

Introduction à la théorie des graphes : Définitions, applications et techniques de preuves

Bruno Courcelle
Université Bordeaux 1, LaBRI (CNRS UMR 5800)

April 20, 2004

1 Bibliographie

Les ouvrages qui suivent peuvent se consulter en bibliothèque.

Claude Berge, Graphes et hypergraphes, Bordas 1973, (300 pages).

Nguyen Huy Xuong, Mathématiques discrètes et informatique, Masson, 1997

Aimé Sache, La théorie des graphes, Que-Sais-Je ?, 1974 ; réédition prévue en 2004 chez Cassini.

M. Kaufmann, Des points des flèches, la théorie des graphes, Dunod, Sciences-poche, épuisé.

Alan Gibbons, Algorithmic graph theory, Cambridge University Press, 1985

Reinhard Diestel, Graph Theory, Second Edition, Springer-Verlag, 2000.

Bojan Mohar, Carsten Thomassen, Graphs on surfaces, John Hopkins University Press, Baltimore, 2001.

2 Introduction

La théorie des graphes est un très vaste domaine, en évolution constante tant du point de vue des recherches fondamentales que de celui des applications. Concernant la recherche fondamentale, il faut citer les travaux de P. Seymour et de ses collaborateurs qui ont démontré récemment la Conjecture des Graphes Parfaits formulée par Claude Berge, amélioré la preuve du Théorème dit des 4 Couleurs et surtout, développé la théorie des mineurs de graphes.

D'autre part, les applications sont très nombreuses. Elles justifient une recherche importante en algorithmique. L'importance des réseaux de transport et de communication, qui sont d'ailleurs de plus en plus des réseaux évolutifs (pour

gérer les connexions des utilisateurs à des serveurs de téléphonie mobile, ou à des réseaux pair-à-pair), explique le foisonnement des problématiques.

La théorie des graphes offre d'autre part un intérêt pédagogique certain. En effet, les définitions sont simples et de véritables problèmes de recherche peuvent être posés sous forme de « jeux mathématiques » dont la formulation ludique peut recouvrir de grandes difficultés. Si comme en géométrie le support est visuel, les raisonnements y sont plus rigoureux, car il n'y a pas de risque de biais lié aux « cas de figure », et l'axiomatique de base est plus rigoureuse que celle de la géométrie au niveau élémentaire. Le raisonnement par récurrence y a une place de choix. Comme les graphes modélisent de très nombreuses situations, les problèmes posés sont des plus « naturels ». C'est ainsi que les sujets proposés par les équipes Math.en.JEANS relèvent très souvent de la théorie des graphes et plus largement de la combinatoire. Le livre de C. Berge illustre les concepts par de nombreux exemples de jeux et de problèmes concrets.

La multiplicité des applications explique également la variété des définitions, ou des variantes de définitions. Ainsi un article de théorie des graphes doit toujours « fixer les définitions », ce qui n'est pas le cas pour les groupes ou les espaces vectoriels.

Pour fixer les idées, je partirai de la définition suivante (qui n'est pas la plus générale): un *graphe* G est un couple (S, A) , où A est une relation binaire sur un ensemble S . Les éléments de S sont les *sommets du graphe*, chaque couple (x, y) dans A en est un *arc*. La terminologie n'est pas totalement fixée : Bourbaki ne s'est pas intéressé aux graphes, et maintenant c'est trop tard. Et les traductions rapides de l'anglais créent de la confusion. Ainsi, on trouve parfois les termes *nœud* ou *point* pour sommet, *arête* ou *flèche* pour arc. Ces variantes peuvent incorporer des précisions sur l'orientation, comme nous le verrons.

Une valeur, très souvent numérique mais pas exclusivement, est souvent attachée aux sommets et/ou aux arcs. La valeur d'un arc pourra être la distance entre deux villes, ou la durée d'un trajet, ou une capacité dans le cas d'un réseau de transport d'électricité ou de pétrole. L'algorithmique des graphes porte très souvent sur des graphes ainsi valués.

Il ne faut pas confondre un graphe et son dessin : un même graphe peut être dessiné de plusieurs façons. La lisibilité de la visualisation est une question importante, et le dessin de graphes est à lui seul un domaine de recherche dont les applications sont nombreuses : fabrication de cartes routières, représentation d'interactions entre sous-systèmes, aide graphique à l'ordonnancement des tâches, pour ne prendre que quelques exemples.

Un graphe est planaire s'il admet un dessin dont les arcs sont représentés par des segments de courbes qui ne se coupent pas. Un graphe planaire possède des dessins non planaires, éventuellement plus lisibles que des dessins planaires car ils peuvent faire mieux voir les symétries. Les deux graphes non planaires les plus petits sont K_5 , le graphe complet à cinq sommets, et $K_{3,3}$, le graphe complet

biparti à 3+3 sommets, dit aussi graphe des « 3 usines et des 3 maisons ». Ces deux graphes jouent un rôle important dans la caractérisation des graphes planaires.

L'un des défauts de la théorie des graphes est le foisonnement des définitions de « classes de graphes » particulières, et des résultats les concernant. On est conduit à se demander où sont les principaux résultats, quels sont les objets centraux de la théorie. Une autre difficulté est l'absence (sauf dans des cas très particuliers) de notation algébrique pour les graphes. Ainsi, en ordonnant les polynômes par degrés décroissants des monômes, on peut repérer sans peine leur égalité. Un polynôme factorisé peut être manipulé par composantes. Par contre, un graphe est toujours un objet global, dont la définition formelle peut être soit une matrice de 0 et de 1, soit une liste de sommets et de couples de sommets, soit une description par une propriété, soit un dessin, soit une méthode *ad hoc* combinant plusieurs types de définitions.

Mon objectif est de présenter quelques concepts principaux, et quelques méthodes de preuve les concernant. Les principaux thèmes abordés seront les suivants :

Chemins

Arbres recouvrants et explorations

Théorème de Menger

Coloriage et problèmes NP complets

Planarité et sous-graphes interdits

Arbres syntaxiques

Structuration des graphes et grammaires

Décompositions arborescentes et algorithmes polynomiaux

Graphes dénombrables

3 Chemins

Un peu de terminologie est nécessaire. Soit un graphe défini comme une relation binaire A sur un ensemble S . Un couple (x, y) est un arc. Un couple (x, x) dans A est une *boucle* du graphe. Le graphe est dit *non orienté* si la relation A est symétrique. Dans ce cas, on nomme *arête* l'ensemble $\{x, y\}$ où $(x, y) \in A$. On dessine les arêtes sans pointes de flèches et le dessin en est simplifié. A tout graphe *orienté* est associé son *graphe non orienté sous-jacent*, celui qui correspond à la relation $A \cup A^{-1}$. Faute de précision particulière, un graphe est orienté. Beaucoup de propriétés, la planarité par exemple, ne dépendent pas de l'orientation des arcs, et sont « invariantes par changement d'orientation ». On peut les définir comme des propriétés du graphe non orienté sous-jacent.

Beaucoup d'auteurs désignent par graphe un graphe fini non orienté sans boucle. Ils considèrent que l'on capture ainsi l'essence de la notion de graphe,

et que tout le reste est accessoire. Ceci pour indiquer quelques « pièges terminologiques ».

Un *sous-graphe* de $G = (S, A)$ est un graphe $G' = (S', A')$ où $S' \subseteq S$, et $A' \subseteq A \cap S' \times S'$. Si G est spécifié comme non orienté, alors A' doit être symétrique.

Si un graphe est défini comme une relation, la notion principale est celle de chemin. Un *chemin orienté* d'origine x et d'extrémité y dans un graphe orienté est une suite (x, \dots, y) de sommets deux à deux distincts telle que si v suit u dans cette suite, (u, v) est un arc. Si l'on omet la précision « deux à deux distincts », on a la notion de *cheminement orienté*.

Un *circuit élémentaire* est un chemin orienté (x, \dots, y) , avec la condition supplémentaire que (y, x) est un arc.

Dans un graphe non orienté, on obtient en remplaçant arc par arête les notions de *chemin*, de *cheminement*, de *cycle élémentaire*. Les cycles et les circuits non élémentaires peuvent passer plusieurs fois par un même sommet sans utiliser plusieurs fois le même arc ou la même arête. Le 8 est un exemple de cycle non élémentaire.

Un graphe non orienté est *connexe* si deux sommets quelconques sont reliés par un chemin. Un graphe orienté est connexe si son graphe non orienté sous-jacent est connexe. Un graphe orienté est *fortement connexe* si pour tout couple de sommets distincts x, y il existe un chemin orienté de x à y et un de y à x . Un sous-graphe connexe (fortement connexe) maximal est une *composante connexe* (une *composante fortement connexe*).

Les arbres sont des graphes particulièrement importants. Un *arbre* peut être défini de plusieurs façons équivalentes : comme un graphe non orienté connexe sans cycle, ou bien comme un graphe non orienté tel que deux sommets distincts sont reliés par un et un seul chemin. Un arbre est donc un graphe connexe ayant un nombre d'arêtes minimal. Si on enlève une arête on déconnecte le graphe. En tant que support d'un réseau de communication ou de transport, un arbre est donc très fragile. La moindre « panne » au niveau d'une arête déconnecte le réseau. A l'opposé, un graphe *complet*, c'est à dire un graphe dont les sommets sont tous 2 à 2 adjacents (reliés par une arête) est très solide, mais très coûteux à entretenir.

Une *arborescence* est un graphe orienté sans circuit disposant d'un sommet nommé *racine* tel que pour tout sommet x il existe un et seul chemin orienté de cette racine à x . Cette racine est unique, le graphe non orienté sous-jacent est un arbre. Pour tout sommet a d'un arbre il existe une et une seule façon d'orienter ses arêtes qui fait de ce graphe une arborescence de racine a .

Différents paramètres associés aux graphes ont été introduits pour classer et comparer les graphes. La *distance de deux sommets* est la longueur d'un plus court chemin non orienté qui les relie. Le *diamètre* d'un graphe connexe est la plus grande distance de deux sommets. Le *degré d'un sommet* est le nombre d'arêtes dont ce sommet est une extrémité. On compte une boucle pour 2. Le *degré sortant (entrant)* d'un sommet x est le nombre d'arcs dont x est l'origine (l'extrémité). Le *degré (resp. degré sortant, resp. degré entrant)* d'un graphe est le maximum des degrés de ses sommets (resp. des degrés sortants, resp. des

degrés entrants).

Des inégalités résultent de ces définitions. On peut ainsi majorer le nombre d'arcs en fonction du degré et du nombre de sommets. On peut majorer le nombre de sommets en fonction du degré et du diamètre.

Pour un graphe connexe (fini) dessiné dans le plan sans croisements d'arcs, soit C l'ensemble complémentaire du dessin du graphe. C'est une union d'ouverts connexes que l'on nomme *faces*. (L'une des faces est non bornée). On a alors la relation d'Euler : $n - m + f = 2$ où n est le nombre de sommets, m le nombre d'arcs, f le nombre de faces. La preuve est facile par récurrence sur le nombre d'arêtes, en prenant comme le cas d'un arbre comme base de la récurrence, et en ajoutant les arêtes une par une. Ainsi le nombre de faces ne dépend pas du dessin particulier. Le nombre 2 dans cette formule est caractéristique du plan. Une formule analogue est valide pour les autres surfaces (par exemple le *tore*). On peut améliorer la formule en faisant intervenir le nombre de composantes connexes. De cette formule, on peut déduire que les graphes K_5 et $K_{3,3}$ ne sont pas planaires.

La notion de longueur d'un chemin peut être généralisée en une notion de coût ou de capacité. A chaque arc ou arête on associe une valeur numérique positive. Le *coût d'un chemin* est alors la somme des coûts des différents arcs qui le composent. Si ces valeurs représentent des *capacités* (des flux maxima), c'est le minimum sur un chemin qui sera sa valeur. Le problème classique du voyageur de commerce consiste à *parcourir un graphe*, c'est à dire à passer par tous les sommets, au moyen d'un cheminement fermé (« revenant à la ville de départ »), de coût minimum. L'idéal est le cycle qui optimise ce coût : on parle alors de *cycle Hamiltonien optimal*. Déterminer l'existence d'un cycle Hamiltonien et en trouver un optimal si c'est possible est un problème difficile en ce sens que l'on ne dispose pas de méthode générale évitant d'avoir à faire des essais systématiques nombreux, qui dans des cas extrêmes peuvent nécessiter un temps exponentiel par rapport à la taille du graphe. Seul un temps de calcul polynomial peut être considéré comme raisonnable. Et encore, cela dépend de l'exposant et/ou de la constante qui peuvent être « astronomiques ». Des heuristiques, fournissant de bons résultats proches de l'optimum ont bien sûr été développées.

La définition d'un graphe comme relation permet d'en donner une représentation mathématique et informatique par une matrice. Un graphe orienté à n sommets peut être représenté par une matrice $n \times n$ notée encore A telle que $A[x, y] = 1$ s'il existe un arc de x vers y et 0 sinon.

Dans ce cas $A^p[x, y]$ est le nombre de cheminements orientés de x à y de longueur p .

Supposons maintenant que les produits de matrices soient calculés dans l'algèbre de Boole $\{0,1\}$ avec $1 + 1 = 1$, les autres calculs étant comme dans les entiers.

L'entrée (x, y) de la matrice $(A + I_n)^p$ représente la présence ou l'absence d'un chemin orienté de x à y de longueur au plus p .

La suite des matrices $(A + I_n)^p$ est croissante avec p . Elle est donc nécessairement stationnaire et le plus petit p tel que $(A + I_n)^{p+1} = (A + I_n)^p$ est inférieur à n . Cette observation reflète un principe combinatoire très utile en théorie des automates : s'il existe un cheminement de x à y dans un graphe à n sommets, il existe un chemin de x à y de longueur au plus n , que l'on peut construire par suppressions itérées des circuits inclus dans le cheminement donné.

Si chaque arc (x, y) est donné avec un coût $C[x, y]$ réel positif ou nul, on peut représenter le graphe par une matrice C à valeurs réelles et en prenant $C[x, y] = +\infty$ lorsqu'il n'y a pas d'arc de x à y , et $C[x, x] = 0$ pour tout x . Si l'on effectue les produits de matrices en prenant comme addition le minimum, et comme produit la somme (avec $x \cdot \infty = \infty$ même pour $x = 0$), on obtient en calculant C^n le tableau des coûts des chemins optimaux de x à y de longueur n .

4 Arbres recouvrants et explorations.

Soit un graphe non orienté $G = (S, A)$. Un arbre $T = (S, B)$ où $B \subseteq A$ s'appelle un *arbre recouvrant*. L'existence d'un arbre recouvrant implique que le graphe considéré est connexe. Le plus souvent, on choisit un sommet s , on explore le graphe à partir de ce sommet et on construit une arborescence dont la racine est s . On parle néanmoins "traditionnellement" d'arbre recouvrant et non d'arborescence recouvrante.

Il existe plusieurs méthodes de construction d'un arbre recouvrant.

4.1 Arbre recouvrant en largeur.

On choisit un sommet s , point de départ de l'exploration du graphe G donné, et on ajoute successivement des sommets et des arcs pour constituer l'arborescence désirée. On construit une suite de sommets $x_0, x_1, \dots, x_n, \dots$ et pour chaque $n > 0$ un arc (x_i, x_n) pour quelque $i < n$ de la façon suivante :

On définit $x_0 = s$. La liste (x_0, x_1, \dots, x_n) étant construite, on choisit le plus petit indice i tel que x_i a des voisins non dans la liste. Soient x_{n+1}, \dots, x_{n+p} ces voisins. On les ajoute à la liste et l'on ajoute à l'arborescence en cours de construction des arcs de x_i à chacun de ces nouveaux sommets. Si un tel i n'existe pas, c'est que l'on a obtenu une arborescence qui recouvre la composante connexe de x_0 , (c'est dire l'ensemble des sommets reliés par un chemin à x_0).

Propriété : La distance de x_0 à un sommet x_i est la même dans G et dans l'arbre construit. Les arcs non dans l'arbre relient des sommets de mêmes niveaux ou de niveaux consécutifs.

4.2 Arbre recouvrant en profondeur.

Le principe est le même, mais le mode d'extension d'une liste x_0, x_1, \dots, x_n est différent : on détermine i , le plus grand indice (éventuellement n) tel x_i a des voisins non dans la liste. On prend pour x_{n+1} l'un de ces voisins et on ajoute l'arc (x_i, x_{n+1}) . Si un tel indice i n'existe pas, c'est que l'on a un arbre recouvrant la composante connexe de x_0 .

Un *arbre de Trémaux* T est une arborescence recouvrante telle que tout arc ou arête du graphe qui n'est pas dans T relie deux sommets situés sur un même chemin issu de la racine.

Propriété : Tout graphe fini connexe possède un arbre de Trémaux. Les arbres de Trémaux sont les résultats des explorations en profondeur du graphe.

Pour les graphes orientés, on peut utiliser cet algorithme en prenant pour i le plus grand indice tel qu'il existe un arc $x_i \rightarrow u$ où u n'est pas dans la liste. On ajoute à la liste tel $u = x_{n+1}$ et l'arc $x_i \rightarrow u$. On recouvre ainsi l'ensemble S' des sommets accessibles par un chemin orienté à partir de x_0 , qui peut ne pas être toute la composante connexe.

Propriété : Tout arc entre deux sommets de S' qui n'est pas dans l'arborescence recouvrante relie deux sommets situés sur un même chemin issu de la racine, ou bien va de x_p à x_q avec $p > q$ (c'est un *arc transverse* de droite à gauche si l'on dessine l'arborescence en représentant de gauche à droite les successeurs d'un sommet dans l'ordre où ils sont placés dans la liste construite lors de l'exploration en profondeur).

On peut ainsi déterminer dans un graphe orienté les sommets accessibles à partir d'un sommet, l'existence de cycles (qui équivaut à l'existence « d'arcs remontants »), un tri topologique du graphe considéré s'il est sans circuit (l'inverse de l'ordre de la liste), la composante fortement connexe d'un sommet.

5 Théorème de Menger

Soit G connexe, égal à $H \cup K$ l'union de deux graphes, telle que H et K ont un seul sommet s en commun. Un tel sommet est dit *séparateur*. Soient x et y deux sommets distincts de s , l'un dans H , l'autre dans K . Tout chemin allant de x à y doit passer par s . Il n'existe donc pas de cycle élémentaire passant par x et y . (Un 8 n'est pas un cycle élémentaire). Il n'existe pas de paire de chemins disjoints reliant x à y .

La réciproque est vraie : si dans un graphe connexe, deux sommets x et y ne peuvent être reliés par deux chemins disjoints, alors, c'est qu'il existe un sommet séparateur, un goulot d'étranglement en quelque sorte. (*Preuve* : Considérons un chemin c de x à y dans G . Soit H' le sous-graphe défini comme réunion des

chemins dans G issus de x et qui ne rencontrent pas c (sauf en x). Soit K' la réunion des chemins issus de y et qui ne rencontrent pas c (sauf en y). Soit x' le sommet sur c le plus éloigné de x lié à un sommet de H' . Soit y' le sommet de c le plus loin de y lié à un sommet de K' . On a sur le chemin $c : x \leq x' \leq y' \leq y$ (car si $y' < x'$, on aurait deux chemins disjoints.) N'importe quel sommet entre x' et y' (inclus) est un séparateur de G qui sépare x et y .

Ce résultat se généralise aux ensembles de k chemins disjoints. Un ensemble de sommets U est un k -séparateur d'un graphe connexe et sépare x de y si U a au plus k sommets et si x et y sont dans deux composantes connexes du sous graphe obtenu en retirant U et tous les arcs incidents. Le théorème de Menger (1926) dit que si dans un graphe connexe x et y sont deux sommets non adjacents et s'il n'existe pas k chemins disjoints entre x et y , alors il existe un $(k - 1)$ -séparateur séparant x et y .

D'un point de vue logique ce théorème énonce un changement de quantification : la négation d'une propriété existentielle : « il n'existe pas d'ensemble de k chemins disjoints tels que ... » est montrée équivalente à une propriété de la forme existentielle : « il existe un k -séparateur ... ». Beaucoup de théorèmes importants ont ce type de structure logique. On peut rapprocher cela de l'uniforme continuité sur un intervalle compact, qui remplace une propriété de la forme « quelque-soit il-existe » par une propriété de la forme « il-existe quelque-soit ».

Le théorème de Menger peut se déduire du théorème de Ford et Fulkerson portant sur les flots dans les « réseaux de transport ». Soit un graphe orienté possédant un sommet « source » s de degré sortant non nul et un sommet « but » b de degré entrant non nul. A tout arc (x, y) est associé un entier $c(x, y)$ positif ou nul, sa *capacité*. Un *flot* est une fonction f qui associe à chaque arc (x, y) un entier $f(x, y)$, son *flux*, avec $0 \leq f(x, y) \leq c(x, y)$, et qui satisfait la loi de Kirchhoff, c'est à dire que pour chaque sommet sauf pour la source et le but, la somme des flux des arcs entrants est égale à celle des flux des arcs sortants.

Le flot partout nul satisfait trivialement ces conditions. Le problème est de trouver un flot qui maximise la valeur globale $f(G)$ définie comme la somme des $f(s, x)$ moins la somme des $f(x, s)$. De nombreux problèmes pratiques nécessitent ce type d'optimisation.

Une *coupe* est une partition de S en $A \cup B$ où $s \in A, b \in B$. On note $c(A, B)$ la somme des capacités des arcs de A vers B . Si f est un flot alors, par une sommation globale, on montre que $f(G)$ est égale à la somme des valeurs des arcs de A vers B moins la somme des valeurs des arcs de B vers A . Il en résulte que $f(G) \leq c(A, B)$ pour toute coupe A, B . Le théorème de Ford et Fulkerson (1957) indique que l'égalité peut être atteinte. Un algorithme d'améliorations successives permet de construire un flot qui réalise le maximum issu de cette majoration, c'est à dire le minimum des $c(A, B)$ pour toutes les coupes A, B .

Une variante du théorème de Menger est obtenue en définissant pour un graphe orienté dont on remplace chaque arête par deux arcs opposés :

$f(x, y) =$ le nombre maximum de chemins sans arêtes communes entre x et y ,

$c(x, y) =$ la cardinalité minimale d'un ensemble d'arêtes dont la suppression

déconnecte x et y .

On crée un arc de u à v avec capacité 1 pour tout couple de sommets adjacents u, v . Un flot entre deux sommets x et y correspond au nombre maximum de chemins sans arcs communs entre x et y . On obtient ainsi que le nombre maximum de chemins sans arête commune entre x et y est égal au cardinal minimum d'un ensemble d'arêtes dont la suppression déconnecte x et y .

6 Coloriages et problèmes NP-complets

Soit un ensemble fini C appelé l'ensemble des *couleurs*. Un *coloriage* d'un graphe G est une fonction $c : S(G) \rightarrow C$ qui affecte une couleur à chaque sommet, et qui est telle que deux sommets voisins n'ont pas la même couleur. Un graphe est *k-coloriable* s'il peut être colorié au moyen d'un ensemble de k couleurs. Il existe d'autres notions de coloriage des graphes (on peut demander que chaque arc ait une couleur différente de celle des arcs incidents, on peut aussi mettre d'autres contraintes que la différence de couleur de deux sommets voisins).

Dans tous les cas, on se pose des questions de ce type : est-ce que tous les graphes de tel ou tel type peuvent être coloriés avec un nombre fixé de couleurs? Si oui, comment peut-on déterminer un tel coloriage ?

Le *nombre chromatique* d'un graphe G est le nombre minimum $\chi(G)$ de couleurs d'un coloriage. Il n'est pas très difficile de montrer que $\chi(G) \leq \text{Degré}(G) + 1$. (On construit un arbre recouvrant en profondeur et on colore les sommets dans l'ordre associé en utilisant à chaque fois la plus petite couleur disponible.) L'égalité a lieu pour un graphe complet, ainsi que pour un cycle impair. Un théorème difficile de Brooks (1941) montre que ce sont les seuls cas où l'égalité est atteinte. Ainsi pour un graphe de degré d au moins 3 qui ne contient pas K_{d+1} comme sous-graphe, on a $\chi(G) \leq d$.

Tout graphe planaire peut être colorié avec 4 couleurs, quelque soit son degré. La preuve de ce résultat, conjecturé depuis plus d'un siècle, n'a été donnée qu'en 1976 et nécessite la vérification par ordinateur de nombreux cas. On parle à ce sujet de *coloriage des cartes* pour la raison suivante. Soit une carte de géographie dont les pays sont connexes (pas d'île ni d'enclave dans un pays voisin). Est-il possible de colorier chaque pays par une couleur parmi 4 de telle sorte que deux pays ayant une frontière commune (non réduite à un point) aient des couleurs différentes? Une traduction en termes combinatoires (ne reposant pas sur la notion de région connexe du plan réel), peut être donnée ainsi : on construit un graphe en prenant chaque capitale comme un sommet et on place un arc entre deux capitales si les pays ont une frontière commune.

On obtient un graphe planaire. Un 4-coloriage de ce graphe donnera donc le coloriage demandé de la carte géographique. Les coloriages de graphes ont bien d'autres applications. On citera des allocations de fréquences pour les téléphones mobiles ou les stations de radio : les fréquences de stations voisines doivent être différentes.

Le problème de déterminer si un graphe est 3-coloriable est un problème difficile, et ceci en un sens que l'on va préciser.

On dit qu'un problème est *polynomial* s'il peut être résolu par un algorithme formalisé en termes de *Machine de Turing* (modèle formel simplifié mais universel de calculateur), en temps polynomial par rapport à la taille de la donnée. La notion de « temps polynomial » est robuste. Pour les graphes, elle est indépendante de leur mode de représentation : les mêmes problèmes sont polynomiaux, que le graphe soit donné par une matrice booléenne ou par une liste de sommets et d'arcs. (Par contre, pour des problèmes comportant des nombres entiers comme données, il n'est pas indifférent que les entiers soient donnés en notation binaire ou en notation unaire. Les tailles des représentations ne sont plus comparées par des polynômes mais par des exponentielles.)

Un problème est **NP**, c'est à dire « résoluble en temps Polynomial par une machine Non déterministe » s'il est de la forme : il existe une affectation de valeurs à certaines variables qui satisfait une condition vérifiable en temps polynomial. La recherche de cette valuation des variables peut nécessiter une série d'essais systématiques dont on ne sait pas majorer par un polynôme le temps total de recherche. Par contre, pour chaque valuation particulière, la vérification qu'elle est « correcte » prend un temps polynomial.

On conjecture qu'il existe des problèmes de type **NP** qui n'ont pas d'algorithme polynomial. Cette conjecture a été classée comme l'un des 10 principaux problèmes de mathématiques à un récent Congrès Mondial des Mathématiciens. Parmi ces problèmes de type **NP**, certains, dits **NP-complets**, sont « les plus difficiles » de la classe **NP** toute entière. Cela veut dire, pour faire court, que $\mathbf{P} = \mathbf{NP}$, autrement dit que tout problème de type **NP** peut être résolu en temps polynomial (éventuellement avec des exposants énormes), si et seulement si tel problème **NP-complet**, par exemple celui de l'existence d'un 3-coloriage, possède un algorithme polynomial. Il existe d'autres problèmes **NP-complets** portant sur les graphes : l'existence d'un circuit Hamiltonien, le 3-coloriage des graphes planaires, le coloriage des arcs d'un graphe 3-régulier avec 3 couleurs, l'existence d'un sous-graphe complet de taille au moins k où k fait partie de la donnée du problème.

Déterminer si un graphe est 4-, 5-, ..., k -coloriable pour chaque k fixé est également **NP-complet**. Par contre, un graphe est 2-coloriable si et seulement si son ensemble de sommets S peut être partitionné en deux classes A et B telles qu'aucun arc ne relie deux sommets de A ou deux sommets de B . Déterminer si un graphe est 2-coloriable peut se faire au moyen d'une exploration à partir d'un arbre recouvrant en largeur, donc en temps polynomial.

Il existe des liens entre coloriage et orientations. Soit G un graphe non orienté. Si H est une orientation de G obtenue en choisissant une direction pour chaque arête, on note $L(H)$ le nombre maximum de sommets d'un chemin orienté dans H . Le théorème de Gallai et Roy énonce que $\chi(G)$ est le minimum des entiers $L(H)$ pour toutes les orientations H de G .

Esquisons sa preuve : Si G est colorié par $1, \dots, k$, orientons les arcs de sorte que $x \rightarrow y$ si $c(x) < c(y)$. Un chemin orienté a des couleurs strictement croissantes et donc un nombre de sommets au plus k . Inversement, soit H une orientation qui réalise le minimum de $L(H)$. On prend B un ensemble minimal d'arcs de H tel que $A - B$ est un sous-graphe sans circuit de H . Pour tout x on pose $c(x) =$ le nombre maximal de sommets d'un chemin orienté issu de x . On vérifie sans peine que c'est un k -coloriage.

Voici un autre exemple de relation entre coloriage et orientation : il s'agit de définir une orientation à partir d'un coloriage. Supposons que l'ensemble C des couleurs est l'ensemble des sommets d'un *tournoi* (c'est à dire d'un graphe orienté complet tel qu'aucun n'arc n'a d'arc « retour »). Soit G un graphe non orienté C -colorié. On définit une orientation de ce graphe en orientant un arc de $x \rightarrow y$ si et seulement si $c(x) \rightarrow c(y)$.

Question : Combien faut-il de couleurs pour spécifier ainsi l'orientation d'un graphe ?

Si ce graphe est un arbre, 3 couleurs suffisent, mais 4 sont nécessaires pour un circuit de longueur 4. Pour un graphe planaire, 80 couleurs suffisent, mais l'on n'a pas d'exemple où 80 couleurs sont nécessaires.

Pour un graphe orienté de telle sorte que chaque sommet ait un degré entrant au plus d , le nombre de couleurs est au plus $2^{2^{d+1}}$, alors que le nombre chromatique pour ces graphes est au plus $2d + 1$. (Ces graphes sont les graphes *uniformément d -creux*, c'est à dire tels que pour tout sous-graphe, le nombre d'arcs est au plus d fois le nombre de sommets.)

7 Planarité et configurations interdites

La notion de *graphe planaire*, c'est à dire plongeable topologiquement dans le plan, nécessite la notion d'arc de Jordan, soit un recours aux nombres réels et aux fonctions continues. La caractérisation combinatoire de Kuratowski évite cette difficulté.

On dit que H non orienté, est un *sous-graphe topologique* de G si G admet un sous-graphe K qui est obtenu en remplaçant les arêtes de H par des chemins disjoints (sauf à leurs extrémités), ou encore par subdivision itérée de ses arêtes.

Il est clair que si G est planaire, il en va de même de H et K . Si H est l'un des deux graphes K_5 et $K_{3,3}$, alors G n'est pas planaire. Le théorème de Kuratowski (1930) énonce que la réciproque est vraie : si un graphe ne contient aucun sous graphe topologique isomorphe à K_5 ou à $K_{3,3}$, alors il est planaire.

On dispose ainsi d'une caractérisation des graphes planaires en termes de leurs sous-graphes, ne faisant pas appel à l'analyse et à la géométrie.

On peut également représenter des graphes sur d'autres surfaces que le plan. Sur la sphère on représente les mêmes graphes que sur le plan, mais sur le tore on peut représenter K_5 et $K_{3,3}$. N. Robertson et P. Seymour ont démontré en 1986 que pour toute surface orientable (tores à plusieurs trous) la famille des graphes représentables était caractérisée par un nombre fini de sous-graphes topologiques « interdits ». Le cas des surfaces non-orientables (plan projectif, bouteille de Klein) est dû à Glover *et al.* (1980). Les preuves ne sont pas constructives. On ne connaît la liste de ces graphes que pour deux surfaces, le plan (la sphère) et le plan projectif pour lequel ils sont très nombreux : 103. Il y en a au moins 2200 le tore. On ne dispose même pas d'un algorithme qui, quitte à demander un temps extravagant, pourrait en principe les énumérer.

8 Arbres syntaxiques

La grammaire d'une langue définit des types de mots et de groupes de mots en fonction desquels on peut décomposer une phrase et lui donner une structure arborescente qui est liée à son sens. Cette structure permet de définir des règles dites « d'accord », et elle constitue pour un processus de traduction automatique une représentation « pivot ». Ainsi la traduction d'une langue dans une autre peut être vue comme composée de deux phases, une phase d'analyse syntaxique partant de la phrase donnée et aboutissant à la construction de son arbre syntaxique (avec les problèmes d'ambiguïté inhérents aux langues naturelles), et d'une phase de transformation de cet arbre en une phrase de la langue visée, transformation qui utilisera un lexique de mots et d'expressions idiomatiques, et une grammaire. L'approche grammaticale formalisée des langues naturelles a été introduite par N. Chomsky dans les années 50.

A cette époque ont été construits les premiers ordinateurs, dont le fonctionnement est assuré par des programmes écrits dans des « langages ». L'ordinateur dispose d'un langage interne, son « langage-machine », illisible pour un humain, dont les programmes sont des suites d'instructions définissant des opérations élémentaires, notamment arithmétiques, et des transferts de données d'un emplacement à un autre, sans oublier les instructions d'entrée-sortie. Le programmeur utilise un langage « évolué » pour écrire ses programmes (qui restent néanmoins peu compréhensibles pour le non-spécialiste). Le *compilateur* est un programme qui traduit les programmes écrits dans des langages de programmation en des programmes exécutables.

Comme les langues naturelles, les langages de programmation ont des *grammaires*, conçues, elles, pour éviter toute ambiguïté, et le compilateur a pour tâche la traduction. Le mécanisme en deux temps, *analyse syntaxique* construisant un arbre syntaxique à partir d'un texte, suivi d'une traduction proprement

dite de l'arbre en un programme-objet reste pertinent, même s'il est simplifié. Il en est résulté une importante recherche en *théorie des langages formels et en sémantique des langages de programmation*. Les arbres y jouent un rôle prépondérant.

Un *arbre syntaxique* représente un *terme* au sens de la logique mathématique et de l'algèbre universelle. Un polynôme, par exemple est un terme écrit avec des symboles d'opération, des constantes, des variables. La notation dite *infixée* qui consiste à placer le symbole d'opération entre les arguments est une source d'ambiguïté : comment doit on lire $x + y * z$? et $x - y - z$? Les notations préfixées lèvent ces ambiguïtés car ces deux termes s'écrivent alors $+(x, *(y, z))$ et $-(- (x, y), z)$ mais elles ne correspondent pas à nos habitudes d'écriture. Mais elles ont l'avantage de permettre l'utilisation de fonctions à plus de deux arguments. Un exemple de terme est $f(x, g(y, x, z), h(x), a)$, où f est d'arité 4, g d'arité 3, h d'arité 1.

Un tel terme peut être écrit comme ci-dessus avec des parenthèses et des virgules, mais également représenté par un *arbre étiqueté et ordonné*. Les étiquettes, valeurs des sommets, indiquent les fonctions, les variables et les constantes. Les arcs sont orientés vers les arguments. On obtient donc une arborescence. Mais en plus, l'ensemble des successeurs d'un sommet est totalement ordonné. En effet, l'ordre des arguments est important (sauf dans le cas très particulier de fonctions commutatives). On représente le plus souvent un tel terme comme une arborescence dont la racine en haut et les feuilles en bas (en fait les informaticiens disent « un arbre », car pour eux tout arbre est muni d'une racine).

Une *grammaire* peut être formalisée comme un ensemble de définitions simultanées d'un ensemble de termes. Prenons l'exemple très simple des termes écrits avec un unique symbole binaire f et deux constantes a et b . Quelques exemples de termes en commençant par les plus petits sont :

$$a, b, f(a, b), f(a, a), f(a, f(a, b)), f(f(b, b), a), \text{ etc. } \dots$$

On peut les caractériser par des *règles de formation*. Dans des ouvrages de mathématiques qui ne s'attardent pas à des questions formelles on trouve des définitions de ce genre :

- a et b sont des termes,
- si s et t sont des termes alors $f(s, t)$ est un terme,
- rien n'est un terme que par application des règles qui précèdent.

Pour être correct, il faut préciser qu'un terme est une suite finie de symboles, on dit un *mot* en théorie des langages formels. Cette hypothèse de finitude est nécessaire pour exclure le terme infini $f(f(.,.), f(.,.))$ solution de l'équation $T = f(T, T)$ (sorte de série formelle généralisée). En utilisant le vocabulaire de la théorie des langages formels, on écrirait que l'ensemble des termes est le langage engendré par la grammaire composée des règles de réécriture :

- $\langle \text{terme} \rangle \longrightarrow a,$
- $\langle \text{terme} \rangle \longrightarrow b,$
- $\langle \text{terme} \rangle \longrightarrow f(\langle \text{terme} \rangle, \langle \text{terme} \rangle).$

Une manière plus élégante de formuler cela est de dire que l'ensemble des termes est le plus petit ensemble L de mots qui contient a, b et inclut l'ensemble de mots $f(L, L)$. (On désigne par $f(L, L)$ l'ensemble des mots de la forme $f(s, t)$ pour s et t dans L .)

L'analyse syntaxique des langages de programmation s'appuie sur la notion de grammaire où, au lieu d'une unique catégorie d'expressions on en utilise plusieurs, par exemple <instruction>, <déclaration>, <expression booléenne>, <expression arithmétique>, <appel de procédure>. De façon semblable, les phrases des langues naturelles sont analysées en <groupe verbal>, <adjectif>, <proposition relative> etc. . . La notion de grammaire peut également s'appliquer à la description d'autres objets que des mots et en particulier à des graphes comme nous le verrons bientôt.

9 Structuration des graphes et grammaires

Indiquons quelques résultats de décomposition canonique.

Tout graphe est l'union disjointe de ses composantes connexes.

Tout graphe connexe est d'une manière unique, un recollement arborescent de ses composantes 2 connexes (un graphe est 2-connexe si tout couple de sommets appartient à un cycle).

Un graphe 2-connexe peut également être décrit de manière unique comme recollement arborescent de ses composantes 3-connexes (deux sommets sont reliés par 3 chemins disjoints).

Les composantes fortement connexes d'un graphe 2-connexe orienté forment un graphe orienté sans circuit que l'on obtient en contractant chaque composante en un sommet.

Une autre notion importante est la *décomposition modulaire*. Elle s'appuie sur la notion de *substitution* dans un graphe d'un graphe à un sommet. Ainsi, $G[H/v]$ où G et H sont disjoints (s'ils ne le sont pas on remplace H par une copie isomorphe) est le graphe K dont l'ensemble de sommets est $S(G) \cup S(H) - \{v\}$, et dont les arcs sont ceux de H , ceux de G non incidents à v , et pour tout x dans $G, x \neq v$, pour tout y dans H , on place un arc de $x \longrightarrow y$ (de $y \longrightarrow x$) si et seulement si il y a dans G un arc de $x \longrightarrow v$ (de $v \longrightarrow x$).

Un graphe est *premier* s'il ne peut pas s'écrire comme $G[H/v]$ sauf avec G ou H réduit à un unique sommet. Cette opération de substitution se généralise à des substitutions simultanées à plusieurs sommets : $G[H_1/v_1, \dots, H_n/v_n]$. Les substitutions peuvent être imbriquées, c'est à dire que l'on peut définir un graphe comme $G[H_1/v_1, \dots, K[L_1/x_1, \dots, M_1/y_1, \dots], \dots, H_n/v_n]$.

Tout graphe (fini) peut s'écrire ainsi et de façon unique comme le résultat de substitutions itérées portant sur des graphes premiers. On considère comme premiers les graphes formés de sommets sans arcs, les graphes complets, et les graphes des ordres totaux.

Ces décompositions sont utiles pour ramener des preuves relatives à tous les graphes d'un certain type aux cas particuliers de leurs constituants, qui sont, selon les cas, les graphes connexes, les graphes fortement connexes et sans circuits, ou les graphes premiers. D'autre part, pour améliorer l'efficacité et la modularité d'un algorithme, il est souvent utile de réduire le problème considéré à des problèmes identiques ou similaires pour des composants tels que les graphes premiers. Malheureusement, ces décompositions restent superficielles. A titre d'exemple, un graphe aléatoire à n sommets a une probabilité d'être premier qui tend vers 1 quand n tend vers l'infini.

On citera plus loin d'autres décompositions, d'une certaine façon plus puissantes, qui n'ont plus la propriété d'unicité, mais qui néanmoins diminuent le temps de calcul de certains algorithmes.

Un graphe « général » est un objet non structuré, défini de manière globale et non par composition d'entités plus petites. Cela ne facilite pas la rédaction des preuves. Les choses sont plus simples pour les arbres, car on peut dans ce cas faire des récurrences sur leur structure. Il existe toutefois des familles de graphes que l'on peut engendrer au moyen d'un nombre fini d'opérations, et que l'on peut donc représenter par des termes.

Je prendrai l'exemple des cographes. Ces graphes sont définissables à partir de deux opérations l'*union disjointe* et le *produit complet*, notées respectivement \oplus et \otimes .

On ne considère ici que des graphes non orientés, sans boucles.

Le graphe $G = H \oplus K$ est défini à partir de deux copies disjointes des graphes H et K : on en prend l'union des sommets et l'union des ensembles d'arcs. Le graphe $G = H \otimes K$ est défini comme $H \oplus K$ augmenté de tous les arcs possibles entre les sommets de H et ceux de K . Ces graphes ne sont définis qu'à isomorphisme près puisque, à chaque étape, on prend des copies disjointes. Notons $\mathbf{1}$ le graphe réduit à un sommet isolé. Les *cographe*s sont les graphes définis par les termes finis écrits avec $\mathbf{1}$, \oplus et \otimes .

On peut également écrire que l'ensemble des cographe L est l'unique solution dans les ensembles de graphes finis de l'équation de l'équation :

$$L = L \oplus L \cup L \otimes L \cup \{\mathbf{1}\}.$$

Chacun des graphes engendrés a une représentation arborescente.

Est-elle unique ? Non car les opérations \oplus et \otimes sont associatives et commutatives (mais c'est tout). Donc plusieurs termes représentent le même cographe (à isomorphisme près). Cela dit, on peut définir une représentation unique par des termes à opérateurs commutatifs d'arité variables.

10 Décompositions arborescentes et algorithmes polynomiaux

Voici un exemple d'utilisation algorithmique des décompositions de graphes.

Soit D un ensemble fini d'étiquettes. Un D -graphe est un graphe muni d'une injection $s : D \longrightarrow S(G)$. On appelle $s(a)$ la a -source de G .

On définit sur ces graphes une opération de *recollement*. Soit H un D -graphe et K un E -graphe. On note $G = H//K$ le graphe obtenu à partir de l'union disjointe de H et de K en fusionnant la a -source de H et la a -source de K pour tout a dans $D \cap E$. Le graphe G est donc un $D \cup E$ -graphe. Si h est une injection: $D \longrightarrow E$, on note $Ren(h)$ le *renommage de sources* défini de la façon suivante: Si H est un E -graphe, $Ren(h)(H)$ est le D -graphe obtenu en prenant $s \circ h$ comme application désignant les sources.

En utilisant des sous-ensembles d'un ensemble fixé de k étiquettes de sources et des graphes de base à au plus k sommets, on définit une famille de graphes que l'on peut désigner par des termes. Le nombre minimal k d'étiquettes à utiliser pour construire un graphe de cette manière est un paramètre introduit indépendamment pour d'autres motivations par Robertson et Seymour et dénommé sa *largeur arborescente* (*tree-width*). Les graphes de largeur arborescente au plus k sont d'une certaine façon très particuliers. Ils sont uniformément $(k + 1)$ -creux. (Mais les graphes planaires bien qu'uniformément 3-creux, ne sont pas de largeur arborescente bornée).

Montrons l'intérêt algorithmique de cette notion, en prenant l'exemple du problème NP-complet de la recherche d'une 3-coloration. L'idée est de chercher à vérifier la 3-colorabilité d'un graphe G en partant d'une décomposition arborescente, c'est-à-dire de ramener la vérification que $G = H//K$ ou que $G = Ren(h)(H)$ est 3 coloriable à des vérifications de ce type pour H et K . Il est clair que si G est 3 coloriable, ses sous-graphes H et K le sont aussi. Mais la réciproque est fautive. Ainsi, comme souvent pour une preuve par récurrence, il faut renforcer la propriété et faire la récurrence sur une propriété plus forte.

Pour un D -graphe, on note $c : V(G) \longrightarrow \{1, 2, 3\}$ un 3-coloriage. On note $C(G)$ l'ensemble de tous les 3-coloriages de G . On va extraire de $C(G)$ une information finie qui « passe à la récurrence ».

A un coloriage c on associe sa restriction aux sources, c'est à dire l'application $c \circ s : D \longrightarrow \{1, 2, 3\}$ qui associe à chaque étiquette de source la couleur du sommet correspondant. On note $c^\#(G)$ cette application pour un D -graphe G .

L'observation clef est que l'on peut déterminer si $d = c^\#(H//K)$ en fonction de $c^\#(H)$ et $c^\#(K)$.

Une fois connue une expression de G au moyen de recollements et de renommages de sources, on peut calculer l'ensemble $c^\#(H)$ pour tous les sous-graphes de G associés aux sous-termes du terme qui le définit, inductivement et en montant dans l'arbre syntaxique du terme. On obtient donc un algorithme en temps linéaire (supposant le terme connu), et pour un nombre d'étiquettes maximal k fixé. Les informations à calculer inductivement doivent rester dans un ensemble fini, même très grand.

Cette méthode s'applique à bien d'autres problèmes. En particulier à la recherche d'un cycle Hamiltonien optimal, et plus généralement à tout problème de graphe que l'on peut formuler par une formule de la *logique du second-ordre monadique*, c'est à dire d'une formule logique écrite avec des quantifications existentielles et universelles sur des variables désignant des sommets, des arcs,

des ensembles de sommets et des ensembles d'arcs. Cette logique exclut des conditions du genre : « il existe une bijection d'un ensemble X sur un ensemble Y ».

11 Graphes dénombrables : utilisation du Lemme de Koenig.

Jusqu'ici nous avons surtout parlé de graphes finis, mais la plupart des définitions s'étendent sans difficulté aux graphes infinis. Nous nous limiterons à des graphes dénombrables.

Le *Lemme de Koenig* établit que dans une arborescence infinie dont chaque sommet est de degré fini (mais non nécessairement borné), il existe un chemin infini issu de la racine. (*Preuve* : On construit pas à pas ce chemin en choisissant à chaque fois de continuer en allant vers un sommet qui est la racine d'une sous-arborescence infinie). On peut prouver ainsi que tout sous-ensemble infini de l'intervalle réel $[0,1]$ possède un point d'accumulation.

Ce lemme permet d'étendre aux graphes dénombrables diverses propriétés des graphes finis. Si un graphe dénombrable G est tel que tout sous-graphe fini est k -coloriable, alors il est lui même k -coloriable. La preuve se fait ainsi : on choisit une suite G_i , $i \in \mathbf{N}$, de graphes finis, croissante pour l'inclusion, dont la réunion est G . On construit une arborescence infinie dont les sommets sont les coloriage c des graphes G_i . On définit un arc $c \rightarrow c'$ si et seulement si c' est un coloriage de G_i et c est la restriction de c' à G_{i-1} . On obtient une arborescence infinie (car chaque G_i possède au moins un k -coloriage). Il n'y a qu'un nombre fini de k -coloriages d'un graphe fini, donc chaque sommet de l'arborescence est de degré fini. Le lemme de Koenig montre qu'il existe un chemin infini. Ces coloriage sont tous compatibles et l'on a ainsi un coloriage de l'union des graphes G_i , c'est-à-dire de G .

On peut donc montrer ainsi que la validité du Théorème des 4 Couleurs pour les graphes finis implique son extension aux graphes dénombrables. Il en va de même de la définition des orientations de degré entrant au plus d au moyen d'un graphe de couleurs. Par contre, certains résultats nécessitent une preuve particulière. On a vu que tout graphe fini avait un arbre de Trémaux. On montre l'existence d'un tel arbre par une exploration du graphe « en profondeur ». Dans le cas d'un arbre infini, cette exploration peut omettre complètement une partie du graphe, et donc ne permet pas de conclure à l'existence d'un arbre de Trémaux. Néanmoins, il en existe. Je laisserai cette démonstration aux lecteurs.

Pour contacter l'auteur :
<http://www.labri.fr/~courcell/SafeMail.php>