



HAL
open science

Algorithme polynomial pour les graphes isomorphes, tous les cas

Mohamed Mimouni

► **To cite this version:**

Mohamed Mimouni. Algorithme polynomial pour les graphes isomorphes, tous les cas. 2018. <hal-01822100v2>

HAL Id: hal-01822100

<https://hal.science/hal-01822100v2>

Preprint submitted on 4 Jul 2018 (v2), last revised 26 Jan 2020 (v5)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Algorithme polynomial pour les graphes isomorphes, tous les cas

Mohamed mimouni

20 rue kadissia 60000 Oujda Maroc

mimouni.mohamed@gmail.com

RESUME. Dans ce papier, je propose un algorithme capable de résoudre le problème de graphes isomorphes en temps polynomial. En premier lieu, je défini un pseudo-arbre qui nous permet de définir pour chaque sommet une étiquette ou un label. En second lieu, j'applique le pseudo-arbre pour le premier graphe puis je calcul les étiquette de chaque sommet de premier graphe, puis je fais la même chose pour le deuxième graphe. En troisième lieu je cherche pour chaque sommet de graphe1 les sommets de graphe2 qui ont la même étiquette, si ou moins un sommet de premier graphe son étiquette n'est pas dans les étiquette de deuxième graphe on déduit que les deux pseudo-arbres ne sont pas isomorphes. Dans l'autres cas je génère des solutions et je les vérifie en temps polynomiale... Cet algorithme permet donc pour les graphes isomorphes de calculer l'image de chaque sommet en temps polynomiale.

ABSTRACT. In this paper, I propose an algorithm capable of solving the problem of isomorphic graphs in polynomial time. First, I define a pseudo-tree that allows us to define for each vertex a label or a label. In the second place, I apply the pseudo-tree for the first graph and I calculate the labels of each vertex of the first graph, then I do the same thing for the second graph. Thirdly, for each vertex of graph 1, I look for the vertices of graph2 which have the same label, if at least a vertex of the first graph its label is not in the labels of the second graph we deduce that the two pseudo-trees are not isomorphic. In the other case I generate solutions and I check them in polynomial time ... This algorithm thus makes it possible for the isomorphic graphs to compute the image of each vertex in polynomial time.

MOTS-CLES : graphes isomorphes, isomorphisme, automorphisme, hypergraphes isomorphes

KEY-WORDS: isomorphic graphs, isomorphism, automorphism, isomorphic hypergraphs

Définitions

1 Un graphe est un ensemble de points nommés *nœuds* (parfois *sommets* ou *cellules*) reliés par des traits (segments) ou flèches nommées *arêtes* (ou *liens* ou *arcs*).

2 Un hypergraphe est un graphe dont les arêtes relient un ou deux ou plusieurs sommets.

3 un isomorphisme f entre les graphes G et H est une bijection entre les sommets de G et ceux de H , telle qu'une paire de sommets $\{u, v\}$ de G est une arête de G si et seulement si $\{f(u), f(v)\}$ est une arête de H .

Construire un pseudo-arbre

Soit G un graphe non orienté et s un sommet. Le pseudo-arbre en tête s se fait de la manière suivante :

1. Le sommet s dans niveau 0.
2. Les adjacents de s dans niveau 1.
3. Les adjacents de chaque sommet de niveau1 sans répétition, qui ne sont pas déjà dans les autres niveaux, dans le niveau2.
4. Fin quand il n'y a pas des nouveaux adjacents.

Si dans le cas (cas de graphe non connecté) où il reste des sommets de graphe G , qui ne sont pas dans le pseudo-arbres on doit génère d'autres pseudo-arbres.

Etiquette

Pour chaque sommet de G une étiquette se forme de quatre éléments :

1. n le niveau où il se trouve.
2. h le nombre des adjacents qui sont dans le niveau en haut.
3. m le nombre des adjacents qui sont dans le même niveau.
4. b le nombre des adjacents qui sont dans le niveau en bas.

Algorithme

Soit G un graphe $\{g_1, g_2, g_3, \dots, g_n\}$ et H un graphe $\{h_1, h_2, h_3, \dots, h_n\}$

Les deux graphes G et H sont isomorphe si pour la fonction isomorphisme $f(g_i) = h_i$, on a l'étiquette de g_i égal l'étiquette de h_i .

$$label(g_i) = label(h_i) = n - h - m - b$$

Pour le graphe G :

Construire un ou plusieurs pseudo-arbres de manière d'avoir une étiquette pour chaque sommet de graphe G .

Pour le graphe H :

Construire des pseudo-arbres et calculer les étiquettes pour chaque sommet de H .

Chercher dans le graphe G les sommets qui ont la même étiquette.

Si au moins un sommet de H a une étiquette inégale des étiquettes des sommet de H , on modifier l'entête et refaire les calculs des étiquettes des sommets de H .

Si chaque sommet de H a une étiquette égale a des étiquettes de graphe G , on fait une validation ou une vérification des solutions. Si la solution n'est pas validée on change les entêtes.

Fin si une solution est validée ou vérifier.

Preuve mathématique

Soit dans deux graphes G et H , deux groupes des pseudo-arbres P pour G et Q pour H .

On a G et P sont isomorphes, car P est construit à partir de graphe G sans modification. Même chose pour Q et H .
donc si les deux P et Q sont strictement isomorphes alors G et H sont deux graphes isomorphes.

Donc on peut dire que le problème d'isomorphisme de G et H se réduit en temps polynomial vers un problème d'isomorphisme entres les pseudo-arbres P et Q .

Premier cas un sommet de H ($n-h-m-b$) n'a pas d'image dans G donc $f(g_1) \neq h_1$

Deuxième cas $f(g_i) = h_t \Leftrightarrow label(g_i) = label(h_t)$ pour chaque sommet de G et chaque sommet de H .

On ne peut pas affirmer que les deux graphes son isomorphes seulement si la solution trouver est valider.

Donc l'algorithme recoudre l'isomorphes des deux graphes.

Complexité d'algorithme

Chaque sommet doit être dans une pseudo-arbres, la complexité est donc $O(n^2)$.

Pour le deuxième graphe la complexité peut atteindre $O(n^3)$.

La calcule des étiquettes de chaque sommet est de $O(n)$.

La validation de solution est de $O(n^2)$.

Donc cet algorithme est polynomial.

Exemple1 : graphes connecte

Soit les deux graphes **G** et **H**, présenter par listes des adjacents suivants :

Graphe G	Graphe H
a:b,d,e,h	1:4,6,7,8,9
b:a,c,e,g,i	2:6,7,8,9
c:b,f,g,h,i	3:5,6
d:a,e,f,i	4:1,5,6,9
e:a,b,d,h,i	5:3,4,7,9
f:c,d	6:1,2,3,4,8
g:b,c,h	7:1,2,5,9
h:a,c,e,g	8:1,2,6
i:b,c,d,e	9:1,2,4,5,7

Pseudo-arbre de tête a pour graphe **G** :

a	niveau 0
b,d,e,h	niveau 1
f,i,c,g	niveau 2

Label de chaque sommet de graphe **G** :

sommet	adjacents	niveau	haut	même	bas	label
a	b,d,e,h	0	vide	a	b,d,e,h	0-0-0-4
b	a,c,e,g,i	1	a	b,d,e,h	f,i,c,g	1-1-1-3
d	a,e,f,i	1	a	b,d,e,h	f,i,c,g	1-1-1-2
e	a,b,d,h,i	1	a	b,d,e,h	f,i,c,g	1-1-3-1
h	a,c,e,g	1	a	b,d,e,h	f,i,c,g	1-1-1-2
f	c,d	2	b,d,e,h	f,i,c,g	vide	2-1-1-0
i	b,c,d,e	2	b,d,e,h	f,i,c,g	vide	2-3-1-0
c	b,f,g,h,i	2	b,d,e,h	f,i,c,g	vide	2-2-3-0
g	b,c,h	2	b,d,e,h	f,i,c,g	vide	2-2-1-0

Après avoir fini pour graphe **G**, on passe au graphe **H** :

Label (a)=0-0-0-4 donc pour graphe **H**, on teste $f(a)=2$, $f(a)=4$, $f(a)=5$ et $f(a)=7$.

Pseudo-arbre de tête 2,4,5 et 7 pour graphe **H** :

2	4	5	7	niveau 0	a
6,7,8,9	1,5,6,9	3,4,7,9	1,2,5,9	niveau 1	b,d,e,h
3,1,4,5	3,8,2,7	6,1,2	6,8,3,4	niveau 2	f,i,c,g
		8		niveau 3	

Nous avons pour le sommet **5** quatre niveaux donc $f(a) \neq 5$.

Les labels... :

tête 2

sommet	adjacents	niveau	haut	même	bas	label	images
2	6,7,8,9	0	vide	2	6,7,8,9	0-0-0-4	a
6	1,2,3,4,8	1	2	6,7,8,9	3,1,4,5	1-1-1-2	d,h
7	1,2,5,9	1	2	6,7,8,9	3,1,4,5	1-1-1-2	d,h
8	1,2,6	1	2	6,7,8,9	3,1,4,5	1-1-1-1	vide

Pour le sommet **8** le label (1-1-1-1) n'existe pas dans les labels de graphe **G**, donc $f(a) \neq 2$.

tête 4

sommet	adjacents	niveau	haut	même	bas	label	images
4	1,5,6,9	0	vide	4	1,5,6,9	0-0-0-4	a
1	4,6,7,8,9	1	4	1,5,6,9	3,8,2,7	1-1-2-2	vide

Pour le sommet **1** le label (1-1-2-2) n'existe pas dans les labels de graphe **G**, donc $f(a) \neq 4$.

tête 7

sommet	adjacents	niveau	haut	même	bas	label	images
7	1,2,5,9	0	vide	7	1,2,5,9	0-0-0-4	a
1	4,6,7,8,9	1	7	1,2,5,9	6,8,3,4	1-1-1-3	b
2	6,7,8,9	1	7	1,2,5,9	6,8,3,4	1-1-1-2	d,h
5	3,4,7,9	1	7	1,2,5,9	6,8,3,4	1-1-1-2	d,h
9	1,2,4,5,7	1	7	1,2,5,9	6,8,3,4	1-1-3-1	e
6	1,2,3,4,8	2	1,2,5,9	6,8,3,4	vide	2-2-3-0	c
8	1,2,6	2	1,2,5,9	6,8,3,4	vide	2-2-1-0	g
3	5,6	2	1,2,5,9	6,8,3,4	vide	2-1-1-0	f
4	1,5,6,9	2	1,2,5,9	6,8,3,4	vide	2-3-1-0	i

Vérifier les solutions :

sommet	adjacents	images	images adjacents	sommet	adjacents	etat
7	1,2,5,9	a	(b),(d,h),(d,h),(e)	a	b,d,e,h	ok
1	4,6,7,8,9	b	(i),(c),(a),(g),(e)	b	a,c,e,g,i	ok
2	6,7,8,9	d	(e),(a),(g),(e)	d	a,e,f,i	non
2	6,7,8,9	h	(c),(a),(g),(e)	h	a,c,e,g	ok
5	3,4,7,9	d	(f),(i),(a),(e)	d	a,e,f,i	ok
5	3,4,7,9	h	(f),(i),(a),(e)	h	a,c,e,g	non
9	1,2,4,5,7	e	(b),(d,h),(i),(d,h),(a)	e	a,b,d,h,i	ok
6	1,2,3,4,8	c	(b),(d,h),(f),(i),(g)	c	b,f,g,h,i	ok
8	1,2,6	g	(b),(d,h),(c)	g	b,c,h	ok
3	5,6	f	(d,h),(c)	f	c,d	ok
4	1,5,6,9	i	(b),(d,h),(c),(e)	i	b,c,d,e	ok

Dans ce tableau on a f(2)=h et pas d. et f(5)=d et pas h.

La vérification à donner 2 erreurs, on doit alors faire une autre vérification :

sommet	adjacents	images	images adjacents	sommet	adjacents	etat
7	1,2,5,9	a	(b),(h),(d),(e)	a	b,d,e,h	ok
1	4,6,7,8,9	b	(i),(c),(a),(g),(e)	b	a,c,e,g,i	ok
2	6,7,8,9	h	(c),(a),(g),(e)	h	a,c,e,g	ok
5	3,4,7,9	d	(f),(i),(a),(e)	d	a,e,f,i	ok
9	1,2,4,5,7	e	(b),(h),(i),(d),(a)	e	a,b,d,h,i	ok
6	1,2,3,4,8	c	(b),(h),(f),(i),(g)	c	b,f,g,h,i	ok
8	1,2,6	g	(b),(h),(c)	g	b,c,h	ok
3	5,6	f	(d),(c)	f	c,d	ok
4	1,5,6,9	i	(b),(d),(c),(e)	i	b,c,d,e	ok

Maintenant aucune erreur donc les deux graphes G et H sont isomorphes et voici la solution finale :

sommet	Image
a	7
b	1
c	6
d	5
e	9
f	3
g	8
h	2
i	4