Solution TD Branch and bound

Exercice1

1) D'après le schéma d'arbre, les chemins réalisables (de la racine jusqu'aux feuilles) constituent les solutions possibles sont : 28, 31, 27,35 et le minimum est 27

- La meilleure borne inferieure est **LB=27,** parce que toutes les évaluations des nœuds qui suivent immédiatement la racine sont supérieures ou égales à 27.
- La meilleure borne supérieure est **UB=28,** parce qu'en utilisant une méthode gloutonne basée sur la recherche en profondeur d'abord (DFS) , on peut trouver une première solution acceptable pour ce problème. **(0)→(1) →(3)**

2) selon les LB et UB trouvés dans l'étape précédente :

Les nœuds sondés (recherchés)	Les nœuds séparés
0 (racine) 1 → eval(1)>=LB **et** eval(1)<=UB 2 → eval(2)>=LB **et** eval(2)<=UB	4 → eval(4)>=LB **MAIS eval(4)>UB** Donc 4 est séparé

Exercice2 Branch and bound pour Max 3-SAT

Pour une formule CNF donnée, B&B explore l'espace de recherche de toutes les affectations possibles dans une recherche en profondeur d'abord. Le schéma de l'algorithme branch&bound pour le problème MAX 3-SAT est décrit par l'algorithme suivant :

```
Algorithme 5.5 : un algorithme  B&B
Données : une formule ϕ en CNF, une borne supérieure ub
Résultat : le nombre minimum de clauses fausses.
  début
   si (ϕ = Ø ou ϕ  ne contient que des clauses vides) Alors
      Retourner le nombre de clauses vides de ϕ ;
   fin
   si (le nombre de clauses vides de ub ≤ ϕ)  alors
       Retourner  ∞ ;
   fin
  v= variable sélectionnée par une heuristique;
  ub= min(ub,  B&B (Simplifier(ϕ,¬x),ub));
  Retourner min(ub,  B&B ( Simplifier(ϕ,x), ub));
  Fin
```

L'algorithme B&B commence par sélectionner une variable x non valuée grâce à une heuristique de branchement donnée. Cette variable est ensuite instanciée et la formule est simplifiée en fonction de cette valuation : quand x est mise à vrai (resp. faux), les clauses contenant le littéral x (resp. ¬x) sont supprimées et toutes les occurrences du littéral ¬x (resp. x) sont effacées des clauses les contenant. Ce processus est connu sous le nom de règle du littéral unique. Une clause dont tous les littéraux ont été effacés (clause vide) correspond à une clause fausse. Si le nombre de clauses fausses obtenues avec cette valuation est supérieur à la borne supérieur (ub) qui représentant le nombre de clauses fausses obtenues par la meilleure affectation trouvée depuis le début de la recherche alors aucune affectation construite à partir de ce nœud ne peut fournir un nombre de clauses fausses plus petit que ub. La branche correspondante est donc coupée. À ce moment, soit la variable courante est flippée (vrai à faux ou faux à vrai), soit, si les deux valeurs de vérité possibles ont été testées, l'algorithme effectue un backtrack (la valeur de vérité de la variable courante est effacée et la variable précédente est inspectée).

Quand une affectation complète dont le nombre de clauses vides correspondant est inférieur à ub est trouvée, alors ub est mise à jour avec cette nouvelle valeur. Le processus ne s'arrête que quand tous les backtracks possibles ont été effectués. Le nombre minimum de clauses fausses pour la formule initialement donnée est la valeur courante de ub. Au début de la recherche, ub doit être initialisée. Au moins deux possibilités existent. La première consiste à démarrer avec une valeur égale au nombre de clauses de la formule mais une telle initialisation mène à une large exploration de l'arbre de recherche. Pour réduire cette exploration, la seconde technique consiste à utiliser un algorithme approché qui retourne une affectation engendrant peu de clauses fausses. Il est à noter que, étant donné la bonne efficacité de ces algorithmes, la valeur initiale obtenue de cette manière est souvent de très bonne qualité. En fait, dans beaucoup de cas, cette valeur ub correspond à la meilleure valeur possible.

<u>Exercice 3</u>

1) la relaxation linaire continue consiste à supprimer les conditions **x1, x2 entiers** et le problème devient un problème purement linaire avec des variables réelles :

$$\max z = 4x1 - x2$$
$$7x1 - 2x2 \leq 14$$
$$x2 \leq 3$$
$$2x1 - 2x2 \leq 3$$

$x1, x2 \in R^+$

Pour résoudre ce problème linéaire, on utilise la méthode du simplex (voir les cours PL)

La solution trouvée pour sous problème est :

X1=2.85714286
X2=3
Z= 8.42857143

La valeur Z représente <u>la borne supérieure</u> de la solution du problème linéaire en ajoutant les contraintes entières(PLNE).

Mais la solution dans un programme en PLNE doit être entière et non fractionnaire

2) La résolution du problème par B&B

Ce problème est facilement résoluble par la méthode B&B parce que la contrainte x2 ≤ 3 nous aide énormément pour énumérer les nœuds de l'arbre de résolution.

Afin de séparer les nœuds il faut **une borne inferieure :**

Comme la solution exacte da la relaxation continue est (2.86, 3), on voit clairement que x2=3 et un entier, on peut donc remplacer la valeur de x2 dans le système afin de trouver une borne inférieure.

Max z = 4x1 −3

$7x1 - 6 \leq 14$

$2x1 - 4 \leq 3$

Après la résolution de ce système simple on trouve la valeur de x1=2

On a une solution admissible (2,3)

et z=5 donc : la meilleure solution si elle existe doit être supérieure ou égale à 5, on peut prendre cette valeur comme une borne inferieure LB.

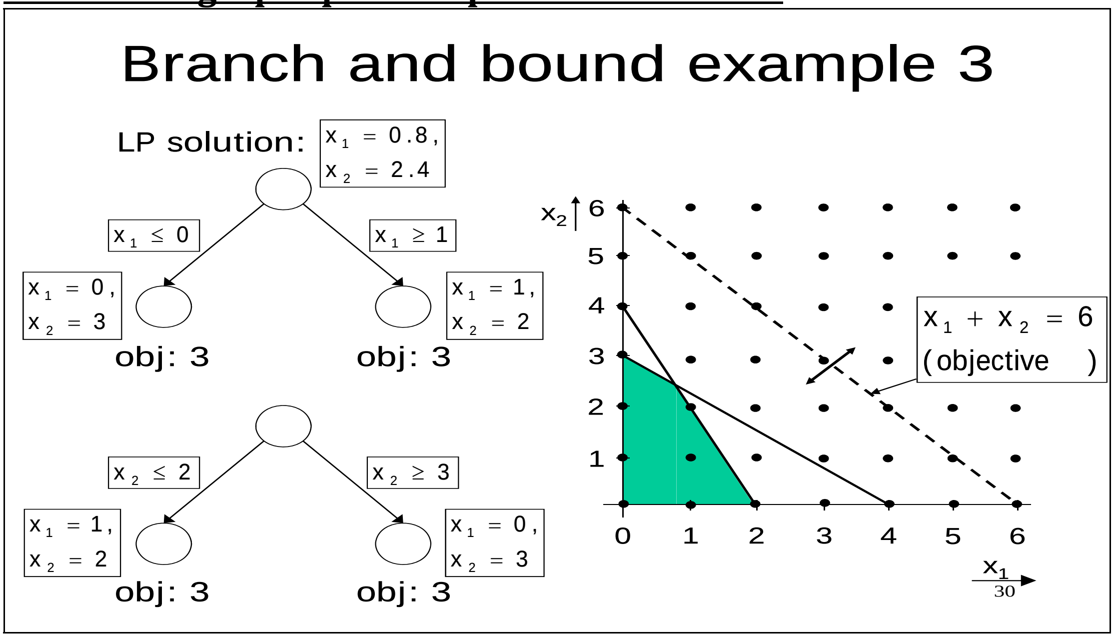
NB : Il existe plusieurs façons pour résoudre ce problème

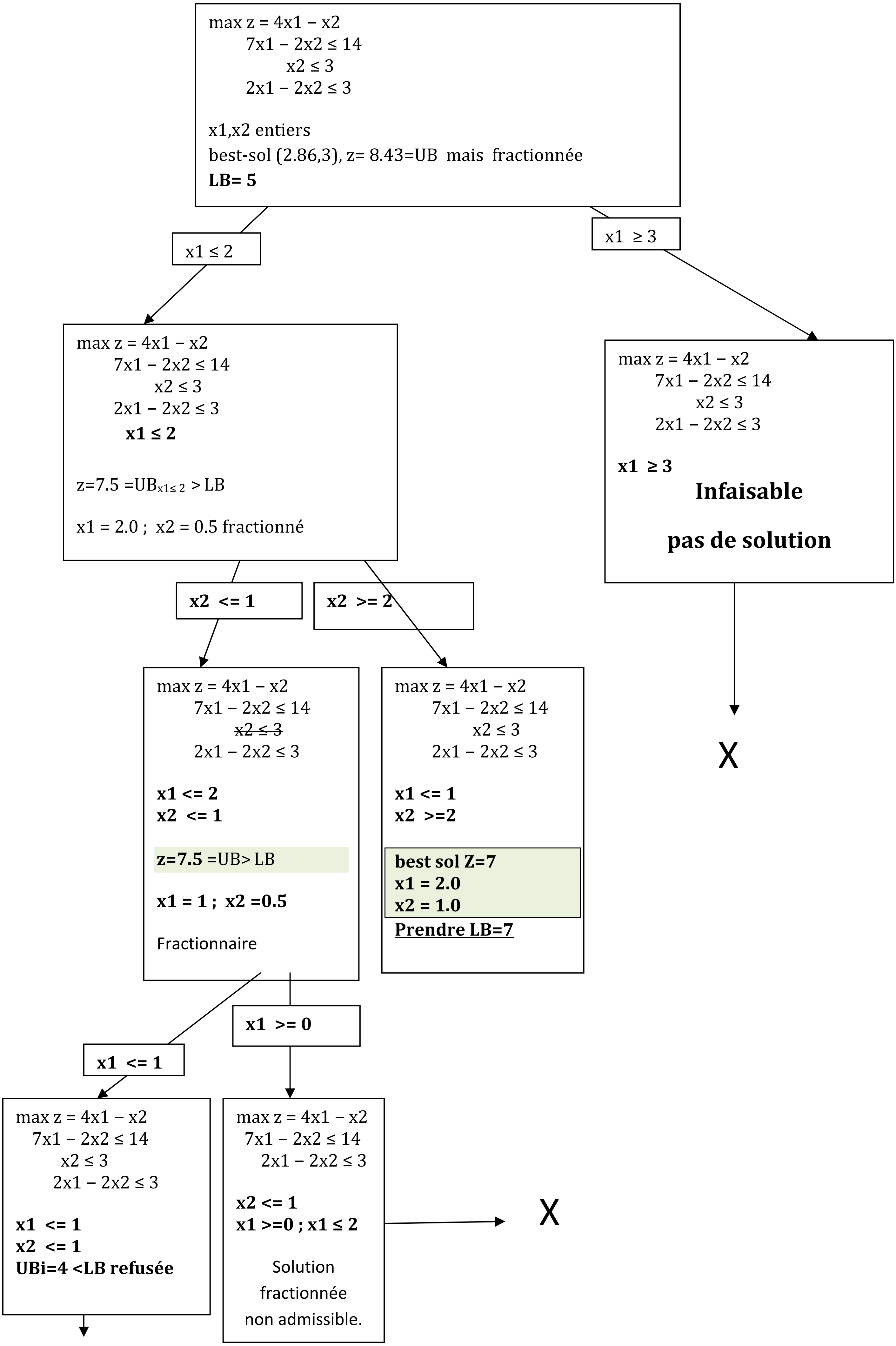
<u>Méthode de résolution générale d'un problème en PLNE par B&B</u>

1. Initialisation :
   - Calculer un LB par une heuristique qui donne des solutions entières
   - Calculer un UB par la méthode du simplex

2. choisir une variable xi et une constante xi* et ajouter la contrainte suivante xi ≥ xi*+1 (xi ≤ xi*)
3. diviser le problème selon ces deux contraintes (voir exemple)
2. Pour chaque nœud i calculer UBi de ce nœud par la méthode du simplexe
Si (UBi < LB)
Séparer ce nœud
Sinon (si UBi est fractionné)
Aller à 2
Sinon (si UBi est entier)
Mettre LB=UBi
Aller à 2

<u>Résolution graphique d'un problème en PLNE</u>




max z = 4x1 − x2
7x1 − 2x2 ≤ 14
x2 ≤ 3
2x1 − 2x2 ≤ 3
x1,x2 entiers
best-sol (2.86,3), z= 8.43=UB mais fractionnée
LB= 5
x1 ≤ 2
x1 ≥ 3
max z = 4x1 − x2
7x1 − 2x2 ≤ 14
x2 ≤ 3
2x1 − 2x2 ≤ 3
x1 ≤ 2
z=7.5 =$UB_{x1 \le 2}$ > LB
x1 = 2.0 ; x2 = 0.5 fractionné
max z = 4x1 − x2
7x1 − 2x2 ≤ 14
x2 ≤ 3
2x1 − 2x2 ≤ 3
x1 ≥ 3
Infaisable
pas de solution
X
x2 <= 1
x2 >= 2
max z = 4x1 − x2
7x1 − 2x2 ≤ 14
~~x2 ≤ 3~~
2x1 − 2x2 ≤ 3
x1 <= 2
x2 <= 1
z=7.5 =UB> LB
x1 = 1 ; x2 =0.5
Fractionnaire
max z = 4x1 − x2
7x1 − 2x2 ≤ 14
x2 ≤ 3
2x1 − 2x2 ≤ 3
x1 <= 1
x2 >=2
best sol Z=7
x1 = 2.0
x2 = 1.0
Prendre LB=7
x1 <= 1
x1 >= 0
max z = 4x1 − x2
7x1 − 2x2 ≤ 14
x2 ≤ 3
2x1 − 2x2 ≤ 3
x1 <= 1
x2 <= 1
UBi=4 <LB refusée
max z = 4x1 − x2
7x1 − 2x2 ≤ 14
2x1 − 2x2 ≤ 3
x2 <= 1
x1 >=0 ; x1 ≤ 2
Solution
fractionnée
non admissible.
X