ALGORITHMIQUE DES STRUCTURES DISCRÈTES

Algorithmes sur les graphes
Algorithmes sur les mots

Livre recommandé:

Introduction à l'algorithmique,
T.H. Cormen, C.E. Leiserson, R.L. Rivest,
Dunod, Paris, 1994 (MIT Press 1990).

Algorithmes sur les graphes

1) **Représentation de graphes**
2) **Problèmes NP-complets et classes de graphes**
3) **Coloration et domination**
   - Graphes d'intervalles
   - Graphes de permutation
4) **Algorithmes randomisés**
   - Tri randomisé
   - Coupure minimal dans un graphe
5) **Algorithmes d'approximation**
   - Couverture de sommet
   - Couverture d'un ensemble

Chapître I.
Représentation de graphes

1. Graphes orientés

Un **graphe orienté** G est représenté par un couple (V, A), ou V est l'ensemble de **sommets** de G et A est l'ensemble des **arcs** de G. Si (u, v) est un arc d'un graphe orienté $G = (V, A)$, on dit que (u, v) **part** du sommet u et **arrive** au sommet v. On dit que l'arc (u, v) est **incident** aux sommets u et v.

Représentation graphique:

2. Graphes non orientés

Un **graphe non orienté** G est représenté par un couple (V, E), ou V est l'ensemble de **sommets** de G et E est l'ensemble des **arêtes** de G. On note n le nombre des sommets et m le nombre des arêtes (ou des arcs).

Si $\{u, v\}$ est un arête d'un graphe non orienté $G = (V, E)$, on dit que les sommets u et v sont **adjacents**. Le **degré** $d(v)$ d'un sommet v est le nombre d'arêtes qui lui sont incidentes.

Représentation graphique:

3. Représentation des graphes dans un ordinateur

Il y a deux façons classiques de représenter un graphe:

a) par listes d'adjacence

consiste en un tableau Adj de n listes, une pour chaque sommet de G

et pour chaque $u \in V$, la liste d'adjacence $Adj[u]$ est une liste chaînée de tous les sommets adjacents à u dans G

Exemple:

b) par matrice d'adjacence

consiste en une matrice $|V| \times |V|$
$M = (a_{ij})$ telle que $a_{ij} = 1$ si
$(v_i, v_j) \in A$ (ou $\{v_i, v_j\} \in E$)
et $a_{ij} = 0$ sinon

on suppose que les sommets sont
étiquetés $v_1, v_2, \ldots v_n$

Exemple:

Chapître II.
Problèmes NP-complets

De nombreux problémes d'intérêt théorique ou pratique sont NP-complets. En particulier une grande partie de problèmes des graphes et réseaux est connu d'être NP-complets.

Livre recommandé:
Garey, Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, W.H. Freeman, San Francisco, 1979.

La plupart de problèmes sont trop importants pour qu'on laisse les tomber, mais s'il y a très peu de chance de trouver un algorithme polynomial capable de résoudre un problème NP-complet exactement.

2. Attaquer un problème NP-complet

Supposons notre problème particulier (comme par exemple ENSEMBLE STABLE) est NP-complet. C'est-à-dire il ne faut pas perdre de temps en cherchant un **algorithme efficace** capable de le résoudre **pour toutes les entrées** avec **exactitude**.

Cependant sachant que le probléme est NP-complet nous essayons de relaxer une ou plusieurs des conditions trop strictes et alors nous arrivons à des differents types d'attaque possible:

0) petite entrée
1) algorithmes d'approximation
2) algorithmes randomisés
3) cas particuliers

3. Quelques problèmes de graphes NP-complets

Les problèmes suivants sont des problèmes de graphes NP-complets bien connus est bien étudiés.

Independent Set (ensemble stable)

Clique

Coloring (coloration)

Dominating Set (ensemble dominant)

Hamiltonian Circuit (cycle hamiltonien)

4. Ensemble stable et Clique

Définition. Soit $G = (V, E)$ un graphe non orienté.

1. $V' \subseteq V$ est un *ensemble stable* de G si
$$\forall u \in V' \ \forall v \in V' \ \{u, v\} \notin E.$$
2. $V' \subseteq V$ est une *clique* de G si
$$\forall u \in V' \ \forall v \in V' \ \{u, v\} \in E.$$

On note $\alpha(G)$ la plus grande taille d'un ensemble stable du graphe G.

On note $\omega(G)$ la plus grande taille d'une clique du graphe G.

Observation. $S \subseteq V$ est un ensemble stable du graphe $G = (V, E)$ si et seulement si S est une clique du graphe complementaire de G. Donc $\alpha(G) = \omega(\overline{G})$.

Exemple:

5. Coloration

Définition. Une *coloration* d'un graphe $G = (V, E)$ est une fonction f qui associe a chaque sommet de G une couleur tel que les couleurs de deux sommets adjacents sont differents.
Une k-*coloration* d'un graphe $G = (V, E)$ est une fonction $f : V \Rightarrow \{1, 2, \ldots, k\}$ si

$$\forall u \in V \;\; \forall v \in V \;\; f(u) \neq f(v).$$

On note $\chi(G)$ le plus petit nombre de couleur dans une coloration de G. On appele $\chi(G)$ le *nombre chromatique* de G.

Exemple:

Observation. Une k-coloration f d'un graphe $G = (V, E)$ induit une partition de V d'un k ensemble stable $S_1, S_2, \ldots, S_k$, où $S_i = \{v \in V : \ f(v) = i\}$.

Donc minimiser le nombre de couleurs d'une coloration de G est equivalent à minimiser le nombre d'ensembles stable d'une partition du ensemble de sommet de G en ensemble stable.

Lemme. *Soit $G = (V, E)$ un graphe non orienté, soit S un ensemble stable de G et C une clique de G. Alors $|S \cap C| \leq 1$.*

Par consequent $\chi(G) \geq \omega(G)$ pour un graphe G quelconque.

6. Ensemble dominant

Définition. $V' \subseteq V$ est un *ensemble dominant* si

$$\forall u \in V \setminus V' \;\; \exists v \in V' \;\; \{u, v\} \in E.$$

On note $\gamma(G)$ la plus petite taille d'un ensemble dominant de G.

Exemple:

7. Cycle hamiltonien

Définition. $(v_1, v_2, \ldots, v_n)$ est un *cycle hamiltonien* du graphe $G = (V, E)$ si $(v_1, v_2, \ldots, v_n)$ est un cycle de G qui passe precisement une fois par chaque sommet de G.

On appele un graphe G *hamiltonien* s'il y a un cycle hamiltonien de G.

Exemple:

Exercice: Montrer que le graphe de Petersen est un graphe non hamiltonien.

8. Une collection des problems NP-complets

Formellement les problèmes NP-complets sont des problèmes de decision.

Hamiltonian Circuit
$= \{G :$ il y a un cycle hamiltonien de $G\}$

Independent Set
$= \{(G, k) :$ il y a un ensemble stable $V' \subseteq V$ de G tel que $|V'| \geq k\}$

Clique
$= \{(G, k) :$ il y a une clique $V' \subseteq V$ de G tel que $|V'| \geq k\}$

Dominating Set
$= \{(G, k) :$ il y a un ensemble dominant $V' \subseteq V$ de G tel que $|V'| \leq k\}$

Coloring
$= \{(G, k) :$ il y a une k-coloration $f : V \Rightarrow \{1, 2, \ldots, k\}$ de $G\}$

9. Cas particulier

Beaucoup d'autres problèmes de graphes sont NP-complets.

Heureusement on besoin rarement un algorithme qui resout le problème NP-complet à resoudre pour **tous les graphes**.

Donc un approache pour attaquer un problème NP-complet de graphe est de relaxer la condition disant qu'il faut resoudre le problème pour tous les entrées.

C'est pourquoi il faut resoudre un problème NP-complet Π pour tous les graphe d'une classe $\mathcal{G}$.

Peut-etre "Π reduit sur $\mathcal{G}$" reste NP-complet, peut-etre il y a un algorithme polynomial pour resoudre Π pour tous les graphe de $\mathcal{G}$.