

Optimisation linéaire

Romain DUJOL, Jean-Paul FOREST

CY TECH – INGI-IN



2022 – 2023

OLNE – Séparation et évaluation (« Branch-and-bound »)

Séparation (« branch »)

Évaluation (« bound »)

Commençons par un exemple...

Règles de choix

Problème à traiter

Variable de séparation

Exemple complet

« Branch-and-cut »

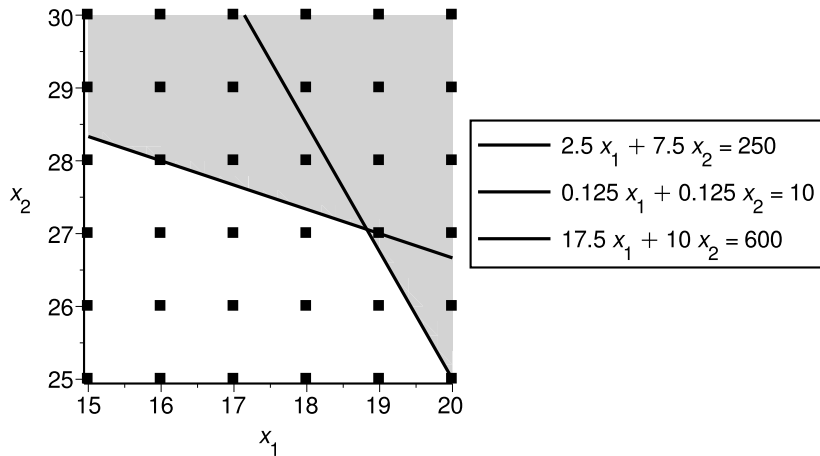
$$(P) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \in \mathbb{N}^n \end{cases} \quad (\bar{P}) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \geq 0 \end{cases}$$

Intérêt du problème relaxé (\bar{P})

- ▶ Si un optimum \bar{x} de (\bar{P}) est entier, c'est un optimum de (P) .
- ▶ Sinon, (\bar{P}) est plus facile à résoudre que (P) .
- ▶ Pour tout optimum x^* de (P) , $(c|x^*) \leq (c|\bar{x})$.

Comme pour la méthode des coupes, on cherche à se ramener à la résolution de problèmes d'optimisation linéaire en variables réelles.

Problème en nombres entiers et problème relaxé



Séparation (« branch »)

$$\begin{array}{ccc}
 (P) \left\{ \begin{array}{l} \max_x (c|x) \\ Ax \leq b, x \in \mathbb{N}^n \end{array} \right. & & \\
 \swarrow & & \searrow \\
 (P_1) \left\{ \begin{array}{l} \max_x (c|x) \\ A_1x \leq b_1, x \in \mathbb{N}^n \end{array} \right. & & (P_2) \left\{ \begin{array}{l} \max_x (c|x) \\ A_2x \leq b_2, x \in \mathbb{N}^n \end{array} \right.
 \end{array}$$

Objectif

► Convergence

Le domaine du problème relaxé de (P) contient strictement la réunion des deux sous-domaines des problèmes relaxés de (P_1) et (P_2)

$$\{Ax \leq b, x \geq 0\} \supsetneq \{A_1x \leq b_1, x \geq 0\} \cup \{x, A_2x \leq b_2, x \geq 0\}$$

► Exhaustivité

Le domaine du problème de (P) est la réunion des deux sous-domaines des problèmes (P_1) et (P_2) .

$$\{Ax \leq b, x \in \mathbb{N}^n\} = \{A_1x \leq b_1, x \in \mathbb{N}^n\} \cup \{A_2x \leq b_2, x \in \mathbb{N}^n\}$$

$$(P) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \in \mathbb{N}^n \end{cases}$$

$$(\bar{P}) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \geq 0 \end{cases}$$

Méthode

1. On calcule un optimum \bar{x} du problème relaxé (\bar{P}) .

$$(P) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \in \mathbb{N}^n \end{cases} \quad (\bar{P}) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \geq 0 \end{cases}$$

Méthode

1. On calcule un optimum \bar{x} du problème relaxé (\bar{P}) .
Si $\bar{x} \in \mathbb{N}^n$, c'est un optimum de (P) .

$$(P) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \in \mathbb{N}^n \end{cases} \quad (\bar{P}) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \geq 0 \end{cases}$$

Méthode

1. On calcule un optimum \bar{x} du problème relaxé (\bar{P}) .
Si $\bar{x} \in \mathbb{N}^n$, c'est un optimum de (P) .
2. Sinon, on choisit $i \in \llbracket 1, n \rrbracket$ tel que $\bar{x}_i \notin \mathbb{N}$.

$$(P) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \in \mathbb{N}^n \end{cases} \quad (\bar{P}) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \geq 0 \end{cases}$$

Méthode

1. On calcule un optimum \bar{x} du problème relaxé (\bar{P}) .
Si $\bar{x} \in \mathbb{N}^n$, c'est un optimum de (P) .
2. Sinon, on choisit $i \in \llbracket 1, n \rrbracket$ tel que $\bar{x}_i \notin \mathbb{N}$.
3. On construit les sous-problèmes :

$$(P_-) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x_i \leq \lfloor \bar{x}_i \rfloor, x \in \mathbb{N}^n \end{cases}$$

$$(P_+) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x_i \geq \lfloor \bar{x}_i \rfloor + 1, x \in \mathbb{N}^n \end{cases}$$

$$(P) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \in \mathbb{N}^n \end{cases} \quad (\bar{P}) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \geq 0 \end{cases}$$

Méthode

1. On calcule un optimum \bar{x} du problème relaxé (\bar{P}) .
Si $\bar{x} \in \mathbb{N}^n$, c'est un optimum de (P) .
2. Sinon, on choisit $i \in \llbracket 1, n \rrbracket$ tel que $\bar{x}_i \notin \mathbb{N}$.
3. On construit les sous-problèmes :

$$(P_-) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x_i \leq \lfloor \bar{x}_i \rfloor, x \in \mathbb{N}^n \end{cases}$$

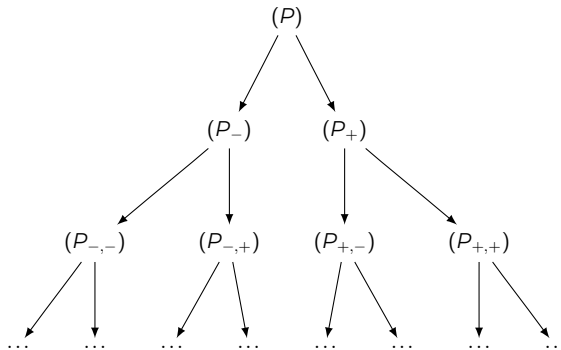
$$(P_+) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x_i \geq \lfloor \bar{x}_i \rfloor + 1, x \in \mathbb{N}^n \end{cases}$$

Remarque

x^- optimum de $(P_-) \Rightarrow (c|x^-) \leq (c|\bar{x})$

x^+ optimum de $(P_+) \Rightarrow (c|x^+) \leq (c|\bar{x})$

Arborescence des séparations



Les séparations peuvent se faire sur des variables différentes :

- ▶ d'un niveau à l'autre;
- ▶ au même niveau.

$$(P) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \in \mathbb{N}^n \end{cases} \quad (\bar{P}) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \geq 0 \end{cases}$$

(\bar{P}) est le problème relaxé de (P)

Quand est-il inutile d'appliquer la séparation à (P) ?

$$(P) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \in \mathbb{N}^n \end{cases} \quad (\bar{P}) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \geq 0 \end{cases}$$

(\bar{P}) est le problème relaxé de (P)

Quand est-il inutile d'appliquer la séparation à (P) ?

- lorsque le domaine des contraintes de (\bar{P}) est vide
(les séparations suivantes feraient de même)

$$(P) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \in \mathbb{N}^n \end{cases} \quad (\bar{P}) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \geq 0 \end{cases}$$

(\bar{P}) est le problème relaxé de (P)

Quand est-il inutile d'appliquer la séparation à (P) ?

- ▶ lorsque le domaine des contraintes de (\bar{P}) est vide
(les séparations suivantes feraient de même)
- ▶ lorsque l'optimum de (\bar{P}) est entier [et donc optimum de (P)]
(les séparations suivantes donneraient des optima moins bons, entiers ou non)

Évaluation (« bound »)

Évaluation (« bound »)

Commençons par un exemple...

Exemple

$$(P) \left\{ \begin{array}{l} \max \quad (3x_1 + 5x_2) \\ x_1 + 2x_2 \leq 3 \\ 6x_1 + 8x_2 \leq 15 \\ x_1, x_2 \in \mathbb{N} \end{array} \right.$$

- Optimum du problème relaxé :

$$(x_1, x_2) = (3/2, 3/4) \Rightarrow (c|x) = 33/4 = 8,25$$

Exemple

$$(P) \left\{ \begin{array}{l} \max \quad (3x_1 + 5x_2) \\ x_1 + 2x_2 \leq 3 \\ 6x_1 + 8x_2 \leq 15 \\ x_1, \quad x_2 \in \mathbb{N} \end{array} \right.$$

- ▶ Optimum du problème relaxé :
 $(x_1, x_2) = (3/2, 3/4) \Rightarrow (c|x) = 33/4 = 8,25$
- ▶ Séparation selon x_1 :

Exemple

$$(P) \begin{cases} \max & (3x_1 + 5x_2) \\ & x_1 + 2x_2 \leq 3 \\ & 6x_1 + 8x_2 \leq 15 \\ & x_1, x_2 \in \mathbb{N} \end{cases}$$

- Optimum du problème relaxé :

$$(x_1, x_2) = (3/2, 3/4) \Rightarrow (c|x) = 33/4 = 8,25$$

- Séparation selon x_1 :

$$(P^-) \begin{cases} \max & (3x_1 + 5x_2) \\ & x_1 + 2x_2 \leq 3 \\ & 6x_1 + 8x_2 \leq 15 \\ & x_1 \leq 1 \\ & x_1, x_2 \in \mathbb{N} \end{cases} \quad (P^+) \begin{cases} \max & (3x_1 + 5x_2) \\ & x_1 + 2x_2 \leq 3 \\ & 6x_1 + 8x_2 \leq 15 \\ & x_1 \geq 2 \\ & x_1, x_2 \in \mathbb{N} \end{cases}$$

Exemple

$$(P) \begin{cases} \max & (3x_1 + 5x_2) \\ & x_1 + 2x_2 \leq 3 \\ & 6x_1 + 8x_2 \leq 15 \\ & x_1, x_2 \in \mathbb{N} \end{cases}$$

- Optimum du problème relaxé :

$$(x_1, x_2) = (3/2, 3/4) \Rightarrow (c|x) = 33/4 = 8,25$$

- Séparation selon x_1 :

$$(P^-) \begin{cases} \max & (3x_1 + 5x_2) \\ & x_1 + 2x_2 \leq 3 \\ & 6x_1 + 8x_2 \leq 15 \\ & x_1 \leq 1 \\ & x_1, x_2 \in \mathbb{N} \end{cases} \quad (P^+) \begin{cases} \max & (3x_1 + 5x_2) \\ & x_1 + 2x_2 \leq 3 \\ & 6x_1 + 8x_2 \leq 15 \\ & x_1 \geq 2 \\ & x_1, x_2 \in \mathbb{N} \end{cases}$$

Optimum du problème relaxé :

$$(x_1, x_2) = (1, 1)$$

$$\Rightarrow (c|x) = 8$$

Exemple

$$(P) \begin{cases} \max & (3x_1 + 5x_2) \\ & x_1 + 2x_2 \leq 3 \\ & 6x_1 + 8x_2 \leq 15 \\ & x_1, x_2 \in \mathbb{N} \end{cases}$$

- Optimum du problème relaxé :

$$(x_1, x_2) = (3/2, 3/4) \Rightarrow (c|x) = 33/4 = 8,25$$

- Séparation selon x_1 :

$$(P^-) \begin{cases} \max & (3x_1 + 5x_2) \\ & x_1 + 2x_2 \leq 3 \\ & 6x_1 + 8x_2 \leq 15 \\ & x_1 \leq 1 \\ & x_1, x_2 \in \mathbb{N} \end{cases}$$

Optimum du problème relaxé : Optimum du problème relaxé :

$$(x_1, x_2) = (1, 1)$$

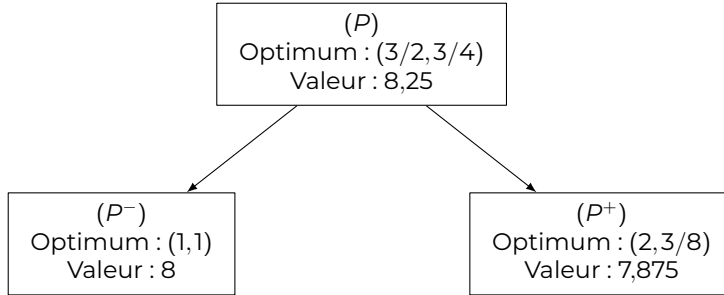
$$\Rightarrow (c|x) = 8$$

$$(P^+) \begin{cases} \max & (3x_1 + 5x_2) \\ & x_1 + 2x_2 \leq 3 \\ & 6x_1 + 8x_2 \leq 15 \\ & x_1 \geq 2 \\ & x_1, x_2 \in \mathbb{N} \end{cases}$$

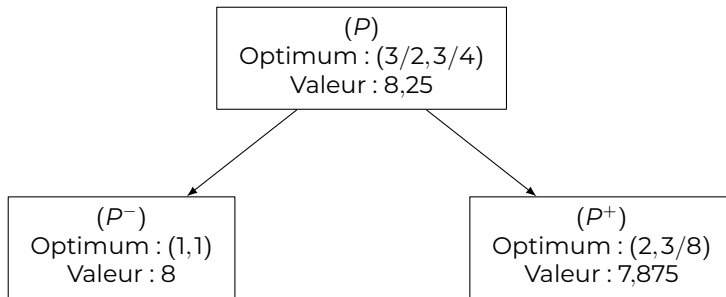
$$(x_1, x_2) = (2, 3/8)$$

$$\Rightarrow (c|x) = 63/8 = 7,875$$

Arborescence de la séparation

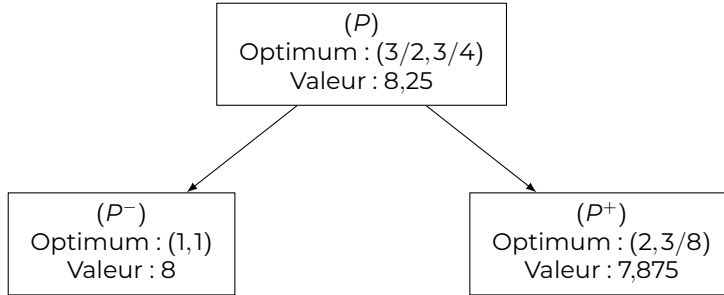


Arborescence de la séparation



Élagage par évaluation

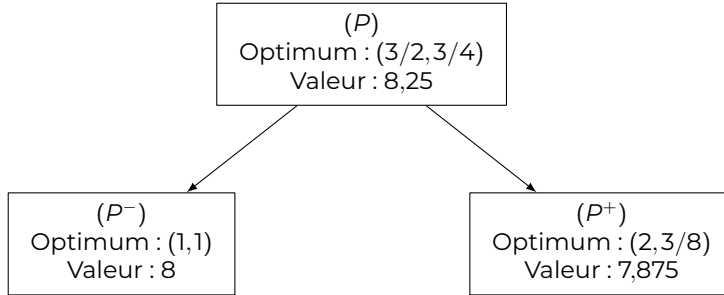
Arborescence de la séparation



Élagage par évaluation

- Toute solution issue d'une séparation de (P^+) aura une valeur strictement inférieure à 7,875... donc à 8.

Arborescence de la séparation



Élagage par évaluation

- Toute solution issue d'une séparation de (P^+) aura une valeur strictement inférieure à 7,875... donc à 8.

⇒ Inutile de séparer (P^+)

$$(P) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \in \mathbb{N}^n \end{cases} \quad (\bar{P}) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \geq 0 \end{cases}$$

(\bar{P}) est le problème relaxé de (P)

Quand est-il inutile d'appliquer la séparation à (P) ?

- ▶ lorsque le domaine des contraintes de (\bar{P}) est vide
(les séparations suivantes feraient de même)
- ▶ lorsque l'optimum de (\bar{P}) est entier [et donc optimum de (P)]
(les séparations suivantes donneraient des optima moins bons, entiers ou non)

$$(P) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \in \mathbb{N}^n \end{cases} \quad (\bar{P}) \begin{cases} \max_{x \in \mathbb{R}^n} (c|x) \\ Ax \leq b, x \geq 0 \end{cases}$$

(\bar{P}) est le problème relaxé de (P)

Quand est-il inutile d'appliquer la séparation à (P) ?

- ▶ lorsque le domaine des contraintes de (\bar{P}) est vide
(les séparations suivantes feraient de même)
- ▶ lorsque l'optimum de (\bar{P}) est entier [et donc optimum de (P)]
(les séparations suivantes donneraient des optima moins bons, entiers ou non)
- ▶ lorsque l'optimum de (\bar{P}) est moins bon que le meilleur optimum entier courant
(les séparations suivantes donneraient des optima moins bons, entiers ou non)

Algorithme général

1. On part avec le problème initial, non traité.

Algorithme général

1. On part avec le problème initial, non traité.
2. Tant qu'il existe des problèmes non traités,

Algorithme général

1. On part avec le problème initial, non traité.
2. Tant qu'il existe des problèmes non traités,
 - 2.1 on choisit un problème non traité

Algorithme général

1. On part avec le problème initial, non traité.
2. Tant qu'il existe des problèmes non traités,
 - 2.1 on choisit un problème non traité
 - 2.2 si l'ensemble des contraintes de son problème relaxé n'est pas vide, on résout le problème relaxé associé;

Algorithme général

1. On part avec le problème initial, non traité.
2. Tant qu'il existe des problèmes non traités,
 - 2.1 on choisit un problème non traité
 - 2.2 si l'ensemble des contraintes de son problème relaxé n'est pas vide, on résout le problème relaxé associé;
 - 2.3 si l'optimum obtenu est meilleur que le meilleur optimum entier courant :

Algorithme général

1. On part avec le problème initial, non traité.
2. Tant qu'il existe des problèmes non traités,
 - 2.1 on choisit un problème non traité
 - 2.2 si l'ensemble des contraintes de son problème relaxé n'est pas vide, on résout le problème relaxé associé;
 - 2.3 si l'optimum obtenu est meilleur que le meilleur optimum entier courant :
 - ▶ si l'optimum est entier, c'est le nouveau meilleur optimum entier;

Algorithme général

1. On part avec le problème initial, non traité.
2. Tant qu'il existe des problèmes non traités,
 - 2.1 on choisit un problème non traité
 - 2.2 si l'ensemble des contraintes de son problème relaxé n'est pas vide, on résout le problème relaxé associé;
 - 2.3 si l'optimum obtenu est meilleur que le meilleur optimum entier courant :
 - ▶ si l'optimum est entier, c'est le nouveau meilleur optimum entier;
 - ▶ sinon, on applique la méthode de séparation et les nouveaux problèmes obtenus sont considérés comme non traités.

Algorithme général

1. On part avec le problème initial, non traité.
2. Tant qu'il existe des problèmes non traités,
 - 2.1 on choisit un problème non traité
 - 2.2 si l'ensemble des contraintes de son problème relaxé n'est pas vide, on résout le problème relaxé associé;
 - 2.3 si l'optimum obtenu est meilleur que le meilleur optimum entier courant :
 - ▶ si l'optimum est entier, c'est le nouveau meilleur optimum entier;
 - ▶ sinon, on applique la méthode de séparation et les nouveaux problèmes obtenus sont considérés comme non traités.
 - 2.4 on considère ce problème comme traité

Algorithme général

1. On part avec le problème initial, non traité.
2. Tant qu'il existe des problèmes non traités,
 - 2.1 on choisit un problème non traité
 - 2.2 si l'ensemble des contraintes de son problème relaxé n'est pas vide, on résout le problème relaxé associé;
 - 2.3 si l'optimum obtenu est meilleur que le meilleur optimum entier courant :
 - ▶ si l'optimum est entier, c'est le nouveau meilleur optimum entier;
 - ▶ sinon, on applique la méthode de séparation et les nouveaux problèmes obtenus sont considérés comme non traités.
 - 2.4 on considère ce problème comme traité

⇒ Complexité exponentielle

Règles de choix

Règles de choix

Problème à traiter

Algorithme général

1. On part avec le problème initial, non traité.
2. Tant qu'il existe des problèmes non traités,
 - 2.1 on choisit un problème non traité
 - 2.2 si l'ensemble des contraintes de son problème relaxé n'est pas vide, on résout le problème relaxé associé;
 - 2.3 si l'optimum obtenu est meilleur que le meilleur optimum entier courant :
 - ▶ si l'optimum est entier, c'est le nouveau meilleur optimum entier;
 - ▶ sinon, on applique la méthode de séparation et les nouveaux problèmes obtenus sont considérés comme non traités.
 - 2.4 on considère ce problème comme traité

⇒ Complexité exponentielle

Lorsque plusieurs problèmes peuvent être traités...

Lorsque plusieurs problèmes peuvent être traités...

Règle de la plus grande profondeur

On choisit le problème le plus profond dans l'arborescence de séparation.

Lorsque plusieurs problèmes peuvent être traités...

Règle de la plus grande profondeur

On choisit le problème le plus profond dans l'arborescence de séparation.

Règle de la plus grande valeur

On choisit le problème dont l'optimum du problème relaxé a la plus grande valeur.

Règles de choix

Variable de séparation

Algorithme général

1. On part avec le problème initial, non traité.
2. Tant qu'il existe des problèmes non traités,
 - 2.1 on choisit un problème non traité
 - 2.2 si l'ensemble des contraintes de son problème relaxé n'est pas vide, on résout le problème relaxé associé;
 - 2.3 si l'optimum obtenu est meilleur que le meilleur optimum entier courant :
 - ▶ si l'optimum est entier, c'est le nouveau meilleur optimum entier;
 - ▶ sinon, on applique la méthode de séparation et les nouveaux problèmes obtenus sont considérés comme non traités.
 - 2.4 on considère ce problème comme traité

⇒ Complexité exponentielle

Choix de la variable de séparation

Lorsque plusieurs variables de séparation sont possibles...

Choix de la variable de séparation

Lorsque plusieurs variables de séparation sont possibles...

Règle de la plus grande distance à un entier

On choisit la variable dont la partie fractionnaire est la plus proche de 0,5.

Choix de la variable de séparation

Lorsque plusieurs variables de séparation sont possibles...

Règle de la plus grande distance à un entier

On choisit la variable dont la partie fractionnaire est la plus proche de 0,5.

Règle... aléatoire!

On choisit aléatoire une variable parmi celles qui sont non entières.

Choix de la variable de séparation

Lorsque plusieurs variables de séparation sont possibles...

Règle de la plus grande distance à un entier

On choisit la variable dont la partie fractionnaire est la plus proche de 0,5.

Règle... aléatoire!

On choisit aléatoire une variable parmi celles qui sont non entières.

Règles à base de pénalités

L'évaluation des pénalités se fait en résolvant les problèmes issus de la séparation pour toutes les variables possibles.

⇒ **Trop coûteux**

Il n'y a pas de règle universelle !

L'utilisation d'une règle plutôt qu'une autre dépend grandement du type de problème considéré.

Il n'y a pas de règle universelle !

L'utilisation d'une règle plutôt qu'une autre dépend grandement du type de problème considéré.

C'est donc une histoire d'expérience...

Exemple complet

Formulation du problème

$$(P_0) \left\{ \begin{array}{l} \max \quad (39x_1 + 69x_2) \\ 2,5x_1 + 7,5x_2 \leq 250 \\ 0,125x_1 + 0,125x_2 \leq 10 \\ 17,5x_1 + 10x_2 \leq 600 \\ x_1, x_2 \in \mathbb{N} \end{array} \right.$$

Règles utilisées

Choix du problème à traiter

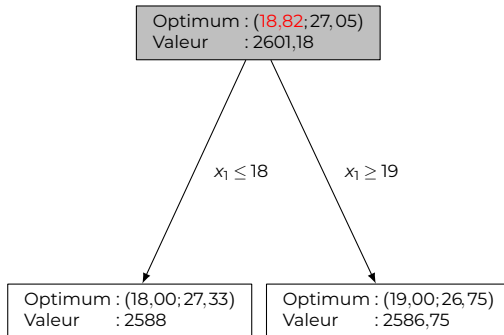
Règle de la plus grande valeur

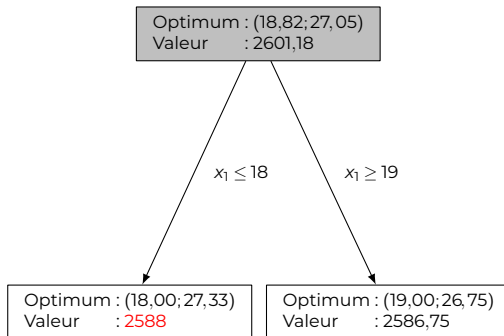
Choix de la variable de séparation

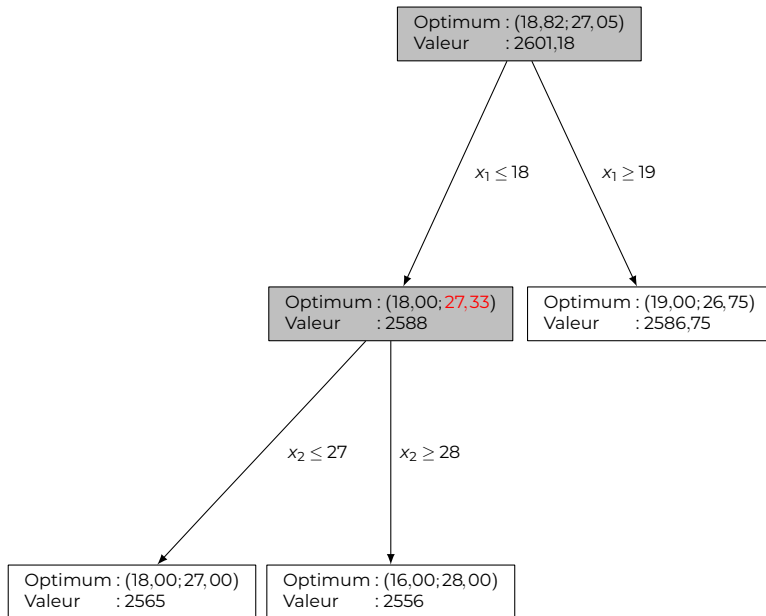
Règle de la plus grande distance à un entier

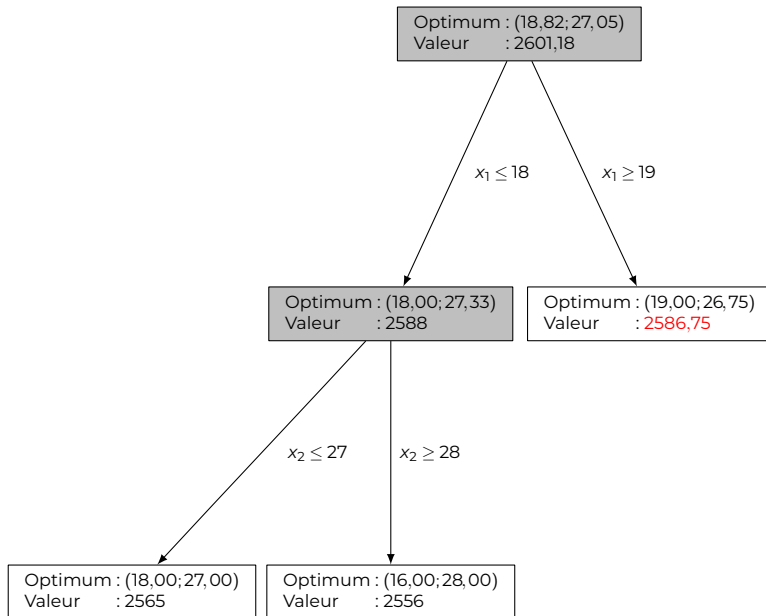
Meilleure solution entière : \emptyset

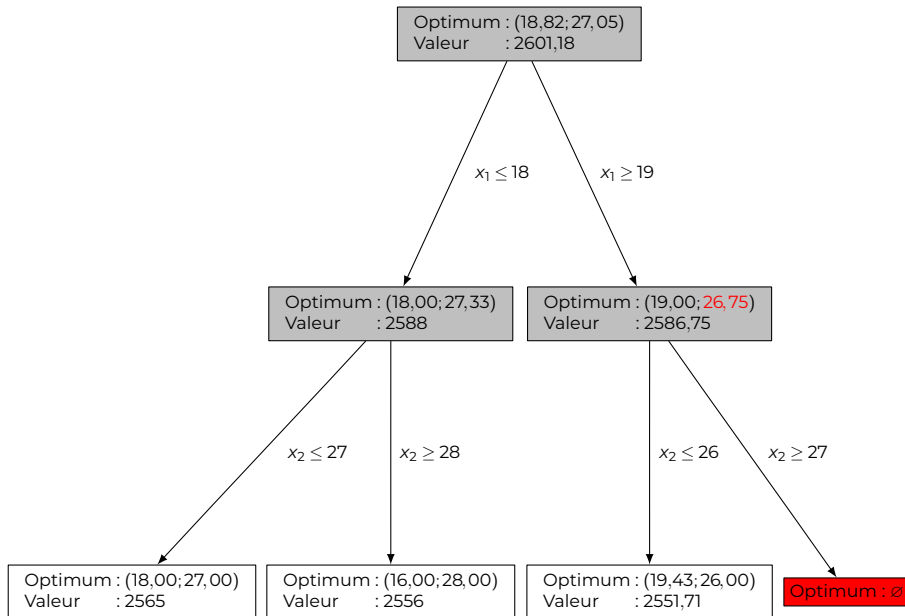
Optimum : (18,82; 27,05)
Valeur : 2601,18

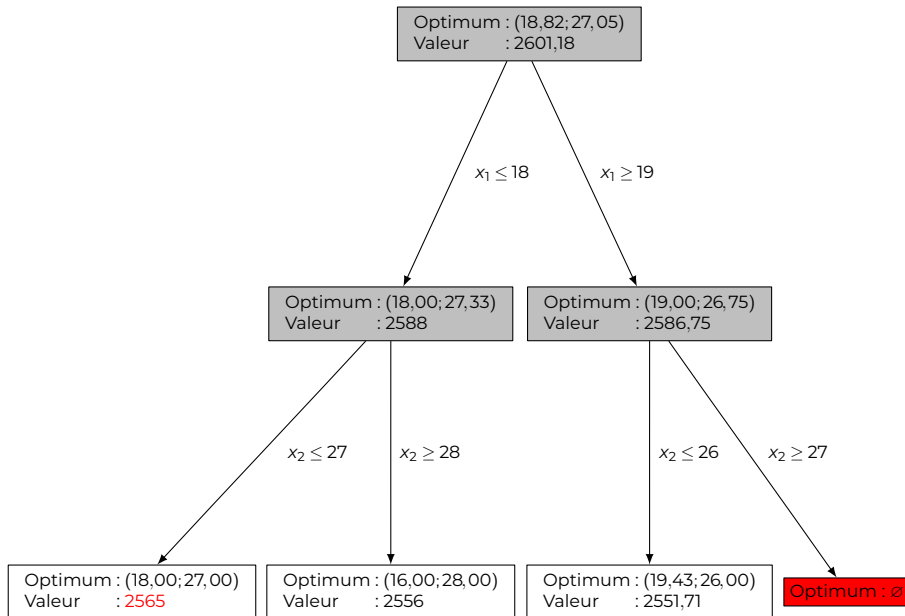




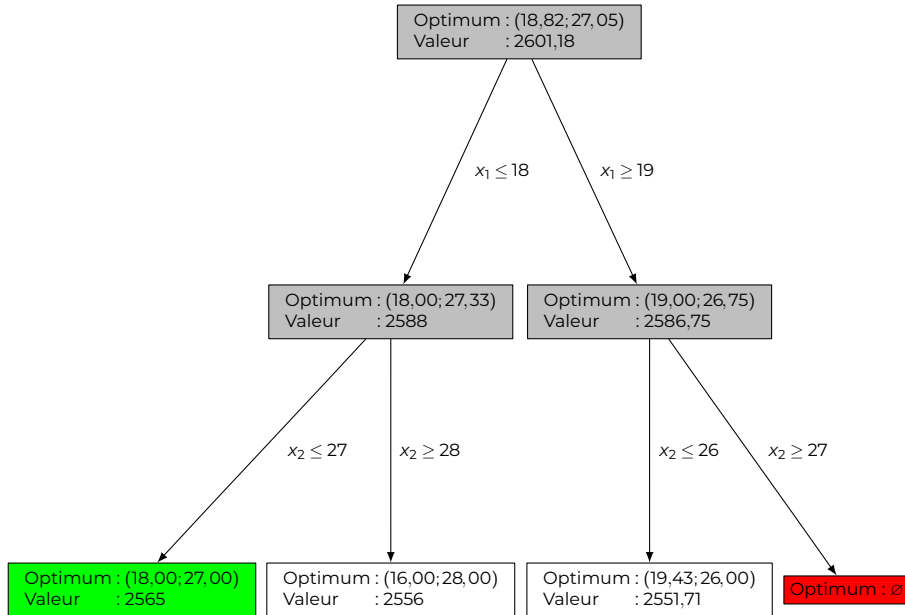




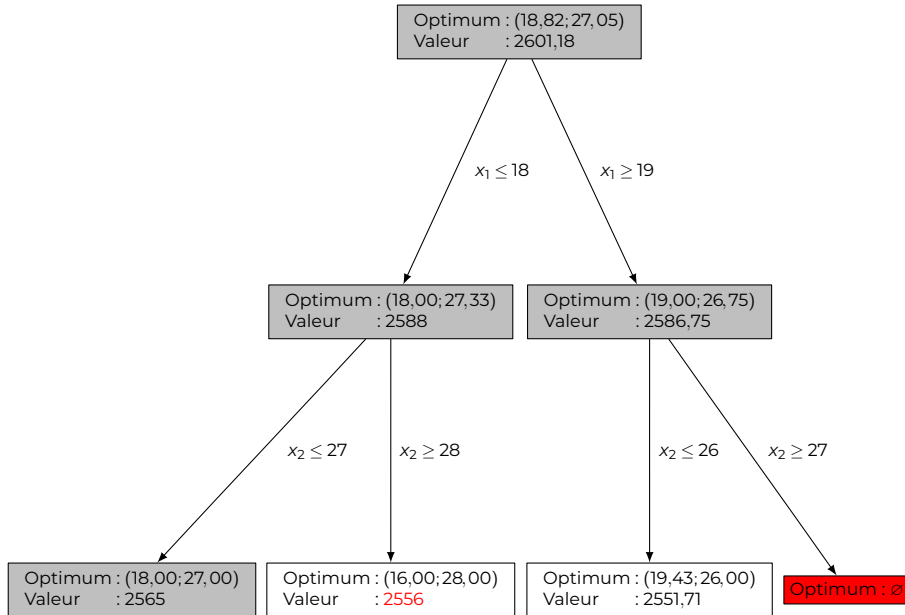




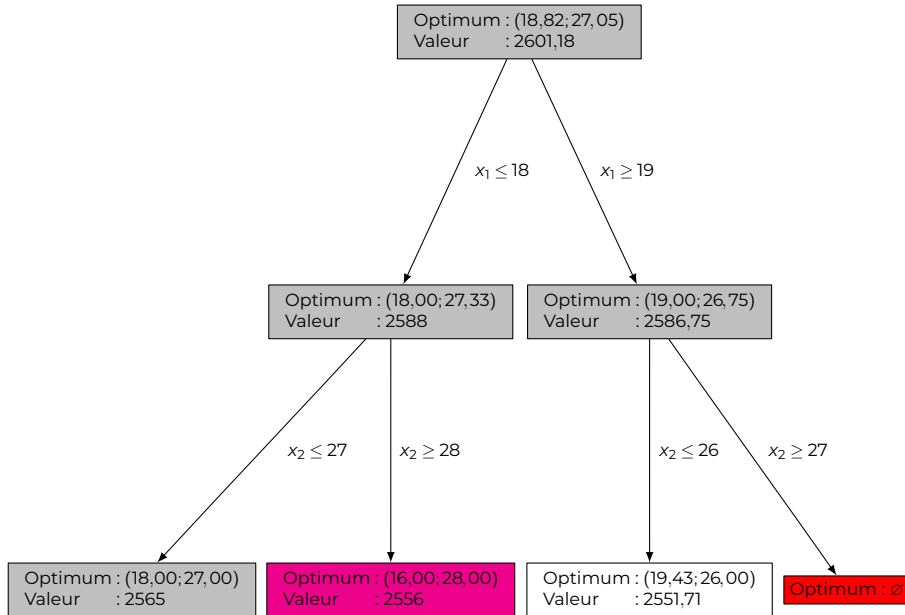
Meilleure solution entière : (18,27)



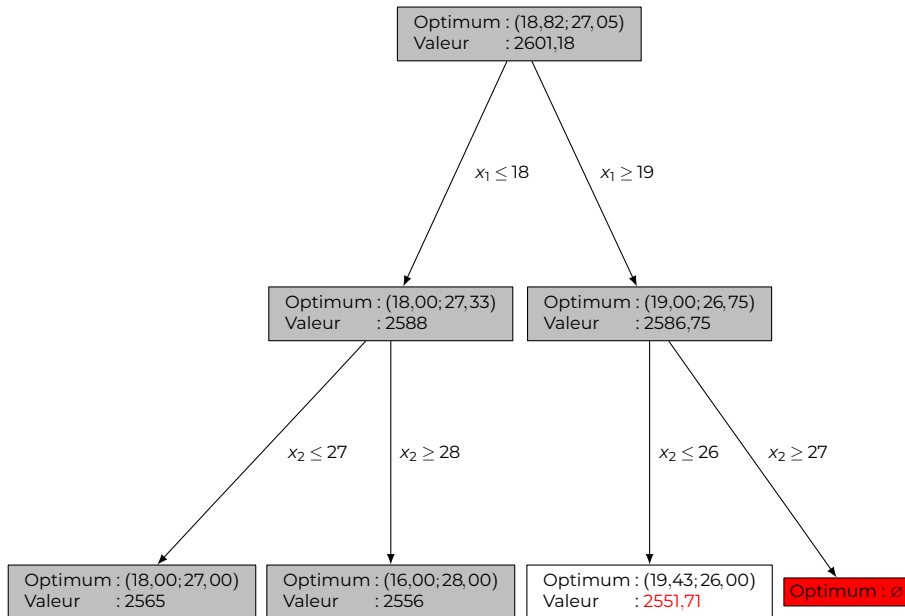
Meilleure solution entière : (18,27)



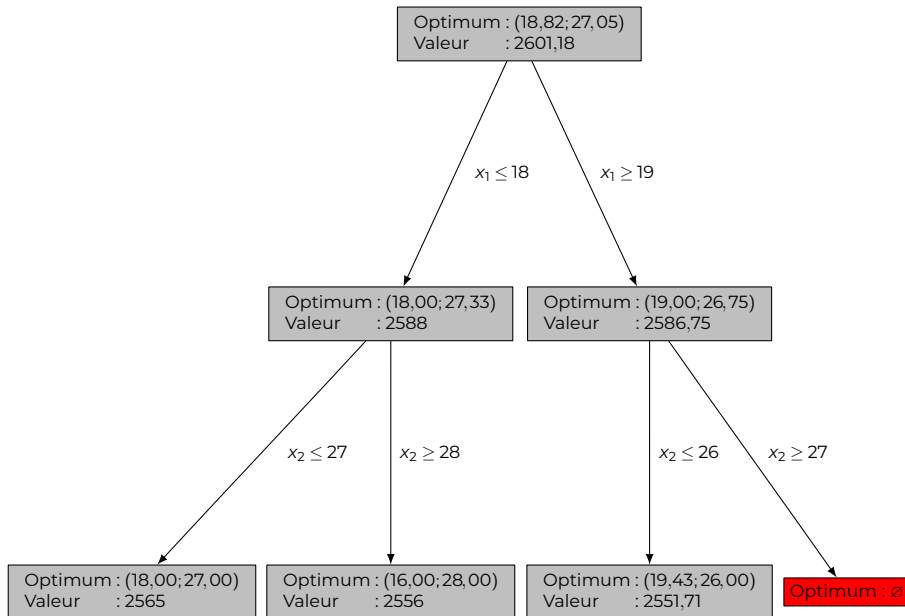
Meilleure solution entière : (18,27)



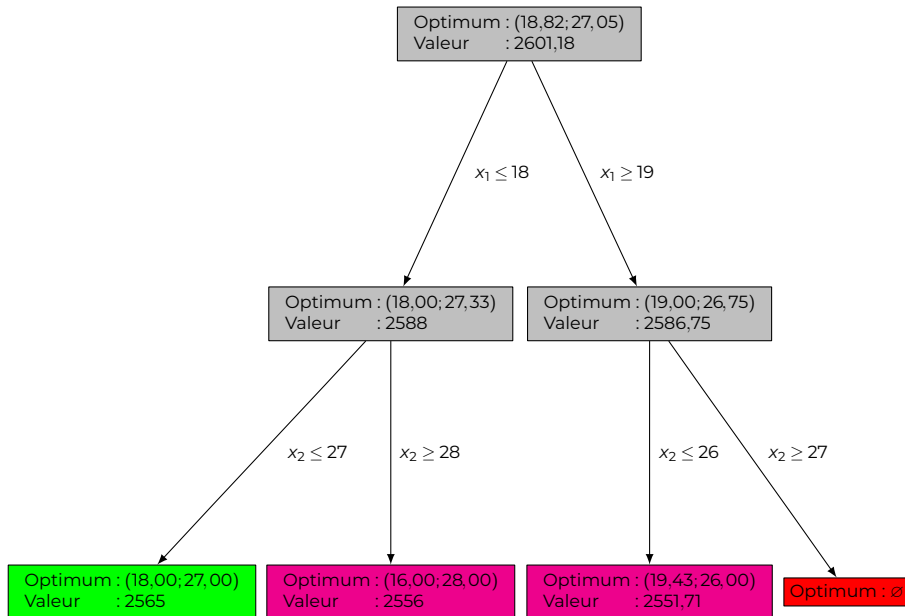
Meilleure solution entière : (18,27)



Optimum final : (18,27)



Optimum final : (18,27)



« Branch-and-cut »

Algorithme général

1. On part avec le problème initial, non traité.
2. Tant qu'il existe des problèmes non traités,
 - 2.1 on choisit un problème non traité
 - 2.2 si l'ensemble des contraintes de son problème relaxé n'est pas vide, on résout le problème relaxé associé;
 - 2.3 si l'optimum obtenu est meilleur que le meilleur optimum entier courant :
 - ▶ si l'optimum est entier, c'est le nouveau meilleur optimum entier;
 - ▶ sinon, on applique la méthode de séparation et les nouveaux problèmes obtenus sont considérés comme non traités.
 - 2.4 on considère ce problème comme traité

⇒ Complexité exponentielle

Algorithme général

1. On part avec le problème initial, non traité.
2. Tant qu'il existe des problèmes non traités,
 - 2.1 on choisit un problème non traité
 - 2.2 si l'ensemble des contraintes de son problème relaxé n'est pas vide, on résout le problème relaxé associé;
 - 2.3 si l'optimum obtenu est meilleur que le meilleur optimum entier courant :
 - ▶ si l'optimum est entier, c'est le nouveau meilleur optimum entier;
 - ▶ sinon, **appliquer la méthode des coupes tant que c'est possible**
on applique la méthode de séparation et les nouveaux problèmes obtenus sont considérés comme non traités.
 - 2.4 on considère ce problème comme traité

⇒ Complexité exponentielle

Contexte

- ▶ Résoudre un problème relaxé est (relativement) facile.
- ▶ Mais la valeur obtenue n'est pas forcément efficace... de même que la séparation qui s'ensuivrait.

Idée : prétraitement avant séparation

On rajoute des coupes afin de rapprocher la valeur du problème obtenu de celle du problème non relaxé.

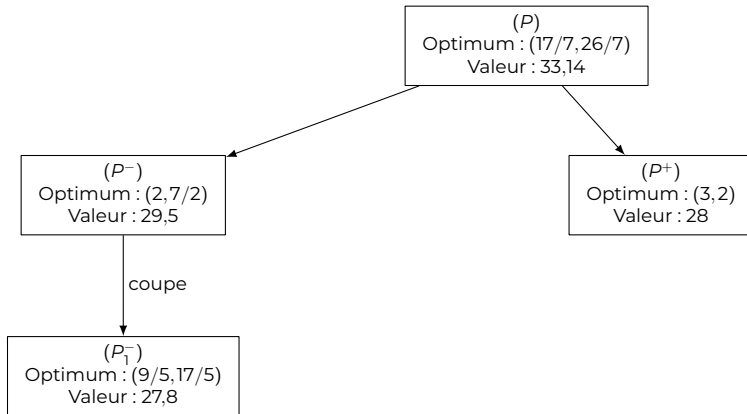
Intérêt : réduire le nombre de séparations

- ▶ Chaque coupe diminue la valeur du problème relaxé.
- ▶ Si la valeur devient suffisamment faible, on peut alors réaliser un élagage du noeud.

L'efficacité de cette méthode dépend donc de la qualité de la réduction de la valeur du problème.

Séparation/coupe et évaluation

Exemple



Sans coupe : $29,5 > 28$

Séparation de (P^-) nécessaire

Avec coupe :

$27,8 < 28$

Élagage de (P_1^-)