

Malheureusement ce minimum n'est que celui du problème généralisé, et non celui du problème initial. En général, le vecteur trouvé par la méthode du simplexe n'a pas de coordonnées entières, et l'on doit remettre en action le couteau à fromage pour transformer le polytope du problème linéaire en polytope du problème initial. Pour ce travail supplémentaire, un coup de couteau est insuffisant. Pis encore, on ne sait pas comment retailler le polytope du problème linéaire. Les programmes linéaires explicites, pour la résolution d'un circuit, sont aujourd'hui limités à 9 villes. Plus précisément, le polytope associé au problème du voyageur de commerce à 9 villes est défini par 9 équations et par 42 104 442 inéquations, avec 36 variables qui correspondent aux 36 possibilités de liaison directe entre 9 sommets. Pour des polytopes avec 8, 7 et 6 villes, on obtient respectivement 194 187, 3 437 et 100 inéquations à 28, 21 et 15 variables. Pour 10 villes, le nombre d'inéquations atteindrait au moins 51 043 900 866 (cela reste à démontrer).

Pour des problèmes encore plus complexes, on ne connaît pas, même théoriquement, le jeu d'équations qui définit le polytope, et, même si on le connaissait, il serait peu utile, car il est beaucoup trop gros. Enfin, pour des villes encore plus nombreuses, on ne pourrait même pas trouver ce jeu d'équations. Le nombre des inéquations pour le problème des 16 villes de l'Odyssée serait, en tout cas, vertigineux.

On peut trouver une inéquation linéaire qui distingue le minimum trouvé de celui du problème étudié. Quand on ajoute cette inéquation au système d'inéquations déjà trouvé, on obtient un nouveau minimum, et, si l'on a de la chance, ce point peut même appartenir au polytope réel : on dispose alors d'une solution minimale du problème initial. Cependant, en général, ce point n'a pas de coordonnées entières, ce qui impose de répéter le processus.

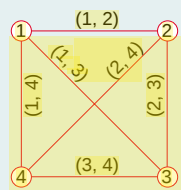
Une autre difficulté s'ajoute : généralement on ne peut pas se fonder sur le polytope des points admissibles du

problème d'optimisation, car on devrait tenir compte de toutes les contraintes, qui seraient également en nombre considérable dès que le nombre de villes est élevé. Paradoxalement on peut toutefois résoudre beaucoup de problèmes plus complexes par la méthode de la programmation linéaire. Comment cela est-il possible ?

Remplaçons à nouveau un problème par un autre, plus vaste, mais plus simple : parmi le nombre considérable de contraintes, ne retenons que certaines d'entre elles (nous découpons le morceau de fromage de manière grossière). Nous identifions d'abord un vecteur solution, c'est-à-dire le coin optimal du trop grand polytope, puis nous l'améliorons en introduisant une contrainte qui permet de couper un morceau aussi grand que possible ; on répète l'opération plusieurs fois.

Dans le cas du problème d'Ulysse, nous pourrions d'abord commencer par utiliser, comme contrainte, que chaque ville ne figure que dans deux

Les circuits et l'optimisation linéaire



On résout parfois des problèmes en les compliquant d'abord. Considérons quatre villes 1, 2, 3, 4. Le graphe complet, pour ces quatre villes, a six arêtes, repérées par les numéros des villes qu'elles relient (le plus petit numéro figure toujours en premier). On peut caractériser chaque circuit en attribuant aux arêtes qui appartiennent au circuit la valeur 1, et la valeur 0 aux autres arêtes :

CÔTÉ	(1,2)	(1,3)	(1,4)	(2,3)	(2,4)	(3,4)
	1	0	1	1	0	1
	1	1	0	0	1	1
	0	1	1	1	1	0

Avec ce codage, chaque circuit peut être considéré comme un vecteur d'un espace à six dimensions où deux valeurs sont possibles pour chaque dimension. De même, nous pourrions représenter l'Odyssée par un vecteur de 120 composantes, dont 16 seraient un chiffre 1, et les autres un chiffre 0. Inversement, tous les vecteurs ayant 16 composantes égales à 1 et 104 composantes égales à 0 ne sont pas des circuits. Parmi les 16 arêtes qui relient chaque ville, seulement deux peuvent être affectées de la valeur 1 : le chemin par lequel Ulysse arrive à cette ville, et le chemin par lequel il en sort. Cette condition s'exprime sous la forme d'une équation linéaire : la somme de toutes les composantes du vecteur qui appartiennent aux arêtes correspondant à une ville doit être égale à deux. Un vecteur qui remplit cette condition ne correspond pas

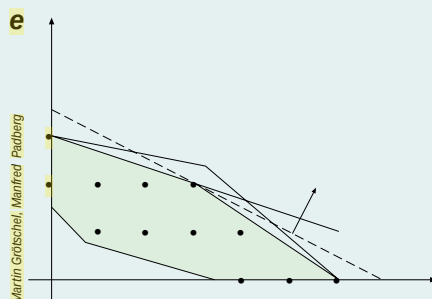
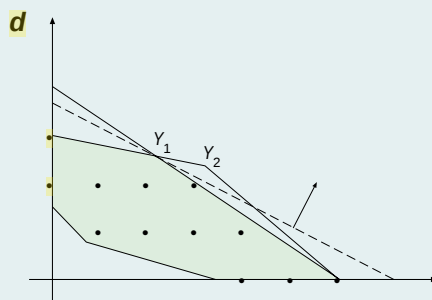
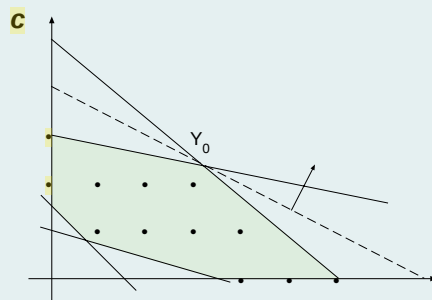
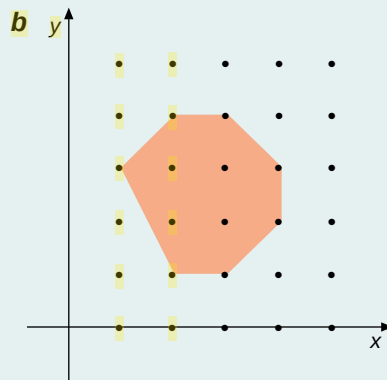
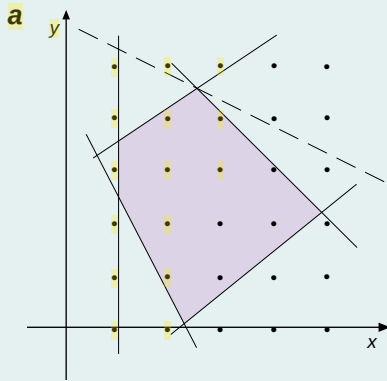


toujours à un circuit : les arêtes choisies pourraient constituer plusieurs circuits partiels (voir la figure en haut à droite) plutôt qu'un grand circuit. D'autres contraintes sont nécessaires.

Si l'on considère les sept villes du circuit partiel le plus à l'Ouest, on observe que toutes les arêtes qui relient ces villes sont affectées de la valeur 1 dans le vecteur. Au moins un des ces chiffres 1 est de trop, car sept arêtes reliant sept villes, chaque ville n'étant associée qu'à deux arêtes, constituent nécessairement un circuit fermé, ce qui exclut un circuit passant par les 16 villes. Plus généralement, on peut formuler cette condition de la façon suivante : pour chaque groupe de k villes, $k-1$ arêtes au plus entre ces villes pourront être affectées de la valeur 1. Cette condition se formalise par une inéquation linéaire : la somme de toutes les composantes associées à un groupe de k villes est nécessairement inférieure ou égale à $k-1$.

Chaque groupe d'au moins trois villes conduit à une telle condition, ce qui, pour n villes, donne $2^n - 1$ conditions (et non 2^n , car la condition d'exclusion pour un groupe est identique à celle du groupe complémentaire). Pour de grands nombres de villes, un tel nombre de conditions est bien au-delà de ce qui est praticable.

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).

