RAIRO. RECHERCHE OPÉRATIONNELLE

B. PEROCHE

Complexité de l'arboricité linéaire d'un graphe

RAIRO. Recherche opérationnelle, tome 16, n° 2 (1982), p. 125-129

<http://www.numdam.org/item?id=RO_1982__16_2_125_0>

© AFCET, 1982, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (http://www.numdam.org/conditions). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
http://www.numdam.org/

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 16, n° 2 mai 1982, p. 125 à 129)

COMPLEXITÉ DE L'ARBORICITÉ LINÉAIRE D'UN GRAPHE (*)

par B. PEROCHE [1]

Résumé. — *$la(G)$ étant le nombre minimal de forêts de chaînes partitionnant les arêtes d'un graphe G, on montre que déterminer $la(G)$ est un problème N.P. complet, y compris dans le cas où G est de degré maximal 4.*

Mots clés : graphe, arboricité linéaire, problème N.P. complet.

Abstract. — *We prove that the determination of the linear arboricity of a graph is a N.P.-complete problem, even when the graph has maximum degree four.*

Keywords: graph, linear arboricity, N.P.-complexity.

1. INTRODUCTION

Soit $G=(X, E)$ un graphe. On appellera arboricité linéaire de G, et on notera $la(G)$, le nombre minimal de forêts de chaînes (i.e. de sous-graphes partiels de G dont les composantes connexes sont des chaînes) qui partitionnent les arêtes de G. Dans [3], le résultat suivant concernant $la(G)$ a été obtenu : si G a pour degré maximal 4, $2 \leqq la(G) \leqq 3$.

Un tel résultat conduit à se demander si on ne pourrait pas trouver un « bon » algorithme permettant de fournir la réponse au problème suivant, noté LA : G étant de degré maximal 4, a-t-on $la(G)=2$? On va montrer, dans ce qui suit, que LA est $N.P.$ complet et donc qu'on a toutes les raisons de penser qu'il n'est pas possible de trouver un algorithme polynomial pour le résoudre.

Dans ce qui suit, la terminologie utilisée concernant la complexité sera celle de [1].

Pour montrer que LA est $N.P.$ complet; on va utiliser le problème de *la* 3-satisfiabilité : soient $x_1, x_2, \ldots, x_t$ des variables booléennes et soit $C=C_1 . C_2 . \ldots . C_p$ une fonction booléenne sous forme canonique conjonctive

(*) Reçu septembre 1981.
(1) I.U.T. Villetaneuse, Université Paris-Nord, rue J. B.-Clément, 93430 Villetaneuse, France.

© AFCET-Bordas-Dunod.

des variables x_i, telle que $\forall j$, $C_j = \tilde{x}_{j_1} + \tilde{x}_{j_2} + \tilde{x}_{j_3}$. Le problème : « l'équation $C=1$ admet-elle une solution ? » sera noté 3-SAT. On sait qu'il est $N.P.$ complet [1].

Il est facile de voir que LA est dans la classe $N.P.$ Pour obtenir le résultat annoncé, on va exhiber une réduction polynomiale de 3-SAT à LA.

Pour terminer, signalons que certaines des idées utilisées ici sont inspirées de [2].

2. DESCRIPTION DES GRAPHES UTILISÉS

Étant donnée une fonction booléenne C du problème 3-SAT, on va lui associer un graphe $G=(X, E)$ de degré maximal 4 qui aura la propriété suivante : on pourra partitionner E en deux forêts de chaînes si et seulement si C est satisfiable, c'est-à-dire si l'équation $C=1$ admet au moins une solution. Pour construire G, on va utiliser un certain nombre de graphes que l'on va définir ci-dessous, et dont on donnera les propriétés qui nous seront utiles.

2.1. Le graphe G_1 (voir *fig.* 1)

Ce graphe est de degré maximal 4 et deux paires d'arêtes en sortent : $\{a, b\}$ et $\{c, d\}$. Il est facile de voir que $la(G_1)=2$, et on peut vérifier :

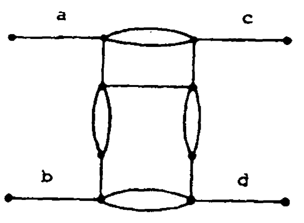

Fig. 1.

LEMME 1 : *Il n'existe qu'une seule partition des arêtes de G_1 par deux forêts de chaînes f_1 et f_2 si on impose à a et à b d'appartenir à la même forêt f_1 (resp. si on impose à a d'appartenir à f_1 et à b d'appartenir à f_2). De plus, dans cette partition, $c \in f_1$ et $d \in f_2$ (resp. c et d appartiennent à f_1).*

Pour ce graphe G_1, chaque paire $\{a, b\}$ et $\{c, d\}$ sera considérée comme une « entrée-sortie ». De plus, et dans toute la suite, on adoptera la convention suivante : si les deux arêtes d'une entrée-sortie appartiennent à une même forêt de chaînes, on dira que l'entrée-sortie représente la valeur « fausse »; sinon, c'est-à-dire si les deux arêtes d'une entrée-sortie n'appartiennent pas à la même forêt de chaînes, elle représentera la valeur « vraie ». On peut donc dire que G_1 est un graphe qui permet d'échanger « vrai » et « faux ».

2.2. Le graphe G_2. A partir de G_1, on construit un graphe G_2 comme indiqué figure 2

G_2 est de degré maximal 4, a trois entrées-sorties $\{a, b\}$, $\{c, d\}$ et $\{e, f\}$, et vérifie $la(G_2)=2$. De plus, on a :

LEMME 2 : *Il n'existe qu'une seule partition des arêtes de G_2 en deux forêts de chaînes f_1 et f_2, si on impose à une des entrées-sorties d'être « fausse » (resp. « vraie »). De plus, dans cette partition, les trois entrées-sorties sont alors toutes « fausses » (resp. toutes « vraies »).*

2.3. Le graphe G_3 **(voir** *fig.* 3)

G_3 est de degré maximal 4 et vérifie $la(G_3)=2$. Il a trois entrées-sorties : $\{a, b\}$, $\{c, d\}$ et $\{e, f\}$ et vérifie la propriété suivante :

LEMME 3 : *Dans toute partition des arêtes de G_3 en deux forêts de chaînes, on a soit une, soit deux, soit trois entrées-sorties « vraies ».*

3. RÉSULTAT PRINCIPAL

On peut maintenant démontrer le résultat annoncé :

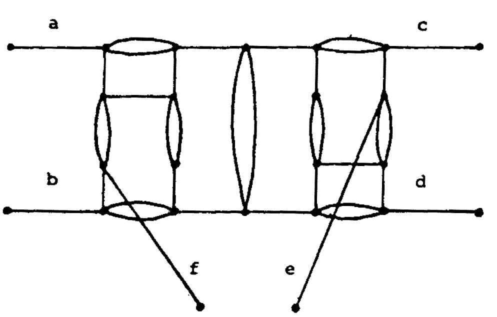

Fig. 2.

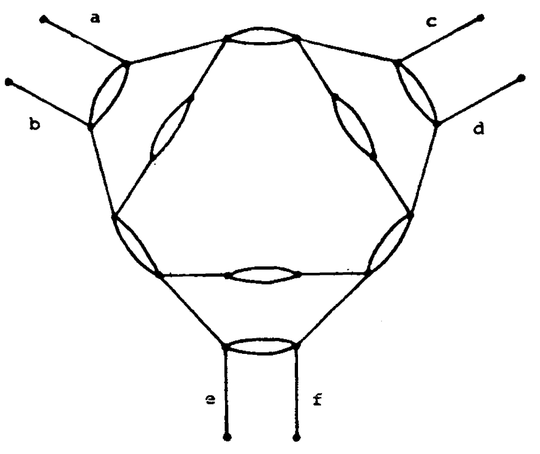

Fig. 3.

THÉORÈME 4 : *Déterminer si* $la(G)=2$ *pour un graphe* G *de degré maximal* 4 *est un problème N.P. complet.*

Preuve : Il est facile de voir que LA est dans la classe $N.P.$ On va obtenir maintenant une réduction polynomiale de 3-SAT à LA.

Soit C une fonction booléenne des variables x_i, $1 \leqq i \leqq t$, $C=C_1 . C_2 . \ldots C_p$ associée à un problème 3-SAT.

1[re] *étape :* A chaque variable x_i, on associe un graphe $G(i)$ de la manière suivante : on définit $q(i)$ comme étant le nombre d'apparitions de x_i ou de $\bar{x}_i$ parmi les clauses $C_1, C_2, \ldots, C_p$:

(1) si $q(i)=1$, on prend $G(i)=G_1-\{c, d\}$;

(2) si $q(i) \geqq 2$, on construit $G(i)$ à partir de $q(i)$ graphes G_2 que l'on raccorde de la manière suivante : si on note s_i, s'_i et σ_i les trois entrées-sorties du i-ième graphe G_2^i, on raccorde s'_i et σ_{i+1} pour $1 \leqq i \leqq q$ (les indices étant pris modulo q).

2[e] *étape :* A chaque clause C_j, on associe un graphe H_j du type G_3.

Ensuite, pour tout k, $1 \leqq k \leqq 3$, si la variable $\tilde{x}_{j_k}$ de C_j est x_i, on identifie une sortie quelconque de H_j non encore utilisée avec une sortie quelconque non encore utilisée de $G(i)$; si la variable $\tilde{x}_{jk}$ de C_j est $\bar{x}_i$ on place un graphe G_1 entre une sortie quelconque non encore utilisée de H_j et une sortie quelconque non encore utilisée de $G(i)$.

On fait cela pour toutes les variables et toutes les clauses C_j, et on obtient ainsi un graphe G de degré maximal 4, qui peut être construit à partir de C en utilisant un algorithme polynomial.

Du lemme 2, on peut déduire que les arêtes de $G(i)$ sont partitionnées en deux forêts de chaînes si et seulement si toutes les entrées-sorties sont soit « vraies », soit « fausses ».

Supposons que $C=1$ ait une solution. On peut partitionner les arêtes de chaque $G(i)$, $1 \leqq i \leqq p$, par deux forêts de chaînes de telle sorte que les sorties soient « vraies » si $x_i=1$ et « fausses » sinon. Comme $C=1$ est satisfiable, les entrées de chaque H_j ne sont pas toutes « fausses » donc, d'après le lemme 3, on peut partitionner chaque H_j par deux forêts de chaînes. Donc, si C est satisfiable, on peut partitionner les arêtes de G avec deux forêts de chaînes.

Réciproquement, supposons que $C=1$ n'ait pas de solution. Pour une valeur des variables, il y a donc au moins une clause C_j qui vaut 0. Donc les entrées du graphe correspondant H_j seront toutes « fausses », donc il faudra trois forêts de chaînes pour partitionner les arêtes de H_j, d'après le lemme 3, donc pour partitionner les arêtes de G.

On a donc prouvé que $la(G)=2$ si et seulement si C est satisfiable, ce qui prouve que LA est $N.P.$ complet.

BIBLIOGRAPHIE

1. M. Chein, *Cours de D.E.A. : Graphes et Informatique*, 1974-1975, Institut de Programmation, Paris-VI.
2. J. Holyer, *The N.P. Completness of Edge-Colouring*, S.I.A.M. J. of Computing, vol. 10, 1981, p. 718-720.
3. B. Peroche, *On Partitions of Graphs into Linear Forests and Dissections*, Rapport de Recherche, n° 2 du G.R. 22 du C.N.R.S., Paris-VI, 1980.