La programmation mathématique en nombres entiers

Résolution approchée ou exacte, algorithme de branchement/évaluation

Stanislas Francfort

Orange Labs

5 Juin 2017

Enumération, l'explosion combinatoire

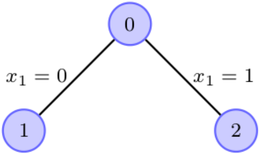
- Considérons le problème du sac à dos avec 6 objets :

$$
\begin{aligned}
Max\ & 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6 \\
& 5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14 \\
& x_1, \ldots, x_6 \in \{0, 1\}
\end{aligned}
$$

- n variables binaires
- Donc 2^n possibilités si on les énumère

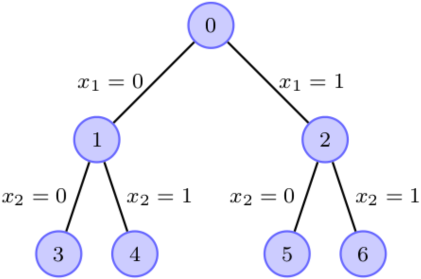
Enumération, l'explosion combinatoire

- on commence par un noeud racine où aucune variable est fixée
- puis on s'occupe de la variable x_0
- on pose deux sous-problèmes : $x_0 = 0$ et $x_0 = 1$

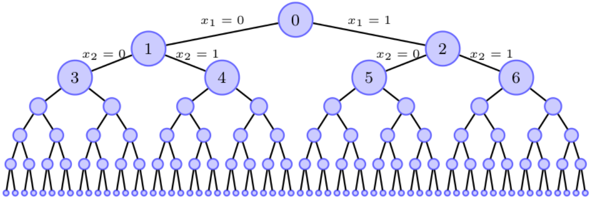


- on recommence avec les autres variables

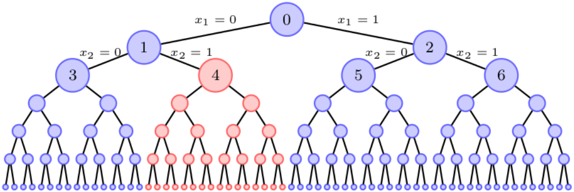
Enumération, l'explosion combinatoire



Enumération, l'explosion combinatoire



Enumération, l'explosion combinatoire



- Si on sait qu'il n'y a pas de solutions optimale dans ce sous arbre, on peut ne pas l'explorer
- But : connaitre les sous-arbres stériles le plus vite possible

Enumération, l'explosion combinatoire

- déterminer une solution $x^\star$ qui soit supérieure a toute solution d'un ensemble discret peut être fait par énumération
- **Mais** si n est la taille du problème, on a souvent $n!$ solutions.

n	2^n	Blue Gene
10	3628800	$0.000098sec$
15	1.2×10^{12}	$5.4sec$
20	2.4×10^{18}	150 jours
1000	10^{301}	14.3 milliards d'années

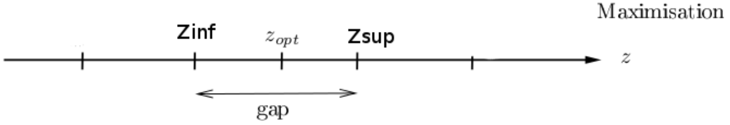
- ce phénomène d'accroissement exponentiel s'appelle l'explosion combinatoire.
- but : contourner l'explosion combinatoire par des outils mathématiques et algorithmiques.

Cas polynomiaux

- certains problèmes ont un nombre exponentiel de solutions
- mais on sait pourtant les résoudre efficacement
- problèmes de plus courts chemins dans des graphes
- problèmes de flots dans des graphes
- ...

Encadrement de la solution

- on a trouvé une solution ayant pour valeur une borne inf z_{inf} (heuristique par exemple)
- si on sait obtenir une borne sup z_{sup}
- alors on obtient un encadrement : $z_{inf} \leq z_{opt} \leq z_{sup}$



- $gap = 100 * \frac{|z_{sup} - z_{inf}|}{|z_{inf}|}$
- on connait alors la distance max de z_{inf} à z_{opt} : gap
- dans le pire des cas, $z_{opt} = z_{sup}$
- dans ce cas la solution trouvée z_{inf} est à gap% de l'optimum.

Algorithme de Branchement-Évaluation

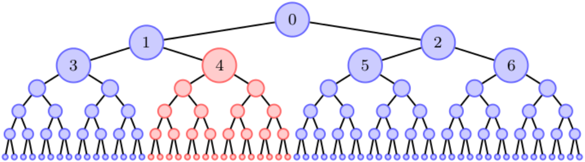
- un algorithme de branchement-évaluation utilise la méthode "diviser pour régner"
- **Branchement** : qui consiste a diviser le problème en sous-problèmes plus simples a résoudre
- **Évaluation** : nous allons éviter d'explorer tous les sous-problèmes.

Illustration : retour sur le problème du sac à dos

$$\begin{aligned} Max\ & 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6 \\ & 5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14 \\ & x_1, \dots, x_6 \in \{0, 1\} \end{aligned}$$

- Supposons qu'on ait obtenue la solution : $\hat{x} = (1, 1, 0, 0, 0, 0)$
- de valeur : $\hat{v} = 38$
- Quel sous-arbre cela nous donne-il ?

Illustration : retour sur le problème du sac à dos



- examinons maintenant le sous-arbre du noeud $(0, 1, *, *, *, *)$

$$
\begin{aligned}
Max\ & 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6 \\
& 5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14 \\
& x_0 = 0,\ \ x_1 = 1 \\
& x_1, \dots, x_6 \in \{0, 1\}
\end{aligned}
$$

Illustration : retour sur le problème du sac à dos

$$\begin{aligned} Max\ & 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6 \\ & 5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14 \\ & x_0 = 0,\ x_1 = 1 \\ & x_1, \ldots, x_6 \in \{0, 1\} \end{aligned}$$

- on peut calculer une borne supérieur de ce problème
- par une relaxation linéaire pas exemple
- cela donne la valeur 44 pour ce sous-arbre
- on sait donc que la valeur optimale entière est comprise entre 38 et 44
- et si nous avions une solution réalisable de valeur 45 ?
- alors nous pourrions arrêter d'explorer ce sous-arbre

Illustration par le problème du voyageur de commerce

- Exemple : le problème du voyageur de commerce
- Soit n villes $v_0, \ldots, v_n$
- et les distances inter-villes d_{ij} pour $i, j \in \{0, \ldots, n\}$
- But : minimiser un tour qui part de v_0 et revient à v_0
- en passant une et une seule fois par chacune des villes
- une solution peut s'exprimer pas une permutation $\sigma(1), \ldots, \sigma(n)$
- on a donc $n!$ solutions possibles.

- énumération trop longue
- un algorithme de branchement/évaluation permet de ne pas parcourir toutes les solutions

Illustration par le problème du voyageur de commerce

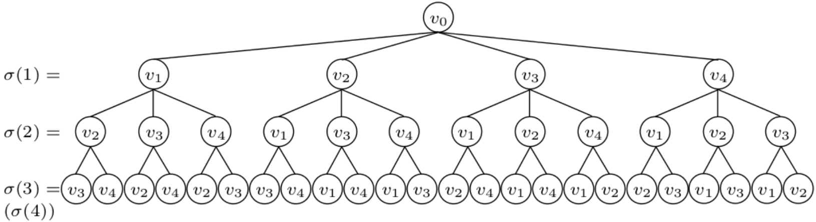
- Exemple : 5 villes v_0, v_1, v_2, v_3 et v_4 et les distances suivantes :

	v_0	v_1	v_2	v_3	v_4
v_0	-	8	4	2	3
v_1	9	-	7	1	6
v_2	3	7	-	6	6
v_3	2	1	6	-	4
v_4	7	6	6	2	-

- Pour construire une méthode énumérative, il faut choisir une façon d'énumérer les solutions

Illustration par le problème du voyageur de commerce

- énumération des solutions : représentation par un arbre



- A chaque niveau de l'arborescence en partant de la racine, on fixe quelle ville sera visitée a la position *i*

Illustration par le problème du voyageur de commerce

- une construction d'une solution : un parcourt de l'arborescence
- but : décider si l'exploration d'une **branche** est utile avant d'en énumérer toutes les solutions
- Pour cela, on va donner une **évaluation** z des solutions que peut porter une branche
- c'est-a-dire donner une borne inférieure $z_{inf}^{branche}$ des coûts des solutions de la branche
- Si $z_{inf}^{branche} \geq z_{inf}$ que l'on a déjà obtenu, on peut éviter de l'explorer.

- commençons par trouver une bonne solution
- par exemple, une heuristique gloutonne : le plus proche d'abords

Illustration par le problème du voyageur de commerce

- heuristique $\Rightarrow$ on obtient une solution réalisable
- dans notre exemple : $s_1 = \{v_0, v_3, v_1, v_4, v_2\}$
- de coût $c_1 = 18$
- on a donc une borne supérieur du problème : 18

- pouvons-nous évaluer cette solution ?
- si on obtient que la solution optimale au moins de coût $z_{opt} = 18$
- alors nous avons obtenu une solution optimale.

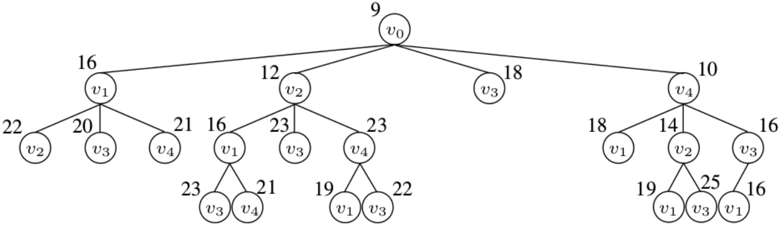
- Une idée : on doit passer par toutes les villes
- donc on doit sortir de chacune des villes
- Donc : le trajet pour aller de v_i à une autre ville est au moins la plus petite distance issue de v_i
- ici : $\textit{borne inf} = 2 + 1 + 3 + 2 + 2 = 9$
- La meilleure solution : 18, nous devons donc commencer l'énumération.

Illustration par le problème du voyageur de commerce

- première branche : on place v_1 en première place, $c(v_0 v_1) = 8$
- plus petite distance issue des sommets restants : $1, 3, 2, 2$, donc au minimum 8
- ceci donne un parcourt au minimum de coût $8 + 8 = 16$
- comme $16 < 18$, on continue
- poursuivons par v_2 : coût du parcourt $= 8 + 7 = 15$
- plus petite distance des sommets restants : $3 + 2 + 2 = 7$
- ceci donne un parcourt de $15 + 7 = 22$
- $22 \geq 18$, on arrête, aucune solution de moindre coût dans cette branche

Illustration par le problème du voyageur de commerce

- Voici le résultat



- Nous avons exploré 21 noeuds au lieu de 41
- meilleur solution : coût = 16
- une solution : v_0, v_4, v_3, v_1

Sous-problèmes, évaluation et stérilité

- On appelle *branchement* : le fait de diviser un problème en un ensemble de sous-problèmes tels que l'union de leurs espaces de solutions forme l'espace des solutions du problème-père
- On appelle les sous-problèmes obtenus des *problèmes-fils*

Sous-problèmes, évaluation et stérilité

- types de branchements fréquents :
  - choisir l'une des variables $x_i \in \{1, \ldots, k\}$
  - définir un problème-fils pour chaque valeur de x_i
- on peut avoir des branchements plus généraux $(x_1 \leq 3; x_1 > 3)$
- ou même sur des contraintes plus complexes $(x_1 + x_2 \leq 5; x_1 + x_2 > 5)$

Sous-problèmes, évaluation et stérilité

- appelle *évaluation* une borne supérieure de la valeur optimale de ce sous-problème.
- **Important** : cette évaluation est également une évaluation de tous les sous-problèmes issus de ce sous-problème
- un sous-problème est dit *stérile* dans deux cas :
  - il ne contient aucune solution (il est infaisable)
  - il est faisable et on en connaît la meilleur solution

Efficacité d'un algorithme de branchement

- l'efficacité d'un algorithme de branchement : comparaison entre des évaluations et minimums locaux et globaux
  - la meilleure des solutions valides rencontrées est une borne inférieure
  - la plus grande des évaluations d'un niveau est une borne supérieure pour le PLNE
- on stoppe l'exploration de la branche quand l'évaluation de ce sous-problème est inférieure a la meilleure solution connue pour le PLNE

Efficacité d'un algorithme de branchement

- Comment rendre cette méthode plus efficace ?
- favoriser l'apparition de sous-problèmes stériles
- Il est très utile de posséder une bonne heuristique pour le problème P
- Eliminer la symétrie, car un algorithme de branchement va devoir explorer toutes les solutions symétriques proches de l'optimum

Efficacité d'un algorithme de branchement

- Utiliser la structure pour éviter une exploration systématique
- Premier cas :
    - Si les valeurs des solutions sont liées par une récurrence
    - Programmation Dynamique
    - Exemple : Djikstra pour les plus courts chemins)
- Deuxième cas :
    - Si les solutions sont fortement liées entre elles
    - on peut détecter des "dominances" entre solutions
    - si tout sous-problème descendant d'un sous-problème a est au moins aussi bon que tout descendant d'un sous-problème b
    - alors il suffit d'explorer a
    - Exemple : quelques problèmes d'ordonnancement ou d'optimisation sur les graphes, où les solutions ont des points ou des parties en communs.

Schéma de l'algorithme

- La création de sous-problème (5) est appelé *le choix du branchement*
- dans le cas le plus simple : choisir quelle variable sera utilisée pour brancher

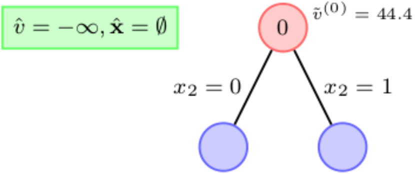
- L'ordre dans lequel sont traités les sous-problèmes stockés dans la structure L a une influence sur le fonctionnement de l'algorithme
- On parle alors de *stratégie de branchement*
- La stratégie de branchement est cruciale pour une bonne exploration
- classique : parcourir l'arborescence des sous-problèmes
    - en largeur
    - en profondeur
    - en recherchant le sous-problème non traité de meilleur évaluation

Exemple complet sur le problème du sac à dos

- On rappelle :

$$
\begin{aligned}
Max\ & 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6 \\
& 5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14 \\
& x_1, \ldots, x_6 \in \{0, 1\}
\end{aligned}
$$

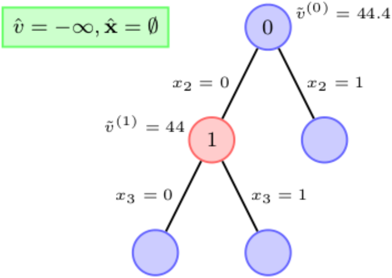
Exemple complet sur le problème du sac à dos



- Relaxation linéaire : $\tilde{v}^0 = 44.4$ et $\tilde{x}^0 = (1, 0.43, 0, 0, 0, 1)$

Exemple complet sur le problème du sac à dos

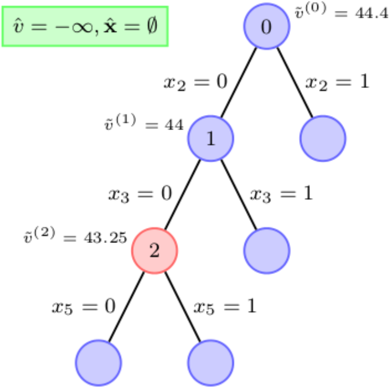
- on branche sur x_2 (on le fixe à 0)



- Relaxation linéaire : $\tilde{v}^1 = 44$ et $\tilde{x}^1 = (1, 0, 0.75, 0, 0, 1)$

Exemple complet sur le problème du sac à dos

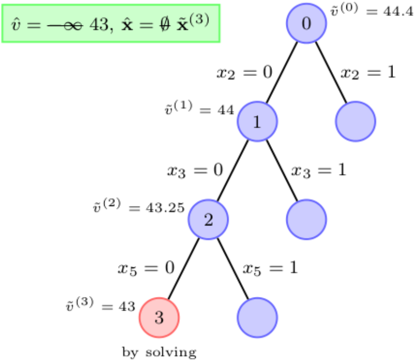
- on branche sur x_3



- Relaxation linéaire : $\tilde{v}^2 = 43.25$ et $\tilde{x}^2 = (1, 0, 0, 0, 0.75, 1)$

Exemple complet sur le problème du sac à dos

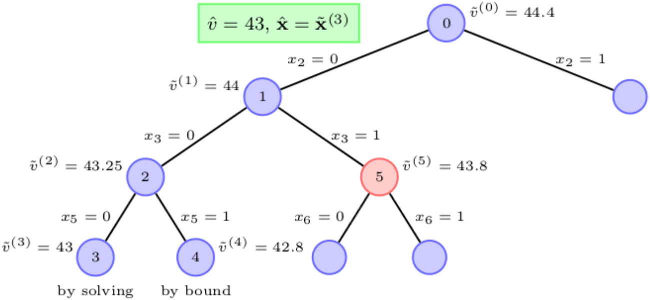
- on branche sur x_5



- Relaxation linéaire : $\tilde{v}^3 = 43$ et $\tilde{x}^3 = (1, 0, 0, 1, 0, 1)$

Exemple complet sur le problème du sac à dos

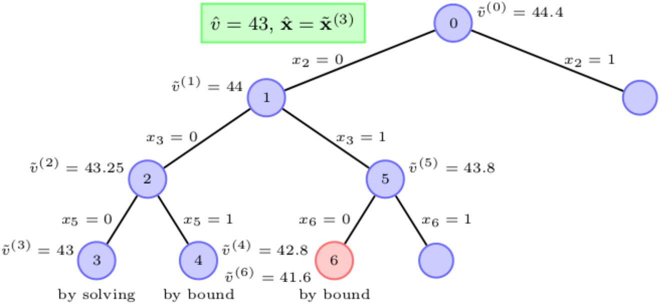
- remplacer le solution réalisable obtenue
- explorer l'autre branchement de x_5



- Relaxation linéaire : $\tilde{v}^5 = 43.8$ et $\tilde{x}^5 = (1, 0, 1, 0, 0, 0.83)$

Exemple complet sur le problème du sac à dos

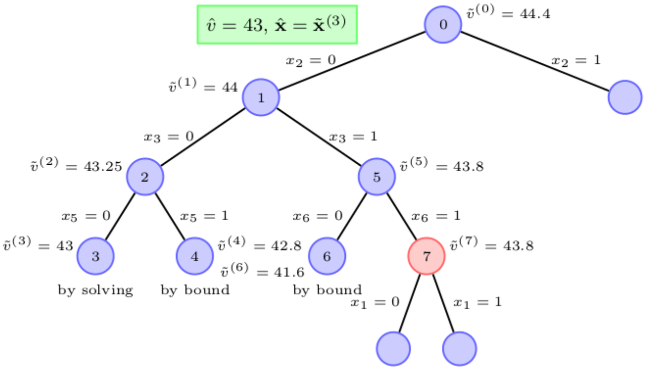
- on branche sur x_6



- Relaxation linéaire : $\tilde{v}^6 = 41.6$ et $\tilde{x}^6 = (1, 0, 1, 0.33, 1, 0)$

Exemple complet sur le problème du sac à dos

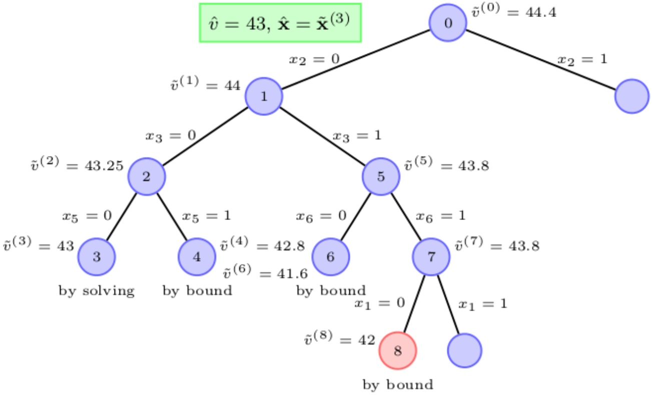
- on termine l'exploration du noeud 6



- Relaxation linéaire : $\tilde{v}^7 = 43.8$ et $\tilde{x}^7 = (0.8, 0, 1, 0, 0, 1)$

Exemple complet sur le problème du sac à dos

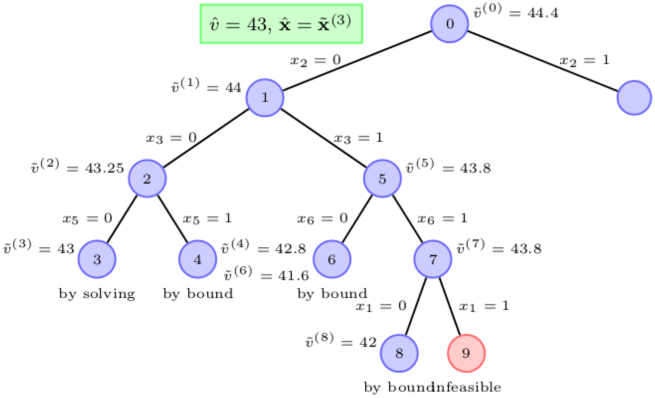
- on branche sur x_1



- Relaxation linéaire : $\tilde{v}^8 = 42$ et $\tilde{x}^8 = (0, 0, 1, 0, 1, 1)$

Exemple complet sur le problème du sac à dos

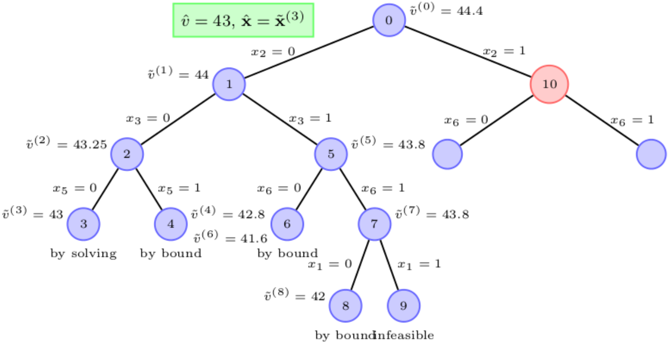
- on stoppe l'exploration du noeud 8



- Relaxation linéaire : pas de solution réalisable ! !

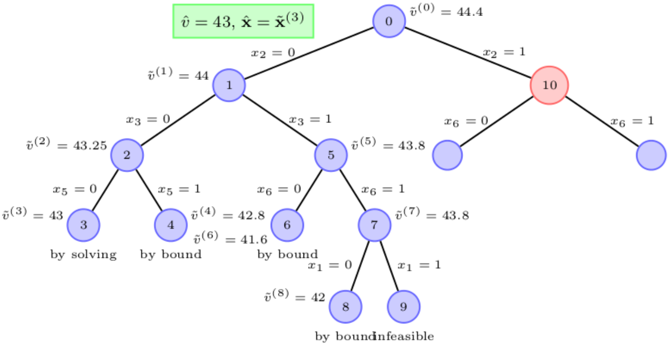
Exemple complet sur le problème du sac à dos

- on stoppe l'exploration du noeud 9



- Relaxation linéaire : $\tilde{v}^{10} = 44.3$ et $\tilde{x}^{10} = (1, 1, 00, 0, 0.33)$
- on branche sur x_6
- et on continue ainsi de suite.... jusqu'à ce qu'il n'y ait plus de noeud disponibles

Exemple complet sur le problème du sac à dos



- on a résolu 29 relaxations linéaires
- on a trouvé la solution optimale à l'itération 3
- mais on n'a pas pu conclure avant...