# Algorithmes Probabilistes
# TP2

Christos Kotsalos

2 Mars 2020

Vous devez rendre votre travail sur Moodle dans le répertoire TP2 avant la séance d'excercises du 16 Mars 2020. Votre rendu doit être une archive contenant votre rapport ainsi que votre code.

## 1 Introduction

Le but de cette série d'exercices est d'implémenter un algorithme de Monte-Carlo pour trouver la coupe minimale d'un graphe. On commence par rappeler la définition d'une coupe d'un graphe:

**Définition 1.** *Soit un graphe $G = (V, E)$ non-dirigé et une partition de $V$ en deux ensembles $A$ et $B$ ($A \cap B = \emptyset$), une **coupe** $C$ est définie comme l'ensemble des arcs reliants les sommets de $A$ aux sommets de $B$.*

La **coupe minimale** $C_{min}$ d'un graphe est la coupe dont la cardinalité est minimale.

Dans cette série deux algorithmes probabilistes permettant de trouver la coupe minimale d'un graphe, vous sont présentés: l'algorithme de Karger et l'algorithme de Karger-Stein. Une partie de votre travail consistera à implémenter le premier algorithme et à écrire le pseudo-code du deuxième.

## 2 Algorithme de Karger

L'algorithme de Karger est un des algorithmes que vous avez étudié en cours. Cet algorithme est basé sur un opérateur de contraction d'arc qui est défini ci-dessous.

**Définition 2.** *Soit un multigraphe $G = (V, E)$ sans boucle, une **opération de contraction** d'un arc $e = \{u, v\}$ avec $e \in E$ et $u, v \in V$ est définie comme:*

1. *Remplacer le sommet $u$ et $v$ par un nouveau sommet $w$.*

2. *Remplacer tous les arcs* $\{u, x\}$ *et* $\{v, x\}$ *par un arc* $\{w, x\}$

3. *Supprimer les boucles sur* $w$.

*On dénote cette contraction par* $G/e$.

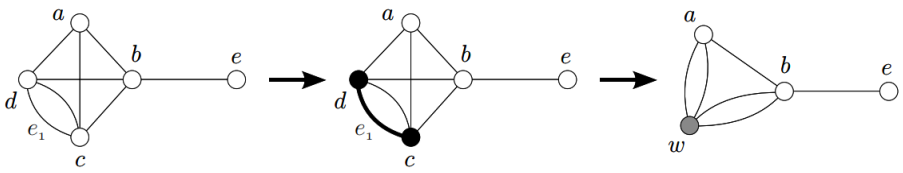


Figure 1: Exemple d'opération de réduction sur l'arc $e_1$: les sommets c et d ont été remplacés par le sommet w, les arcs entre c et d ont été supprimés et les arcs reliant c et d au reste du graphe ont été reconnectés au nouveau sommet w.

La figure 1 illustre un exemple de contraction sur un arc reliant deux sommets. Avec l'opérateur de contraction, l'algorithme s'écrit simplement comme:

**Input:** un graphe $G$ non-orienté composé de $n$ sommets.
  $C \leftarrow \emptyset$
  $s \leftarrow \infty$
  **for** $i = 1 \rightarrow \frac{n^2}{2}$ **do**
    **repeat**
      $e$ un arc de $G$ choisi au hasard.
      $G/e$
    **until** $|V| = 2$
    **if** $|\text{cut}(G)| < s$ **then**
      $C \leftarrow \text{cut}(G)$
      $s \leftarrow |C|$
    **end if**
  **end for**
  **return** $C$

Un avantage de cet algorithme est que son implémentation reste très simple en comparaison avec les algorithmes déterministes trouvant la coupe minimale d'un graphe.

## 2.1 Travail à accomplir

Implémentez cet algorithme en utilisant le langage de programmation C++ ou Python. Vous trouverez sur Moodle (TP2 → Graphs) un graphe de petite taille (g_8.txt) pour tester votre implémentation.

On vous a montré en cours que la borne inférieure de la probabilité qu'une suite de $n-2$ contractions trouve la coupe minimale est de $2/n^2$ (avec $n = |V|$). Vous

avez aussi vu que l'algorithme de Karger avec $n^2/2$ itérations a une probabilité de succès est de $1 - e^{-1} \approx 0.63$. Avec les graphes fournis sur Moodle (TP2 $\rightarrow$ Graphs), dont la coupe minimale est connue, vérifiez expérimentalement la probabilite de succès pour une suite de $n - 2$ contractions ainsi que pour l'algorithme de Karger.

Les estimations sont-elles optimistes, pessimistes ou bien correctes? Si vos estimations diffèrent de la théorie, tentez de l'expliquer.

Pendant le cours, vous avez vu que contracter un graphe jusqu'à ce qu'il ne reste que deux sommets a une complexité en temps de $O(n^2)$. Mesurez expérimentalement la complexité d'une suite de $n - 2$ contractions et présentez vos résultats clairement de sorte à ce que la complexité en temps de votre implémentation soit mise en évidence. Vous pouvez aussi utiliser les graphes mis à disposition sur Moodle.

# 3 L'algorithme de Karger-Stein

L'algorithme de Karger-Stein est aussi un algorithme probabiliste permettant de trouver la coupe minimale d'un graphe. Son avantage est qu'il a une complexité en temps de $O(n^2 logn)$ alors que l'algorithme de Karger a une complexité en temps de $O(n^4)$ en appliquant $n^2$ itérations.

L'idée de l'algorithme de Karger-Stein est de grouper les opérations de contraction en deux phases successives:

1. La phase $\ll$ *sûre* $\gg$ réduit partiellement le graphe de $n$ sommet à $\lfloor n/\sqrt{2} \rfloor$ sommets en applicant $n - \lfloor n/\sqrt{2} \rfloor$ contractions aléatoires.
2. La phase $\ll$ *risquée* $\gg$ construit un arbre binaire de graphes partiellement contractés jusqu'à ce que les graphes soient composés de 2 sommets (c.f. le cours d'algorithmes probabilistes).

## 3.1 Travail à accomplir

Écrivez le pseudo-code de l'algorithme de Karger-Stein afin que cet algorithme retourne la coupe minimale. Pensez à commenter votre pseudo-code.

Calculez aussi la probabilité de choisir un mauvais arc dans la phase $\ll$ *sûre* $\gg$. Comprenez-vous pourquoi cette phase est considérée comme *sûre* ?

# 4 Consignes

Votre code implémentant l'algorithme de Karger doit être écrit en C++ ou Python et doit être facilement compilé. Si vous utilisez une librairie externe, par exemple pour représenter les graphes, celle-ci doit aussi être incluse avec votre code.

Si vous utilisez une librairie externe implémentant un opérateur de contraction ou l'algorithme de Karger, vous ne devez pas l'utiliser pour implémenter votre version de l'algorithme de Karger. L'utilisation d'une librairie pour implémenter l'algorithme de Karger n'apportera aucun point.

Votre programme doit pouvoir prendre en paramètre un fichier comme ceux mis à disposition sur Moodle et doit afficher les arcs appartenant à la coupe minimale trouvée. Le format d'affichage de la coupe doit être de la forme suivante:

Taille de la coupe: 3
3 −− 4
5 −− 8
1 −− 9

où la première ligne donne la taille de la coupe trouvée et les lignes suivantes énumèrent les arcs appartenant à cette coupe.

Vous devez aussi rendre un rapport répondant aux différentes questions posées dans l'énoncé de cette série.