Connexité - cycles – cocycles – arbres – coarbres – arborescences

Définitions

- Deux sommets font partie d'une même **composante connexe** d'un graphe G si et seulement si il existe une chaîne dans G qui les relie.
- Un graphe est **connexe** s'il ne comporte qu'une composante connexe.
- Un **arbre** est un graphe connexe sans cycle simple.

Soit G=(V,E) un graphe connexe et $E' \subseteq E$.

- Le sous-graphe partiel G'=(V,E') est un **arbre de G** si E' ne contient aucun cycle de G alors que $E' \cup \{e\}$ en contient un pour tout $e \in E-E'$. On note C_e le cycle obtenu en rajoutant e à E'.
- Le sous-graphe partiel G'=(V,E') est un **coarbre de G** si E' ne contient aucun cocycle de G alors que $E' \cup \{e\}$ en contient un pour tout $e \in E-E'$. On note W_e le cocycle obtenu en rajoutant e à E'.

Propriété : G'=(V,E') est un arbre de G $\Leftrightarrow$ G''=(V,E-E') est un coarbre de G
(voir démonstration à la fin de ce document)

Un peu d'algèbre

- On peut représenter chaque cycle C par un vecteur **c** à $|E|$ composantes booléennes :

  c_i=1 si la i^e arête du graphe fait partie du cycle; c_i=0 sinon.

- On peut représenter chaque cocycle [W,V-W] par un vecteur **w** à $|E|$ composantes booléennes :

  w_i=1 si la i^e arête du graphe fait partie du cocycle; w_i=0 sinon.

- L'espace des vecteurs cycles muni de l'addition modulo 2 est orthogonal à l'espace des vecteurs cocycles muni de l'addition modulo 2 :

  c•w=0 pour tout vecteur cycle **c**, et pour tout vecteur cocycle **w**.

- Le nombre de vecteurs cycles dans une base de cycles est appelé **le nombre cyclomatique** et vaut $|E|-|V|+p$, où p est le nombre de composantes connexes du graphe considéré.
- Le nombre de vecteurs cocycles dans une base de cocycles est appelé **le nombre cocyclomatique** et vaut $|V|-p$, où p est le nombre de composantes connexes du graphe considéré.

- Pour obtenir **une base de cycles** dans un graphe connexe G=(V,E), il suffit de construire un arbre formé d'un sous-ensemble d'arêtes $E' \subseteq E$, et de mettre dans la base chaque cycle C_e créé par l'ajout d'une arête $e \in E-E'$ dans l'arbre.

- Pour obtenir **une base de cocycles** dans un graphe connexe G=(V,E), il suffit de construire un arbre formé d'un sous-ensemble d'arêtes $E' \subseteq E$, et de mettre dans la base chaque cocycle W_e créé par le retrait d'une arête $e \in E'$ de l'arbre.

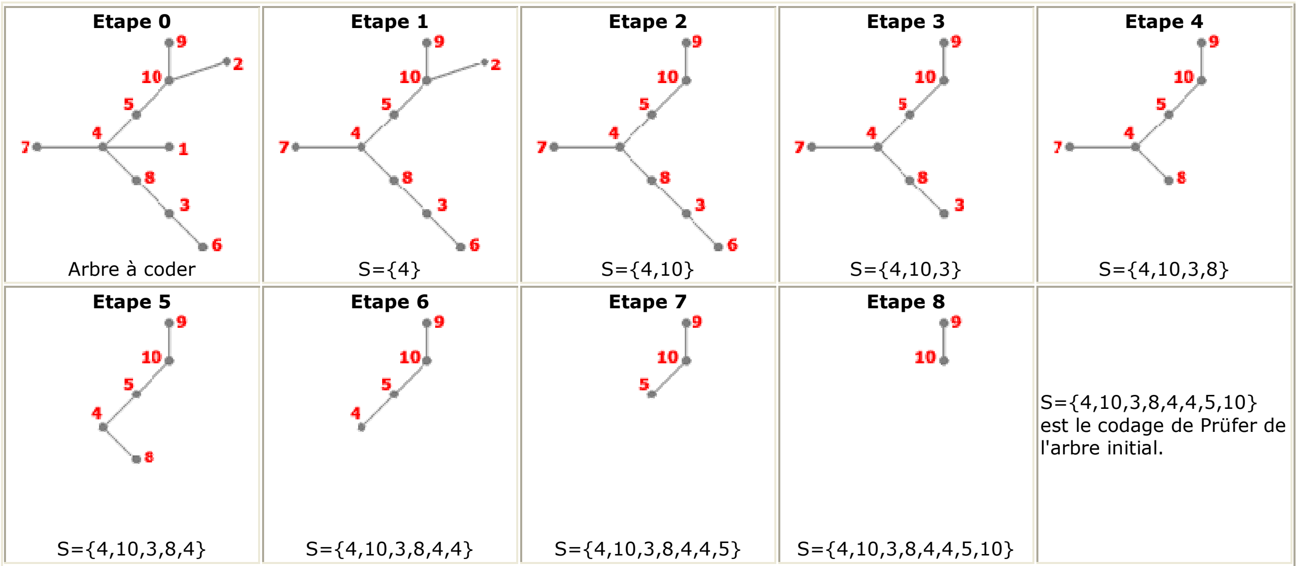
Codage

Codage d'un arbre (Prüfer)

Soit l'arbre T = (V, E) et supposons V = {1, 2, ..., n}. L'algorithme ci-dessous fournira le code de T, c'est-à-dire une suite S de n-2 termes employant (éventuellement plusieurs fois) des nombres choisis parmi {1, ..., n}.

Répéter ce qui suit tant qu'il reste plus de deux sommets dans l'arbre courant T :
- identifier le sommet pendant v de l'arbre courant ayant le numéro minimum ;
- ajouter à la suite S le seul sommet s adjacent à v dans l'arbre T courant ;
- enlever de l'arbre T courant le sommet v et l'arête incidente à v.

Exemple de codage

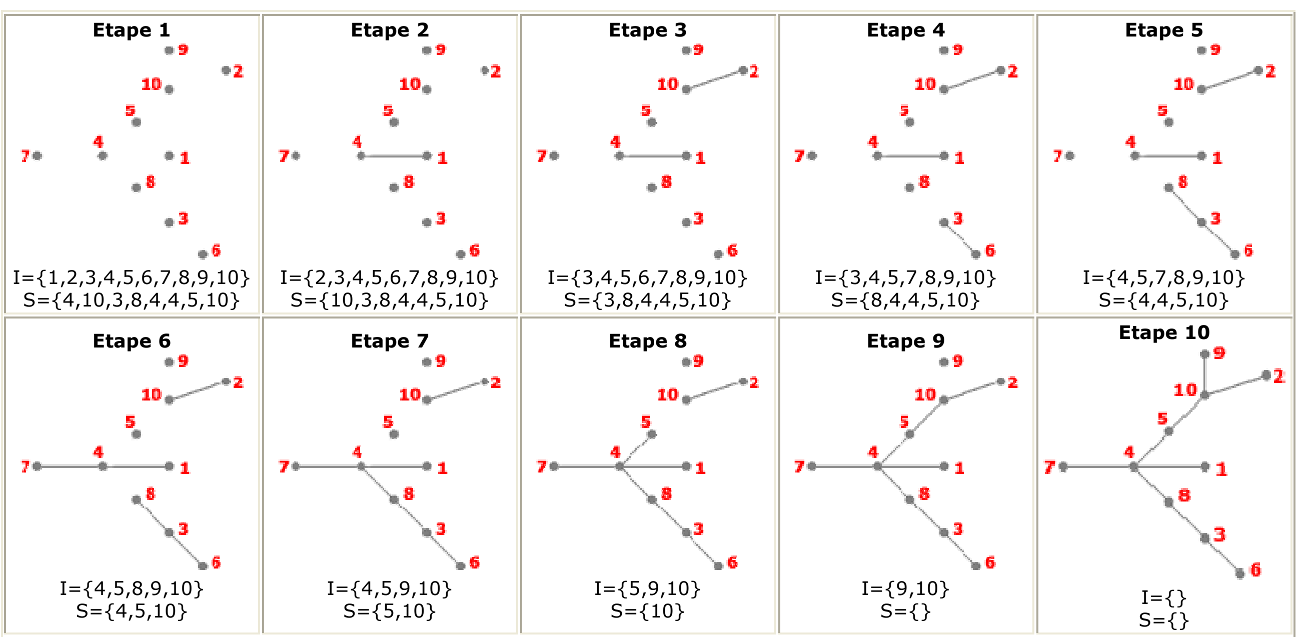


Décodage d'un arbre

Donnée: une suite S de n-2 nombres pris dans {1, ..., n}.

> Poser I = {1, ..., n};
> Répéter ce qui suit tant qu'il reste des éléments dans S et plus de deux éléments dans I :
> - identifier le plus petit élément i de I n'apparaissant pas dans la suite S ;
> - relier par une arête de T le sommet i avec le sommet s correspondant au premier élément de la suite S ;
> - enlever i de I et s de S.
>
> Les 2 éléments qui restent dans I à la fin de l'algorithme constituent les extrémités de la dernière arête à ajouter à T.

Exemple de décodage



On peut vérifier que les deux algorithmes constituent les deux sens d'une bijection entre les arbres sur n sommets numérotés et les mots de n-2 lettres sur l'alphabet à n lettres. On peut donc énoncer le théorème suivant.

Théorème de Cayley (1857)
Le nombre d'arbres que l'on peut construire sur n (n>1) sommets numérotés est égal à n^{n-2}.

Arbre de poids minimum

Supposons qu'une distance d_e est associée à chaque arête e du graphe G=(V,E). On désire déterminer un sous-graphe partiel de G qui soit un arbre de distance totale minimum (aussi appelé arbre de poids minimum).

Propriété

Soit G'=(V,E') un sous-graphe partiel de G qui est un arbre. Les trois propriétés suivantes sont équivalentes :

(a) G' est un arbre de poids minimum

(b) $d_e = \max_{e' \in C_e} d_{e'} \quad \forall e \notin E'$

(c) $d_e = \min_{e' \in W_e} d_{e'} \quad \forall e \in E'$

<u>Algorithme de *Kruskal* de construction d'un arbre de poids minimum (basé sur (b))</u>

(1) Classer les arêtes par distance non décroissante

(2) Parcourir la liste des arêtes et rajouter une arête à l'arbre si celle-ci ne crée pas de cycle avec les arêtes déjà rajoutées

Complexité : $O(|V||E|)$

<u>Algorithme de *Prim* de construction d'un arbre de poids minimum (basé sur (c))</u>

(1) Choisir un sommet quelconque de G et le mettre dans A

(2) Pour i allant de 1 à $|V|$-1 faire

Déterminer une arête de coût minimum dans le cocycle [A,V-A]

Rajouter cette arête dans l'arbre et mettre dans A l'extrémité qui n'y était pas.

Complexité : $O(|V|^2)$

Arborescence de poids minimum

Une **arborescence de racine r** dans un graphe orienté G=(V,A) est un sous-graphe partiel connexe G'=(V,A') tel que chaque sommet de V autre que la racine à un unique arc entrant.

Problème : déterminer une arborescence de racine r de poids minimum (c-à-d de distance totale minimum), où d_a représente la distance d'un arc a de G.

<u>*Préliminaire*</u>. Soit C un circuit dans un graphe G. On note G/C le multi-graphe obtenu en remplaçant les sommets de C par un seul sommet c, et en remplaçant tout arc entrant en C ou sortant de C par des arcs entrant ou sortant de c.

<u>Algorithme de construction d'une arborescence de racine r de poids minimum.</u>

(1) t :=0; G^0 :=G; $d^0(a)$:=d_a pour tout *a* dans A

(2) Soit A^t le sous-graphe partiel de G^t obtenu en choisissant, pour chaque sommet autre que la racine, l'arc entrant de distance minimum (par rapport à d^t)

Si A^t n'a pas de circuit, aller à (3) sinon aller à (4).

(3) Décontracter A^t pour déterminer l'arborescence optimale dans G : STOP

(4) Soit C un circuit dans A^t. Poser G^{t+1} :=G^t/C et poser $d^{t+1}(a)=d^t(a)$ pour tout arc *a* n'entrant pas dans C, et $d^{t+1}(a)=d^t(a)-d^t(a')$ pour tout arc *a* entrant dans C, où *a'* est l'arc de C ayant la même extrémité terminale que *a*.

Poser t :=t+1 et retourner à (2)

Dans le cas où la racine n'est pas fixée d'avance on peut déterminer une arborescence de poids minimum de la façon suivante

(1) Rajouter un sommet r ainsi qu'un arc de r vers chacun des sommets de G, le coût de chaque arc (r,x) étant égal à D+1, où D est la somme totale des distances sur les arcs de G.

(2) Déterminer une arborescence de racine r de poids minimum à l'aide de l'algorithme ci-dessus.

(3) Ôter r de l'arborescence.

Théorème de Minty (lemme des arcs colorés)

Soit G=(V,A) un graphe orienté et *a* un arc dans A. Considérons n'importe quelle coloration des arcs de A tel que *a* est NOIR et les autres arcs sont BLEUS, ROUGES ou NOIRS.

Alors on se trouve forcément dans exactement l'un des deux cas suivants

1) il existe un cycle passant par *a*, coloré uniquement en ROUGE et NOIR, et dont tous les arcs NOIRS sont orientés dans le même sens que *a*.

2) il existe un cocycle contenant *a*, coloré uniquement en BLEU et NOIR, et dont tous les arcs NOIRS traversent la frontière dans le même sens que *a*.

Corollaire

Dans un graphe orienté, chaque arc fait partie soit d'un circuit, soit d'un cocircuit (ou exclusif)

Démonstration que G'=(V,E') est un arbre de G=(V,E) si et seulement si G''=(V,E-E') est un coarbre de G

Tout d'abord, orientons de façon quelconque les arêtes de G.

(1) Soit G'=(V,E') un arbre de G et soit E''=(E-E').

- Soit *a* un arc quelconque de E'. Colorons *a* en NOIR, tous les arcs de E' sauf *a* en ROUGE et tous les arcs de E'' en BLEU. Comme E' ne contient pas de cycle, il n'existe pas de cycle ROUGE et NOIR. Il existe donc un cocycle NOIR et BLEU, ce qui montre que E''$\cup\{a\}$ contient un cocycle.
- Soit *b* un arc quelconque de E''. Colorons *b* en NOIR, tous les arcs de E' en ROUGE et tous les arcs de E'' sauf *b* en BLEU. Comme E'$\cup\{b\}$ contient un cycle passant par *b*, il existe un cycle ROUGE et NOIR avec un unique arc NOIR. Il n'existe donc pas de cocycle NOIR et BLEU, ce qui montre que E''ne contient pas de cocycle.

On en déduit donc que G''=(V,E-E') est un coarbre de G

(2) Soit G''=(V,E'') un coarbre de G et soit E'=(E-E'').

- Soit *a* un arc quelconque de E''. Colorons *a* en NOIR, tous les arcs de E'' sauf *a* en BLEU et tous les arcs de E' en ROUGE. Comme E'' n'a pas de cocycle, il n'existe pas de cocycle BLEU et NOIR. Il existe donc un cycle NOIR et ROUGE, ce qui montre que E'$\cup\{a\}$ contient un cycle.
- Soit *b* un arc quelconque de E'. Colorons *b* en NOIR, tous les arcs de E'' en BLEU et tous les arcs de E' sauf *b* en ROUGE. Comme E''$\cup\{b\}$ contient un cocycle passant par *b*, il existe un cocycle BLEU et NOIR avec un unique arc NOIR. Il n'existe donc pas de cycle NOIR et ROUGE, ce qui montre que E' ne contient pas de cycle.

On en déduit donc que G'=(V,E-E'') est un arbre de G