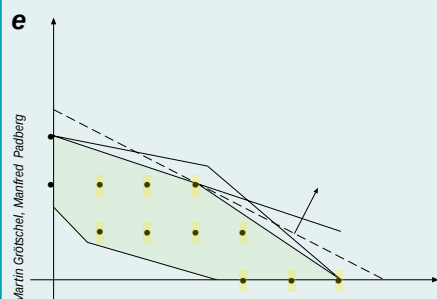
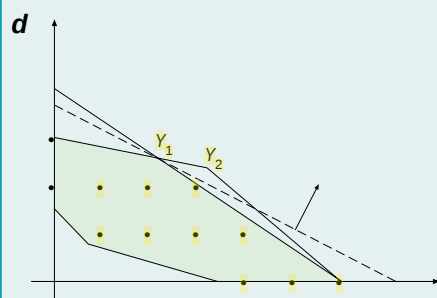
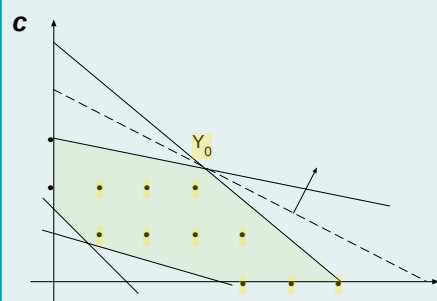
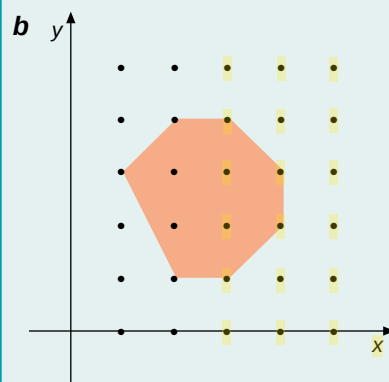
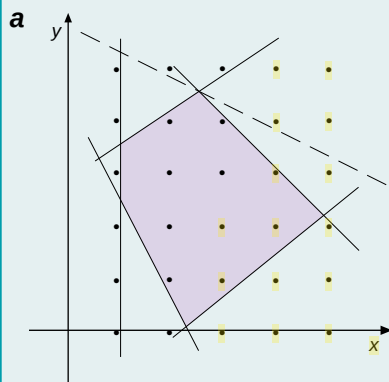


L'optimisation linéaire



La découpe des polytopes est un art difficile, car on ne sait pas où la commencer. Pour illustrer cette difficulté, examinons des exemples en deux dimensions (au lieu de trois, voire de 120). À deux dimensions, l'espace est décrit à l'aide de deux variables x et y . Une condition linéaire (exprimée par une inéquation) correspond à une droite dans le plan : tous les points situés d'un côté de la droite sont éliminés. Plusieurs conditions délimitent un polygone (dans le cas général, un polytope) qui contient les solutions acceptables du problème (a). La fonction coût est également linéaire. C'est pourquoi il existe des droites (dans le cas général, des hyperplans) sur lesquelles la fonction coût est constante. La droite en pointillés de la figure a est une telle droite : les points situés au-dessous à gauche sont tous plus chers, et les points situés à droite et au-dessus sont moins chers que ceux qui sont situés sur la droite elle-même. La solution optimale du problème (le point le moins cher du polygone) doit être un des coins de ce polygone : c'est le seul point du polygone qui rencontre la droite. Parfois l'un des côtés du polygone est parallèle à cette droite des coûts constants. En pareil cas, les deux points qui délimitent la droite, ainsi que les points intermédiaires, constituent des solutions optimales de même valeur.

Le problème linéaire qui généralise le problème initial est ici celui qui considère tous les points de l'espace, et non seulement les points de coordonnées entières (points noirs). Dans la recherche du circuit le plus court entre les 16 étapes d'Ulysse, les valeurs entières acceptables se réduisent même à 0 et à 1. Nous ne cherchons pas ici la solution optimale au problème linéaire, car elle n'est généralement pas située sur un tel ensemble de points, mais plutôt la solution optimale parmi les points situés dans la grille de points à l'intérieur du polygone. Il nous faudrait donc rectifier ce polygone de sorte qu'il ne contienne plus que les points acceptables, ainsi que tous les points situés entre eux, délimitant une enveloppe convexe qui englobe les points acceptables (b). Une telle rectification impose l'ajout de conditions supplémentaires. Malheureusement les trouver toutes est difficile, car il peut en exister un nombre gigantesque. On adopte alors une démarche itérative.

Pour identifier un faux optimum (un optimum y_0 du problème linéaire, avec des coordonnées non entières) (c), on cherche une droite (une condition exprimée par une

inéquation) qui exclut le faux optimum tout en laissant acceptables les points qui le sont. Parmi les nombreuses droites qui possèdent cette propriété, il en existe de meilleures, c'est-à-dire des droites qui découpent un gros morceau du polygone. On résout le problème linéaire à l'aide de cette condition supplémentaire (d), on trouve un nouvel optimum y_1 qui se révèle être à nouveau un faux optimum, et l'on répète le processus jusqu'à ce que l'optimum soit situé sur un point de la grille (e).

Cet algorithme de coupe est simple dans un plan, à seulement deux dimensions (déterminer la droite de coupe la plus favorable), mais il devient très difficile à mettre en œuvre dans un espace à 120 dimensions. En 1958, à l'Université Princeton, Ralph Gomory a trouvé un algorithme qui résout le problème après un nombre fini d'étapes, mais qui se révèle trop lent pour les gros problèmes. Aujourd'hui on ne cherche plus systématiquement les hyperplans de coupe, mais on se limite à ceux qui s'adaptent le mieux aux facettes du polytope.

En considérant les conditions secondaires, la solution non admissible au problème du circuit de l'*Odyssée* décrite dans l'encadré de la page 96 reste une solution non admissible, mais qui se rapproche de la solution admissible la plus courte (la solution optimale, f) et s'améliore encore à l'étape suivante, où quelques arêtes (en pointillés) sont affectées de la valeur 1/2 (g). L'étape suivante conduit à la solution optimale (h).

