**DS R5.A.FI.11- 2024**

**Durée : 1h30**

Accès autorisé aux supports de cours, TDs et corrections et uniquement au site geogebra.org

**Exercice 1 (7 points) : Programmation linéaire**

Une entreprise de logistique doit transporter des marchandises de plusieurs entrepôts vers différents magasins tout en minimisant les coûts de transport. Chaque entrepôt a une capacité limitée, et chaque magasin a une demande spécifique. Les coûts de transport varient en fonction de la distance entre les entrepôts et les magasins. L’entreprise souhaite minimiser ses coûts de transport.

**Données :**

Il y a 3 entrepôts : E1​, E2​, E3​, avec les capacités suivantes :

* E​1 : 120 unités.
* E2​ : 150 unités.
* E3 : 180 unités.

Il y a 4 magasins : M1, M2, M3 et M4​, avec les demandes suivantes :

* M1​ : 50 unités.
* M2​ : 60 unités.
* M3​ : 70 unités.
* M4​ : 100 unités.

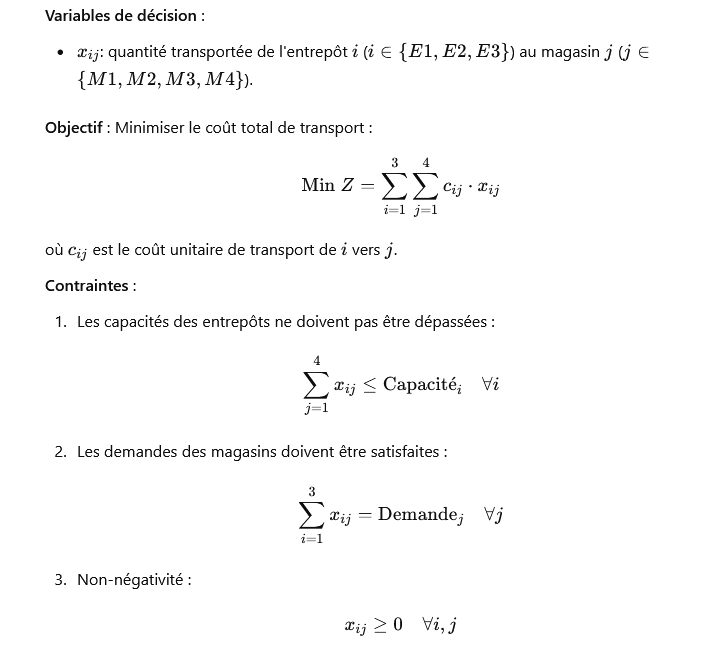
Les coûts de transport par unité (en euros) entre les entrepôts et les magasins sont donnés par la matrice suivante :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | M1 | M2 | M3 | M4 |
| E1 | 4 | 6 | 9 | 8 |
| E2 | 5 | 4 | 8 | 7 |
| E3 | 6 | 5 | 7 | 6 |

**Questions :**

1. Formulez ce problème sous la forme d’un programme linéaire.
2. Écrivez un programme en Python pour résoudre ce problème et trouvez la solution optimale. Collez le code de votre programme ici.
3. Donnez la solution optimale en indiquant la répartition des unités entre les entrepôts et les magasins.

Correction :



Code Python :

from pulp import LpProblem, LpMinimize, LpVariable, lpSum

# Données

capacities = [120, 150, 180]  # Capacité des entrepôts

demands = [50, 60, 70, 100]  # Demande des magasins

costs = [  # Coûts de transport

    [4, 6, 9, 8],

    [5, 4, 8, 7],

    [6, 5, 7, 6]

]

# Modèle

model = LpProblem("Transport\_Optimization", LpMinimize)

# Variables de décision

x = [[LpVariable(f"x\_{i+1}\_{j+1}", lowBound=0) for j in range(4)] for i in range(3)]

# Fonction objectif

model += lpSum(costs[i][j] \* x[i][j] for i in range(3) for j in range(4))

# Contraintes de capacité des entrepôts

for i in range(3):

    model += lpSum(x[i][j] for j in range(4)) <= capacities[i], f"Capacity\_E{i+1}"

# Contraintes de demande des magasins

for j in range(4):

    model += lpSum(x[i][j] for i in range(3)) == demands[j], f"Demand\_M{j+1}"

# Résolution

model.solve()

# Résultats

print("Statut :", model.status)

print("Coût total optimal :", model.objective.value())

for i in range(3):

    for j in range(4):

        print(f"x\_{i+1}\_{j+1} = {x[i][j].value()} unités")

1. Solution optimal:

Coût total minimal : 890 €.

Répartition des unités :

E1 -> M1 : 50 unités.

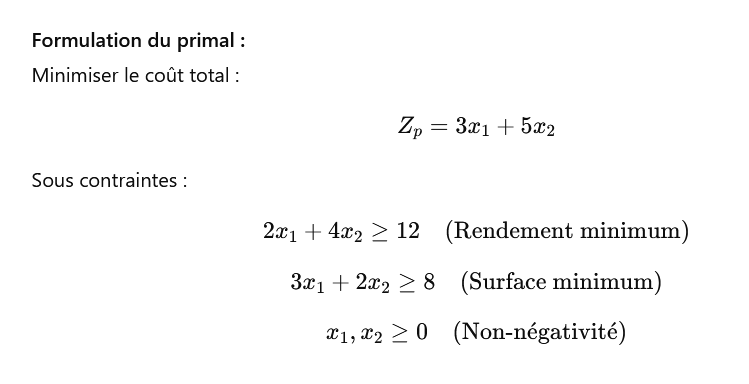
E1 -> M2 : 60 unités.

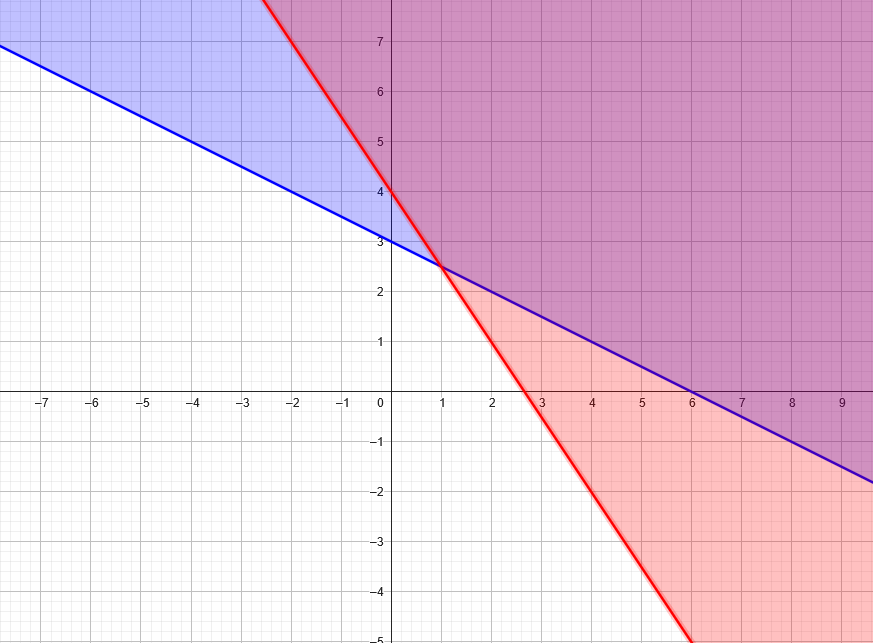
E2 -> M3 : 70 unités.

E3 -> M4 : 100 unités.

**Exercice 2 (5 points) : Problème Dual**

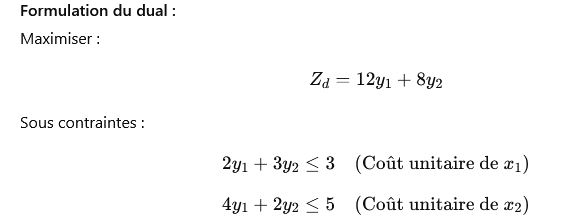
Un agriculteur doit allouer ses ressources entre deux cultures afin de minimiser les coûts de production, tout en satisfaisant les contraintes de rendement et de surface.



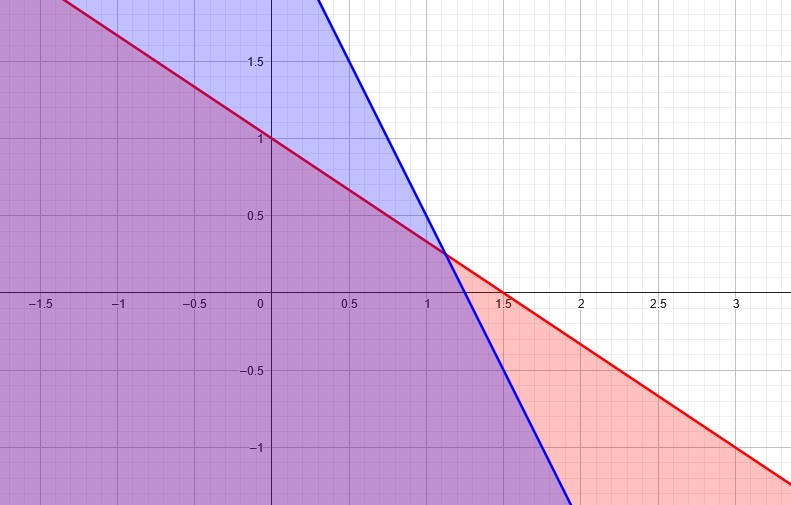
1. Donner une représentation graphique de ce problème et délimiter la zone des solutions réalisables.
2. 
3. Evaluer les solutions aux différents sommets de la zone des solutions réalisables et donnez la solution optimale.

|  |  |  |
| --- | --- | --- |
| X1 | X2 | z |
| 6 | 0 | 18 |
| 0 | 4 | 20 |
| 1 | 2.5 | 15.5 |

1. Formuler le problème Dual



1. Résoudre le problème Dual graphiquement et donner la solution optimale.



La solution optimale est donnée par le point suivant :

Y1= 1.125 et y2= 0.25 et Z=15.5

1. Qu’en déduisez-vous ?

Les valeurs optimales des deux programmes Primal et Dual sont égales, conformément au **théorème de dualité forte**.

**Exercice 3 (8 points) : Optimisation de la gestion énergétique**

Une entreprise gère un parc de centrales électriques et doit sélectionner une combinaison de centrales pour répondre à une demande d’énergie donnée. Les deux objectifs sont :

1. Maximiser la production énergétique (en kWh).
2. Minimiser les coûts de production (en euros).

Vous avez sept centrales potentielles (C1, C2, C3, C4, C5, C6, C7) avec les caractéristiques suivantes :

1. **Production énergétique (kWh)** :
   * C1 : 800 000 kWh
   * C2 : 750 000 kWh
   * C3 : 1 000 000 kWh
   * C4 : 600 000 kWh
   * C5 : 850 000 kWh
   * C6 : 745 000 kWh
   * C7 : 900 000 kWh
2. **Coût de production (en euros)** :
   * C1 : 40 000 €
   * C2 : 35 000 €
   * C3 : 50 000 €
   * C4 : 30 000 €
   * C5 : 42 000 €
   * C6 : 38 000 €
   * C7 : 55 000 €

**Étape 1 : Visualisation des solutions**

1. **Représentation graphique :** Créer un graphique bidimensionnel, placez la production énergétique en abscisse (kWh) et les coûts de production en ordonnée (euros). Chaque centrale doit être représentée comme un point avec ses coordonnées (kWh, €) et coller le graphique sur votre feuille de réponse.
2. **Pareto Front :** Identifier graphiquement le Pareto Front. Quelles solutions appartiennent aux Pareto Front ? Justifier votre réponse.

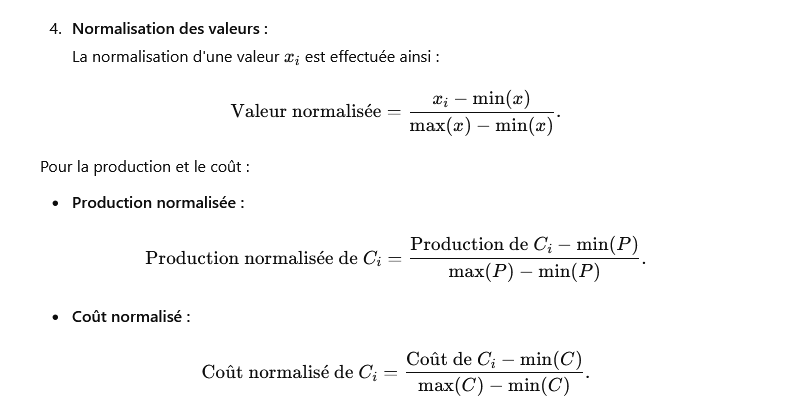
**Le Pareto Front contient les centrales de C1 jusqu’à C5**

**Étape 2 : Utilisation de la somme agrégée**

1. On compteutiliser la méthode de la somme agrégée avec des poids pour combiner les deux objectifs. Donner la formule générale de la fonction objective permettant de combiner les deux objectifs.

P= w(coût) \* Coût – w(production) \* Production

1. Normaliser les valeurs de production énergétique et de coût de chaque centrale pour avoir des valeurs comprises entre 0 et 1 pour les deux critères.



1. On considère les poids de préférences suivants : w1 = 0.6 pour le coût et w2 = 0.4 pour la production énergétique. Calculez la valeur agrégée pour chaque centrale. Quelle est la meilleure centrale ?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Centrale** | **Production (MillekWh)** | **Production normalisée** | **Coût (k€)** | **Coût normalisé** | **Somme agrégée** |
| C1 | 800 | 0,5 | 40 | 0,5 | -0,1 |
| C2 | 750 | 0,375 | 35 | 0,25 | 0 |
| C3 | 1000 | 1 | 50 | 1 | -0,2 |
| C4 | 600 | 0 | 30 | 0 | 0 |
| C5 | 850 | 0,625 | 42 | 0,6 | -0,11 |

**Étape 3 : Méthode de sur-classement**

1. Expliquez brièvement en quoi consiste la méthode de sur-classement et comment elle est utilisée pour comparer des solutions multicritères.

La méthode de sur-classement compare les options deux à deux sur chaque critère. Une solution est dite "préférée" si elle surclasse les autres sur un critère donné. Un score global est attribué en pondérant les performances de chaque option sur tous les critères.

1. Classer les centrales selon les deux critères de production énergétique et coût en attribuant à chaque centrale un score entier de votre choix en fonction de son classement.
2. Calculer le poids de préférence global de chaque centrale en considérant les poids de préférences suivants par critère : w1 = 0.6 pour le coût et w2 = 0.4 pour la production énergétique.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Centrale** | **Production (MillekWh)** | **Classement production** | **Coût (k€)** | **classement Coût** | **Poids global** |
| C1 | 800 | 3 | 40 | 3 | 3 |
| C2 | 750 | 2 | 35 | 4 | 3,2 |
| C3 | 1000 | 5 | 50 | 1 | 2,6 |
| C4 | 600 | 1 | 30 | 5 | 3,4 |
| C5 | 850 | 4 | 42 | 2 | 2,8 |

1. Donner la meilleure centrale dans ce cas.

C4