



**AVIGNON**  
UNIVERSITÉ

# Rapport Application BI

Groupe 9

**Florian LAMBERT**  
**Maxime JULLIEN**

**8 décembre 2024**

**Master 2 Informatique**  
**ILSEN / IA**

**UE BUSINESS INTELLIGENCE**  
**ECUE AMS Application BI**

**Responsables**  
Remy KESSLER

**UFR**  
**SCIENCES**  
**TECHNOLOGIES**  
**SANTÉ**



**CENTRE**  
**D'ENSEIGNEMENT**  
**ET DE RECHERCHE**  
**EN INFORMATIQUE**  
[ceri.univ-avignon.fr](http://ceri.univ-avignon.fr)

## Sommaire

<b>Titre</b>	<b>1</b>
<b>Sommaire</b>	<b>2</b>
<b>1 Présentation</b>	<b>4</b>
1.1 Contexte	4
1.2 Organisation	4
<b>2 Méthodes</b>	<b>4</b>
2.1 Outils d'aide à la décision	5
2.1.1 Prométhée I	5
2.1.2 Prométhée II	6
2.1.3 Prométhée avec seuil de préférences	6
2.1.4 Electre Iv	7
2.1.5 Électre Is	8
2.1.6 Somme pondérée	9
2.1.7 Borda	9
2.2 Implémentation	10
2.2.1 Programme principal	10
2.2.2 Partie parseur du programme principal	10
2.2.3 Partie optimisation du programme principal	11
<b>3 Données</b>	<b>12</b>
3.1 Traitement des Déchets	12
3.1.1 Caractéristiques des Données	12
3.1.2 Nettoyage des Données	12
3.1.3 Analyse Descriptive	12
3.2 Modèle de Voiture	13
3.2.1 Caractéristiques des Données	13
3.2.2 Nettoyage des Données	13
3.2.3 Analyse Descriptive	14
<b>4 Résultats et interprétations</b>	<b>14</b>
4.1 Sélection de la meilleur déchetterie	14
Résultat de Prométhée	14
Résultat d'Electre	16
Résultat de Somme pondérée	16
Résultat de Borda	17
Comparaison des résultats	17
4.1.1 Sélection de la meilleure déchetterie par le thème "Environnement naturel"	17
4.1.2 Sélection de la meilleure déchetterie par le thème "Acceptabilité technique"	18
4.1.3 Sélection de la meilleure déchetterie par le thème "Environnement socio-économique"	18
4.1.4 Sélection de la meilleure déchetterie par le thème "Économie"	19
4.1.5 Interprétation des différents résultat	19
4.2 Sélection de la meilleure voiture	20
Résultat de Prométhée	20
Résultat d'Electre	22
Résultat de Somme pondérée	23

Résultat de Borda . . . . .	24
Comparaison des résultats . . . . .	24
4.2.1 Sélection de la meilleur voiture pour les Étudiants . . . . .	24
4.2.2 Sélection de la meilleur voiture pour les personnes âgée . . . . .	25
4.2.3 Interprétation des différents résultat . . . . .	26
<b>5 Conclusion . . . . .</b>	<b>26</b>
5.1 Général . . . . .	26
5.2 Florian . . . . .	27
5.3 Maxime . . . . .	27

# 1 Présentation

## 1.1 Contexte

Ce projet a pour objectif d'appliquer, dans un cadre pratique, des méthodes étudiées en cours et en travaux dirigés, en combinant programmation et analyse des résultats.

Ce projet permet la collaboration entre les filières de Master, les ILSN et les IA. Les étudiants en IA ont utilisé les méthodes pour la prise de décision multicritère, tandis que les étudiants en ILSN se sont concentrés sur la partie fouille de données et l'explication des informations trouvées.

À la suite de nos discussions, nous avons donc choisi deux sujets de type "aide à la décision" afin de travailler sur la sélection d'une solution selon un ensemble de critères.

Pour la réalisation de ce projet, nous avons inclus différentes méthodes comme la somme pondérée, Prométhée I et II, Électre IV et IS. Pour cela, nous allons réaliser nous-mêmes l'implémentation de ces méthodes dans le langage de programmation Python.

Les données que nous allons traiter concernent le jeu des déchets et des voitures.

## 1.2 Organisation

Le groupe est composé de Florian, spécialisé en ILSN, et Maxime, spécialisé en IA. Étant donné que le programme ILSN ne comprenait pas de cours sur l'aide à la décision, Maxime possédait une meilleure maîtrise des concepts liés au projet.

Le travail s'est déroulé principalement pendant les heures de cours, de manière concentrée et structurée. Dans un premier temps, l'effort a été orienté vers le développement des méthodes et algorithmes en Python. Florian s'est chargé de créer un parseur de fichiers CSV, utilisant la bibliothèque Pandas pour lire les données et extraire les poids réels. Pendant ce temps, Maxime a développé les méthodes Prométhée et Sommes pondérées pour générer les premiers résultats.

Une fois le code fusionné sur la plateforme Git ([lien](#)), les tâches ont été redistribuées. Florian a travaillé sur la méthode Électre, tandis que Maxime s'est concentré sur la méthode de Borda. Cette organisation a permis une répartition équilibrée des tâches, tirant parti des compétences spécifiques de chaque membre pour mener à bien le projet.

L'ajout de nouvelles fonctionnalités se faisait par création de branches dans Git. Nous avons essayé d'ajouter dans chaque commit de branche des nouvelles fonctionnalités afin d'éviter de réaliser un énorme commit contenant toutes les fonctionnalités en bloc. Cependant, durant le développement, cette règle n'a pas toujours été respectée, tout comme la prévention des bugs par l'ajout d'exceptions sur les inputs des algorithmes.

# 2 Méthodes

Dans cette partie, nous allons expliquer les différents types d'algorithmes que nous avons utilisés, ainsi que l'implémentation que nous avons réalisée et le recodage effectué sur un jeu de données.

## 2.1 Outils d'aide à la décision

### 2.1.1 Prométhée I

La méthode de Prométhée consiste à savoir le ou les candidats qui surclassent les autres candidats. Cette méthode sert à comparer entre deux candidats leurs différents critères. Pour cela, nous allons réaliser une matrice donnant une somme de poids de critères qui démontre qui a les meilleures valeurs de critères.

Pour expliquer cette méthodologie nous allons réaliser un exemple. On va prendre juste un cas avec deux candidats qui seront des joueurs, avec deux critères : Capacité physique avec un poids de 0.6 et Capacité mentale avec un poids de 0.4.

Exemple :

	Capacité Physique (0.6)	Capacité Mentale (0.4)
Joueur 1	19	6
Joueur 2	12	13

**Table 1.** Exemple de valeurs à traiter

On a donc les informations des deux joueurs. Ensuite, on va comparer par catégorie le joueur 1 et le joueur 2. Dans notre cas, les critères "Capacités Physique" et "Capacité Mentale" doivent être maximisés, cela veut dire qu'un candidat est meilleur qu'un autre si celui-ci a une valeur plus grande que l'autre candidat. Et inversement si on devait minimiser le critère. Revenons à notre exemple, pour la capacité physique, le joueur 1 est plus fort que le joueur 2 ( $19 > 12$ ) donc il remporte le poids associé, soit 0.6. Pour la capacité mentale, c'est le joueur 2 qui est plus fort que le joueur 1, donc il remporte le poids 0.4.

On arrive donc au tableau suivant :

	Joueur 1	Joueur 2
Joueur 1	-	0.6
Joueur 2	0.4	-

**Table 2.** Tableau de concordance

Dans ce tableau, on a pour chaque joueur, si on le compare à un autre joueur, la somme des poids où il est meilleur. On obtient ainsi le tableau de concordance entre les joueurs. Ensuite, on rajoute une ligne et une colonne correspondant à  $\Phi^-$  et  $\Phi^+$ . Dans  $\Phi^-$ , on fera la somme des lignes pour une colonne, et dans  $\Phi^+$ , la somme des colonnes pour une ligne.

On arrive donc au tableau suivant :

	Joueur 1	Joueur 2	$\Phi^+$
Joueur 1	-	0.6	0.6
Joueur 2	0.4	-	0.4
$\Phi^-$	0.4	0.6	

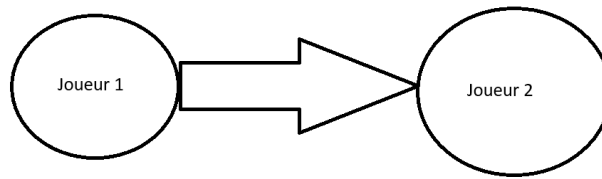
**Table 3.** Tableau de Prométhée I

Ainsi, on obtient pour  $\Phi^+ = [0.6, 0.4]$  et  $\Phi^- = [0.4, 0.6]$ .

Il nous reste à définir l'ordre selon  $\Phi^+$  et  $\Phi^-$ . Pour  $\Phi^+$ , le meilleur est celui qui a la plus grande valeur. Pour  $\Phi^-$ , c'est celui qui a la valeur la plus petite qui est le meilleur.

On obtient donc :

- $\Phi^+$  : Joueur 1 > Joueur 2 (car  $0.6 > 0.4$ )
  - $\Phi^-$  : Joueur 1 > Joueur 2 (car  $0.4 < 0.6$ )
- Le graphique obtenu sera donc :



**Figure 1.** Exemple graphique Prométhée I

Dans notre cas, les meilleurs joueurs seront ceux qui ne sont pas surclassés par d'autres. Ici, il n'y a que le joueur 1. Ainsi, d'après Prométhée I, le meilleur joueur est le joueur 1.

### 2.1.2 Prométhée II

Pour la méthodologie de Prométhée II, cela va commencer pareil que Prométhée I. On revient donc à notre dernier résultat (voir 3).

Pour continuer, on va juste faire la différence de  $\Phi^+$  à  $\Phi^-$ , ce qui nous obtiendra  $\Phi$ .

On obtiendra donc  $\Phi = \Phi^+ - \Phi^- = [0.6 - 0.4, 0.4 - 0.6] = [0.2, -0.2]$ .

Dans le cas de  $\Phi$ , on veut qu'il soit trié par ordre décroissant. On obtient donc Joueur 1 > Joueur 2 (car  $0.2 > -0.2$ ).

On obtient donc le graphique obtenu dans Prométhée I (voir 3) et le même résultat, qui est que la meilleure option est le Joueur 1.

### 2.1.3 Prométhée avec seuil de préférences

Prométhée avec seuil de préférence, ce sont juste des versions de Prométhée I et II qui vont utiliser un seuil lors de la comparaison.

Reprenons notre exemple avec Prométhée I.

L'importance du seuil est lorsqu'on réalise la matrice de concordance. Donc, lorsqu'on va comparer le Joueur 1 et le Joueur 2 pour une catégorie spécifique. Reprenons le cas avec la catégorie Capacité physique. On a bien Joueur 1 qui est meilleur que le Joueur 2 ( $19 > 12$ ). Cependant, au lieu de donner au Joueur 1 forcément le poids de la catégorie, on va d'abord vérifier si la différence entre Joueur 1 et Joueur 2 est supérieure à un seuil. Si c'est supérieur, alors on donne la totalité du poids au Joueur 1, sinon on donne (la différence divisée par le seuil) multipliée par le poids.

Donc donnons comme seuil 10 pour chaque catégorie.

- Pour la capacité physique, on a Joueur 1 qui est meilleur que Joueur 2 ( $19 > 12$ ). Le Joueur 1 obtient la valeur suivante :  $((19 - 12)/10) \times 0.6 = (7/10) \times 0.6 = 0.42$ .
- Pour la capacité mentale, on a Joueur 2 qui est meilleur que Joueur 1 ( $13 > 6$ ). Le Joueur 2 obtient la valeur suivante :  $((13 - 6)/10) \times 0.4 = (7/10) \times 0.4 = 0.28$ .

On obtient donc le nouveau tableau de concordance suivant :

	Joueur 1	Joueur 2
Joueur 1	-	0.42
Joueur 2	0.28	-

**Table 4.** Tableau de concordance

Nous aurons aussi des valeurs de  $\Phi^+$  et  $\Phi^-$  différentes. Cependant, on peut avoir des résultats différents ou non.

#### 2.1.4 Electre Iv

La méthode d'Électre IV consiste à récupérer des candidats qui ne sont surclassés par aucun autre candidat. Il se peut qu'avec cette méthode, nous arrivions à une boucle, par exemple :  $C1 > C2$ ,  $C2 > C3$ , mais  $C3 > C1$ . Dans ces cas, nous supprimerons les liens ayant les scores de concordance les plus faibles (décrit plus tard). Durant cette méthode, nous allons devoir réaliser deux tableaux : un tableau de concordance et un tableau de discordance. Commençons par le tableau de concordance.

Le tableau de concordance correspond, pour chaque candidat, au score de concordance qu'il reçoit par rapport à un autre candidat. Pour cela, nous allons prendre un exemple de Prométhée I en modifiant une valeur.

	Capacité Physique (0.6)	Capacité Mentale (0.4)
Joueur 1	19	13
Joueur 2	12	13

**Table 5.** Exemple de valeur à traiter

Pour rappeler, tous les critères doivent être maximisés. On aura donc pour le critère "Capacité Physique", le joueur 1 qui est supérieur ou égal au joueur 2 ( $19 \geq 12$ ), donc il gagne 0.6. Pour le critère "Capacité Mentale", les deux joueurs vont avoir les points, car ils ont les mêmes valeurs ( $13 \geq 13$ ). C'est l'une des différences entre Électre et Prométhée. On obtient donc le tableau de concordance suivant :

	Joueur 1	Joueur 2
Joueur 1	-	1
Joueur 2	0.4	-

**Table 6.** Tableau de concordance

Nous avons le tableau de concordance, maintenant il faut construire le tableau de discordance. Ce tableau sera réalisé lorsqu'entre deux candidats, celui qui est le moins bon a une différence avec l'autre candidat supérieure ou égale à la valeur de veto pour ce critère. Si on prend comme veto =  $[6, 5]$ , où 6 est le veto du critère "Capacité Physique" et 5 celui de "Capacité Mentale".

Prenons le critère "Capacité Physique". Le joueur 2 est moins bon que le joueur 1. Cependant, la différence entre les deux est supérieure à 6 ( $19 - 12 = 7 \geq 6$ ). On mettra donc un 0 dans la case où l'on compare le joueur 2 au joueur 1. Dans les autres cas, si on ne dépasse pas le veto, alors on met un 1. On obtient le tableau suivant :

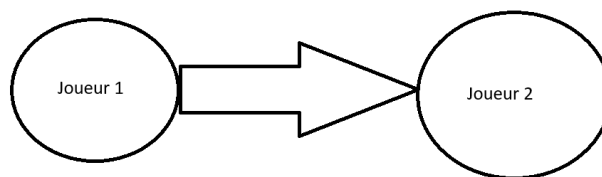
	Joueur 1	Joueur 2
Joueur 1	-	1
Joueur 2	0	-

**Table 7.** Tableau de non-discordance

Après avoir réalisé ces deux tableaux, nous allons avoir besoin d'un seuil de concordance. Ce seuil permet de déterminer quels liens nous allons conserver dans le tableau de concordance. Dans notre cas, si nous fixons le seuil à 0, nous conservons tous les liens qui respectent cette condition. Tous les liens le respectent.

Ensuite, nous regardons le tableau de non-discordance pour vérifier si les liens que nous avons conservés ont une valeur de 1 dans ce tableau. Si ce n'est pas le cas, nous supprimons le lien. Dans cet exemple, nous supprimons le lien Joueur1 → Joueur2.

On obtient finalement le graphe suivant :



**Figure 2.** Exemple graphique Électre IV

Dans ce graphique, on peut voir que seul le joueur 1 n'est pas surclassé par d'autres joueurs. Le noyau est donc : [Joueur1]. La solution est donc le joueur 1.

### 2.1.5 Électre Is

La méthode d'Électre Is correspond à la même méthodologie que celle d'Électre IV. Cependant, lorsqu'on calcule le tableau de concordance, on applique des seuils de préférence en fonction du type de critère. Contrairement à Prométhée avec seuils, où même si un candidat n'a pas de meilleur résultat qu'un autre, il peut obtenir une partie des poids en fonction de la différence entre les deux valeurs.

Reprenons notre exemple sur les données d'ElectreIV (voir 5).

Nous allons mettre un seuil pour chaque critère à 9. Si on regarde le critère "Capacité Mental", cela ne changera rien, les deux candidats auront le poids de ce critère. Cependant, si on regarde le critère "Capacité Physique" on a le joueur2 qui est inférieur au joueur1 ( $12 < 19$ ). Mais la différence entre les 2 valeurs est inférieure au seuil, donc le joueur2 va gagner une partie des poids. Le joueur2 gagnera donc  $((9 - (19 - 12)) / 9) * 0.6 = ((9 - 7) / 9) * 0.6 = 2 / 9 * 0.6 = 0.1333333$ .

On aura donc le tableau de concordance suivant :

	Joueur 1	Joueur 2
Joueur 1	-	1
Joueur 2	0.533333	-

**Table 8.** Tableau de concordance

Avec cette méthodologie, il est possible d'obtenir des résultats plus élevés que ceux de la méthode d'Electre IV pour le tableau de concordance. L'utilisation du seuil n'affecte pas le tableau de non-discordance. Et on se retrouve à récupérer les liens entre 2 candidats si



la valeur dans le tableau de concordance dépasse le seuil de concordance et qu'il y est 1 dans le tableau de non-discordance. Cette méthodologie permet de rajouter de nouveaux liens dans le graphique.

### 2.1.6 Somme pondérée

La méthodologie de somme pondérée permet de déterminer quel(s) candidat(s) sont les meilleurs par rapport aux autres. Pour cela, cette méthodologie consiste à faire la somme des valeurs des critères d'un candidat, multipliées par le poids des critères.

Pour réaliser cette méthodologie, nous devons d'abord normaliser les données. Cette étape est obligatoire, car l'ordre de grandeur des valeurs des critères peut varier considérablement, on peut passer d'un critère avec des valeurs entre 0 et 4 à un critère avec des valeurs entre 0 et 10000000000. Pour cela, nous utilisons la méthode des résidus studentisés (studentized residual method) pour normaliser les données des différents critères. Les données seront donc sous la forme suivante :

$$x_{ij} = \frac{x_{ij} - \bar{C}_j}{\sigma_{C_j}}.$$

À la suite de cette normalisation, nous allons calculer la somme des valeurs multipliées par les poids des critères pour chaque candidat. Nous obtenons ainsi, pour chaque candidat :

$$resultat_i = \sum_{j=1} x_{ij} \cdot w_j$$

Après avoir obtenu ces résultats, nous les classons par ordre croissant. Enfin, nous sélectionnons le ou les meilleurs résultats, c'est-à-dire celui ou ceux qui sont supérieur aux autres.

### 2.1.7 Borda

La méthodologie de Borda est principalement utilisée dans les théories du choix social. Cependant, avec une représentation dite de Borda, nous pouvons également effectuer de l'optimisation multicritère discrète. Cette méthode ne prend pas en compte les valeurs exactes des critères, mais se base uniquement sur le classement des candidats pour un critère donné.

Pour appliquer cette méthode, nous devons d'abord représenter, pour chaque critère, quels candidats sont les meilleurs ou les pires. Les valeurs seront attribuées de 1 (correspondant au plus mauvais) à  $n$  (le nombre de candidats), où  $n$  correspond au meilleur pour un critère donné, en fonction du contexte (maximiser ou minimiser).

Prenons un exemple avec le jeu de données suivant (où chaque critère doit être maximisé) :

	C1	C2	C3
A1	2	4	4
A2	1	1	6
A3	4	1	5
A4	3	7	3
A5	10	5	1

**Table 9.** Table des valeurs

Avec cette table des valeurs, on va donner pour chaque candidat la position s'il est bien ou pas dans le classement pour une catégorie donnée. Cela nous donne pour cet exemple :

	C1	C2	C3
A1	2	3	3
A2	1	1.5	4
A3	4	1.5	5
A4	3	5	2
A5	5	4	1

**Table 10.** Table des points de Borda

Pour expliquer le cas de  $C_2$  entre les candidats  $A_2$  et  $A_3$ , on peut constater que ces deux candidats ont la même valeur, qui est 1. Ainsi,  $A_2$  et  $A_3$  recevront les scores 1 et 2. Cependant, pour déterminer quel score attribuer à chacun, nous utilisons la règle suivante : nous prenons le numéro de la prochaine valeur à afficher (dans notre cas, 3), puis nous le divisons par le nombre de candidats ayant la même valeur (ici, 2). Cela donne  $3/2 = 1.5$ . Par conséquent, la valeur attribuée aux candidats  $A_2$  et  $A_3$  pour le critère  $C_2$  sera 1.5.

Après avoir défini les scores pour chaque critère, nous calculons ensuite la somme des scores pour chaque candidat. Nous obtenons ainsi :

	C1	C2	C3	B(Ai)
A1	2	3	3	8
A2	1	1.5	4	6.5
A3	4	1.5	5	10.5
A4	3	5	2	10
A5	5	4	1	10

**Table 11.** Table des points avec le décompte de Borda

On obtient donc le tableau suivant  $B(Ai) = [8, 6.5, 10.5, 10, 10]$ . Puis on remet par ordre croissant les résultats de  $B(Ai)$ . On obtient donc à la fin que  $A_3 > A_4 = A_5 > A_1 > A_2$ .

Implémentation :

## 2.2 Implémentation

### 2.2.1 Programme principal

Avant de pouvoir lancer le programme, il faut installer les librairies nécessaires à son installation. Il faudra, avant toute installation, exécuter la commande suivante : `python3 -m pip install -r requirements.txt` (les versions Python 3.10, Python 3.12 peuvent être utilisées, mais il faut au minimum que ce soit Python 3 pour lancer le programme).

Le programme principal sera lancé par `python3 main.py`. Pour l'utilisation de ce programme, vous pourrez le faire en ligne de commande ou directement dans le programme.

Durant la première partie du programme, il faudra sélectionner le type d'action que l'on veut effectuer : soit on va vouloir de l'optimisation, soit on va parser les données afin de pouvoir les utiliser dans le programme.

### 2.2.2 Partie parseur du programme principal

Afin de pouvoir utiliser les algorithmes d'optimisation, nous avons utilisé une mise en forme spécifique des données en CSV. Il faudra obligatoirement au moins deux noms de

colonnes. On aura des colonnes C\*, par exemple : C1 ou C10, qui représentent un candidat par colonne. Les lignes correspondent donc aux critères des différents candidats. Et enfin, on aura la colonne TrueWeight, qui représentera les poids que l'on va utiliser pour les différents algorithmes. La somme de ces poids doit valoir 1.

La partie parseur du programme permet donc de préparer les données pour l'optimisation. Pour lancer cette partie, il faudra exécuter `python3 main.py parseur`.

Durant cette utilisation, plusieurs éléments vous seront proposés, et ceux-ci pourront également être effectués en ligne de commande :

- `-file filepath, --filepath filepath` : Représente le fichier que l'on veut parser.
- `-transpose` : Réalisation d'une transposition des données.
- `-data_type` : Va mettre en place les noms des colonnes en forme de C\*. Cela prendra toutes les colonnes au début du programme pour changer les noms en forme C\*. On peut mettre plusieurs fois cet argument.
- `-categorie filename` : Représente les données d'une catégorie, au format colonne, que l'on veut rajouter.
- `-categorie_name name` : Représente le nom de la colonne que l'on veut rajouter. On peut mettre plusieurs fois cet argument.
- `-weight filename` : Permet d'utiliser les données d'un fichier, au format ligne, et de les mettre dans une colonne Weight.
- `-true_weight filename` : Permet d'utiliser les données d'un fichier, au format ligne, et de les mettre dans une colonne TrueWeight.
- `-compute_true_weight` : Permet de calculer une TrueWeight par rapport au Weight et à la catégorie Catégorie.
- `-compute_true_weight_filtered name [name...]` : Permet de calculer une TrueWeight par rapport au Weight et à la catégorie Catégorie que l'on aura donnée précisément.

Sinon sans ligne de commande, le programme vous demandera si vous voulez faire certains action. Ce programme permet aussi de lire les fichiers contenant les seuils ou le veto de changer la mise en forme colonne en mise en forme ligne qui sera requis pour la partie optimisation.

### 2.2.3 Partie optimisation du programme principal

Après avoir mis en forme les données que l'on doit traiter, nous allons lancer les algorithmes d'optimisation sur ces données. Nous pourrions choisir entre 4 types d'optimisation : Prométhée, Électre, Borda, Somme pondérée (weightsum). Cette partie du programme fonctionne avec ou sans ligne de commande. Au début, le programme prendra 3 arguments :

- `-f name, --function name` : Le type d'optimisation de fonction que nous voulons réaliser.
- `-file filepath, --filepath filepath` : Permet de savoir sur quelles données l'optimisation doit se réaliser.
- `-op filepath, --operation filepath` : Permet de savoir, pour chaque ligne, quel type d'opération on souhaite faire (si on veut maximiser le critère ou le minimiser).

Mais nous aurons :

- Pour Prométhée :
  - -version\_algorithme name : Pour savoir quelle version de Prométhée nous avons choisie (1 pour ProméthéeI et 2 pour ProméthéeII).
  - -seuil\_filepath : Permet, si on le souhaite, de donner les seuils pour les critères, au format ligne, afin de réaliser une version de Prométhée avec seuil.
- Pour Électre :
  - -veto\_filepath : Permet de récupérer les valeurs des veto, au format ligne; par critères nécessaires à l'algorithme.
  - -concordance\_float : Définit le seuil de concordance que l'on souhaite utiliser (doit être compris entre 0 et 1).
  - -seuil\_filepath : Permet, si on le souhaite, de donner les seuils pour les critères afin de réaliser ÉlectreIS, au format ligne.
- Pour Résultat de Somme pondérée
  - -type\_result\_name : Permet de savoir si la sortie doit être triée par ordre croissant (min) ou décroissant (max).
- Pour le cas de Borda, il n'y a aucun argument à donner en plus.

## 3 Données

### 3.1 Traitement des Déchets

#### 3.1.1 Caractéristiques des Données

Les données relatives au traitement des déchets concernent six sites potentiels envisagés pour l'installation d'un complexe de gestion des déchets.

Le dossier de ce sujet contenait : un fichier csv avec les données des attributs (en colonnes) des différents sites sans aucune entête, un autre fichier csv contenant la liste des poids de chaque critère (en ligne), et enfin, un fichier pdf contenant les caractéristiques de la table. Le pdf est le plus intéressant car il décrit et détaille les catégories et thèmes des critères.

#### 3.1.2 Nettoyage des Données

Dans ces données il n'y avait pas d'erreur manquante ou de grande incohérences, cependant, les données étaient techniquement dans trois fichiers différents et nous avons décidé de fusionner les données ensembles. Prenant comme base le fichier contenant les valeurs numériques des attributs, nous avons rajouter plusieurs colonnes afin que le fichier csv ressemble à celui du pdf.

Les colonnes catégories, thèmes, critères et weight spécifier dans la table 1 du pdf furent rajouter ainsi que des entêtes afin de pouvoir mieux visualisé directement les données.

Finalement, avec l'ajout des catégories, nous devions aussi ajouter deux colonnes pour pouvoir jouée avec les poids et catégories, ces colonnes devinrent ThemeValue et TrueWeight.

#### 3.1.3 Analyse Descriptive

Il existe quatre principales catégories : Acceptabilité technique, Environnement naturel, Environnement socio-économique et Économie. Chaque catégorie est liée à plusieurs thèmes et critères.

La catégorie *acceptabilité technique* se concentre sur les caractéristiques du terrain et l'accès au site. Elle soulève des questions comme : « Est-il facile de se rendre sur le site ? Le terrain est-il adapté et sans danger ? ». Les thèmes abordés incluent la géologie

des sols, l'hydrologie des aquifères, les eaux superficielles et la morphologie du terrain, qui se rapportent aux caractéristiques naturelles et aux contraintes techniques du site. Cette catégorie prend également en compte des aspects pratiques comme l'accès, la distance et le trafic, ainsi que les coûts fonciers, qui évaluent le prix des terrains et le nombre de propriétaires concernés. Bien que ces derniers puissent être rattachés à la catégorie Économie, ils restent ici pour leur pertinence dans l'analyse technique de la faisabilité du projet.

La catégorie *environnement naturel* s'intéresse aux impacts sur le cadre paysager et l'écosystème local. Elle répond à des préoccupations comme : « La création de cette déchetterie ruinera-t-elle la vue dans ce village alpin ? ». Elle évalue également l'état et la qualité de l'environnement naturel en se demandant si le site pourrait affecter la faune et la flore locales. Cette catégorie est essentielle pour préserver l'environnement et éviter des conflits avec des tiers, tels que des organisations ou des autorités environnementales.

La catégorie *environnement socio-économique* analyse les impacts sur les populations locales et les activités économiques existantes. Elle examine les densités d'habitat pour privilégier les zones moins peuplées, les implications pour le tourisme et l'urbanisme afin de protéger le patrimoine et les loisirs, ainsi que les effets potentiels sur des secteurs comme l'agriculture ou l'artisanat. Cette catégorie vise à anticiper les risques pour les acteurs locaux et à évaluer les conséquences de l'implantation du projet sur leur quotidien.

Enfin, la catégorie *économie* se concentre sur les coûts associés au projet, tels que les dépenses de construction, les frais de transport et la valeur foncière des terrains. Cette dernière notion, bien qu'évoquée dans l'acceptabilité technique, reste pertinente ici pour évaluer globalement les sites les plus économiquement viables. L'objectif de cette catégorie est de minimiser les coûts tout en optimisant les ressources disponibles pour le projet.

## 3.2 Modèle de Voiture

### 3.2.1 Caractéristiques des Données

Ce jeu de données porte sur le choix d'un modèle de voiture turbo diesel européenne parmi dix modèles différents. Le dossier du sujet comprenait un fichier CSV contenant les données brutes, sans en-têtes, pour les différents modèles, ainsi qu'un fichier PDF fournissant des explications détaillées sur ces données.

### 3.2.2 Nettoyage des Données

Contrairement au premier jeu de données disponible sur le site de traitement des déchets, où les critères étaient disposés en colonnes et les modèles de voitures en lignes, nous avons inversé cette disposition. Désormais, les critères sont présentés en lignes et les modèles de voitures en colonnes. Cette modification vise à faciliter la réutilisation de nos algorithmes et à rendre le tableau plus lisible et visuellement clair.

En utilisant les données et les critères décrits dans le PDF *donneesvoiture*, nous avons ajouté des en-têtes pour améliorer la lisibilité des informations. Les modèles de voitures ne sont pas nommés avec leur nom complet, mais sont identifiés par des abréviations simples telles que C1, C2, C3, etc.

De plus, une colonne nommée *TrueWeight* a été ajoutée. Cette colonne permet d'attribuer manuellement un poids aux différents critères (étant donné que ces poids n'étaient pas inclus dans les données initiales), afin de mieux identifier les critères les plus importants pour orienter la prise de décision.

### 3.2.3 Analyse Descriptive

Il y a moins de données ici par rapport au premier jeu de données, avec seulement sept critères et aucune catégorie définie. Si nous devons créer des catégories, il y en aurait trois principales. La première serait centrée sur les performances du véhicule, regroupant des critères tels que la vitesse maximale atteignable, la distance de freinage et l'accélération.

La deuxième catégorie concernerait les aspects utilitaires du véhicule, comme le volume du coffre et le confort.

Enfin, la dernière catégorie relèverait des aspects économiques, incluant le prix du véhicule et sa consommation moyenne.

Ces catégories permettraient de mieux identifier le type de véhicule qu'un utilisateur pourrait rechercher en ajustant les poids attribués à chaque critère. Par exemple, préférez-vous une voiture peu coûteuse, un véhicule performant ou un modèle utilitaire ?

## 4 Résultats et interprétations

### 4.1 Sélection de la meilleur déchetterie

Pour sélectionner le meilleur candidat, tous les critères doivent avoir le score le plus bas possible (pour notre jeu de données).

Pour les seuils de préférence que nous utiliserons pour Electre et Prométhée, nous fixerons toutes les valeurs à 2.

#### Résultat de Prométhée

- Pour Prométhée<sup>+</sup> :  $\Phi^+ = C4 > C2 > C3 > C1 > C5 > C6$   $\Phi^- = C3 > C4 > C2 > C6 > C1 > C5$ .
- Pour Prométhée<sup>ell</sup> :  $\Phi = C3 > C4 > C2 > C1 > C6 > C5$ .
- Pour Prométhée<sup>el</sup> avec seuil :  $\Phi^+ = C2 > C4 > C1 > C3 > C5 > C6$   $\Phi^- = C2 > C4 > C3 > C6 > C1 > C5$ .
- Pour Prométhée<sup>ell</sup> avec seuil :  $\Phi = C2 > C4 > C3 > C1 > C6 > C5$ .

Résultat de Prométhée avec image :

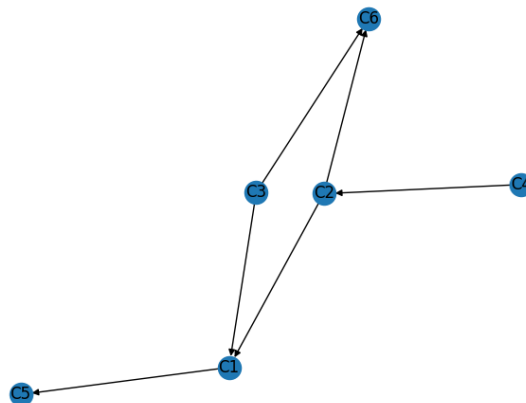
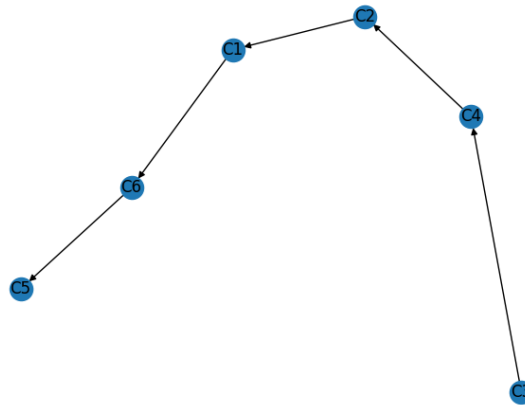
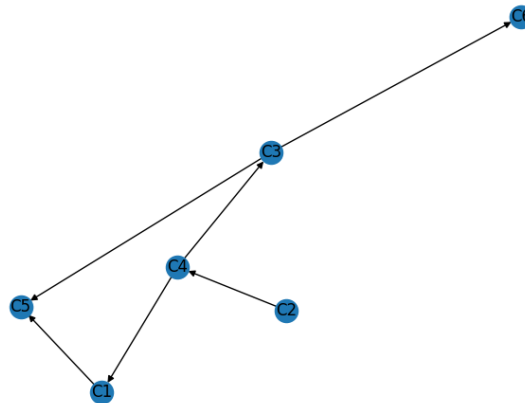


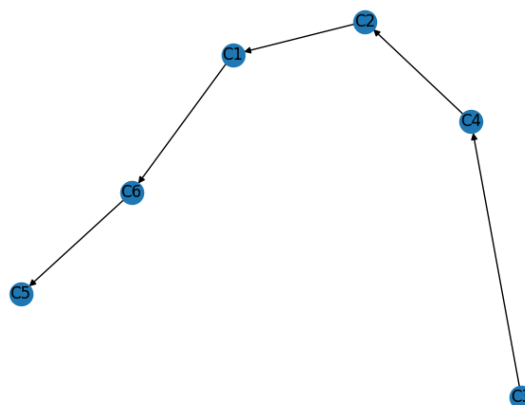
Figure 3. Résultat de Prométhée sur les données des déchetteries



**Figure 4.** Résultat de Prométhée I sur les données des déchetterie



**Figure 5.** Résultat de Prométhée II avec seuil sur les données des déchetterie



**Figure 6.** Résultat de Prométhée I avec seuil sur les données des déchetterie

Avec ces graphiques, nous pouvons conclure qu'avec la méthode de Prométhée I, C4 et C3 sont les candidats qui ne sont surclassés par aucun autre candidat. En utilisant la méthode de Prométhée II, c'est le candidat C4 qui surclasse les autres candidats. Cependant, en utilisant les seuils de préférence, les méthodes de Prométhée I et II indiquent que C2 est le meilleur candidat.

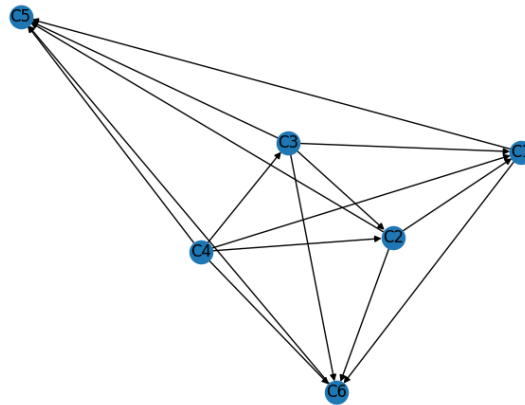
### Résultat d'Electre

Pour le cas d'Electre, le veto de chaque critère est fixé à une valeur supérieure à 4. Le veto est défini à cette valeur car il est précisé que les candidats ne doivent pas être affectés par ce veto. Le seuil de concordance est fixé à 0,7.

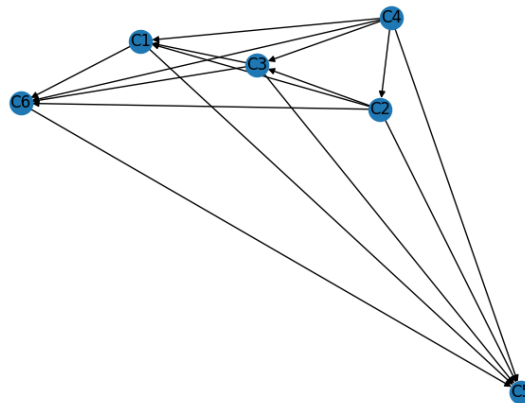
Voici les résultats obtenus à partir des différentes méthodologies Electre :

- Pour Electrelv, le noyau sont C2, C4.
- Pour Electrels, le noyau est C4.

Résultat d'Electre avec des graphiques :



**Figure 7.** Résultat de ElectreIV sur les données des déchetterie



**Figure 8.** Résultat de ElectreIS sur les données des déchetterie

D'après les résultats des méthodes Electre, nous pouvons observer qu'avec Electre IV, les candidats C2 et C4 ne sont surclassés par aucun autre candidat. En revanche, avec la méthode Electre IS, c'est uniquement le candidat C4 qui n'est surclassé par aucun autre candidat.

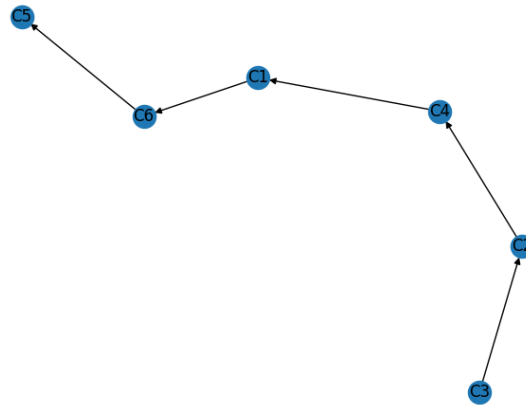
### Résultat de Somme pondérée

Le résultat de la méthodologie de somme pondérée est le suivant :

- $C3 > C2 > C4 > C1 > C6 > C5$

Voici les résultats sous forme de graphique :





**Figure 9.** Résultat de la somme pondérée sur les données des déchetterie

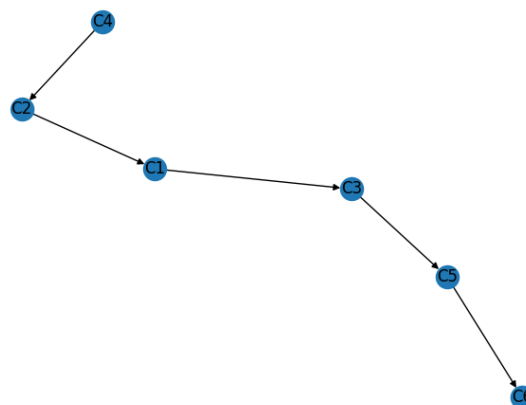
Le candidat C3 est le meilleur d'après la méthode de somme pondérée.

### Résultat de Borda

Le résultat de la méthodologie de Borda sur ces données est le suivant :

- $C4 > C2 > C1 > C3 > C5 > C6$

Voici les résultats sous forme de graphique :



**Figure 10.** Résultat de Borda sur les données des déchetterie

Le candidat C4 est le meilleur d'après la méthodologie de Borda.

### Comparaison des résultats

D'après les différents résultats des méthodologies, les meilleurs candidats sont C4, C3 et C2. En comptant le nombre de fois où chaque candidat a été choisi comme le meilleur selon les différentes méthodologies, nous pouvons conclure que C4 est le meilleur candidat.

Dans les prochaines sous-parties, nous examinerons le meilleur candidat en fonction des différents thèmes de critères pour la déchetterie. Nous ne réaliserons pas de méthode de Borda sur ces données, car le résultat serait identique à celui précédemment présenté.

#### 4.1.1 Sélection de la meilleure déchetterie par le thème "Environnement naturel"

Dans ce cas, nous attribuerons un poids de 0 aux critères qui ne font pas partie du thème "Environnement naturel". Pour les critères de ce thème, les valeurs des TrueWeight seront ajustées de manière à ce que la somme des poids soit égale à 1.

### Résultat de Prométhée

- Pour Prométhée 1 :  $\Phi^+ = C6 > C5 > C3 > C4 > C1 > C2$   $\Phi^- = C6 > C3 > C4 > C5 > C1 > C2$ .
- Pour Prométhée 2 :  $\Phi = C6 > C3 > C5 > C4 > C1 > C2$ .
- Pour Prométhée 1 avec seuil :  $\Phi^+ = C6 > C3 > C4 > C5 > C1 > C2$   $\Phi^- = C6 > C4 > C3 > C5 > C2 > C1$ .
- Pour Prométhée 2 avec seuil :  $\Phi = C6 > C3 > C4 > C5 > C1 > C2$ .

### Résultat d'Electre

- Pour Electrelv, le noyau est C6.
- Pour Electrels, le noyau est C6.

### Résultat de Somme pondérée

Le résultat de la méthodologie de somme pondérée est le suivant :

- $C6 > C3 > C4 > C5 > C1 > C2$

D'après les différents résultats des méthodologies, les meilleurs candidats sont C6 et C4. En comptabilisant le nombre de fois où chaque candidat a été choisi comme le meilleur, nous pouvons conclure que, pour le thème de l'"Environnement naturel", le candidat C6 est la meilleure solution.

#### 4.1.2 Sélection de la meilleure déchetterie par le thème "Acceptabilité technique"

Dans ce cas, nous attribuerons un poids de 0 aux critères qui ne font pas partie du thème "Acceptabilité technique". Pour les critères de ce thème, les valeurs des TrueWeight seront ajustées de manière à ce que la somme des poids soit égale à 1.

##### Prométhée :

- Pour Prométhée 1 :  $\Phi^+ = C4 > C2 > C3 > C1 > C5 > C6$   $\Phi^- = C4 > C3 > C2 > C1 > C5 > C6$ .
- Pour Prométhée 2 :  $\Phi = C4 > C3 > C2 > C1 > C5 > C6$ .
- Pour Prométhée 1 avec seuil :  $\Phi^+ = C4 > C2 > C3 > C1 > C5 > C6$   $\Phi^- = C4 > C3 > C2 > C1 > C5 > C6$ .
- Pour Prométhée 2 avec seuil :  $\Phi = C4 > C3 > C2 > C1 > C5 > C6$ .

### Résultat d'Electre

- Pour Electrelv, le noyau est C4.
- Pour Electrels, le noyau est C4.

### Résultat de Somme pondérée

Le résultat de la méthodologie de somme pondérée est le suivant :

- $C4 > C3 > C2 > C1 > C5 > C6$

D'après les différents résultats des méthodologies, nous pouvons conclure que, pour le thème de l'"Acceptabilité technique", le candidat C4 est la meilleure solution.

#### 4.1.3 Sélection de la meilleure déchetterie par le thème "Environnement socio-économique"

Dans ce cas, nous attribuerons un poids de 0 aux critères qui ne font pas partie du thème "Environnement socio-économique". Pour les critères de ce thème, les valeurs des TrueWeight seront ajustées de manière à ce que la somme des poids soit égale à 1.

##### Prométhée :

- Pour Prométhée 1 :  $\Phi^+ = C2 > C5 > C1 > C4 > C6 > C3$   $\Phi^- = C2 > C5 > C6 > C3 > C1 > C4$ .
- Pour Prométhée 2 :  $\Phi = C2 > C5 > C1 > C6 > C3 > C4$ .
- Pour Prométhée 1 avec seuil :  $\Phi^+ = C2 > C5 > C1 > C6 > C4 > C3$   $\Phi^- = C2 > C5 > C6 > C1 > C3 > C4$ .
- Pour Prométhée 2 avec seuil :  $\Phi = C2 > C5 > C1 > C6 > C3 > C4$ .

**Résultat d'Electre**

- Pour Electrelv, le noyau est C2.
- Pour Electrels, le noyau est C2.

**Résultat de Somme pondérée**

Le résultat de la méthodologie de somme pondérée est le suivant :

- $C2 > C5 > C3 > C1 > C6 > C4$

D'après les différents résultats des méthodologies, les meilleurs candidats sont C2 et C4. En comptabilisant le nombre de fois où chaque candidat a été choisi comme le meilleur, nous pouvons conclure que, pour le thème de l' "Environnement socio-économique", le candidat C2 représente la meilleure solution.

**4.1.4 Sélection de la meilleure déchetterie par le thème "Économie"**

Dans ce cas, nous attribuerons un poids de 0 aux critères qui ne font pas partie du thème "Économie". Pour les critères de ce thème, les valeurs des TrueWeight seront ajustées de manière à ce que la somme des poids soit égale à 1.

**Prométhée :**

- Pour Prométhée 1 :  $\Phi^+ = C1 > C4 > C2 > C3 > C6 > C5$   $\Phi^- = C2 > C3 > C4 > C1 > C6 > C5$ .
- Pour Prométhée 2 :  $\Phi = C2 > C4 > C1 > C3 > C6 > C5$ .
- Pour Prométhée 1 avec seuil :  $\Phi^+ = C2 > C4 > C1 > C3 > C6 > C5$   $\Phi^- = C4 > C3 > C2 > C1 > C6 > C5$ .
- Pour Prométhée 2 avec seuil :  $\Phi = C4 > C1 > C2 > C3 > C6 > C5$ .

**Résultat d'Electre**

- Pour Electrelv, le noyau est C2.
- Pour Electrels, le noyau est C4.

**Résultat de Somme pondérée**

Le résultat de la méthodologie de somme pondérée est le suivant :

- $C2 > C4 > C3 > C1 > C6 > C5$

À partir des différents résultats des méthodologies, les meilleurs candidats identifiés sont C1, C2 et C4. En comptabilisant le nombre de fois où chaque candidat est considéré comme le meilleur choix, nous pouvons conclure que, pour le thème de l' "Économie", le candidat C2 représente la meilleure solution.

**4.1.5 Interprétation des différents résultats**

Pour résumer, le candidat C4 serait le meilleur choix pour une déchetterie en général ou selon le thème de l' "Acceptabilité technique". Le candidat C2 serait le meilleur pour une déchetterie selon les thèmes de l' "Économie" ou de l' "Environnement socio-économique". Enfin, le candidat C6 se démarquerait comme le meilleur pour le thème de l' "Environnement naturel".

Nous pouvons donc conclure qu'il n'existe pas de candidat unique qui soit le meilleur sur tous les thèmes. Bien que C4 apparaisse comme le meilleur candidat en général, il n'est pas forcément le meilleur selon certains critères spécifiques. Cette analyse a toutefois permis de réduire le nombre de candidats à trois.

Il sera donc nécessaire de faire un choix en fonction des priorités retenues pour la sélection de la déchetterie, en tenant compte des critères les plus importants pour le projet.

## 4.2 Sélection de la meilleure voiture

Pour pouvoir choisir la meilleure voiture, il faut savoir quelles sont les caractéristiques préférables pour l'acheteur. Ce choix sera fait lors de la sélection des poids selon les différents critères. Ensuite, il faut choisir ce que l'on veut faire pour un critère. Par exemple, on veut avoir le prix le plus bas ou le plus haut. À la suite de la lecture du type de critère, nous avons donc décidé du type de travail à effectuer sur ces critères :

- Minimisation du prix.
- Maximisation de la vitesse maximale.
- Minimisation de la consommation moyenne.
- Minimisation de la distance de freinage.
- Maximisation du confort.
- Maximisation du volume du coffre.
- Minimisation du temps d'accélération pour atteindre 100 km/h.

Les candidats pour les voitures seront représentés sous la forme  $C^*$ . Nous aurons donc les candidats suivants :

- Alfa Romeo 156 2.4 JTD = C1
- Audi A4 1.9 TDI = C2
- Citroën Xantia 2.1 TD = C3
- Peugeot 406 2.1 TD SV = C4
- Saab 9.3 TID = C5
- Renault Laguna 2.2 dT RXT = C6
- Volkswagen Passat 1.9 TDI Sport = C7
- BMW 320d = C8
- Citroën Wara 2.0 HDi = C9
- Renault Safrane 2.2 dt RXE = C10

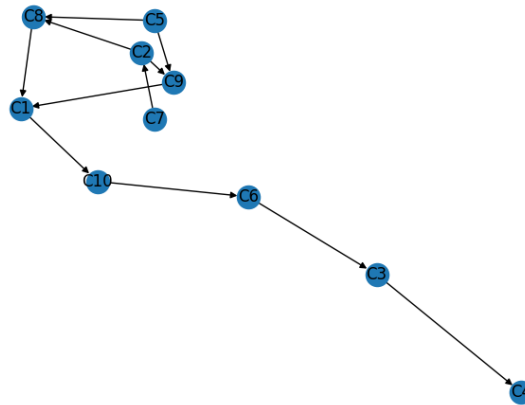
Voici les seuils de préférence que nous allons utiliser pour les différents critères :

- Pour le prix , seuil de préférence = 500.
- Pour la Vitesse maximal , seuil de préférence = 20.
- Pour la Consommation Moyenne , seuil de préférence = 1.
- Pour la Distance de freinage , seuil de préférence = 2.
- Pour le Confort , seuil de préférence = 1,5.
- Pour le Volume de Coffre , seuil de préférence = 20.
- Pour le temps d'accélération , seuil de préférence = 0,5.

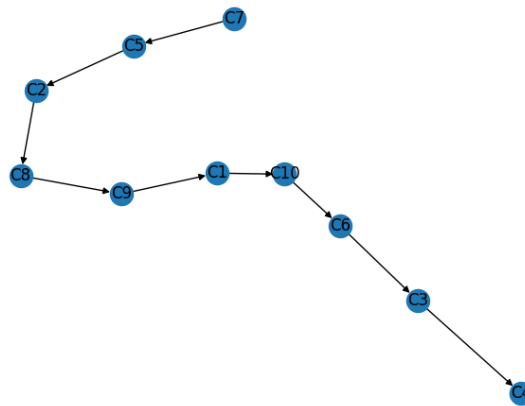
### Résultat de Prométhée

- Pour Prométhée 1 :  $\Phi^+ = C5 > C7 > C2 > C9 > C8 > C1 > C10 > C6 > C3 > C4$   $\Phi^- = C7 > C2 > C5 > C8 > C9 > C1 > C10 > C6 > C3 > C4$ .
- Pour Prométhée 2 :  $\Phi = C7 > C5 > C2 > C8 > C9 > C1 > C10 > C6 > C3 > C4$ .
- Pour Prométhée 1 avec seuil :  $\Phi^+ = C9 > C2 > C7 > C8 > C5 > C1 > C10 > C6 > C3 > C4$  et  $\Phi^- = C2 > C7 > C5 > C8 > C6 > C1 > C9 > C10 > C3 > C4$
- Pour Prométhée 2 avec seuil :  $\Phi = C2 > C7 > C9 > C5 > C8 > C1 > C6 > C10 > C3 > C4$

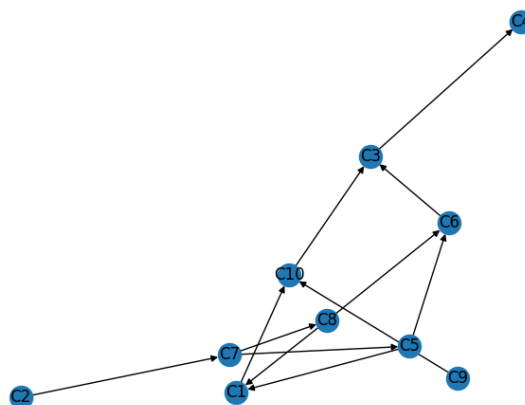
Voici les résultats de Prométhée sous forme graphique :



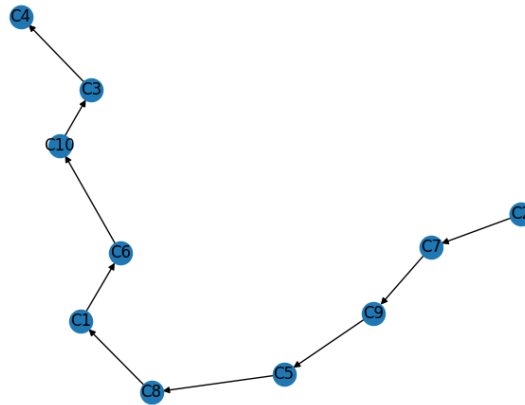
**Figure 11.** Résultat de ProméthéeI sur les données des déchetterie



**Figure 12.** Résultat de ProméthéeII sur les données des déchetterie



**Figure 13.** Résultat de ProméthéeI avec seuil sur les données des déchetterie



**Figure 14.** Résultat de ProméthéeII avec seuil sur les données des déchetterie

Avec ces graphiques, nous pouvons donc conclure qu'avec la méthode de Prométhée I, C5 et C7 sont les candidats qui ne sont surclassés par aucun autre candidat. En utilisant la méthode de Prométhée II, c'est le candidat C7 qui surclasse les autres candidats. Pour le cas avec des seuils de préférence, on a pour Prométhée I, C9 et C2 qui sont les meilleurs candidats. Et pour Prométhée II, on a C2 qui est le meilleur candidat.

### Résultat d'Electre

Pour le cas d'Electre, nous allons définir un veto spécifique pour chaque caractéristique. Ces différents veto sont définis d'après nous. On peut avoir des valeurs totalement différentes selon les envies de l'acheteur. Voici les différentes valeurs que nous avons définies :

- Pour le prix, veto = 3000.
- Pour la Vitesse maximal, veto = 69.
- Pour la Consommation Moyenne, veto = 2.
- Pour la Distance de freinage, veto = 42.
- Pour le Confort, veto = 3.
- Pour le Volume de Coffre, veto = 50.
- Pour le temps d'accélération, veto = 2.

Voici donc les noyaux obtenus avec un seuil de concordance à 0,5.

- Pour ElectreIV, le noyau est C1, C4, C9.
- Pour ElectreI, le noyau est C1, C7, C9.

Voici les résultats d'Electre sous forme graphique :

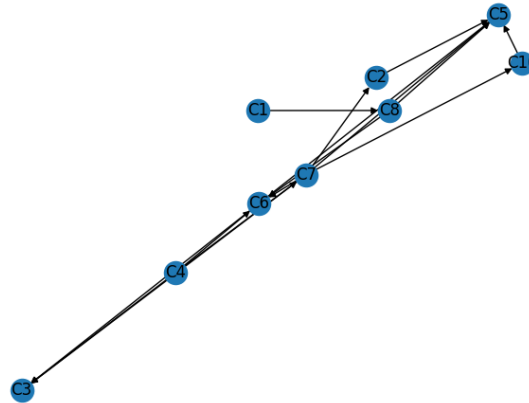


Figure 15. Résultat de ElectreIV sur les données des Voitures

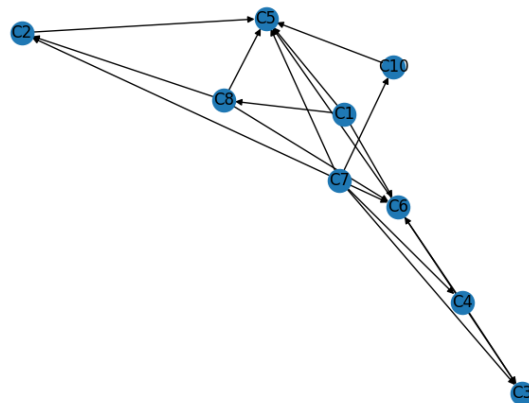


Figure 16. Résultat de ElectreIS sur les données des Voitures

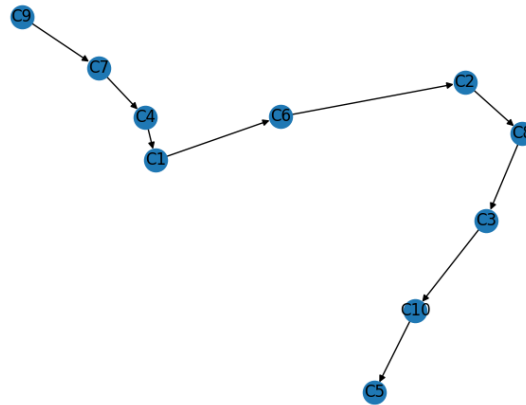
D'après les résultats des méthodes Electre, nous pouvons observer qu'avec Electre IV, les candidats C1, C4 et C9 ne sont surclassés par aucun autre candidat. Avec la méthode Electre IS, les candidats C1, C7 et C9 ne sont surclassés par aucun autre candidat.

### Résultat de Somme pondérée

Le résultat de la méthodologie de somme pondérée est le suivant :

- $C9 > C7 > C4 > C1 > C6 > C2 > C8 > C3 > C10 > C5$

Voici les résultats sous forme graphique :



**Figure 17.** Résultat de la somme pondérée sur les données des Voitures

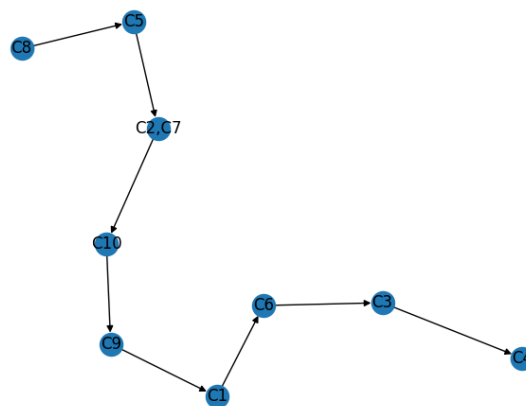
Le candidat C9 est donc le meilleur d'après la méthodologie de somme pondérée.

### Résultat de Borda

Le résultat de la méthodologie de borda est le suivant :

- $C8 > C5 > C2 > C7 > C10 > C9 > C1 > C6 > C3 > C4$

Voici les résultats sous forme graphique :



**Figure 18.** Résultat de Borda sur les données des Voitures

Le candidat C8 est donc le meilleur d'après la méthodologie de borda.

### Comparaison des résultats

D'après les différentes méthodologies, voici les candidats sélectionnés : C1, C2, C4, C5, C7, C8 et C9. Si nous cherchons à identifier le candidat qui a été le plus souvent considéré comme le meilleur à travers les différentes méthodologies, nous constatons que C9 se distingue comme étant le meilleur candidat.

Dans les prochaines sous-parties, nous examinerons le meilleur candidat en fonction des différents thèmes que nous avons choisis : étudiant et personne âgée. Nous ne réaliserons pas de méthode de Borda sur ces données, car le résultat serait identique à celui précédemment présenté.

#### 4.2.1 Sélection de la meilleur voiture pour les Étudiants

Dans ce cas, nous nous plaçons dans la perspective d'un étudiant, en priorisant l'importance du prix par rapport aux autres critères. Il s'agit uniquement d'un exemple : d'autres



étudiants pourraient accorder moins d'importance au prix ou, au contraire, privilégier davantage la consommation moyenne. Nous aurons donc les poids suivants :

- Pour le prix , le poids sera 0.8.
- Pour la vitesse maximal , le poids sera 0.01.
- Pour la consommation moyenne , le poids sera 0.075.
- Pour la distance de freinage , le poids sera 0.01.
- Pour le confort , le poids sera 0.075.
- Pour le volume de coffre , le poids sera 0.02.
- Pour la vitesse d'accélération , le poids sera 0.01.

#### Résultat de Prométhée

- Pour Prométhée 1 :  $\Phi^+ = C9 > C7 > C6 > C1 > C3 > C2 > C4 > C5 > C8 > C10$   $\Phi^- = C9 > C7 > C6 > C1 > C3 > C2 > C4 > C5 > C8 > C10$ .
- Pour Prométhée 2 :  $\Phi = C9 > C7 > C6 > C1 > C3 > C2 > C4 > C5 > C8 > C10$ .
- Pour Prométhée 1 avec seuil :  $\Phi^+ = C9 > C7 > C6 > C1 > C2 > C3 > C4 > C8 > C5 > C10$  et  $\Phi^- = C9 > C7 > C6 > C1 > C2 > C3 > C4 > C5 > C8 > C10$
- Pour Prométhée 2 avec seuil :  $\Phi = C9 > C7 > C6 > C1 > C2 > C3 > C4 > C5 > C8 > C10$

#### Résultat d'Electre

Voici donc les noyaux obtenus avec un seuil de concordance à 0,5.

- Pour Electrelv, le noyau sont C1, C7, C9.
- Pour Electrels, le noyau sont C1, C7, C9.

#### Résultat de Somme pondérée

Le résultat de la méthodologie de somme pondérée est le suivant :

- $C9 > C7 > C6 > C1 > C3 > C2 > C4 > C8 > C5 > C10$

D'après les différentes méthodologies, voici les candidats sélectionnés : C1, C7, C8 et C9. Si nous cherchons à identifier le candidat qui a été le plus souvent considéré comme le meilleur à travers les différentes méthodologies, nous constatons que C9 se démarque comme le meilleur candidat.

### 4.2.2 Sélection de la meilleur voiture pour les personnes âgées

Ici, nous nous plaçons dans la perspective d'une personne âgée, en priorisant l'importance du confort par rapport aux autres critères. Il s'agit uniquement d'un exemple : d'autres personnes âgées pourraient accorder moins d'importance au confort ou, au contraire, privilégier davantage la vitesse. Nous aurons donc les poids suivants :

- Pour le prix , le poids sera 0.01.
- Pour la vitesse maximal , le poids sera 0.02.
- Pour la consommation moyenne , le poids sera 0.03.
- Pour la distance de freinage , le poids sera 0.03.
- Pour le confort , le poids sera 0.8.
- Pour le volume de coffre , le poids sera 0.1.
- Pour la vitesse d'accélération , le poids sera 0.01.

#### Résultat de Prométhée

- Pour Prométhée 1 :  $\Phi^+ = C9 > C2 > C5 > C1 > C8 > C6 > C7 > C3 > C10 > C4$   $\Phi^- = C9 > C2 > C5 > C1 > C8 > C6 > C7 > C3 > C4 > C10$ .

- Pour Prométhée 2 :  $\Phi = C9 > C2 > C5 > C1 > C8 > C6 > C7 > C3 > C4 > C10$ .
- Pour Prométhée 1 avec seuil :  $\Phi^+ = C9 > C2 > C1 > C5 > C8 > C6 > C7 > C3 > C10 > C4$  et  $\Phi^- = C2 > C9 > C5 > C1 > C8 > C6 > C7 > C3 > C4 > C10$
- Pour Prométhée 2 avec seuil :  $\Phi = C9 > C2 > C1 > C5 > C8 > C6 > C7 > C3 > C4 > C10$

### Résultat d'Electre

Voici donc les noyaux obtenus avec un seuil de concordance à 0,5.

- Pour Electrelv, le noyau sont C1, C4, C8, C9, C10.
- Pour Electrels, le noyau est C1, C4, C8, C9, C10.

### Résultat de Somme pondérée

Le résultat de la méthodologie de somme pondérée est le suivant :

- $C10 > C4 > C3 > C7 > C8 > C6 > C5 > C1 > C2 > C9$

Le résultat est identique à celui obtenu lors de la sélection de la meilleure voiture en général (voir le résultat 4.2).

D'après les différentes méthodologies, voici les candidats sélectionnés : C1, C2, C4, C8, C9, et C10. Si nous cherchons à identifier le candidat qui a été le plus souvent considéré comme le meilleur à travers les différentes méthodologies, nous constatons que C9 se démarque comme étant le meilleur candidat.

#### 4.2.3 Interprétation des différents résultat

D'après les différents résultats obtenus avec nos jeux de poids, de veto et de seuils de préférence, nous pouvons affirmer que, dans ces cas, le candidat C9, la Citroën Wara 2.0 HDi, serait la meilleure voiture à choisir. Cependant, ce résultat dépend étroitement des choix que nous avons faits en termes de seuils, de poids et de veto. Selon les préférences des acheteurs, ces valeurs pourraient être totalement différentes, ce qui entraînerait également des résultats possiblement différents.

## 5 Conclusion

### 5.1 Général

Dans ce projet, nous avons réussi à sélectionner la meilleure déchetterie ainsi que le meilleur modèle de voiture en utilisant des méthodologies multicritères telles que Prométhée, Electre, la somme pondérée et la méthode de Borda. Ces techniques nous ont permis d'analyser et de comparer les différentes options en fonction de multiples critères, en fournissant des résultats fiables et pertinents.

Cependant, plusieurs pistes d'amélioration peuvent être envisagées pour de futurs projets. Premièrement, l'automatisation du processus de fusion des données et de l'application des méthodologies via des scripts ou un tableau de bord interactif pourrait considérablement améliorer l'efficacité et la fluidité du travail. L'interface console utilisée dans ce projet, bien que fonctionnelle, pourrait être remplacée par un système plus dynamique et intuitif, ce qui permettrait de traiter plus rapidement et efficacement des ensembles de données plus vastes. De plus, intégrer des outils permettant de manipuler, ajouter, fusionner ou corriger les données de manière plus interactive offrirait une plus grande flexibilité et faciliterait l'adaptation aux besoins spécifiques des utilisateurs.

## 5.2 Florian

En rétrospective, je trouve que ce projet a été intéressant. Il m'a permis de découvrir les bases de l'aide à la décision tout en développant et en implémentant différents algorithmes liés à ce domaine. Cependant, je dois souligner une difficulté majeure : l'absence de cours préalable sur l'aide à la décision avant de débiter ce projet. Cela a considérablement ralenti ma contribution initiale, car j'ai dû investir du temps supplémentaire pour comprendre les concepts fondamentaux et m'adapter au projet. Malgré cela, je suis satisfait du travail accompli et de l'organisation de l'équipe. Nous avons su répartir efficacement les tâches en fonction des compétences de chacun.

## 5.3 Maxime

De mon point de vue, le projet était bien structuré. Nous avons réussi à nous répartir les tâches de manière convenable, et chacun a pu travailler sur toutes les parties du programme, allant du parseur à l'algorithme d'optimisation. Je suis satisfait du travail accompli par nous deux. Le seul point noir de cette matière est que, selon la partie que nous aurions choisie, l'un de nous aurait dû consacrer plus de temps à comprendre les méthodes et les algorithmes à mettre en œuvre. Il aurait été préférable que, pour les deux parties (aide à la décision ou fouille de données), nous ayons eu accès aux cours associés, même si nous appartenons à des formations différentes.