Programmation linéaire – Résolution en nombre entier – hors PL binaire

I. Préambule :

La programmation linéaire en nombres entiers (PLNE) permet de modéliser de nombreux problèmes pratiques dans lesquels les variables doivent être entières : nombre de véhicules, personnel affecté (oui ou non) à une tâche.
Ainsi des décisions peuvent être aussi incluses sous forme de variables booléennes.
Souvent un PLNE résulte d'une formulation en termes de PL, avec la contrainte supplémentaire que les variables doivent être entières.

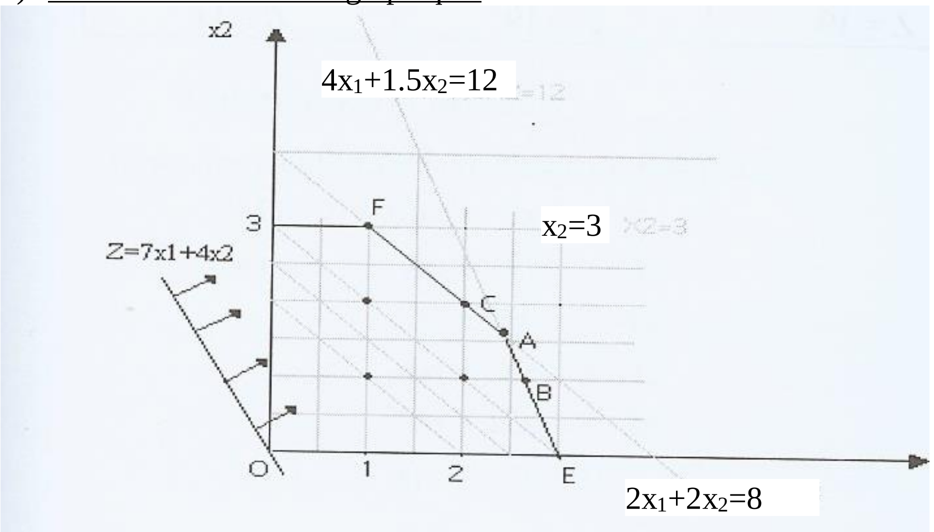
Exercice 1 :

Considérons le problème suivant :
Max $Z = 7x_1 + 4x_2$
Sous :

$$\begin{array}{rcl} x_2 & \le & 3 \\ 4x_1 + 3/2x_2 & \le & 12 \\ 2x_1 + 2x_2 & \le & 8 \end{array}$$

$x_1 \ge 0$, $x_2 \ge 0$,

1) Effectuez la résolution graphique.



2) Donnez la solution optimale en variables réelles:
Vous trouvez : x1=2.4, x2=1.6 et Z =23.2. Cette solution est donnée par le point A sur le graphique.

Le PL continu (c'est-à-dire avec des valeurs continues, donc réelles) est appelé PL relaxé.

3) Calculez la solution "arrondie" en variables entières :
Pour une maximisation, on choisit l'arrondi des variables de décision par la valeur inférieure
Pour une minimisation, on choisit l'arrondi des variables de décision par la valeur supérieure.

Quelques remarques :

- L'optimum entier n'est pas forcément un sommet du polyèdre : point C(2;2).
- "L'arrondi" de la solution continue, ici (2;1) n'est pas forcément optimal, voire ne pas être réalisable (c'est-à-dire en dehors du polyèdre des solutions admissibles).
- Le polyèdre continu peut être non vide, mais ne contenir aucune valeur solution entière.

Soient :
Z_{PLNE}, la valeur de la solution optimale du PLNE (par énumération de la fonction à optimiser pour chaque point de coordonnées entières ET appartenant au polygone des solutions admissibles)

Z_{PL}, la valeur de la solution optimale relaxée. (Calcul du simplexe en réels)

Z_A, la solution "arrondie" du PL (obtenue en arrondissant les coordonnées de la solution optimale du PL).

Dans l'exercice 1,
$Z_{PLNE} = 22$, avec le point C (2;2)
$Z_{PL} = 23{,}2$, avec le point A (2.4;1.6)
$Z_A = 18$ avec le point (2;1).

Soit α, l'écart entre la partie entière de Z_{PL} et la valeur de Z_A
Lorsque α est très petit (≤ 5%), alors on peut se contenter de la solution approchée.

A condition que cette solution soit réalisable (dans le polyèdre des solutions admissibles), ce qui est malheureusement rarement le cas.

Exercice 2 :

Max $Z = 2x_1 + x_2$

Sous :

$$\begin{array}{rcrcl} x_1 & + & x_2 & \le & 5 \\ -x_1 & + & x_2 & \le & 0 \\ 6x_1 & + & 2x_2 & \le & 21 \end{array}$$

x_1 : entier>=0, x_2 : entier>=0

1) Effectuez la résolution graphique du PL
2) Faites apparaitre toutes les solutions réalisables entières. Donnez en la liste
3) Calculez dans l'ordre : Z_{PL}, puis Z_A puis Z_{PLNE}, tout en spécifiant les points concernés.
4) Apportez votre conclusion au problème

Résolution sur Excel :
Vous pouvez résoudre le simplexe avec le solveur d'Excel comme vous avez appris à le faire avec des valeurs réelles.

La seule différence est de rajouter une contrainte qui spécifie que les variables sont entières

L'algorithme appliqué est l'algorithme de Dakin (branch and bound)

Méthode arborescente de DAKIN (technique du Branch and Bound):

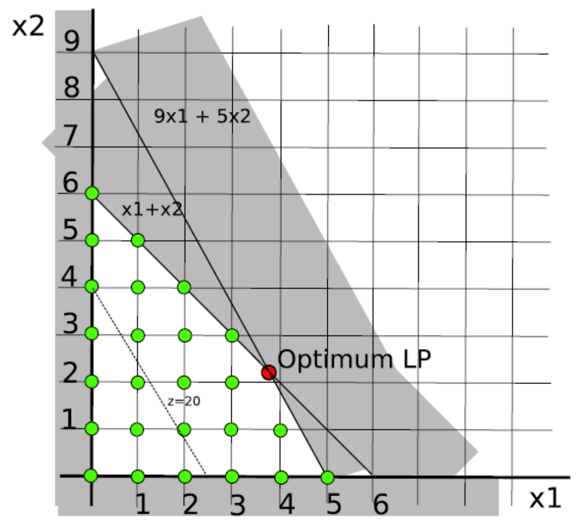
Exercice fil rouge du cours :

Une entreprise fabrique des armoires et des tables.

- Une armoire nécessite 1h de travail et 9m2 de bois
- Une table nécessite 1h de travail et 5 m2 de bois.
- 6h de travail et 45 m2 de bois sont disponibles
- Chaque armoire génère un profit de 8€ et chaque table génère un profit de 5€. Il s'agit de maximiser ce profit.

Formulez le PL en entiers :

Représentation graphique :



Probleme initial (1)

Liste des solutions entières admissibles (avec boucle sur y incluse dans une boucle sur x):

{(0;0),(0;1),(0;2),(0;3),(0;4),(0;5),(0;6),(1;0),(1;1),(1;2),(1;3),(1;4),(1;5),(2;0),(2;1),(2;2),(2;3),(2;4),(3;0),(3;1),(3;2),(3;3),(4;0),(4;1),(5;0)}

Le traitement démarre avec une résolution conventionnelle en PL relaxé.
Si la solution est entière, c'est la solution optimale du PLNE.
On obtient Z=165/4 avec x1= 15/4 et x2=9/4.
Za (solution entière arrondie qui va servir de 1ere évaluation de Z_{PLNE})

Algorithme de Dakin :

1) ***Initialisation*** : Résoudre le Z_{PL}. S'il est égal à un Z_{PLNE}, alors l'algorithme est fini. Calculez Za (initialisation), solution entière arrondie.

   La valeur obtenue servira de **valeur de référence du Z_{PLNE}** pour la 1ere évaluation.

2) **Tant qu**'il reste un nœud de l'arbre à explorer (de préférence en profondeur plutôt qu'en largeur), recommencer les traitements 3 à 7
3) ***Séparer*** pour chaque variable la partie entière de la partie fractionnaire.
4) ***Sélectionnez*** la variable qui présente la partie fractionnaire la plus grande pour une maximisation (la plus petite pour une minimisation), c'est la variable qui va être traitée.
5) Dédoublez la variable choisie en deux valeurs :
   ≤ Entière(xi) et ≥ Entière(xi) + 1.
6) ***Sélectionnez*** la branche que vous allez explorer :
   a. ***Evaluez la pertinence*** de résoudre le simplexe des deux branches. La nouvelle contrainte apportée à chaque branche rend peut-être l'une des 2 branches impropre à toute solution admissible.
   b. Résolvez par la méthode du simplexe le PL restants, ou s'il reste 2 choix, alors arbitrairement bien l'un des deux PL restant (l'un avec une contrainte de borne supérieure sur la variable choisie, l'autre avec une contrainte de borne inférieure)
7) ***Evaluez*** le résultat obtenu :
   1) Si le Z_{PL} du simplexe réel est "moins bon"" que le Z_{PLNE} de référence, abandonnez la branche courante
   2) Si (Z_{PL} est obtenu avec des valeurs xi TOUTES entières), alors l'exploration de la branche actuelle s'arrête.
      i. Si la nouvelle valeur du Z_{PLNE} est "meilleure" que la valeur du Z_{PLNE} de référence, prenez cette nouvelle solution comme nouvelle valeur de référence du Z_{PLNE}

Lexique et remarques:

- ***Branch and bound*** = Separation et Evaluation
- On dit qu'un nœud que l'on n'explore plus est ***stérilisé***, si ce nœud propose
  - 6a) : une solution non réalisable en entier
  - 7a) : une solution relaxée "moins bonne" que la solution Z_{PLNE} de référence courante
  - 7b) : une solution réalisable en entier (une nouvelle séparation n'apportera aucun amélioration)
- Le Branch and Bound n'est pas glouton, il propose des retours arrière (***backtrack***)
- L’algorithme divise pour régner, et ***converge nécessairement*** dû au fait qu’il élimine toujours des points à chaque étape de séparation, alors que ceux-ci sont en nombre fini.

Dans l'exemple :

1) Za correspond à x1=3 et x2=2 et Za = 8*3+5*2 = 34
   Constatez que alpha = 17%

   Cette valeur servira de référence minimale lors d'évaluations

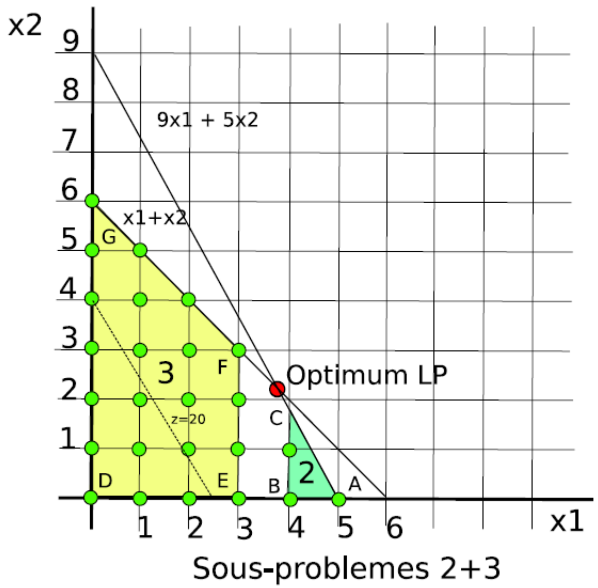
2) x1 = 12/4 + 3/4 = 3 + 3/4
   x2 = 8/4 + 1/4 = 2 + 1/4

3) Nous choisirons x1.
4) Nous avons donc x1<= 3 ou x1>= 4

<u>On a donc maintenant deux sous-problèmes :</u>

2. Problème initial + contrainte x1 >= 4
3. Problème initial + contrainte x1 <= 3

Voici la représentation graphique de la séparation en 2 sous-problèmes :



Liste des solutions entières admissibles au sous-problème 2 : {(0;0),(0;1),(0;2),(0;3),(0;4),(0;5),(0;6),(1;0),(1;1),(1;2),(1;3),(1;4),(1;5),(2;0),(2;1),(2;2),(2;3),(2;4),(3;0),(3;1),(3;2),(3;3)}

Liste des solutions entières admissibles au sous-problème 3 : {(4;0),(4;1),(5;0)}

<u>D'où problème 2 :</u>

Max $Z = 8x_1 + 5x_2$

Sous :
$$\begin{array}{rcrcl} x_1 & + & x_2 & <= & 6 \\ 9x_1 & + & 5x_2 & <= & 45 \\ x_1 & & & >= & 4 \end{array}$$

x_1 : entier>=0, x_2 : entier>=0

<u>Et problème 3 :</u>

Max $Z = 8x_1 + 5x_2$

Sous :
$$\begin{array}{rcrcl} x_1 & + & x_2 & <= & 6 \\ 9x_1 & + & 5x_2 & <= & 45 \\ x_1 & & & <= & 3 \end{array}$$

x_1 : entier>=0, x_2 : entier>=0

5) On choisit (arbitrairement) de résoudre d'abord le problème 2 :
   On obtient en PL relaxé la solution suivante :
   Z_{PL} = 41, x1 = 4 et x2 = 9/5.

7b) La valeur de x1 étant entière, l'algorithme n'est réitéré que sur x2

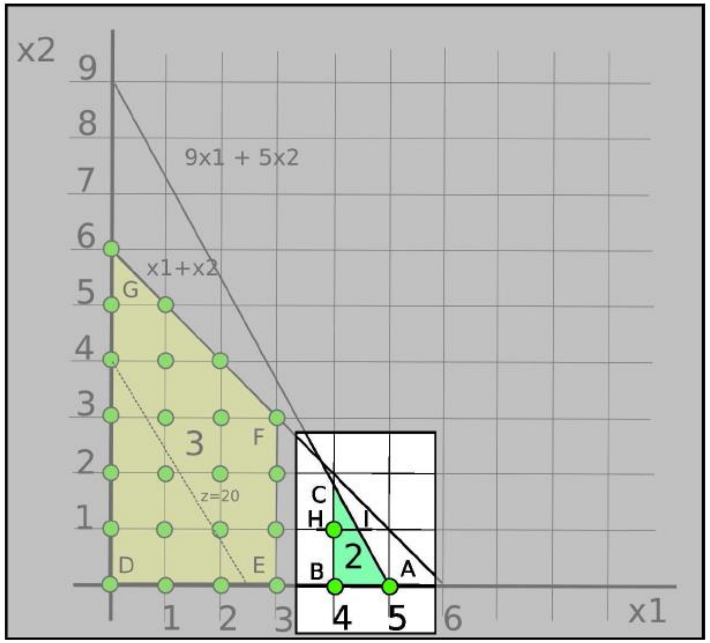
x2 = 5/5 + 4/5 = 1 + 4/5.

La contrainte se dédouble en deux nouvelles contraintes :

x2 <= 1 et x2 >=2

Nous voici avec 2 nouveaux sous-problèmes :

4. Problème 2 + contrainte x2 >= 2
5. Problème 2 + contrainte x2 <=1



On pourrait constater que le problème 4 (x2 >= 2) n'est pas réalisable.

Le PL correspondant au problème 5 (x2 <= 1) donne la solution suivante :
Z_{PL} = 365/9, x1=40/9 et x2 = 1.

Il faut donc de nouveau séparer sur x1, avec les contraintes suivantes :

6. Problème 5 + contrainte x1 >= 5
7. Problème 5 + contrainte x1 <= 4

Si l'on choisit de résoudre arbitrairement le problème 7, on obtient :

Z_{PL} = 37, x1 = 4, x2 = 1

7a) C'est une solution candidate réalisable, car de borne inférieure au PL initial, et de plus en valeurs entières.
La solution générée par le problème 6 n'est cependant pas optimal.
Pas d'algorithme supplémentaire pour cette branche.

Si l'on résout le problème 6, on obtient :

Z_{PL}= 40, avec x1= 5 et x2 = 0.

7a) C'est aussi une solution candidate réalisable, car de borne inférieure au PL initial, et de plus en valeurs entières.
La solution générée par le problème 6 n'est cependant pas optimal.
Pas de cycle supplémentaire pour cette branche.

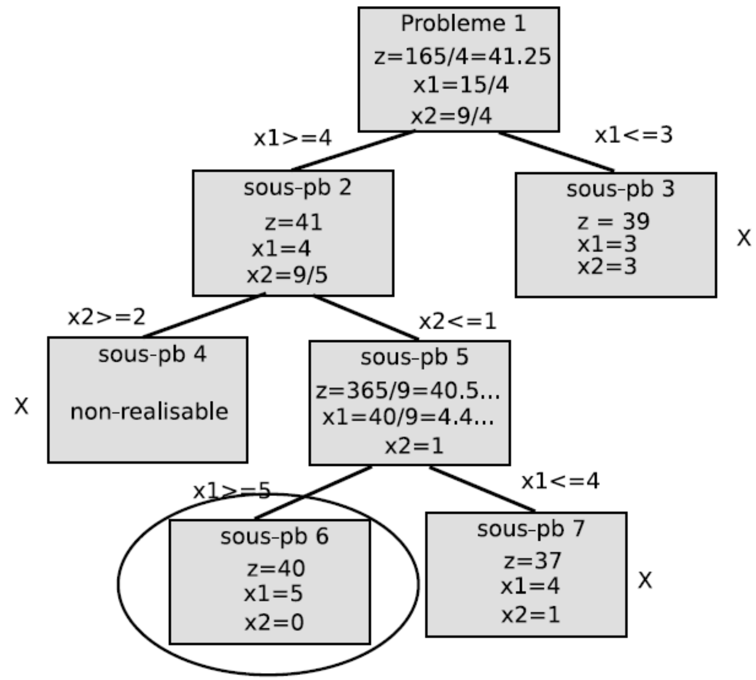
Il reste à évaluer le problème 3.
On obtient :
Z_{PL} = 39, x1 = 3 et x2 = 3.

7a) C'est aussi une solution candidate réalisable, car de borne inférieure au PL initial, et de plus en valeurs entières.
La solution générée par le problème 3 n'est cependant pas optimal.
Pas de cycle supplémentaire pour cette branche.

1) Il ne reste plus de nœud de l'arbre à explorer, on a donc trouvé notre optimum en nombre entier :

C'est à dire, fabriquer 5 armoires et 0 tables pour 40€ de bénéfice

Voici l'arborescence des traitements effectués :



Traitement à faire sur Excel :
Créez un onglet pour la situation initiale (problème 1)
Résolvez le PL relaxé
Vérifiez que vous ne vous trouvez pas sur un nœud stérilisé (cf page 3/6), ce qui stopperait toute extension de l'algorithme à partir du nœud actuel.

Conformément à l'algorithme de Dakin,
Créez un onglet pour chacun des sous-problèmes 2 et 3
Résolvez le PL relaxé à chaque fois.
Vérifiez que vous ne vous trouvez pas sur un nœud stérilisé (cf page 3/6), ce qui stopperait toute extension de l'algorithme à partir du nœud actuel.

Etc …

Vous obtiendrez alors autant de feuilles de calcul que de nœud sur l'arborescence illustrant l'algorithme de Dakin.

Exercice 2bis :

Nous reprenons **l'exercice 2** précédemment traité :

Max Z= 2x + y

Sous :	x	+	y	<=	5
	-x	+	y	<=	0
	6x	+	2y	<=	21

x : entier >=0, y : entier >=0

Nous avions trouvé x1 = 11/4, x2= 9/4 et Z_{PL} = 31/4.

1) Effectuez le traitement arborescent de DAKIN en explicitant chaque calcul
2) Effectuez le traitement arborescent de Dakin à l'aide du solveur d'Excel.