Isomorphisme de graphe: extension polynomiale et optimisation

Daniel Porumbel*

Université d'Artois, LGI2A, Béthune

- Problème d'isomorphisme de graphe (`GI`):
  - Étant donnés deux graphes $G(V, E)$ et $G'(V', E')$, sont-ils isomorphes? Il faut chercher une bijection $f : V \to V'$ tel que:

$$\{a, b\} \in E \ \underline{\text{si et seulement si}}\ \{f(a), f(b)\} \in E'$$

  - Classe de complexité: encore inconnue, on ne sais pas si `GI` est NP-complet ou pas
  - Il existe des dizaines d'algorithmes d'isomorphisme mais pas beaucoup qui utilisent des techniques de recherche opérationnelle
- Un nouveau algorithme (`GI-EXT`) a été proposé
  - `GI-EXT` est basé sur une <u>extension polynomiale</u> de graphe qui construit des graphes plus hétérogènes, plus facilement comparables

Invariants de sommets et d'arêtes

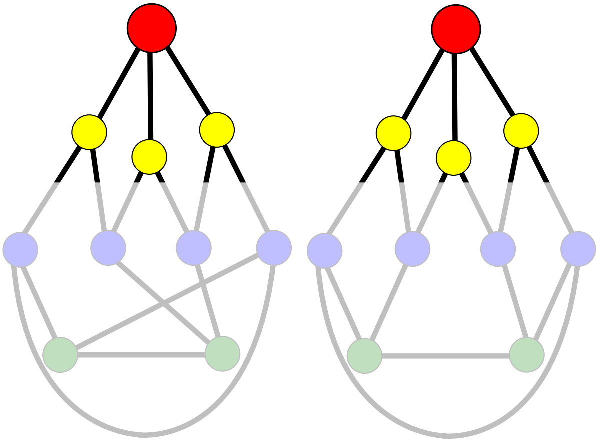
Des nombreux invariants ont été proposés depuis des années [Scott Fortin, The Graph Isomorphism Problem, 1996, University of Alberta]

- Le plus simple exemple: le degré d_a (pour tout sommet a)
- S'il existe un isomorphisme f tel que $f(a) = a'$, alors $d_a = d_{a'}$

Tout invariant d'arête doit avoir la même valeur sur $\{a, b\} \in E$ et sur $\{f(a), f(b)\}$.

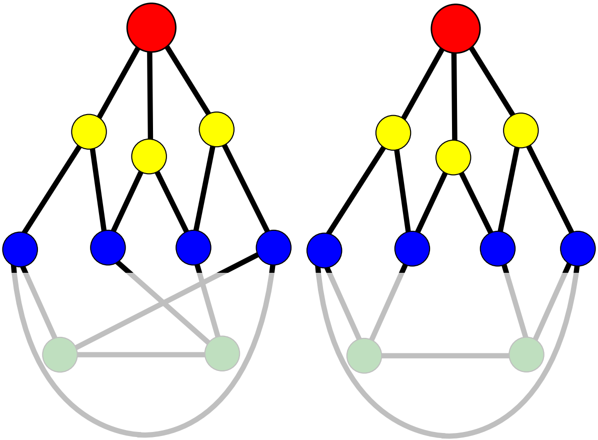
- Si $\{a, b\}$ fait partie de m cycles, alors $\{f(a), f(b)\}$ doit faire partie de m cycles aussi.
- Le nombre de trajet de longueur 5 entre a et b doit être égal au nombre de trajets de longueur 5 entre $f(a)$ et $f(b)$
- ......

L'extension: plusieurs invariants, une seule construction



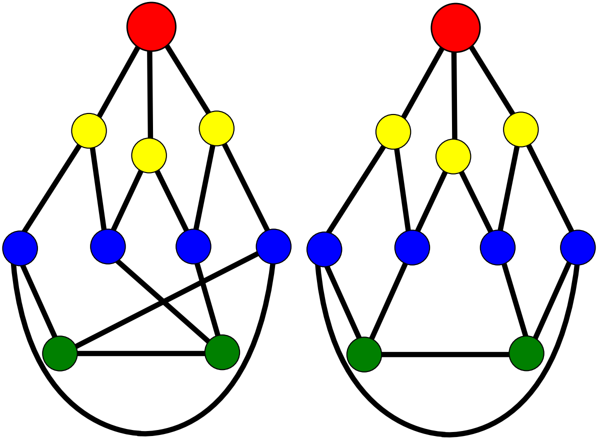
Est-il possible de mapper Rouge à Rouge (via un isomorphisme)?

L'extension: plusieurs invariants, une seule construction



Est-il possible de mapper Rouge à Rouge (via un isomorphisme)?
Peut-être: les deux sommets Rouges ont 3 voisins directs et 4 voisins d'ordre 2 (distance 2)

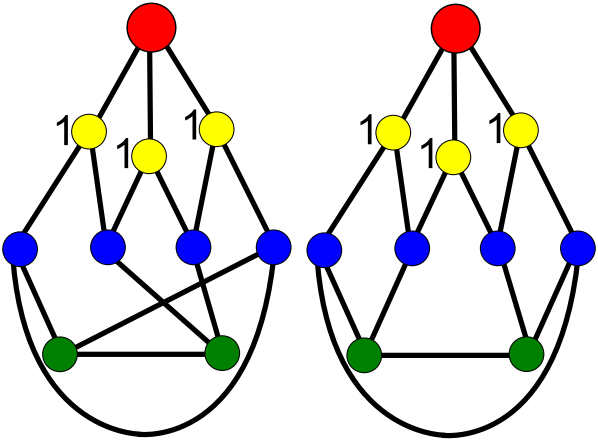
L'extension: plusieurs invariants, une seule construction



Est-il possible de mapper Rouge à Rouge (via un isomorphisme)?

Isomorphisme possible: les deux sommets Rouges ont deux voisins d'ordre 3

L'extension: plusieurs invariants, une seule construction

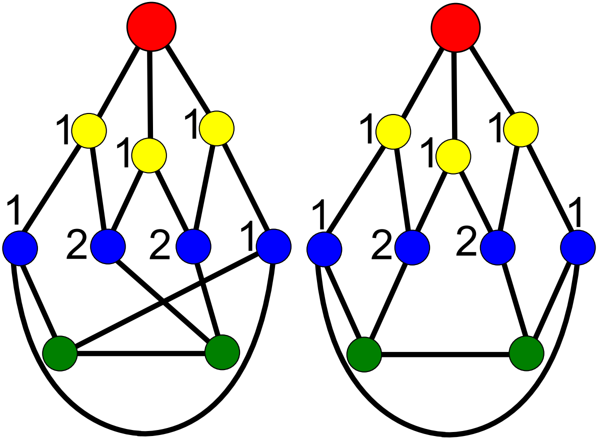


Définir la matrice D^1 :

D^1_{ij}=nombre de chemins de longueur 1 (liaisons directes) entre i et j

(la matrice d'adjacence)

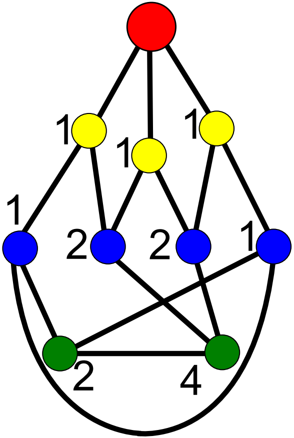
L'extension: plusieurs invariants, une seule construction



Définir la matrice D^2:

D^2_{ij}=nombre de chemins de longueur 2 entre i et j

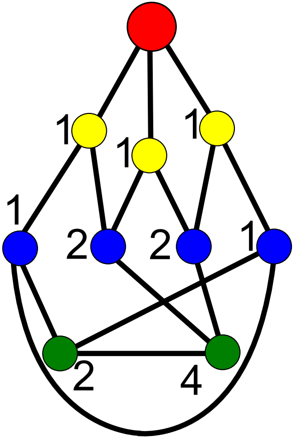
L'extension: plusieurs invariants, une seule construction

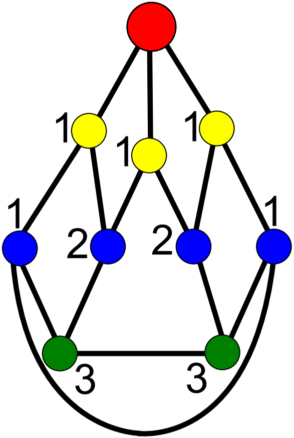


Définir la matrice D^3:

D^3_{ij}=nombre de chemins de longueur 3 entre i et j

L'extension: plusieurs invariants, une seule construction





Le valeurs de $D^3[vert, rouge]$ sont différentes dans les deux graphes

$\rightarrow$

Un isomorphisme ne peut pas mapper Rouge à Rouge

Définition de l'extension de graphe

Soit un graphe G avec matrice d'adjacence D:

- $D^1 = D$: D^1_{ij}=nombre de chemins directs (arêtes) entre i et j
- D^2: D^2_{ij}= nombre de chemins de longueur 2 entre i et j

...

- D^n: D^n_{ij}= nombre de chemins de longueur n entre i et j

$\Longrightarrow$ Définir $\overline{D} = \alpha D^1 + \alpha^2 D^2 + \cdots + \alpha^n D^n$. Cette matrice représente la matrice d'adjacence du graphe étendu (de l'`extension`)

$$\Longrightarrow$$

Deux graphes sont isomorphes si et seulement si leurs extensions sont isomorphes

Définition de l'extension de graphe

Soit un graphe G avec matrice d'adjacence D:

- $D^1 = D$: D^1_{ij}=nombre de chemins directs (arêtes) entre i et j
- D^2: D^2_{ij}= nombre de chemins de longueur 2 entre i et j

...

- D^n: D^n_{ij}= nombre de chemins de longueur n entre i et j

$\Longrightarrow$ Définir $\overline{D} = \alpha D^1 + \alpha^2 D^2 + \cdots + \alpha^n D^n$. Cette matrice représente la matrice d'adjacence du graphe étendu (de l'`extension`)

$$\Longrightarrow$$

Deux graphes sont isomorphes si et seulement si leurs extensions sont isomorphes

Problème d'optimisation

Conditions nécessaires pour que f soit un isomorphisme:

- $\overline{D}_{ab} = \overline{D}_{f(a),f(b)}$

Problème d'optimisation

- Espace de recherche: toutes les injections $f : V \to V'$
- Fonction objectif à minimiser:

$$\sum_{a,b \in V^2} [\overline{D}_{ab} \neq \overline{D'}_{f(a),f(b)}],$$

l'opérateur $[expr]$ est 1 si l'expression $expr$ est vraie ou 0 sinon.

Restreindre l'espace de recherche

Peut-il exister un isomorphisme f tel que $f(a) = a'$? Réponse négative en plusieurs situations:

- Les valeurs de la ligne a de D^1 différentes des valeurs de la ligne a' de D' (différence de degré)
- Les valeurs de la ligne a de D^k différentes des valeurs de la ligne a' de D'^k (différence de nombre de voisins d'ordre k)
- $D^k_{aa} \neq D'^k_{a'a'}$: pas le même nombre de cycles de longueur k

Difficile de construire deux graphes non-isomorphes et qui respectent les propriétés ci-dessus.

Algorithme à base d'extensions: conclusions et perspectives

Un algorithme en deux étapes a été proposé [D.C. Porumbel, Isomorphism Testing Via Polynomial-Time Graph Extensions, Journal of Mathematical Modelling And Algorithms, 2011]:

1. Construire l'extension en $O(n^4)$ au maximum, $n = |G|$
   - Chaque pas d'extension demande $O(n^3)$
   - On a besoin de n extensions au maximum $\Rightarrow O(n^4)$
2. Appliquer une heuristique sur les graphes étendus
   - Définir une fonction objectif sur la matrice d'adjacence étendue
   - Nous avons utilisé le `Recuit Simulé`

Perspectives: L'étape 2 est très rapide et elle peut être remplacée par une méthode d'optimisation exacte

Implémentation pratique

L'algorithme a été testé en pratique sur plusieurs classes de graphes

graphes aléatoires résolution rapide sans problèmes (comme beaucoup d'autres algorithmes)

graphes régulières aléatoires résolution en $O(n^3)$ même pour des graphes à densité proche de $\frac{1}{2}$

Il y a des raisons: après quelques étapes d'extension, les valeurs D^k deviennent très hétérogènes même si chaque sommet a le même degré

- Même si a et b ont le même nombre de voisins, ils n'ont pas le même nombre de voisins communs: D^2_{ab} varie beaucoup (sauf pour les graphes *strongly-regular*

Conclusions

- L'algorithme (GI-Ext) nécessite juste des manipulations de matrices
- Il est plus facile à implémenter que d'autres algorithmes
  - les algorithmes nauty/bliss ne sont pas toujours plus efficaces et ils ont été raffiné et améliorés depuis les années 1970s
  - les algorithmes à base de programmation par contraintes: une (nouvelle) approche, mais même les constructions à base de filtrages sont plus complexes que l'extension de graphe
- Une méthode exacte qui utilise la même extension a été envisagée