**TD 1 Correction**

**Exercice 1 :**

**Formulation du problème de programmation linéaire :**

Variables de décision :

* x : nombre de chaises à produire par jour.
* y : nombre de tables à produire par jour.

Fonction objectif à maximiser : Z=10x+20y

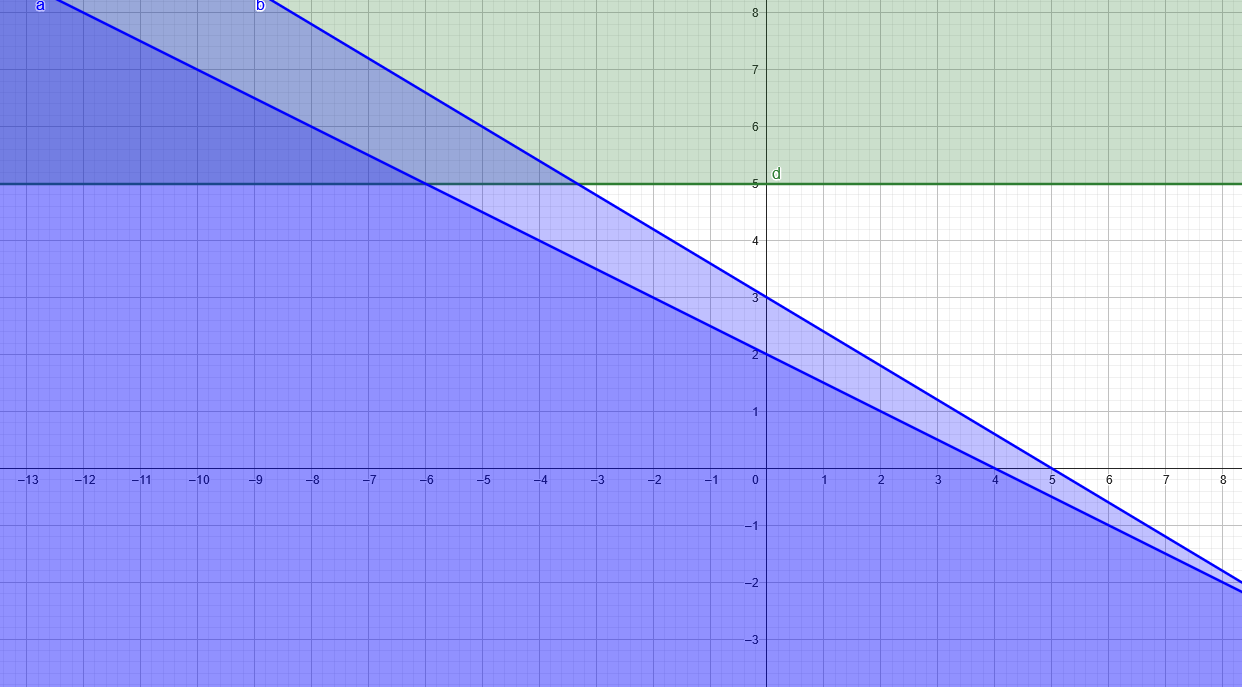
Contraintes :

1. Contrainte de temps de travail :
   * 2x+4y≤8 (8 heures de travail disponibles).
2. Contrainte de disponibilité de planches de bois :
   * 3x+5y≤15 (15 planches de bois disponibles).
3. Non-négativité des variables de décision : x≥0 et y>=0

**La solution optimale est Z=40 pour x=0 et y=2. Graphiquement, la zone des solutions réalisables est représentée par la surfaces en vert ci-dessous :**

****

**Question : 8 : on rajoute une contrainte de y>=5. Graphiquement, on voit bien qu’il n’y a plus de solutions réalisables.**



**Exercice 2 :**

**Formulation du problème de programmation linéaire :**

Variables de décision :

* xij​ : quantité de tâche i attribuée au serveur j.

**Fonction objectif à minimiser** :

Contraintes :

1. Contrainte de capacité des serveurs :

Capacité (j) pour chaque serveur j.

1. Contrainte d'attribution des tâches :

(chaque tâche doit être attribuée exactement à un serveur

1. Non-négativité des variables de décision : xij appartient à {0,1} pour tous i et j.

**Question 2 : voir le fichier texte exo2.py**

La solution que vous devez trouver en exécutant le code est la suivante :

Assigner la tâche 1 au serveur A

Assigner la tâche 2 au serveur A

Assigner la tâche 3 au serveur A

Assigner la tâche 4 au serveur C

Assigner la tâche 5 au serveur C

Consommation totale d'énergie: 400.0 Wh

**Question 3 :**

Mathématiquement, cela peut être formulé com me suit :

J’ai modifié le seuil de temps d’exécution pour rendre le problème impossible à résoudre avec 4 au lieu de 500.

Où :

* Temps(i,j) représente le temps nécessaire pour exécuter la tâche i sur le serveur j.

**Question 4 :**

**Voir le fichier exo2-question4.py**

**Exercice 3 :**

**Formulation du problème en programmation linéaire en nombres entiers :**

Variables de décision :

* xi: 1 si le projet i est sélectionné, 0 sinon (variable binaire).

Fonction objectif à maximiser : Z=∑i Profit(i)⋅xi

Contrainte budgétaire : ∑i Coût(i)⋅xi ≤ 10000

Contrainte de variables binaires : xi ∈ {0,1} pour i=1,2,3,4,5

**Pour le code : voir le fichier exo3.py**

**Exercice 4 :**

**Les variables de décision de ce problème peuvent être exprimées comme suit :**

* xij​ : quantité de colis livrée du point i au point j.

Fonction objectif à minimiser : Z=∑i≠j Distance(i,j)⋅xij

Contraintes :

1. Contrainte de demande des clients :
   * ∑j≠i xij = Demande(i) pour chaque client i.
2. Contrainte de flux sortant :
   * ∑j≠i xij=1 pour chaque client i.
3. Contrainte de flux entrant :
   * ∑j≠i xji=1 pour chaque client i.

Chaque variable xij représente la quantité de colis livrée du point i au point j. Les contraintes assurent que chaque client est desservi exactement une fois et que la demande de colis est satisfaite. La fonction objectif vise à minimiser la distance totale parcourue.

**Le code python de la solution utilisant l’algorithme Branch and Bound est dans le fichier exo4.py**

**Exercice 5 :**

**Question 1 : La solution optimale est :**

Z = 220  
X1 = 2  
X2 = 1.5

Le problème Dual est :

Une image contenant texte, Police, écriture manuscrite, calligraphie

Description générée automatiquement

La solution optimale du problème Dual est :

Z = 220  
y1 = 3.3333333333333  
y2 = 20  
y3 = 0

**Exercice 6 :**

La solution optimale du problème primal est :

Z = 440000  
X1 = 800  
X2 = 0  
X3 = 300

Le problème Dual est :

Une image contenant texte, Police, écriture manuscrite, calligraphie

Description générée automatiquement

La solution optimale du problème Dual est :

Z = 440000  
y1 = 120  
y2 = 220  
y3 = 0