

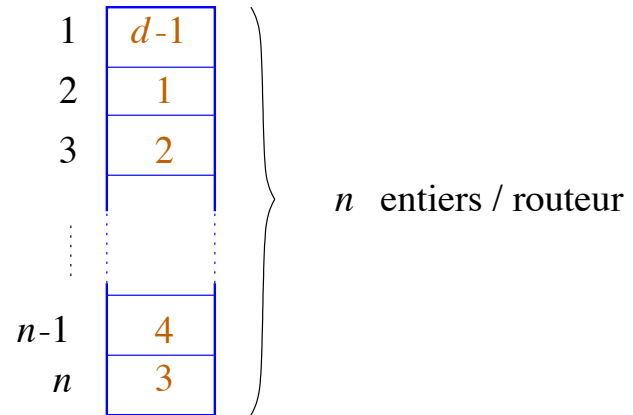
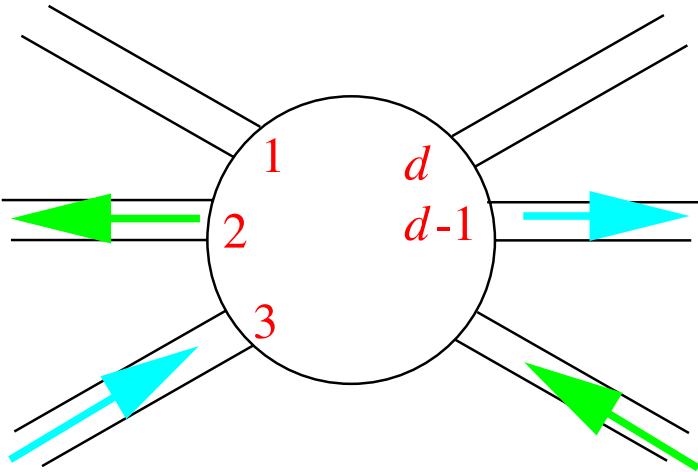
# Tables compactes de routage pour graphes de genre borné

Cyril Gavoille, Nicolas Hanusse - [LaBRI, Bordeaux](#)

## Plan de l'exposé

- Définitions
  - Réseaux communicants : problème du routage
  - Topologie : graphe simple connexe  $n$  sommets,  $m$  arêtes
  - Modèle du schéma de routage
- Routage dans les graphes planaires
- Plongement en  $k$ -page des graphes, partitions non croisées, ensembles dominants

## Tables de routage



message vers le sommet 3  
message vers le sommet 1

- $n \log d$  bits par routeur,
- contrôle des chemins (plus courts chemins),
- routage "direct": temps  $O(\log n)$  (linéaire).

**Résultats** : pour tout graphe de genre borné par  $\gamma \geq 0$ , nous construisons des tables compactes de routage de taille  $n \log(\gamma + 1) + O(n)$  bits (pour les graphes planaires :  $8n$  bits) avec temps poly-log pour l'extraction locale du plus courts chemin

⇒ la représentation "aide" au routage

## Différents modèles pour le routage

**Schéma de routage** = algorithme qui construit les fonctions de routage pour chaque graphe

- avec ou sans modification des entêtes,
- schéma de routage **universel** ou **dédié**,
- avec **facteur d'étirement borné** ( $s = 1$  pour les plus courts chemins),
- approche **centralisée** ou **distribuée**.

Nous considérons les schémas de routage universel sans modification d'entête et de plus courts chemins

## Une approche centralisée

- Chaque sommet contient la carte **complète** du réseau.
- Pour router de  $u$  à  $v$ , on applique un algorithme de plus courts chemin (Dijkstra/Bellman:  $O(m + n \log n)$ )
- **Avantages :**
  - faible complexité **mémoire** si le graphe peut être décrit succinctement (codage compact du graphe, Cayley, ...).
- **Inconvénients :**
  - **Temps** local d'extraction du plus court chemin :  $\Omega(n)$
  - **Complexité** exponentielle : temps/taille de l'entrée =  $\Omega(n)/O(\log n)$

## Approche distribuée

- Les sommets **ne possède pas** la description complète de la topologie

mais ...

**Théorème (Gavoille - Pérénnes 1996)** : Il existe des graphes où tout schéma de routage selon les plus courts chemins nécessite  $\Omega(n \log d)$  bits/routeur.

... alors que

$n \log d$  bits/routeur suffisent avec les **tables de routage**

## Graphes de genre $\gamma$

Le genre d'un graphe  $G$  : plus petit entier  $\gamma$  tel que  $G$  se plonge sur une surface de genre  $\gamma$  (isomorphe à une sphère avec  $\gamma$  trous) sans arête qui se croisent.

Graphes planaires :  $\gamma = 0$

- Aucune borne fine connue pour le routage dans les planaires :
  - Graphes de genre  $\gamma$  avec  $p$ -plans (Frederickson - Janardan '88) :  $O(d(p + \gamma) \log n)$  bits/routeur  $O(\log n)$  temps d'extraction de la route ... mais pire cas avec  $p = \Theta(n)$ , même pour  $\gamma = 0$ .
  - Graphes planaires (Frederickson - Janardan '89) :  $O(n^{1/3} \log n)$  bits/routeur en moyenne ... mais  $O(n \log n)$  bits sur certains sommets dans le pire cas. De plus  $s \neq 1$  ! ...

## Routage dans les planaires

- Codage compact des graphes planaires à  $n$  sommets
  - Théoriquement :  $\lim_{n \rightarrow +\infty} \mathcal{U}(n) = c^n$  (Denise, Vasconcellos, Welsh '96) avec  $3.24 < \log c < 6.25$  (Tutte '62)  
 $\Rightarrow$  codage avec  $6.25n$  bits mais non “efficace”.
  - En pratique : codage avec  $9n + o(n)$  bits (Chuang & al. '98) avec test d'adjacence en temps  $O(1)$
  - Nécessite un calcul de plus courts chemins :  $O(n)$  pour les planaires (Klein & al. '94)

$\Rightarrow$  Aucun schéma de routage de plus courts chemins connu pour les planaires qui utilise moins de  $n \log d$  bits de mémoire locale ( $n \log n$  dans le pire cas) avec un temps d'extraction poly-logarithmique.



## Table compacte de routage pour les planaires à $n$ sommets

Table de routage taille en bits	temps de routage	
$n \log n$	$O(\log n)$	table de routage standard (en bit-opération)
$6.25n$	$2^{cn} ?$	“Information-theoretic” upper bound (Denise & al. '96, Tutte '62)
$9n + o(n)$	$O(n)$	codage compact des graphes planaires non étiquetés avec temps $O(1)$ pour l'adjacence (Chuang & al. '98) et algo. de plus courts chemins optimal (Klein & al. '94)
$8n + o(n)$	$O(\log^{2+\epsilon} n)$	arbres en $k$ -pages + ensembles dominants (en bit-operations) [ <b>cet exposé</b> ]

Table compacte de routage avec  
plongement en  $k$ -pages et ensemble dominants

Classes de graphes	Mémoire requise (en bits)	Temps
graphes planaires	$8n + o(n)$	$O(\log^{2+\epsilon} n)$
graphes planaires et sommets de degré $d > \frac{n}{8}$	$9n - 8d + o(n)$	$O(\log^{2+\epsilon} n)$
graphes de genre $\gamma$	$n \log(\gamma + 1) + O(n)$	$O(\log^{2+\epsilon} n)$
graphes en $k$ -pages	$2n \log k + 4n + o(n)$	$O(\log^{2+\epsilon} n)$

## Plongement en $k$ -pages et nombre de pages

Plongement en  $k$ -page :

- ordre linéaire des sommets (dessinés sur une ligne ou un cercle)
- partition des arêtes en  $k$  pages telle que deux arêtes d'une même page ne se croissent pas.

Nombre de page d'un graphe : plus petit entier  $k$  tel que  $G$  possède un plongement en  $k$  pages.

Théorème (Yannakakis '86) : si  $G$  est planaire, son nombre de pages est au plus 4.

Exemple:

nombre de page de  $G = 1$  si et seulement si  $G$  est outer-planaire

nombre de page de  $K_n = \lceil n/2 \rceil$

## Codage d'un $k$ -pages

Coder un graphe de nombre de page  $k$  :  $9kn$  bits suffisent (Jacobson '89). Optimal ?

**Théorème 1** : Le nombre de graphes connexes à  $n$  sommets non étiquetés de nombre de page  $\mathcal{U}_{n,k}$ ,  $1 \leq k \leq n/2$ , vérifie

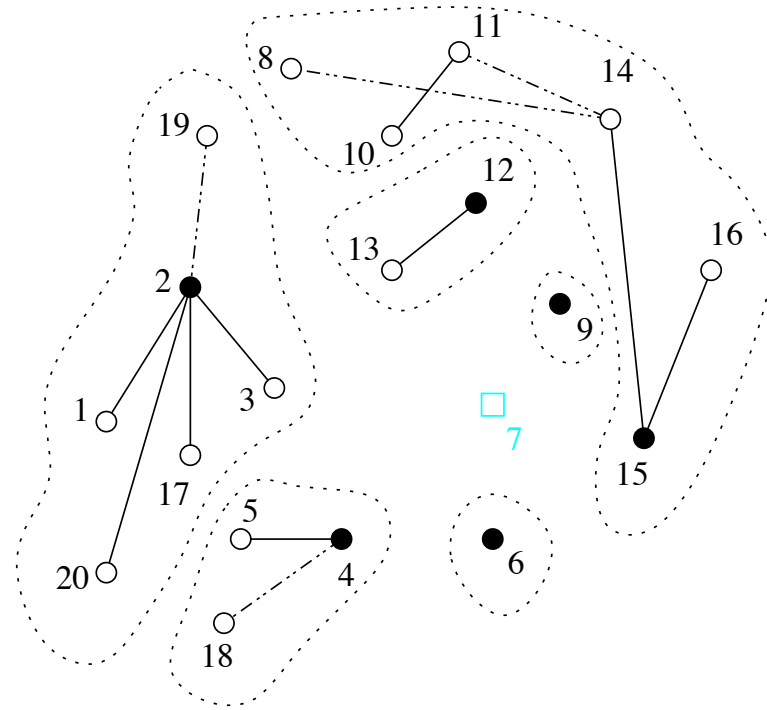
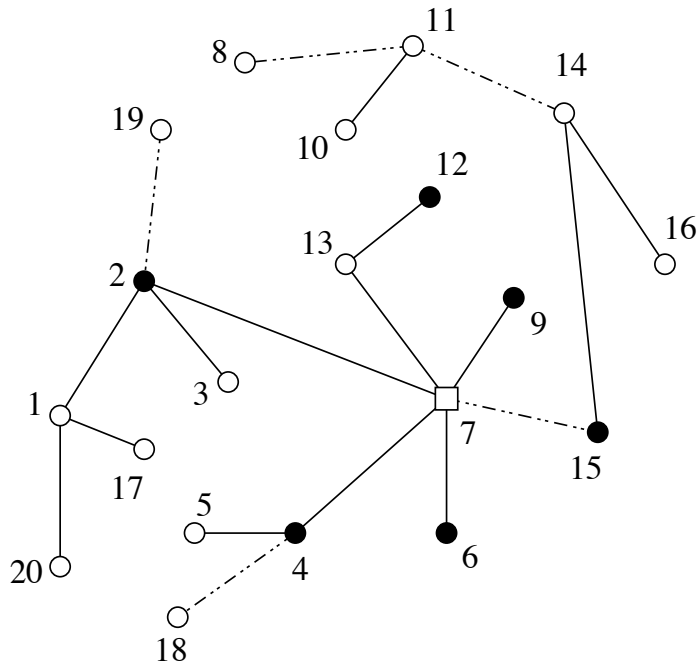
$$\mathcal{U}_{n,k} \geq \frac{\left(2^k (2^k - 1)\right)^{n/2-k}}{2k!} \sim 2^{kn-O(k \log k)} .$$

$\Rightarrow \Omega(kn)$  bits sont nécessaire pour coder un graphe en  $k$ -pages.

**Théorème 2** : Tout graphe non étiquetés en  $k$ -pages et  $m$  arêtes peut être représenté avec  $2m \log k + 4m$  bits. De plus, avec un supplément de  $o(m)$  bits, il est possible de tester l'adjacence en  $O(1)$  opérations entières.

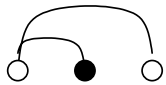
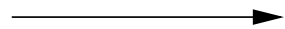
## Schéma de routage dans un $k$ -page

- Le sommet  $u$  contient un arbre recouvrant de poids minimum  $T$ , enraciné en  $u$
- Pour aller de  $u$  à  $v$  le routage consiste à sélectionner l'unique ancêtre de  $v$  dans  $T$  qui est la racine ( $\bullet$ ) d'une "région"
- On stocke  $R_T$ , le "graphe-région", à la place de  $T$ .



tree  $T$

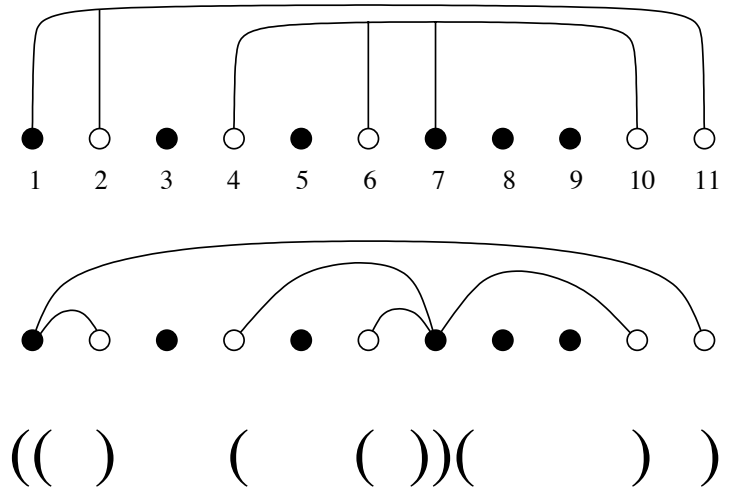
region graph  $R_T$



## Remarque sur le “graphe-région”

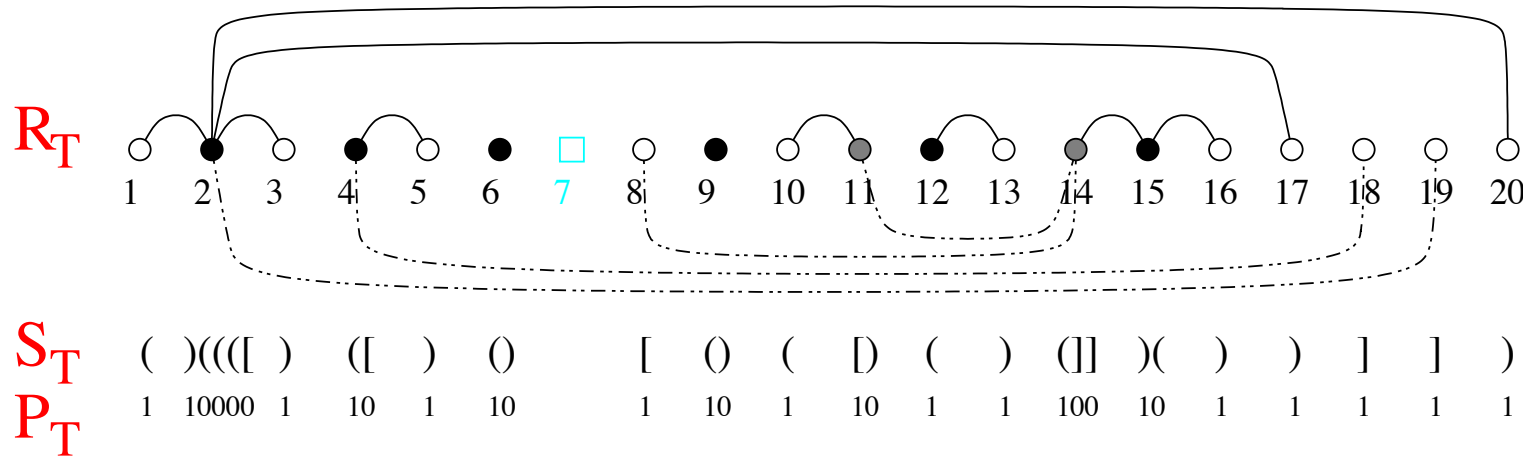
Classes de graphes	Nombre	Codage (en bits)
arbres non étiquetés enracinés	$\frac{1}{n+1} \binom{2n}{n}$ (Catalan)	$2n - O(\log n)$
arbres étiquetés sans racine	$n^{n-2}$ (Cayley)	$n \log n - O(\log n)$
partition non croisées	$\frac{1}{n} \binom{n}{m} \binom{n}{m-1}$ (Narayana)	$2n - O(\log n)$
arbres sur $k$ -page	??	$4n + 2n \log k + o(n)$

- Une partition  $P_1 \cup \dots \cup P_m$  sur un ensemble ordonné  $\mathcal{P}, n = |\mathcal{P}|$ , est *non croisée* si pour tout  $a, b, c, d \in \mathcal{P}$  tels que  $a < b < c < d$ , si  $a, c \in P_i$  et  $b, d \in P_j$ , alors  $i = j$ .
- $R_T$  est l'union, sur toutes les pages, de la représentation de  $k$  partitions non croisées.



## Codage et cheminement dans le “graphe-région” $R_T$

- Une chaîne  $S_T$  de parenthèses typées ( $2k$  types) pour les arêtes
- Un vecteur binaire  $P_T$  indiquant la position de chaque sommet dans  $S_T$  :  $|P_T| = |S_T| = 2(n - 1 - d + i)$



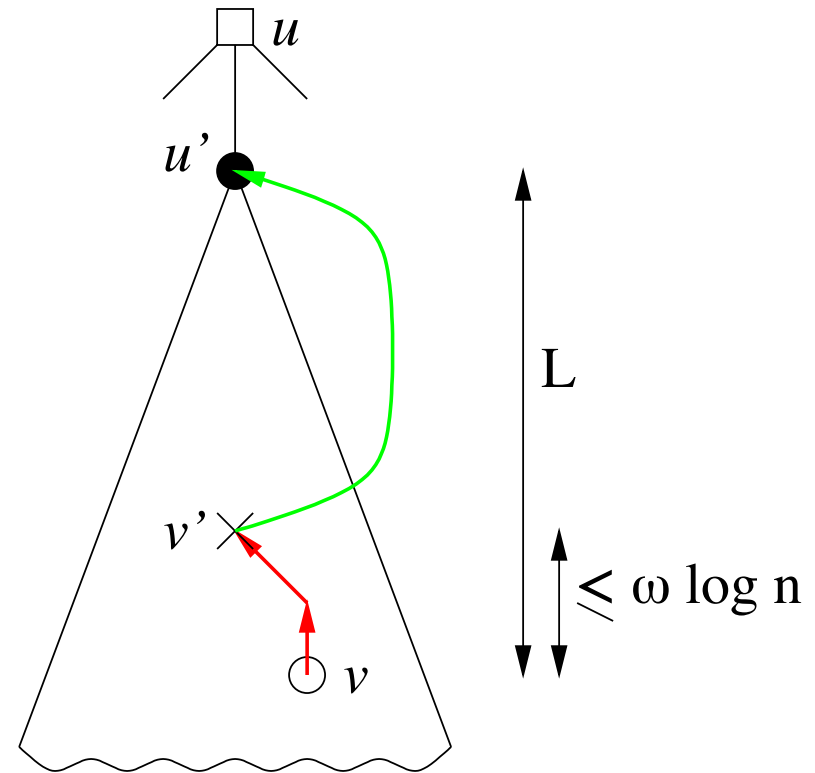
**Lemme** : L’algorithme Route1 s’exécute en  $O(L\omega \log n + \omega^2 \log^2 n)$  bit-opérations avec des tables de routage de taille  $2n \log k + 4n + o(n)$  bits avec  $\omega$  tel que  $n/\omega = o(n)$  et  $L \approx dist(u, v)$

**Problème** : si  $L > \omega \log n$



## Vers un routage en $O(\omega \log n)$ étapes : ensembles dominants

- $\forall v \in R_T, D_1 = \{v' \mid d(v, v') \leq \omega \log n\}$ ,  $D_1$  est un ensemble  $\omega \log n$ -dominant,
- $D_1$  peut être construit en temps linéaire (Upfal - Peleg '89) et  $|D_1| \leq \frac{n}{\omega \log n}$ ,
- la liste  $D_1$  codable sur  $|D_1| \log n = n/\omega = o(n)$  bits.



**Théorème 3** : On peut router selon les plus courts chemins dans un graphe  $k$ -page de  $n$  sommets en  $O(\omega^2 \log^2 n)$  bit-opérations avec des tables de taille  $2n \log k + 4n + o(n)$  bits.

## Applications

Théorème (Malitz '88) :  $k = O(\sqrt{\gamma})$ .

Choissant  $\omega = O(\log^{\epsilon/2} n)$  :

**Corollaire** : Tout graphe pondéré de  $n$  sommets et de genre  $\gamma \geq 0$  possède des tables de routage selon des chemins de coûts minimaux de taille  $n \log(\gamma + 1) + O(n)$  bits/sommet, et avec une extraction de la route en  $O(\log^{2+\epsilon} n)$  bit-opérations/sommet, constante  $\epsilon > 0$ .

**Corollaire** : Tout graphe planaire pondéré de  $n$  sommets possède des tables de routage selon des chemins de coûts minimaux de taille  $8n + o(n)$  bits/sommet, et avec une extraction de la route en  $O(\log^{2+\epsilon} n)$  bit-opérations/sommet.

## Conclusion

On peut montrer que, pour une famille donnée d'arbres recouvrants de poids minimum,  $\Omega(n \log k)$  bits sont requis pour router selon les chemins définis par les arbres de la famille (la borne du Théorème 3 est fine).

### Problèmes ouverts :

- Est-ce que  $\Omega(n)$  bits sont requis pour router selon **des** plus courts chemins dans un graphes planaire ou outer-planaire ( $k = 1$ ) ? La meilleure borne inférieure est  $\Omega(\sqrt{n})$  bits pour les arbres (Eilam, Gavioille, Peleg '98).
- Si le facteur d'étirement est  $s > 1$  ?