TP11 - Algorithme de Karger

Étude théorique

Un *multigraphe* est un graphe non orienté dans lequel il peut y avoir plusieurs arêtes entre deux sommets. Formellement, on peut considérer qu'un multigraphe est un graphe pondéré $G = (V, E, \rho)$ où $\rho : E \to \mathbb{N}^*$ donne la multiplicité de chaque arête : $\rho(xy) = 3$ signifie qu'il y a trois arêtes reliant le sommet x au sommet y (c'est une autre manière de dire que E est un multi-ensemble).

Une *coupe* est une partition $X \cup Y$ de V, sa *taille* est le nombre d'arêtes (avec multiplicité) qui relient un sommet de X à un sommet de Y. Une *coupure* est dite *minimale* si sa taille l'est.

Question 1. Que peut-on dire de la taille d'une coupure minimale dans un graphe non connexe ? On suppose dans toute la suite que le graphe est connexe.

Les sommets sont numérotés de 0 à $n-1$ et on identifie initialement le sommet k au singleton $\{k\}$. On définit l'opération de *contraction* d'une arête xy :

- les deux sommets x et y disparaissent du graphe, et sont remplacés par un sommet $x \cup y$;
- les éventuelles arêtes qui reliaient x à y disparaissent ;
- pour chaque z différent de x et de y, les arêtes qui reliaient z à x et z à y relient à présent z à $x \cup y$.

On note G/xy le graphe obtenu à partir de G en contractant l'arête xy.

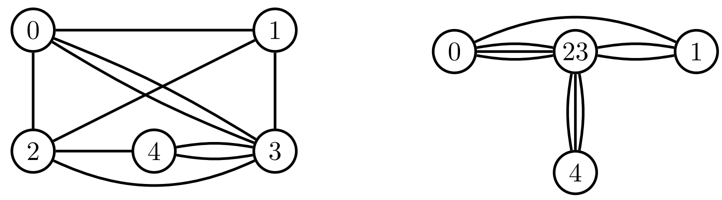

Figure 1 – Contraction de l'arête $\{2, 3\}$.

Question 2. En supposant qu'il existe k coupes minimales, déterminer en fonction de n et k la probabilité qu'une coupe choisie aléatoirement (et uniformément parmi toutes les coupes) soit minimale.

On cherche à calculer efficacement une coupure minimale du graphe G et on proposé l'algorithme suivant.

Algorithme 1 : Algorithme de Karger

Entrée : un multigraphe connexe $G = (V, E)$
Sortie : une coupe de G
tant que $|V| > 2$ **faire**
  choisir une arête uniformément dans E
  $G \leftarrow G/xy$
renvoyer V

Le choix uniforme de l'arête xy se fait en tenant compte des multiplicités : dans l'exemple ci-dessus, on aurait 2 chances sur 10 de choisir l'arête $\{0, 3\}$.

Question 3. Justifier que cet algorithme termine bien en renvoyant une coupure de G.

Question 4. On veut minorer la probabilité p que cet algorithme renvoie une coupe minimale de G. On fixe pour cela une coupe minimale $X \cup Y$, avec C l'ensemble des arêtes *coupées* (reliant un sommet de X à un sommet de Y) et k = |C|.
□ **4.1.** Justifier que |E| $\geqslant nk/2$.
□ **4.2.** En déduire que la probabilité que la première arête contractée n'appartienne pas à C est minorée par $1 - 2/n$.
□ **4.3.** Montrer alors que $p \geqslant 2/n(n-1)$. Que peut-on dire par rapport à la question 2 ?

Question 5. On suppose possible le tirage uniforme d'une arête et la contraction en temps O(n). Donner un algorithme permettant de renvoyer une coupe minimale avec probabilité d'erreur majorée par $1/n^c$ (où c est une constante positive), et calculer sa complexité en temps en fonction de n et c.

Programmation

On s'intéresse à présent à la programmation effective de cet algorithme. On représente un multigraphe par une matrice d'adjacence à coefficients entiers, où le coefficient (i, j) donne le nombre d'arêtes reliant le sommet i au sommet j. On souhaite définir un type permettant de :

- représenter des graphes partiellement contractés, leurs sommets étant des ensembles de sommets du graphe initial ;
- effectuer un tirage uniforme d'une arête en temps au plus linéaire en le nombre de sommets ;
- contracter une arête en temps au plus linéaire en le nombre de sommets.

Question 6. On note $E_n = \{0, \dots, n-1\}$ l'ensemble des sommets du graphe initial et $V_k = \{x_0, \dots, x_{k-1}\}$ les sommets après $n-k$ contractions. Que peut-on dire de V_k par rapport à E_n ? Si l'on contracte une arête $x_i x_j$ pour passer de V_k à V_{k-1}, à quelle opération mathématique cela correspond-il ? Quelle structure de données vous semble appropriée ?

Question 7. Proposer un type OCaml permettant de représenter un multigraphe partiellement contracté, et de réaliser les opérations de tirage uniforme d'une arête et de contraction d'une arête en temps linéaire en le nombre de sommets.

Question 8. Programmer les fonctions de contraction et de tirage uniforme.

Question 9. Programmer l'algorithme de Karger.