Unité d'Enseignement MCR – Compléments et Outils de Recherche Opérationnelle

Méthodes arborescentes et PLNE

ENSIIE

Sourour Elloumi et Eric Soutil

Définition PLNE

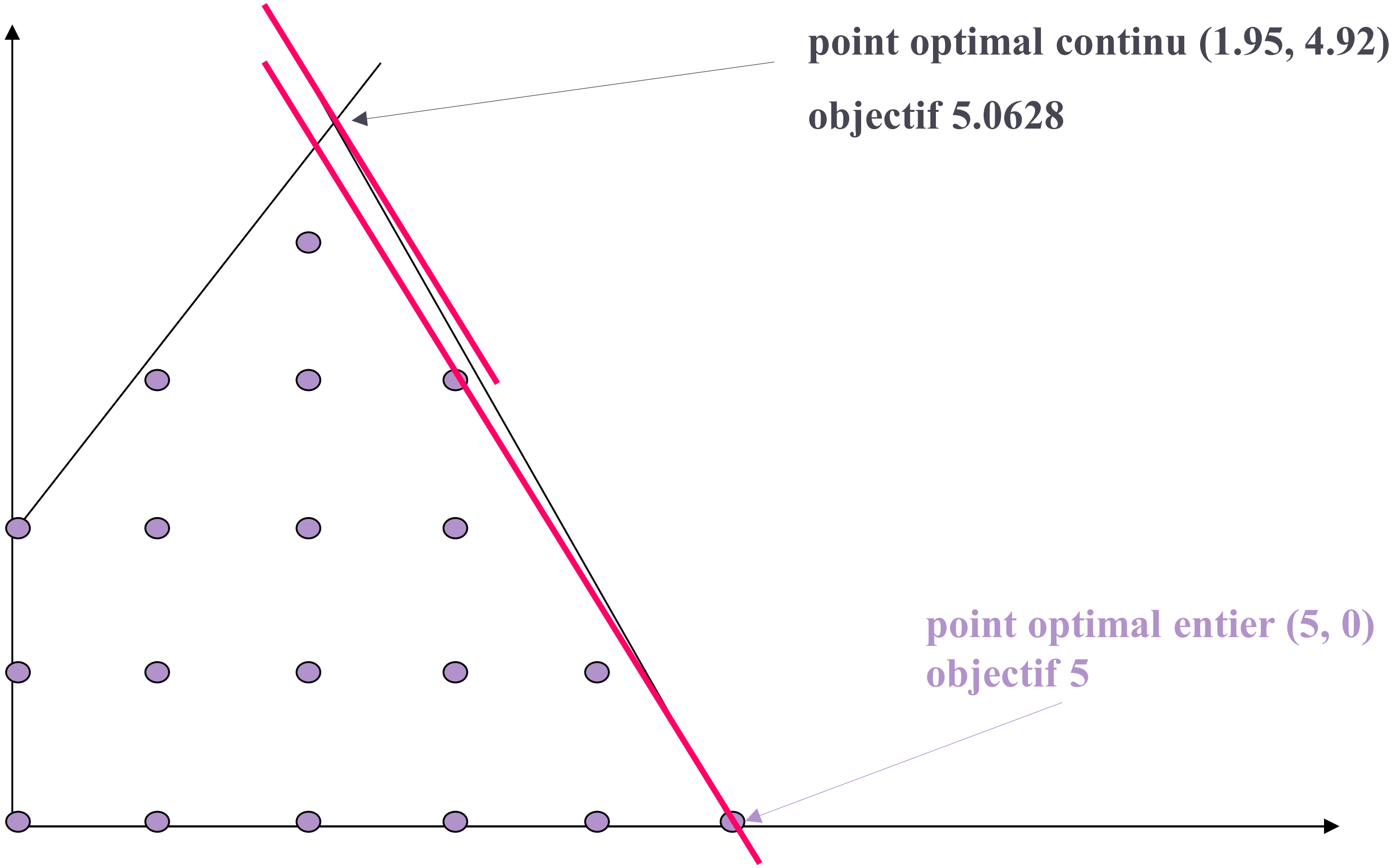
- **Programmation linéaire en nombres entiers (PLNE)** : Programmation linéaire où certaines variables ne peuvent prendre que des valeurs entières

  - **Programmation pure** (resp. **mixte**) en NE : la totalité (resp. un sous-ensemble) des variables sont entières
  - **Programmation 0-1** ou binaire : les variables entières ne peuvent être que 0 ou 1

Exemple 1

$$
\begin{cases}
\max x_1 + 0.64 x_2 \\
\text{s.c.}: \\
\quad 50x_1 + 31x_2 \leq 250 \\
\quad 3x_1 - 2x_2 \geq -4 \\
\quad x_1, x_2 \geq 0 \text{ et entiers}
\end{cases}
$$

Exemple 1- Suite



Définition PLNE

- PL et PLNE sont TRÉS différentes
- Optimisation continue convexe / Optimisation discrète

Suite de ce cours :

- Utilité de la PLNE
- Résolution des PLNE

Application 1 : choix d'usines et d'entrepôts

- But : choisir de nouveaux emplacements pour construire des usines et des entrepôts
- Deux emplacements : Lyon et Toulouse
- On ne peut construire plus d'un entrepôt
- On ne peut construire un entrepôt que dans une ville où l'on a aussi une usine
- On connaît, pour chaque ville :
  - Coefficient de rentabilité usine ou entrepôt
  - Coût de construction usine ou entrepôt
- Le coût total de construction ne peut pas dépasser 10
- Objectif : maximiser la rentabilité

Application 1 : données

	Rentabilité	Coût
Usine à L	9	6
Usine à T	5	3
Entrepôt à L	6	5
Entrepôt à T	4	2

Application 1 : modèle

- Variables de décision :
  - $x_1 = 1$ si une **usine** est construite à **Lyon** et 0 sinon
  - $x_2 = 1$ si une **usine** est construite à **Toulouse** et 0 sinon
  - $y_1 = 1$ si un **entrepôt** est construit à **Lyon** et 0 sinon
  - $y_2 = 1$ si un **entrepôt** est construit à **Toulouse** et 0 sinon

Application 1 : modèle

- **Contraintes**

  1- On ne peut construire plus d'un entrepôt

$$y_1 + y_2 \leq 1$$

  2- On ne peut construire un entrepôt que dans une ville où l'on a aussi une usine

$$y_1 \leq x_1$$
$$y_2 \leq x_2$$

  3- Le coût total de construction ne peut pas dépasser 10

$$6x_1 + 3x_2 + 5y_1 + 2y_2 \leq 10$$

- **Objectif**

$$\max \quad 9x_1 + 5x_2 + 6y_1 + 4y_2$$

Application 1 : résumé du modèle

$$
\begin{cases}
\max Z = 9\,x_1 + 5x_2 + 6y_1 + 4y_2 \\
\text{s.c.} \quad \left| \begin{array}{l} y_1 + y_2 \leq 1 \\ y_1 \leq x_1 \\ y_2 \leq x_2 \\ 6x_1 + 3x_2 + 5y_1 + 2y_2 \leq 10 \\ 0 \leq x_1, x_2, y_1, y_2 \leq 1 \ \text{ et entiers} \end{array} \right.
\end{cases}
$$

C'est notre exemple de base

Exemple de base: résolution par un solveur/modeleur (glpk)


```
#Exemple du cours

#variables
var x1 binary;
var x2 binary;
var y1 binary;
var y2 binary;

#fonction objectif
maximize Z : 9*x1 + 5*x2 + 6*y1 + 4*y2 ;

#contraintes
subject to
c1: y1 + y2 <= 1;
c2: y1 <= x1;
c3: y2 <= x2;
c4: 6*x1 +3*x2 +5*y1 +2*y2 <= 10;
```

```
printf "------Début de la résolution -----\n";
solve;

printf "------Fin de la résolution -----\n";
display Z;
display x1;
display x2;
display y1;
display y2;

end;
```

Exemple de base: résolution par un solveur/modeleur (Glpk)

Valeur optimale : $Z^* = 14$

valeur de x1 : 1

valeur de x2 : 1

valeur de y1 : 0

valeur de y2 : 0

Langage de modélisation et solveur : Glpk

Résolution des PLNE

- Nous allons voir comment résoudre les PLNE

1. Dans le cas particulier des problèmes purement binaires PL01
2. Dans le cas général des PLNE

Résolution des PL01- Exemple de base

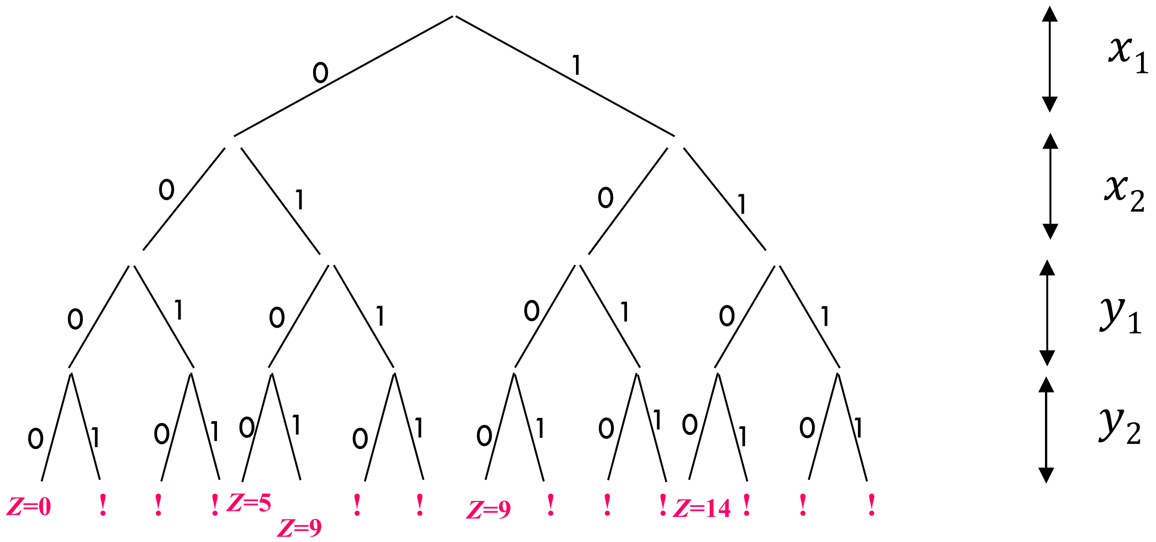
$$\begin{cases} \max Z = 9\,x_1 + 5x_2 + 6y_1 + 4y_2 \\ \quad \text{s.c.} \quad \left| \begin{array}{l} y_1 + y_2 \leq 1 \\ y_1 \leq x_1 \\ y_2 \leq x_2 \\ 6x_1 + 3x_2 + 5y_1 + 2y_2 \leq 10 \\ 0 \leq x_1, x_2, y_1, y_2 \leq 1 \ \text{ et entiers} \end{array} \right. \end{cases}$$

Résolution des PL01

1ère idée : énumération

Résolution des PL01- Énumération

- Idée : il existe un nombre fini de solutions. Enumérons-les !



Résolution des PL01- Enumération

- Pour n variables binaires, 2^n cas possibles
- Pour $n=20$, plus d'un million
- Pour $n=30$, plus d'un milliard …

- **Idée d'énumération de tous les cas possibles impraticable en Optimisation Discrète en général**

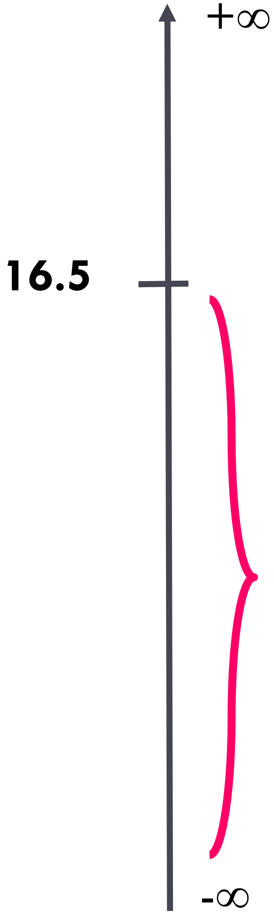
Résolution des PL01

2ème idée : encadrement de la valeur optimale

Relaxation continue - exemple de base

- **Relaxation continue** = on « oublie » le caractère entier des variables
- On obtient un programme linéaire (continu) qu'on sait résoudre, par exemple par la méthode du simplexe :
  - Solution $(x_1, x_2, y_1, y_2) = \left(\frac{5}{6}, 1{,}0{,}1\right)$
  - Valeur optimale : $Z = \frac{33}{2} = 16.5$

Relaxation continue: interprétation



Valeurs possibles de toutes les solutions admissibles, entières ou non

Relaxation continue: interprétation

Conclusion : la valeur optimale (en entier) ≤ 16.5

Propriété générale

- Pour un problème de maximisation :

  la valeur optimale en entier $\leq$ la valeur optimale en continu

- Pour un problème de minimisation :

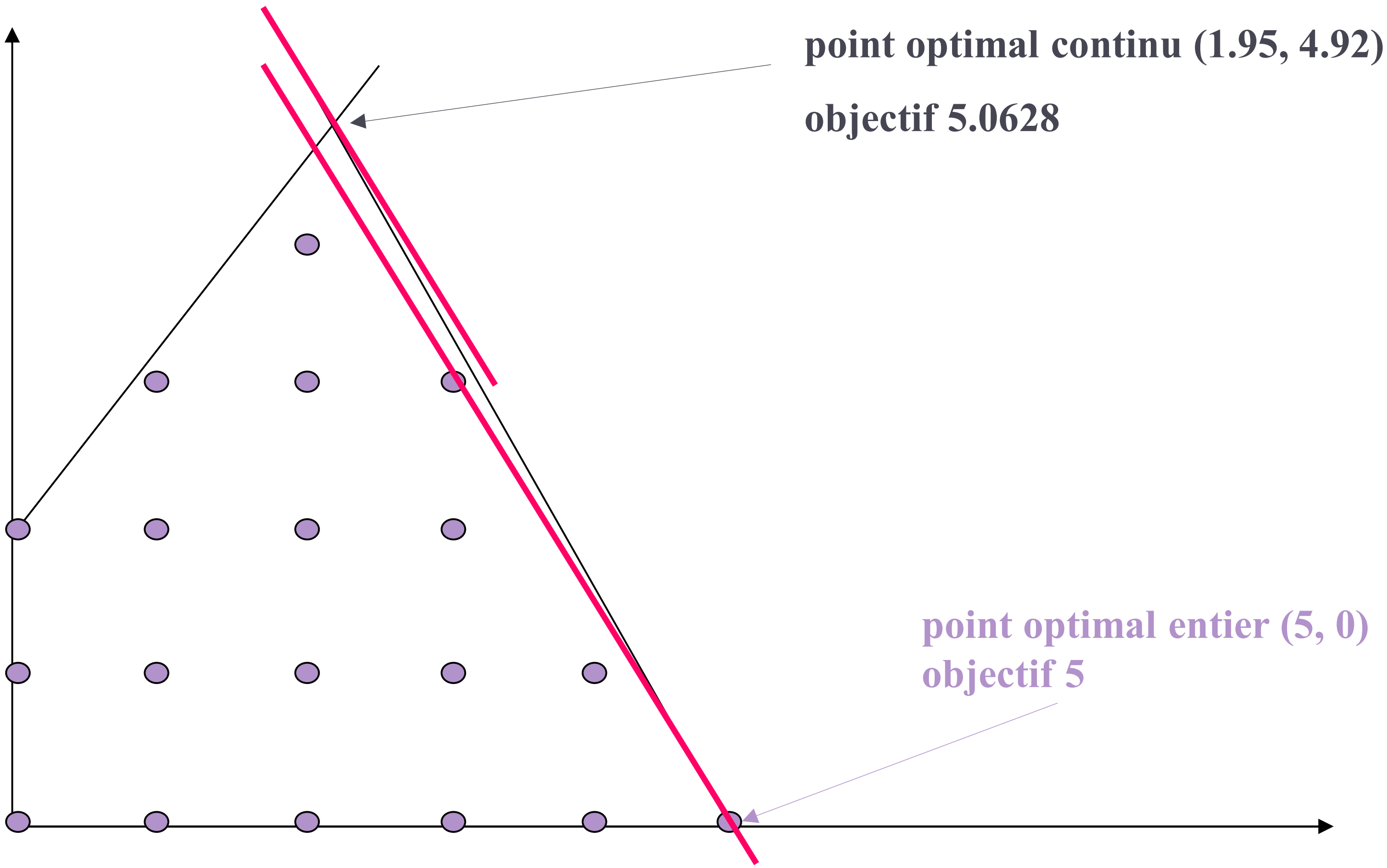
  la valeur optimale en entier $\geq$ la valeur optimale en continu

On dit que la valeur optimale de la relaxation continue est

- une **borne supérieure** (si objectif de maximisation) ou
- une **borne inférieure** (si objectif de minimisation)

de la valeur optimale en entier

Exemple 1- Rappel (max)



Connaissance d'une solution admissible

- **Question** : quelle information a-t-on si l'on dispose d'une solution admissible ?

Connaissance d’une solution admissible-exemple de base

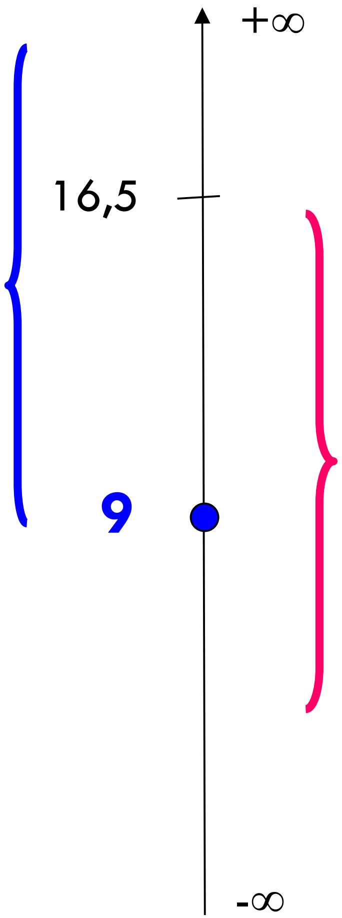
- Admettons que l’on connaisse la solution

$$(x_1, x_2, y_1, y_2) = (1,0,0,0)$$

de valeur $Z = \mathbf{9}$

Connaissance d'une solution admissible- exemple de base

Valeurs possibles de toutes les solutions entières **optimales**



Valeurs possibles de toutes les solutions admissibles, entières ou non

Connaissance d'une solution admissible-exemple de base

Conclusion : la valeur d'une solution optimale est comprise entre **9** et 16.5

Propriété générale :

- Pour un problème de maximisation,

$$\underbrace{\text{val. d'une solution admissible}}_{\text{borne inférieure}} \leq \textbf{val. optimale en entier} \leq \underbrace{\text{val. optimale en continu}}_{\text{borne supérieure}}$$

- Pour un problème de minimisation,

  val. optimale en continu ≤ **val. optimale en entier** ≤ val. d'une solution admissible

Algorithme Branch-and-Bound (Séparation et évaluation) pour résoudre un PLNE

- Ingrédients de base :

  - encadrement de la valeur optimale (borne inférieure, borne supérieure)

  - énumération limitée dans le but d'obtenir un encadrement de plus en plus fin

B&B- Exemple de base

- Dans la solution de la relaxation continue :

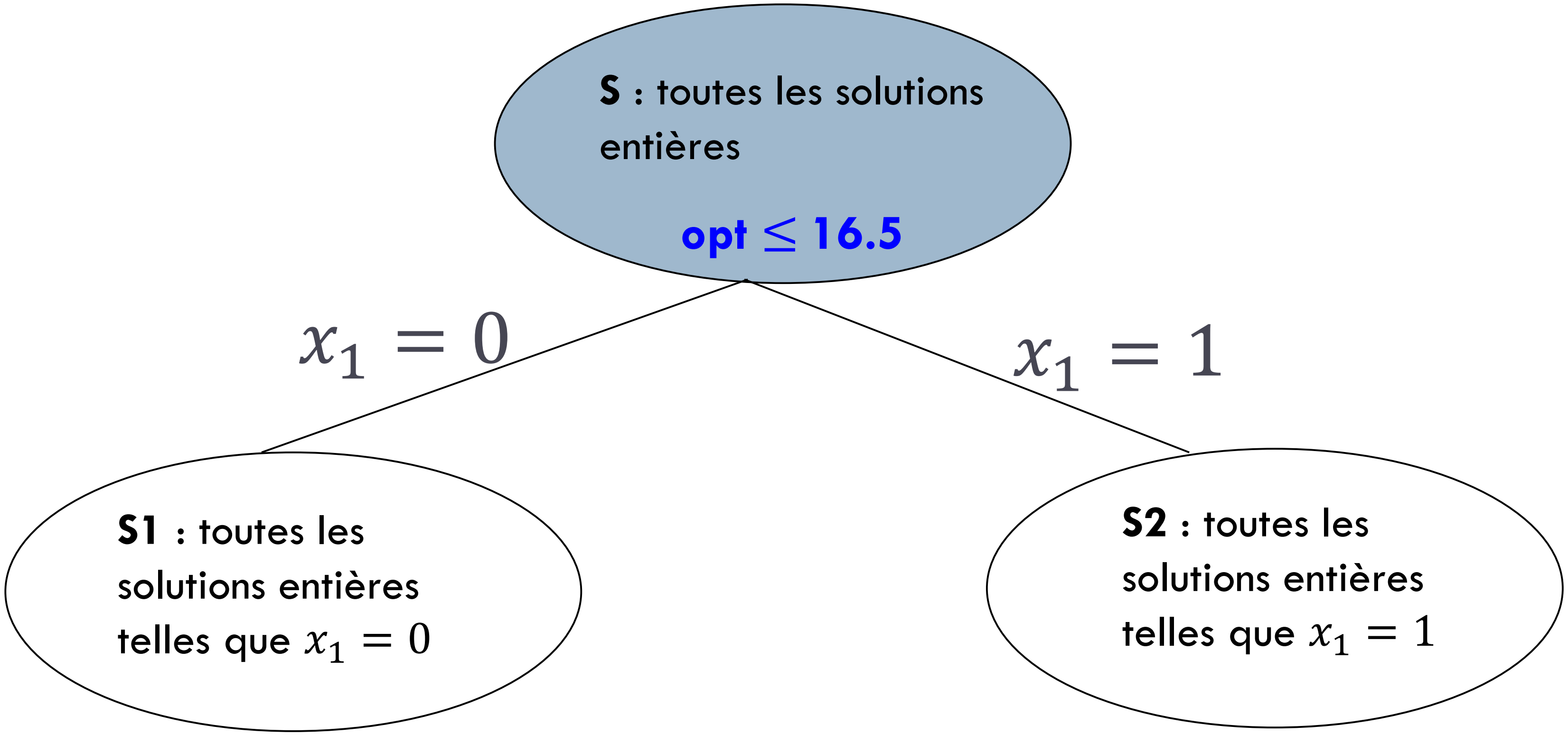
$$(x_1, x_2, y_1, y_2) = \left(\frac{5}{6}, 1,0,1\right),$$

x_1 n'est pas entier.

- On va « brancher » selon les deux valeurs possibles de x_1 : 0 et 1

B&B - Exemple de base

Solution courante : 9



B&B- Exemple de base

Sous-ensemble **S1** : $x_1 = 0$

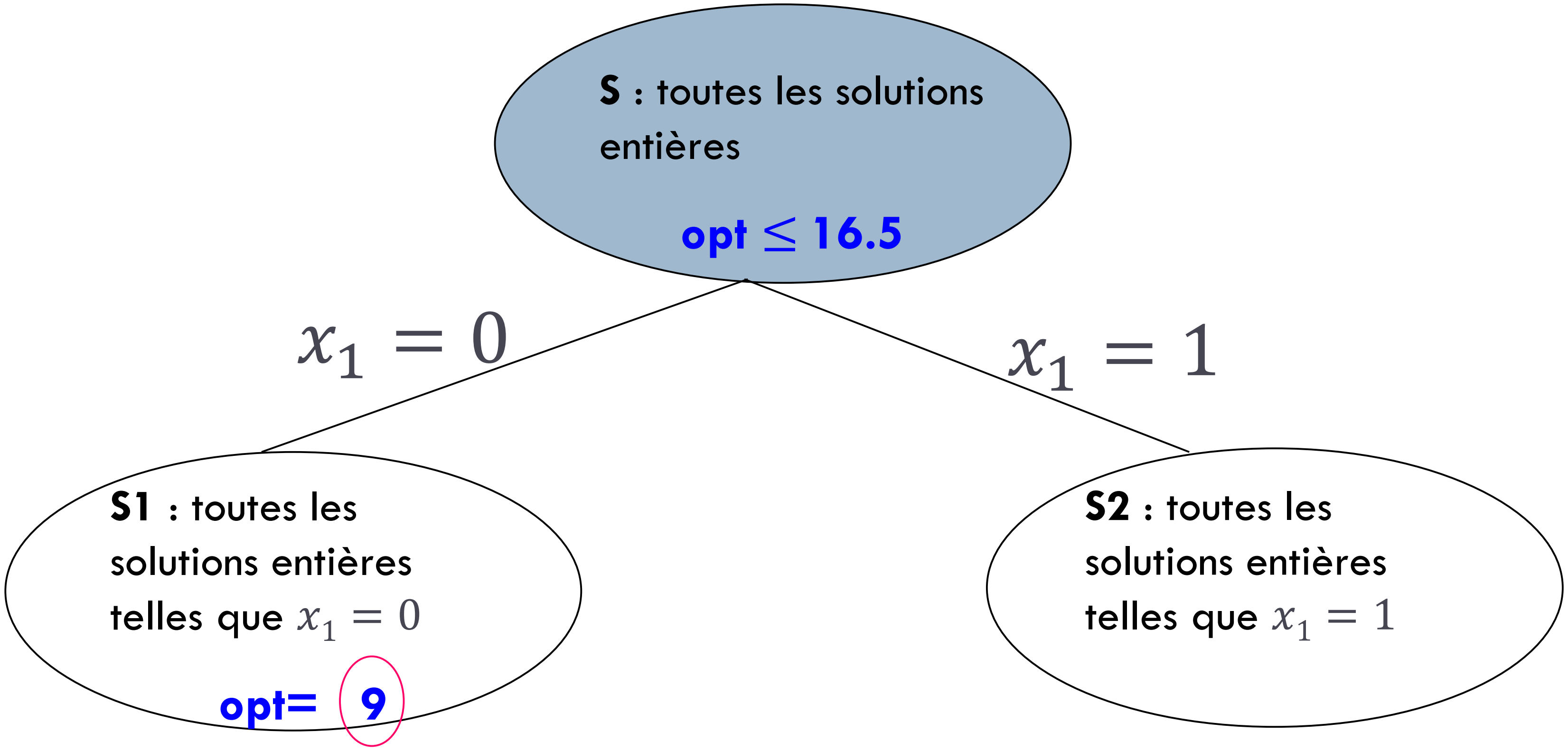
$$\begin{cases} \max Z = 5x_2 + 6y_1 + 4y_2 \\ \text{s.c.} \quad \left| \begin{array}{l} y_1 + y_2 \leq 1 \\ y_1 \leq 0 \\ y_2 \leq x_2 \\ 3x_2 + 5y_1 + 2y_2 \leq 10 \\ 0 \leq x_2, y_1, y_2 \leq 1 \text{ et entiers} \end{array} \right. \end{cases}$$

Solution de la relaxation continue : $(x_2, y_1, y_2) = (1,0,1)$, Z= 9,

et c'est une solution entière !

B&B- Exemple de base

Solution courante : 9



B&B- Exemple de base

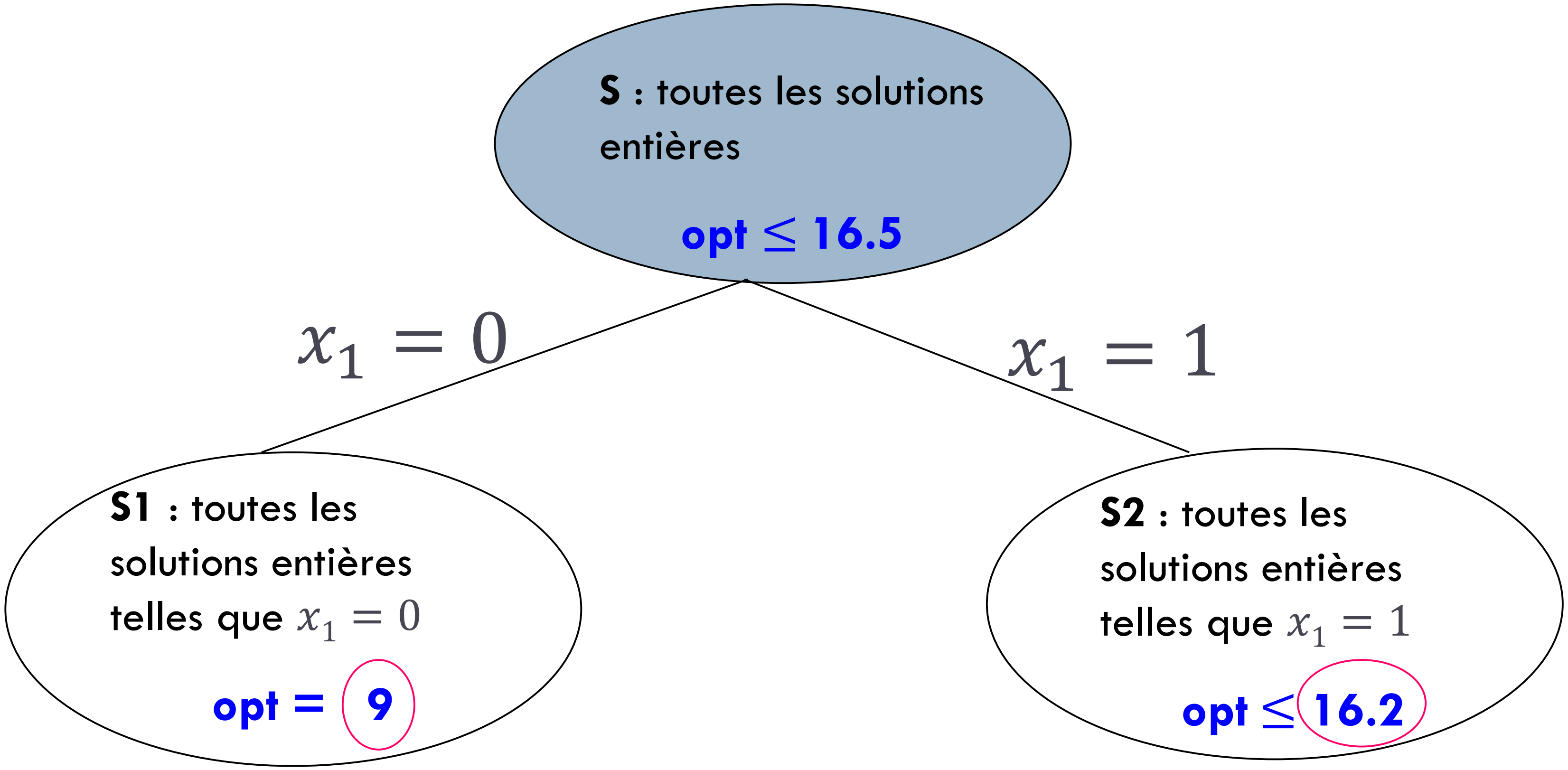
Sous-ensemble **S2** : $x1=1$

$$\begin{cases} \max Z = 9 + 5x_2 + 6y_1 + 4y_2 \\ \text{s.c.} \quad \left| \begin{array}{l} y_1 + y_2 \leq 1 \\ y_1 \leq 1 \\ y_2 \leq x_2 \\ 3x_2 + 5y_1 + 2y_2 \leq 4 \\ 0 \leq x_2, y_1, y_2 \leq 1 \text{ et entiers} \end{array} \right. \end{cases}$$

Solution de la relaxation continue : $(x_2, y_1, y_2) = \left(\frac{4}{5}, 0, \frac{4}{5}\right)$ et $Z = 16.2$

B&B- Exemple de base

Solution courante : 9



B&B- Exemple de base

- Conclusion actuelle :
  - Meilleure solution admissible connue (**solution courante**) de valeur 9
  - Meilleure borne supérieure connue : 16.2
  - La valeur optimale est donc comprise entre 9 et 16.2

- Comment continuer ?

B&B- Exemple de base

Le nœud **S1** peut être **élagué** car on connaît la valeur d'une solution entière optimale dans cet ensemble.

Critère d'élagage :

La relaxation continue a une solution optimale entière

S1 : toutes les solutions entières telles que $x_1 = 0$

opt = 9

B&B- Exemple de base

Pour le nœud **S2**, on applique le même traitement que pour **S**

S2 : toutes les solutions entières telles que $x_1 = 1$

opt ≥ 16.2

Solution de la relaxation continue :

$(x_1, x_2, y_1, y_2) = \left(1, \frac{4}{5}, 0, \frac{4}{5}\right)$. Et $Z = 16.2$.

On va brancher sur x_2.

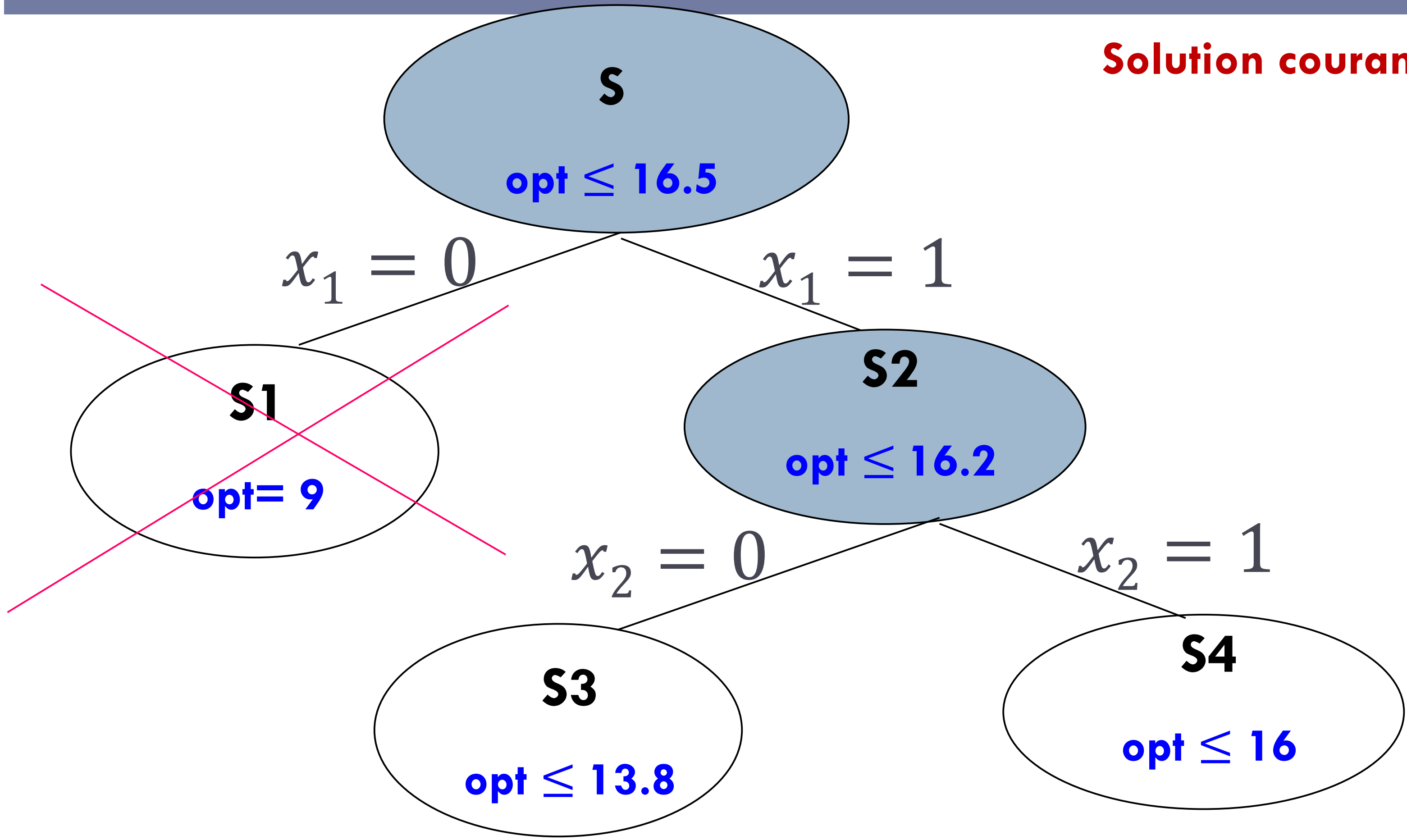
B&B- Exemple de base

- Sous-ensemble **S3** : x_1=1 et x_2=0. Solution de la relaxation continue : $(x_1, x_2, y_1, y_2) = \left(1,0,\frac{4}{5},0\right)$ et $Z = 13.8$

- Sous-ensemble **S4** : $x_1 = 1$ et $x_2 = 1$. Solution de la relaxation continue : $(x_1, x_2, y_1, y_2) = \left(1,1,0,\frac{1}{2}\right)$ et $Z = 16$

B&B- Exemple de base

Solution courante : 9



B&B- Exemple de base

- Conclusion actuelle :
  - Valeur meilleure solution connue : 9
  - Meilleure borne supérieure connue :16

- On ne peut élaguer ni **S3** ni **S4**

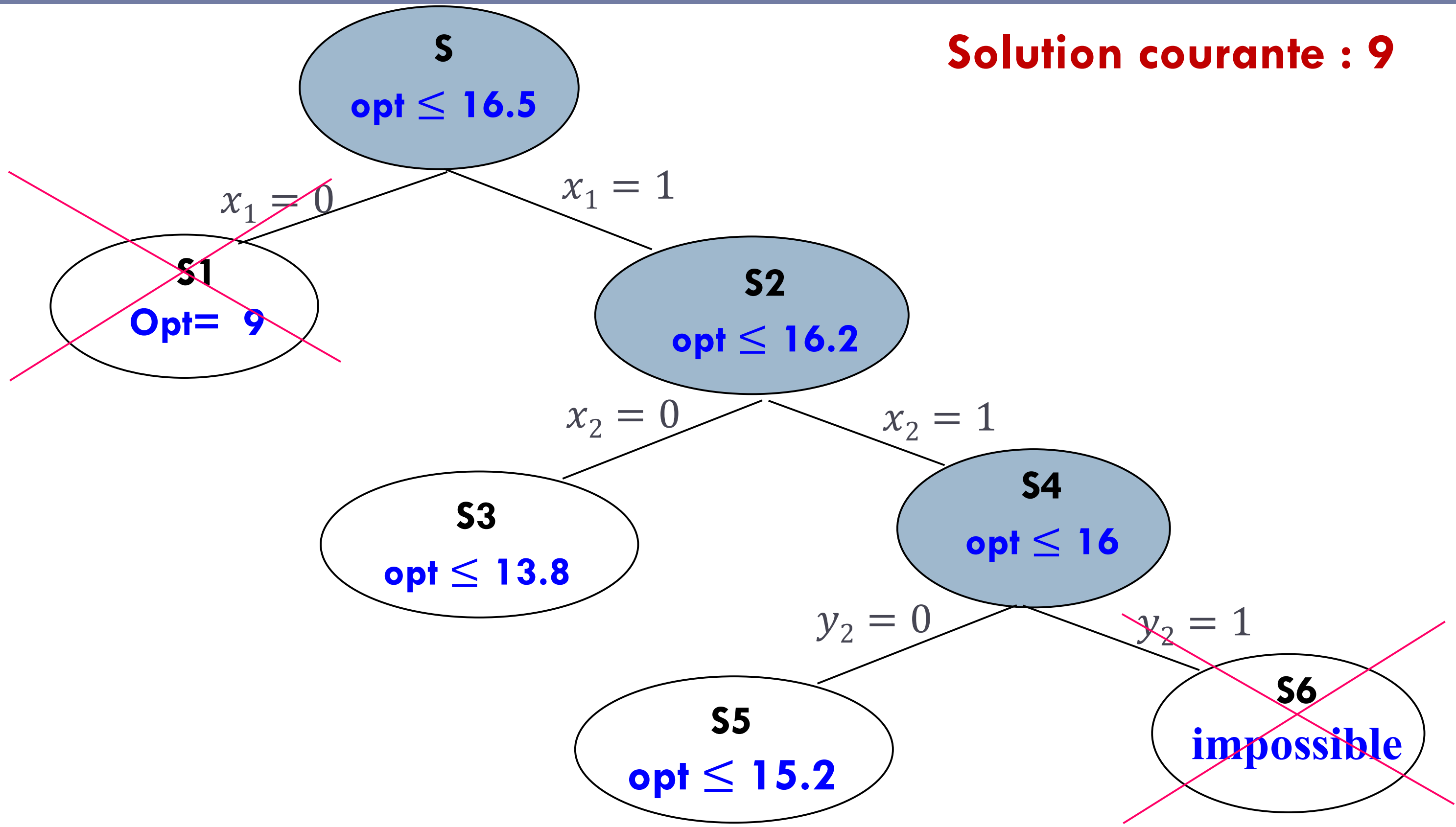
- On « repart » avec **S4** qui a la plus grande borne supérieure. On branche sur y_2.

B&B- Exemple de base

- Sous-ensemble **S5** : $x_1 = 1$, $x_2 = 1$ et $y_2 = 0$
  Solution de la relaxation continue :
  $(x_1, x_2, y_1, y_2) = \left(1,1,\frac{1}{5},0\right)$ et $Z = 15.2$

- Sous-ensemble **S6** : $x_1 = 1$, $x_2 = 1$ et $y_2 = 1$
  impossible. **S6** peut donc être élagué

B&B- Exemple de base



B&B- Exemple de base

- Conclusion actuelle :
  - Valeur meilleure solution connue : 9
  - Meilleure borne supérieure connue : 15.2

- On peut repartir soit avec **S3** soit avec **S5**, on repart avec **S5**, on branche sur y_1.

B&B- Exemple de base

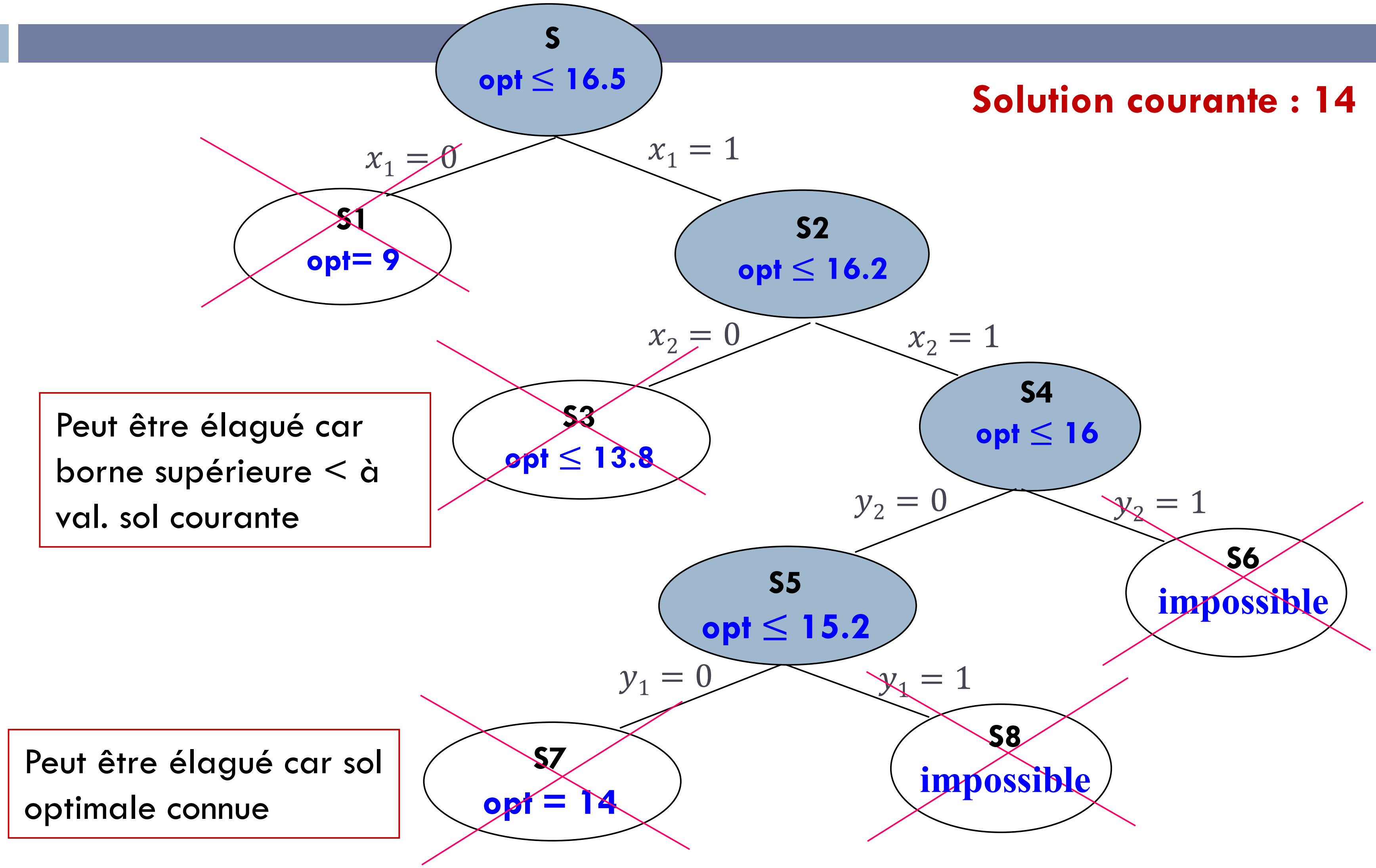
- Sous-ensemble **S7** :

  $x_1 = 1, x_2 = 1, y_2 = 0$ et $y_1 = 0$. Solution unique entière et $Z = 14$ (>9 donc **nouvelle solution courante**)

- Sous-ensemble **S8** :

  $x_1 = 1, x_2 = 1, y_2 = 0$ et $y_1 = 1$ impossible. S8 peut donc être élagué

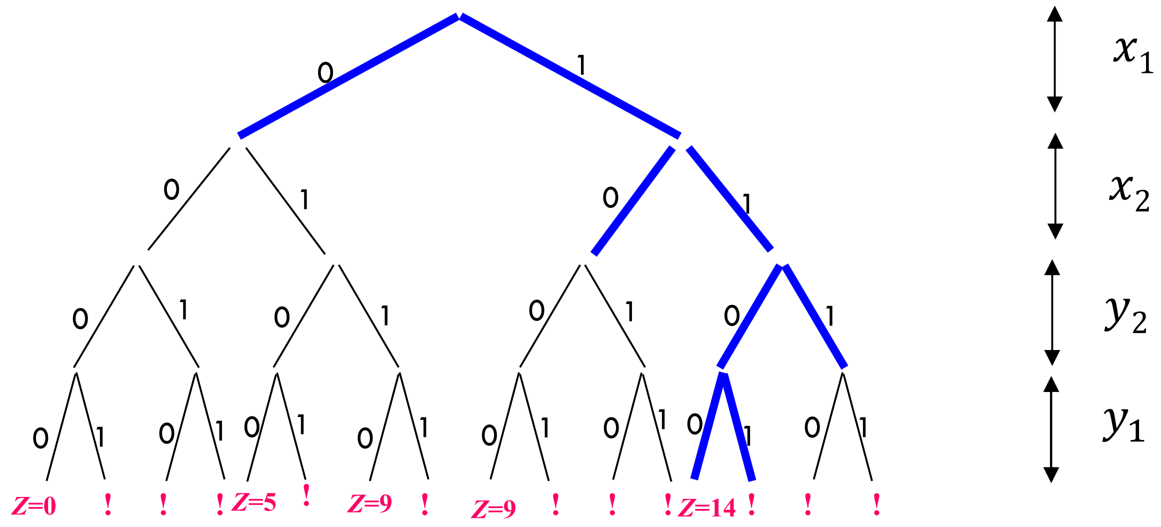
B&B- Exemple de base



B&B- Exemple de base

- Conclusion : on peut s'arrêter car tous les nœuds ont été élagués. On prouve ainsi que la solution de valeur 14 est optimale !

B&B- Exemple de base, gain par rapport à l'énumération complète



Algorithme B&B (max)- Résumé

- Initialisation
  - Calculer une solution admissible de valeur Z^* ou poser $Z^*=-\infty$
  - Résoudre la relaxation continue et mettre à jour éventuellement Z^* (évaluation)
  - Appliquer les tests d'élagage
- Tant qu'il reste des nœuds non élagués
  - Choisir un nœud non élagué
  - Brancher sur une des variables (séparation)
  - Pour chacun des 2 nouveaux nœuds, résoudre la relaxation continue et mettre à jour éventuellement Z^*
  - Appliquer les tests d'élagage
- Fin tant que
- La solution courante Z^* est optimale

Algorithme B&B (max)- Résumé- suite

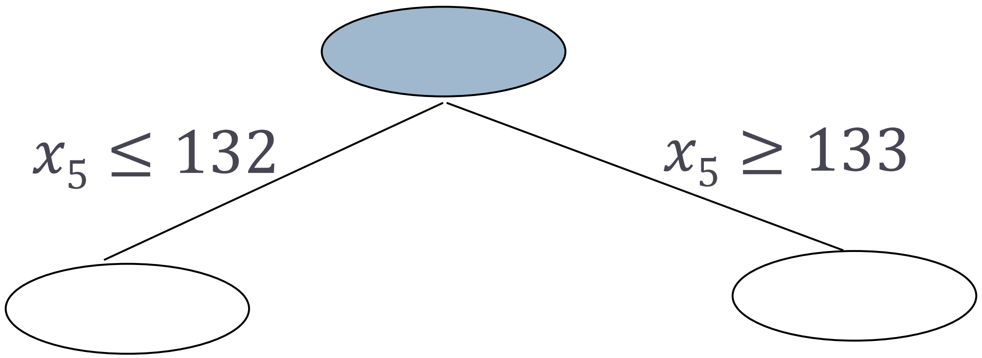
- Un nœud est élagué si :
  - La relaxation continue n'a pas de solution
  - La valeur optimale de la relaxation continue $\leq Z^*$
  - Solution entière de la relaxation continue
  - Pas de solution admissible entière : très difficile à tester en général

- La mise en place de l'algorithme nécessite de préciser :
  - La règle de sélection : quel nœud non élagué choisir ?
  - La règle de branchement : sur quelle variable brancher ?

Algorithme B&B (max)- Résumé- suite

- Dans le cas général des variables entières (non seulement 0-1), on choisit une variable de valeur fractionnaire dans la solution optimale de la relaxation continue et on branche sur l'arrondi supérieur et inférieur de cette valeur

Exemple : $x_5 = 132.48$



Détermination des solutions admissibles

- Ce n’est pas toujours facile (pb général NP-complet)
- Il n’existe pas de méthode générale rapide
- On peut se contenter de celles qu’on trouve lors de la résolution des relaxations continues
- Il existe des algorithmes qui fonctionnent bien dans des cas particuliers, par exemple l’arrondi

Problème d'efficacité

- C'est le nombre de nœuds explorés qui déterminera le temps de calcul. À chaque nœud, on résout un programme linéaire (continu)

- On ne peut pas prévoir à l'avance le nombre maximal de nœuds qu'il faudra explorer

- En règle générale, un programme linéaire continu se résout « vite »

- Un PLNE nécessite du temps …

Efficacité- Exemple

$$\min Z = \sum_{j=1}^{n} c_j x_j$$

s.c. :

$$\sum_{j=1}^{n} a_{1j} x_j \le b_1$$

$$\sum_{j=1}^{n} a_{2j} x_j \le b_2$$

$$x_j \quad \text{variables binaires}$$

- Problème avec $n = 1000$ variables données générées aléatoirement
- Relaxation continue : 0.03 secondes
- Résolution en entier : 43 secondes ($251\,402$ nœuds)

Quelques implémentations classiques des B&B

- <u>**Méthode de Balas pour les PL01 (minimisation)**</u> :
  - Ré-écrire z (à minimiser) de sorte que z n'ait pas de coefficients négatifs et les variables soient triées en ordre croissant de ces coefficients (tout programme peut être exprimé de cette façon)
  - Séparation : fixer une variable soit à 0 soit à 1
  - Si un sous-problème a un second-membre positif ou nul, on a déjà une solution réalisable (en fixant toutes les variables restant à 0) et c'est la solution optimale de ce (sous-)problème
  - Parcourir l'arbre en profondeur
  - Plusieurs règles d'élagage permettent de savoir qu'un problème donné est infaisable ou ne peut donner une solution meilleure d'une solution déjà trouvée.

Quelques implémentations classiques des B&B

- **Méthode de Dakin pour les PLNE** :

  - Choisir à chaque itération le nœud possédant la meilleure solution de son programme relaxé

  - Séparer sur une nouvelle variable dont la valeur (dans la solution optimale du programme relaxé) est la plus proche d'un entier k. Cette variable a une fourchette de valeurs possibles, disons [min,max]. On construit deux nouveaux problèmes, un avec [min,k], et l'autre avec [k+1,max]