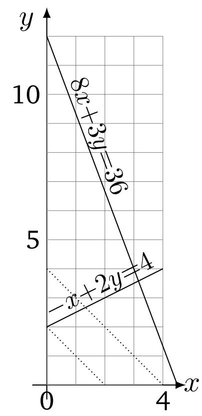
Optimisation linéaire en nombres entiers

C'est la famille des problèmes d'optimisation de la forme :

(1) Maximiser $\sum a_i x_i$ (2) sous les contraintes $\sum c_{ij} x_i \leq d_j$	où	• les a_i, c_{ij} et d_j sont des nombres (entiers ou réels), et • les inconnues sont les x_i, à valeurs entières.

On cherche un vecteur d'entiers $(x_1, x_2, \ldots)$ qui maximise l'expression (1) tout en vérifiant les contraintes (2).



Exercice 1 On considère l'instance de PLNE suivante :

$$\text{Maximiser } x + y \text{ sous les contraintes : } -x + 2y \leq 4 \text{ , } 3x + y \leq 12 \text{ , } x, y \in \mathbf{N}$$

Question 1.1 Résoudre graphiquement cette instance. Quelle est la solution *réelle* optimale ?
Question 1.2 Expliquer comment on peut résoudre un problème de ce type à l'aide d'une méthode de recherche locale (comment définit-on le voisinage ?)
Question 1.3 Donner un minimum local qui n'est pas un minimum global.

Exercice 2 Exprimer les problèmes suivants sous forme de problèmes de programmation linéaire en nombres entiers : sac-à-dos, déménageurs, coloration de graphe.

L'optimisation linéaire en nombre entiers est un problème difficile (*NP-complet*).

Remarque Les meilleurs algorithmes peuvent actuellement traiter des problèmes avec quelques dizaines de variables (quelques milliers pour l'optimisation linéaire continue).

Problème relaxé : lorsqu'on enlève la contrainte qu'on cherche des solutions entières.
Alors le problème est soluble en temps polynomial, et donne une borne (inférieure ou supérieure selon qu'on minimise ou maximise) de la solution entière.
C'est important, car cette borne permet d'accélérer la recherche d'une solution entière, en éliminant rapidement certaines solutions qui ne peuvent être optimales. Attention : arrondir la solution du problème relaxé ne donne pas la solution entière !

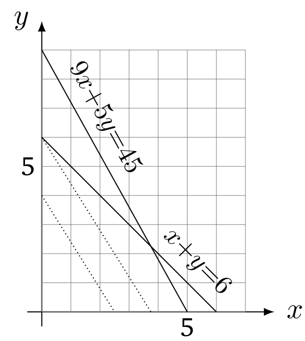
Exemple Le sac à dos, avec deux types d'objets : les uns de poids 10 et de valeur 10, les autres de poids 12 et de valeur 11. Le sac a une capacité maximale de 59, on peut prendre autant d'objet de chaque type qu'on veut, il faut maximiser la valeur totale. La solution optimale entière est très éloignée de l'optimum du problème relaxé.

Méthode générale de résolution exacte = recherche arborescente :

- on branche sur une variable, en divisant son espace de valeurs possibles en 2
- ça créée deux problèmes plus contraints. Pour chaque problème, on calcule la valeur optimale du problème relaxé. Si cette borne est moins bonne qu'une solution entière déjà trouvée, on coupe la branche. Sinon, on coupe la branche si le problème relaxé n'a pas de solution.
- Parcourir l'arbre en profondeur d'abord : 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

Exemple Soit à résoudre le problème d'optimisation ci-contre, avec $x, y \in \mathbf{N}$.
La solution relaxée est en $y = 2.25$, $x = 3.75$, la valeur de l'objectif est alors 41.25.

$$\begin{array}{l} \text{Max. } 8x + 5y \\ x + y \leq 6 \\ 9x + 5y \leq 45 \\ x \geq 0, y \geq 0 \end{array}$$



On peut séparer l'espace de recherche en deux parties, en ajoutant :

1. la contrainte $x \geq 4$, l'optimum relaxé est alors en $x = 4$, $y = 1.8$, où l'objectif vaut 41 ; on peut séparer en 2 à nouveau avec $y \geq 2$ et $y \leq 1 \ldots$ finalement on arrive à l'optimum en $x = 5$, $y = 0$, où l'objectif vaut 40.
2. la contrainte $x \leq 3$, l'optimum est alors en $x = y = 3$, où l'objectif vaut 39, on ne trouvera donc pas de meilleure solution que celle déjà trouvée.