



## Algorithme randomisé

L'objectif de cette séance est d'étudier un algorithme randomisé de solution du problème Coupe-Min (MinCut). On dispose d'un graphe connecté  $G = (S, A)$ , avec  $|S| = n$  (ici  $|\cdot|$  est la cardinalité).

On desire d'enlever le nombre minimal d'arêtes de  $G$  pour obtenir un nouveau graph dans lequel au moins un paire de noeuds ne sont pas relié par un chemin. On appelle ceci *une coupe*.

Plus formellement : *une coupe* dans  $G$  est un ensemble des arêtes telles que leur effacement produit la coupure de  $G$  en 2 ou plus graphes séparés.

La *coupe-min* est la coupe de cardinalité minimale.

### Algorithme randomisé

algorithme *MinCutR*(graph  $G$ )

tant que ( $|A| > 2$ )

choisir une arête  $(x, y) \in A$  au hasard

$A = A \setminus (x, y)$

créer nouveau sommet  $z$

pour  $(u, x) \in A$

$A = A \setminus (u, x)$

$A = A \cup (u, z)$

fin

pour  $(u, y) \in A$

$A = A \setminus (u, y)$

$A = A \cup (u, z)$

fin

/\* les deux boucles "pour" servent à "coller" les sommets  $x$  et  $y$  en un seul sommet  $z$  \*/

fin

retourner  $|A|$

Voici un exemple d'application de *MinCutR*. Le coût de la coupe est 2 et la solution sont les arêtes 3 et 4.



**Question :** quelle est la probabilité que *MinCutR* retourne la bonne réponse, c.-à-d. quelle est la valeur de  $p$  pour laquelle *MinCutR* est  $p$ -correct ?

Pour répondre à cette question

- on suppose que la taille de la coupe minimale est  $k$  ;
- on considère un ensemble  $C$  d'arêtes qui donne une des coupes minimales ;
- finalement, **on estime la probabilité que *MinCutR* sort seulement les arêtes de  $C$ .**

**Exercice 1** (Analyse de *MinCutR*)

- 1) Quel est le nombre minimal d'arêtes du graphe  $G$  ?
- 2) Quelle est la probabilité de tomber sur une arête de  $C$  au premier tirage aléatoire ?
- 3) Quel est le nombre minimal d'arêtes du graphe  $G'$  après la première "reduction" ? Quelle est la probabilité de tomber sur une des arêtes de  $C$  au second tirage ?
- 4) Procéder par récurrence pour obtenir la probabilité de tomber sur une des arêtes de  $C$  au  $i$ -ème tirage.
- 5) Calculer la probabilité que *MinCutR* trouve la coupe minimale.

On peut répéter *MinCutR*  $t$  fois comme dans la procédure suivante :

```
algorithme RepeatMinCutR(graph  $G = (S, A)$ )
int  $cut = |S|$ 
pour ( $i = 1; i \leq t; i++$ )
 $C = MinCutR(\text{graph } G)$ 
si  $|C| < cut$  alors  $cut = |C|$ 
fin
retourner  $cut$ 
```

**Exercice 2** Comment faut-il choisir  $t$  pour obtenir la probabilité de trouver la coupe minimale  $\sim e^{-1}$  ? Quelle est la valeur de  $t$  qui permet d'obtenir  $p = 0.9$  pour  $n = 10$  ?