Méthode de Branch and Bound

I Introduction

En pratique, il arrive fréquemment que dans un programme linéaire, certaines variables soient astreintes à être entières. Une entreprise ne saurait construire $x_1 = 1{,}45$ entrepôts, acquérir $x_2 = 2{,}37$ camions ou encore affréter $x_3 = 0{,}41$ avion... comme pourrait lui indiquer la solution d'un Programme Linéaire en variables continues !

Il existe des modèles classiques importants pour les applications en logistique comme dans les plannings de transports de personnes (autobus, métros, trains, avions). On va en citer quelques-uns.

1 Exemples

Exemple 1. Un camion peut transporter une charge maximale de $b = 14$ tonnes, de $n = 4$ marchandises différentes. Le poids de chacune des marchandises est respectivement de 4, 6, 8 et 10 tonnes. Enfin, le bénéfice de la vente de chacune des marchandises, après son transport, vaut respectivement 2000, 2700, 3600 et 4400 euros. Déterminer le chargement du camion permettant de maximiser le bénéfice.
Le problème s'écrit

$$\left\{\begin{array}{lccccccccc} \max z = & 2000x_1 & + & 2700x_2 & + & 3600x_3 & + & 4400x_4 & & \\ s.c. & 4x_1 & + & 6x_2 & + & 8x_3 & + & 10x_4 & \leqslant & 14 \\ & x_1 & , & x_2 & , & x_3 & , & x_4 & = & 0 \text{ ou } 1 \end{array}\right.$$

Exemple 2. Considérons n villes v_0, $v_1, \cdots, v_n$ et les distances inter-villes $d_{i,j}$, $i, j \in \{0, \cdots, n\}$.
Le problème du voyageur de commerce consiste à donner une tournée allant et revenant à la ville v_0 en passant une fois par chacune des villes.
Une solution de ce problème s'écrit donc sous la forme d'une liste de villes données dans l'ordre de la tournée, que l'on peut noter $\sigma(1), \cdots, \sigma(n)$ où σ est une permutation des n villes.
On a donc clairement $n!$ solutions possibles. L'explosion combinatoire d'une énumération la rend difficilement envisageable dans tous les cas (voir exercice).

Exemple 3. Un établissement de restauration spécialisé en "produits de la mer" propose 2 types d'assiettes-repas :
a) 5 oursins, 2 douzaines de moules, 1 huitre en nombre x_1 : bénéfice 8 euros pour le restaurant ;
b) 3 oursins, 3 douzaines de moules, 3 huitres en nombre x_2 : bénéfice 6 euros pour le restaurant.
L'approvisionnement du restaurant est de 30 oursins, 24 douzaines de moules, 18 huitres. On cherche à réaliser le bénéfice maximum :

$$\left\{\begin{array}{lccccl} \max z = & 8x_1 & + & 6x_2 & & \\ s.c. & 5x_1 & + & 3x_2 & \leqslant & 30 \text{ (oursins) (1)} \\ & 2x_1 & + & 3x_2 & \leqslant & 24 \text{ (moules) (2)} \\ & x_1 & + & 3x_2 & \leqslant & 18 \text{ (huitres) (3)} \\ & x_1 & , & x_2 & \in & \mathbb{N} \end{array}\right.$$

2 DÉFINITION D'UN PROGRAMME LINÉAIRE EN NOMBRES ENTIERS

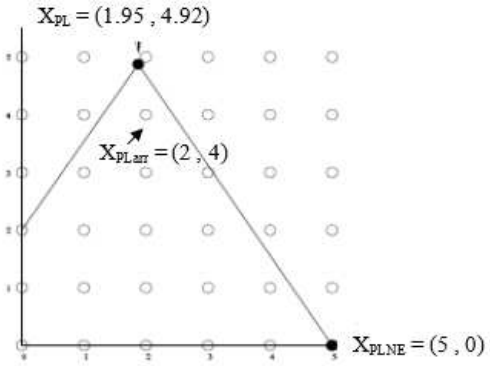
On considère le programme suivant ($PLNE$) :

$$\begin{cases} \min z = & \langle c,x \rangle \\ s.c. & Ax = b \\ & x_j \in \mathbb{N} \text{ pour } j = 1,\cdots,n \end{cases}$$

où A est une matrice de taille $m \times n$ et ses coefficients $a_{i,j}$ sont entiers, c est un vecteur de taille $n \times 1$ et les c_j sont entiers, b est un vecteur de taille $m \times 1$ et les b_i sont entiers.

Une des premières idées que l'on pourrait avoir serait de d'abord résoudre le programme linéaire continu correspondant ($x \in \mathbb{N}^n \rightarrow x \in \mathbb{R}^n$), puis de prendre la valeur entière la plus proche, un arrondi de la solution optimale non entière que l'on aurait trouvé. Malheureusement, cette solution arrondie ne correspond que dans certains cas à la solution optimale. Elle peut même être très éloignée de la solution optimale du $PLNE$, comme on peut le voir dans l'exemple suivant :

$$(PLNE) \begin{cases} \max z = & x_1 + 0{,}64x_2 \\ s.c. & 50x_1 + 31x_2 \leqslant 250 \\ & 3x_1 - x_2 \geqslant -4 \\ & x_1, x_2 \in \mathbb{N} \end{cases}$$



Solution optimale du PL : $x_{PL} = (1.95,4.92)$, $z_{PL} = 5.098$
Solution Arrondie : $x_{PLarr} = (2,4)$, $z_{PLarr} = 4.56$
Solution optimale du $PLNE$: $x_{PLNE} = (5,0)$, $z_{PLNE} = 5$.

Néanmoins, une propriété importante est mise en évidence.
Soit :
- z_{PLNE} : valeur de la fonction objectif de la solution optimale du $PLNE$;
- z_{PL} : valeur de la fonction objectif de la solution optimale du PL ;
- z_{PLarr} : valeur de la fonction obtjectif de la solution arrondie du PL.

Nous avons, pour le maximum, $z_{PLarr} \leqslant z_{PLNE} \leqslant z_{PL}$.
Ainsi, la solution optimale réelle z_{PL} représente une borne supérieure par rapport à la solution optimale du $PLNE$ z_{PLNE} dans le cas du maximum.

Remarque :
Lorsqu'on arrondit, il faut que la solution reste réalisable.

II LA MÉTHODE DE RECHERCHE ARBORESCENTE BRANCH AND BOUND

1 PRINCIPE GÉNÉRAL

La méthode que l'on va voir pour résoudre les $(PLNE)$ est la méthode de recherches arborescentes par "séparation et évaluation" ou méthode "Branch and Bound" ($\mathscr{B}\&\mathscr{B}$). Elle n'intervient pas seulement en programmation linéaire, mais dans tous les problèmes d'optimisation, où l'on travaille avec des arbres de décision.
On cherche à minimiser une fonction z sur un ensemble $S \subset \mathbb{Z}^n$ contenant un nombre fini de solutions.

SÉPARATION OU BRANCHEMENT

À chaque étape, on va subdiviser l'ensemble S de toutes les solutions réalisables (problème initial) en un nombre fini de sous-ensembles $S^{(i)}$, $i = 1, \cdots, p$ en veillant à ce que

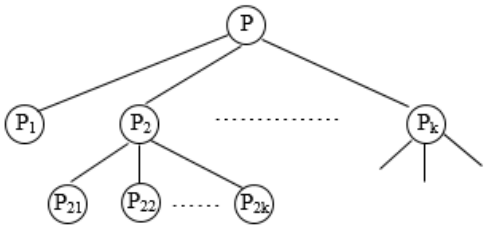
$$\bigcup_{1 \leqslant i \leqslant p} S^{(i)} = S \text{ avec } S^{(i)} \cap S^{(j)} = \emptyset \text{ pour tous } i \neq j.$$

C'est la phase "séparation" (Branch).
De cette façon, le problème se réduit à l'étude des p problèmes P_i plus restreints :

$$(P_i) \begin{cases} \min & z(x) \\ s.c. & x \in S^{(i)} \end{cases}$$

Ces subdivisions successives sont représentées à l'aide d'une arborescence
- le sous-ensemble de niveau zéro (la "racine" de l'arborescence) correspond au problème initial (P) avec $S^{(0)} = S$;
- les sous-ensembles de niveau 1 sont repérés par un seul indice ;
- chaque sous-ensemble de niveau q (> 1) est repéré par q indices dont les $q-1$ premiers indiquent le sous-ensemble "parent" dont il est issu.

- Le nombre de sous-ensembles créés n'est pas nécessairement identique à chaque séparation.



Le branchement le plus courant consiste à choisir l'une des variables x_i, puis à définir un problème-fils pour chaque valeur de x_i prise dans $\{1, \cdots, k\}$. Il est souvent utile de considérer des branchements plus généraux où l'on branche sur des sous-problèmes plus larges ($x_1 \leqslant 3$; $x_1 > 3$) par exemple, ou même sur des contraintes plus complexes ($x_1 + x_2 \leqslant 5$; $x_1 + x_2 > 5$).

ÉVALUATION OU BOUND

Divers critères sont alors utilisés pour identifier les sous-ensembles qui peuvent contenir la solution optimale et les sous-ensembles qui ne doivent pas être explorés plus à fond car ils ne peuvent pas contenir la solution optimale. C'est la phase "évaluation" (Bound).

La procédure d'évaluation consiste à évaluer la valeur optimale de la fonction objectif du sous-problème P_i, plus précisément à déterminer une borne inférieure (supérieure pour un problème de maximisation) de cette

valeur. L'obtention de cette borne - appelée fonction d'évaluation - est généralement obtenue à l'aide d'une relaxation du problème P_i.

Une fonction d'évaluation est une fonction $v : S^{(i)} \mapsto v_i = v(S^{(i)})$ telle que (pour une minimisation) $v_i \leqslant \min_{x \in S^{(i)}} z(x)$.
Il est important de remarquer qu'alors cette évaluation est également une évaluation de tous les sous-problèmes issus de ce sous-problème : étant donné $S^{(i,k)} \subset S^{(i)}$, on a $v_i \leqslant v_{i,k}$.

Soit x_s une solution du problème (déterminée par une méthode heuristique/astucieuse/chanceuse quelconque, ou qu'au pire, on met à $+\infty$), et $z_s = \langle c, x_s \rangle$ le coût associé.

Si, quand on parcourt l'arbre, on a $v(S_{i,j}) > z_s$, alors $S_{i,j}$ ne peut pas contenir de meilleure solution que x_s, puisque $v(S_{i,j})$ était déjà un minorant de ce que pouvait contenir $S_{i,j}$ comme solution.

On ne parcourt donc les sommets de l'arbre que si on a $v(S_{i,j}) \leqslant z_s$, et on réduit ainsi beaucoup le nombre de sommets à examiner, ce qui est bien ce que l'on cherchait.

Le processus de séparation d'un sous-ensemble P_i s'arrête donc dans l'un des cas suivants (pour une minimisation) :
▷ lorsque la borne inférieure de P_i est supérieure à la meilleure solution trouvée jusqu'à maintenant pour le problème initial ;
▷ lorsque P_i n'admet pas de solution réalisable : on dit que ce sous-problème est *infaisable*.
▷ lorsqu'on connaît une solution optimale locale sur P_i, c'est-à-dire, pour un $PLNE$, si la meilleure solution du sous-problème est entière : il ne sert alors à rien de brancher de nouveau ce sous-problème.

On dit alors que le problème P_i est *stérilisé*.

Si un sous-problème n'est pas stérile et qu'on en connaît la meilleure solution, on parle alors de la solution associée ou optimum local de ce sous-problème.
Une borne inférieure est la plus petite des évaluations parmi tous les sous-problèmes d'un même niveau de l'arborescence. On doit attendre d'avoir évalué tous les sous-ptoblèmes d'un même niveau de l'arborescence pour avoir une nouvelle borne inférieure.

La meilleure des solutions (entières) rencontrées au cours de l'exploration constitue une borne supérieure pour le $PLNE$.

2 ALGORITHME GÉNÉRAL (CAS D'UNE MINIMISATION)

• À chaque instant, on maintient :
- une liste de sous-problèmes actifs P_i
- un coût z de la meilleure solution obtenue jusqu'à maintenant (initialisée à $+\infty$ ou à celui d'une solution initiale connue).

• À une étape typique :
- sélectionner un sous problème actif P_i
- si P_i n'est pas réalisable, le supprimer (stop branch) sinon déterminer $v_i \leqslant z_{\text{inf}}(P_i)$
- si $v_i > z$ supprimer P_i (stop branch)
- si $v_i \leqslant z$, soit résoudre P_i directement, soit créer de nouveaux sous-problèmes et les ajouter à la liste des sous-problèmes actifs. Mettre éventuellement z à jour.

3 EXEMPLE D'APPLICATION

On considère le *PLNE* suivant :

$$(P)\begin{cases} \min z = & x_1 - 2x_2 \\ s.c. & -4x_1 + 6x_2 \leqslant 9 \\ & x_1 + x_2 \leqslant 4 \\ & x_1, x_2 \in \mathbb{N} \end{cases}$$

• La solution optimale réelle obtenue graphiquement ou par le simplexe de P relaxé est $x^* = (1.5, 2.5)$ et $z^* = -3.5$.

Ceci constitue donc une borne inférieure de P : $z_{\text{inf}} = z^* = -3.5$.

L'arrondi de cette solution donne une solution réalisable (borne supérieure z_{sup}). On a $\bar{x} = (1,2)$ et $\bar{z} = z_{\text{sup}} = -3$.

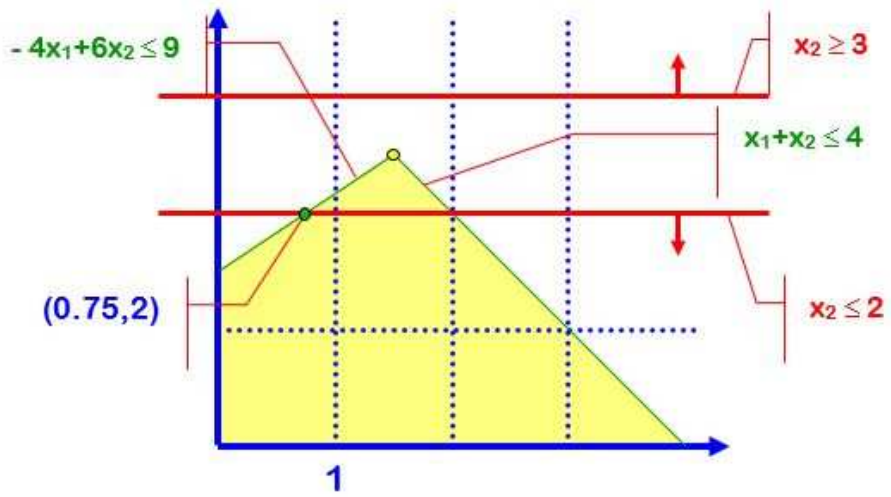
Ainsi, on peut borner la solution optimale du *PLNE* comme suit :

$$-3.5 \leqslant z_{\mathbb{R}opt} \leqslant -3.$$

• Création des sous-problèmes P_1 et P_2 en rajoutant les contraintes $x_2 \leqslant 2$ et $x_2 \geqslant 3$:

P_1		P_2	
$\min z =$	$x_1 - 2x_2$	$\min z =$	$x_1 - 2x_2$
$s.c.$	$-4x_1 + 6x_2 \leqslant 9$	$s.c.$	$-4x_1 + 6x_2 \leqslant 9$
	$x_1 + x_2 \leqslant 4$		$x_1 + x_2 \leqslant 4$
	$x_2 \geqslant 3$		$x_2 \leqslant 2$
	$x_1, x_2 \in \mathbb{N}$		$x_1, x_2 \in \mathbb{N}$

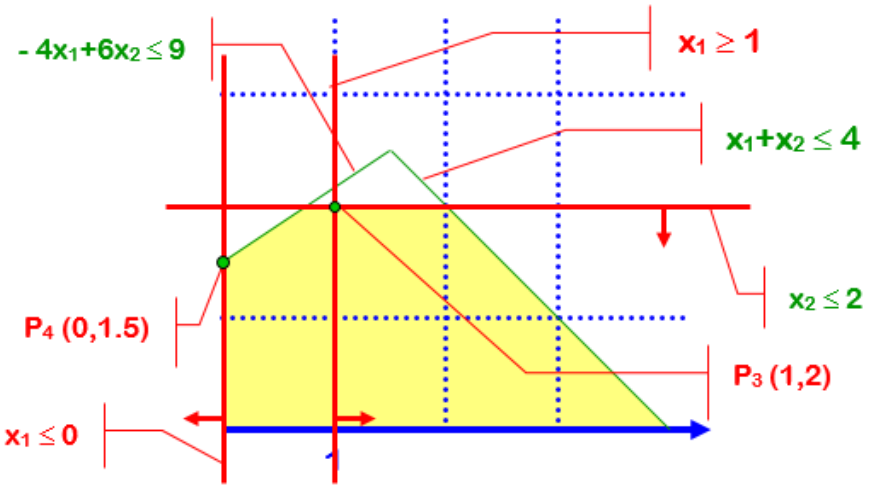
Liste des sous-problèmes actifs : $\{P_1, P_2\}$.



Le problème P_1 n'est pas réalisable : $\rightarrow$ Liste des problèmes actifs : $\{P_2\}$.

• La solution optimale de P_2 relaxé est $x^* = (0.75, 2)$ avec $z^* = -3.25$ et, la borne inférieure trouvées (-3.25) étant inférieure à la meilleure solution trouvée (-3), on crée des nouveaux sous-problèmes P_3 et P_4 en rajoutant les contraintes $x_1 \leqslant 0$ et $x_1 \geqslant 1$.

P_3		P_4	
$\min z =$	$x_1 - 2x_2$	$\min z =$	$x_1 - 2x_2$
$s.c.$	$-4x_1 + 6x_2 \leqslant 9$	$s.c.$	$-4x_1 + 6x_2 \leqslant 9$
	$x_1 + x_2 \leqslant 4$		$x_1 + x_2 \leqslant 4$
	$x_2 \leqslant 2$		$x_2 \leqslant 2$
	$x_1 \geqslant 1$		$x_1 \leqslant 0$
	$x_1, x_2 \in \mathbb{N}$		$x_1, x_2 \in \mathbb{N}$



La liste des problèmes actifs est maintenant $\{P_3, P_4\}$.

La solution optimale de P_3 relaxé est entière donc $x = (1,2)$ et $z = -3$.

Cette branche est arrêtée car z^*(borne inférieure)$\geqslant -3$.

La solution optimale est donc $\boxed{x_{PLNE} = (1,2)}$ avec $\boxed{z_{opt} = -3}$.