Les méthodes exactes

- L'objectif de ces méthodes est de trouver systématiquement la solution optimale du problème d'optimisation visé et de prouver que cette solution est bien optimale, c'est à dire que toutes les autres solutions sont plus mauvaises.
- Il existe différentes familles de méthodes exactes : Branch & Bound, Branch and Cut, Génération de Colonnes, Programmation Par Contraintes, ...
- Ces méthodes, soumises à l'explosion combinatoire, peuvent prendre des temps d'exécution très très longs.
- La plupart de ces méthodes sont basées sur une recherche arborescente où le problème original est progressivement divisé en problèmes plus simples (stratégie "Divide and Conquer").

Le branchement - Branching strategy

Un noeud de l'arborescence de recherche n'a pas besoin d'être séparé (divisé en plusieurs sous problèmes appelés ses fils) dès lors :

- qu'il ne comporte plus aucune solution réalisable - une des contraintes ne peut plus être satisfaite,
- qu'il comporte une seule solution ou que la solution optimale du noeud est connue $->$ éventuelle mise à jour de la valeur de la meilleure solution,
- que son évaluation (voir la suite) est plus mauvaise que la valeur de la meilleure solution connue.

Dans tous les autres cas, il faut"brancher" et dans la majorité des cas, ce branchement se fait sur une variable (encore faut-il décider laquelle) et est dichotomique (2 branches). Lorsque la variable est boléenne, cette dichotomie est naturelle. Lorsque la variable est entière et faiblement bornée, on peut envisager un branchement polytomique (plus de 2 branches).

La fonction d'évaluation

Une fonction d'évaluation s'applique à un sous ensemble de solutions $S' \subseteq S$. Cette fonction notée g doit vérifier les propriétés suivantes (problème de minimisation)

- $\forall s \in S',\ g(S') \leq f(s)$ (on parle de borne inférieure)
- Si S' se réduit à une solution unique s (feuille de l'arborescence), Alors $g(S') = f(s)$
- Si $S'' \subseteq S'$ Alors $g(S'') \geq g(S')$.

L'élagage

Un sous ensemble de solutions S' est dit stérile si on connait une solution réalisable s^*, pas forcément optimale, mais au moins aussi bonne que toute solution de S'.

Un tel sous-ensemble ne pouvant contenir une solution meilleure que s^* n'a pas besoin d'être étudié plus en détail et peut donc être **élagué** (on coupe la branche).

Si $g(S') \geq val(s^*)$, (*problème de minimisation*) alors on est sur que S' est stérile.

Plus on parvient à élaguer d'importantes branches, plus la procédure B&B est efficace.

Algorithme Branch and Bound

A : liste des noeuds en attente
BS : valeur de la meilleure solution connue
S_0 : racine de l'arborescence = probleme general
Tant que $A \neq \emptyset$ faire
  SELECTIONNER S, le noeud de plus grande priorite de A
  SEPARER S : soient S_i ses fils
  Pour chaque S_i, faire :
    Si S_i est une solution (feuille)
      ALORS Si (val(S_i) < BS)
        BS = Val(S_i) et **ELAGUER** A
      SINON
        EVALUER S_i : $g(S_i)$
        Si (g(S_i) < BS) Alors Inserer S_i dans A
        (sinon Elaguer S_i)

Stratégies de recherche

A partir du moment où plusieurs noeuds sont en attente, lequel développer en premier = SELECTIONNER le noeud de plus grande priorité
=> Mode de parcours de l'arborescence de recherche.

- **Profondeur d'abord** : priorité à l'un des noeuds les plus profonds + Backtrack.
  Optimal en mémoire - une seule branche mais éventuellement beaucoup de noeuds finalement inutiles parcourus.
- **Meilleur d'abord** : priorité au noeud de meilleure évaluation.
  Optimal en nombre de sommets parcourus mais beaucoup de noeuds à stocker.
- **Largeur d'abord** : peu utilisé par le B&B mais par d'autres méthodes arborescentes.

Obtention de bornes - Relaxation simple

Soit un problème quelconque d'optimisation discrète :
Minimiser $f(x) = c^T x$
$Ax \leq b$ (contraintes "sympas")
$Dx \leq e$ (contraintes "dures")
$x \in F$ (contraintes d'intégrité)

- Effectuer une relaxation consiste à relacher (omettre) certaines contraintes et à résoudre optimalement le problème restant.
- Relacher les contraintes d'intégrité (accepter que les variables puissent prendre des valeurs réelles) = Relaxation continue. Si le problème est un PL, on peut alors utiliser la méthode du simplexe ou faire appel à un solveur.
- En relachant une contrainte, on élargit le domaine d'admissibilité => obtention d'une borne inférieure que l'on peut utiliser dans une procédure Branch & Bound.
- La plupart du temps, ce n'est qu'une borne dans la mesure où la solution du problème relaché ne satisfait pas la contrainte relachée. Si par chance elle la satisfait, on a obtenu la solution optimale du problème de départ.
- La borne obtenue peut être très imprécise.

Relaxation continue

De nombreuses applications pour lesquelles on a :

- Une fonction économique linéaire
- Des contraintes linéaires elles aussi
- Des variables qui doivent être entières ou booléennes

Le champ applicatif est élargi par des *techniques de linéarisation* qui permettent de transformer certaines fonctions non linéaires en fonctions linéaires au prix de l'ajout de quelques variables et contraintes.

Dans ces cas, la relaxation consiste à considérer les variables réelles, positives et à résoudre le Programme Linéaire. Mais le plus souvent, dans la solution optimale, les variables ne sont pas entières ou booléennes (solution non réalisable)

B&PL : Exemple

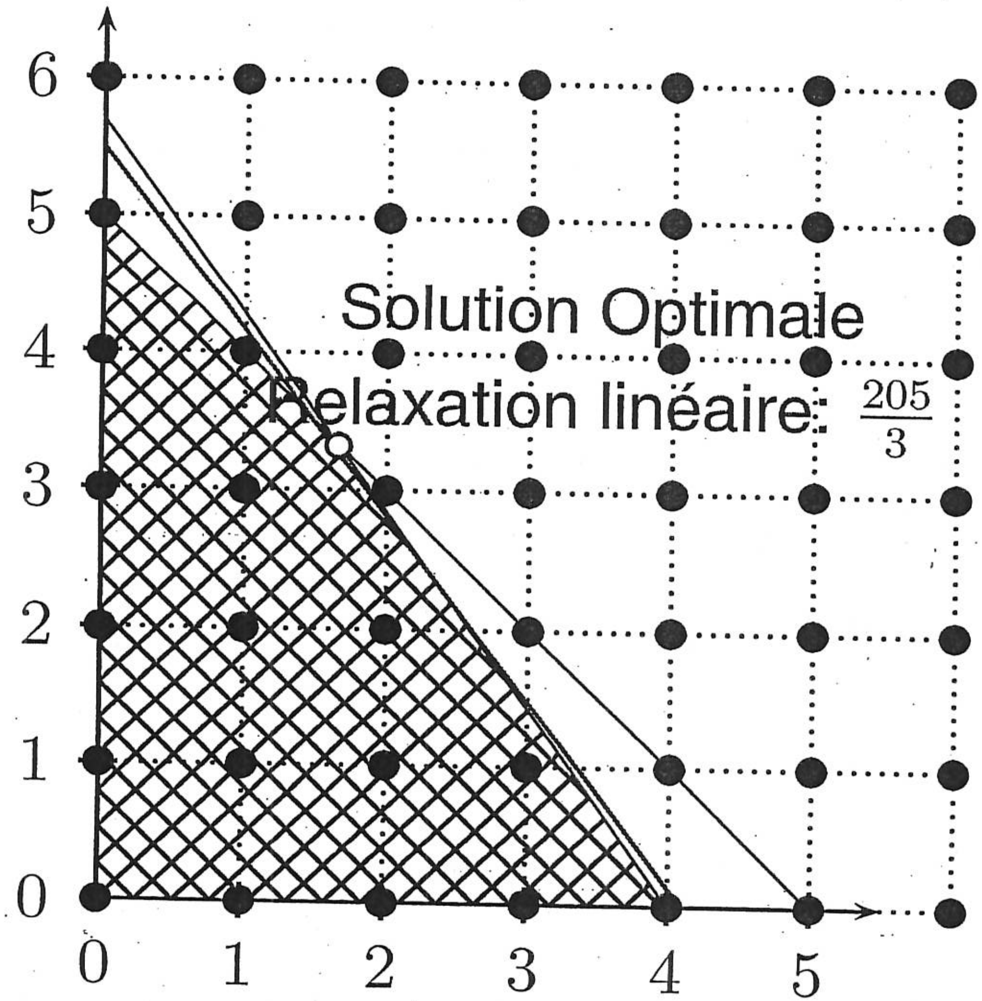
Résolution exacte du problème modélisé par le PL suivant :

$$
\begin{array}{llcll}
\text{max} & 17x_1 & + & 12x_2 & \\
\text{s.c.} & 10x_1 & + & 7x_2 & \leq 40 \\
 & x_1 & + & x_2 & \leq 5 \\
 & x_1 & , & x_2 & \geq 0 \\
 & x_1 & , & x_2 & \mathit{entiers}
\end{array}
$$

opt : $x_1 = 5/3$ $x_2 = 10/3$

BPL-Ex : Graphiquement ?

Relaxation linéaire :



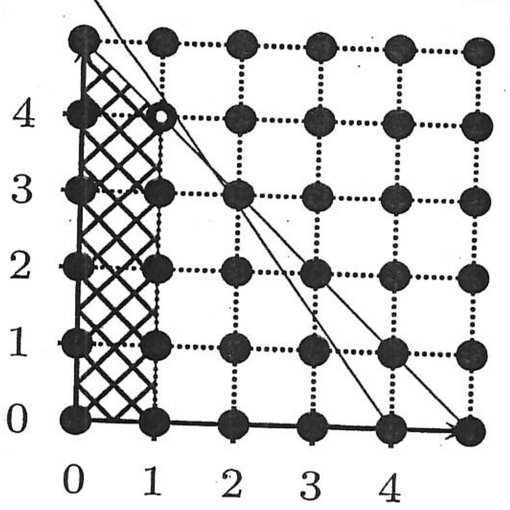
BPL-Ex : Arbre

$R : F(1.6, 3.3) = 68.33$

$x_1 \leq 1$ → P_1 :???

$x_1 \geq 2$ → P_2 :???

$$
\begin{array}{llllll}
\text{max} & 17x_1 & + & 12x_2 & & \\
\text{s.c.} & 10x_1 & + & 7x_2 & \leq 40 & \\
 & x_1 & + & x_2 & \leq 5 & \\
 & x_1 & & & \leq 1 & \\
 & & & & & \\
 & x_1 & & & = 1 & \\
 & & & x_2 & = 4 &
\end{array}
$$



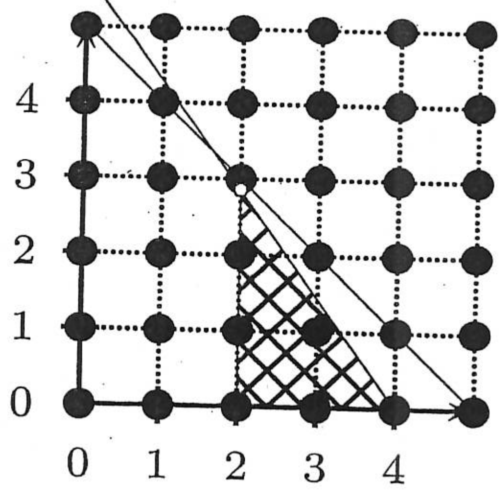
BPL-Ex : Arbre

$R : F(1.6, 3.3) = 68.33$

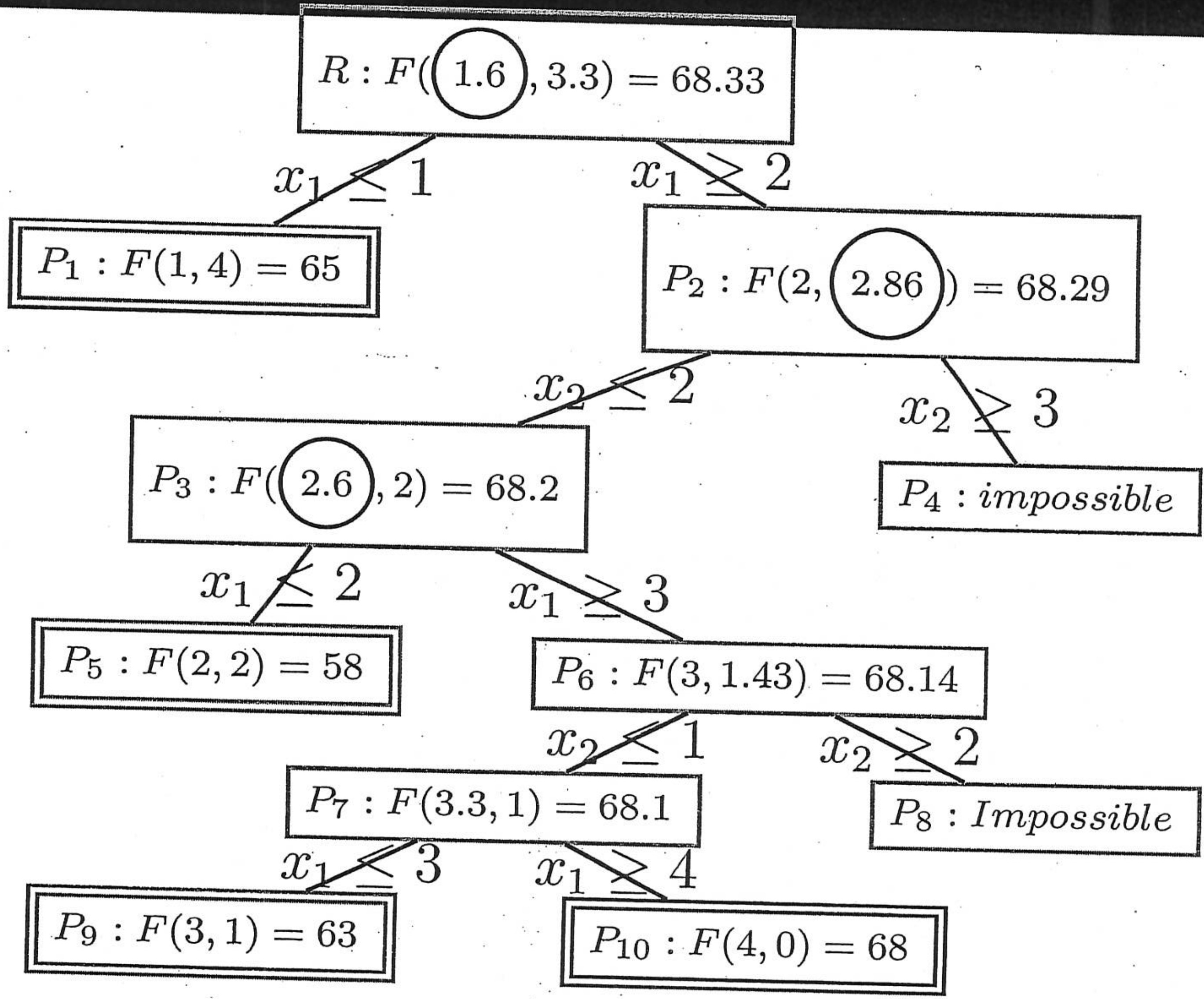
$x_1 \leq 1$ → $P_1 : F(1,4) = 65$

$x_1 \geq 2$ → P_2 :???

$$
\begin{array}{llllll}
\text{max} & 17x_1 & + & 12x_2 & & \\
\text{s.c.} & 10x_1 & + & 7x_2 & \leq 40 & \\
 & x_1 & + & x_2 & \leq 5 & \\
 & x_1 & & & \geq 2 & \\
 & & & & & \\
 & x_1 & & & = 2 & \\
 & & & x_2 & = 2.86 &
\end{array}
$$



BPL-Ex : Arbre



$R : F(1.6, 3.3) = 68.33$
$x_1 \leq 1$
$x_1 \geq 2$
$P_1 : F(1,4) = 65$
$P_2 : F(2, 2.86) = 68.29$
$x_2 \leq 2$
$x_2 \geq 3$
$P_3 : F(2.6, 2) = 68.2$
$P_4 : impossible$
$x_1 \leq 2$
$x_1 \geq 3$
$P_5 : F(2,2) = 58$
$P_6 : F(3, 1.43) = 68.14$
$x_2 \leq 1$
$x_2 \geq 2$
$P_7 : F(3.3, 1) = 68.1$
$P_8 : Impossible$
$x_1 \leq 3$
$x_1 \geq 4$
$P_9 : F(3,1) = 63$
$P_{10} : F(4,0) = 68$