Rapport de TP1
Modélisation et Résolution de PL / PLNE avec GLPK

Sommaire

Réalisé par :
Ralph Khairallah
Baptiste Rembert

Groupe de TP : M2
Date : December 26, 2025

1 Introduction

Ce rapport présente les travaux réalisés dans le cadre du premier sujet de TP de Recherche Opérationnelle. L'objectif principal est de mettre en application les concepts de modélisation mathématique vus en cours et de se familiariser avec l'utilisation d'outils de résolution numérique pour des problèmes d'optimisation sous contraintes.

Nous aborderons trois problèmes distincts : un problème d'assemblage de production, un problème d'affectation de tâches avec préférences, et enfin des problématiques logistiques liées à l'e-commerce (affectation de commandes et tournées de livraison).

1.1 Objectifs et Outils

Le but de ce TP est double : d'une part, formaliser des problèmes concrets sous forme de Programmes Linéaires (PL) ou de Programmes Linéaires en Nombres Entiers (PLNE); d'autre part, implémenter ces modèles informatiquement pour obtenir des solutions optimales.

Pour ce faire, nous utilisons le logiciel **GLPK**. Plus précisément, nous utilisons le solveur exécutable `glpsol`, capable de lire des fichiers de modèles, de résoudre les systèmes d'équations et de fournir les résultats détaillés.

La démarche adoptée pour chaque exercice est la suivante :

- Identification des variables de décision, de la fonction objectif et des contraintes.
- Écriture du modèle mathématique formel.
- Implémentation du modèle dans un fichier compatible avec GLPK.
- Résolution numérique et analyse de la cohérence des résultats.

1.2 Justification générale des formats (LP vs GMPL)

GLPK permet l'utilisation de deux formats principaux pour décrire les problèmes d'optimisation: le format standard (`.lp`) et le langage de modélisation GMPL (`.mod`. Le choix du format dépend de la complexité et de la nature des données du problème.

- **Le format LP (`.lp`) :** Ce format décrit le problème de manière explicite, équation par équation. Il est très lisible pour des instances de petite taille où les contraintes sont peu nombreuses et fixées. Cependant, il devient rapidement inutilisable lorsque la taille du problème augmente, car il ne permet pas de factoriser les expressions (par exemple, une somme $\sum$ doit être écrite terme par terme : $x_1 + x_2 + ... + x_n$).
- **Le format GMPL (`.mod` + `.dat`) :** Ce langage de modélisation algébrique permet de séparer la structure du modèle (les équations logiques) des données spécifiques à l'instance (les paramètres).
  - Le fichier `.mod` contient la déclaration des ensembles, des paramètres et des variables, ainsi que l'écriture générique des contraintes (utilisation de quantificateurs $\forall$ et de sommes $\sum$).
  - Le fichier `.dat` contient les valeurs numériques (matrices de coûts, capacités, etc.).

Dans ce TP, nous avons privilégié le format **GMPL** pour les exercices nécessitant une manipulation de données matricielles ou d'ensembles indicés (comme le problème d'affectation ou le e-commerce), car il offre une meilleure modularité et permet de résoudre différentes instances sans réécrire le modèle. Le format **LP** a pu être utilisé pour les premiers tests ou les problèmes très simples ne nécessitant pas l'utilisation de données externes.

2 Problème d'Assemblage

2.1 Modélisation Mathématique (PL / PLNE)

Modèle PL

Le modèle linéaire correspondant est donné ci-dessous, sous format LP compatible GLPK:

```
\* Problem: Assemblage de velos *\

Maximize
    Benefice: 650 VC + 250 VS

Subject To
    Surface : 3.5 VC + 1.5 VS <= 2000
    Quantite : VC <= 750
    Temps : 0.13 VC + 0.11 VS <= 40

End
```

Variables de décision

$$VC \geq 0, \quad VS \geq 0$$

Fonction objectif

$$\max\ 650\,VC + 250\,VS$$

Contraintes

$$3.5\,VC + 1.5\,VS \leq 2000 \quad \text{(Surface)}$$
$$VC \leq 750$$
$$0.13\,VC + 0.11\,VS \leq 40 \quad \text{(Temps)}$$

Résolution du modèle PL

La sortie du solveur GLPK (résumée ci-dessous) indique que la solution optimale est atteinte :

```
Problem:
Rows:       3
Columns:    2
```

```
Non-zeros:  5
Status:     OPTIMAL
Objective:  Benefice = 200000 (MAXimum)

Row name     Activity   Bound   Marginal
Surface        1076.92   <=2000
Quantite        307.69   <=750
Temps              40.0   <=40      5000

Column name  Activity   Lower bound  Upper bound  Marginal
VC              307.69       0
VS                0.00       0                     -300
```

Solution optimale

$$VC^\star = 307.692, \qquad VS^\star = 0$$

$$\text{Bénéfice maximal} = 200000$$

Analyse

- La contrainte de **temps** est active : $0.13 \times 307.692 = 40$, ce qui explique la borne saturée.
- Les contraintes de surface et de quantité ne sont pas limitantes.
- La variable VS a un coût réduit de -300, indiquant qu'en produire abaisserait le bénéfice : **il est optimal de ne produire que des vélos cargo**.

Modèle PLNE

Nous imposons maintenant que le nombre de vélos produits soit entier. Le modèle devient alors un programme linéaire en nombres entiers (PLNE).
Voici le code GLPK correspondant :

```
\* Problem: Assemblage de velos *\

Maximize
    Benefice: 650 VC + 250 VS

Subject To
    Surface : 3.5 VC + 1.5 VS <= 2000
    Quantite : VC <= 750
    Temps : 0.13 VC + 0.11 VS <= 40

Integer
    VC
    VS

End
```

Les variables VC et VS sont désormais contraintes à être entières :

$$VC \in N, \qquad VS \in N.$$

Résolution du modèle PLNE

La résolution donne :

```
Problem:
Rows:       3
Columns:    2 (2 integer, 0 binary)
Non-zeros:  5
Status:     INTEGER OPTIMAL
Objective:  Benefice = 199550 (MAXimum)

Row name       Activity   Bound
Surface          1074.5   <=2000
Quantite            307   <=750
Temps             39.91   <=40

Column name    Activity   Bound
VC                  307     integer
VS                    0     integer
```

Solution optimale entière

$$VC^\star = 307, \qquad VS^\star = 0$$

$$\text{Bénéfice maximal entier} = 199550$$

Analyse

- Le résultat est très proche du modèle continu (PL), qui donnait $VC = 307.692$. L'arrondi à l'entier inférieur est logique car VC doit être entier.
- La contrainte de **temps** reste la ressource limitante :
$$0.13 \times 307 = 39.91 \leq 40.$$
- Comme pour le PL, le solveur ne produit aucun vélo standard ($VS = 0$), car son coût marginal reste défavorable.
- Tous les tests d'optimalité sont satisfaits, garantissant la validité de la solution entière.

Comparaison PL vs PLNE

- PL : $VC = 307.692$, bénéfice = 200000
- PLNE : $VC = 307$, bénéfice = 199550

L'écart est très faible :

$$200000 - 199550 = 450$$

Ce résultat confirme que le passage à l'intégralité n'affecte presque pas l'optimalité du problème.

3 Problème d'Affectation

3.1 Modélisation Mathématique (PLNE)

Ce problème consiste à affecter exactement une tâche à chaque personne et exactement une personne à chaque tâche, tout en maximisant la somme des préférences individuelles.

Variables de décision

$$Q_{i,j} = \begin{cases} 1 & \text{si la personne } i \text{ est affectée à la tâche } j, \\ 0 & \text{sinon.} \end{cases}$$

Paramètres

$$preference_{i,j} : \text{score de préférence de la personne } i \text{ pour la tâche } j.$$

Modèle PLNE

Fonction objectif

$$\max \sum_{i \in PERSONNES} \sum_{j \in TACHES} preference_{i,j} \cdot Q_{i,j}$$

Contraintes Chaque tâche doit être attribuée à une personne :

$$\forall j \in TACHES, \quad \sum_{i \in PERSONNES} Q_{i,j} = 1$$

Chaque personne doit recevoir une tâche :

$$\forall i \in PERSONNES, \quad \sum_{j \in TACHES} Q_{i,j} = 1$$

Nature des variables

$$Q_{i,j} \in \{0, 1\}$$

Code GLPK du modèle

```
#  GLPK model file created by SUN for RO teaching

set PERSONNES;
set TACHES;

var Q{i in PERSONNES, j in TACHES}, binary;

param preference{i in PERSONNES, j in TACHES};

s.t. RespectPreferenceDeChaquePersonne{j in TACHES}:
  sum{i in PERSONNES} Q[i,j] = 1;

s.t. RespectPreferenceDeChaqueTache{i in PERSONNES}:
```

```
  sum{j in TACHES} Q[i,j] = 1;

maximize PreferenceTotal:
    sum{i in PERSONNES, j in TACHES} Q[i,j] * preference[i,j];

end;
```

Fichier de données

```
data;

set PERSONNES := P1 P2 P3;
set TACHES := T1 T2 T3;

param preference: T1 T2 T3 :=
P1 4 8 6
P2 7 1 3
P3 5 5 7;

end;
```

3.2 Implémentation et Résultats

Modèle LP généré

Le modèle GMPL est automatiquement transformé en LP par GLPK.

```
* Problem: ModelPreference *

Maximize
 PreferenceTotal: + 4 Q(P1,T1) + 7 Q(P2,T1) + 5 Q(P3,T1)
 + 8 Q(P1,T2) + Q(P2,T2) + 5 Q(P3,T2)
 + 6 Q(P1,T3) + 3 Q(P2,T3) + 7 Q(P3,T3)

Subject To
 RespectPreferenceDeChaquePersonne(T1): Q(P1,T1)+Q(P2,T1)+Q(P3,T1)=1
 RespectPreferenceDeChaquePersonne(T2): Q(P1,T2)+Q(P2,T2)+Q(P3,T2)=1
 RespectPreferenceDeChaquePersonne(T3): Q(P1,T3)+Q(P2,T3)+Q(P3,T3)=1
 RespectPreferenceDeChaqueTache(P1): Q(P1,T1)+Q(P1,T2)+Q(P1,T3)=1
 RespectPreferenceDeChaqueTache(P2): Q(P2,T1)+Q(P2,T2)+Q(P2,T3)=1
 RespectPreferenceDeChaqueTache(P3): Q(P3,T1)+Q(P3,T2)+Q(P3,T3)=1

Generals
 Q(P1,T1) Q(P2,T1) Q(P3,T1)
 Q(P1,T2) Q(P2,T2) Q(P3,T2)
 Q(P1,T3) Q(P2,T3) Q(P3,T3)

End
```

Résultat de la résolution

```
Status:     INTEGER OPTIMAL
Objective:  PreferenceTotal = 22 (MAXimum)

Row name                           Activity   Bound
RespectPreferenceDeChaquePersonne(T1)    1 = 1
RespectPreferenceDeChaquePersonne(T2)    1 = 1
RespectPreferenceDeChaquePersonne(T3)    1 = 1
RespectPreferenceDeChaqueTache(P1)       1 = 1
RespectPreferenceDeChaqueTache(P2)       1 = 1
RespectPreferenceDeChaqueTache(P3)       1 = 1

Column name      Activity
Q(P1,T1)         0
Q(P2,T1)         1
Q(P3,T1)         0
Q(P1,T2)         1
Q(P2,T2)         0
Q(P3,T2)         0
Q(P1,T3)         0
Q(P2,T3)         0
Q(P3,T3)         1
```

Solution optimale

$$P2 \to T1$$
$$P1 \to T2$$
$$P3 \to T3$$

$$\text{Préférence totale maximale} = 22$$

Analyse

- Toutes les contraintes d'assignation sont satisfaites : chaque tâche reçoit une personne et chaque personne reçoit une tâche.
- La solution est intuitive : chaque personne reçoit une tâche où elle possède l'un de ses meilleurs scores.

4 Applications E-commerce

4.1 Cas 1 : Affectation de commandes

Ce premier cas correspond à un problème logistique dans lequel plusieurs magasins doivent satisfaire la demande de différents clients (demandes), en fournissant des produits dits « fluides ». Chaque magasin possède un stock limité, chaque demande doit être satisfaite intégralement, et chaque unité expédiée entraîne un coût. L'objectif est de **minimiser le coût total** d'expédition.

4.1.1 Modélisation Mathématique (PL / PLNE)

Nous considérons trois ensembles :

$$MAGASINS, \quad DEMANDES, \quad FLUIDES.$$

Variable de décision

$$Choix_{i,j,k} \geq 0$$

Quantité du fluide j envoyée par le magasin i pour satisfaire la demande k.

Paramètres

$$\begin{aligned} demande_{k,j} &: \text{ quantité du fluide } j \text{ demandée par } k, \\ stock_{i,j} &: \text{ quantité en stock du fluide } j \text{ dans le magasin } i, \\ cout_{i,j} &: \text{ coût unitaire d'envoi du fluide } j \text{ par le magasin } i. \end{aligned}$$

Fonction objectif

$$\min \sum_{i \in MAGASINS} \sum_{j \in FLUIDES} \sum_{k \in DEMANDES} cout_{i,j} \cdot Choix_{i,j,k}$$

Contraintes **1. Limite de stock :**

$$\forall i \in MAGASINS,\ \forall j \in FLUIDES, \quad \sum_{k \in DEMANDES} Choix_{i,j,k} \leq stock_{i,j}$$

2. Satisfaction complète des commandes :

$$\forall j \in FLUIDES,\ \forall k \in DEMANDES, \quad \sum_{i \in MAGASINS} Choix_{i,j,k} = demande_{k,j}$$

Code GLPK du modèle (PL)

```
#  GLPK model file created by SUN for RO teaching

set MAGASINS;
set DEMANDES;
set FLUIDES;

var Choix{i in MAGASINS, j in FLUIDES, k in DEMANDES}, >=0;

param demande{i in DEMANDES, j in FLUIDES};
param stock{i in MAGASINS, j in FLUIDES};
param cout{i in MAGASINS, j in FLUIDES};

s.t. RespectDisponibilite{i in MAGASINS, j in FLUIDES}:
  sum{k in DEMANDES} Choix[i,j,k] <= stock[i,j];

s.t. RespectCommandeComplete{j in FLUIDES, k in DEMANDES}:
  sum{i in MAGASINS} Choix[i,j,k] = demande[k,j];
```

```
minimize CoutTotal:
  sum{i in MAGASINS, j in FLUIDES, k in DEMANDES}
      cout[i,j] * Choix[i,j,k];

end;
```

Solution GLPK (PL)

```
Problem:
Rows:       10
Columns:    12
Non-zeros:  24
Status:     OPTIMAL
Objective:  CoutTotal = 10 (MINimum)

RespectDisponibilite(M1,F1)  Activity 2.5  Bound <=2.5
RespectDisponibilite(M2,F2)  Activity 2    Bound <=2 (binding)

Choix(M1,F1,D1) = 1
Choix(M1,F1,D2) = 1.5
Choix(M2,F1,D2) = 1.5
Choix(M2,F2,D1) = 1
Choix(M2,F2,D2) = 1

All other variables = 0
```

4.1.2 Résultats comparatifs

Solution optimale

$$CoutTotal^{\star} = 10$$

Répartition optimale :

$$\begin{aligned}
&\text{Magasin M1 fournit F1 à D1 : 1}\\
&\text{Magasin M1 fournit F1 à D2 : 1.5}\\
&\text{Magasin M2 fournit F1 à D2 : 1.5}\\
&\text{Magasin M2 fournit F2 à D1 : 1}\\
&\text{Magasin M2 fournit F2 à D2 : 1}
\end{aligned}$$

Analyse

- La solution exploite fortement le magasin M1 pour F1 jusqu'à saturation de son stock.
- Le fluide F2 est fourni exclusivement par M2, car son coût est minimal.
- M3 n'intervient pas dans la solution optimale en raison de coûts moins avantageux.
- Toutes les contraintes sont satisfaites : stocks respectés, demandes entièrement honorées.

Cas 1.2 : Modèle PLNE

Dans cette seconde variante, nous imposons que les quantités expédiées soient des *entiers*. Cette contrainte est cohérente dans certains contextes logistiques où les fluides sont en réalité des colis indivisibles, ou lorsque l'unité choisie doit être comptée sans fraction.

La modélisation mathématique reste identique au cas précédent, à l'exception de la contrainte suivante :

$$Choix_{i,j,k} \in Z_{\geq 0}$$

Code GLPK du modèle (PLNE)

```
#  GLPK model file created by SUN for RO teaching

set MAGASINS;
set DEMANDES;
set FLUIDES;

var Choix{i in MAGASINS, j in FLUIDES, k in DEMANDES},
    integer, >=0;

param demande{i in DEMANDES, j in FLUIDES};
param stock{i in MAGASINS, j in FLUIDES};
param cout{i in MAGASINS, j in FLUIDES};

s.t. RespectDisponibilite{i in MAGASINS, j in FLUIDES}:
  sum{k in DEMANDES} Choix[i,j,k] <= stock[i,j];

s.t. RespectCommandeComplete{j in FLUIDES, k in DEMANDES}:
  sum{i in MAGASINS} Choix[i,j,k] = demande[k,j];

minimize CoutTotal:
  sum{i in MAGASINS, j in FLUIDES, k in DEMANDES}
      cout[i,j] * Choix[i,j,k];

end;
```

Solution GLPK (PLNE)

```
Status:     INTEGER OPTIMAL
Objective:  CoutTotal = 10 (MINimum)

Choix(M1,F1,D1) = 0
Choix(M1,F1,D2) = 2
Choix(M2,F1,D1) = 1
Choix(M2,F1,D2) = 1
Choix(M2,F2,D1) = 1
Choix(M2,F2,D2) = 1
```

Analyse du résultat

- Le coût optimal reste inchangé :

$$CoutTotal^{\star} = 10.$$

- La solution diffère légèrement de celle du PL, car les anciennes quantités fractionnaires (ex. 1.5) doivent désormais être arrondies.
- Les magasins M1 et M2 restent les plus utilisés :
  - M1 envoie uniquement du fluide F1 pour la demande D2.
  - M2 prend en charge tout le reste, notamment le fluide F2 où il est le moins coûteux.
- Le magasin M3 n'est toujours pas utilisé, les coûts étant moins compétitifs.
- Toutes les contraintes de demande et de stock sont respectées strictement.
- Le solveur signale une solution entière parfaitement admissible (conditions KKT satisfaites).

Comparaison PL vs PLNE.

- Les deux modèles donnent le **même coût optimal**.
- Le PL autorise des fractions (ex. 1.5 unités), ce qui est possible pour un fluide divisible.
- Le PLNE impose des quantités entières, modifiant légèrement la répartition optimale.
- L'écart entre solutions reste faible car les coûts favorisent fortement M1 et M2.

4.1.3 Intégration des coûts fixes et variables

Dans cette variante, nous introduisons des **coûts fixes** qui s'appliquent dès qu'un magasin traite une commande, en plus des **coûts variables** proportionnels aux quantités.

Pour modéliser cela correctement, il est impératif de lier la variable de décision binaire (activation du coût fixe) à la variable de flux (quantité transportée) via une contrainte de type *Big-M*.

Variables de décision

$$\begin{aligned} Choix_{i,j,k} &\geq 0 \quad && : \text{quantité du fluide } j \text{ envoyée du magasin } i \text{ vers la demande } k, \\ Q_{i,k} &\in \{0,1\} \quad && : \text{variable binaire indiquant si le magasin } i \text{ traite la demande } k. \end{aligned}$$

Paramètres

$$demande_{k,j} : \text{demande du fluide } j \text{ pour le client } k,$$

$$stock_{i,j} : \text{stock disponible du fluide } j \text{ dans le magasin } i,$$

$$coutfixe_{k,i} : \text{coût fixe si le magasin } i \text{ sert la demande } k,$$

$$coutvariable_{k,i} : \text{coût variable unitaire si } i \text{ sert } k.$$

$$BigM : \text{une grande constante, supérieure à la demande totale possible.}$$

Fonction objectif

$$\min \sum_{i \in MAGASINS} \sum_{k \in DEMANDES} \left(coutfixe_{k,i} \cdot Q_{i,k} + \sum_{j \in FLUIDES} coutvariable_{k,i} \cdot Choix_{i,j,k} \right)$$

Contraintes Les contraintes de stock et de satisfaction des commandes restent identiques au cas précédent :

$$\forall i, j, \quad \sum_k Choix_{i,j,k} \leq stock_{i,j},$$

$$\forall j, k, \quad \sum_i Choix_{i,j,k} = demande_{k,j}.$$

En plus de ces contraintes, nous ajoutons la **contrainte de liaison** :

$$\sum_{j \in FLUIDES} Choix_{i,j,k} \leq M \cdot Q_{i,k} \quad \forall i, k$$

Code GLPK du modèle (PL, coûts fixes + variables)

```
#  GLPK model file created by SUN for RO teaching

set MAGASINS;
set DEMANDES;
set FLUIDES;

var Choix{i in MAGASINS, j in FLUIDES, k in DEMANDES}, >=0;
var Q{i in MAGASINS, k in DEMANDES}, binary;

param demande{i in DEMANDES, j in FLUIDES};
param stock{i in MAGASINS, j in FLUIDES};
param coutfixe{i in DEMANDES, j in MAGASINS};
param coutvariable{i in DEMANDES, j in MAGASINS};
param BigM := sum{k in DEMANDES, j in FLUIDES} demande[k,j];

s.t. RespectDisponibilite{i in MAGASINS, j in FLUIDES}:
  sum{k in DEMANDES} Choix[i,j,k] <= stock[i,j];

s.t. RespectCommandeComplete{j in FLUIDES, k in DEMANDES}:
  sum{i in MAGASINS} Choix[i,j,k] = demande[k,j];
```

```
s.t. RelationFixeVariable {i in MAGASINS, k in DEMANDES}:
  sum{j in FLUIDES} Choix[i,j,k] <= BigM * Q[i,k];

minimize CoutTotal:
  sum{i in MAGASINS, k in DEMANDES}
    ( coutfixe[k,i] * Q[i,k]
    + sum{j in FLUIDES} coutvariable[k,i] * Choix[i,j,k] );

end;
```

Données utilisées

```
data;

set MAGASINS := M1 M2 M3;
set DEMANDES := D1 D2;
set FLUIDES  := F1 F2;

param demande: F1 F2 :=
D1 1 1
D2 3 1;

param stock: F1 F2 :=
M1 2.5 1
M2 3   2
M3 1   3;

param coutfixe: M1 M2 M3 :=
D1 100 120 110
D2 110  90 100;

param coutvariable: M1 M2 M3 :=
D1 12  7  1
D2  4 15 20;

end;
```

4.1.4 Résultats comparatifs

Le modèle a été résolu avec les données fournies. Le résultat est identique que l'on déclare les variables *Choix* comme continues (PL) ou entières (PLNE) en raison de la nature des données.

```
Status:     INTEGER OPTIMAL
Objective:  CoutTotal = 262 (MINimum)

Choix(M2,F1,D2) = 3
Choix(M2,F2,D2) = 1
Choix(M3,F1,D1) = 1
```

```
Choix(M3,F2,D1) = 1

Q(M2,D2) = 1
Q(M3,D1) = 1
```

Analyse

- **Coût total :** 262. Ce coût est nettement supérieur au cas sans coûts fixes, car l'activation des trajets est désormais pénalisante.
- **Stratégie logistique :**
  - Le magasin **M2** est activé ($Q = 1$) pour livrer l'intégralité de la commande du client **D2** (coût fixe de 90, le moins cher pour ce client).
  - Le magasin **M3** est activé ($Q = 1$) pour livrer l'intégralité de la commande du client **D1**.
  - Le magasin **M1** n'est pas utilisé (flux nuls, $Q = 0$), car ses coûts fixes ou variables n'étaient pas compétitifs.
- **Intégrité de la solution :** Les demandes et les stocks étant des nombres entiers, la structure du problème de transport conduit naturellement à des flux entiers. Par conséquent, la résolution en variables continues ou entières aboutit exactement au même résultat (262).

4.2 Cas 2 : Tournée de livraison

4.2.1 Modélisation du cycle

	ALPHA	**C1**	**C2**	**C3**	**C4**	**C5**
ALPHA	-	1	2	10	12	15
C1	1	-	1	8	10	11
C2	2	1	-	8	13	10
C3	10	8	8	-	1	1
C4	12	10	13	1	-	2
C5	15	11	10	1	2	-

Table 1: Matrice des distances entre le magasin ALPHA et les cinq clients.

Ce problème correspond au cas classique du **Voyageur de Commerce**. L'objectif est de définir une tournée unique partant du dépôt (ALPHA), visitant chaque client exactement une fois, et revenant au point de départ en minimisant la distance totale.

Variables de décision Nous utilisons deux types de variables pour modéliser ce problème :

$$Trajet_{i,j} \in \{0, 1\} \quad : \text{ 1 si le véhicule va directement de } i \text{ vers } j$$
$$u_i \in Z_{\geq 1} \quad : \text{ ordre de visite du client } i \text{ dans la tournée}$$

Fonction objectif

$$\min \sum_{i,j} distance_{i,j} \cdot Trajet_{i,j}$$

Contraintes Outre les contraintes de flux classiques (une entrée et une sortie par nœud), nous ajoutons les contraintes d'élimination:

$$\sum_{j \neq i} Trajet_{i,j} = 1 \quad \forall i \qquad \text{(Sortie unique)}$$

$$\sum_{i \neq j} Trajet_{i,j} = 1 \quad \forall j \qquad \text{(Entrée unique)}$$

$$u_i - u_j + n \cdot Trajet_{i,j} \leq n - 1 \quad \forall i, j \neq \text{ALPHA}$$

Code GLPK du modèle (TSP)

```
#  GLPK model file created by SUN for RO teaching

set POINTS;
param n := card(POINTS);
param distance{i in POINTS, j in POINTS};

var Trajet{i in POINTS, j in POINTS : i != j}, binary;
var u{POINTS} >= 1, <= n;

# Contraintes de flux (Entree/Sortie unique)
s.t. RespectEntreeUnique{i in POINTS}:
  sum{j in POINTS : i!=j} Trajet[i,j] = 1;

s.t. RespectSortieUnique{j in POINTS}:
  sum{i in POINTS : i!=j} Trajet[i,j] = 1;

# Elimination des sous-tours (MTZ)
s.t. RespectParcoursTotalite {i in POINTS, j in POINTS :
    i != 'ALPHA' and j != 'ALPHA' and i != j}:
    u[i] - u[j] + n * Trajet[i,j] <= n - 1;

minimize CoutTotal:
    sum{i in POINTS, j in POINTS : i!=j} distance[i,j]*Trajet[i,j];
end;
```

Données utilisées

```
data;

set POINTS :=
ALPHA
C1
C2
```

```
C3
C4
C5;

param distance: ALPHA C1 C2 C3 C4 C5 :=
ALPHA 0 1 2 10 12 15
C1 1 0 1 8 10 11
C2 2 1 0 8 13 10
C3 10 8 8 0 1 1
C4 12 10 13 1 0 2
C5 15 11 10 1 2 0;

end;
```

Résultats (GLPK)

```
Status:     INTEGER OPTIMAL
Objective:  CoutTotal = 24 (MINimum)

Trajet(ALPHA,C1) = 1    u(C1) = 1
Trajet(C1,C3)    = 1    u(C3) = 2
Trajet(C3,C4)    = 1    u(C4) = 3
Trajet(C4,C5)    = 1    u(C5) = 4
Trajet(C5,C2)    = 1    u(C2) = 5
Trajet(C2,ALPHA) = 1
```

Analyse

- Le coût optimal est :
$$CoutTotal^{\star} = 24.$$
- La tournée reconstruite est : **ALPHA → C1 → C3 → C4 → C5 → C2 → ALPHA**.
- **Validation technique :** Les variables auxiliaires u suivent une suite croissante stricte $(1, 2, 3, 4, 5)$ le long du parcours C1...C2. Cela confirme que la contrainte d'élimination des sous-tours a fonctionné correctement (il n'y a pas de cycles isolés).
- **Cohérence et justification :**
  - On observe l'utilisation de l'arc $C5 \to C2$ qui a un coût élevé de **10**.
  - Ce choix, bien que localement coûteux, est globalement optimal. Une approche gloutonne (aller au plus proche) aurait mené de C1 vers C2 (coût 1), mais aurait ensuite bloqué le livreur dans une zone où tous les coûts de sortie sont très élevés (vers C3, C4 ou C5).
  - Le solveur a donc "sacrifié" un trajet coûteux pour bénéficier de la série très économique $C1 \to C3 \to C4 \to C5$ (coûts 8 + 1 + 2) et du retour $C2 \to ALPHA$ (coût 2).

5 Conclusion

5.1 Synthèse des résultats

Ce TP avait pour objectif de modéliser et résoudre différents problèmes d'optimisation à l'aide de GLPK, en comparant les formulations PL et PLNE. Les travaux réalisés montrent clairement l'importance du choix des variables, des contraintes, et de la structure du modèle.

Assemblage. Les modèles PL et PLNE conduisent à des solutions très proches : seule la contrainte d'intégralité réduit légèrement le bénéfice optimal. Le PL fournit une bonne approximation, et la ressource limitante identifiée est le temps de production.

Affectation. Le modèle PLNE d'affectation permet d'obtenir une solution optimale intuitive maximisant la somme des préférences. Toutes les contraintes d'unicité sont respectées et la structure du problème se prête très bien à une formulation binaire.

Cas e-commerce. Les modèles successifs montrent clairement l'impact des hypothèses:

- le PL permet une répartition fine des flux,
- le PLNE impose des arrondis qui modifient légèrement la solution,
- l'ajout de coûts fixes incite à concentrer les flux sur un nombre réduit de magasins pour éviter de multiplier les frais d'activation.

Tournée de livraison. Ce cas met en évidence la capacité du solveur à traiter un problème combinatoire complexe. La méthode utilisée permet d'éliminer les sous-tours, et la tournée optimale obtenue minimise la distance totale tout en respectant les contraintes.

Au final, ce TP met en lumière :

- la pertinence des outils de modélisation (PL/PLNE),
- l'importance de formuler correctement les variables et contraintes,
- la capacité de GLPK à résoudre efficacement des problèmes variés,
- le rôle central de la modélisation dans la recherche opérationnelle.