Optimisation Linéaire

Optimisation en nombres entiers

Optimisation en nombres entiers

De nombreux problèmes d'optimisation peuvent être formulés comme des problèmes d'optimisation linéaire, avec la contrainte supplémentaire que certaines des variables sont entières ou binaires.

Dans certains cas, ces contraintes supplémentaires sont faciles à prendre en compte. Dans d'autres cas, le problème devient considérablement plus difficile à résoudre. Nous présentons une méthode de résolution des problèmes d'optimisation en nombre entiers: la méthode de branch and bound.

Optimisation en nombres entiers

Optimisation en nombres entiers.

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^T x \\ \text{tel que} \quad & Ax \geq b, \\ & x_i \text{ entier pour tout } i. \end{aligned}$$

Optimisation mixte.

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^T x \\ \text{tel que} \quad & Ax \geq b, \\ & x_i \text{ entier pour } i \in \mathcal{I} \subset \{1, 2, \ldots, n\}. \end{aligned}$$

Optimisation en variables binaires.

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^T x \\ \text{tel que} \quad & Ax \geq b, \\ & x \in \{0, 1\}^n. \end{aligned}$$

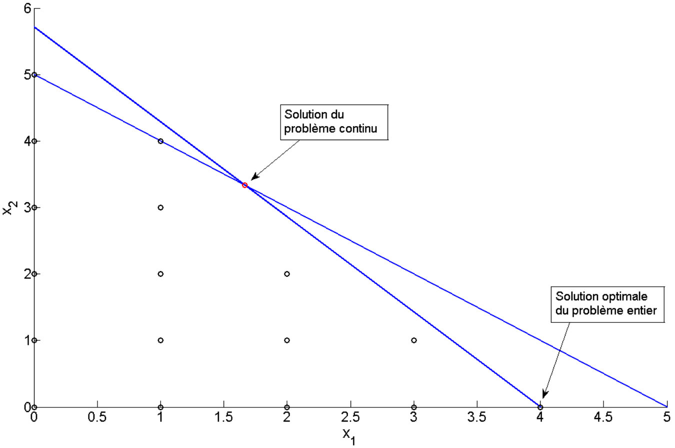
Avec des entiers, ça se complique. . .

$$\max_x \quad 17x_1 + 12x_2 \quad \text{tel que} \quad \begin{aligned} 10x_1 + 7x_2 &\leq 40, \\ x_1 + x_2 &\leq 5, \\ x_1, x_2 &\geq 0 \text{ et entiers.} \end{aligned}$$

Le problème linéaire obtenu sans la contrainte 'x_1, x_2 entiers' est une relaxation du problème initial. La solution optimale du problème relaxé est donnée par $(5/3; 10/3)$. La valeur correspondante de la fonction objectif est $205/3 = 68.33$.

En arrondissant les composantes de la solution optimale du problème relaxé aux entiers les plus proches, on trouve la solution (2, 3) qui n'est pas admissible. La solution admissible la plus proche de la solution optimale du problème relaxé est (1, 3). Cette solution n'est pas optimale. La solution optimale est (4, 0). Cette solution est éloignée de la solution optimale du problème relaxé.

Avec des entiers, ça se complique. . .



La situation est pire encore dans certains cas. Par exemple le cas où les variables sont binaires et la solution optimale du problème relaxé est (0.5, 0.5, ..., 0.5); ce qui ne donne pas beaucoup d'information sur la solution optimale en nombres entiers...

En fait il est souvent difficile de déterminer si un problème en variables binaires possède une solution admissible.

Exemple. Variables de décision

Quatre décisions $x_1, x_2, x_3, x_4 \in \{0, 1\}$ doivent être prises. Le bénéfice résultant de décisions positives est donné, respectivement, par 9, 5, 6 et 4.

Conditions exclusives. Les décisions 3 et 4 sont mutuellement exclusives; c'est 3 ou 4, mais pas les deux: $x_3 + x_4 \leq 1$.

Contraintes alternatives. Au moins une des décisions 1 ou 3 doit être positive; c'est 1 ou 3, ou, éventuellement, les deux: $x_1 + x_3 \geq 1$.

Conditions contingentes. La décision 3 ne peut âtre positive que si la décision 2 l'est: $x_3 \leq x_2$.

$$
\begin{aligned}
\min_x \quad & 9x_1 + 5x_2 + 6x_3 + 4x_4 \\
\text{tel que} \quad & x_3 + x_4 \leq 1, \\
& x_1 + x_3 \geq 1, \\
& x_2 - x_3 \geq 0, \\
& x_1, x_2, x_3, x_4 \in \{0, 1\}.
\end{aligned}
$$

Exemple. Contraintes alternatives

L'ensemble des vecteurs qui satisfont les contraintes $0 \leq x \leq 1$, $a_1^T x \leq b_1$ **et** $a_2^T x \leq b_2$ est un polyèdre.

L'ensemble des vecteurs qui satisfont les contraintes $0 \leq x \leq 1$, $a_1^T x \leq b_1$ **ou** $a_2^T x \leq b_2$ n'est pas un polyèdre. Pour modéliser ces contraintes, nous introduisons des variables binaires y_1 et y_2 et

$$a_1^T x - b_1 \leq M(1 - y_1),$$

$$a_2^T x - b_2 \leq M(1 - y_2),$$

et $y_1 + y_2 = 1$.

M est une constante suffisemment grande pour que les contraintes $a_i^T x - b_i \leq M$ soient toujours satisfaites pour $0 \leq x \leq 1$. (Par exemple, $M \geq \max(||a_1||_1 + |b_1|, ||a_2||_1 + |b_2|)$).

Ces contraintes sont équivalentes à la condition qu'une au moins une des contraintes $a_i^T x \leq b_i$ est satisfaite.

Problème d'affectation (Assignment problem)

Répartition optimale de tâches: Il y a n personnes disponibles et n tâches à accomplir. A chaque personne, on affecte exactement une tâche. On souhaite que toutes les tâches soient affectées. Le coût de la réalisation de la tâche j par la personne i est de c_{ij}. On cherche une affectation des tâches qui minimise le coût total.

Problème équivalent: Dans une matrice carrée on choisit exactement un élément par ligne et un élément par colonne. La somme des éléments choisis est minimum.

Formulation

1. **Choix des variables**:

$$x_{ij} = \begin{cases} 1 & \text{si la personne } i \text{ effectue la tâche } j, \\ 0 & \text{sinon.} \end{cases}$$

2. **Contraintes**:

$$\sum_{j=1}^{n} x_{ij} = 1 \ \forall i, \qquad \text{(chaque personne a une tâche)}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \ \forall j, \qquad \text{(chaque tâche est effectuée)}$$

$$x_{ij} \in \{0, 1\} \text{ pour tout } i, j.$$

3. **Fonction objectif**: le coût total est minimum

$$\min_x \sum_{i,j} c_{ij} x_{ij}.$$

Problème du sac à dos (Knapsack problem)

On rempli un sac à dos d'objets. Il y a n objets susceptibles de rentrer dans le sac. L'objet i est de poids a_i et procure une satisfaction c_i (par exemple, son prix). Parmi les ensembles d'objets dont la somme des poids ne dépasse pas b trouvez ceux pour lesquels la somme des satisfactions est maximum.

Formulation

1. **Choix des variables**:

$$x_i = \begin{cases} 1 & \text{si le produit } i \text{ est inclus,} \\ 0 & \text{sinon.} \end{cases}$$

2. **Contraintes**:

$$\sum_{i=1}^{n} a_i x_i \leq b \; \forall i, \qquad \text{(capacité du sac à dos)}$$

$$x \in \{0,1\}^n.$$

3. **Fonction objectif**: la satisfaction globale est maximum

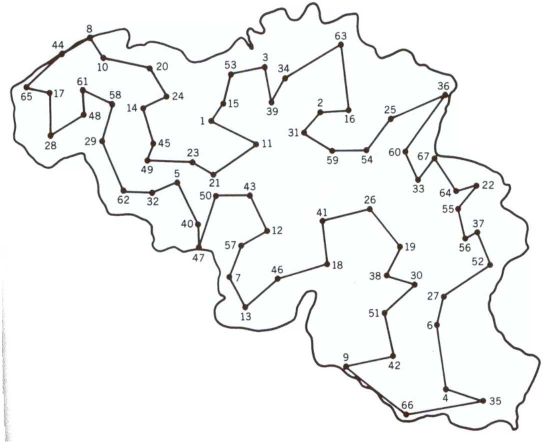
$$\max_x \sum_{i=1}^{n} c_i x_i.$$

Problème du voyageur de commerce (Traveling salesman problem)

Un voyageur de commerce visite n villes. Il passe par chaque ville exactement une fois et retourne finalement à sa ville de départ. Le coût (temps, distance, etc.) du trajet de la ville i à la ville j est de c_{ij}. Trouver le circuit qui minimise la somme des coûts.

C'est sans doute le problème le plus connu d'optimisation. Le problème est simple à expliquer et il appelle une solution.

Le problème se manifeste de multiples façons. Ce peut être le choix de la tournée d'un livreur, d'un facteur ou d'une infirmière à domicile, ou encore la programmation d'une machine qui place des composants sur un circuit imprimé, etc.


8
44
63
3
53
34
10
20
36
61
58
17
65
24
14
15
39
2
16
25
48
1
31
11
29
45
28
23
60
67
59
54
49
5
21
43
33
22
50
62
32
64
26
41
55
40
12
37
57
19
56
47
52
46
18
7
38
30
27
51
13
6
9
42
4
35
66

L'histoire du TSP

1920's: The mathematician and economist Karl Menger publicizes it among his colleagues in Vienna.

1930's: The problem reappears in the mathematical circles of Princeton.

1940's: The problem is studied by statisticians in connection with an agricultural application and the mathematician Merill Flood popularizes it among his colleagues at the RAND Corporation. Eventually, the TSP gained notoriety as the prototype of a hard problem in combinatorial optimization: examining the tours one by one is out of the question because of their large number, and no other idea was on the horizon for a long time.

1954: George Dantzig and co-authors publish a description of a method for solving the TSP and illustrate the power of this method by solving an instance with 49 cities, an impressive size at that time. They created this instance by picking one city from each of the 48 states in the U.S.A. (Alaska and Hawaii became states only in 1959) and adding Washington, D.C.; the costs of travel between these cities were defined by road distances.

L'histoire du TSP

1962: Proctor and Gamble runs a contest requiring solving a TSP on a specified 33 cities. There was a tie between many people who found the optimum.

1977: Groetschel finds the optimal tour of 120 cities from what was then West Germany.

1987: Padberg and Rinaldi find the optimal tour of 532 AT&T switch locations in the USA.

1998: Applegate, Bixby, Chvatal, and Cook find the optimal tour of the 13,509 cities in the USA with populations > 500.

From: `http://www.keck.caam.rice.edu/tsp/`. Applegaten, Bixby, Chvatal, Vasek and Cook, On the solution of traveling salesman problems. Proceedings of the International Congress of Mathematicians, Vol. III (Berlin, 1998).

Formulation

1. **Choix des variables**:

$$x_{ij} = \begin{cases} 1 & \text{si le voyageur va directement de } i \text{ à } j \\ 0 & \text{sinon.} \end{cases}$$

   (Pas de variable x_{ii}.)

2. **Contraintes**:

$$\sum_{j=1, j \neq i}^{n} x_{ij} = 1 \ \forall i, \qquad \text{(quitte à la ville } i \text{ une fois)}$$

$$\sum_{i=1, i \neq j}^{n} x_{ij} = 1 \ \forall i, \qquad \text{(arrive à la ville } j \text{ une fois)}$$

3. **Fonction objectif**: le coût total est minimisé

$$\min_x \sum_{i,j=1}^{n} c_{ij} x_{ij}.$$

Mais cela ne suffit pas. . .

Le tour doit être connecté

Contraintes de coupe. Quelle que soit la partition $S_1 \cup S_2 = \{1, 2, \dots, n\}$ avec $S_1 \cup S_2 = \emptyset$ et $S_1, S_2 \neq \emptyset$, il doit y avoir un arc entre S_1 et S_2

$$\sum_{i \in S_1} \sum_{j \in S_2} x_{ij} \geq 1.$$

Absence de petits circuits. Il n'y a pas de circuit de taille inférieure à n. Quel que soit le sous-ensemble S avec $2 \leq |S| \leq n - 1$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1.$$

Il existe d'autres formlations plus efficaces.

Enumération

Tous ces problèmes peuvent être résolus par énumération puisque l'ensemble des solutions admissibles est chaque fois fini. Il est fini mais il est grand...

- **Problème d'affectation.** Il y a n personnes et n tâches. Il y a donc $n!$ affectations possibles.
- **Problème du sac à dos**. Il y a n objets et chaque objet peut être choisi ou laissé de côté. Il y a 2^n choix possibles.
- **Problème du voyageur de commerce.** Au départ de la ville 1, il y a $n-1$ choix possibles, ensuite $n-2$, etc. Il y a donc $(n-1)!$ circuits possibles en tout. Dans un graphe complet de 101 noeuds il y a $100! \approx 9.3310^{157}$ circuits possibles.

Relaxation

Soit le problème en nombres entiers

$$\begin{aligned} \max_x \quad & c^T x \\ \text{tel que} \quad & Ax \leq b \\ & x \geq 0 \text{ et } x \text{ entier.} \end{aligned}$$

On construit la relaxation de ce problème en éliminant la contrainte 'x entier':

$$\begin{aligned} \max_x \quad & c^T x \\ \text{tel que} \quad & Ax \leq b \\ & x \geq 0. \end{aligned}$$

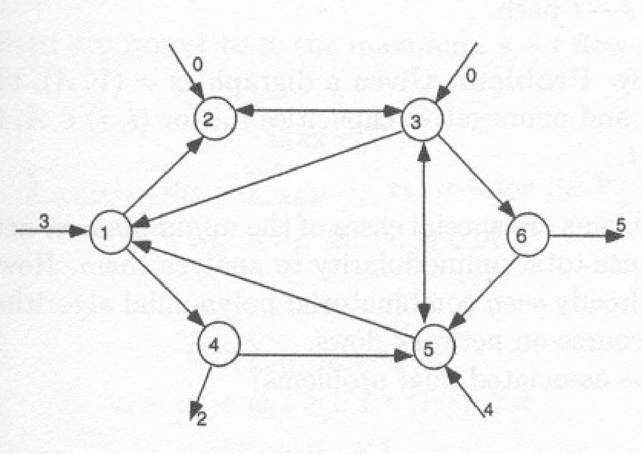
Si le problème relaxé ne possède pas de solution admissible, le problème de départ n'en possède pas non plus. Une solution optimale du problème relaxé fournit une borne supérieure sur l'objectif optimal du problème initial. Si la solution optimale du problème relaxé est entière, alors cette solution est également optimale pour le problème initial.

Problème du flot de coût minimum (Min cost flow)

Transmission dans un réseau de communication.

On considère un graphe dirigé donné par un ensemble de noeuds V et d'arcs E. En chaque noeud i il y a une quantité b_i qui entre $(b_i > 0)$ ou qui sort $(b_i < 0)$. Nous supposons que les quantité entrantes et sortantes s'équilibrent, $\sum_i b_i = 0$.

L'arc (i, j) a une capacité maximale de h_{ij} et est de coût unitaire c_{ij}. Nous souhaitons trouver un flot admissible de coût minimum.


0
0
2
3
3
1
6
5
4
5
2
4

Problème du flot de coût minimum

On définit

$V^+(i) = \{k|(i,k) \in E\}$. (noeuds dont l'arête avec i sort de i)
$V^-(i) = \{k|(k,i) \in E\}$. (noeuds dont l'arête avec i va vers i)

1. **Choix des variables**: x_{ij} est le flot dans l'arc (i,j).
2. **Contraintes**: Le flot est conservé en chaque noeud:

$$\sum_{k \in V^+(i)}^{n} x_{ik} - \sum_{k \in V^-(i)}^{n} x_{ki} = b_i \; \forall\, i \in V.$$

   Les capacités maximales ne sont pas dépassées:

$$0 \leq x_{ij} \leq h_{ij} \quad \forall\, (i,j) \in E.$$

3. **Fonction objectif**: le coût total est minimisé

$$\min_x \sum_{(i,j) \in E} c_{ij} x_{ij}.$$

Flot de coût minimum: cas entier

Théorème. Si les demandes b_i et les capacités h_{ij} d'un problème de flot de coût minimum sont entières alors il existe une solution optimale entière.

Preuve (esquisse). Le problème peut s'écrire sous la forme

$$\min_x \quad c^T x \quad \text{tel que} \quad \begin{array}{r} Ax \leq b, \\ x \geq 0 \text{ et } x \text{ entier,} \end{array}$$

où les entrées de A et b sont entières. La relaxation de ce problème est

$$\min_x \quad c^T x \quad \text{tel que} \quad \begin{array}{r} Ax \leq b, \\ x \geq 0. \end{array}$$

Les solutions admissibles de base de ce problème s'écrivent sous la forme (après permutation) $x = (x_B, x_N) = (A_B^{-1}b, 0)$ où A_B est une sous-matrice de $[A, I]$. Ces sous matrices ont toutes un déterminant égal à ± 1. (Ce sont des matrices unmodulaires; voir par exemple
`http://fr.wikipedia.org/wiki/Matrice_unimodulaire`).

Problème du transport

Un produit est transporté de m origines vers n destinations. Le produit est disponible en quantités $a_1, a_2, \ldots, a_m$ aux origines et les demandes aux destinations sont de $b_1, b_2, \ldots, b_n$. Le coût de transport d'une unité du produit de l'origine i à la destination j est de c_{ij} (par exemple, la distance entre i et j).

On désire déterminer les quantités du produit à transporter de i à j de manière à satisfaire les demandes tout en minimisant le coût total des transports.

Problème de l'affectation

Répartition optimale de tâches: Il y a n personnes disponibles et n tâches à accomplir. A chaque personne, on affecte exactement une tâche. On souhaite que toutes les tâches soient affectées. Le coût de la réalisation de la tâche j par la personne i est de c_{ij}. On cherche une affectation des tâches qui minimise le coût total.

Ce problème est un problème de transport avec un graphe biparti.

Problème du plus court chemin

La longueur du plus court chemin entre deux noeuds dans un graphe peut être obtenue en résolvant un problème de flot de coût minimum.

Soit $G = (V, E)$ un graphe. Nous cherchons la longueur du plus court chemin entre les noeuds k et l. Posons $b_k = 1$, $b_l = -1$ et $b_i = 0$ pour $i \neq k, l$. Définissons des coû unitaires égaux à 1 pour tous les arcs. Le coût du flot de coût minimum est alors égal à la longueur du plus court chemin de k à l.

Complexité

Si les données sont entières, pour les problèmes:

- ◇ Problème du flot de coût minimum;
- ◇ Problème du transport;
- ◇ Problème de l'affectation;
- ◇ Problème du plus court chemin;
- ◇ Problème du flot maximum;

il y a toujours une solution optimale du problème relaxé qui est entière.

Ce n'est par contre pas toujours le cas des problèmes:

- ◇ Problème du sac à dos;
- ◇ Problème de partitionnement;
- ◇ Problème de recouvrement;
- ◇ Problème de voyageur de commerce; etc.

Que faire pour ceux-là?

Bornes supérieures et inférieures

Soit le problème en nombres entiers

$$\begin{aligned} \max_x \quad & c^T x \\ \text{tel que} \quad & Ax \le b, \\ & x \ge 0 \text{ et } x \text{ entier.} \end{aligned}$$

Toute solution admissible $x^\dagger$ fournit une borne inférieure de l'optimum: $\underline{f} = c^T x^\dagger \le f^*$.

Soit le problème relaxé

$$\begin{aligned} \max_x \quad & c^T x \\ \text{tel que} \quad & Ax \le b, \\ & x \ge 0. \end{aligned}$$

Toute solution optimale x^* fournit une borne supérieure de l'optimum: $\bar{f} = c^T x^* \ge f^*$.

Diviser pour conquérir

Soit $z = \max_{x \in S} c^T x$. Pour trouver la solution on divise l'ensemble S en deux sous-ensembles $S = S_1 \cup S_2$, et on cherche à résoudre les problèmes distincts:

$$z_1 = \max_{x \in S_1} c^T x \quad \text{et} \quad z_2 = \max_{x \in S_2} c^T x.$$

Comme $S = S_1 \cup S_2$, on a $z = \max(z_1, z_2)$.

En général on ne dispose pas d'expressions exactes pour z_1 et z_2 mais bien de bornes supérieures et inférieures: $\underline{z_1} \leq z_1 \leq \bar{z_1}$ et $\underline{z_2} \leq z_2 \leq \bar{z_2}$.

Les bornes inférieures s'obtiennent au moyen de solutions admissibles, les bornes supérieures s'obtiennent par relaxation.

Comme $z = \max(z_1, z_2)$, nous avons

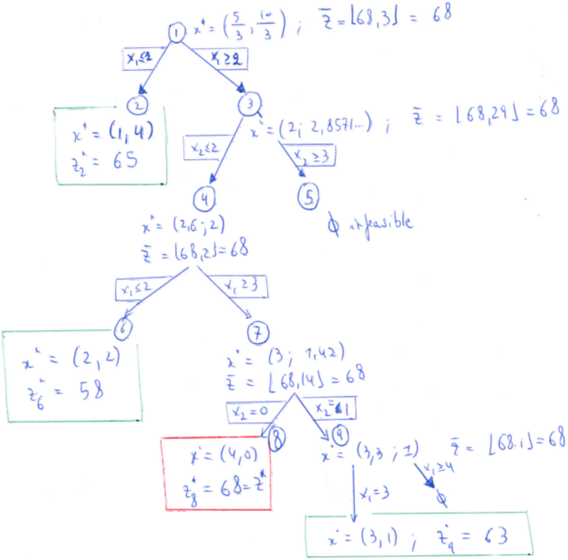
$$\max(\underline{z_1}, \underline{z_2}) \leq z \leq \max(\bar{z_1}, \bar{z_2}).$$

Branch and Bound: Exemple

Comment résoudre le problème

$$\begin{aligned} \max_{x} \quad & 17x_1 + 12x_2 \\ \text{tel que} \quad & 10x_1 + 7x_2 \leq 40, \\ & x_1 + x_2 \leq 5, \\ & x_1, x_2 \geq 0 \text{ et entiers?} \end{aligned}$$

Branch and Bound: Exemple



Branch and Bound

Le processus peut être poursuivi et nous conduit à la construction d'un arbre d'énumération.

Le processus de construction de l'arbre ne doit pas être poursuivi en tous les noeuds. Plusieurs situations donnent lieu à des simplifications.

Optimalité. Si pour un noeud donné on a $\underline{z} = \bar{z}$, le coût optimal pour le noeud est connu et le processus de division s'arrête.

Sous-optimalité. Si au noeud X on obtient une borne supérieure qui est inférieure à la borne inférieure en un autre noeud, alors le processus s'arrête au noeud X.

Ensemble admissible vide. Après décomposition d'un ensemble S en deux ensembles, il se peut qu'un des ensembles soit vide. Le processus s'arrête au noeud d'ensemble associé vide.

Construction de l'arbre

Comment construire l'arbre?

1. Comment choisir les décompositions $S = S_1 \cup S_2$?

   Une stratégie simple consiste à choisir la variable dont la partie fractionnaire est la plus proche de $1/2$.

2. Dans quel ordre choisir les noeuds a explorer?

Choix de l'ordre des noeuds

Recherche en largeur. Résolution de tous les problèmes d'un niveau donné avant de poursuivre.

Recherche en profondeur. On descend le plus rapidement possible dans l'arbre afin d'obtenir une solution admissible. Il y a un intérêt évident à obtenir rapidement une solution admissible puisqu'une telle solution permet d'élaguer toutes les branches dont les bornes supérieures sont faibles. Par ailleurs, une solution entière admissible nous permet d'arrêter le processus à n'importe quel moment et de proposer une solution admissible. Enfin, dans une recherche en profondeur, les programmes linéaires successifs sont obtenus en ajoutant ou en affinant les bornes pour une variable. La méthode du dual simplexe est particulièrement efficace pour traiter ce genre de modification.

Stratégie du meilleur noeud. Choisir un noeud pour lequel la borne supérieure est la plus élevée. De cette manière on ne développe jamais un noeud dont la borne supérieure est inférieure à l'optimum.

Exemple: problème du sac à dos

On dispose d'un sac capable de supporter un poids maximal de 8kg et d'un ensemble d'objets ayant chacun un poids (en kg) et une valeur (en euro). L'objectif est de remplir le sac d'objets sans dépasser le poids maximal tout en maximisant la somme des valeurs des objets qu'il contient. Vous disposez de 5 objets de poids respectifs 4, 3, 3, 2 et 2 kg et de valeurs respectives 3, 5, 2, 2 et 3 €.

On introduit une variables binaire par objet ($x_i = 1$ si on prend l'objet i, sinon $x_i = 0$):

$$\begin{aligned} \max_{x \in \mathbb{R}^5} \quad & 3x_1 + 5x_2 + 2x_3 + 2x_4 + 3x_5 \\ \text{tel que} \quad & 4x_1 + 3x_2 + 3x_3 + 2x_4 + 2x_5 \leq 8, \\ & x_i \in \{0, 1\} \text{ pour } 1 \leq i \leq 5. \end{aligned}$$

Relaxation linéaire (exemple)

On relache $x_i \in \{0, 1\}$ par $0 \le x_i \le 1$ pour tout i :

$$\begin{aligned} f^* \; = \; & \max_{x \in \mathbb{R}^5} && 3x_1 + 5x_2 + 2x_3 + 2x_4 + 3x_5 \\ & \text{tel que} && 4x_1 + 3x_2 + 3x_3 + 2x_4 + 2x_5 \le 8, \\ & && 0 \le x_i \le 1 \text{ pour } 1 \le i \le 5. \end{aligned}$$

Pour calculer la solution, il faut classer les objets par ordre de préférence. Cet ordre est donné par la valeur par unité de poids: $2 \succ 5 \succ 4 \succ 1 \succ 3$. En effet, on a

$$\frac{5}{3} \ge \frac{3}{2} \ge \frac{2}{2} \ge \frac{3}{4} \ge \frac{2}{3}\,.$$

On a donc d'abord intérêt à prendre l'objet 2, puis 5, puis 4 puis 1 puis 3. On rempli le sac dans cet ordre jusqu'a ce qu'il soit rempli: la solution du problème relaché est

$$x^* = (1/4, 1, 0, 1, 1).$$

Branch and Bound (exemple)

On a la solution du problème relaché qui est

$$x^* = (1/4, 1, 0, 1, 1),$$

avec $f(x^*) = \bar{f} = 10.75$. Cela implique que la valeur optimale f^* du problème en nombres binaires sera plus petite (car le problème est plus contraint, et que l'on maximise): $\bar{f} = 10.75 \geq f^*$. De plus, comme f^* doit être entier, on sait en fait que $f^* \leq 10$.
On peut maintenant commencer la procédure de Branch and Bound: on résoud deux relaxations linéaires afin de se 'débarasser' de la variables x_1^* non entière:

- Noeud $x_1 = 0$: solution optimale (0 , 1 , 1/3 , 1 , 1) avec $f^* = 10.66$
  - Noeud $x_1 = 0, x_3 = 0$: (0,1,0,1,1) avec $f^* = 10 \Rightarrow$ solution optimale.
  - Noeud $x_1 = 0, x_3 = 1$: (0,1,1,0,1) avec $f^* = 10 \Rightarrow$ solution optimale.
- Noeud $x_1 = 1$: solution optimale (1 , 1 , 0 , 0 , 1/2) avec $f^* = 9.5$: on peut stopper, on a déjà trouvé une solution strictement meilleure.

Exercice

Vous disposez d'une somme de 96 Euros. Vous désirez acheter des boissons, disponibles en divers conditionnements.

Boisson	A	B	C	D	E
Volume unitaire (litres)	15	16	17	18	19
Prix unitaire (Euros)	16	17	18	19	20

Vous désirez acheter un volume le plus grand possible, sans dépasser votre budget.
(a) Formulez ce problème comme un problème d'optimisation linéaire en nombres entiers.
(b) Combien de conditionnements de chaque type allez-vous acheter ? Détaillez votre raisonnement.