


Modélisation mathématique et implémentation sur python d'un système de parrainage automatisé

Résumé : *Le travail présenté dans cet article a pour objectif de créer un système de parrainage partiellement ou totalement automatisé s'appuyant sur des algorithmes d'affectation. Pour se faire, nous commencerons par une analyse exploratoire des données qui englobera étude du type de chaque covariable, description statistique (moyenne, variance, minimum, maximum) mais aussi des techniques d'embeddings afin de construire une métrique permettant de quantifier le taux de compatibilité entre parrains et filleuls. Nous présenterons dans un deuxième temps les avantages et limites de quelques algorithmes classiques faisant partie de l'art a savoir l'algorithme de Gale-Shapley et l'algorithme hongrois. Dans un troisième temps, nous détaillerons le cadre théorique notre modèle en utilisant des techniques d'optimisation convexe sous contraintes, analyse et algèbre linéaire. L'étude sera complétée par une section de résultats numériques où nous mettrons en évidence la pertinence des résultats théoriques, la convergence des algorithmes construits et les performances du modèle en termes de niveau d'automatisation et maximisation de compatibilité. Enfin, nous présenterons deux pistes d'améliorations avec les résultats numériques associés.*

Mots clefs : *optimisation convexe sous contraintes , mesures de similarité, statistiques, analyse convexe*

Table des matières

1 Introduction

De nos jours, l'utilisation de l'intelligence artificielle est de plus en plus répandue pour automatiser les tâches manuelles complexes et répondre au besoin des utilisateurs. Dans ce contexte s'intègre cette étude ayant pour objectif de répondre à un besoin en utilisant des techniques d'optimisation convexe sous contraintes et des outils de modélisation statistique.

En effet, nous désirons créer un système de parrainage visant à accompagner les nouveaux élèves internationaux en suggérant pour chacun un parrain afin de faciliter leur intégration et maximiser leur expérience. Pour se faire, nous nous baserons sur les données des élèves internationaux (filleuls) ainsi que celle des parrains afin de déterminer une métrique pertinente évaluant la compatibilité entre parrain et filleul. Nous modéliserons ensuite le besoin sous forme d'un problème d'optimisation sous contraintes. Enfin, nous présenterons les résultats obtenus, améliorerons ces derniers et discuterons limites et perspectives du modèle.

Tout au long de l'article, nous veillerons à bien détailler les outils mathématiques (analyse fondamentale, algèbre linéaire, optimisation, statistiques) utilisés ainsi que le schéma algorithmique exploitant les résultats théoriques. Nous nous assurerons aussi de démontrer les propositions les plus importantes et commenterons les résultats numériques obtenus.

2 Présentation des données

En termes d'entrées, nous disposons de deux bases de données (une pour les parrains et une pour les nouveaux étudiants) contenant les caractéristiques (âge, sexe, centres d'intérêts, campus, motivation pour rejoindre le programme de parrainage,...). Par souci de confidentialité et de RGPD, nous présenterons les résultats pour des bases de données synthétiques représentant parfaitement la réalité.Lesfigures 1 et 2 présentent quelques échantillons de données pour les deux bases. Les figures 3 et 4 présentent une description statistique élémentaire de chaque variable explicative.

	interests	study_level	age	univ_buddy_motivation	user_id	campus	sexe
0	arts	PhD	23	5	INT-00001	Campus Est	female
1	literature	PhD	20	2	INT-00002	Campus Ouest	male
2	travel	Master	25	10	INT-00003	Campus Ouest	female
3	technology	license	23	7	INT-00004	Campus Ouest	female
4	technology	license	22	5	INT-00005	Campus Central	female

FIGURE 1 – Premiers échantillons de la base de données des filleuls

	interests	study_level	age	univ_buddy_motivation	user_id	campus	sexe
0	technology	license	20	6	FR-00001	Campus Central	male
1	arts	PhD	19	6	FR-00002	Campus Nord	female
2	travel	PhD	22	10	FR-00003	Campus Sud	female
3	technology	PhD	22	8	FR-00004	Campus Central	male
4	literature	Master	25	1	FR-00005	Campus Central	female

FIGURE 2 – Premiers échantillons de la base de données des parrains

	age	univ_buddy_motivation
count	60.000000	60.000000
mean	21.966667	5.500000
std	2.216782	2.795274
min	18.000000	1.000000
25%	20.000000	3.000000
50%	22.000000	5.000000
75%	24.000000	7.250000
max	25.000000	10.000000

FIGURE 3 – Description statistique de la base de données des filleuls

	age	univ_buddy_motivation
count	60.000000	60.000000
mean	21.500000	5.666667
std	2.339781	2.784136
min	18.000000	1.000000
25%	19.750000	3.000000
50%	21.000000	6.000000
75%	24.000000	8.000000
max	25.000000	10.000000

FIGURE 4 – Description statistique de la base de données des parrains

Commentaire :

Nous pouvons remarquer que nos bases de données contiennent des données catégorielles comme le sexe, les centres d'intérêts, le niveau d'études et des données numériques comme l'âge et la motivation pour rejoindre le programme de parrainage.La moyen d'âge des participants est vers 21 ans et la motivation moyenne dépasse 5.

3 État de l'art

3.1 Définition d'une distance pertinente entre les échantillons de données

Supposons que nous disposons d'une base de données que nous notons X constituée de n lignes et p colonnes Nous notons $X^{i,j}$ la valeur de la j-ème caractéristique du i-ème échantillon de données. Nous définissons une distance entre deux échantillons i_1 et i_2 par :

$$d(X^i, X^j) = \sum_{k=1}^{p} \text{dist}(X^{i,k}, X^{j,k}) \tag{1}$$

où dist est la mesure de similarité définie selon la nature de la variable. En effet :

$$\text{dist}(X^{i,k}, X^{j,k}) = \begin{cases} \mathbb{1}_{X^{i,k} \neq X^{j,k}} \text{ si } X^{,k} \text{ categorielle} \\ \frac{|X^{i,k} - X^{j,k}|}{max(X^{,k}) - min(X^{,k})} \text{ si } X^{,k} \text{ numerique} \\ cos(X^{i,k}, X^{j,k}) \text{ si } X^{,k} \textit{ est du texte} \end{cases}$$

Remarque 1. *En général, si nous disposons de données catégorielles nous choisissons une indicatrice comme mesure. Cette dernière est pertinente car elle conserve l'interprétabilite de données catégorielles contrairement à une distance*

euclidienne généralement utilisée pour les variables numériques. Pour du texte, nous utilisons des modèles d'embeddings afin de les transformer en vecteur numérique puis nous calculons la mesure de similarité cosinus voir chapitre 1 dans [3]

Dans notre cas, nous allons calculer des distances entre les filleuls et les parrains. Sachant que les deux bases de données ont les mêmes caractéristiques (variables explicatives et taille), nous pouvons calculer $d(X^i_{filleuls}, X^j_{\text{parrains}})$ ou X_{filleuls} désigne la base de données des filleuls et $X_{parrains}$ désigne la base de données des parrains.

Dans notre étude, nous allons assumer que l'âge est une variable numérique et la motivation pour rejoindre le programme de parrainage est catégorielle. En effet, nous ne désirons pas constituer des binômes où les deux personnes ne sont pas motivées par le programme. Ceci dit nous allons fortement pondérer cette contrainte par rapport aux autres variables afin qu'elle soit toujours vérifiée.

3.2 Algorithme de Gale-Shapley

L'algorithme de Gale-Shapley (voir [2]) appelé aussi algorithme des mariages stables est une méthode permettant de donner des solutions stables à des problèmes d'appariement a savoir l'affectation des étudiants aux universités, des résidents aux hôpitaux. Pour clarifier son fonctionnement nous supposons avoir à disposition deux populations M pour les hommes et W pour les femmes. Nous considérons aussi pour chaque femme et homme une liste de préférence ordonnée et choisissons $\mathcal{R}$ comme une la relation de liaison entre un homme et une femme. Une configuration est dite instable si :

$$(\exists(m, m', w, w') \in M^2 \times W^2) \; tel \; que \begin{cases} m\mathcal{R}w \\ m'\mathcal{R}w' \\ w' >_m w \\ m >_{w'} m' \end{cases} \tag{2}$$

Concrètement cela veut dire qu'il existe au moins un binome de couples dans lequel un homme de l'un des deux couples aimerait avoir comme partenaire la femme de l'autre et cette dernière préférerait l'homme en question à son partenaire actuel. L'algorithme de Gale-Shapley est avantageux en terme de complexité temporelle cette dernière étant quadratique. Cependant, pour des problèmes d'appariement qui sont valides sur une échelle temporelle de l'ordre du mois, nous pouvons nous permettre de façon générale des temps de calcul en complexité cubique ou surcubique tant que les performances sont meilleurs.

Algorithm 1 Algorithme de Gale-Shapley

Entrées: Deux ensembles égaux M (hommes) et W (femmes), avec chacun des préférences strictement ordonnées sur l'autre ensemble

Initialisation :
Chaque homme $m \in M$ est libre.
Chaque femme $w \in W$ est libre.
Chaque homme m a une liste de femmes à qui il peut proposer, initialement toutes les femmes en ordre de préférence.
while il existe un homme libre m qui n'a pas encore proposé à toutes les femmes **do**
  $w \leftarrow$ la première femme sur la liste de m à qui m n'a pas encore proposé.
  if w est libre **then**
    m et w se fiancent.
  else
    m' est le fiancé actuel de w.
    if w préfère m à m' **then**
      m' devient libre.
      m et w se fiancent.
    end if
  end if
  Retirer w de la liste de m.
end while

Sorties: liste des couples fiancés comme la correspondance stable

Pour notre problème, nous pouvons très bien considérer deux classes F (pour Filleuls) et P (pour parrains), puis calculer une liste de préférence pour chacun basée sur le taux de compatibilité. Nous appliquons par la suite l'algorithme 2 afin de former les binomes.

Limites du modèle :

L'inconvénient des mariages stables est sa bidirectionnalité. En effet, nous prenons en compte les préférences des deux classes (parrains et filleuls). Dans un modèle réel, nous aimerions plutôt favoriser les filleuls et maximiser leur satisfaction, l'objectif du parrainage étant plus orienté vers l'intégration des filleuls dans leur nouvel environnement. Pour se faire, nous présentons dans ce qui suit l'algorithme hongrois développé par Harold Kuhn et qui permet de maximiser la compatibilité des filleuls avec leurs parrains.

3.3 Algorithme hongrois

L'algorithme hongrois est un algorithme qui permet de résoudre le problème d'affectations étant données N équipes et N taches de telles façon a minimiser la durée de travail total. Dans notre contexte, cela revient à appliquer la métrique définie plus haut pour calculer une matrice **P** dite matrice de pondération et qui vérifie :

$$(\forall i \in [\![1, N]\!])(\forall j \in [\![1, N]\!]) : \quad P_{i,j} = d(X^i_{filleuls}, X^j_{parrains})$$

Ceci dit, $P_{i,j}$ désigne une distance entre le filleul i et le parrain j. Pour créer les binomes nous pouvons résoudre le problème suivant :

$$\min_{x_{i,j} \in \{0,1\}^{N^2}} \sum_{i,j} p_{i,j} x_{i,j} \tag{3}$$

sous les contraintes suivantes :

$$(\forall i \in [\![1, N]\!]) : \sum_j x_{i,j} = 1$$

$$(\forall j \in [\![1, N]\!]) : \sum_i x_{i,j} = 1$$

Remarque 2. *Ces contraintes ont pour objectif de créer des binomes uniques,ainsi la coordonnée $x_{i,j}$ vaudra 1 si le filleul i a comme binome le parrain j et 0 dans le cas contraire.*

Nous présentons dans l'algorithme 2 un schéma de résolution du problème (3)

Algorithm 2 Algorithme Hongrois pour le problème d'affectation

Entrées : Une matrice carrée **P** de taille $n \times n$ représentant les coûts
Étape 1 : Soustraire le plus petit élément de chaque ligne de la matrice **P**
for $i = 1$ **to** n **do**
    $p_{i,j} = p_{i,j} - \min_j p_{i,j}$ pour tout j
end for
Étape 2 : Soustraire le plus petit élément de chaque colonne de la matrice résultante.
for $j = 1$ **to** n **do**
    $p_{i,j} = p_{i,j} - \min_i p_{i,j}$ pour tout i
end for
Étape 3 : Couvrir tous les zéros de la matrice résultante en utilisant un minimum de lignes horizontales et verticales.
while le nombre de lignes de couverture $< n$ **do**
    Trouver la plus petite valeur non couverte : m.
    Soustraire m de toutes les valeurs non couvertes.
    Ajouter m à toutes les valeurs aux intersections des lignes et colonnes de couverture.
end while

Étape 4 : Construire une solution optimale à partir de la matrice ajustée.
Identifier une affectation correspondant aux zéros non couverts en suivant les contraintes d'affectation unique.
Étape 5 : Vérification et itération.
if l'affectation trouvée n'est pas optimale **then**
    Ajuster la matrice et répéter depuis l'étape 3.
end if
Sorties : L'affectation optimale et le coût total minimal.

Nous pouvons montrer que le problème admettra une solution unique en utilisant la théorie d'optimisation linéaire en nombres mixtes. Cependant, nous nous contenterons de citer le chapitre 2 de [5] faisant référence aux travaux de Harold Kuhn sur le sujet.

Par rapport à notre problème, l'algorithme hongrois donne une solution optimale en complexité cubique. Cependant, il est très sensible aux paramètres. Pour expliquer mes propos nous supposons avoir a disposition deux bases de données constituées respectivement de trois parrains et trois filleuls. Nous construisons la matrice de pondération suivante et nous rappelons que cette dernière contient les distances entre parrains et filleuls :

$$\mathbf{P}_{hongrois} = \begin{pmatrix} 0.9 & 0.8 & 0.7 \\ 0.2 & 0.3 & 0.1 \\ 0.4 & 0.1 & 0.2 \end{pmatrix} \tag{4}$$

En examinant cette configuration et en résolvant l'équation 3 les binomes formés sont les suivants :

indice filleul	indice parrain
1	1
2	3
3	2

TABLE 1 – Affectations parrains/filleuls

Remarque 3. *Vu la taille du problème la résolution pourra se faire manuellement en choisissant la meilleure combinaison , cette dernière devant minimiser la distance totales.*

Limites du modèle :

Le filleul d'indice 1 représente des taux de compatibilité très faible (grande distance) avec les parrains de la base de données, cependant il s'est vu attribué le parrain d'indice 1 (celui avec lequel il matche le moins). Nous pouvons donc conclure que cette unicité de solution est une contrainte très restrictive ne permettant pas de gérer au mieux les cas particuliers. Pour remédier à cette problématique, nous nous proposons d'introduire un modèle beaucoup plus flexible qui dans un premier temps utilisera la métrique définie ci-dessus pour déterminer les plus proches voisins d'un filleul parmi la classe des parrains et dans un deuxième temps utilisera les résultats obtenus pour construire une matrice de poids où tous les proches voisins de rang égal auront la même pondération indépendemment de la valeur de la métrique. Cela permettra d'avoir une solution optimale mais non unique (sujette à un aléas) dont la mesure ou plusieurs configurations optimales sont possibles. L'avantage est de donner autant de chances aux cas particuliers qu'aux filleuls les plus privilégiés lors de la construction de binomes.

4 Contributions

4.1 Approche théorique

4.1.1 Détermination des plus proches voisins

Le choix d'une distance pertinente nous permettra de déterminer les plus proches voisins pour chaque filleul. Nous choisissons dans un premier temps de prendre les trois plus proches parrains pour chaque filleul en terme de la métrique définie ci-dessus. Nous montrons dans les figures 5 et 6 les attributions effectuées pour les 15 premiers filleuls de la base de données et le taux de compatibilité calculé vis à vis de la métrique.

	id_filleul	id_1NN_parr	id_2NN_parr	id_3NN_parr
0	INT-00001	FR-00021	FR-00043	FR-00045
1	INT-00002	FR-00041	FR-00050	FR-00010
2	INT-00003	FR-00049	FR-00042	FR-00059
3	INT-00004	FR-00058	FR-00055	FR-00032
4	INT-00005	FR-00001	FR-00040	FR-00004
5	INT-00006	FR-00004	FR-00050	FR-00001
6	INT-00007	FR-00019	FR-00031	FR-00012
7	INT-00008	FR-00039	FR-00030	FR-00014
8	INT-00009	FR-00020	FR-00046	FR-00012
9	INT-00010	FR-00048	FR-00017	FR-00004
10	INT-00011	FR-00011	FR-00012	FR-00049
11	INT-00012	FR-00041	FR-00025	FR-00024
12	INT-00013	FR-00002	FR-00053	FR-00020
13	INT-00014	FR-00060	FR-00050	FR-00045
14	INT-00015	FR-00008	FR-00019	FR-00016
15	INT-00016	FR-00042	FR-00034	FR-00013

FIGURE 5 – Attribution des trois plus proches parrains aux filleuls

	id_filleul	matching_score_1NN	matching_score_2NN	matching_score_3NN
0	INT-00001	0.968254	0.873016	0.857143
1	INT-00002	1.000000	0.873016	0.825397
2	INT-00003	0.984127	0.936508	0.904762
3	INT-00004	0.857143	0.857143	0.857143
4	INT-00005	0.968254	0.952381	0.888889
5	INT-00006	0.984127	0.888889	0.873016
6	INT-00007	0.904762	0.888889	0.888889
7	INT-00008	0.968254	0.873016	0.873016
8	INT-00009	0.904762	0.888889	0.888889
9	INT-00010	0.936508	0.888889	0.888889
10	INT-00011	1.000000	0.984127	0.888889
11	INT-00012	0.873016	0.873016	0.857143
12	INT-00013	0.888889	0.888889	0.888889
13	INT-00014	0.888889	0.888889	0.888889
14	INT-00015	0.936508	0.920635	0.888889
15	INT-00016	0.873016	0.873016	0.873016

FIGURE 6 – Taux de compatibilité entre les trois plus proches parrains et les filleuls

Remarque 4. *Le terme 'NN' ou 'nearest neighbors' fait référence au plus proche voisin et les termes 'Fr' et 'INT' désigne respectivement France et International et font référence aux parrains et filleuls.*

En examinant le tableau des plus proches voisins, nous pouvons remarquer qu'il existe des doublons sur une même colonne de choix. Ceci dit, l'attribution manuelle des parrains peut être une tâche compliquée surtout pour des bases de données de grande de taille. Nous rappelons que nous travaillons sur soixante échantillons de données (60 filleuls et 60 parrains) et qu'en réalité nous avons des données de taille plus importante. Pour automatiser le processus, nous allons utiliser des techniques d'optimisation sous contraintes et des algorithmes de type maximisation de taux de satisfaction que nous introduirons dans la section suivante.

4.1.2 Maximisation du taux de satisfaction

L'algorithme de maximisation de satisfaction se base sur la pondération des plus proches voisins selon leur importance. Prenons l'exemple suivant où nous supposons que nous disposons de données de trois filleuls et trois parrains. Nous présentons dans le tableau 2 une répartition hypothétique de plus proches voisins supposée calculée sur la base de la distance définie ci-dessus .

id filleul	id 1NN parr	id 2NN parr	id 3NN parr
1	2	3	1
2	2	1	3
3	1	3	2

TABLE 2 – Exemple de calcul de plus proches voisins en dimension réduite

En examinant le tableau 2, nous pouvons pondérer le plus proches voisins avec un coefficient 0.5 , le deuxième plus proche avec un coefficient moins élevé (0.3 par exemple) et le troisième avec un coefficient de 0.2. La pondération est donc une fonction décroissante de la distance. Considérons dorénavant pour chaque filleul d'identifiant j un vecteur $x_j = (x_{j,1}, x_{j,2}, x_{j,3})$. La composante i du vecteur désigne la probabilité d'attribution du i-ème plus proche voisin au filleul j , l'espace de probabilité étant bien évidemment défini sur l'ensemble des filleuls ayant le i-ème plus proche voisin en question autant que voisin (non nécessairement i-ème). En se basant sur l'exemple ci-dessus, nous pouvons définir la matrice de pondération que nous noterons $\mathbf{P}$ et le vecteur des probabilités que nous noterons x par :

$$x = (x_{1,1}, x_{1,2}, x_{1,3}, x_{2,1}, x_{2,2}, x_{2,3}, x_{3,1}, x_{3,2}, x_{3,3})^T$$

$$\mathbf{P} = \begin{pmatrix} 0 & 0 & 0.2 & 0 & 0.3 & 0 & 0.5 & 0 & 0 \\ 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0.2 \\ 0 & 0.3 & 0 & 0 & 0 & 0.2 & 0 & 0.3 & 0 \end{pmatrix}$$

Remarque 5. *La matrice $\boldsymbol{P}$ est de taille (n, m) avec n le nombre de parrains et m le nombre de choix ou plus proches voisin (égal à 3 dans l'exemple) multiplié par le nombre de filleuls.*

Nous définissons aussi la matrice de contrainte $\mathbf{C}$ exprimant le caractère probabiliste des composantes du vecteur x et qui vérifie :

$$(\forall i \in [\![1, n]\!])(\forall j \in [\![1, m]\!]) : C_{i,j} = \begin{cases} 1 \text{ si le } i-\text{eme parrain est un plus proche de voisin du filleul} \\ 0 \text{ sinon} \end{cases}$$

Dans notre exemple cela revient à remplacer les coefficients non nuls de la matrice de pondération par 1 c'est à dire :

$$\mathbf{C} = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Le problème d'optimisation s'exprime alors comme suit :

$$\hat{x} \in \underset{x \in \mathbb{R}^m}{Argmax} \frac{1}{2} ||\mathbf{P}x||_2^2 \ \ (s.c \ \ \mathbf{C}x = z \text{ et } x \geq 0) \tag{5}$$

Remarque 6. *Nous avons choisi la norme l2 compte tenu de ces propriétés de convexité et de classe C^1. z désigne le vecteur colonne tel que z_j vaut 1 si le parrain j est proche voisin d'au moins un filleul, 0 sinon. La contrainte*

exprime que les probabilités d'attribution d'un certain parrain somment à 1. Nous pouvons particulièrement remarquer que si un parrain j n'est un proche voisin d'aucun des filleuls alors la j-ème ligne de la matrice de pondération $\boldsymbol{P}$ *et de la matrice de contrainte* $\boldsymbol{C}$ *est identiquement nulle.*

Formulation du besoin en problème d'optimisation :

Nous supposons avoir déterminé les trois plus proches voisins pour chaque filleul et construit la matrice de pondération $\mathbf{P}$ avec les coefficients 0.5 (pour le plus proche voisin) , 0.3 (pour le deuxième plus proche voisin) et 0.2 (pour le troisième plus proche voisin) .

Nous cherchons :

$$\hat{x} \in \underset{x\in\mathbb{R}^m}{Argmax}\ \frac{1}{2}||\mathbf{P}x||_2^2\ \ (s.c\ \ \mathbf{C}x = z\ et\ x \geq 0)$$

Commençons tout d'abord par reformuler ce problème de maximisation en problème de minimisation ce qui nous permettra en pratique d'utiliser des algorithmes d'optimisation d'ordre 1 de type descente de gradient. Pour se faire, remarquons que sous les contraintes $\mathbf{C}x = z$ et $x \geq 0$:

$$(\forall j \in [\![1, n]\!])\ (\mathbf{P}x)_j \leq 0.5 \tag{6}$$

En effet, pour un parrain j étant un proche voisin d'au moins un filleul nous avons :

$$(\mathbf{P}x)_j = \sum_{i=1}^{k_j} a_i x_i^j \Rightarrow (\mathbf{P}x)_j \leq (\underset{i\in[\![1,k_j]\!]}{max}\ a_i)(\sum_{i=1}^{k_j} x_i^j)$$

avec :

$$(\forall i \in [\![1, k_j]\!])\ a_i \in \{0.5, 0.3, 0.2\}$$

$$\sum_{i=1}^{k_j} x_i^j = 1$$

Par conséquent :

$$(5) \Leftrightarrow \hat{x} \in \underset{x\in\mathbb{R}^m}{Argmin}\ \frac{1}{2}||0.5 \times \mathbb{1}_{\mathbb{R}^n} - \mathbf{P}x||_2^2\ \ (s.c\ \ \mathbf{C}x = z\ et\ x \geq 0)$$

Pour des raisons de simplification, nous poserons dans ce qui suit : $y = 0.5 \times \mathbb{1}_{\mathbb{R}^n}$

Optimisation sous contraintes : théorie et méthodes de résolution

Dans cette section, nous présenterons dans un premier temps le cadre théorique pour la résolution du problème et dans un deuxième temps les techniques

d'optimisation spécifiques aux problèmes contraints. Nous considérons le problème général suivant :

$$\min_{x\in\mathbb{R}^m} \frac{1}{2}||y-\mathbf{P}x||_2^2 \ \ (s.c\ \ \mathbf{C}x = z\ et\ x \geq 0) \tag{7}$$

Montrons tout d'abord que (7) est bien défini c'est à dire qu'il existe une solution au problème en question.

Proposition 1. *Soient* $(n,m) \in (\mathbb{N}^*)^2$ *tq* $m > n$, $y \in \mathbb{R}^n$ *et* $z \in \mathbb{R}^n$. *Nous posons* $\mathcal{A} = \{w \in \mathbb{R}^m \ tq\ \boldsymbol{C}w = z\} \cap (\mathbb{R}^+)^m$ *alors :*

$$(\exists \hat{x} \in \mathcal{A}) : \frac{1}{2}||y-\boldsymbol{P}\hat{x}||_2^2 = \min_{x\in\mathcal{A}} \frac{1}{2}||y-\boldsymbol{P}x||_2^2$$

Démonstration. Pour la preuve nous nous inspirons [1] qui contient différentes techniques de l'analyse convexe en général et l'optimisation convexe en particulier.

$$(\forall x \in \mathcal{A}) : \frac{1}{2}||y-\mathbf{P}x||_2^2 \geq 0$$

On peut donc poser $a = \inf_{x\in\mathcal{A}} \frac{1}{2}||y-\mathbf{P}x||_2^2$

Montrons que $a = \min_{x\in\mathcal{A}} \frac{1}{2}||y-\mathbf{P}x||_2^2$ ou en d'autres termes :

$(\exists x^* \in \mathcal{A})$ *tel que* $a = \frac{1}{2}||y-\mathbf{P}x^*||_2^2$

Pour se faire on exploite la caractéristique séquentielle de l'infimum . Considérons alors : $(x_k)_{k\geq 0}$ une suite d'éléments de $\mathcal{A}$ tel que :

$$\lim_{k\to\infty} \frac{1}{2}||y-\mathbf{P}x_k||_2^2 = a$$

La suite $(x_k)_{k\geq 0}$ est bornée. En effet :

$$(\forall y \in \mathcal{A}) : \ ||y||_\infty \leq ||z||_\infty$$

Remarquons que $\mathcal{A}$ est un fermé (car intersection de deux fermés) et borné en dimension finie. Ainsi en appliquant le théorème de Bolzano-Weirstrass, il existe ϕ extractrice (application de $\mathbb{N}$ vers $\mathbb{N}$ strictement croissante) tel que : $(x_{\phi(k)})_{k\geq 0}$ converge vers un certain x^*. Par conséquent :

$$\lim_{k\to\infty} \frac{1}{2}||y-\mathbf{P}x_{\phi(k)}||_2^2 = \frac{1}{2}||y-\mathbf{P}x^*||_2^2 = a$$

Nous avons exploité la continuité de l'application $x \to \frac{1}{2}||y-\mathbf{P}.||_2^2$ et la propriété de convergence d'une sous suite d'une suite convergente.

□

Nous cherchons dans ce qui suit :

$$x^* \in \underset{x\in\mathcal{A}}{Argmin}\ \frac{1}{2}||y-\mathbf{P}x||_2^2 \tag{8}$$

Un tel x^* existe d'après la proposition 1.

Optimisation sous contraintes : pénalisation quadratique [4]

Pour résoudre le problème (8), nous utiliserons une pénalisation quadratique (voir chapitre 10) dans [4] pour traduire les contraintes d'égalités et d'inégalités. Soit $(\mu_k)_{k\geq 0}$ une suite strictement croissante de réels strictement positifs qui tend vers $+\infty$. Nous définissons la suite de problèmes suivante ;

$$(P_k) : x_k \in \underset{x\in\mathbb{R}^m}{\text{Argmin}} \frac{1}{2}||y - \mathbf{P}x||_2^2 + \frac{1}{2}\mu_k(||\mathbf{C}x - z||_2^2 + ||x^-||_2^2) \tag{9}$$

avec $x^- = min(0, x_j)_{j\in[\![1,m]\!]}$

Remarque 7. *L'idée des pénalisations est de rapprocher de plus en plus la solution obtenue du domaine admissible au fur et à mesure qu'on augmente k. En effet, plus k augmente plus le coefficient de régularisation μ_k est grand et les contraintes prépondérent. Nous arrêtons l'algorithme quand on obtient une solution satisfaisante.*

Montrons tout d'abord qu'un tel x_k existe. Pour se faire nous commençons par poser :

$$(\forall x \in \mathbb{R}^m)(\forall \mu \in \mathbb{R}^+) : \ F(x,\mu) = \frac{1}{2}||y - \mathbf{P}x||_2^2 + \frac{1}{2}\mu(||\mathbf{C}x - z||_2^2 + ||x^-||_2^2)$$

Proposition 2. *Soit $\mu \in \mathbb{R}^{+*}$, la fonction $F(.,\mu) : x \to F(x,\mu)$ est alors coercive et continue sur $\mathbb{R}^m$*

Démonstration. Soit $\mu \in \mathbb{R}^{+*}$. Montrons que : $\lim\limits_{||x||\to+\infty} F(x,\mu) = +\infty$

Soit $x \in \mathbb{R}^m$:

Nous avons $x = x^- + x^+$ avec $x^- = min(0, x_j)_{j\in[\![1,m]\!]}$ et $x^+ = max(0, x_j)_{j\in[\![1,m]\!]}$

De plus : $||x||_2^2 = ||x^-||_2^2 + ||x^+||_2^2$. Dans la suite de la démonstration nous noterons $|||\mathbf{P}|||_2$ la norme d'opérateur induite par la norme euclidienne et associée à $\mathbf{P}$. Nous cherchons dans ce qui suit une inégalité liant F, $||x^-||_2^2$ et $||x^+||_2^2$

$$\begin{aligned}
\frac{1}{2}||y - \mathbf{P}x||_2^2 + \frac{1}{2}\mu(||\mathbf{C}x - z||_2^2 + ||x^-||_2^2) &\geq \frac{1}{2}||y - \mathbf{P}x||_2^2 + \frac{1}{2}\mu||x^-||_2^2 \\
&\geq \frac{1}{2}||y - (\mathbf{P}x^+ + \mathbf{P}x^-)||_2^2 + \frac{1}{2}\mu||x^-||_2^2 \\
&\geq -\frac{1}{2}||y||_2^2 + \frac{1}{2}||(\mathbf{P}x^+ + \mathbf{P}x^-)||_2^2 + \frac{1}{2}\mu||x^-||_2^2 \\
&\geq -\frac{1}{2}||y||_2^2 + \frac{1}{2}||\mathbf{P}x^+||_2^2 - ||\mathbf{P}x^-||_2^2 + \frac{1}{2}\mu||x^-||_2^2 \\
&\geq -\frac{1}{2}||y||_2^2 + \frac{1}{2}||\mathbf{P}x^+||_2^2 - |||\mathbf{P}|||_2^2||x^-||_2^2 + \frac{1}{2}\mu||x^-||_2^2 \\
&\geq -\frac{1}{2}||y||_2^2 + \frac{1}{2}||\mathbf{P}x^+||_2^2 + \frac{1}{2}(\mu - |||\mathbf{P}|||_2^2)||x^-||_2^2 \\
&\geq \frac{1}{2}(0.2)^2 \times ||x^+||_2^2 + \frac{1}{2}(\mu - |||\mathbf{P}|||_2^2)||x^-||_2^2 - \frac{1}{2}||y||_2^2
\end{aligned}$$

Nous avons utilisé :

$$||\mathbf{P}x^+||_2^2 \geq (0.2)^2||x^+||_2^2$$

Remarque 8. *Cette inégalité est vraie car* $\boldsymbol{P}$ *et* x^+ *sont à coefficients positifs. De plus, chaque coordonnée du vecteur* x^+ *est associée à un coefficient de pondération appartenant* $\{0.5, 0.3, 0.2\}$*. Nous pouvons particulièrement remarquer que le raisonnement reste valable peu importe le nombre de plus proches voisins ou bien les coefficients de pondérations choisis.*

Nous avons alors :

$$F(x,\mu) \geq \frac{1}{2}(0.2)^2 \times ||x^+||_2^2 + \frac{1}{2}(\mu - |||\mathbf{P}|||_2^2)||x^-||_2^2 - \frac{1}{2}||y||_2^2 \tag{10}$$

De plus :

$$F(x,\mu) \geq \frac{1}{2}\mu||x^-||_2^2$$

Il s'en suit que :

$$\frac{|||\mathbf{P}|||_2^2}{\mu}F(x,\mu) \geq \frac{1}{2}|||\mathbf{P}|||_2^2||x^-||_2^2 \tag{11}$$

En combinant (10) et (11) nous obtenons :

$$\begin{aligned}(\frac{|||\mathbf{P}|||_2^2}{\mu} + 1)F(x,\mu) + \frac{1}{2}||y||_2^2 &\geq \frac{1}{2}(0.2)^2 \times ||x^+||_2^2 + \frac{1}{2}\mu||x^-||_2^2 \\ &\geq \frac{1}{2}min(0.04,\mu) \times (||x^+||_2^2 + ||x^-||_2^2) \\ &\geq \alpha||x||_2^2\end{aligned}$$

Nous retenons que :

$$(\forall x \in \mathbb{R}^m)(\forall \mu \in \mathbb{R}^+) : \ (\frac{|||\mathbf{P}|||_2^2}{\mu} + 1)F(x,\mu) + \frac{1}{2}||y||_2^2 \geq \frac{1}{2}min(0.04,\mu) \times ||x||^2 \tag{12}$$

Nous pouvons alors déduire que $F(.,\mu)$ est coercive. $F(.,\mu)$ est continue comme somme et composées de fonctions continues, elle admet alors un minimiseur.

□

Dans ce qui suit, nous montrerons que la suite $(F(x_k,\mu_k))_{k\geq 0}$ converge bien vers $\underset{x\in\mathcal{A}}{min}\frac{1}{2}||y - \mathbf{P}x||_2^2$.

Remarquons tout d'abord que :

$$(\forall x \in \mathcal{A}) : F(x,\mu) = \frac{1}{2}||y - \mathbf{P}x||_2^2$$

Ceci dit :

$$(\forall k \geq 0) : \ F(x_k,\mu_k) \leq \underset{x\in\mathcal{A}}{min}\frac{1}{2}||y - \mathbf{P}x||_2^2 \tag{13}$$

La suite $(F(x_k,\mu_k))_{k\geq 0}$ est croissante. En effet, $(u_k)_{k\geq 0}$ est strictement croissante et :

$$\begin{aligned}(\forall x\in\mathbb{R}^m)(\forall \mu,\nu\in(\mathbb{R}^{+*})^2)\ : \mu<\nu &\Rightarrow F(x,\mu)<F(x,\nu)\\ &\Rightarrow \min_{x\in\mathbb{R}^m} F(x,\mu)\leq \min_{x\in\mathbb{R}^m} F(x,\nu)\end{aligned}$$

Il s'en suit que $(F(x_k,\mu_k))_{k\geq 0}$ est croissante et majorée donc convergente.

Montrons que sa limite est $\min_{x\in\mathcal{A}}\frac{1}{2}||y-\mathbf{P}x||_2^2$.

Remarquons dans un premier temps que la suite $(x_k)_{k\geq 0}$ est bornée. En effet, en exploitant (12), (13) et la stricte croissance de $(\mu_k)_{k\geq 0}$ nous pouvons écrire :

$$(\forall k\in\mathbb{N}):\ (\frac{|||\mathbf{P}|||_2^2}{\mu_0}+1)(\min_{x\in\mathcal{A}}\frac{1}{2}||y-\mathbf{P}x||_2^2)+\frac{1}{2}||y||_2^2\geq 0.02\times||x_k||^2$$

Nous considérons ϕ une extraction (application strictement croissante de $\mathbb{N}$ vers lui même) tel que $(x_{\phi_k})_{k\geq 0}$. Une telle application existe d'après le théorème de Bolzano-Weirstrass car $(x_k)_{k\geq 0}$ est bornée en dimension finie. Nous notons alors $x_{lim}=\lim_{k\to+\infty} x_{\phi_k}$. Montrons tout d'abord que $x_{lim}\in\mathcal{A}$.

On suppose par l'absurde que que x_{lim} n'est pas dans $\mathcal{A}$ alors $||\mathbf{C}x_{lim}-z||_2^2+||x_{lim}^-||_2^2>0$ et :

$$\begin{aligned}\lim_{k\to+\infty} F(x_{\phi_k},\mu_{\phi_k}) &= \frac{1}{2}||y-\mathbf{P}x_{lim}||_2^2+\frac{1}{2}\lim_{k\to+\infty}\mu_{\phi_k}(||\mathbf{C}x_{lim}-z||_2^2+||x_{lim}^-||_2^2)\\ &= +\infty\ (\textit{contradiction})\end{aligned}$$

La contradiction relève de la majoration de $(F(x_k,\mu_k))_{k\geq 0}$ (voir équation (13)).

Utilisons alors ce résultat pour prouver la convergence de $(F(x_k,\mu_k))_{k\geq 0}$.

$$\begin{aligned}x_{lim}\in\mathcal{A}\Rightarrow \frac{1}{2}||y-\mathbf{P}x_{lim}||_2^2 &\geq \min_{x\in\mathcal{A}}\frac{1}{2}||y-\mathbf{P}x||_2^2\\ \Rightarrow \lim_{k\to+\infty} F(x_{\phi_k},\mu_{\phi_k}) &\geq \min_{x\in\mathcal{A}}\frac{1}{2}||y-\mathbf{P}x||_2^2\\ \Rightarrow \lim_{k\to+\infty} F(x_k,\mu_k)=\lim_{k\to+\infty} F(x_{\phi_k},\mu_{\phi_k}) &= \min_{x\in\mathcal{A}}\frac{1}{2}||y-\mathbf{P}x||_2^2\end{aligned}$$

Remarque 9. *Remarquons que la limite de $(F(x_{\phi_k},\mu_{\phi_k}))_{k\geq 0}$ existe bien autant que sous-suite de $(F(x_k,\mu_k))_{k\geq 0}$ et les deux suites convergent vers la même limite (par separabilité de $\mathbb{R}$). De plus, en combinant (13) avec la minoration de $\lim_{k\to+\infty} F(x_{\phi_k},\mu_{\phi_k})$ nous obtenons le résultat.*

Jusqu'à présent, nous avons montré que $(F(x_k,\mu_k))_{k\geq 0}$ converge vers $\min_{x\in\mathcal{A}}\frac{1}{2}||y-\mathbf{P}x||_2^2$ et que l'ensemble des valeurs d'adhérence de $(x_k)_{k\geq 0}$ sont dans $\mathcal{A}$.

De plus, si une solution $x_k \in \mathcal{A}$ alors $F(x_k, \mu_k) = \min_{x \in \mathcal{A}} \frac{1}{2}||y - \mathbf{P}x||_2^2$.

Descente de gradient : algorithme et métriques

Nous présenterons dans cette partie l'algorithme de descente de gradient à pas variable que nous allons utiliser pour la résolution du problème 7. Commençons tout d'abord par calculer le gradient de $F(., \mu)$ pour $\mu > 0$.

Soit $x \in \mathbb{R}^m$,$\mu \in \mathbb{R}^{+*}$:

$$\begin{aligned}\nabla F(x, \mu) &= \frac{1}{2}(\nabla||y - \mathbf{P}.||_2^2 + \mu(\nabla||\mathbf{C}. - z||_2^2 + \nabla||.^-||_2^2))(x) \\ &= \mathbf{P}^T(\mathbf{P}x - y) + \mu(\mathbf{C}^T(\mathbf{C}x - z) + x^-) \\ &= (\mathbf{P}^T\mathbf{P} + \mu\mathbf{C}^T\mathbf{C})x - \mathbf{P}^T y - \mu\mathbf{C}^T z + x^-\end{aligned}$$

L'algorithme de descente de gradient à pas variable se base principalement sur la condition d'Armijo (14). En effet, l'idée est de chercher un itéré sur une direction de descente qui assure une décroissance de la fonction objectif.

$$F(x_k + \alpha_k p_k, \mu) \leq F(x_k, \mu) + c_1 \alpha_k p_k \nabla F(x_k, \mu) \tag{14}$$

— p_k est la direction de descente à l'itération k.
— c_1 est la constante d'Armijo.

Remarque 10. *En procédant ainsi, nous obtenons bien un algorithme monotone car p_k étant une direction de descente la condition $\langle p_k, \nabla F(x_k, \mu)\rangle \leq 0$ est vérifiée. Un choix qui parait évident pour p_k serait $-\nabla F(x_k, \mu)$.*

Nous présentons dans l'algorithme 3 le schéma de résolution d'un problème d'optimisation général en utilisant la condition d'Armijo. Nous choisirons comme métrique de convergence l'observation de la décroissance vers 0 de $\nabla F(., \mu)$. Nous présentons aussi dans l'algorithme 4 le schéma de recherche de la solution du problème 7 à partir d'une descente de gradient à pas variable.

Algorithm 3 Schéma algorithmique de la descente de gradient à pas variable

Entrées: x_{init}, μ, $\epsilon > 0$, $c_1 < 1$, $\beta < 1$, $\alpha > 0$,$j_{max} \in \mathbb{N}^*$

$x_0 \leftarrow x_{init}$
$j \leftarrow 0$
while $j \leq j_{max}$ and $\nabla F(x_j, \mu) \geq \epsilon$ **do**
  $\alpha_j \leftarrow \alpha$
  $p_j \leftarrow -\nabla F(x_j, \mu)$
  $x_{j+1} \leftarrow x_j - \alpha_j \nabla F(x_j, \mu)$
  while $F(x_{j+1}, \mu) > F(x_j, \mu) + c_1 \alpha_j p_j \nabla F(x_j, \mu)$ **do**
    $\alpha_j \leftarrow \beta \times \alpha_j$ {Nous diminuons α_j si la condition n'est pas vérifiée}
    $x_{j+1} \leftarrow x_j - \alpha_j \nabla F(x_j, \mu)$ {Mise à jour de x_{j+1}}
  end while
end while

Sorties: $x^{optim} \leftarrow x_{j+1}$

Remarque 11. *L'algorithme 3 permet de trouver un minimiseur de $F(.,\mu)$. Nous calculons x_{j+1} en fonction de x_j par recherche sur une direction de descente. Nous allons exploiter cette architecture dans l'algorithme 4 pour calculer à chaque fois un minimiseur de $F(.,\mu_k)$.*

Algorithm 4 Schéma algorithmique pour la résolution du problème d'optimisation sous contraintes

Entrées: μ_0, $\beta < 1$, $\alpha > 0$, $c_1 < 1$, k_{max},x_{init}, ϵ

1: $k \leftarrow 0$
2: On calcule $x_0^{optim} \in \underset{x\in\mathbb{R}^m}{Argmin} F(.,\mu_0)$ en utilisant l'algorithme 3 et en initialisant avec x_{init}
3: **while** $k \leq k_{max}$ and $dist(x_k^{optim}, \mathcal{A}) \geq \epsilon$ **do**
4: On calcule $x_{k+1}^{optim} \in \underset{x\in\mathbb{R}^m}{Argmin} F(.,\mu_{k+1})$ en utilisant l'algorithme 3 et en initialisant avec x_k^{optim}
5: $k \leftarrow k+1$
6: **end while**

Sorties: $x^* \leftarrow x_k^{optim}$

Remarque 12. *Nous avons utilisé comme métrique de convergence numérique la distance séparant les estimées des vecteurs de probabilités x_k au domaine admissible. Comme nous l'avons démontré plus haut, si la distance est nulle alors x_k est une solution faisable du problème d'optimisation sous contraintes sinon nous avons convergence théorique vers 0 quand k tend vers $+\infty$.*

Attribution des parrains :

Considérons $x^* \in \underset{x\in\mathcal{A}}{Argmin} \frac{1}{2}||y - \mathbf{P}x||_2^2$ calculé à partir de l'algorithme 4.

Le vecteur x^* est de taille $3 \times n$ où n désigne le nombre de filleuls. On peut alors le redimensionner sous forme d'une matrice $X \in \mathcal{M}_{n,3}(\mathbb{R})$ où $X_{i,j}$ désigne la probabilité d'attribuer le j-ème proche voisin du filleul i à ce dernier.

Procédure d'attribution :

Nous notons $\mathcal{P}$ l'ensemble des identifiants des parrains étant proche voisin d'au moins un filleul.

Soit $j \in \mathcal{P}$:

Nous posons $F_j = \{j_1, j_2, ..j_k\}$ l'ensemble des identifiants des k filleuls ayant le parrain identifié par j comme proche voisin et nous posons $\{r_{j_1}, r_{j_2}, ...r_{j_k}\}$ l'ensemble des rangs du parrain en question pour chaque filleul. Nous avons alors :

$$(\forall i \in [\![1,k]\!]) : \ r_{j_i} \in [\![1,3]\!]$$

Pour attribuer le parrain d'identifiant j nous cherchons :

$$i^* \in \underset{i\in[\![1,k]\!]}{Argmax}\{X_{j_i,r_{j_i}} \ tel \ que \ i \in [\![1,k]\!]\}$$

Nous attribuons alors le parrain identifié par j au filleul d'identifiant j_{i^*} et éliminons la pair de l'ensemble des parrains/filleuls. Nous pouvons particulièrement remarquer qu'en théorie une certaine cohérence s'établit dans la mesure

où les probabilités les plus pondérées seront les plus importantes. Cependant, nous ne sommes pas sûrs d'attribuer tous les parrains (outliers qui ne matchent avec aucun filleul dans le sens de la distance définie et qui ne figurent pas dans le tableau représenté dans la figure 5). Il pourrait éventuellement y avoir des parrains et des filleuls qui n'ont pas eu du binôme vu la configuration du problème. Nous verrons dans la section résultats numériques différentes méthodes pour remédier à cette problématique et avoir une automatisation totale du processus.

5 Résultats numériques

Dans un premier temps, nous vérifierons la convergence de nos algorithmes puis nous appliquerons l'algorithme d'attribution de parrains et vérifierons le pourcentage d'automatisation. Enfin, nous présenterons quelques méthodes pour converger vers une automatisation totale

5.1 Vérification de la convergence de l'algorithme

Dans cette partie nous vérifierons la convergence de l'algorithme 3 de descente de gradient à pas variable et celle de l'algorithme 4 de résolution du problème sous-contraintes. Pour se faire, nous commençons par définir une suite croissante de valeurs du coefficient de régularisation. Nous allons choisir une suite $(\mu_k)_{k\geq 0}$ géométrique de raison 3. Nous visualisons dans les figure 7 , figure 8 et figure 9 respectivement l'évolution du gradient , de la fonction objectif et des accroissements sur la fonction objectif pour différentes valeurs de μ.

Choix des paramètres :

Nous choisissons comme paramètres d'entrée un coefficient de rebroussement $\beta = 0.9$, une constante d'Armijo $c_1 = 0.1$,un seuil de convergence $\epsilon = 10^{-2}$ pour la norme du gradient et enfin un nombre d'itérations maximales $j_{max} = 1000$. Nous initialisons avec $\mu_0 = 1$.

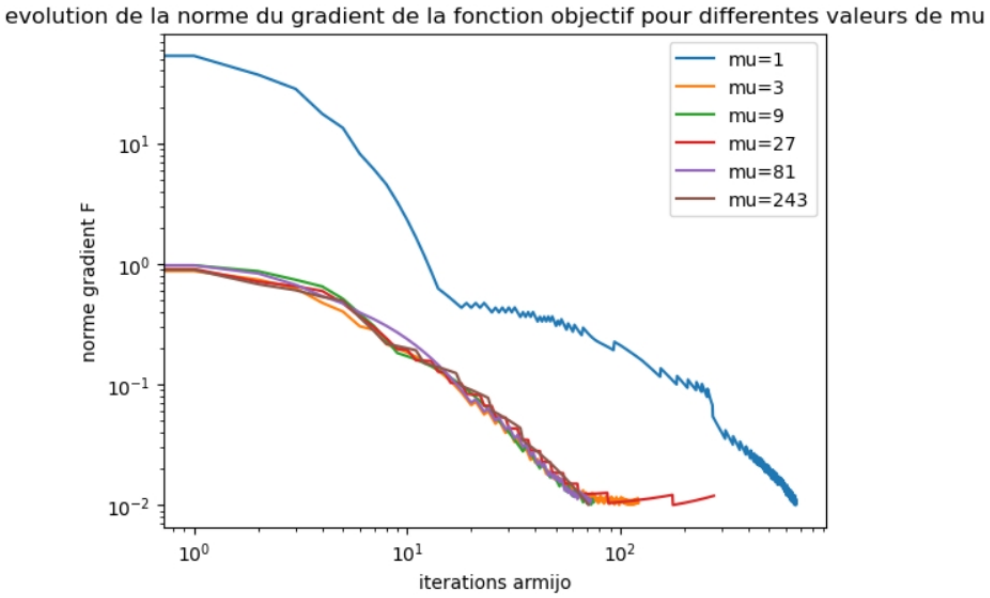

FIGURE 7 – Évolution de la norme du gradient en fonction du nombre d'itérations pour différentes valeurs de μ

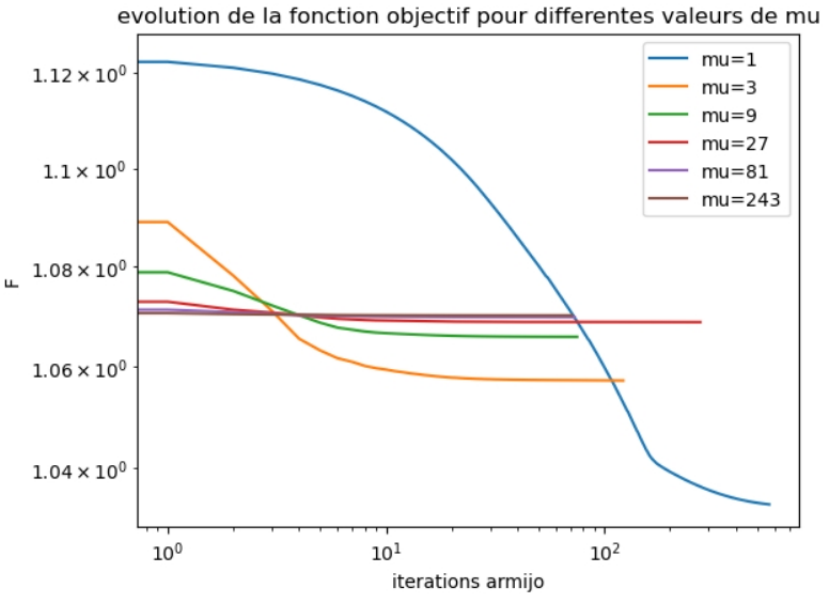

FIGURE 8 – Évolution de la fonction objectif en fonction du nombre d'itérations pour différentes valeurs de μ

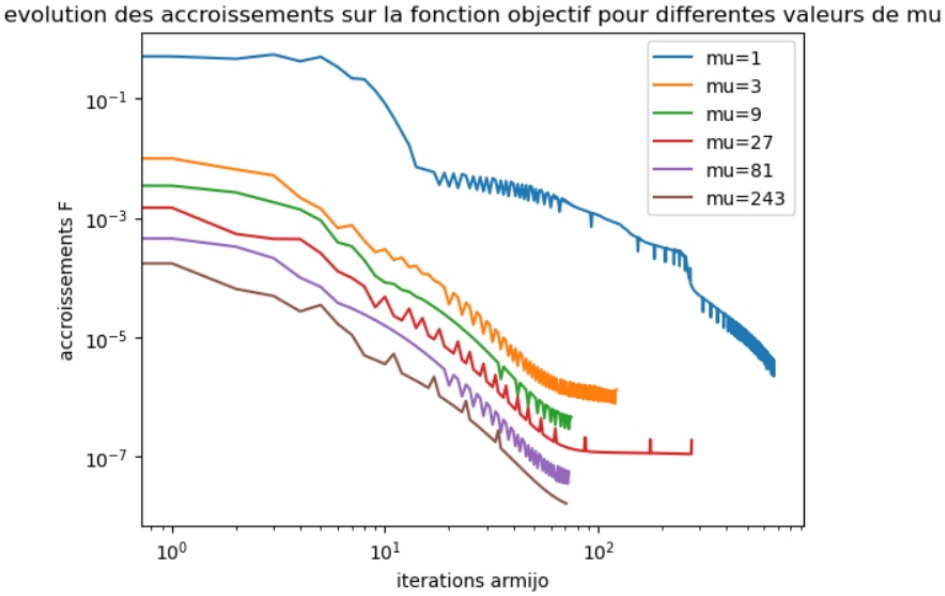

FIGURE 9 – Évolution des accroissements sur la fonction objectif en fonction du nombre d'itérations pour différentes valeurs de μ

Commentaire :

Pour toutes les valeurs du coefficient de régularisation μ explorées,la norme du gradient de la fonction F converge vers 0. De plus, la fonction objectif décroît fortement (car la condition d'Armijo s'applique) et converge vers un plateau de saturation. Nous remarquons aussi que les accroissements décroissent vers 0. Nous pouvons donc conclure que l'algorithme de descente de gradient à pas variable est fonctionnel. Nous vérifierons dans la suite la convergence de l'algorithme général de résolution du problème sous contraintes.

Nous traçons dans les figures 10 et 11 respectivement l'évolution de la distance des estimées des vecteurs de probabilité x_k au domaine admissible en fonction de μ_k et l'évolution de $F(x_k, \mu_k)$ en fonction de μ_k.

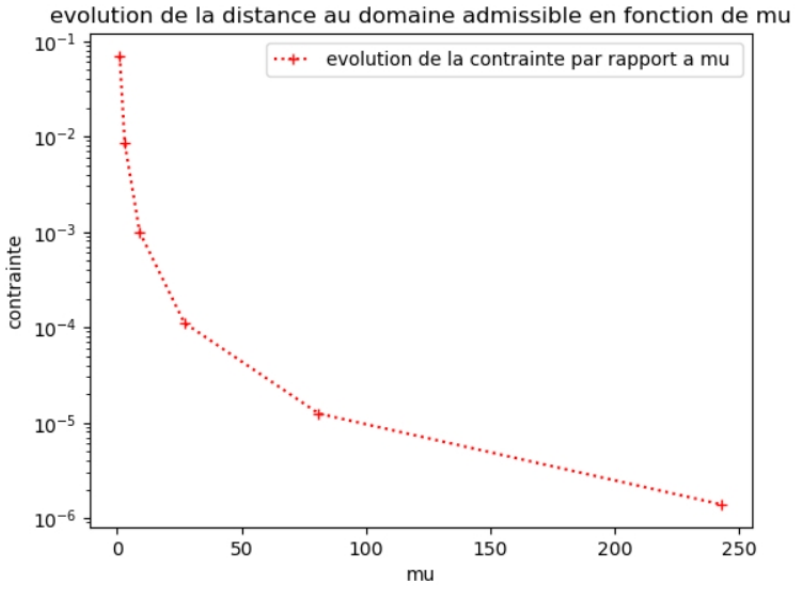

FIGURE 10 – évolution de la distance des estimées des vecteurs de probabilité x_k au domaine admissible en fonction de μ_k

Commentaire :

La distance entre les estimées des vecteurs de probabilité x_k et le domaine admissible décroît vers le seuil de convergence que nous avons fixé à 10^{-6} ce qui témoigne d'une convergence vers 0 de cette dernière. Nous remarquons aussi que l'algorithme parcours 5 valeurs de la suite $(\mu_k)_{k\geq 0}$ avant de s'arrêter.

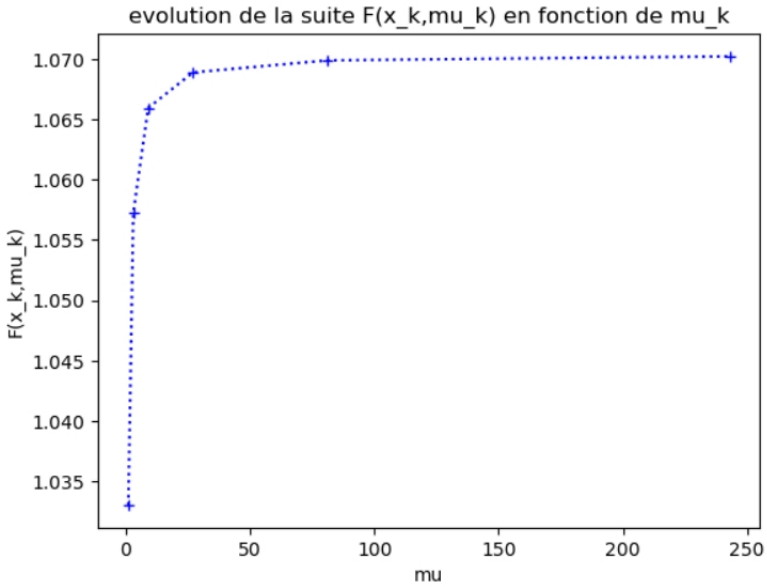

FIGURE 11 – évolution de $F(x_k, \mu_k)$ en fonction de μ_k

Commentaire :

Nous remarquons que la suite $(F(x_k, \mu_k))_{k\geq 0}$ est bien croissante. Le plateau de saturation qui se forme témoigne d'une convergence vers une valeur limite qui se place autour de 1.07. Nous tracerons dans la figure 12 le vecteur des probabilités x^* obtenu.

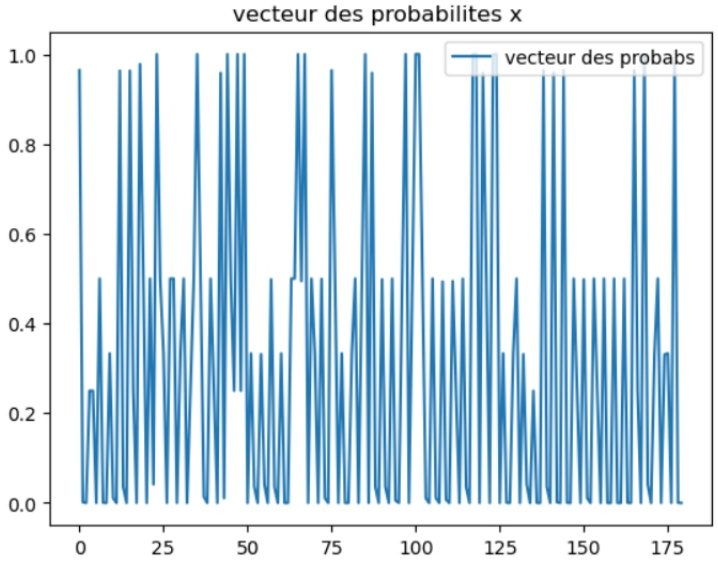

FIGURE 12 – vecteur de probabilité d'attribution des parrains

5.2 Création des pairs et taux de compatibilité

Dorénavant, nous exploiterons le vecteur des probabilités x^* pour effectuer les attributions selon un processus de maximisation de la compatibilité. Nous montrons dans la figure 13 les binômes formés ainsi que le taux de compatibilité correspondant. Nous rappelons que nous avons utilisé une base de données constituées de 60 parrains et 60 filleuls. Nous représentons également dans les figures 14 et 15 respectivement les parrains n'ayant pas obtenu de filleul et les filleuls n'ayant pas obtenu de parrain.

Remarque 13. *En réalité, nous pourrons bien envisager une base de données contenant plus de parrains (étudiants français) que de filleuls (étudiants internationaux), cette hypothèse étant assez logique. Cela n'affecte en rien la démarche théorique et algorithmique et les résultats démontrés plus hauts resteront valides. En effet, seul le nombre de ligne des matrices* $\boldsymbol{P}$ *et* $\boldsymbol{C}$ *augmentera. Il est à noter que la priorité est pour le filleul de trouver un parrain qui partage les mêmes centres d'intérêts et qui l'aidera dans son intégration.*

	id_filleul	id_parrain	matching_rate
0	INT-00005	FR-00001	0.9682539682539683
1	INT-00013	FR-00002	0.8888888888888888
2	INT-00018	FR-00003	0.873015873015873
3	INT-00006	FR-00004	0.9841269841269842
4	INT-00053	FR-00006	0.9047619047619048
5	INT-00040	FR-00007	0.8888888888888888
6	INT-00015	FR-00008	0.9365079365079365
7	INT-00022	FR-00009	0.873015873015873
8	INT-00060	FR-00010	0.9206349206349207
9	INT-00011	FR-00011	1.0
10	INT-00044	FR-00012	0.9841269841269842
11	INT-00016	FR-00013	0.873015873015873
12	INT-00008	FR-00014	0.873015873015873
13	INT-00033	FR-00015	0.8412698412698413
14	INT-00010	FR-00017	0.8888888888888888
15	INT-00034	FR-00018	0.8888888888888888
16	INT-00007	FR-00019	0.9047619047619048
17	INT-00009	FR-00020	0.9047619047619048
18	INT-00001	FR-00021	0.9682539682539683
19	INT-00032	FR-00022	0.8888888888888888
20	INT-00041	FR-00023	0.8888888888888888
21	INT-00012	FR-00024	0.8571428571428572
22	INT-00028	FR-00025	0.873015873015873
23	INT-00048	FR-00026	1.0
24	INT-00024	FR-00028	0.8888888888888888
25	INT-00029	FR-00029	0.8253968253968254
26	INT-00030	FR-00030	0.873015873015873
27	INT-00026	FR-00031	0.873015873015873
28	INT-00017	FR-00032	0.8888888888888888
29	INT-00056	FR-00033	0.873015873015873
30	INT-00038	FR-00034	0.873015873015873
31	INT-00036	FR-00035	0.8571428571428572
32	INT-00019	FR-00036	0.8888888888888888
33	INT-00047	FR-00040	0.9841269841269842
34	INT-00002	FR-00041	1.0
35	INT-00031	FR-00042	1.0
36	INT-00020	FR-00043	0.8888888888888888
37	INT-00042	FR-00045	1.0
38	INT-00059	FR-00046	0.8571428571428572
39	INT-00046	FR-00047	0.8571428571428572
40	INT-00003	FR-00049	0.9841269841269842
41	INT-00014	FR-00050	0.8888888888888888
42	INT-00057	FR-00051	0.8888888888888888
43	INT-00037	FR-00053	0.873015873015873
44	INT-00023	FR-00055	0.9523809523809523
45	INT-00049	FR-00057	0.8888888888888888
46	INT-00004	FR-00058	0.8571428571428572
47	INT-00052	FR-00060	0.873015873015873

FIGURE 13 – Binômes et taux de compatibilité

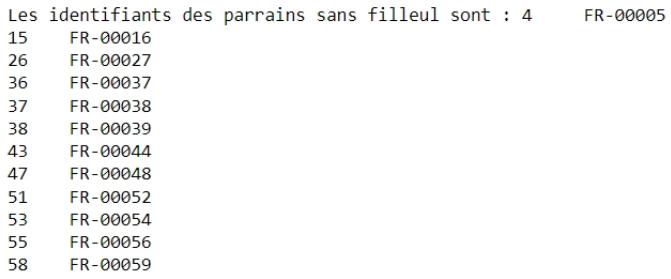

FIGURE 14 – identifiants des parrains sans filleul

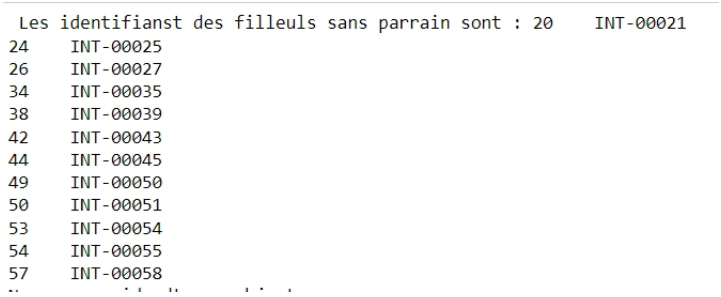

FIGURE 15 – identifiants des filleuls sans parrain

Commentaire :

L'analyse des résultats obtenus montre que nous avons pu former 48 binomes sur les 60 avec un taux de compatibilité minimal supérieur à 0.82. Nous avons donc obtenu un pourcentage d'automatisation intéressant égal à 80 pourcent. Les résultats obtenus ne sont pas incohérents. En effet, plusieurs facteurs pourraient justifier une automatisation partielle. D'une part, nous avons le nombre de proches voisins que nous avons fixé à 3 tout au long de l'étude et dont il découle que des parrains peuvent ne pas être choisi en tant que proches voisins. En pratique, cela correspond à une ligne de la matrice $\mathbf{P}$ identiquement nulle. D'autre part, l'architecture de la table représentée dans la figure 5 (doublons, faible combinatoire, forte corrélation entre les proches voisins) joue un rôle important dans l'efficacité de l'automatisation. Nous verrons dans la section suivante quelques techniques pour améliorer le processus voir converger vers une automatisation totale.

5.3 Vers une automatisation totale du processus d'appariement

L'étude des résultats numériques que nous avons obtenu plus haut met la lumière sur quelques problématiques notamment le pourcentage d'automatisation du processus d'attribution. Pour se faire nous verrons deux techniques : La première méthode consiste à effectuer un bouclage. Ceci dit, nous commençons par effectuer une première attribution puis nous définissons une base de données avec le reste des filleuls et des parrains et nous effectuons une deuxième

attribution. Nous répétons ce processus jusqu'à ce qu'il n'y ait plus de filleuls en réserve. La deuxième approche consiste à augmenter la combinatoire du problème en permettant beaucoup plus de proches voisins en espérant favoriser la decorrélation entre les proches voisins de nos filleuls.

5.3.1 Attribution et Bouclage

Dans cette partie, nous étudions l'effet du bouclage. Pour se faire, nous utilisons toujours un nombre de proches voisins égal à 3. Ensuite, nous effectuons des attributions en boucle en formant à chaque fois une nouvelle base de données avec les filleuls et parrains restants. La première attribution a donne les résultats présentés dans la figure 13. Nous montrons dans les figures 16 et 17 respectivement les résultats de la deuxième et troisième attribution.

	id_filleul	id_parrain	matching_rate
0	INT-00051	FR-00016	0.7619047619047619
1	INT-00027	FR-00027	0.7619047619047619
2	INT-00055	FR-00037	0.7777777777777778
3	INT-00045	FR-00038	0.7777777777777778
4	INT-00021	FR-00039	0.746031746031746
5	INT-00054	FR-00044	0.7777777777777778
6	INT-00035	FR-00048	0.7777777777777778
7	INT-00025	FR-00052	0.8412698412698413
8	INT-00050	FR-00054	0.7619047619047619
9	INT-00043	FR-00059	0.8095238095238095

FIGURE 16 – Binômes et taux de compatibilité après premier bouclage

	id_filleul	id_parrain	matching_rate
0	INT-00058	FR-00005	0.25396825396825395
1	INT-00039	FR-00056	0.31746031746031744

FIGURE 17 – Binômes et taux de compatibilité après deuxième bouclage

Commentaire :

Cette approche permet d'automatiser totalement la création de binômes. cependant, en examinant la figure 17 nous pouvons remarquer que la compatibilité baisse drastiquement (0.25 et 0.31). En effet, la reconstruction des bases de données à partir des filleuls et parrains restants entraîne une diminution de l'éventail des choix possibles à effectuer et donc à priori une détérioration de la compatibilité. Pour répondre à cette problématique en gardant une très bonne compatibilité entre filleuls et parrains tout en augmentant le taux d'automatisation, nous étudierons dans la partie suivante l'effet d'une augmentation de la combinatoire.

5.3.2 Augmentation de la combinatoire

Dans cette partie, nous étudions l'effet d'une augmentation de la combinatoire i.e une augmentation du nombre de proches de voisins sur le taux d'automatisation. Pour se faire, nous commençons par définir une fonction qui génère des coefficients de pondération selon le nombre de proches voisins que nous aurions choisi au préalable (voir figure 18). Nous mettons dans le tableau 3

les performances obtenues pour différentes valeurs du nombre de plus proches voisins.

Remarque 14. *Nous remarquerons plus particulièrement qu'une modification des pondérations ou du nombre de proches voisins est conforme avec l'approche théorique pour la démonstration de l'existence d'une solution problème d'optimisation sous contrainte. En effet, il suffira de changer les paramètres et la taille des matrices* $\boldsymbol{P}$ *et* $\boldsymbol{C}$ *et le raisonnement tient toujours.*

```
## cette fonction va generer une liste de ponderation coherente
def genere_liste_ponder(k):
    return [1-(i/k) for i in range(k)]
```

FIGURE 18 – Fonction génératrice de coefficients de pondération

Nbr voisins	Nbr binômes	automatisation	matching rate minimal	Nbr filleuls sans parrains
3	48	0.8	0.82	12
5	51	0.85	0.77	9
7	54	0.9	0.77	6
9	57	0.95	0.76	3

TABLE 3 – Performances obtenues en augmentant la combinatoire

Commentaire :

L'analyse des résultats obtenus montre que le pourcentage d'automatisation est une fonction croissante du nombre de proches voisins. En effet, une augmentation de la combinatoire favorise une decorrélation entre les parrains des différents filleuls. Cependant, une telle approche entraîne une détérioration du taux de compatibilité minimal et du temps de calcul, ce dernier étant logiquement lié à la taille des données. En examinant les deux techniques d'amélioration, nous pouvons conclure que la deuxième est plus avantageuse en terme de maximisation de la compatibilité quoique plus coûteuse en temps de calcul.

6 Conclusions

Jusqu'à présent, nous avons pu modéliser le problème de parrainage en utilisant des techniques d'optimisations sous contraintes. Dans un premier temps, nous avons étudier la nature des inputs de notre base de données afin de définir une métrique pertinente permettant de quantifier le degré de similarité entre parrains et filleuls. Dans un deuxième temps, nous avons déterminé les proches voisins de chaque filleul vis a vis de notre métrique avant de modéliser le besoin sous forme d'un problème d'optimisation sous contraintes.Dans un troisième temps, nous avons établi l'existence d'une solution pour le problème susmentionné et avons présenté la démarche algorithmique à suivre afin de générer les binomes.

L'analyse des résultats numériques obtenus et plus particulièrement les métriques montre que les algorithmes construits sont fonctionnels dans le sens ou ils convergent bien vers les solutions voulus. De plus, nous avons pu atteindre un taux d'automatisation de 0.8 avec un nombre de proches voisins égal à 3.

Pour améliorer les résultats obtenus, nous avons utilisé deux techniques : Une première qui se base sur le bouclage et qui permet d'assurer une automatisation totale du processus. Cependant, en procédant ainsi nous formons des binômes qui ne sont pas compatibles. Pour répondre à cette problématique en gardant une très bonne compatibilité entre filleul et parrain tout en augmentant le taux d'automatisation, nous avons décidé d'augmenter la combinatoire du problème en agissant sur le nombre de proches voisins de chaque filleuls. Cela nous a permis d'améliorer le taux d'automatisation tout en assurant une très bonne compatibilité entre parrains et filleuls.

7 Limites du modèle et perspectives

Le modèle ainsi construit fournit ,certes, de très bons résultats par rapport à une approche manuelle qui est sûrement plus coûteuse et moins précise en terme de maximisation de compatibilité. Cependant, nous n'arrivons toujours pas à atteindre une automatisation totale avec en choisissant un nombre raisonnable de proches voisin. Cela est peut être dû à la nature des données caractérisant parrains et filleuls d'une part et au nombre de parrains d'autre part , ce dernier étant égal au nombre de filleuls alors qu'en réalité nous avons plus de français que d'internationaux. De plus, nous pouvons très bien nous confronter à des situations où un filleul est très peu compatible avec tous les parrains.

Pour résoudre ce problème, il est possible de considérer une hiérarchie sur les filleuls se basant sur leurs taux de compatibilité avec les parrains. Par exemple, un filleul très peu compatible avec tous les parrains devra être prioritaire. En pratique cela revient a garder la hiérarchie sur les pondérations des plus proches voisins tout en ajoutant une hiérarchie sur les filleuls. De plus, une telle approche nous permettra en théorie d'obtenir une solution unique du problème d'optimisation sous contraintes ce qui le rendra très bien défini au sens de Hadamard.

8 Bibliographie

Références

[1] Heinz H Bauschke, Patrick L Combettes, Heinz H Bauschke, and Patrick L Combettes. *Correction to : Convex analysis and monotone operator theory in hilbert spaces*. Springer, 2017.

[2] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1) :9–15, 1962.

[3] Daniel Jurafsky and James H Martin. Speech and language processing : An introduction to natural language processing, computational linguistics, and speech recognition.

[4] Mykel J Kochenderfer and Tim A Wheeler. *Algorithms for optimization*. Mit Press, 2019.

[5] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2) :83–97, 1955.