

être aussi proche que possible de celle du problème initial. Dans le cas de l'*Odyssée*, une généralisation qui consiste à supposer toutes les distances entre les villes égales à la plus petite des distances entre villes conduit à une borne inférieure.

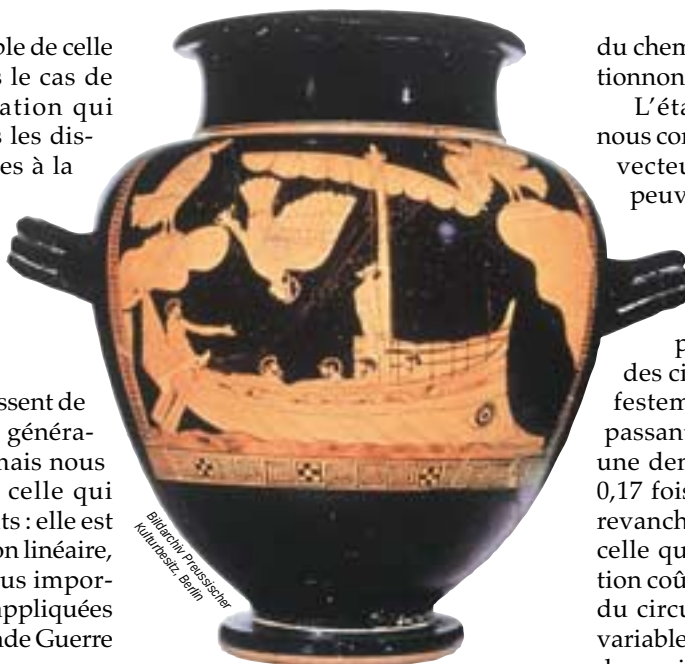
La programmation linéaire

Les mathématiciens connaissent de nombreuses techniques de généralisation (ou d'inclusion), mais nous ne considérerons ici que celle qui donne les meilleurs résultats : elle est fondée sur la programmation linéaire, l'une des techniques les plus importantes des mathématiques appliquées découvertes depuis la Seconde Guerre mondiale (voir *Gestion et programmation linéaire*, par Robert Bland, *Pour la Science*, août 1981). Son nom est trompeur, car elle n'a pas de rapport direct avec l'informatique ; c'est plutôt une optimisation de problèmes linéaires. On l'emploie quand on veut répartir de façon optimale des ressources limitées ; elle résout des problèmes de planification dans les compagnies de transport aérien, dans les raffineries de pétrole, dans les usines automobiles...

Pour inclure le problème d'Ulysse dans un problème que l'on puisse étudier par les techniques de programmation linéaire, on commence par dresser la liste des 120 liaisons directes entre les 16 villes, dans un ordre quelconque, et l'on affecte chaque liaison de la valeur 1 si Ulysse les utilise au cours de son circuit, de la valeur 0 dans le cas contraire. Ainsi chaque circuit est décrit par un ensemble de 120 chiffres binaires, ce que l'on nomme un vecteur dans un espace à 120 dimensions. Autrement dit, un circuit est associé à un point dans un espace à 120 dimensions.

Inversement, tout point de l'espace à 120 dimensions ne correspond pas à un circuit : il faut que 16 des coordonnées du point soient égales à 1, et que les 104 autres soient égales à 0.

Parmi les 653 837 184 000 circuits, comment choisir ceux dont la longueur totale est minimale ? Calculons d'abord leur longueur en multipliant chaque coordonnée (0 ou 1) par la longueur



4. QUAND ON SE CONTENTE de choisir la solution la plus directement séduisante, on ne parvient pas toujours à l'objectif fixé. Le «prévoyant Ulysse», lui, voyait loin : afin d'échapper au chant des sirènes, il demanda à ses compagnons de l'attacher au mât de son navire, tandis que ses compagnons avaient les oreilles bouchées à la cire (le vase qui représente cette scène date de 450 avant notre ère).



5. TEL LE CYCLOPE Polyphème aveuglé par Ulysse, les algorithmes de recherche de circuits n'ont pas de vue d'ensemble du problème, et ils ne progressent qu'à tâtons. Pour des raisons de gain de temps, ils ne vérifient pas, chaque fois, que chaque contrainte est respectée, et ils doivent accepter de faire parfois fausse route. C'est ainsi qu'Ulysse échappa à Polyphème en se cachant sous le ventre d'un bélier (le peintre surnommé le Maître de Sapho a réalisé cette œuvre vers 510 avant notre ère).

du chemin correspondant, puis additionnons tous les produits.

L'étape décisive vient ensuite : nous considérons les composantes du vecteur comme des variables qui peuvent prendre n'importe quelle valeur comprise entre 0 et 1. On obtient ainsi des points de l'espace à 120 dimensions qui ne correspondent généralement pas à des circuits : Ulysse ne peut manifestement pas effectuer un circuit passant par les 16 villes s'il se rend une demi-fois d'Ithaque à Troie, ou 0,17 fois de Messine à Gibraltar. En revanche, cette généralisation est de celle que nous cherchions. La fonction coût cherchée, la longueur totale du circuit, est une fonction des 120 variables : toute perturbation de l'une des variables modifie la fonction coût d'une valeur proportionnelle à la perturbation, le facteur de proportionnalité étant égal à la longueur de l'étape correspondante.

On connaît un procédé efficace pour chercher le minimum d'un problème linéaire quand on impose des contraintes qui sont également linéaires : c'est l'algorithme du simplexe. Considérons tous les points (ou vecteurs) de l'espace habituel dont les coordonnées sont comprises entre 0 et 1 : ils constituent un cube dont le côté a une longueur égale à 1 et dont un sommet se trouve à l'origine de l'espace. De même, dans un espace à 120 dimensions, les vecteurs dont toutes les composantes sont comprises entre 0 et 1 forment un hypercube, à 2^{120} sommets. Une contrainte exprimée par une inéquation linéaire est comme un couteau qui élimine une partie d'un fromage : on obtient un solide limité par des faces planes.

Dans l'espace à 120 dimensions, les points qui vérifient ainsi les contraintes forment un polytope (un volume limité par des hyperplans, tout comme un polyèdre est limité par des plans) convexe. Le minimum d'une fonction coût linéaire sera toujours dans un coin (ou dans plusieurs coins) du polytope. L'algorithme du simplexe examine systématiquement, mais en restant myope, les coins du polytope jusqu'à ce qu'il soit bloqué : le coin atteint est le minimum cherché (ou l'un des divers minima possibles).

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.

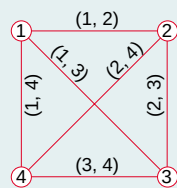
Une autre difficulté s'ajoute : généralement on ne peut pas se fonder sur le polytope des points admissibles du

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.