Programmation Linéaire en Nombres Entiers (PLNE)

- Toutes les variables sont obligées de prendre des valeurs entières

- En général les coefficients $a_{i,j}$ sont aussi entiers

- Et, donc, on peut se limiter à des constantes b_j entières

- Si, les variables sont aussi contraintes d'être entre 0 et 1, elles n'ont que les deux valeurs possibles 0 et 1 et on parle de $PL\ en\ 0-1$

Maintenant on peut exprimer des contraintes telles $x_1 >= 1\ ou\ x_2 >= 1$
par la contrainte $x_1 + x_2 >= 1$ parce que des "solutions" comme $x_1 = x_2 = 0.5$ sont interdites.

Le PL ordinaire avec les mêmes variables, contraintes et fonction objective s'appelle le *PL relaxé* du PLNE.
Sa solution optimale donne une borne (majorant pour un problème de maximisation) sur la solution optimale du PLNE.

(Mais la solution optimale du PLNE peut être très différente ou même ne pas exister.)

Aspects Théoriques

- Chercher une solution réalisable du PLNE équivaut à chercher un point dans le polytope de solutions réalisables avec tous ses coordonnés entiers

- Si le polytope est long et étroit il n'est pas évident qu'il contient de tels points

- Décider si un PLNE possède des solutions réalisables est un exemple d'un problème NP-complet une classe de problèmes pour lesquels personne ne sait s'il existe ou non un algorithme efficace (opérant en temps polynomial)

- Les meilleurs algorithmes connus sont capables de traiter des programmes avec quelques dizaines de variables
  (par comparaison, on peut traiter des PL de plusieurs milliers de variables)

- Méthodes heuristiques

Des cas où les deux programmes sont équivalents

- Pour quelques types de PL, on sait que (s'il existe une solution optimale,) il existe une solution optimale entière

- Donc, le PLNE et le PL ont la même valeur optimale
  et c'est beaucoup plus vite de résoudre le PL

- Essentiellement des problèmes sur des réseaux

- Des programmes où chaque variable est présente dans deux contraintes au maximum

- Et avec des coefficients de ± 1

Les Méthodes de "Séparation et Evaluation"

- On choisit une variable qui a une fourchette de valeurs possibles, disons $[min, max]$

- On construit deux nouveaux problèmes, un avec $[min, mid]$, et l'autre avec $[mid+1, max]$ **séparation**

- On utilise la borne sur chaque problème fournie par son programme relaxé **évaluation**

- Si cette borne est inférieure (pour un problème de maximisation) à une solution déjà trouvée, terminer cette branche

- Et bien sûr teminer la branche si le programme relaxé n'est pas faisable

- Sinon continuer

A chaque moment il y a un arbre de problèmes à résoudre et la solution optimale est ou la meilleure solution déjà trouvée ou la meilleure de celles des problèmes contenus dans l'arbre

La Méthode de Dakin pour le PLNE général

- Parcourir l'arbre en profondeur (ce qui a une bonne chance de produire une solution réalisable assez vite)

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

- Séparer sur une variable dont la valeur (dans la solution optimale du programme relaxé) est la plus proche d'un entier

- mid est cette valeur (arrondie)

La Méthode de Balas pour les PLNE en 0-1

- Considérer les cas où z (à minimiser) n'a pas de coefficients négatifs (pas de restriction; tout programme peut être exprimé de cette façon)

- Ici séparation consiste à fixer une variable soit à 0 soit à 1

- Ce qui permet à simplifier énormément les sous-problèmes

- Si un problème n'a pas de b_j négatif, 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 largeur

- 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.