# *SOMMAIRE*

**Introduction**

1. **Contexte du problème de recherche opérationnelle**
2. **Programme linéaire**
3. **Résolution du programme linéaire**
4. **Bilan et comparaison**
5. **difficultés et perspectives**

**Conclusion**

# Introduction

Dans le cadre de notre projet de recherche opérationnelle, nous avons exploré divers concepts et projets afin d'analyser et d'optimiser des problématiques concrètes liées aux opérations industrielles et logistiques. Parmi les nombreuses idées examinées, notre attention s'est particulièrement portée sur les opérations de terrassement, un domaine crucial dans les projets d'infrastructures de grande envergure.

L'idée directrice de notre étude a été inspirée par la construction du barrage de Nachtigal, un projet phare réalisé par la Camerounaise de Construction du Barrage de Nachtigal (CCN). Ce chantier nous a offert une base de données riche et pertinente, que nous avons analysée dans une perspective d'adaptabilité et d'application à d'autres projets similaires.

Notre objectif est d’identifier des stratégies efficaces pour optimiser les opérations de terrassement, en tenant compte des contraintes logistiques, des ressources disponibles et des coûts associés. Ce travail s’inscrit dans une démarche visant à généraliser les bonnes pratiques identifiées, afin qu’elles puissent être implémentées sur d'autres projets potentiels, dans des contextes similaires notamment celui de la construction du **barrage de Kikot**, est en préparation.

Ainsi, ce rapport présente les démarches méthodologiques adoptées, les analyses effectuées, ainsi que les résultats et recommandations découlant de cette étude.

# Problème de recherche opérationnelle

L’entreprise CCN a réalisé une opération majeure lors des travaux de terrassement pour la construction du barrage de Nachtigal, qui ont nécessité le déplacement de 2 millions de mètres cubes de terre pendant une durée de 20 mois. Pour accomplir cette tâche, 22 engins de terrassement de caractéristiques différentes étaient disponibles et pouvaient être mobilisés. Chaque engin était associé à une benne de 24 m³ de volume.

Les informations sur les engins disponibles sont présentées dans les tableaux n°1 et n°2 ci-dessus. Les données du tableau n°1 ont été directement fournies par l’entreprise, tandis que celles du tableau n°2 ont été obtenues à la suite de calculs nécessaires à la modélisation.

L'objectif principal de cette étude est de minimiser le nombre d'engins à utiliser tout en respectant plusieurs contraintes :

* Le volume total de terre à déplacer,
* La durée maximale du projet,
* La capacité de dépassement,
* Le nombre maximal d’engins pouvant fonctionner simultanément,
* La durée maximale d’utilisation pour chaque type d’engin.

En outre, nous proposerons un calendrier optimal pour l’utilisation des engins.

**Paramètres du Projet**:

**Volume total à déplacer** : V = 2 000 000 m³

**Durée de l’opération** : 20 mois (400 jours effectifs)

**Temps de travail effectif par jour et par engin** : basé sur 8 heures de travail par jour, avec 1 heure de pause par jour et le temps d’attente d’un engin entre les versements (entre 8-15 minutes = )

**Fréquence de travail des engins** : 2 tours par minute

**Nombre maximal d’engins simultanés sur le site** : 12 engins



Table 1 : Liste des engins disponibles avec leurs caractéristiques.



Tableau n° 2

Ce tableau est obtenu en insérant les différentes formules suivantes dans une feuille Excel pour faciliter les conversions :

* =

Dans le tableau 2, , choisi initialement comme temps moyen.

# Programme linéaire

1. **Variables de décision**

* : Variable binaire indiquant si l’engin *i* est utilisé pendant le mois *t*.

=

* : Variable binaire indiquant si l’engin *i* est utilisé pendant au moins un mois.

=

## **2. Fonction Objectif**

L’objectif est de minimiser le nombre total d’engins utilisés au cours des 20 mois. La fonction objective peut être exprimée comme suit :

Minimiser Z =

## **3. Contraintes**

### **3.1 Contrainte de Volume Total**

Cette contrainte garantit que le volume total déplacé par les engins au cours des 20 mois atteint au moins 2 000 000 m³. Le volume déplacé par un engin « i » pendant un mois « t » est calculé selon l’équation suivante :

Volume , pour un i et t donnée

Où :  
- C est la capacité de godet de l’engin (en m³),  
Le volume total déplacé par l’engin « i » pendant les 20 mois est exprimé par :

Volume =

La contrainte s’écrit donc :

≥ 2 000 000

### **3.2 Contrainte du Nombre Maximum d’Engins Simultanés**

Pour chaque mois « t », au plus 12 engins peuvent être utilisés simultanément. Cette contrainte est exprimée comme suit :

Pour tout t=1 à 20

### **3.3 Contrainte de Durée Maximale d’Utilisation**

Chaque engin « i » dispose d’une durée maximale d’utilisation « Mi » (en mois). Cette contrainte est donnée par :

Pour tout i=1 à 22

### **3.4 Contrainte de Continuité d’Utilisation**

Si un engin « i » est utilisé pendant le mois « t » et un mois « k > t », alors il doit être utilisé sans interruption entre ces deux périodes. La contrainte s’écrit :

, pour tout i ∈ {1, ..., 22} et tout t ∈ {1, ..., 19}

### **3.5 Contrainte de Liaison entre Variables**

Pour garantir que yi prend la valeur 1 dès que l’engin « i » est utilisé, on impose :

Et pour s’assurer que = 1 si = 1 pour au moins un mois « t », on ajoute :

, pour t=1 à 20, où M = 20 (nombre maximal de mois)

### **3.6 Contraintes de Non-négativité et Binarité**

Les variables sont binaires et respectent les conditions suivantes :  
- ∈ {0, 1} pour tout i et tout t,  
- ∈ {0, 1} pour tout i.

## **4. Conclusion partielle**

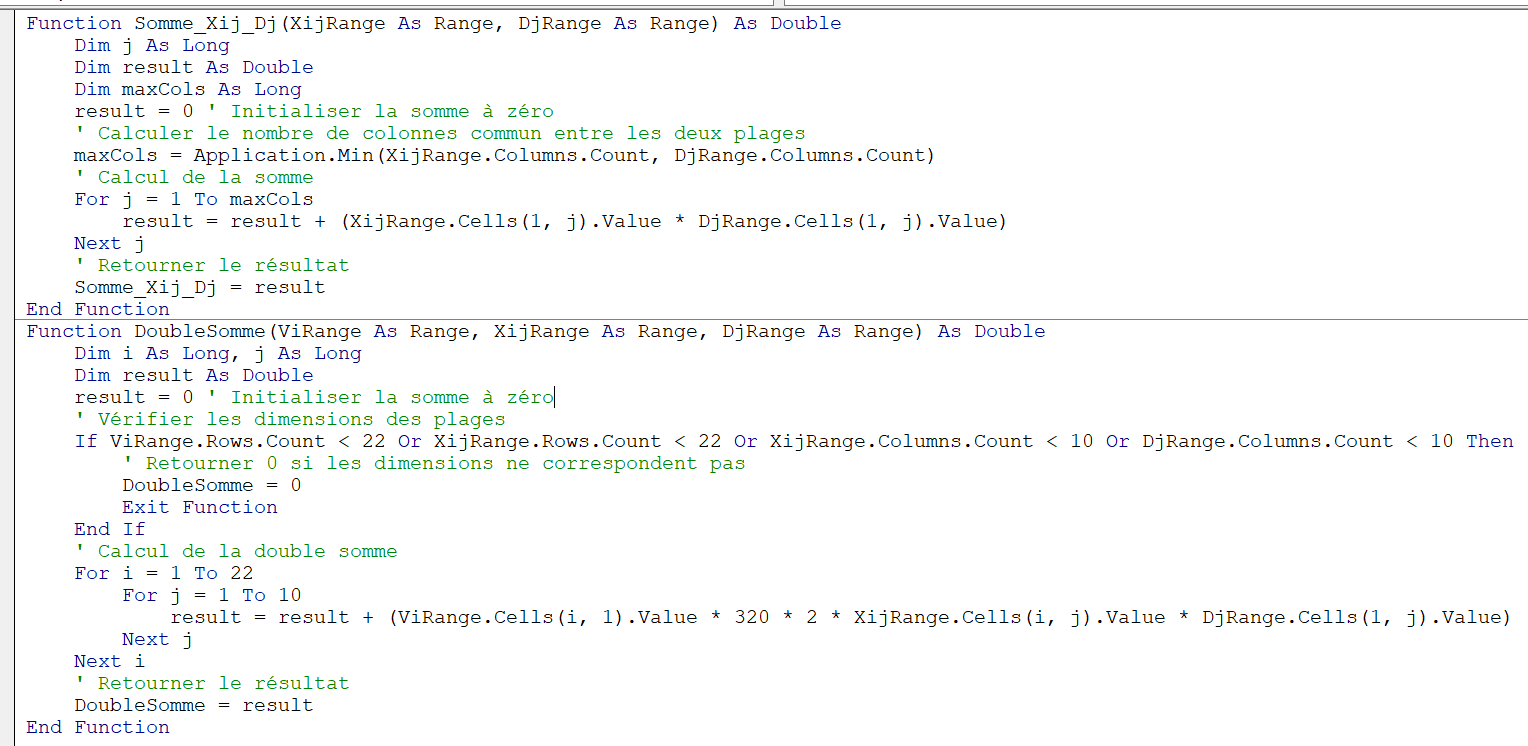
Ce modèle permet d’optimiser l’utilisation des engins en minimisant le nombre total d’engins nécessaires tout en respectant les contraintes opérationnelles : volume déplacé, simultanéité et durée d’utilisation. La solution indiquera les mois précis d’utilisation pour chaque engin, permettant une gestion efficace et économique des ressources.

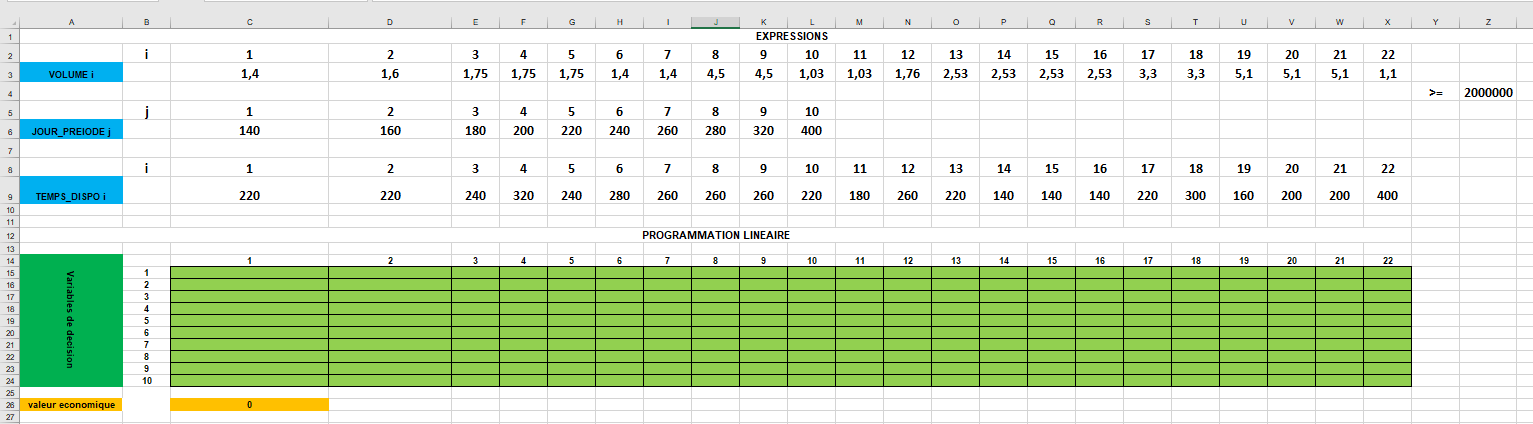
;

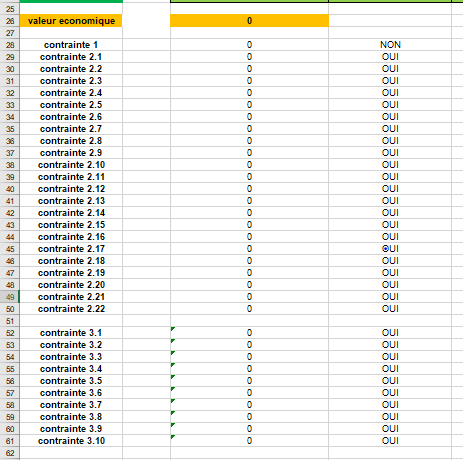
## Résolution du programme linéaire

1. **Résolution du problème linéaire à l’aide du Solveur Excel**

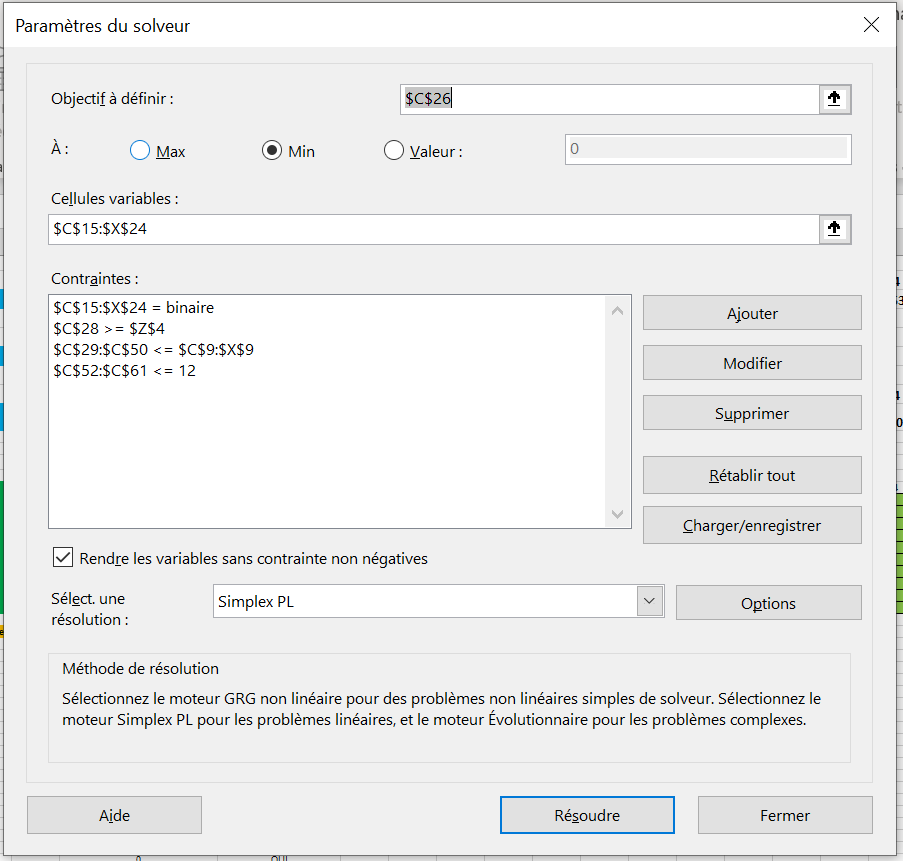
Pour faciliter les calculs des doubles sommes à insérer dans les cellules, nous avons créé des fonctions sur Excel avec VBA :

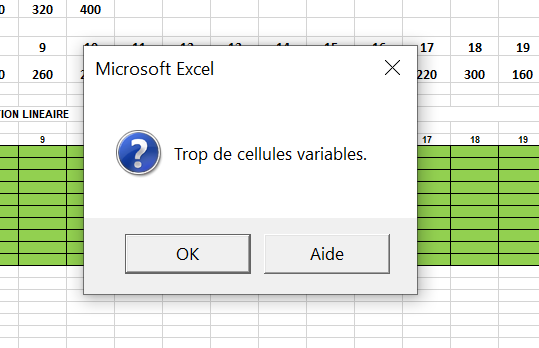


****

****

Après avoir saisi les données dans Excel, nous utilisons le Solveur pour paramétrer notre problème linéaire. Cela consiste à définir la fonction objective, spécifier les variables décisionnelles, et intégrer les différentes contraintes dans l'interface du Solveur. Ce processus permet de structurer et résoudre efficacement le programme linéaire en utilisant les outils intégrés d’Excel.

****

****

Lors de la résolution de notre problème linéaire à l'aide du solveur Excel, nous avons été confrontés à un nombre important de variables décisionnelles, ce qui a rendu le processus complexe et fastidieux. En effet, la limitation d'Excel dans la gestion simultanée de nombreuses variables et contraintes a impacté l'efficacité et la lisibilité des résultats, nécessitant des ajustements constants pour respecter les limites de calcul et structurer correctement les données. Afin de contourner ces limitations et d'automatiser davantage le processus, nous avons choisi d'utiliser Python avec les bibliothèques **scipy.optimize** et **numpy**, qui offrent des outils performants et adaptés pour gérer efficacement des problèmes linéaires de grande taille tout en facilitant les manipulations mathématiques et algébriques.

1. **Résolution du problème linéaire à l’aide de la bibliothèque PuLP de Python**

**Fonctionnalité et utilisation de la bibliothèque Pulp en programmation linéaire :**

Pulp est une bibliothèque open source pour la programmation linéaire en Python. Elle permet de modéliser et de résoudre des problèmes d’optimisation linéaire en utilisant une interface python intuitive. Développé par ANDREW M. Mc GOWAN en 2008, son but était de rendre l’optimisation linéaire accessible et facile à utiliser pour les développeurs et les chercheurs.

Elle met à disposition tous les outils pour modéliser un problème et les résoudre en faisant appel à différents types de solveurs standards utilisant des algorithmes de résolution différents. C'est en fait un Wrapper qui permet la formulation du problème en Python.

**UTILISATION DE LA BIBLIOTHEQUE PULP POUR LA RESOLUTION DUN PROBLEME LINEAIRE :**

Prenons un exemple simple de problème de maximisation de profit :   
Un fabricant de voitures propose 2 modèles à la vente, des grosses voitures et des petites voitures. Les grosses voitures sont vendues à 16000 € unité tandis que les petites voitures à 10000 € unité.  
Son problème vient de l’approvisionnement limité en deux matières premières, le  
caoutchouc et l’acier. La construction d’une petite voiture nécessite l’emploi d’une unité de caoutchouc et d’une unité d’acier, tandis que celle d’une grosse voiture nécessite une unité de caoutchouc mais deux unités d’acier.  
Dans son stock, il ne lui reste plus que 400 unités de caoutchouc et 600 unités d’acier.

Il aimerait savoir combien de petites et de grosses voitures il doit produire afin de maximiser son chiffre d'affaires en prenant en compte son stock actuel.

Nous appellerons **x** le nombre de grosses voitures produites, **y** le nombre de petites voitures produites, et **z** le chiffre d'affaires résultant. Le problème se traduit alors sous la forme:

Maximiser z=16000x+10000y

s.t.  x+y≤400

2x+y≤600

x≥0, y≥0

Explication :  
Nous souhaitons maximiser le chiffre d’affaires z qui est égal à 16000 \* le nombre de grosses voitures vendues + 10000 \* le nombre de petites voitures vendues.  
Les contraintes sont telles que :

* Le nombre d’unités de caoutchouc utilisées (x + y car il faut 1 unité de caoutchouc pour chaque voiture) ne peut excéder 400
* Le nombre d’unités d’acier utilisées (2x + y car il faut 2 unités d’acier pour chaque grosse voiture et 1 unité pour chaque petite voiture) ne peut excéder 600
* Le nombre de petites/grosses voitures produites est positif.

Nos variables de décisions sont x et y. Elles doivent être des valeurs entières positives.

# Nombre de grosses voitures produites

x = LpVariable('x', lowBound=0, cat=LpInteger)

# Nombre de petites voitures produites

y = LpVariable('y', lowBound=0, cat=LpInteger)

Initialisons le problème pour pouvoir lui ajouter nos contraintes et fonction objectif. Nous voulons maximiser notre fonction objective donc nous utilisons LpMaximize (LpMinimize si l’on veut minimiser).

probleme = LpProblem(name='chiffre\_affaires', sense=LpMaximize)

Ajoutons à notre problème les contraintes par rapport au stock :

* e est l’expression à comparer et contient généralement les variables de décision que nous avons défini plus haut
* sense est le sens de l’inégalité avec LpConstraintLE pour “less equal” et LpConstraintGE pour “greater equal”
* Le nom de la contrainte peut servir pour débugger le problème, si nous souhaitons afficher les contraintes non utilisées par exemple
* rhs (right hand-side) est la constante à laquelle l’expression est comparée

contrainte\_caoutchouc = LpConstraint(e=x + y, sense=LpConstraintLE, name='contrainte\_caoutchouc', rhs=n\_caoutchouc)

probleme.add(contrainte\_caoutchouc)

contrainte\_acier = LpConstraint(e=2 \* x + y, sense=LpConstraintLE, name='contrainte\_acier', rhs=n\_acier)

probleme.add(contrainte\_acier)

Nous pouvons aussi les ajouter de la manière suivante car PuLP reconnaît le type d’expression passée (contrainte vs fonction objectif).

probleme += (x+y <= n\_caoutchouc)

probleme += (2\*x+y <= n\_acier)

Ajoutons ensuite la fonction objective à maximiser au problème.

probleme += prix\_grosse\_voiture\*x + prix\_petite\_voiture\*y

Une fois que le problème a été résolu, nous pouvons afficher les valeurs trouvées.

print(f'x = {x.value()}')

print(f'y = {y.value()}')

Ce qui nous donne le résultat suivant :

x = 200.0

y = 200.0

Le fabricant devra donc fabriquer 200 petites et 200 grosses voitures pour maximiser son chiffre d’affaires.

**Fonctionnalité et utilisation de la bibliothèque SCIPY en programmation linéaire :**

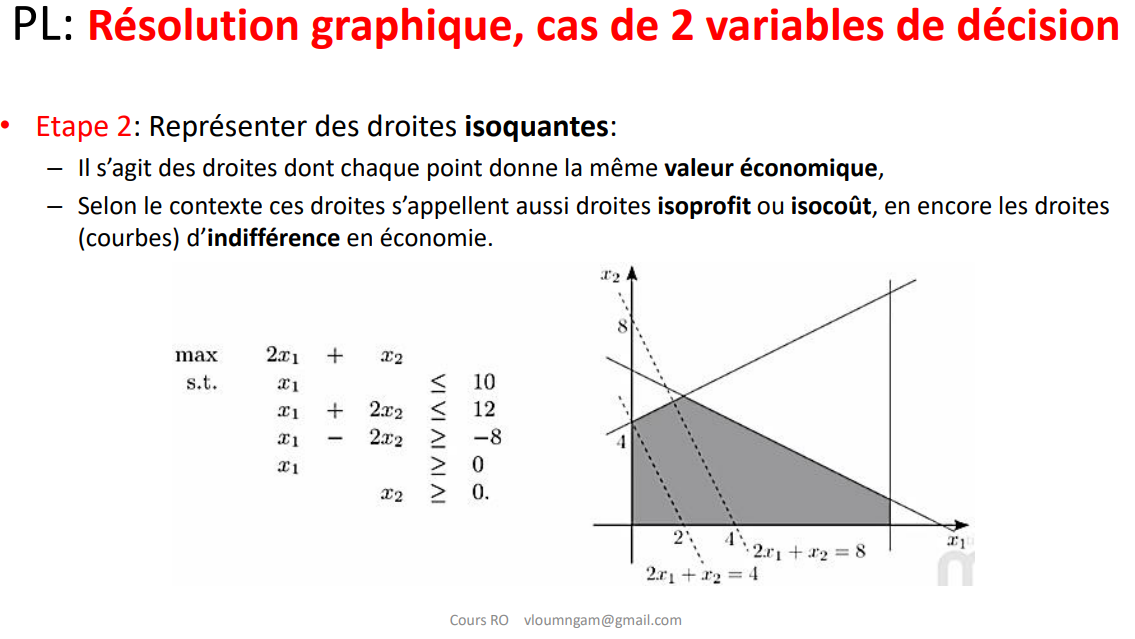
La bibliothèque SCIPY est un outil puissant pour la programmation scientifique en python, incluant des fonctionnalités spécifiques pour la programmation linéaire. SCIPY a été fonde a la fin des années 1990 par TARVIS OLLIPHANT un des principaux contributeurs de NUMPY. Le projet a été motivé par le besoin d’une bibliothèque plus riche en fonctionnalités pour le calcul scientifique.

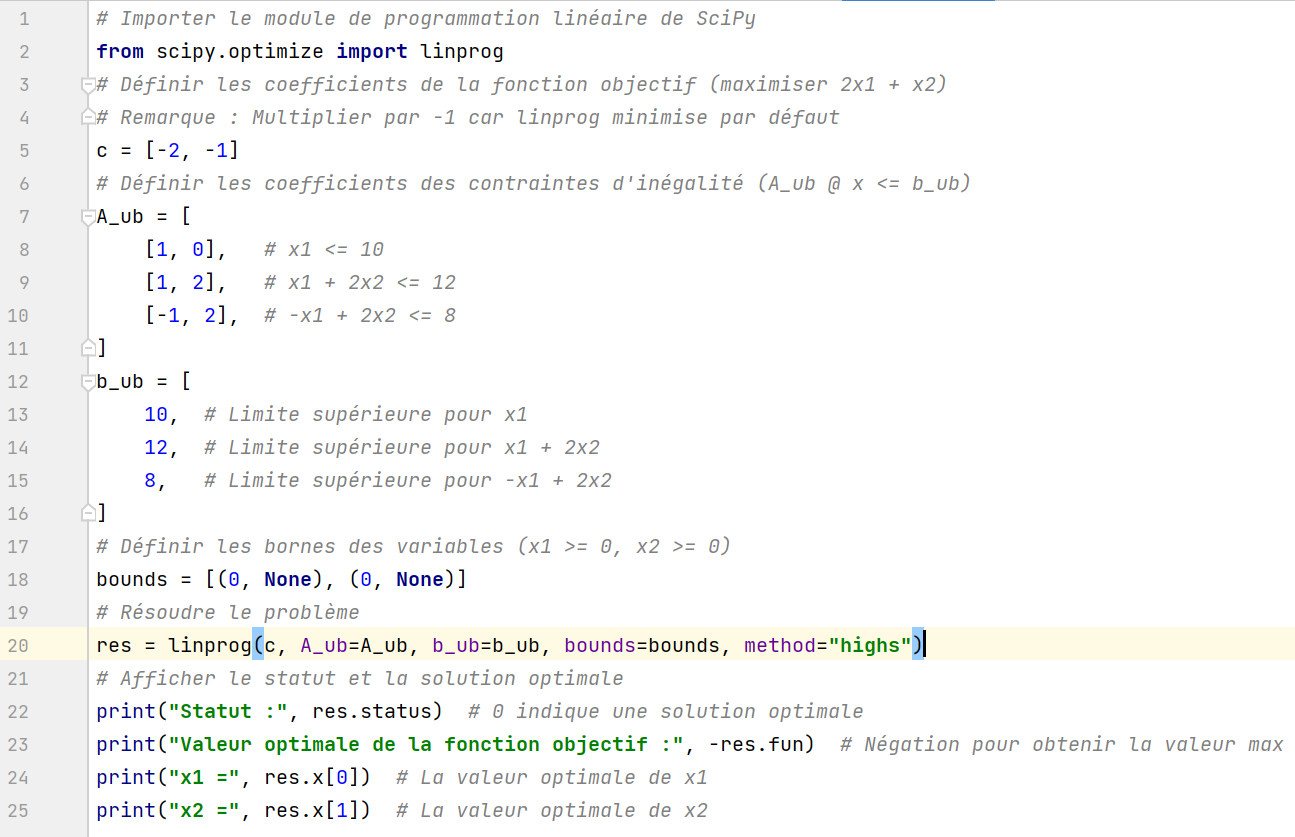
**UTILISATION DE LA BIBLIOTHEQUE SCIPY POUR LA RESOLUTION DUN PROBLEME LINEAIRE :**

Voici un aperçu de ses principales fonctionnalités et de son utilisation dans un problème d’optimisation.

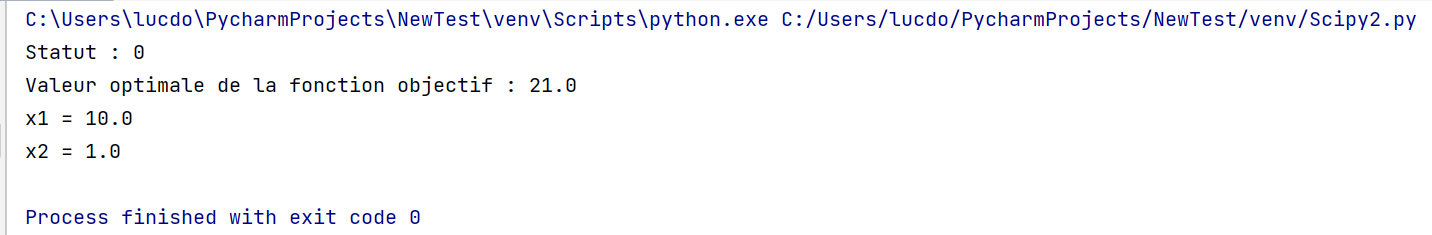
**EXEMPLES DU COUR :**

**FONCTION 1 : CAS D’UN PL VIABLE :**

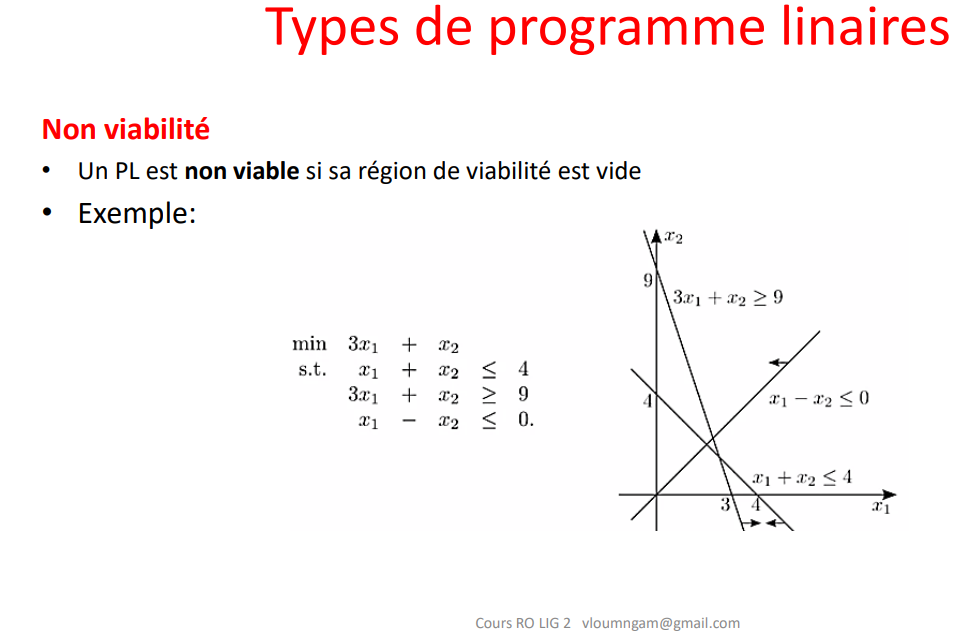
**RESOLUTION :**

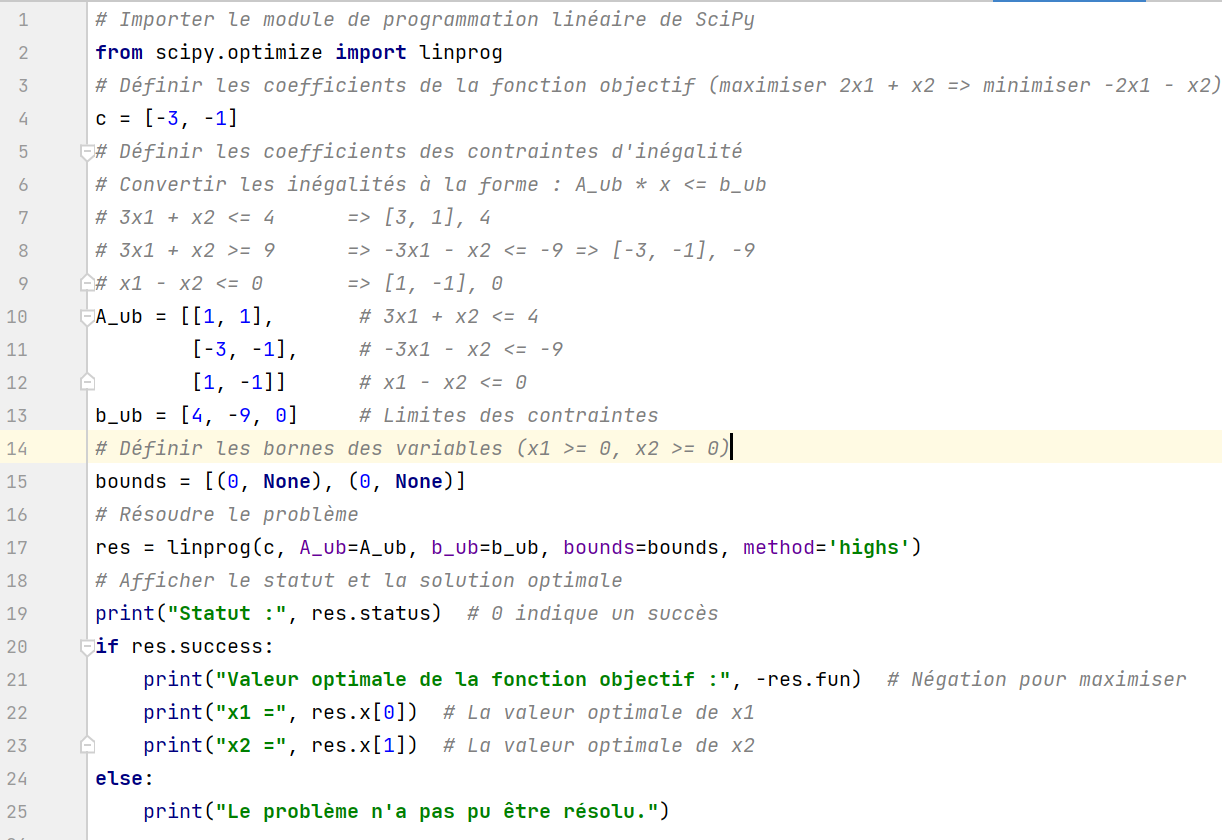
****

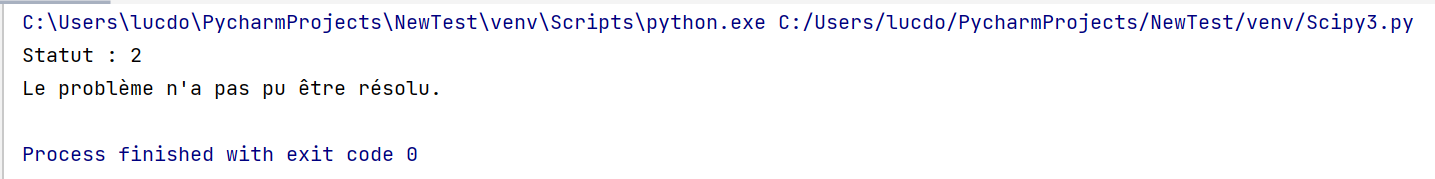
**SOLUTION :**

****

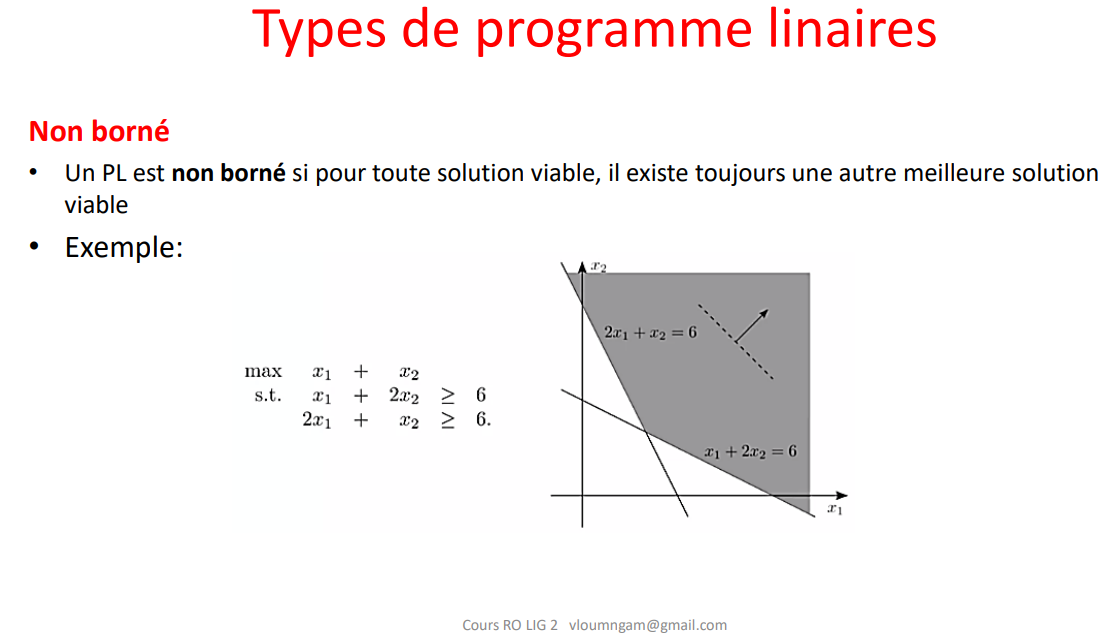
**FONCTION 2 : CAS D’UN PL NON VIABLE :**



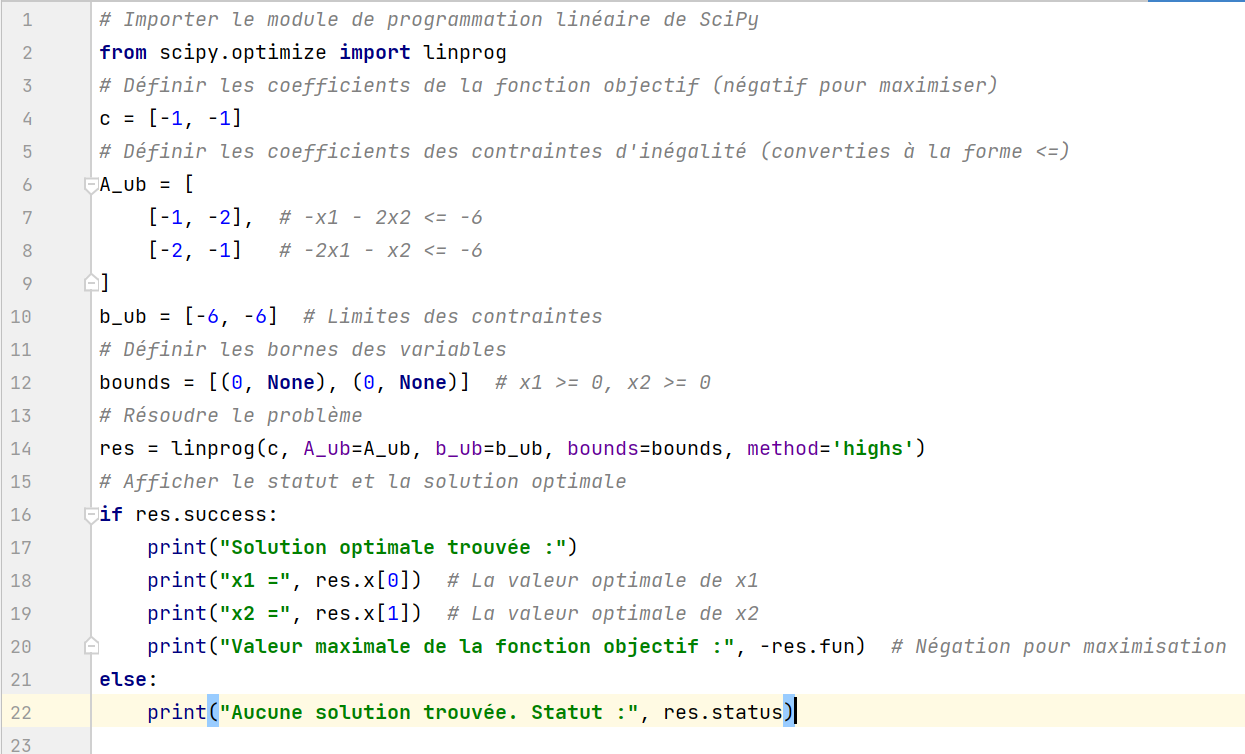
**RESOLUTION :**

**SOLUTION**

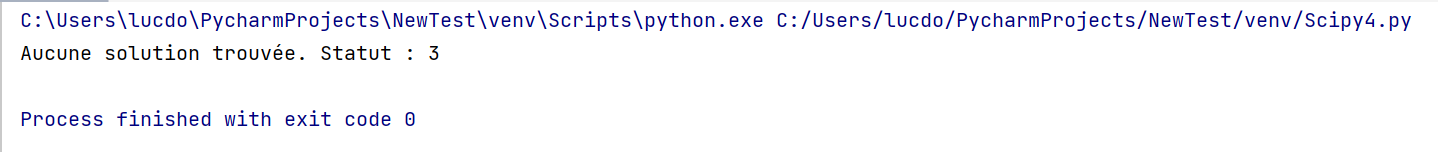
**FONCTION 3 : CAS D’UN PL NON BORNE**



**RESOLUTION :**



**SOLUTION :**

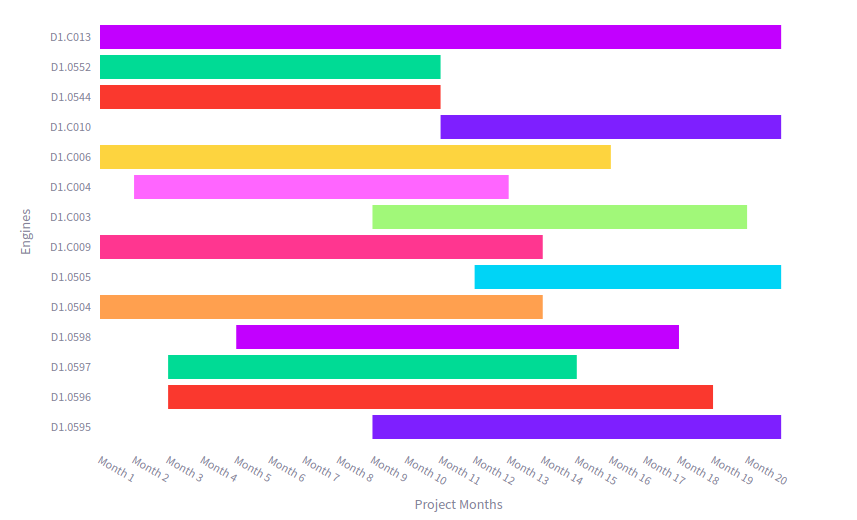
****

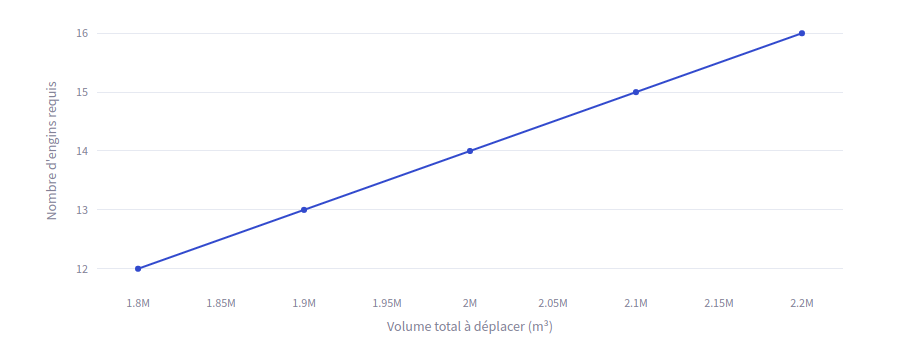
En somme, PULP est idéale pour les problèmes d’optimisation linéaire, tant dis que SCIPY est plus polyvalent offrant des outils pour l’optimisation, l’intégration et l’analyse statistique.

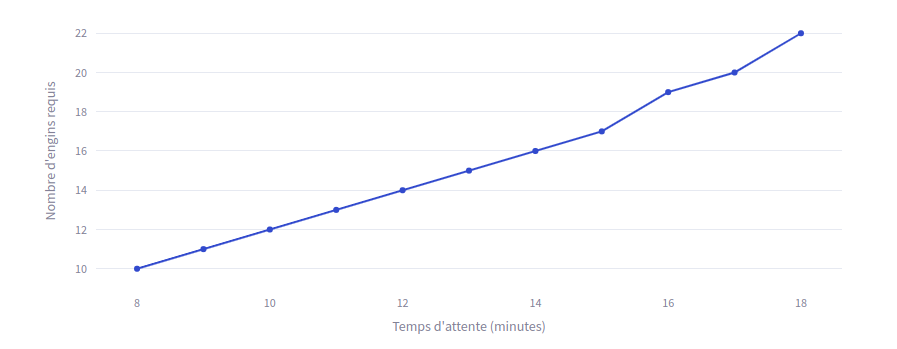
Choisissez PULP pour des problèmes spécifiques d’optimisation linéaire et SCIPY pour des problèmes plus généraux d’optimisation et d’analyse.

**Résolution du programme linéaire proprement dit**

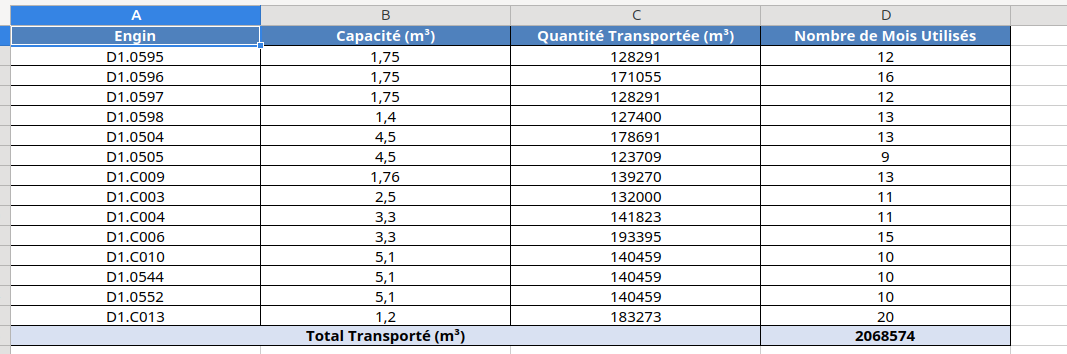


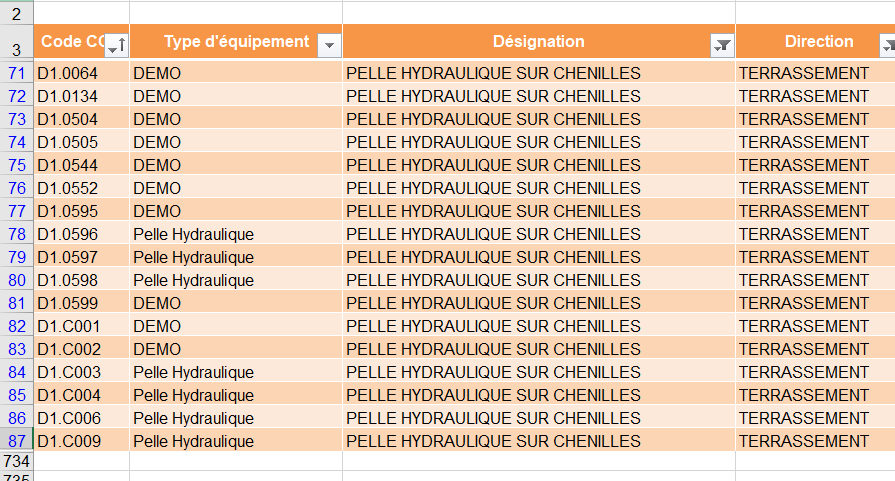






1. Bilan et comparaison





Nous pouvons constater que notre résolution apporte une amélioration significative par rapport à la solution initiale utilisée lors des travaux du barrage de Nachtigal. En effet, notre modèle propose une utilisation optimisée de 14 engins pour accomplir le même volume de terrassement sur la même durée, alors que 17 engins avaient été mobilisés lors de l’opération initiale.

Cette réduction de trois engins traduit une gestion plus rationnelle des ressources, permettant non seulement de réduire les coûts associés à l’utilisation des engins supplémentaires, mais également d’optimiser la logistique globale du chantier. Elle met en évidence la pertinence de notre approche et la valeur ajoutée de l’application des principes de la recherche opérationnelle à ce type de projet.

# difficultés et perspectives

## **Difficultés**

L'une des principales difficultés rencontrées lors de ce projet a été l'analyse du monde réel afin d'extraire des données exploitables pour la modélisation. La collecte et la structuration des informations pertinentes nécessitaient une compréhension fine des opérations de terrassement, ainsi qu’une collaboration étroite avec les experts du domaine. Ce défi a parfois ralenti notre progression et exigé des ajustements méthodologiques pour garantir la fiabilité des données utilisées.

## **Perspectives**

À la suite de la soumission de notre travail, l’ingénieur en génie civil impliqué dans le projet nous a proposé plusieurs pistes d’amélioration pour accroître la pertinence et l’adaptabilité de notre approche pour les futures opérations de terrassement. Ces perspectives incluent :

1. **Étude des vitesses des engins** : Analyser les vitesses des différents engins afin de mieux estimer le temps moyen entre deux chargements d’une benne. Une telle analyse pourrait être renforcée par l’utilisation du processus des chaînes de Markov pour modéliser et optimiser ces temps.
2. **Évaluation des calendriers d’utilisation** : Explorer divers calendriers d’utilisation des engins pour sélectionner ceux qui répondent le mieux aux ambitions et aux objectifs de l’entreprise. Une planification rigoureuse pourrait également permettre d’optimiser les coûts et les délais.
3. **Choix des zones de remblayage optimales** : Identifier les meilleures zones de remblayage en fonction de la distance et des périodes de travail définies. Cette démarche permettrait de réduire les déplacements inutiles et de maximiser l'efficacité des opérations.

Ces perspectives offrent une opportunité d’enrichir notre modèle et de le rendre plus flexible pour répondre aux exigences des projets à venir. Elles soulignent également la nécessité d’une collaboration continue entre les équipes techniques et opérationnelles pour affiner les stratégies adoptées.

## Conclusion

Dans le cadre de ce projet de recherche opérationnelle, nous avons exploré les différentes dimensions d’une opération de terrassement en nous inspirant des travaux réalisés pour la construction du barrage de Nachtigal par la CCN. En partant des données disponibles et des contraintes opérationnelles spécifiques, nous avons conçu un modèle visant à minimiser le nombre total d’engins utilisés tout en respectant des exigences strictes concernant le volume déplacé, les durées maximales d’utilisation, et la simultanéité des engins.

Grâce à une approche méthodologique rigoureuse et à une analyse approfondie des données, notre solution optimise non seulement le nombre d'engins nécessaires mais propose également un calendrier d'utilisation détaillé. Cette optimisation a permis de démontrer qu’une gestion plus rationnelle et efficace des ressources était possible, aboutissant à une solution meilleure que celle utilisée lors des travaux du barrage de Nachtigal.

En outre, les perspectives identifiées, telles que l’étude des vitesses des engins, l’analyse des calendriers d’utilisation et la sélection des zones de remblayage optimales, offrent des pistes prometteuses pour renforcer davantage l’efficacité et l’adaptabilité de notre approche. Ces recommandations ouvrent la voie à une application généralisée de ce modèle dans d’autres projets de terrassement, contribuant ainsi à la maîtrise des coûts et à l’amélioration des performances opérationnelles.

Ce projet met en lumière l’importance de la recherche opérationnelle dans la résolution des problématiques complexes et confirme son rôle stratégique dans la prise de décisions éclairées pour les projets d’ingénierie civile.