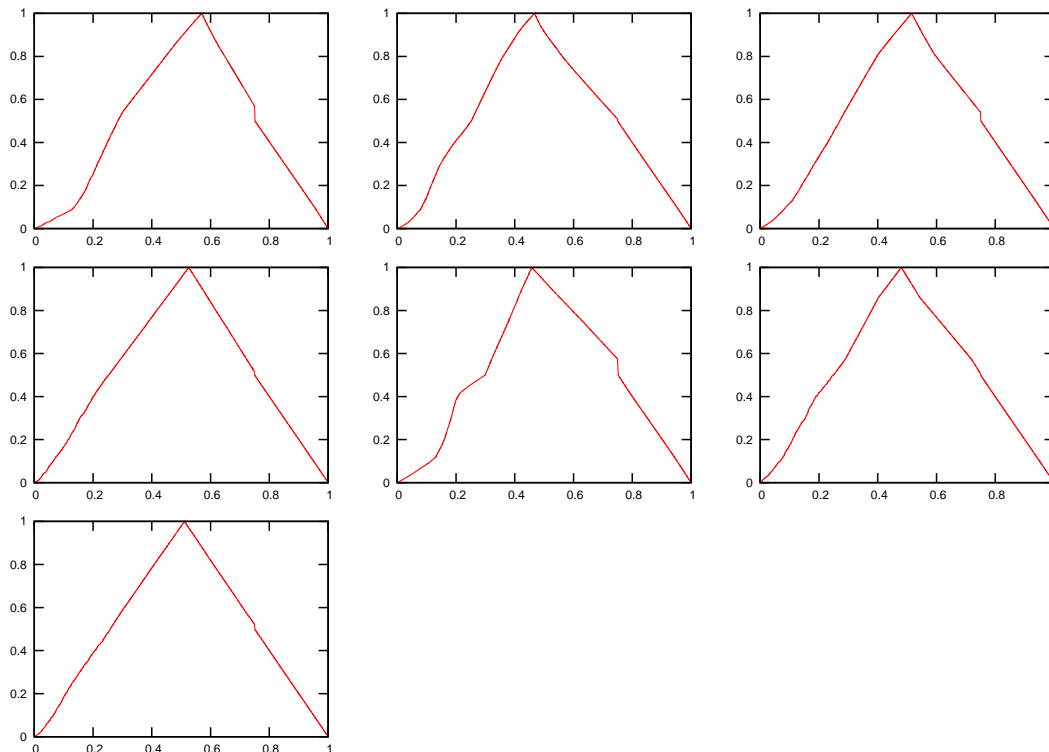


FIGURE 5.10 – Quelques résultats obtenu avec notre algorithme génétique

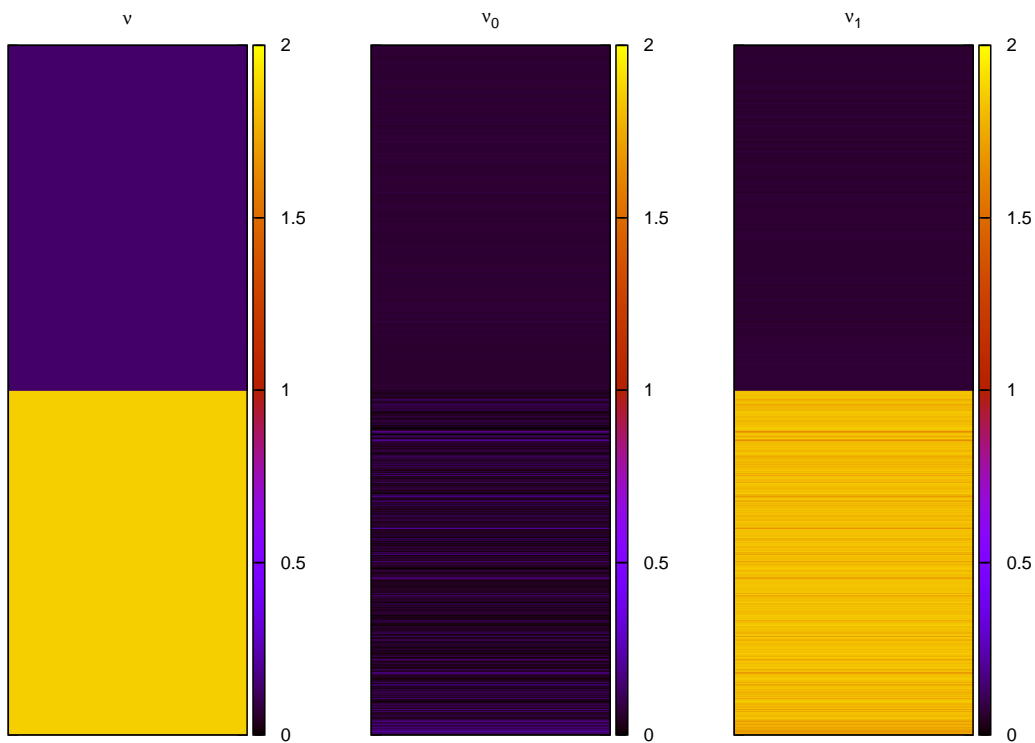
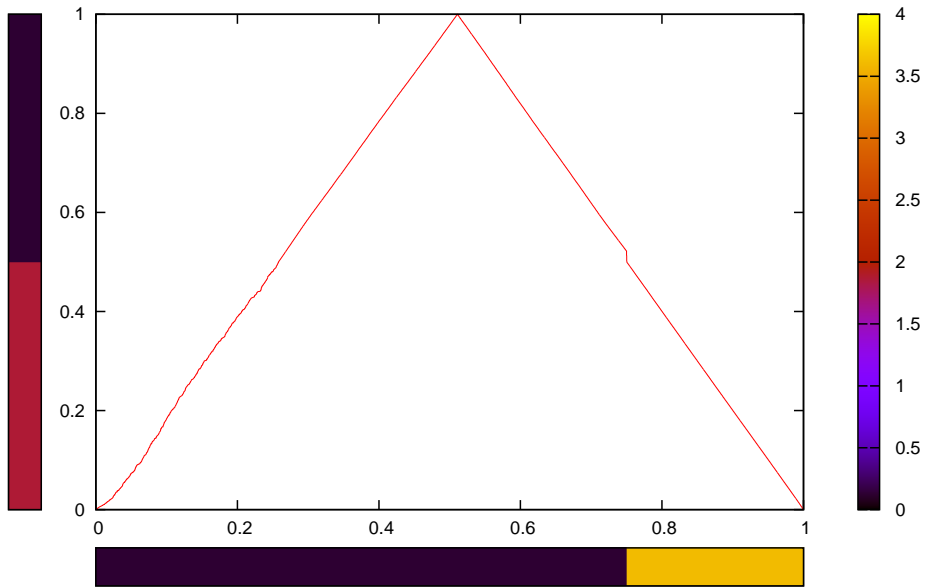


FIGURE 5.11 – Meilleur résultat - $F_\delta(T) \approx 0.223$ pour $\delta = 0.1$

Toutefois, il faut noter que notre algorithme génétique peut, dans de rares cas, donner un résultat correspondant à un minimum local très attractif, comme nous l'avons expliqué dans la partie 4.2.4. Ce transport ainsi que la décomposition $\langle \nu_0, \nu_1 \rangle$ correspondante est donnée à la figure 5.12. Cependant, on peut imaginer raisonnablement que plusieurs lancements en parallèle de cet algorithme avec sélection du meilleur résultat nous donneraient de très bons résultats globaux.

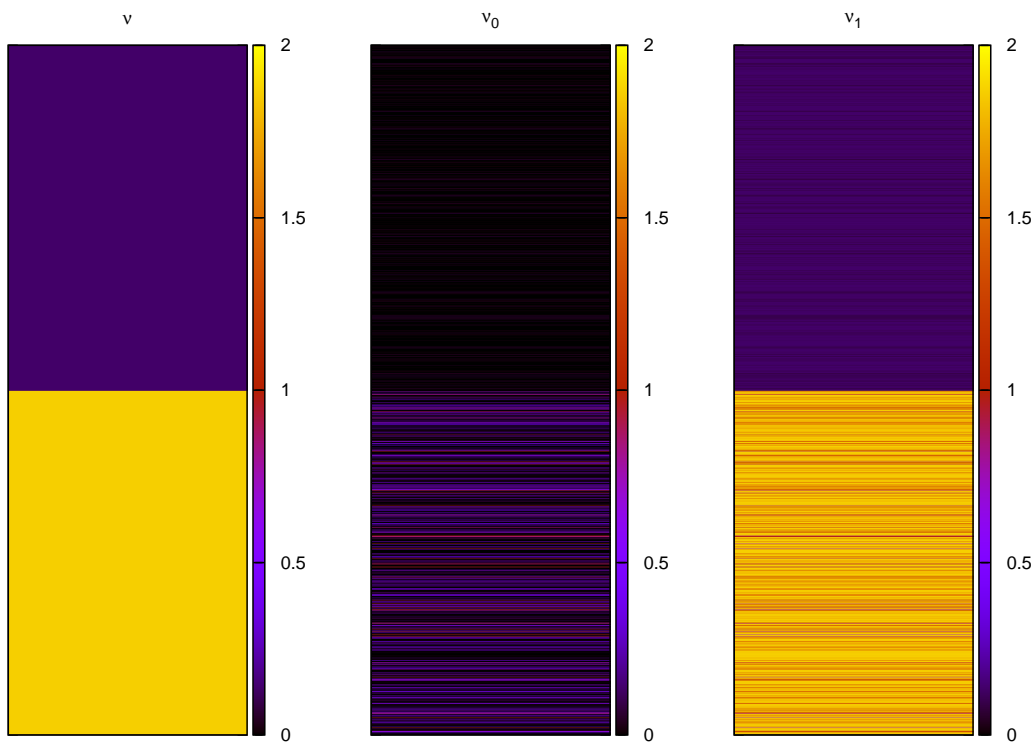
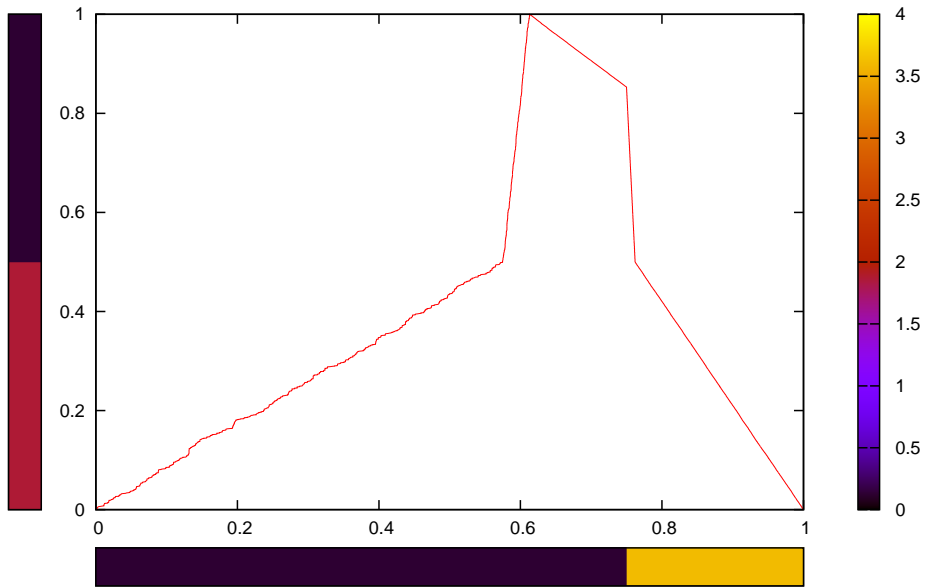


FIGURE 5.12 – Meilleur résultat - $F_\delta(T) \approx 0.557$ pour $\delta = 0.1$

Enfin, l’observation en figure 5.13 de différents transports (correspondants à différents individus) au cours de l’algorithme est très intéressante. En effet, on remarque que l’algorithme, après applications successives de plusieurs mutations par diminution de pente, “converge” vers le minimum local décrit à la figure 5.12 (transports 2 à 4 de la figure 5.13) pour ensuite en sortir après une application d’une mutation par augmentation de pente (transports 5 et 6 de la figure 5.13).

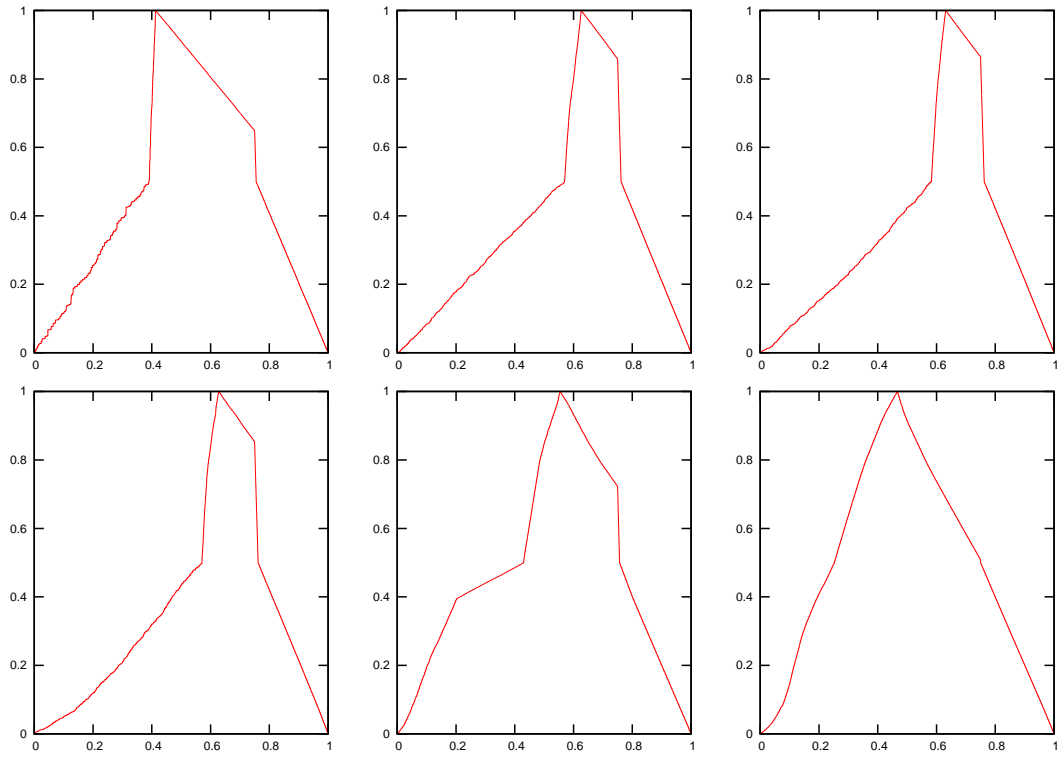


FIGURE 5.13 – Etapes de l’algorithmes, de la gauche vers la droite, du haut vers le bas

Bibliographie

- [1] BRÉZIS Haïm, *Analyse fonctionnelle : Théorie et applications* Dunod, 2005
- [2] RUDIN Walter, *Analyse réelle et complexe* Dunod, 2009
- [3] MONGE Gaspard, *Memoire sur la théorie des déblais et des remblais*, 1781
- [4] SANTAMBROGIO Filippo, *Optimal Transport Course* 2012
- [5] VILLANI Cedric, *Optimal transport : Old and New*, Springer, 2008
- [6] VILLANI Cedric, *Topics in Optimal Transportation*, AMS, 2003
- [7] LUKE Sean, *Essentials of Metaheuristics* 2012
- [8] LESESVRE Didier, PEGON Paul, SANTAMBROGIO Filippo, *Optimal transportation with an oscillation-type cost : the one-dimensional case* 2011
- [9] S. MASCETTI, D. FRENI, C. BETTINI, X.-S. WANG, S. JAJODIA, *Privacy in geo-social networks : proximity notification with untrusted service providers and curious buddies*, VLDB Journal, 2011

Annexe A

Détails de l'algorithme

A.1 Initialisation

L'algorithme pour créer une population initiale d'individus aléatoires a été décrit à la partie 5.1.2. Voici cet algorithme plus en détail sous la forme de pseudo-code.

Algorithm 1 Création d'un individu aléatoire

Require: ν mesure de densité g

```
for  $j = 0; j < m; j ++$  do
   $\alpha \leftarrow$  nombre aléatoire entre 0 et  $\nu(J)$  suivant une loi uniforme
   $p \leftarrow \frac{1}{\alpha} - 1$ 
  if  $j = m - 1$  then
     $\nu_j \leftarrow \nu$ 
  else
    for  $i = 0; i < n - 1; i ++$  do
       $\beta \leftarrow$  nombre aléatoire entre 0 et 1 suivant une loi uniforme
       $g_j(y_i) \leftarrow \beta^p \times g(y_i)$ 
       $g(y_i) \leftarrow (1 - \beta^p) \times g(y_i)$ 
    end for
     $\nu_j \leftarrow$  mesure de densité  $g_j$ 
  end if
end for
return  $\langle \nu_0, \nu_1, \dots, \nu_{m-1} \rangle$ 
```

A.2 Calcul de F_δ

L'algorithme pour calculer F_δ à partir d'un individu a été décrit à la partie 5.1.3.

Calcul des (x_k^j) Voici le détail du calcul des images réciproques des (y_k) par μ sous la forme de pseudo-code.

Algorithm 2 Calcul des (x_i^j)

Require: μ mesure de fonction de répartition F , $\langle \nu_0, \nu_1, \dots, \nu_{m-1} \rangle$ un individu avec G_j fonction de répartition de ν_j

$G \leftarrow 0$

for $j = 0; j < m; j++$ **do**

$k \leftarrow 0$

$v \leftarrow j \bmod 2$

for $i = 0; i < n; i++$ **do**

$G \leftarrow \nu_0(J) + \dots + \nu_{j-1}(J) + v \times [G_j(y_{n-1}) - G_j(y_{n-1-i})] + (1-v) \times G_j(i)$

while $F(k) \leq G \wedge k < n$ **do**

$k \leftarrow k + 1$

end while

$x_i^j \leftarrow (k-1) \times \text{pas}_\mu + \frac{(G-F(k-1)) \times \text{pas}_\mu}{G_j(k) - G_j(k-1)}$

end for

end for

return (x_i^j)

Algorithme dynamique et calcul de F_δ Comme nous l'avons vu dans la partie 5.1.3, le calcul de F_δ repose sur la recherche de couples (l, k) vérifiant les inégalités 5.3. Cette recherche peut se faire naïvement en explorant tous les x_l^j et en recherchant x_k^j se situant à une distance δ ; cette méthode, même avec un algorithme de recherche dichotomique aurait une complexité temporelle de l'ordre de $O(n \cdot m \ln(n \cdot m))$, ce qui est trop important pour une opération qu'on doit répéter un grand nombre de fois.

Le principe de l'algorithme dynamique que nous utilisons est de se souvenir des éléments que nous avons déjà exploré et de ne faire la recherche qu'à partir d'un certain rang. Si le couple d'indices $(l, l + \Delta)$ vérifie les inégalités 5.3, on ne recherche l'indice k tel que $(l + 1, k)$ vérifie 5.3 qu'à partir de $l + \Delta$, comme on l'illustre dans la figure A.1. Cet algorithme a une complexité temporelle de l'ordre de $O(n \times m)$.

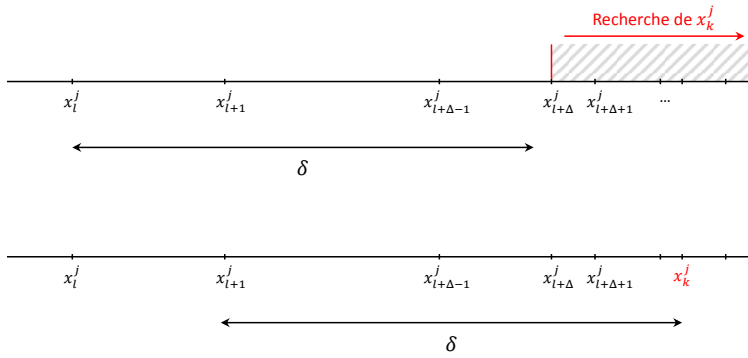


FIGURE A.1 – Recherche des couples vérifiant la condition 5.3

Voici le détail du calcul de $F_\delta(T)$ sous la forme de pseudo-code :

Algorithm 3 Calcul de $F_\delta(T)$

Require:

```
 $k_{\max} \leftarrow 0$ 
for  $j = 0; j < m; j++$  do
   $k \leftarrow 0$ 
   $i \leftarrow 0$ 
  while  $x_i^j + \delta \leq x_{n-1}^j$  do
    while  $x_i^j + \delta > x_k^j \wedge k < n$  do
       $k \leftarrow k + 1$ 
    end while
     $k_{\max} \leftarrow \max(k_{\max}, k - i - 1)$ 
     $i \leftarrow i + 1$ 
  end while
  if  $x_{n-1}^j - \delta \geq x_0^j \wedge n - i - 2 > k_{\max}$  then
     $k_{\max} \leftarrow \max(k_{\max}, n - i - 2)$ 
  end if
end for
return  $k_{\max} \times \text{pas}_\nu$ 
```

A.3 Mutations infructueuses

Comme nous l'avons indiqué dans la partie 5.1.5, nous avons envisagé deux autres mutations qui se sont révélées infructueuse, car elles nous obligeaient à parfois rester dans des minima locaux.

Mutation par création d'une partie de l'individu Notre première idée suivait celle que nous utilisions pour l'initialisation, à savoir créer un nouvel individu à partir de la mesure d'arrivée ν . Ici, on ne recréait qu'une partie de l'individu (sur certains intervalles de discrétisation seulement, déterminés de manière aléatoire) et laissait invariantes les autres parties. Malheureusement, cette technique (surement pas assez spécifique à notre problème) nous empêchait de sortir des éventuels minima locaux, ce qui, comme nous l'avons vu, n'est pas ce qui est recherché chez les algorithmes génétiques.

Mutation par permutation Une autre idée était de permuter de manière aléatoire les mesures d'un individu. En d'autres termes, étant donné un individu $\langle \nu_0, \nu_1, \dots, \nu_{m-1} \rangle$ et une permutation aléatoire σ , on crée un nouvel individu $\langle \nu_{\sigma(0)}, \nu_{\sigma(1)}, \dots, \nu_{\sigma(m-1)} \rangle$. Encore une fois, cette mutation ne permettait pas de se détacher de minima locaux.