USMB

EXERCICES

Méthodes d'optimisation

Melih Cetinkaya
18/02/2025

TABLE DES MATIERES

1. PROGRAMMATION LINEAIRE (PL) – OPTIMISATION DE PRODUCTION

PROBLEME :

Une boulangerie produit deux types de pains : **baguettes traditionnelles** et **pains complets**. Chaque type de pain nécessite un certain nombre d'heures de travail et une quantité de farine.

DONNEES DISPONIBLES :

- Une baguette nécessite **2 heures de travail** et **1 kg de farine**.
- Un pain complet nécessite **1 heure de travail** et **2 kg de farine**.
- La boulangerie dispose de **40 heures de travail** et **30 kg de farine** par jour.
- La vente rapporte **1,50 € par baguette** et **2 € par pain complet**.

QUESTIONS :

1. **Formulez un modèle de programmation linéaire** pour maximiser les revenus de la boulangerie.
2. **Résolvez graphiquement** le problème en représentant l'espace des solutions possibles.
3. **Déterminez la solution optimale** et le revenu maximal.
4. Si la boulangerie décide de ne pas produire plus de **10 baguettes**, comment cela affecte-t-il la solution ?

2. ALGORITHME A* – RECHERCHE DU CHEMIN OPTIMAL

PROBLEME :

Un robot doit se déplacer dans un entrepôt en suivant **le chemin le plus court** pour atteindre une boîte de colis.

DONNEES :

L'entrepôt est représenté sous forme de graphe avec des **nœuds** représentant les intersections et des **arcs pondérés** représentant le temps en secondes pour aller d'un point à un autre. Une **fonction heuristique h(n)** est donnée, correspondant au **temps estimé restant** pour atteindre la boîte.

Départ	**Destination**	**Temps (coût)**
A	B	3
A	C	1
B	D	5
C	D	2
C	E	4
D	E	3

HEURISTIQUE H(N) ESTIMEE :

Nœud	h(n)
A	6
B	4
C	5
D	2
E	0

QUESTIONS :

1. Appliquez **l'algorithme A*** pour déterminer le chemin optimal de **A à E**.
2. Décrivez le contenu des listes **OPEN et CLOSED** à chaque étape.
3. Quelle est la valeur du chemin trouvé et son coût total ?

3. K-MEANS – SEGMENTATION DE CLIENTS POUR UNE ENTREPRISE

PROBLEME :

Une entreprise souhaite regrouper ses clients en fonction de leur **fréquence d'achat** et du **montant total dépensé** pour proposer des offres adaptées.

Elle dispose des **6 clients suivants**, dont les caractéristiques sont :

Client	Fréquence d'achat (X)	Dépense totale (Y)
C1	2	50
C2	8	150
C3	5	90
C4	7	130
C5	3	60
C6	6	100

On applique **l'algorithme K-Means avec 2 clusters**, et les **centres initiaux** sont :

- **Centre 1 : (3, 70)**
- **Centre 2 : (7, 140)**

QUESTIONS :

1. Expliquez le fonctionnement de **l'algorithme K-Means**.
2. **Assignez chaque client** au centre le plus proche en utilisant la **distance euclidienne**.
3. **Recalculez les centres** après cette première répartition.
4. **Répétez l'étape de réassignation** et déterminez les **centres finaux**.

4. DESCENTE DE GRADIENT – REGLAGE D'UNE MACHINE INDUSTRIELLE

PROBLEME :

Une usine cherche à optimiser la **température d'une machine** pour obtenir la **meilleure qualité de production**. On modélise la qualité par la fonction :

Q(T) = −(T−50)2 + 100*Q*(*T*) = −(*T*−50)2 + 100

Où **Q(T)** représente la qualité et **T** la température de la machine en degrés Celsius.

QUESTIONS :

1. **Expliquez la méthode de descente de gradient** et son principe d'optimisation.
2. Si l'usine commence avec une température **T = 70°C** et un pas de **0,1**, appliquez **2 itérations de la descente de gradient** en calculant la dérivée de **Q(T)**.
3. Quelle est la température optimale attendue pour maximiser la qualité ?

5. BRANCH AND BOUND – OPTIMISATION DE LA LIVRAISON DE COLIS

PROBLEME :

Une entreprise de transport doit **minimiser le coût de livraison** entre **quatre villes (A, B, C et D)**. Chaque ville doit être visitée une fois avant de revenir au point de départ (**problème du voyageur de commerce**).

COUTS DE TRANSPORT (EN EUROS) ENTRE VILLES :

De	Vers	Coût
A	B	10
A	C	15
A	D	20
B	C	35
B	D	25
C	D	30

QUESTIONS :

1. Décrivez le **principe de l'algorithme Branch and Bound**.
2. Trouvez le chemin **optimal en minimisant les coûts** en appliquant **Branch and Bound**.
3. Quelle est la solution optimale et son coût total ?

RESUME DES TYPES D'EXERCICES ATTENDUS

Thème	Type d'exercice	Compétences évaluées
Programmation Linéaire	Optimisation de production	Formulation de PL, résolution graphique, contraintes
Algorithme A*	Recherche du chemin optimal	Manipulation des listes OPEN/CLOSED, heuristique
K-Means	Segmentation de clients	Calcul de distances, réassignation, convergence
Descente de Gradient	Optimisation d'un réglage machine	Calcul de dérivée, itérations de gradient
Branch and Bound	Livraison optimisée	Structuration de l'exploration des solutions

Ces exercices couvrent **les concepts principaux du cours** et **les méthodes d'optimisation et d'apprentissage automatique**. Ils sont représentatifs de ce qui pourrait être demandé lors du **Devoir Surveillé**.

CORRECTIONS DETAILLEES

EXERCICE 1 : PROGRAMMATION LINEAIRE – OPTIMISATION DE PRODUCTION (BOULANGERIE)

DONNEES :

- Baguette :
  - 2 heures de travail
  - 1 kg de farine
  - Vente à 1,50 €
- Pain complet :
  - 1 heure de travail
  - 2 kg de farine
  - Vente à 2 €
- Ressources journalières :
  - 40 heures de travail
  - 30 kg de farine

CORRECTION :

1. **Définition des variables :**
   - Soit x le nombre de baguettes produites et y le nombre de pains complets produits.
2. **Fonction objective :**
   On souhaite maximiser le revenu total :

$Z = 1{,}50x + 2y$

3. **Contraintes :**
   - **Temps de travail :**

$2x + y \leq 40$

   - **Farine :**

$x + 2y \leq 30$

   - Non-négativité :

x ≥ 0, y ≥ 0

4. **Représentation graphique et recherche des sommets :**
   - Tracez les droites 2x + y = 40 et x + 2y = 30dans le plan (x,y).
   - Déterminez la zone réalisable, qui est l'intersection des demi-plans définies par les inégalités.
   - Identifiez les points d'intersection (sommets). Par exemple :
     - En posant x = 0 dans x + 2y = 30, on trouve y = 15.
     - En posant y = 0 dans 2x + y = 40, on trouve x = 20.
     - L'intersection des deux droites se calcule avec Geogebra :
5. **Calcul de la fonction objectif en chaque sommet :**
   - Pour (16,67,6,67):

Z ≈ 1,50×16,67 + 2×6,67 ≈ 25 + 13,33 = 38,33 €

Solution optimale :
La solution optimale est obtenue pour x≈16,67*x*≈16,67 baguettes et y≈6,67*y*≈6,67 pains complets, ce qui donne un revenu maximal d'environ 38,33 €.
Remarque : Si l'on impose une contrainte supplémentaire, par exemple x≤10*x*≤10, il faut recalculer la zone réalisable et chercher le nouveau sommet optimal.

EXERCICE 2 : ALGORITHME A* – RECHERCHE DU CHEMIN OPTIMAL

DONNEES DU GRAPHE :

Départ	Destination	Coût
A	B	3
A	C	1
B	D	5
C	D	2
C	E	4
D	E	3

HEURISTIQUE H(N) :

Nœud	h(n)
A	6
B	4
C	5
D	2
E	0

CORRECTION :

1. **Initialisation :**
   - OPEN = {A} avec f(A) = g(A) + h(A) = 0 + 6 = 6

- CLOSED = {}.

2. **Itération 1 :**
   - Sélectionnez A (le nœud avec le plus petit f*f* dans OPEN).
   - Explorez les voisins de A :
     - Pour B : g(B) = 0 + 3 = 3 et f(B) = 3 + 4 = 7.
     - Pour C : g(C) = 0 + 1 = 1 et f(C) = 1 + 5 = 6.
   - Placez A dans CLOSED et mettez à jour OPEN :
     OPEN = {B (f = 7), C (f = 6)}.
3. **Itération 2 :**
   - Sélectionnez C (f=6).
   - Explorez les voisins de C :
     - Pour D : g(D) = g(C) + 2 = 1 + 2 = 3 et f(D) = 3 + 2 = 5.
     - Pour E : g(E) = g(C) + 4 = 1 + 4 = 5 et f(E) = 5 + 0 = 5.
   - Placez C dans CLOSED et mettez à jour OPEN :
     OPEN = {B (f = 7), D (f = 5), E (f = 5)}.
4. **Itération 3 :**
   - Choisissez D ou E (les deux ont f=5). Supposons que vous sélectionniez D.
   - Explorez D :
     - D a pour voisin E. Calcul via D : g(E) = g(D) + 3 = 3 + 3 = 6.
     - E est déjà dans OPEN avec g(E) = 5 (meilleure valeur), donc pas de mise à jour.
   - Placez D dans CLOSED.
     OPEN = {B (f = 7), E (f = 5)}.
5. **Itération 4 :**
   - Sélectionnez E (f=5).
   - Comme E est le but, l'algorithme s'arrête.
   - **Chemin optimal :**
     Pour reconstituer le chemin, on suit les prédécesseurs enregistrés lors de l'exploration (par exemple, A → C → E).
   - **Coût total :** g(E) = 5.

EXERCICE 3 : K-MEANS – SEGMENTATION DE CLIENTS

DONNEES DES CLIENTS :

Client	Fréquence d'achat (X)	Dépense totale (Y)
C1	2	50
C2	8	150
C3	5	90
C4	7	130
C5	3	60
C6	6	100

CENTRES INITIAUX :

- Centre 1 : (3, 70)
- Centre 2 : (7, 140)

CORRECTION :

1. **Affectation initiale :**
   Pour chaque client, calculez la distance euclidienne aux deux centres.
   Exemple pour C1 (2,50) :

   - Distance à Centre 1 :

$d1 = (3-2)2 + (70-50)2 = 1+400 \approx 20,02$

   - Distance à Centre 2 :

$d2 = (7-2)2 + (140-50)2 = 25 + 8100 \approx 90,14$

2. → C1 est assigné au cluster du Centre 1.

3. **Répéter pour tous les clients :**
   Attribuez chaque client au centre le plus proche en comparant les distances.

4. **Recalcul des centres :**
   Une fois tous les clients affectés, calculez les nouveaux centres pour chaque cluster en faisant la moyenne des coordonnées des points affectés.

   - Par exemple, si le cluster 1 regroupe C1, C3, C5 et C6, le nouveau centre sera :

x new = $xC1+xC3+xC5+xC64$,

y new = $yC1+yC3+yC5+yC64$

Itération et convergence :
Réaffectez les clients aux nouveaux centres et recalculez si nécessaire, jusqu'à ce que les centres ne changent plus.

5. **Résultat final attendu :**
   Après quelques itérations, l'algorithme converge vers des centres finaux (par exemple, Cluster 1 : centre autour de (4,75) et Cluster 2 : centre autour de (7,140)) et une répartition stable des clients dans les clusters.

EXERCICE 4 : DESCENTE DE GRADIENT – REGLAGE DE LA TEMPERATURE D'UNE MACHINE

FONCTION DE QUALITE :

$Q(T) = -(T-50)2 + 100$

Cette fonction représente une parabole inversée, dont le maximum se situe pour une température T optimale.

CORRECTION :

1. **Calcul de la dérivée :**
   La dérivée de Q(T) est :

$Q'(T) = -2(T-50)$

2. **Choix du point de départ et du pas :**
   - Température initiale $T0 = 70$
   - Pas $(\alpha) = 0{,}1$
3. **Première itération :**
   - Calculez $Q'(70) = -2(70 - 50) = -2\times20 = -40$
   - Pour maximiser Q(T), on se déplace dans la direction du gradient (car le gradient donne la direction d'augmentation).
   - Mise à jour :

$T1 = T0 + \alpha\times Q'(70) = 70 + 0{,}1\times(-40) = 70-4 = 66.$

4. **Deuxième itération :**
   - Calculez $Q'(66) = -2(66 - 50) = -2\times16 = -32.$
   - Mise à jour :

$T2 = 66 + 0{,}1\times(-32) = 66 - 3{,}2 = 62{,}8.$

5. **Interprétation :**
   On constate que la température diminue vers 50 °C. Le maximum de Q(T) est atteint pour T=50. Le processus convergera vers 50 °C après plusieurs itérations.

EXERCICE 5 : BRANCH AND BOUND – OPTIMISATION DE LA LIVRAISON DE COLIS (TSP)

DONNEES :

De	Vers	Coût
A	B	10
A	C	15
A	D	20
B	C	35
B	D	25
C	D	30

OBJECTIF :

Trouver le chemin qui visite chaque ville une fois et revient à A avec le coût total minimum.

CORRECTION :

1. **Génération des branches :**
   - Commencez par la ville A.
   - Énumérez toutes les permutations possibles pour visiter B, C, D et revenir à A.
     Par exemple, un chemin possible est A → B → C → D → A.
2. **Calcul des coûts totaux :**
   Pour le chemin A → B → C → D → A :
   - Coût = A→B(10) + B→C(35) + C→D(30) + D→A(20) = 95 €.
3. **Application de Branch and Bound :**
   - Pour chaque branche, calculez une borne inférieure (en ajoutant les coûts déjà accumulés et une estimation du coût restant).
   - Si une branche a déjà un coût supérieur à une solution complète déjà trouvée, on l'élimine.
   - Continuez l'exploration des branches jusqu'à identifier le chemin avec le coût minimal.
4. **Recherche de la solution optimale :**
   - Après avoir évalué toutes les branches ou éliminé celles qui dépassent la borne, comparez les coûts des chemins complets restants.
   - La solution optimale est le chemin avec le coût total le plus bas.
   - Par exemple, si le chemin A → C → D → B → A donne un coût total de 80 € et que toutes les autres branches ont un coût supérieur, alors la solution optimale est ce chemin.

Conclusion

Ces corrections illustrent :

- La formulation d'un problème en programmation linéaire, la recherche graphique et l'identification de la solution optimale.
- La simulation d'un algorithme A*, avec la gestion des listes OPEN et CLOSED pour trouver le chemin le plus court.
- L'application de l'algorithme K-Means en affectant des points à des centres, recalculant et réaffectant jusqu'à convergence.
- Le fonctionnement de la descente de gradient pour maximiser une fonction en ajustant progressivement la variable.
- L'utilisation de Branch and Bound pour explorer les solutions d'un problème combinatoire (TSP) et éliminer les branches non prometteuses.

Chaque exercice est traité étape par étape pour garantir une compréhension complète des méthodes et du raisonnement utilisé pour obtenir la solution optimale.