

rend dans la plus proche des villes voisines encore non visitées ; puis, quand toutes les villes ont été visitées, on revient au point de départ.

Quand on applique cette stratégie à des villes quelconques, à l'aide de papier et de crayon, on fait rapidement deux observations : le résultat dépend beaucoup du point de départ, et la solution obtenue est facile à améliorer. Ainsi, lorsque deux arêtes se croisent, on gagnera à remplacer la partie de circuit composée des quatre sommets liés par ces deux arêtes par un circuit où les arêtes sont des bords opposés du quadrilatère. Parfois on gagne même à remplacer les deux côtés du quadrilatère par les autres côtés, et d'autres améliorations de ce type sont possibles.

Pour perfectionner l'algorithme de recherche du circuit le plus court, on doit ainsi corriger les effets néfastes de la myopie. Si l'on s'obstinaît à améliorer les solutions en ne considérant que les solutions voisines, on risquerait de rester piégé dans un minimum local, c'est-à-dire une solution qui n'est pas particulièrement

bonne, mais que les règles d'évolution ne permettent pas d'améliorer (de même, si on laisse rouler une bille sur une tôle déformée, elle ira se loger dans un creux, mais, si elle n'a pas assez d'énergie pour en ressortir, elle risque d'y rester même si le creux n'est pas le point le plus bas de la tôle). On devra donc utiliser des perturbations qui conduiront assez loin de la solution identifiée à chaque étape. De telles méthodes ont souvent des noms amusants, tels que méthode de Monte-Carlo, recuit simulé, évolution, algorithme du déluge...

La programmation de ces méthodes est simple quand les problèmes sont de complexité moyenne. Aussi les problèmes d'optimisation ont-ils été souvent examinés par des amateurs, qui n'ont pas toujours obtenu de bons résultats. Pour étudier des problèmes du voyageur de commerce avec des dizaines de milliers de sommets (un ordre de grandeur courant, en pratique), on doit structurer les données et mettre en œuvre des techniques informatiques raffinées, afin d'obtenir des solutions en un temps acceptable.

Simultanément les heuristiques sont restées très utiles pour la résolution de problèmes pratiques.

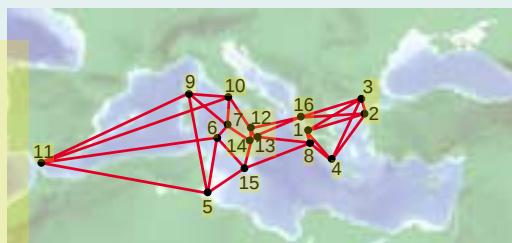
Comment détermine-t-on la borne inférieure de la fonction coût, pour un problème d'optimisation combinatoire ? L'idée fondamentale, apparemment paradoxale, consiste à simplifier le problème en le généralisant : on ajoute au problème initial un autre problème, qui possède davantage de solutions, de sorte que chaque solution du problème initial soit également une solution du problème élargi. On peut, par exemple, rendre moins strictes les conditions secondaires du problème initial, voire les supprimer.

Une solution minimale du problème élargi se caractérise par un coût inférieur à celui de toute autre solution et, par conséquent, inférieur à celui des solutions du problème initial. C'est exactement ce que l'on cherche : une borne inférieure pour le problème initial.

Évidemment le problème élargi doit être plus simple que le problème initial, et sa solution optimale doit

La fiabilité des réseaux

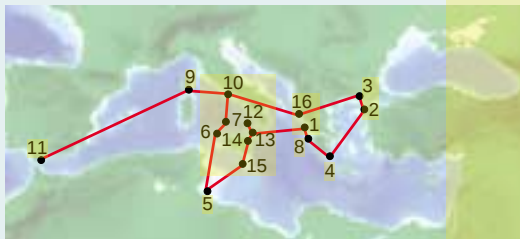
Supposons que nous devions relier les 16 villes étapes d'Ulysse par un réseau de télécommunications qui résiste aux défaillances de ses lignes. Pour des raisons historiques, nous décidons d'attribuer une priorité 1 à Ithaque (1), Troie (2), Djerba (5), Circée (10) et Messine (13) ; les autres villes auraient un niveau de priorité inférieur (0). Autrement dit, les liaisons entre les villes prioritaires devraient être préservées même en cas de panne des connexions directes entre ces villes. Si les coûts de construction sont proportionnels à la longueur des lignes, le réseau le moins coûteux a une longueur de 5 842 kilomètres (ci-dessous). Aujourd'hui les concepteurs de réseau cherchent empiriquement un équilibre raisonnable entre les coûts de réalisation et la fiabilité en modifiant les para-



Toutes les villes sont de rang 3.

mètres. Si l'on peut dépenser plus pour assurer la fiabilité du réseau, par exemple en assignant la valeur 3 à chacune des villes, le réseau le moins cher que l'on puisse concevoir est de longueur égale à 16 112 kilomètres : avec un tel réseau, les connexions subsisteraient si trois connexions de chaque ville tombaient simultanément en panne. Si le central d'une ville était en panne (par suite d'un incendie du central, par exemple), les autres villes resteraient interconnectées. Le réseau ne serait interrompu que si les villes 8 (Zakinthos) et 16 (Corfou) tombaient simultanément en panne. Dans les problèmes réels, tel celui du réseau téléphonique de l'Est des États-Unis, la solution

la plus économique serait celle de la figure en bas à droite, où les villes d'importance 0 sont représentés par un cercle, et les villes prioritaires par un carré. Un tel réseau comporte 116 villes, mais le coût de la liaison entre deux villes n'est pas toujours proportionnel à la distance qui les sépare : le franchissement des fleuves, des gorges ou des montagnes coûte très cher.



Les villes 1, 2, 5, 10 et 13 ont une priorité 1.



Réseau téléphonique de l'Est des États-Unis.

Martin Grötschel, Clyde Monma et Mechthild Stoer

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.

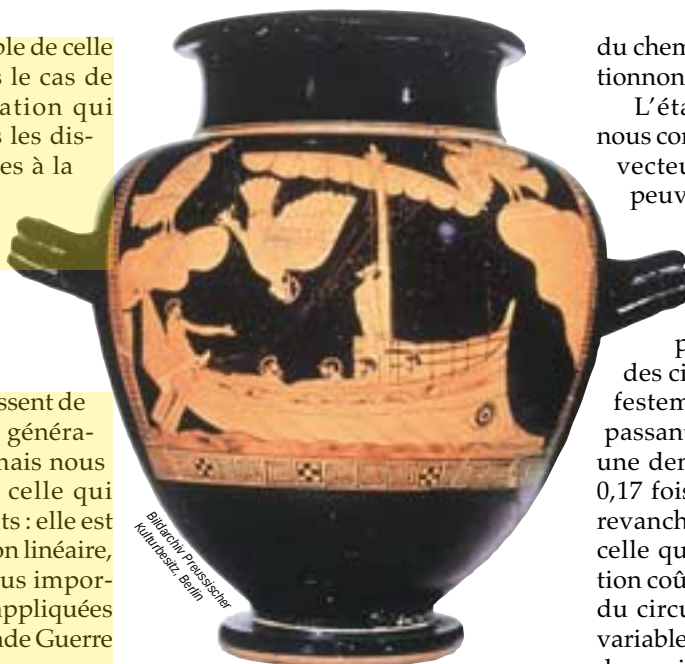
La programmation linéaire

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).