UNIVERSITÉ LIBRE DE BRUXELLES

INFO-F310

Algorithmique et recherche opérationnelle

Optimisation du Transport Multi-Articles : Approches Agrégée et Désagrégée

Étudiants :

Nieto Navarrete Matias - 502920

Gracki Konrad - 500020

2023 - 2024

Résumé

Ce rapport présente les résultats d'une étude comparative entre deux formulations de programmation linéaire - agrégée et désagrégée - appliquées au problème de transport multi-articles. Nous expliquons les formulations mathématiques de chaque modèle, justifions les choix de variables et de contraintes, et analysons les résultats obtenus à partir de différentes instances de problème.

Table des matières

1 Introduction **3**

2 Formulation des Programmes Linéaires **3**
2.1 Approche Agrégée . . . 3
2.1.1 Fonction Objectif . . . 3
2.1.2 Contraintes de Flux et de Capacité . . . 4
2.1.3 Bornes et Types des Variables . . . 5
2.2 Approche Désagrégée . . . 5
2.2.1 Fonction Objectif . . . 5
2.2.2 Contraintes de Flux et de Capacité . . . 6
2.2.3 Bornes et Types des Variables . . . 7

3 Analyse des Résultats **7**
3.1 Comparaison des Modèles . . . 7
3.1.1 Valeurs de la Fonction Objectif . . . 7
3.1.2 Contraintes . . . 9
3.1.3 Temps de Résolution . . . 10
3.2 Analyse des Instances avec Solution Nulle . . . 10
3.3 Interprétation d'un Fichier .sol . . . 11
3.3.1 Contenu du Fichier .sol avec la méthode agrégée . . . 11
3.3.2 Explication des Sections . . . 12
3.3.3 Vérification de Qualité par les Conditions KKT . . . 12

4 Utilisation d'Outils Externes **12**

5 Conclusion **13**

1 Introduction

Optimiser les systèmes de transport est essentiel pour booster l'efficacité de notre logistique et diminuer les dépenses. Dans ce rapport, nous examinons deux méthodes distinctes de programmation linéaire pour relever ce défi. D'un côté, l'approche agrégée, qui simplifie les choses en traitant tous les articles comme un seul et même bloc, et de l'autre, l'approche désagrégée, qui prend en compte chaque type d'article séparément, permettant ainsi une gestion plus fine et adaptée à chaque situation. Notre objectif est de voir quelle stratégie permet de réduire au mieux les coûts de transport tout en respectant les limites de capacité et les besoins spécifiques de chaque point du réseau.

2 Formulation des Programmes Linéaires

Dans cette section, nous détaillons les formulations des programmes linéaires pour les deux approches — agrégée et désagrégée — utilisées pour résoudre le problème de transport multi-articles avec nœuds intermédiaires. Nous expliquons les notations, les variables, les contraintes et les fonctions objectif spécifiques à chaque méthode, en soulignant comment elles modélisent intégralement le problème.

2.1 Approche Agrégée

L'approche agrégée traite les différents types d'articles comme **une seule entité**, simplifiant ainsi le problème de transport multi-articles en éliminant la nécessité de distinguer entre les types d'articles. Cette approche réduit la complexité du modèle et se concentre sur le flux global de marchandises à travers le réseau.

2.1.1 Fonction Objectif

La fonction objectif du modèle agrégé cherche à minimiser le coût total du transport sur l'ensemble du réseau. Elle est formulée comme suit :

$$\text{Minimize} \sum_{\text{edges}} \text{cost}_{\text{edge}} \times x_{\text{edge\_id}}$$

où $\text{cost}_{\text{edge}}$ représente le coût médian de transport sur chaque arc, calculé à partir des coûts associés à différents types d'articles. Cette formulation aide à obtenir une estimation équilibrée du coût. Et $x_{\text{edge\_id}}$ est la variable de décision qui mesure la quantité de marchandises transportées sur l'arc edge_id. Cette variable de flux permet de calculer le coût total du transport en tenant compte de chaque arc spécifique du réseau.

2.1.2 Contraintes de Flux et de Capacité

Dans un réseau de transport modelé comme un graphe bidirectionnel, les arcs peuvent représenter des routes de transport où les marchandises peuvent circuler dans les deux sens. Cette bidirectionalité implique que pour tout nœud (que ce soit une source, une destination ou un nœud intermédiaire), le flux de marchandises entrant peut être différent du flux sortant en fonction des besoins de transport et de la distribution logistique. Par conséquent, il est crucial de considérer la différence entre les flux sortants et entrants pour chaque nœud afin de garantir la précision de la modélisation des flux de transport.

— **Pour les sources**, cette différence permet de s'assurer que la quantité totale des marchandises expédiées ne dépasse pas la capacité maximale disponible à la source. La contrainte est formulée comme suit :

$$\text{cap\_s} = \sum_{\text{outgoing edges}} x_{\text{edge\_id}} - \sum_{\text{incoming edges}} x_{\text{edge\_id}} \leq \text{capacité totale de } s$$

où cap_s représente une contrainte spécifique à chaque source, capacité totale de s est la capacité totale disponible à la source considérée, et $x_{\text{edge\_id}}$ sont les flux sur les arcs entrants et sortants. Cette formulation prend en compte non seulement les marchandises expédiées depuis la source, mais aussi celles qui pourraient y retourner à travers des arcs entrants.

— **Pour les nœuds intermédiaires**, l'équilibre des flux est essentiel pour assurer que toute marchandise qui entre dans un nœud en ressort, respectant ainsi les principes de conservation des flux. Chaque nœud intermédiaire est soumis à la contrainte suivante :

$$\text{flow\_id} = \sum_{\text{incoming edges}} x_{\text{edge\_id}} - \sum_{\text{outgoing edges}} x_{\text{edge\_id}} = 0$$

où chaque terme $x_{\text{edge\_id}}$ représente le flux sur un arc entrant ou sortant du nœud intermédiaire. Cette condition garantit que la quantité totale entrante dans chaque nœud intermédiaire équivaut à la quantité totale sortante, reflétant un équilibre parfait de flux.

— **Pour les destinations**, la contrainte assure que la quantité de marchandises reçue moins celle expédiée est suffisante pour répondre à la demande locale, même si la destination agit également comme un point de transit. Pour chaque destination, cette contrainte est formulée comme suit :

$$\text{demand\_d} = \sum_{\text{incoming edges}} x_{\text{edge\_id}} - \sum_{\text{outgoing edges}} x_{\text{edge\_id}} \geq \text{total demand de } d$$

2.1.3 Bornes et Types des Variables

Dans notre modèle de programmation linéaire, les variables de décision $x_{\text{edge\_id}}$ représentent les quantités de marchandises transportées le long des arcs. Ces variables sont contraintes à être non-négatives et sans limite supérieure explicite, comme indiqué ci-dessous :

$$0 \leq x_{\text{edge\_id}} \leq +\infty$$

Cette spécification permet une adaptation flexible aux divers scénarios de demande, sans restriction de capacité supérieure prédéfinie. Par ailleurs, pour refléter la réalité des opérations de transport où les marchandises ne peuvent pas être fractionnées, les variables sont définies comme entières générales dans le modèle :

$$x_0, x_1, x_2, \ldots$$

Cette approche garantit que les flux que nous modélisons correspondent à des unités complètes qui peuvent être physiquement transportées, assurant ainsi des solutions pratiques et efficaces sur le plan logistique.

2.2 Approche Désagrégée

L'approche désagrégée traite chaque type d'article **individuellement**, ce qui permet une modélisation plus précise des flux spécifiques et répond à des besoins logistiques détaillés pour chaque type d'article. Cette approche complexifie le modèle en conservant la distinction entre les différents articles à travers le réseau.

2.2.1 Fonction Objectif

Dans l'approche désagrégée, la fonction objectif vise à minimiser le coût total de transport en tenant compte des spécificités de chaque type d'article sur chaque arc du réseau. Contrairement à l'approche agrégée, chaque type d'article est traité séparément, permettant une analyse plus détaillée et précise des coûts de transport. La fonction objectif est donc formulée comme suit :

$$\text{Minimize} \sum_{\text{edges},i} \text{cost}_{\text{edge},i} \times x_{\text{edge\_id},i}$$

où $\text{cost}_{\text{edge},i}$ est le coût spécifique au type d'article i sur chaque arc, et $x_{\text{edge\_id},i}$ représente la variable de décision indiquant la quantité de l'article i transportée sur l'arc identifié par edge_id. Cette structure de variable permet de distinguer les flux de chaque type d'article individuellement, assurant ainsi une modélisation précise des coûts et des contraintes associées à chaque type d'article sur chaque segment du réseau.

2.2.2 Contraintes de Flux et de Capacité

Les contraintes pour chaque type d'article tiennent compte de la nature bidirectionnelle du réseau de transport, où les marchandises peuvent circuler dans les deux sens sur les mêmes arcs, nécessitant une gestion précise des flux entrants et sortants pour chaque nœud.

— **Pour les sources**, dans le modèle désagrégé, chaque source et chaque type d'article sont considérés séparément, ce qui permet d'assurer que la capacité de production pour chaque type d'article n'est pas dépassée, tout en intégrant la possibilité de retour de marchandises. Pour chaque source s et chaque type d'article i, la somme des flux sortants moins les flux entrants est contrainte par la capacité maximale disponible pour ce type à cette source. La contrainte est exprimée comme suit :

  Pour chaque source s et type i,

$$\text{cap\_s\_i} = \sum_{\text{outgoing edges}} x_{\text{edge\_id},i} - \sum_{\text{incoming edges}} x_{\text{edge\_id},i} \leq \text{capacity}_{s,i}$$

— **Pour les nœuds intermédiaires**, l'équilibre des flux est essentiel pour assurer que toute marchandise qui entre dans un nœud en ressort, respectant ainsi les principes de conservation des flux pour chaque type d'article. Chaque nœud intermédiaire est soumis à la contrainte suivante pour chaque type d'article i :

$$\text{flow\_id\_i} = \sum_{\text{incoming edges}} x_{\text{edge\_id},i} - \sum_{\text{outgoing edges}} x_{\text{edge\_id},i} = 0$$

  où $x_{\text{edge\_id},i}$ représente le flux de l'article i sur un arc entrant ou sortant du nœud intermédiaire. Cette condition garantit que la quantité totale d'un type d'article i entrante dans chaque nœud intermédiaire équivaut à la quantité totale sortante pour ce même type, reflétant un équilibre parfait de flux pour chaque catégorie d'article.

— **Pour les destinations**, il est essentiel de gérer avec précision les flux entrants et sortants pour répondre aux besoins spécifiques de chaque type d'article. Chaque destination doit non seulement satisfaire sa propre demande, mais peut également agir comme un point de transit. Ainsi, la contrainte appliquée à chaque destination et pour chaque type d'article est définie de la manière suivante :
  Pour chaque demande d et type i,

$$\text{demand\_d\_i} = \sum_{\text{incoming edges}} x_{\text{edge\_id},i} - \sum_{\text{outgoing edges}} x_{\text{edge\_id},i} = \text{demand}_{d,i}$$

  où $x_{\text{edge\_id},i}$ indique le flux de l'article i arrivant ou partant de la destination d. Cette formulation assure que chaque destination reçoit la quantité exacte requise de chaque type d'article, conformément à la demande spécifiée, qui peut être un chiffre positif, nul ou même un besoin de surplus. Cette flexibilité permet de gérer efficacement les cas où certaines destinations peuvent ne pas nécessiter un type d'article spécifique ou pourraient avoir besoin de gérer des retours ou des surplus de manière stratégique.

2.2.3 Bornes et Types des Variables

Dans le modèle désagrégé de programmation linéaire, les variables de décision $x_{\text{edge\_id},i}$ représentent les quantités de chaque type d'article i transportées le long des arcs spécifiés par edge_id. Ces variables sont soumises à des contraintes de non-négativité, et aucune limite supérieure n'est imposée, comme indiqué ci-dessous pour chaque type d'article sur chaque arc :

$$0 \leq x_{\text{edge\_id},i} \leq +\infty$$

Cette approche nous permet de nous adapter facilement à différentes situations pour adapter le modèle à une variété de scénarios de demande sans imposer de restrictions sur la capacité de transport maximale. De plus, pour maintenir l'alignement avec les opérations de transport réelles où les articles sont souvent transportés en quantités entières et non divisibles, les variables dans ce modèle sont définies comme entières générales :

$$x_{0_0}, x_{0_1}, x_{1_0}, x_{1_1}, x_{2_0}, x_{2_1}, \ldots$$

Cette classification des variables comme entières assure que les quantités transportées dans le modèle correspondent à des envois réels, qui doivent être comptabilisés en unités complètes pour refléter précisément la réalité logistique. Chaque combinaison de edge_id et de type d'article i est traitée individuellement, garantissant que les spécifications du transport pour chaque type d'article sont respectées et que les solutions générées sont pratiques et logistiquement viables.

3 Analyse des Résultats

Dans cette section, nous analysons et comparons les performances des modèles agrégé et désagrégé pour le problème de transport multi-articles avec nœuds intermédiaires. Nous discutons les résultats des fonctions objectifs, les temps de résolution pour des instances de taille croissante, et l'impact de contraintes additionnelles. Cette analyse vise à déterminer le modèle le plus efficace et adaptable aux divers scénarios opérationnels.

3.1 Comparaison des Modèles

3.1.1 Valeurs de la Fonction Objectif

Nous comparons les valeurs obtenues pour la fonction objectif des deux modèles sur différentes instances. Pour l'instance **test.txt**, le modèle agrégé, qui regroupe tous les articles en un seul type, affiche une fonction objectif :

$$obj : +1.0x0 + 1.0x1 + 1.0x2 + 1.0x3 + 1.0x4$$

En revanche, le modèle désagrégé, traitant chaque article séparément, montre une fonction objectif plus détaillée :

$$obj : +1.0x0\_0 + 1.0x0\_1 + 1.0x1\_0 + 1.0x1\_1 + 1.0x2\_0 \\ + 1.0x2\_1 + 1.0x3\_0 + 1.0x3\_1 + 1.0x4\_0 + 1.0x4\_1$$

Cette différence montre la complexité plus élevée du modèle désagrégé, où le nombre de variables dépend du nombre d'arcs et du nombre d'articles, chaque combinaison ayant un coût spécifique.

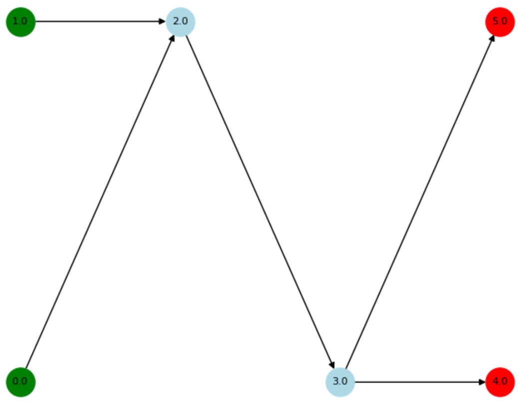

FIGURE 1 – Graphique de l'instance **test.txt**

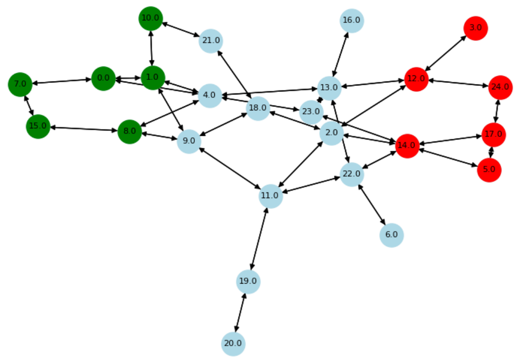

FIGURE 2 – Graphique de l'instance **25_4_ducdame.txt**

Pour une instance plus grande, telle que **25_4_ducdame.txt**, le même principe s'applique, mais la fonction objectif est plus étendue en raison de la taille accrue de l'instance. Le modèle agrégé, par exemple, peut inclure des valeurs négatives qui réduisent le coût total du transport, comme montré ici :

$$obj : +117.0x0 + 147.0x1 + 124.5x2 + ... + 109.0x61 - 35.0x62 + ... + 128.0x75$$

Ces coûts négatifs indiquent que le passage de certains articles par certains arcs permet de diminuer les coûts de transport.

Dans le modèle désagrégé pour la même instance, la fonction objective est construite de la même manière que celle de l'instance **test.txt**. La taille de cette fonction objective dépend directement du nombre d'arcs multiplié par le nombre d'articles présents dans le fichier texte, montrant la complexité de ce modèle par rapport au modèle agrégé.

3.1.2 Contraintes

Si nous observons les contraintes créées pour le modèle agrégé du fichier `test.txt`, nous avons :

— **Capacités** :

$$\begin{aligned}\text{cap\_0} &: x0 \leq 30\\ \text{cap\_1} &: x1 \leq 20\end{aligned}$$

— **Conservation du flux** :

$$\begin{aligned}\text{flow\_2} &: x0 + x1 - x4 = 0\\ \text{flow\_3} &: x4 - x2 - x3 = 0\end{aligned}$$

— **Demandes** :

$$\begin{aligned}\text{demand\_4} &: x2 = 40\\ \text{demand\_5} &: x3 = 10\end{aligned}$$

Tandis que pour le modèle désagrégé, les contraintes sont spécifiées comme suit :

— **Capacités par type d'article** :

$$\begin{aligned}\text{cap\_0\_0} &: x0\_0 \leq 10\\ \text{cap\_0\_1} &: x0\_1 \leq 20\\ \text{cap\_1\_0} &: x1\_0 \leq 5\\ \text{cap\_1\_1} &: x1\_1 \leq 15\end{aligned}$$

— **Conservation du flux par type d'article** :

$$\begin{aligned}\text{flow\_2\_0} &: x0\_0 + x1\_0 - x4\_0 = 0\\ \text{flow\_2\_1} &: x0\_1 + x1\_1 - x4\_1 = 0\\ \text{flow\_3\_0} &: x4\_0 - x2\_0 - x3\_0 = 0\\ \text{flow\_3\_1} &: x4\_1 - x2\_1 - x3\_1 = 0\end{aligned}$$

— **Demandes par type d'article** :

$$\begin{aligned}\text{demand\_4\_0} &: x2\_0 = 10\\ \text{demand\_4\_1} &: x2\_1 = 30\\ \text{demand\_5\_0} &: x3\_0 = 5\\ \text{demand\_5\_1} &: x3\_1 = 5\end{aligned}$$

L'analyse des contraintes des modèles agrégé et désagrégé montre un choix entre simplicité et précision. Le modèle agrégé, en raison de son nombre réduit de variables et de contraintes, permet de trouver rapidement des solutions et est plus simple à gérer, ce qui le rend parfait pour les situations où il n'est pas nécessaire de différencier minutieusement les articles. En revanche, le modèle désagrégé, bien qu'il nécessite plus d'efforts à cause de sa complexité et de son grand nombre de variables, offre une gestion plus fine et détaillée des différents types d'articles, répondant avec précision aux exigences variées.

3.1.3 Temps de Résolution

Les temps de résolution pour les modèles agrégé et désagrégé ont été mesurés sur différentes instances. Cette analyse cherche à évaluer la performance des modèles face à une complexité croissante des données. Les résultats, présentés ci-dessous, montrent les temps moyens de résolution ainsi que les écarts-types pour chaque instance, illustrant la capacité de chaque modèle à gérer efficacement des volumes de données de plus en plus importants.

Instance	**Temps Moyen (s)**	**Écart-Type (s)**
Test	0.73	0.08
22_3_mycoses	0.71	0.01
25_4_ducdame	0.71	0.01

Table 1 – Temps de résolution pour le modèle agrégé

Instance	**Temps Moyen (s)**	**Écart-Type (s)**
Test	0.73	0.06
22_3_mycoses	0.76	0.03
25_4_ducdame	0.76	0.05

Table 2 – Temps de résolution pour le modèle désagrégé

Les analyses quantitatives nous permettent de tirer des conclusions sur la performance algorithmique et la scalabilité des modèles. Bien que les temps de résolution soient assez similaires entre les deux modèles, les écarts-types révèlent des variations dans la stabilité des temps de réponse, en particulier dans le modèle désagrégé où la variabilité est légèrement réduite.

Les analyses quantitatives nous fournissent des informations précieuses sur la performance et la capacité d'évolution des modèles. Même si les temps de résolution semblent proches pour les deux modèles, les écarts-types nous montrent qu'il y a des différences dans la constance des temps de réponse. C'est particulièrement le cas avec le modèle désagrégé, où l'on observe une légère diminution de la variabilité.

3.2 Analyse des Instances avec Solution Nulle

Nous observons plusieurs instances qui aboutissent à une solution de coût total nul :

— Pour l'instance `20_3_didactics.txt`, nous observons des solutions de coût total nul pour les modèles agrégé et désagrégé. Cette conséquence est causée par la bidirectionnalité du graphe et le fait que certains arcs présentent des coûts négatifs. En examinant le fichier texte, nous constatons que les arcs entre les nœuds 14 et 9 (IDs 41 et 42) permettent des échanges d'articles à coût négatif, ce qui conduit à une solution de coût total nul.

— L'instance `21_2_larcenist.txt` présente une situation similaire à celle observée dans `20_3_didactics.txt`. Toutefois, les arcs concernés ici sont ceux connectant les nœuds 12 à 18, avec les IDs 53 et 54 pour l'article 0. Ces connexions bidirectionnelles avec des coûts négatifs permettent également des des échanges d'articles à coût négatif, ce qui conduit à une solution de coût total nul le modèle désagrégé.

— Pour l'instance `25_3_alidade.txt`, nous rencontrons une particularité intéressante pour le modèle désagrégé : un arc cyclique (12 -> 12) associé à l'article 0 avec un cout negatif, identifié par l'ID 50. Dans le fichier LP correspondant, cette variable est déclarée sans contraintes spécifiques qui en limitent la valeur. En raison de son coût négatif, pour minimiser la fonction objectif, la valeur optimale de cette variable pourrait théoriquement tendre vers l'infini, conduisant ainsi à une réduction indéfinie du coût total. Cela explique pourquoi la solution obtenue pourrait suggérer un coût total nul, le modèle exploitant cette absence de contrainte pour diminuer le coût à l'extrême.

— Pour l'instance `25_4_ducdame.txt`, le modèle désagrégé présente une solution nulle. Cette situation résulte du fait que la demande pour l'article 2 dépasse l'offre disponible, rendant le problème insoluble. En conséquence, aucune allocation conforme aux contraintes n'est possible, ce qui se traduit par une absence de solution réalisable pour cette configuration.

3.3 Interprétation d'un Fichier .sol

Nous examinons ici le contenu d'un fichier de sortie `.sol` généré par GLPK pour l'instance `test.txt`, afin de détailler comment interpréter les informations qu'il contient.

3.3.1 Contenu du Fichier .sol avec la méthode agrégée

```
Problem:
Rows:        6
Columns:     5 (5 integer, 0 binary)
Non-zeros:   10
Status:      INTEGER OPTIMAL
Objective:   obj = 150 (MINimum)

   No.   Row name        Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 cap_0                     30                           30
     2 cap_1                     20                           20
     3 flow_2                     0             0             =
     4 flow_3                     0             0             =
     5 demand_4                  40            40             =
     6 demand_5                  10            10             =

   No. Column name       Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 x0           *            30             0
     2 x1           *            20             0
```

```
    3 x2          *           40          0
    4 x3          *           10          0
    5 x4          *           50          0

Integer feasibility conditions:

KKT.PE: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality

KKT.PB: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality
```

3.3.2 Explication des Sections

- **Status : INTEGER OPTIMAL** - Le solveur a trouvé une solution optimale entière, indiquant que toutes les contraintes sont satisfaites sans violations et que la solution est la meilleure possible parmi les entiers.
- **Objective : obj = 150 (MINimum)** - La valeur de la fonction objectif est minimisée à 150, ce qui est le meilleur résultat obtenu sous les contraintes données.
- **Rows et Columns** - Le problème consiste en 6 contraintes (rows) et 5 variables de décision (columns), toutes entières, sans variables binaires.
- **Activities dans Rows** - Les activités (valeurs assignées aux contraintes) montrent que toutes les contraintes sont exactement satisfaites, comme indiqué par les valeurs d'activité correspondant strictement aux bornes inférieures ou égales.
- **Activities dans Columns** - Chaque variable de décision atteint une valeur spécifique (activity) qui contribue à la satisfaction des contraintes et à la minimisation de la fonction objectif.

3.3.3 Vérification de Qualité par les Conditions KKT

Les conditions de Karush-Kuhn-Tucker (KKT) sont également vérifiées pour assurer la qualité de la solution :

- **KKT.PE et KKT.PB** - Les erreurs maximales absolues et relatives sont nulles, ce qui indique une haute qualité de la solution obtenue, avec des calculs précis et sans violations des conditions de primalité et de dualité.

Cette analyse détaillée du fichier `.sol` permet de comprendre non seulement les résultats spécifiques obtenus mais aussi la manière dont le solveur GLPK les a déterminés et vérifiés pour leur exactitude et leur optimalité.

4 Utilisation d'Outils Externes

Nous avons utilisé ChatGPT pour comprendre les fichier .sol, et pour corriger et reformuler les phrases de ce rapport.

5 Conclusion

Dans cette étude, nous avons exploré deux méthodes différentes pour optimiser les systèmes de transport : un modèle agrégé, qui est simple et direct, et un modèle désagrégé, qui est plus détaillé et précis. Le modèle agrégé se montre particulièrement pratique et efficace quand une extrême précision n'est pas cruciale, permettant ainsi de gagner du temps et de réduire les coûts. Par contre, si le projet nécessite une attention minutieuse aux détails, le modèle désagrégé est plus approprié car il peut s'adapter spécifiquement à chaque exigence logistique.

Les tests de performance ont montré que les deux modèles sont assez similaires en termes de temps de résolution, mais le modèle désagrégé est généralement plus stable. Le choix entre ces deux modèles dépendra donc de l'équilibre entre la nécessité de simplicité et la demande de précision, ainsi que des objectifs spécifiques et des conditions du projet.

En conclusion, il est essentiel de choisir le modèle de programmation linéaire adapté au contexte spécifique pour assurer l'efficacité des opérations de transport tout en optimisant les coûts et la performance.

Références

[1] Joshua Emmanuel. *Transshipment Problem - LP Formulation / Solution.* Disponible sur YouTube : `https://www.youtube.com/watch?v=ABMPgSApdUw`