RAIRO. RECHERCHE OPÉRATIONNELLE

B. PEROCHE
B. SADI

Recouvrement et partition en chaînes des arêtes d'un graphe cubique

RAIRO. Recherche opérationnelle, tome 20, nº 2 (1986), p. 163-170
<http://www.numdam.org/item?id=RO_1986__20_2_163_0>

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 20, n° 2, mai 1986, p. 163 à 170)

RECOUVREMENT ET PARTITION EN CHAÎNES DES ARÊTES D'UN GRAPHE CUBIQUE (*)

par B. PEROCHE (1), et B. SADI (2)

Résumé. — *Étant donné un graphe cubique, nous montrons que la détermination de son indice de recouvrement en chaînes est un problème NP-complet et nous exhibons un algorithme linéaire pour obtenir son indice de partition en chaînes.*

Mots clés : Complexité, algorithme, graphe, recouvrement.

Abstract. — *Given a cubic graph, we prove that the determination of its unrestricted path number is a NP-complete problem but we exhibit a linear algorithm for the determination of its path number.*

Keywords: Complexity, algorithm, graph, cover.

1. INTRODUCTION

Un graphe G est constitué d'un ensemble fini non vide de sommets, $X(G)$, et d'une famille finie $E(G)$ de paires de sommets, les arêtes.

La terminologie suivie sera celle de Berge [1].

Soit P une famille de chaînes élémentaires de G. Si chaque arête de G appartient à au moins un élément de P, alors P est un recouvrement en chaînes des arêtes de G; si chaque arête de G appartient à exactement un élément de P, alors P est une partition en chaînes des arêtes de G. On notera $\rho(G)$ [resp. $\pi(G)$] le cardinal minimal d'un recouvrement (resp. partition) en chaînes de $E(G)$. Ces invariants ont été introduits par Harary dans [4] et étudiés par de nombreux auteurs, en particulier [2], [5], [6], [7] et [9].

Dans cet article, nous allons nous intéresser à la complexité de problèmes de détermination des invariants définis ci-dessus. Pour les définitions de complexité utilisées ici, nous renvoyons le lecteur à [3].

(*) Reçu mars 1985.

(1) Département Informatique, École des Mines de Saint-Étienne, 158, cours Fauriel, 42023 Saint-Étienne Cedex 2, France.

(2) Laboratoire d'Économétrie, Université P. et M. Curie, 4, place Jussieu, 75230 Paris Cedex, France.

R.A.I.R.O. Recherche opérationnelle/Operations Research, 0399-0559/86/01 163 8/$ 2.80

Dans [8], les problèmes de détermination de $\pi(G)$ et de $\rho(G)$ ont été classés. Plus précisément, si on note 2-PAR le problème:

donnée: un graphe G;

question: a-t-on $\pi(G)=2$?

et 2-REC le problème:

donnée: un graphe G;

question: a-t-on $\rho(G)=2$?

on a:

THÉORÈME A [8]: 2-PAR *et* 2-REC *sont NP-complets.*

Dans cet article, nous allons nous restreindre à la classe des graphes cubiques. Nous allons montrer que la détermination de $\rho(G)$ reste un problème NP-complet; par contre, nous exhiberons un algorithme linéaire fournissant une partition en chaînes de cardinal minimal des arêtes de G.

A notre connaissance, c'est la première fois que des problèmes correspondants de partition et de recouvrement des arêtes ou des sommets d'un graphe ne figurent pas dans la même classe de complexité. Nous pensons qu'il s'agit là d'un fait qui mérite d'être souligné.

2. DÉTERMINATION DE L'INDICE DE RECOUVREMENT

Considérons le problème suivant, que nous noterons 2-REC/CUBIQUE:

donnée: un graphe cubique G;

question: a-t-on $\rho(G)=2$?

Il est clair que 2-REC/CUBIQUE appartient à NP. Nous allons montrer que ce problème est NP-complet en exhibant une transformation polynomiale de 2-DEC vers 2-REC/CUBIQUE, où 2-DEC est défini de la manière suivante:

donnée: un graphe orienté $G=(X, U)$ tel que $\forall x \in X$, $d^+(x)=d^-(x)=2$;

question: peut-on partitionner U en deux circuits hamiltoniens?

Rappellons que 2-DEC a été étudié dans [8] et qu'il a été classé NP-complet.

Pour définir la transformation, nous allons utiliser le graphe cubique G_0 défini par la figure 1.

Ce graphe vérifie les propriétés suivantes:

LEMME 1:

(i) $\rho(G_0)=2$.

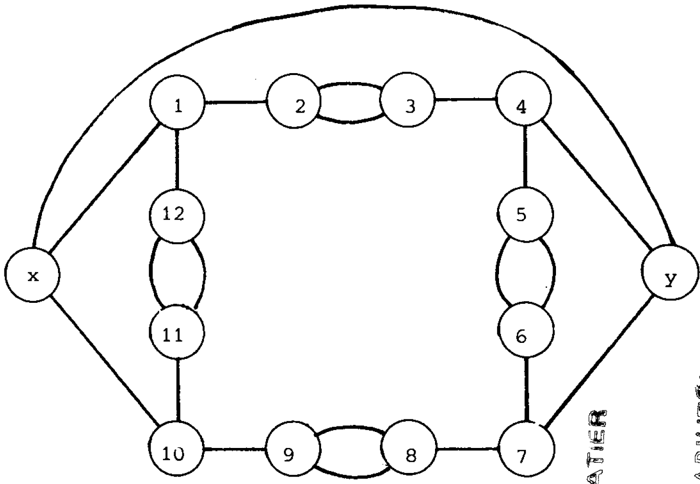

Figure 1

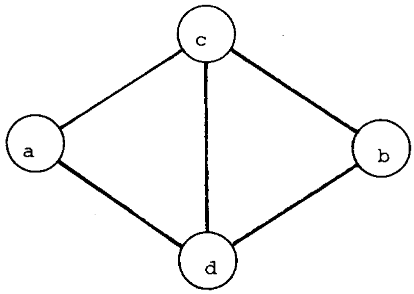

Figure 2

(ii) *Pour tout recouvrement en chaînes* $P=\{P_1, P_2\}$ *des arêtes de* G_0, xy *appartient à* P_1 *et à* P_2.

Preuve: (i) est évident.

Pour prouver (ii), il suffit de montrer que toute chaîne hamiltonienne de G_0 contient xy. Supposons donc qu'il existe une chaîne hamiltonienne μ ne contenant pas xy. Si μ n'a pas d'extrémité en x, μ a nécessairement une extrémité dans $\{11, 12\}$, par exemple 12; l'autre extrémité ne peut donc être que 9 et μ ne peut alors passer à la fois par 5 et par y. Ce cas est donc impossible et μ joint donc x à y. Supposons que μ contienne l'arête $\{x, 1\}$; si $\{1, 2\}$ (resp. $\{1, 12\}$) appartient à μ, 12 (resp. 2) doit être extrémité de μ, ce qui contredit le fait que μ joint x à y. Toute chaîne hamiltonienne de G_0 contient donc xy. □

Remarque: il est possible d'obtenir un graphe ayant la propriété du graphe G_0 et sans arêtes parallèles; pour ce faire, il suffit de remplacer toute paire d'arêtes parallèles entre deux sommets par le graphe H de la figure 2.

On peut maintenant définir la transformation T.

Soit $G=(X, U)$ un graphe orienté tel que $\forall x \in X,\ d^+(x)=d^-(x)=2$. On supposera que $X=\{1, 2, \ldots, n\}$.

(i) On remplace le sommet 1 par un graphe isomorphe à $G_0-xy+xx'+yy'$, où x' et y' sont de nouveaux sommets.

(ii) Pour tout $i \in 2, \ldots, n\}$, on remplace le sommet i par un graphe G_i isomorphe à $H+aa'+bb'$, où a' et b' sont de nouveaux sommets (on supposera que $X(G_i)=\{a'_i, a_i, b_i, b'_i, c_i, d_i\}$).

(iii) Pour tout i et $j \in \{1, 2, \ldots, n\}$, on remplace chaque arc $[i, j]$ par:

– une arête $b'_i a'_j$, si $i \neq 1$ et $j \neq 1$;

– une arête $y' a'_j$, si $i=1$;

– une arête $b'_i x'$, si $j=1$.

T, qui peut être schématisée par la figure 3, est bien polynômiale et associe à G un graphe non orienté cubique G'. On a:

THÉORÈME 2: *G peut être décomposé en deux circuits hamiltoniens si et seulement si* $\rho(G')=2$.

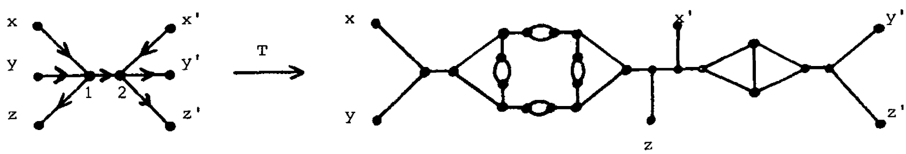

Figure 3

Preuve: Supposons que G soit décomposé par les deux circuits hamiltoniens:

$c_1=[i_1=1, i_2, \ldots, i_n, i_1]$

et

$c_2=[j_1=1, j_2, \ldots, j_n, j_1]$.

Si on note:

$\rightarrow i=[a'_i, a_i, c_i, d_i, b_i, b'_i]$,

$\leftarrow i=[a'_i, a_i, d_i, c_i, b_i, b'_i]$, les deux chaînes recouvrant les arêtes de G_i, on peut définir deux chaînes dans G' par :

$$\mu_1=[9, 8, 7, 6, 5, 4, y, y', \rightarrow i_2, \rightarrow i_3, \ldots, \rightarrow i_n, x', x, 10, 11, 12, 1, 2, 3],$$

$$\mu_2=[12, 11, 10, 9, 8, 7, y, y', \leftarrow j_2, \leftarrow j_3, \ldots, \leftarrow j_n, x', x, 1, 2, 3, 4, 5, 6].$$

Il est facile de vérifier que $\{\mu_1, \mu_2\}$ est un recouvrement en chaînes des arêtes de G'.

Inversement, supposons que $\rho(G')=2$ et soit $P=\{\mu_1, \mu_2\}$ un recouvrement en chaînes minimal des arêtes de G'. Comme G' est cubique, μ_1 et μ_2 ont quatre extrémités distinctes et, d'après le Lemme 1, ces quatre extrémités appartiennent au graphe isomorphe à G_0. Pour $i=1, 2$, chaque chaîne μ_i arrivant en un sommet a'_j traverse donc tous les sommets du graphe G_j et ressort par b'_j. En remplaçant chaque sous-chaîne $[a'_j, \ldots, b'_j]$ par le sommet j, on obtient deux circuits hamiltoniens de G qui sont disjoints, aucune arête $a'_j b'_k$ de G' ne pouvant être couverte deux fois par les chaînes de P. □

Il est facile de voir que le graphe cubique G' obtenu par la transformation T est 2-connexe. Définissons alors le problème 2-REC/3-CONNEXE par :

donnée : un graphe G cubique, 3-connexe;

question : a-t-on $\rho(G)=2$?

Nous proposons la conjecture suivante :

CONJECTURE 1 : 2-REC/3-CONNEXE est NP-complet.

3. DÉTERMINATION DE L'INDICE DE PARTITION

Soit G un graphe cubique connexe simple (c'est-à-dire sans arêtes parallèles); nous noterons $n=|X(G)|$ et $m=|E(G)|$. Nous allons proposer ci-dessous un algorithme linéaire exhibant une partition en chaînes de cardinal minimal de $E(G)$.

Nous fournirons deux versions de cet algorithme. La première présente les idées utilisées et permet de prouver facilement l'algorithme; la seconde, plus détaillée, permet de mesurer la complexité de l'algorithme.

3.1. Version simplifiée

```
début
Tant qu'il existe une arête e non marquée, faire
  début
    prendre une chaîne élémentaire maximale μ contenant e
    marquer toutes les arêtes de μ
  fintantque
fin
```

Pour prouver cet algorithme, nous allons démontrer la propriété suivante:

PROPOSITION 3: *L'algorithme fournit une partition en chaînes élémentaires où tout sommet est exactement extrémité d'une chaîne.*

Preuve: Raisonnons par l'absurde; supposons la propriété fausse et soit x le premier sommet rencontré au cours de l'exécution de l'algorithme en lequel deux chaînes μ_1 et μ_2 aient une extrémité commune. On supposera que x a pour voisins x_1, x_2 et x_3 (comme G est simple, $x_i \neq x_j$ si $i \neq j$). On supposera que $\mu_1 = [x, x_1, \ldots]$ a été obtenue avant $\mu_2 = [x, x_2, \ldots]$ lors de l'exécution de l'algorithme.

Comme μ_1 est maximale par construction, μ_1 passe par x_3; le même raisonnement montre que μ_2 passe par x_3. Comme l'arête xx_3 ne peut appartenir ni à μ_1, ni à μ_2, on a $\mu_1 = [x, x_1, \ldots, x_3]$ et $\mu_2 = [x, x_2, \ldots, x_3]$. Considérons alors le sommet y voisin de x_3 sur μ_2. Comme μ_1 est maximale, μ_1 passe par y; on obtient donc une contradiction car y, de degré 3, devrait être sommet intérieur de deux chaînes distinctes, ce qui est impossible. □

La Proposition 3 permet de valider l'algorithme car elle met en évidence l'invariant de l'algorithme: à chaque fois que l'on visite un arc supplémentaire, la propriété est vérifiée.

Par ailleurs, $\pi(G) \geqq i(G)/2$ (*voir* [7]), où $i(G)$ est le nombre de sommets impairs de G. D'après la Proposition 3, on obtient $i(G)/2$ chaînes élémentaires partitionnant $E(G)$, ce qui prouve que la partition est optimale.

3.2. 2^e^ version

Nous allons présenter maintenant une version plus précise de l'algorithme; cette version sera exprimé en un pseudo-langage proche de PASCAL.

Nous utiliserons les structures de données suivantes:

deux tableaux à une dimension, MARQ et NUM, indicés par les sommets;

un tableau à une dimension, UTILISE, indicé par les arêtes;

une pile PIL, qui pourra contenir les sommets.

3.3. Algorithme PARTIT (G)

```
début
1. pour j de 1 à n, faire
     début
     MARQ [j] ← 0
     NUM [j] ← 0
     finpour
2. pour toute arête xy, faire
     UTILISE [xy] ← 0
3. i ← 1
     choisir une arête e = xy dans G
```

```
     μ_i ← { xy }
     UTILISE [xy] ← 1
     empiler (y, PIL)
     MARQ [x] ← MARQ [x]+1
     MARQ [y] ← MARQ [y]+1
     NUM [x] ← i
     NUM [y] ← i
4. tant que c'est possible, faire
     début
     choisir une arête uv, avec
     u ∈ { origine(μ_i), extrémité(μ_i) }, telle que NUM [v]=0
     μ_i ← μ_i U { uv }
     UTILISE [uv] ← 1
     empiler (v, PIL)
     MARQ [v] ← MARQ [v]+1
     NUM [v] ← i
     fintantque
5. dépiler (PIL, x)
6. tant que PIL non vide, faire
     début
     i ← i+1
7.   répéter
          dépiler (PIL, x)
     jusqu'à ce que (MARQ [x]<2) ou (PIL vide)
8.   si PIL non vide, alors
          début
          choisir l'arête xy telle que UTILISE [xy]=0
          μ_i ← { xy }
          UTILISE [xy] ← 1
          MARQ [x] ← MARQ [x]+1
          MARQ [y] ← MARQ [y]+1
          NUM [x] ← i
          NUM [y] ← i
          empiler (y, PIL)
          x ← y
9.        tant que c'est possible, faire
               début
               choisir une arête xy telle que UTILISE [xy]=0
               et NUM [y]<i
               μ_i ← μ_i U { xy }
               UTILISE [xy] ← 1
               MARQ [y] ← MARQ [y]+1
               NUM[y] ← i
               empiler (y, PIL)
               fintantque
10.         dépiler (PIL, x)
          finsi
     fintantque
fin.
```

Cette version de l'algorithme permet de préciser comment on construit une chaîne maximale contenant une arête e non marquée. Par ailleurs, la pile a été introduite pour obtenir une complexité en temps linéaire de l'algorithme.

3.4. Complexité de l'algorithme

Grâce au tableau UTILISE, chaque arête est utilisée une fois dans l'algorithme. De plus, chaque sommet est empilé au plus deux fois dans PIL, chaque élément de MARQ, NUM, UTILISE reçoit au plus deux affectations, donc le nombre d'opérations élémentaires utilisé par PARTIT (G) est en $O(n+m)$; comme $m=3n/2$, l'algorithme a une complexité en temps en $O(n)$.

La complexité en espace est également linéaire par rapport à n: les tableaux MARQ, NUM et UTILISE et la pile PIL utilisant $O(n+m)$ emplacements.

Remarque: Dans [6], Lovasz a montré que pour tout graphe impair (graphe dont tous les sommets sont de degré impair), $\pi(G)=|X(G)|/2$. Nous retrouvons ici ce résultat dans le cas très particulier des graphes cubiques, mais par une procédure algorithmique, ce qui ne paraît pas être le cas de la preuve de Lovasz.

De plus, nous pensons que notre résultat peut être notablement amélioré:

Notons PAR/IMPAIR le problème:

donnée: un graphe G dont tous les sommets sont impairs;

question: trouver une partition en chaîne optimale des arêtes de G.

Nous proposons la:

CONJECTURE 2: PAR/IMPAIR est polynômial.

BIBLIOGRAPHIE

1. C. BERGE, *Graphes et hypergraphes*, Dunod, Paris, 1970.
2. A. DONALD, *An Upper Bound for the Path Number of a Graph*, J. of Graph Theory, vol. 4, 1980, p. 189-201.
3. M. R. GAREY et D. S. JOHNSON, *Computers and Intractability*, Freeman, San Francisco, 1979.
4. F. HARARY, *Covering and Packing in Graphs I*, Ann. N.Y. Acad. Sc., vol. 175, 1970, p. 198-205.
5. F. HARARY et J. SCHWENK, *Covering and Packing in Graphs II*, in: R. C. READ ed., Graph Theory and Computing, Acad. Press, New York, 1972.
6. L. LOVASZ, *On Covering in Graphs*, in: Theory of graphs, Erdos, Katona eds., Tihany, Acad. Press, New York, 1968, p. 231-236.
7. B. PEROCHE, *Connexité et indices de recouvrement dans les graphes*, Thèse d'État, Université Paris-Nord, 1982.
8. B. PEROCHE, *NP-completeness of Some Problems of Partitioning and Covering in Graphs*, Discrete Applied Math., vol. 8, 1984, p. 195-208.
9. R. STANTON, D. D. COWAN et L. O. JAMES, *Some Results on Path Numbers*, Proc. of the Louisiana Conf. on Combinatorics, Graph Theory and Computing, 1970, p. 112-135.