Chapître III.
Classes de graphes

Tous les graphes qu'on va considerer dans ce chapître sont non orientés.

Il y a un nombre enorme de diverse classes de graphes etudiées dans la littérature (où une classe de graphes est simplement un ensemble de graphes).

Exemple:

-) graphes planaires
-) graphes biparti
-) graphes de permutation
-) graphes cobiparti
-) graphes d'intervalles

1. Livres recommandés

A. Brandstädt, V. Bang Le, J.P. Spinrad,
Graph classes: a survey,
SIAM, 1999.

M.C. Golumbic,
Algorithmic Graph Theory and Perfect Graphs,
Academic Press, 1980

2. Algorithmes de reconnaissance

Soit $\mathcal{G}$ une classe de graphe comme par exemple les graphes planaire.

Définition. Un *algorithme de reconnaissance* pour la classe de graphes $\mathcal{G}$ est un algorithme qui, étant donné un graphe G, verifie si le graphe G appartient à la classe $\mathcal{G}$ ou non. C'est-à-dire, soit l'algorithme accepte l'entrée G en sortant "OUI", soit l'algorithme refuse l'entrée G en sortant "NON".
Souvent l'algorithme sort une certaine representation du graphe G si $G \in \mathcal{G}$.

Une telle representation (modéle, diagramme etc.) est une representation d'un graphes different de liste d'adjacence et d'une matrice d'adjacence. Elle peut être utile pour construire un algorithme efficace pour un ou plusieurs problèmes NP-complet sur la classe de graphes $\mathcal{G}$.

3. Graphes biparti

Définition. Un graphe $G = (V, E)$ est *biparti* s'il y a une partition de V en deux sous-ensembles X et Y tel que chaque arête de G est incident à un sommet de X et à un sommet de Y. On note $G = (X, Y, E)$.

Dans un graphe biparti $G = (X, Y)$ les sous-ensembles X et Y sont ensembles stable. Donc $\chi(G) \leq 2$. Vice versa, si $G = (V, E)$ est 2-colorable alors il y a une partition de V dans deux ensembles stable X et Y, donc G est biparti. Souvent on appele X et Y les classes de couleur d'un graphe biparti $G = (X, Y, E)$.

Exercice: 1. Construire un algorithme qui verifie un temps linéaire si un graphe donné est biparti.
2. Construire un algorithme qui verifie un temps linéaire si un graphe donné est le complement d'un graphe biparti.

Théorème. *Un graphe $G = (V, E)$ est biparti si et seulement si il n'y a pas de cycle de longeur impair de G.*

Démonstration.
Soit $C = (c_1, c_2, \ldots, c_{2k+1})$, $k \geq 1$, un cycle impair de G. Donc $\chi(G) \geq \chi(G[C]) \geq \chi(C_{2k+1}) = 3$, où C_{2k+1} est un "cycle sans chord de longeur $2k+1$". Donc G n'est pas biparti.

Soit G un graphe sans cycle de longeur impair. C'est evident que G est biparti ssi chaque composant connexe de G est biparti. C'est-à-dire on peut supposer que G est connexe.

On applique un *recherche en largeur* à G qui demarre d'un sommet v. On obtient une partition de V en "niveaux" L_i où l'$i$eme niveau est l'ensemble de tous les sommet un distance i de v.

On affirme que chaque L_i est un ensemble stable. Sinon il y a deux sommet $x, y \in L_i$ tel que $\{x, y\}$. On regarde le chemin P_x de x vers v qui passe toujours d'un sommet à son predecesseur dans le recherche en largeur, et le chemin P_y de y vers v qui passe toujours d'un sommet à son predecesseur dans le recherche en largeur. Soit w le sommet commun de P_x et P_y le plus proche à x (resp. y). Donc $(x \quad w \quad y)$ est un cycle impair.

Finalement si $x \in L_i$ et $y \in L_{i+t}$ alors $t \geq 2$ implique $\{x, y\} \notin E$. Donc $X = \bigcup_{\{i:\ i \text{ pair}\}} L_i$ et $Y = \bigcup_{\{i:\ i \text{ impair}\}} L_i$ sont ensembles stable, $X \cap Y = \emptyset$ et $X \cup Y = V$. Donc $G = (X, Y, E)$ est un graphe biparti. $\square$

Exercice: 1. Construire un algorithme de reconnaissance pour les graphe biparti qui s'execute en temps linéaire (au pire des cas) en utilisant le théorème au-dessus.

2. Modifier cet algorithme de reconnaissance pour qu'il sort un certificat pour la sortie "G biparti" (partition de V en deux ensembles stable) et qu'il sort aussi un certificat pour la sortie "G n'est pas biparti" (cycle impair).

4. Quelques problèmes NP-complets sur les graphes biparti

On considère un certain problème NP-complet Π sur les graphes biparti.

Entrée: un graphe biparti $G = (V, E)$

Sortie: une solution du problème Π

Il faut déterminer si la “structure” d’un graphe biparti fait le problème Π plus facile tel qu’il y a un algorithme polynomial pour resousre Π si on regarde que des graphe biparti.

Théorème. *Les problèmes* Dominating Set *et* Hamiltonian Circuit *restent NP-complets sur les graphes biparti.*

Les problèmes Clique et Coloring deviennent trivial parce que $\omega(G) \leq 2$ et $\chi(G) \leq 2$ pour tous les graphes biparti.

Exercice: Donner un algorithme linéaire, qui étant donné un graphe biparti G, calcule une clique de plus grande taille possible de G.

Noter que $\omega(G) = \chi(G)$ pour tous les graphes biparti. En effet, tous les graphes biparti sont parfait.

Finalement pour le problème Independent Set il y a un algorithme polynomial pour resoudre le problème sur les graphe biparti. Cet algorithme utilise un algorithme pour calculer un couplage (*angl.* matching) de plus grande taille possible d'un graphe biparti.

5. Graphes split

Définition. Un graphe $G = (V, E)$ est *split* s'il y a une partition de V en deux sous-ensembles S et C tel que S est un ensemble stable de G et C est une clique de G. On note $G = (S, C, E)$.

Exemple:

Théorème. *Un graphe $G = (V, E)$ est un graphe split si et seulement si aucun sous-graphe induit de G est isomorphe à $2K_2$, C_4 ou C_5.*

Pour une démonstration voir le chapître sur les graphes split dans le livre de Golumbic.

Exercice: Construire un algorithme de reconnaissance pour les graphes split en utilisant le théorème au-dessus. Analyser le temps d'execution de cet algorithme.

Définition. Soit $G = (V, E)$ un graphe. On appele $d_1 \leq d_2 \leq \cdots \leq d_n$ la *séquence de degré* de G si $(d_1, d_2, \ldots, d_n)$ est la séquence trié de tous les degrés de sommets de G.

Exemple:

On peut caracteriser les graphes split par leur séquences de degré.

Théorème. *Soit $G = (V, E)$ un graphe et $d_1 \leq d_2 \leq \cdots \leq d_n$ la séquence de degré de G. Soit $m = \max\{i : \ d_i \geq i-1\}$. Alors G est un graphe split si et seulement si*

$$\sum_{i=1}^{m} d_i = m(m-1) + \sum_{i=m+1}^{n} d_i.$$

En plus, si c'est le cas alors $\omega(G) = m$.

Pour une démonstration voir le chapître sur les graphes split dans le livre de Golumbic.

Exercice: Construire un algorithme linéaire pour la reconnaissance de graphes split en utilisant le théorème au-dessus.

6. Graphes d'intervalles

Définition. Un graphe $G = (V, E)$ est un *graphe d'intervalles* s'il y a une collection $\mathcal{I}$ des intervalles de droite des réels et une bijection qui associe à chaque sommet v de G un intervalle I_v de $\mathcal{I}$ tel que:

$$\forall x \in V \;\; \forall y \in V \quad : \;\; \{x, y\} \in E \Leftrightarrow I_x \cap I_y \neq \emptyset.$$

Si G est un graphe d'intervalles on appele une telle collection $\{I_v : \;\; v \in V\}$ le *modèle d'intervalles* de G.

Exemple: un graphe d'intervalles et un de ses modèles d'intervalles

Il y a plus que sept algorithmes linéaire pour la reconnaissance de graphes d'intervalles dans des journaux scientiques de haute niveau. Le premier etait publié en 1976 par Booth et Lueker. Malheureusement il sont tous trop compliqué pour les presenter dans un cours de DEA.

Pour nos algorithmes sur les graphes des intervalles (par exemple pour calculer un ensemble stable de taille maximum) on suppose que le graphe et un modèle d'intervalles sont données.

(Si uniquement le graphe est donné alors il faut appeler un algorithme de reconnaissance qui trouve un modèle d'intervalles si le graphe est vraiment un graphe d'intervalles.)

7. Graphes de permutation

Soit π une *permutation* des entiers $1, 2, \ldots, n$. Nous regardons π comme une séquence $[\pi_1, \pi_2, \ldots, \pi_n]$ des entiers $1, 2, \ldots, n$. Ensuite on note π_i^{-1} la position de i dans la séquence $\pi = [\pi_1, \pi_2, \ldots, \pi_n]$.

Exemple:
$\pi = [3, 1, 4, 5, 6, 2]$
$\pi_3^{-1} = 1$, $\pi_5^{-1} = 4$

Définition. Soit $\pi = [\pi_1, \pi_2, \ldots, \pi_n]$ une *permutation*.
Soit $G[\pi] = (\{1, 2, \ldots, n\}, E_\pi)$ le *graphe d'inversions* de π où

$$\{i, j\} \in E_\pi \Leftrightarrow (i - j)(\pi_i^{-1} - \pi_j^{-1}) < 0.$$

Un graphe $G = (V, E)$ est un *graphe de permutation* s'il y a une permutation π telle que $G \cong G[\pi]$.

Exemple:

$\pi = [2, 6, 1, 5, 4, 3, 7]$

$G[\pi]$:

Donc on peut representer un graphe de permutation G par une permutation π satisfaisant $G \cong G[\pi]$.

En plus, les graphes de permutation sont une classe de graphes d'intersection comme les graphes d'intervalles. Donc il y a un modèle d'intersection pour les graphes de permutation et ce modèle s'appele *diagramme de permutation*.

8. Diagramme de permutation

Soit π une permutation. Alors on obtient un diagramme de permutation de $G[\pi]$ dans la façon suivante.

Soient $\mathcal{L}_1$ et $\mathcal{L}_2$ deux droite paralleles entre eux dans le plan. Marquer les points $1, 2, \ldots, n$ de gauche à droit sur $\mathcal{L}_1$. Marquer les points $\pi_1, \pi_2, \ldots, \pi_n$ de gauche à droit sur $\mathcal{L}_2$. Pour chaque $i \in \{1, 2, \ldots, n\}$, soit L_i le segment de droite qui joint le point i sur $\mathcal{L}_1$ et le point i sur $\mathcal{L}_2$.
On appele un tel modèle d'intersection $\mathcal{L}$ le diagramme de permutation *de $G[\pi]$.*

Le graphe d'intersection de $\mathcal{L}$ est le graphe $G_{\mathcal{L}} = (\{1, 2, \ldots, n\}, E_{\mathcal{L}})$ où

$$\{i, j\} \in E_{\mathcal{L}} \Leftrightarrow \quad L_i \text{ et } L_j \text{ se croit en } \mathcal{L}.$$

En consequence $G[\pi] = G_{\mathcal{L}}$.

Exemple:

$\pi = [3, 1, 5, 6, 4, 2]$

diagramme de permutation de $G[\pi]$ et le graphe de permutation $G[\pi]$:

Le meilleur algorithme connu pour la reconnaissance de graphes de permutation s'execute un temps linéaire. Cet algorithme est tres compliqué. Il etait etabli par J. Spinrad et R. McConnell en 1999. Si le graphe G donné est vraiment un graphe de permutation alors cet algorithme sort un diagramme de permutation de G.