Vers un routage Compact Distribué

Algotel '12

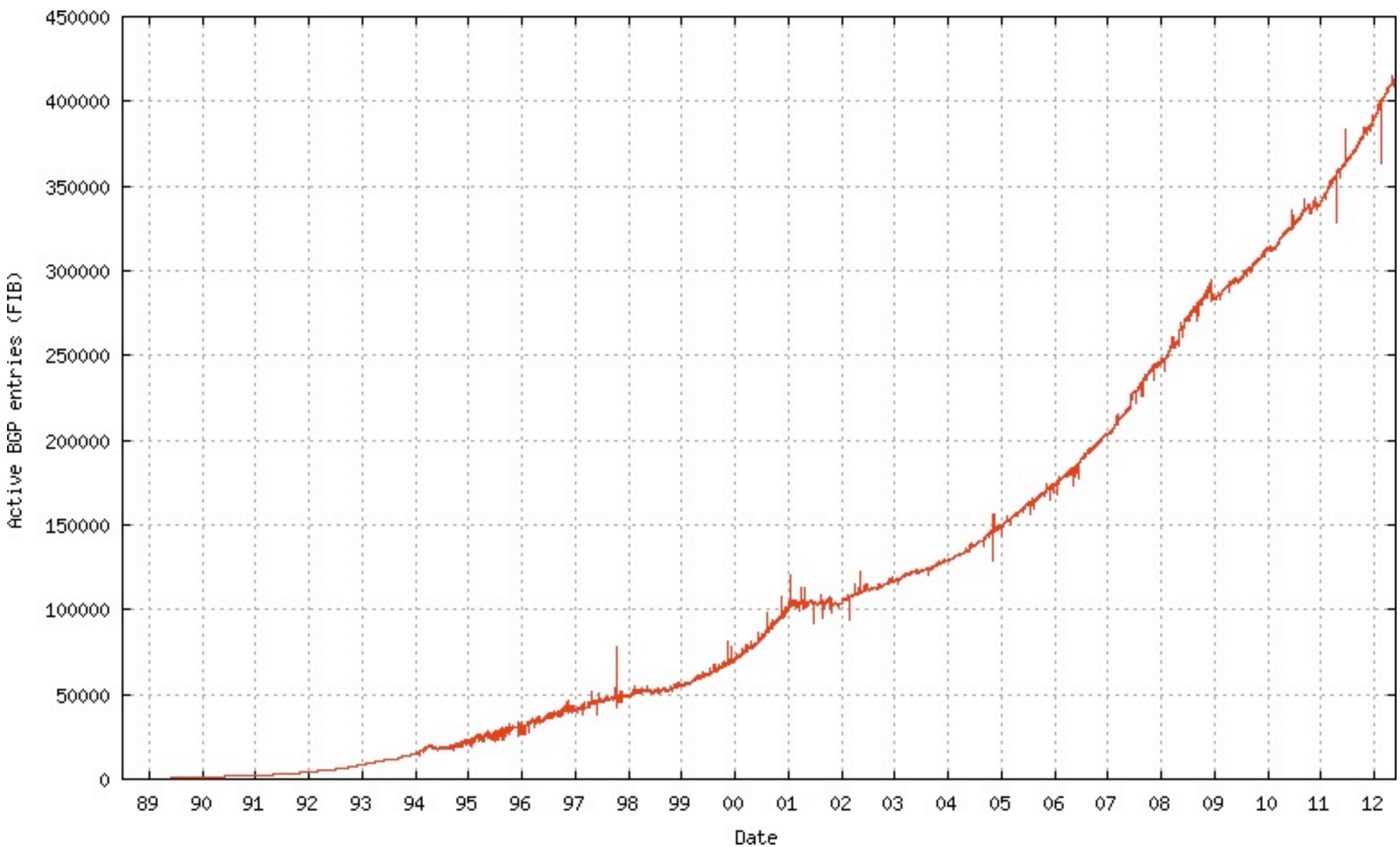
Christian Glacet , Lucas Verdonk

INRIA - Supporté par le projet EULER

Routage inter-domaine BGP

- Routage de plus court chemin
- Connaissance de tout le réseau (une entrée par destination nécessaire)

Routage inter-domaine BGP



Routage inter-domaine BGP

Problème : augmentation des délais de transmission lors du routage

Une solution : routage compact

- Détours bornés
- Tables plus petites (sous-linéaire)

Alternatives existantes

- Centralisées
- Statiques

ne sont pas utilisable en pratique

Alternatives existantes

- Centralisées
- Statiques

ne sont pas utilisable en pratique

- exemples d'algorithmes universels compacts :

  [TZ] Thorup & Zwick - 2001

  [AGMNT] Abraham, Gavoille, Malkhi, Nisan & Thorup - 2008

Objectif

- Proposer un algorithme de construction des tables de routage :
  - distribué
  - asynchrone
  - supportant la dynamique du réseau
  - sans renommage des sommets (noms arbitraires), ie AGMNT.

Objectif

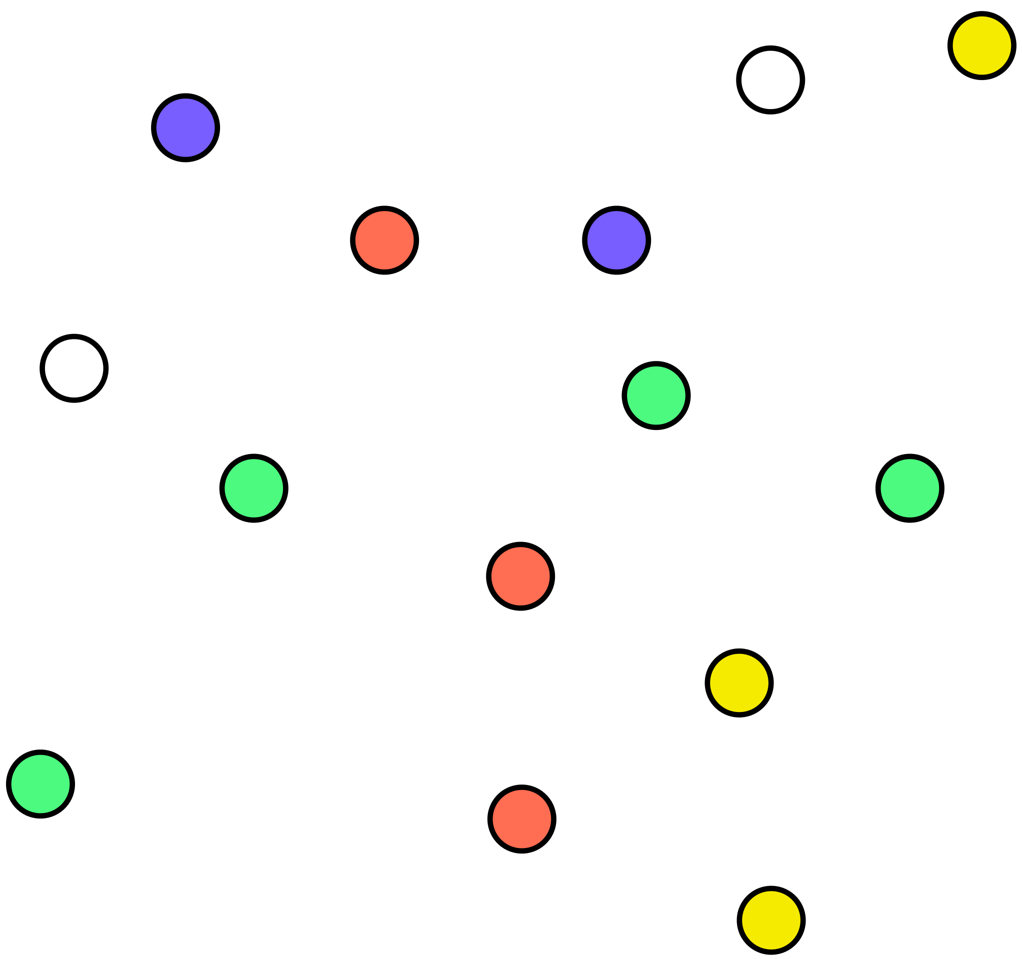
- Proposer un algorithme de construction des tables de routage :
  - **distribué**
  - asynchrone
  - supportant la dynamique du réseau
  - sans renommage des sommets (noms arbitraires), ie AGMNT.

Le principe du routage AGMNT

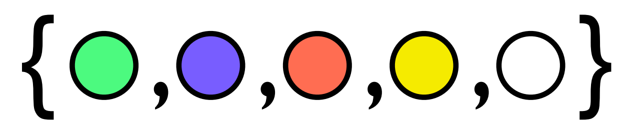
Notations utilisées

- Graphe G=(V,E)
- $|V| = n$, $|E| = m$ et D le diamètre de G
- $u \to v$: <u>route</u> de u vers v
- d(u,v) : distance entre u et v
- $len(u \to v)$: longueur de la route $u \to v$.
- L'étirement est le ratio : $len(u \to v)/d(u,v)$

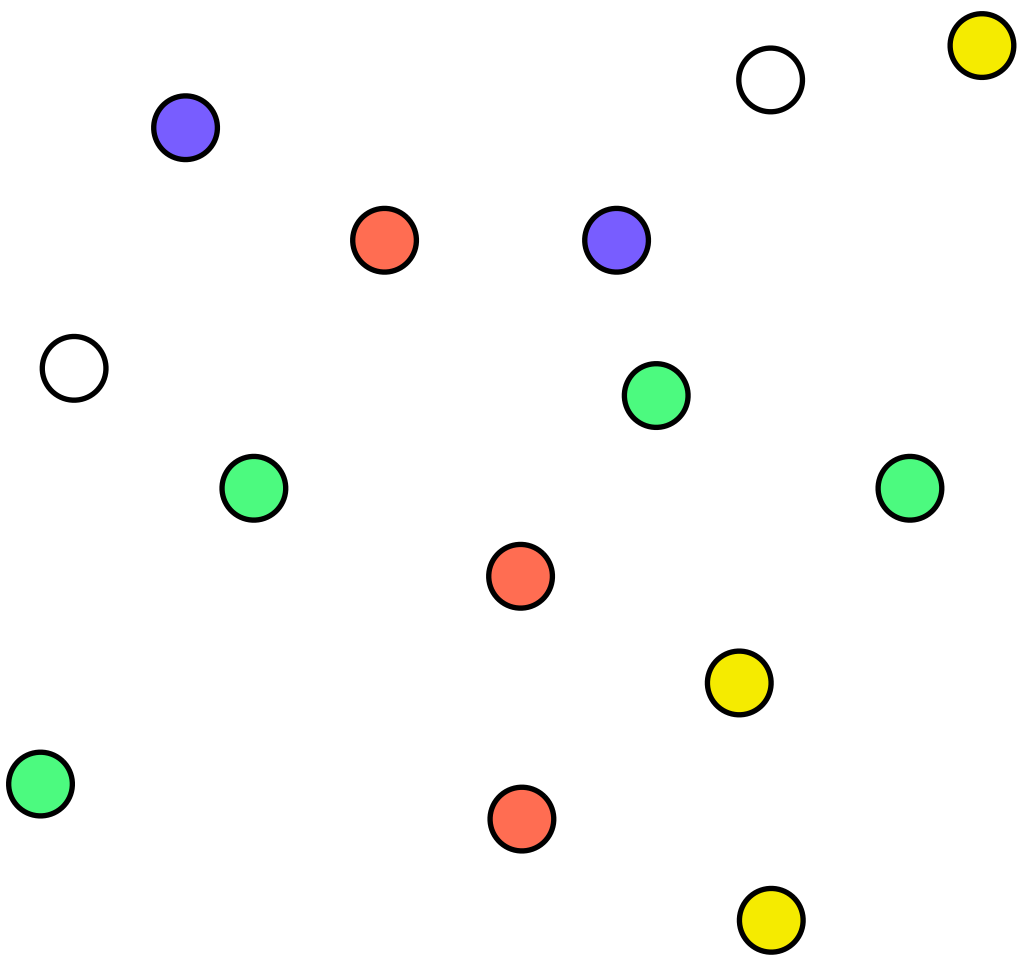
Couleurs



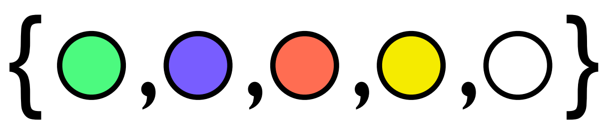
Tout sommet u a une couleur aléatoire $c(u) \in [1,\ldots, k]$:



Couleurs

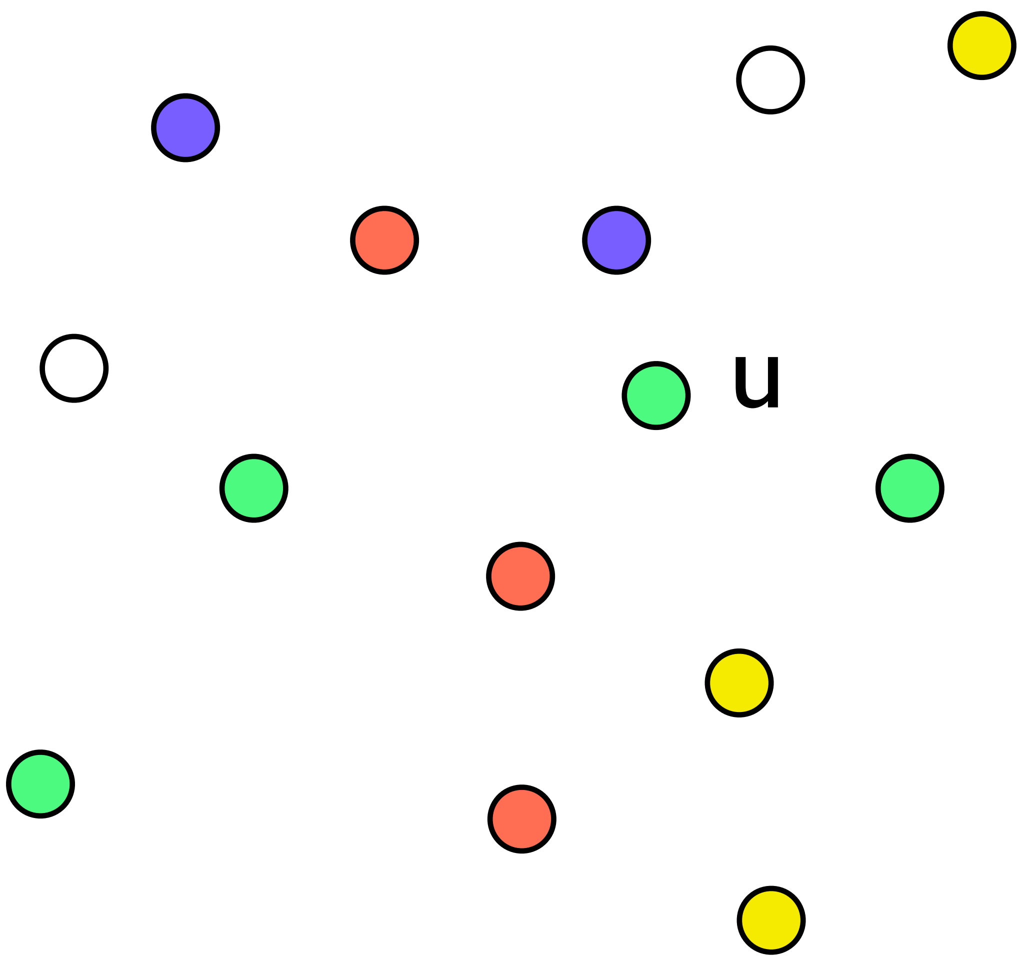


Tout sommet u a une couleur aléatoire $c(u) \in [1,\ldots, k]$:

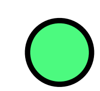


k est un paramètre d'AGMNT, par exemple $k = \sqrt{n}$

Couleurs

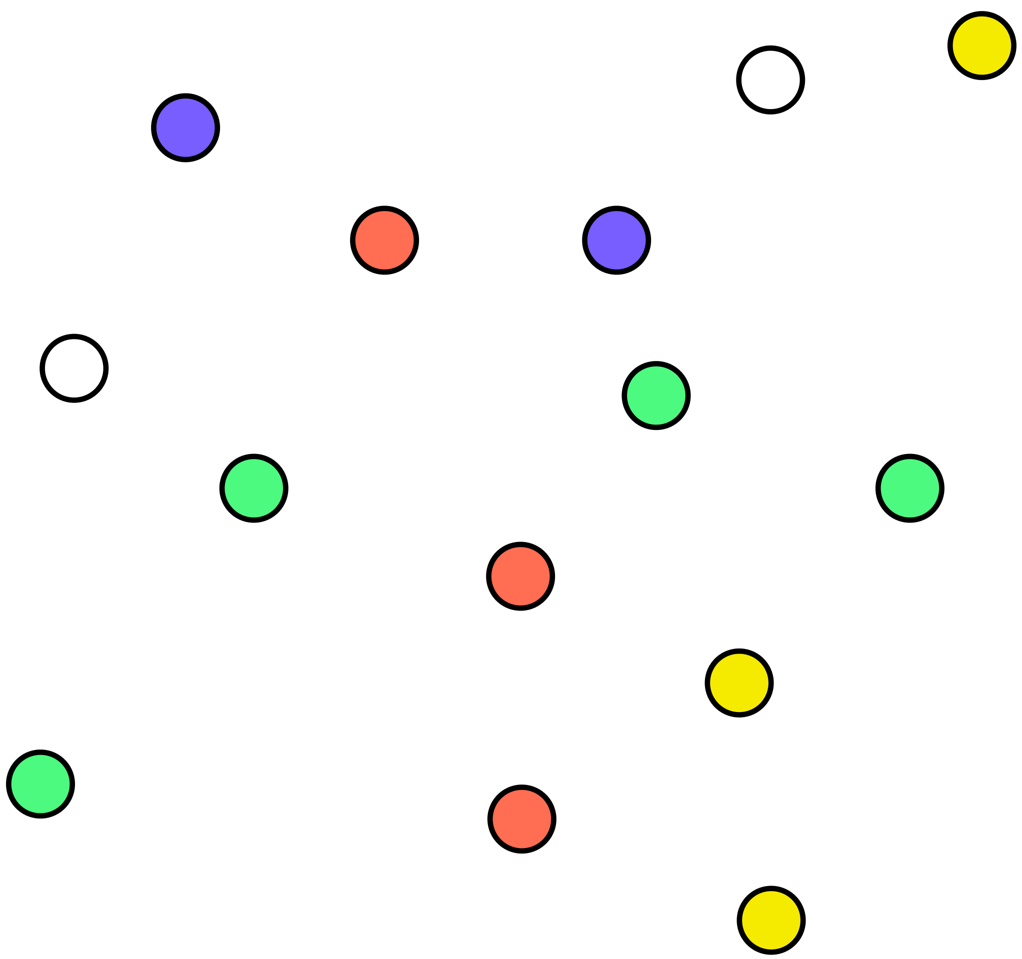


C(u) - Ensemble des sommets ayant la même couleur que u :



$$| C(u) | = O(n/k) = O(\sqrt{n})$$

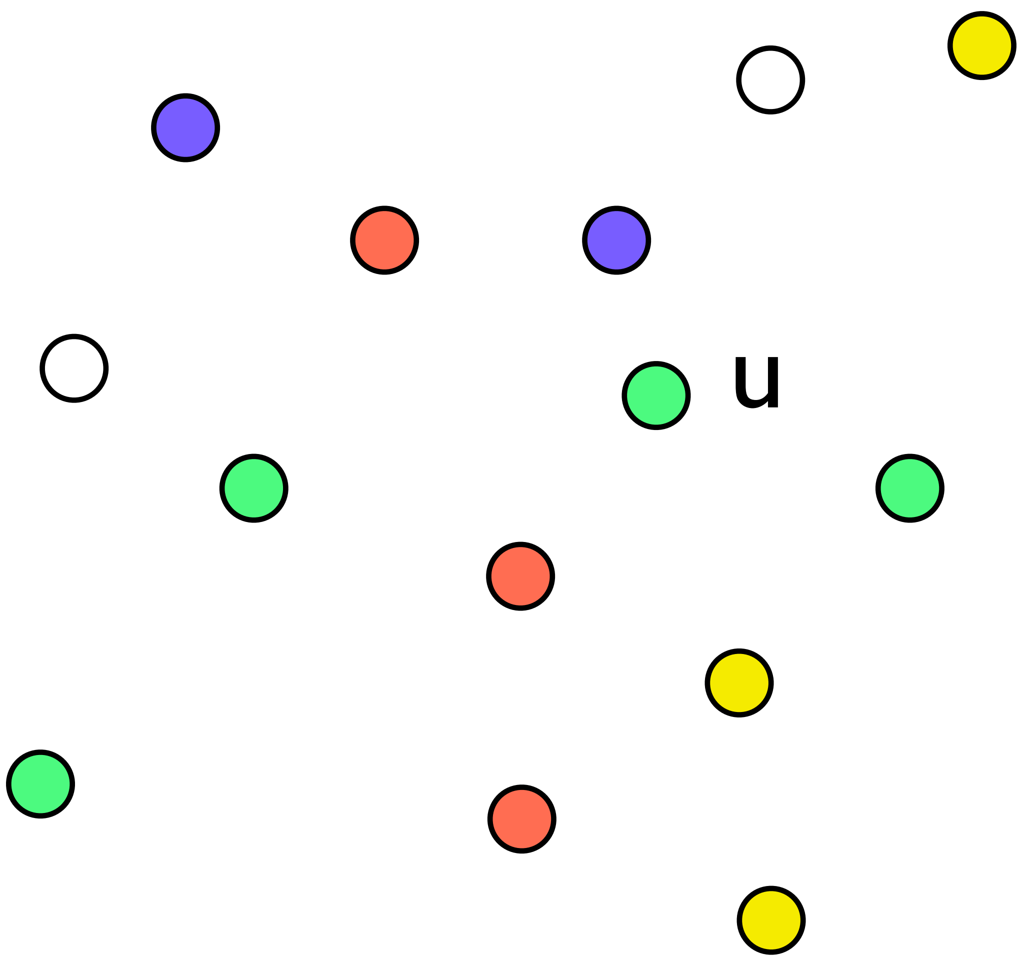
Landmarks



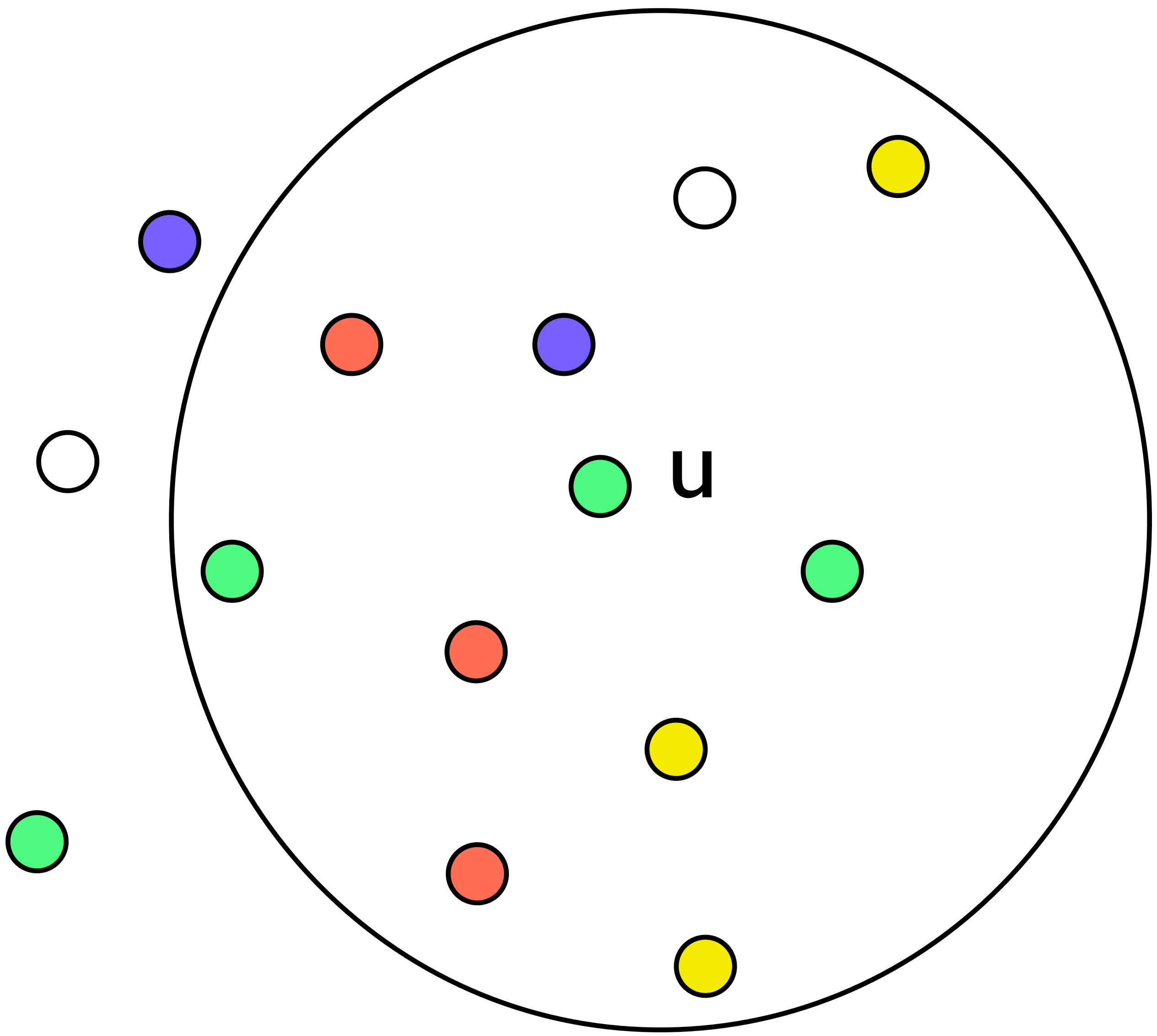
L - un ensemble de sommet d'une couleur donnée (ici blanc), appelés Landmarks :

$O(n/k) = O(\sqrt{n})$ Landmarks

Boules de voisinage



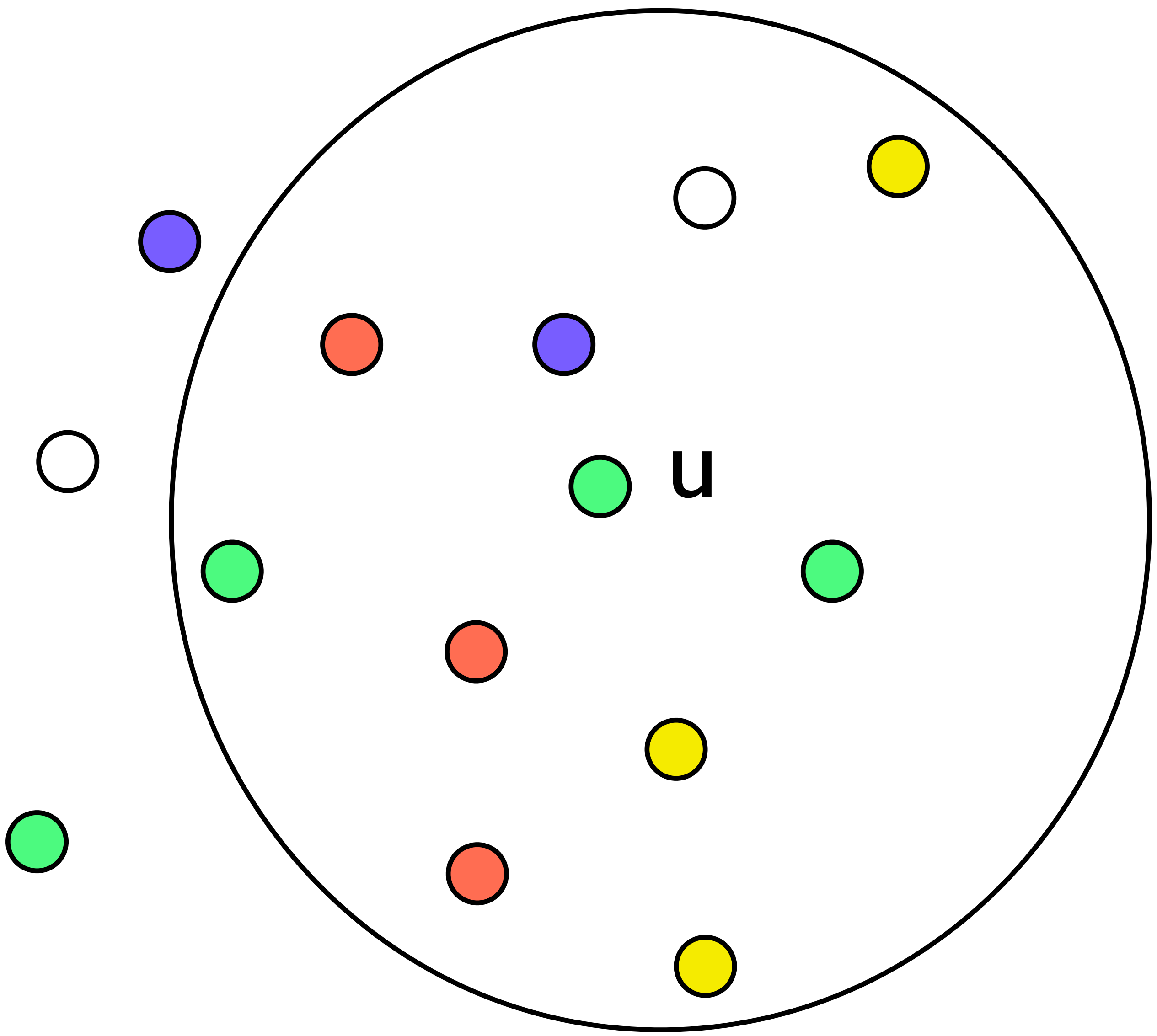
Boules de voisinage



B(u) - Les plus proches sommet de u respectant les propriétés :

- de **complétude**, la boule contient un représentant de chaque couleur,

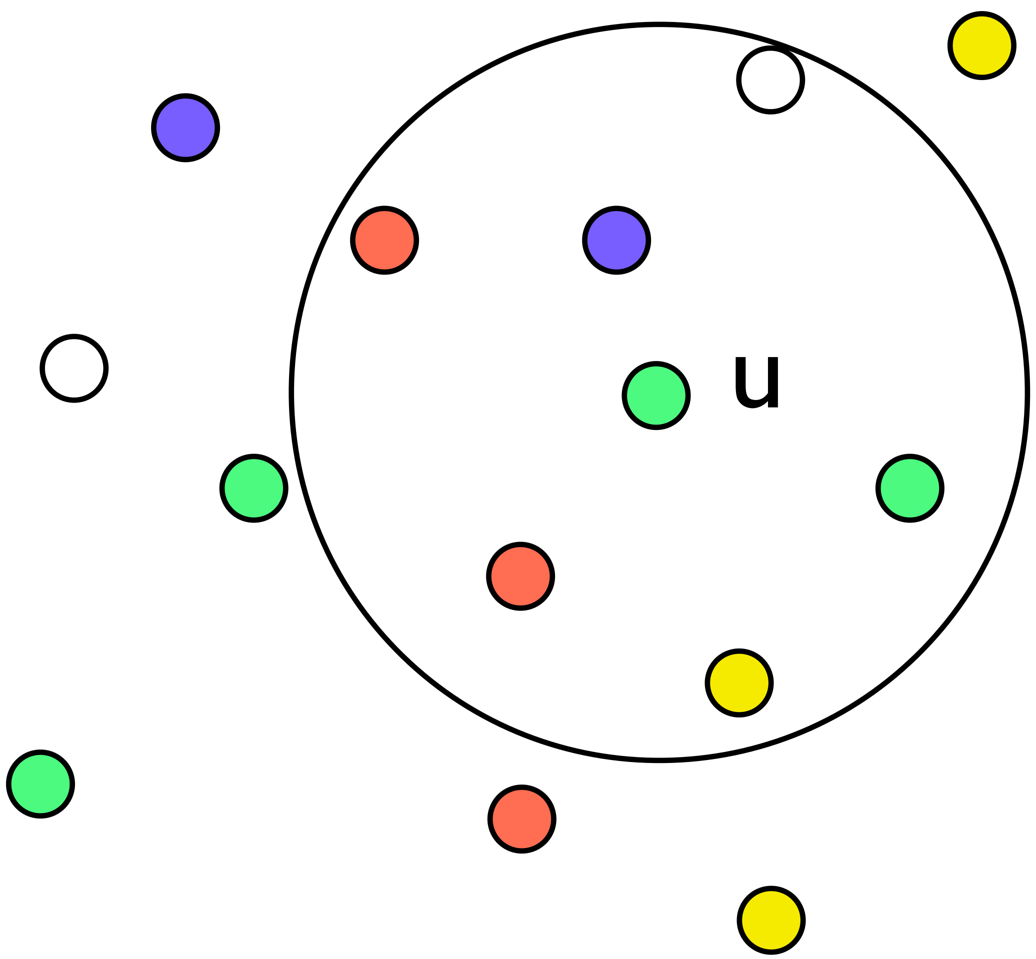
Boules de voisinage



B(u) - Les plus proches sommet de u respectant les propriétés :

- de **complétude**, la boule contient un représentant de chaque couleur,
- de **minimalité**, en nombre de sommet.

Boules de voisinage

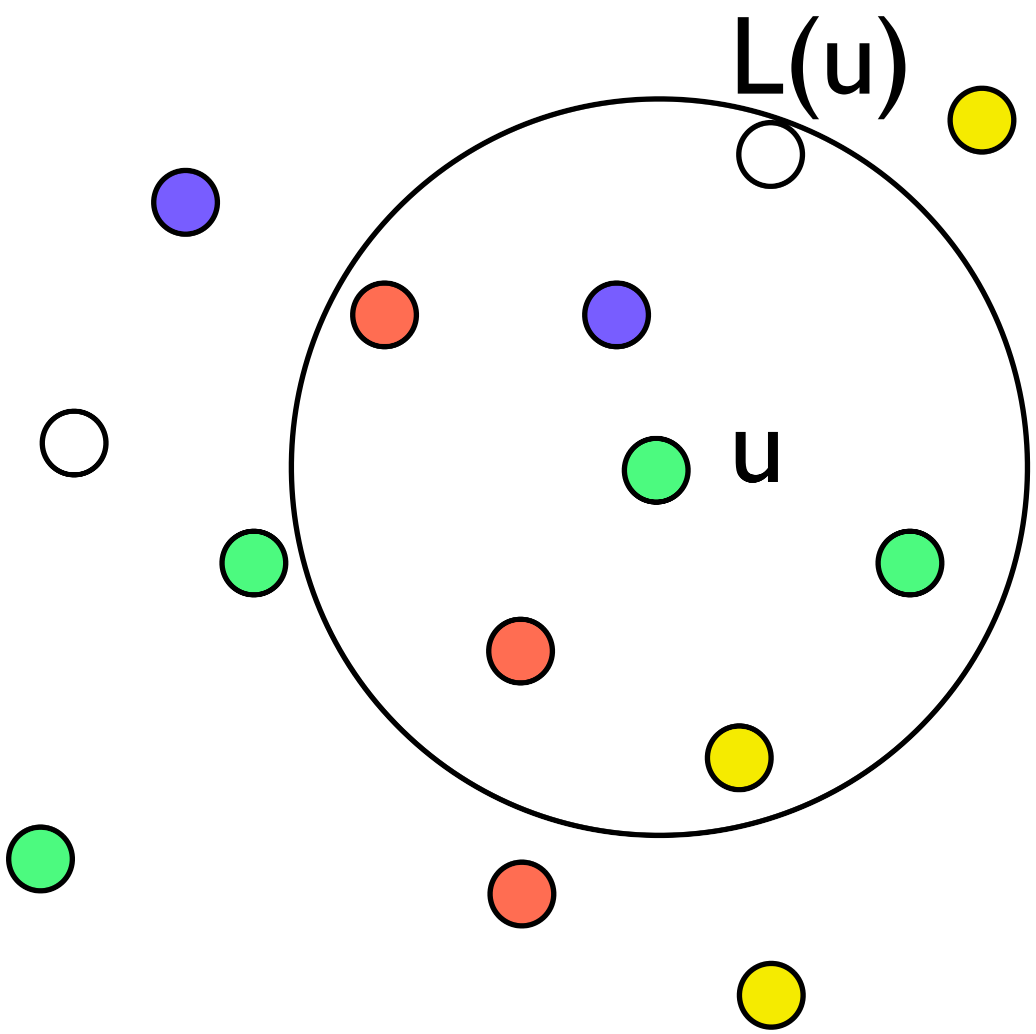


B(u) - Les plus proches sommet de u respectant les propriétés :

- de **complétude**, la boule contient un représentant de chaque couleur,
- de **minimalité**, en nombre de sommet.

$| B(u) | = O(n/k) = O(\sqrt{n})$

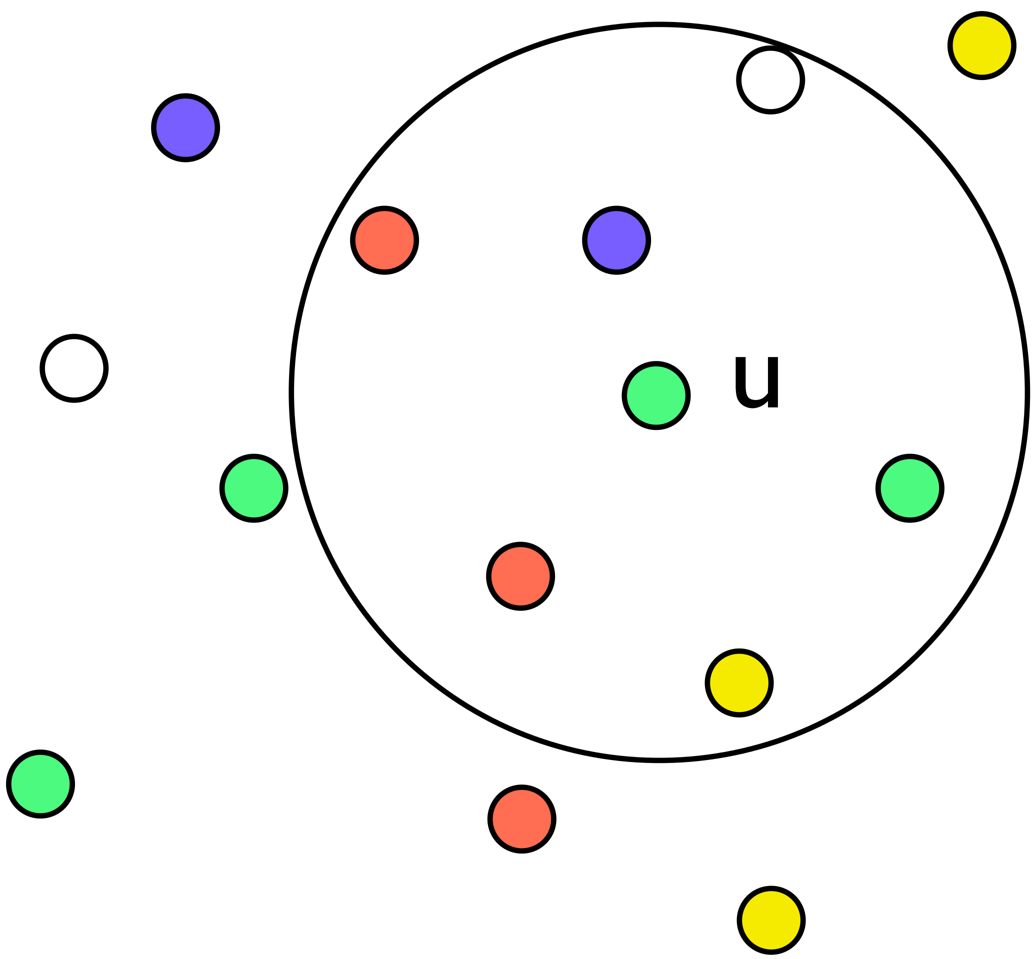
Boules de voisinage



On définit de plus L(u) comme le plus proche landmark de u

$$L(u) \in B(u)$$

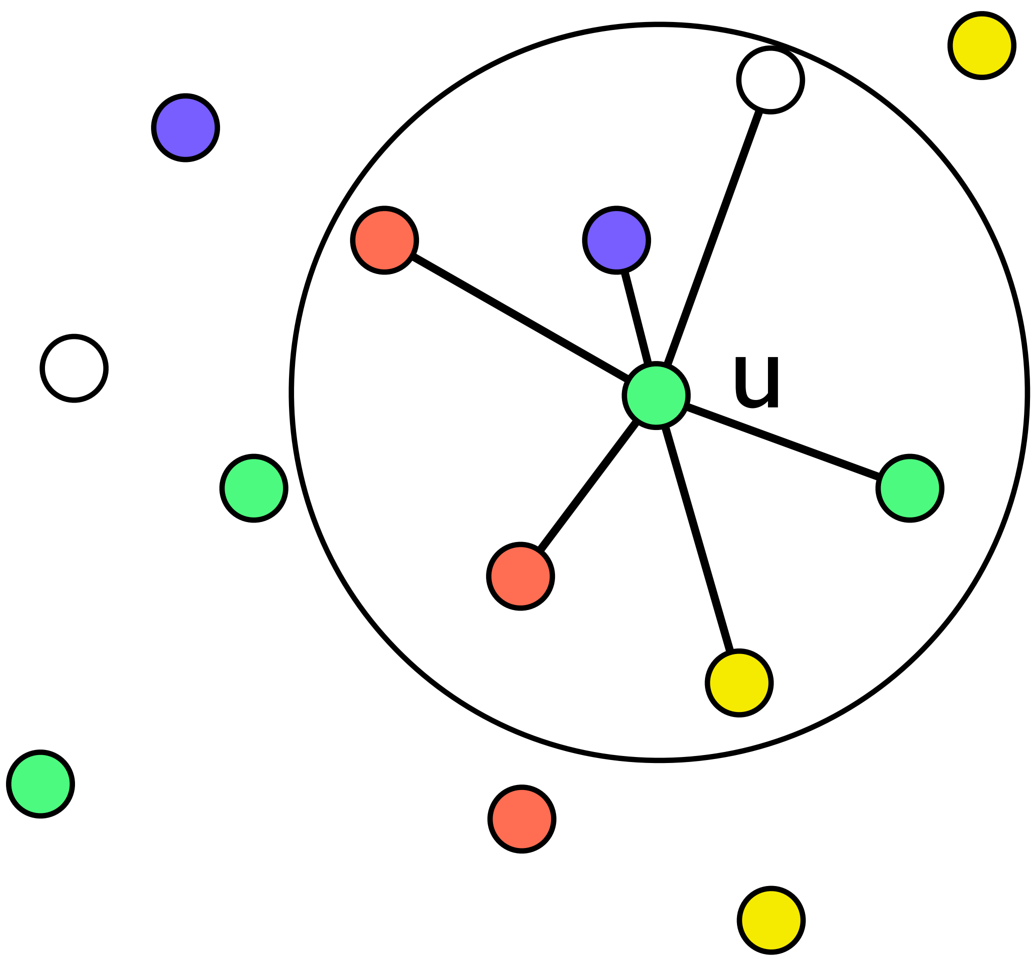
Connaissance
(Tables de routage)



Tout sommet u doit connaître une route vers les sommets :

Connaissance

(Tables de routage)

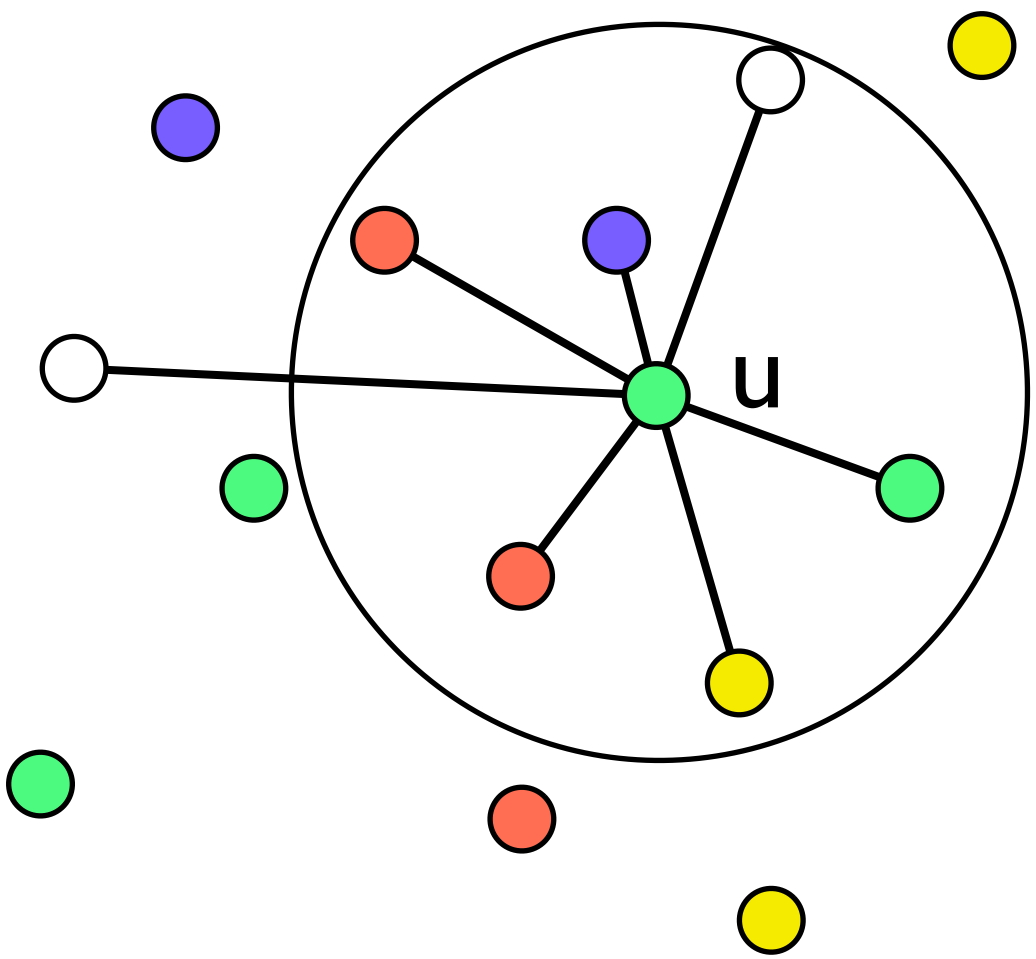


Tout sommet u doit connaître une route vers les sommets :

- B(u), sommets proches

Connaissance

(Tables de routage)

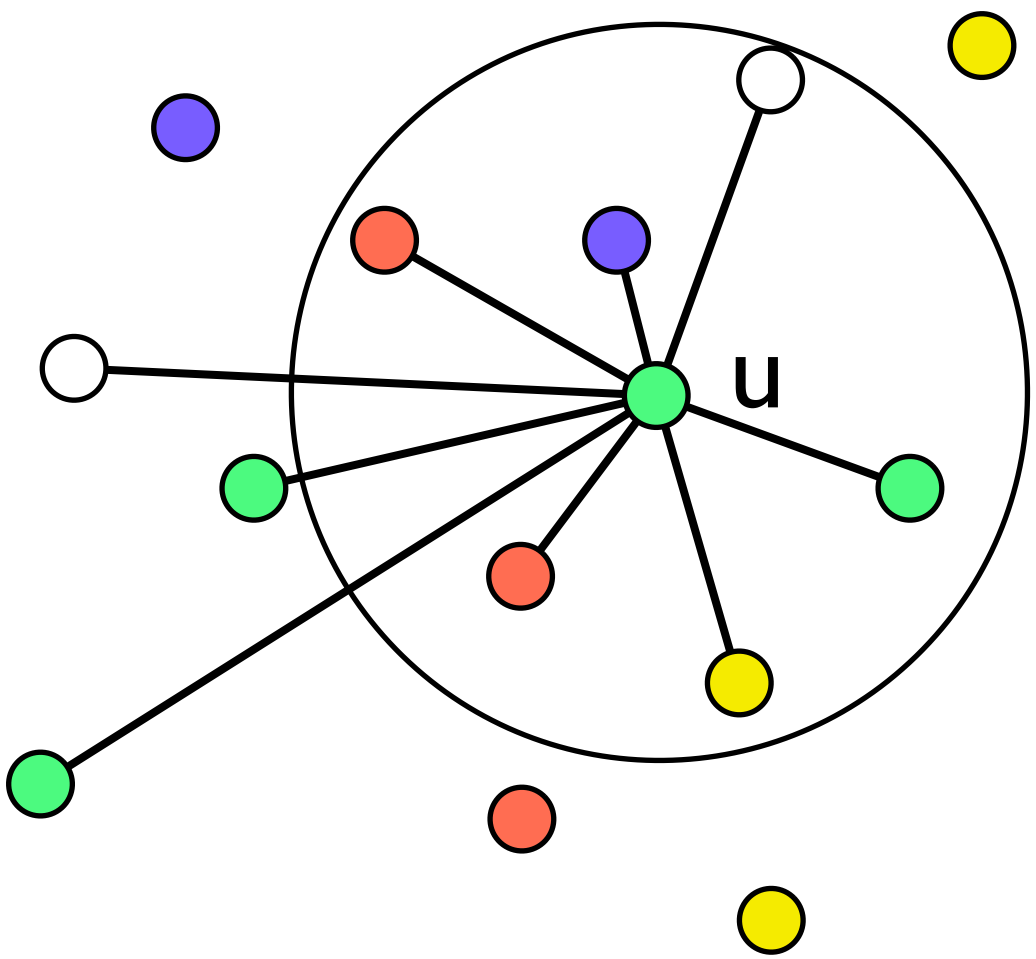


Tout sommet u doit connaître une route vers les sommets :

- B(u), sommets proches
- L, les landmarks

Connaissance

(Tables de routage)

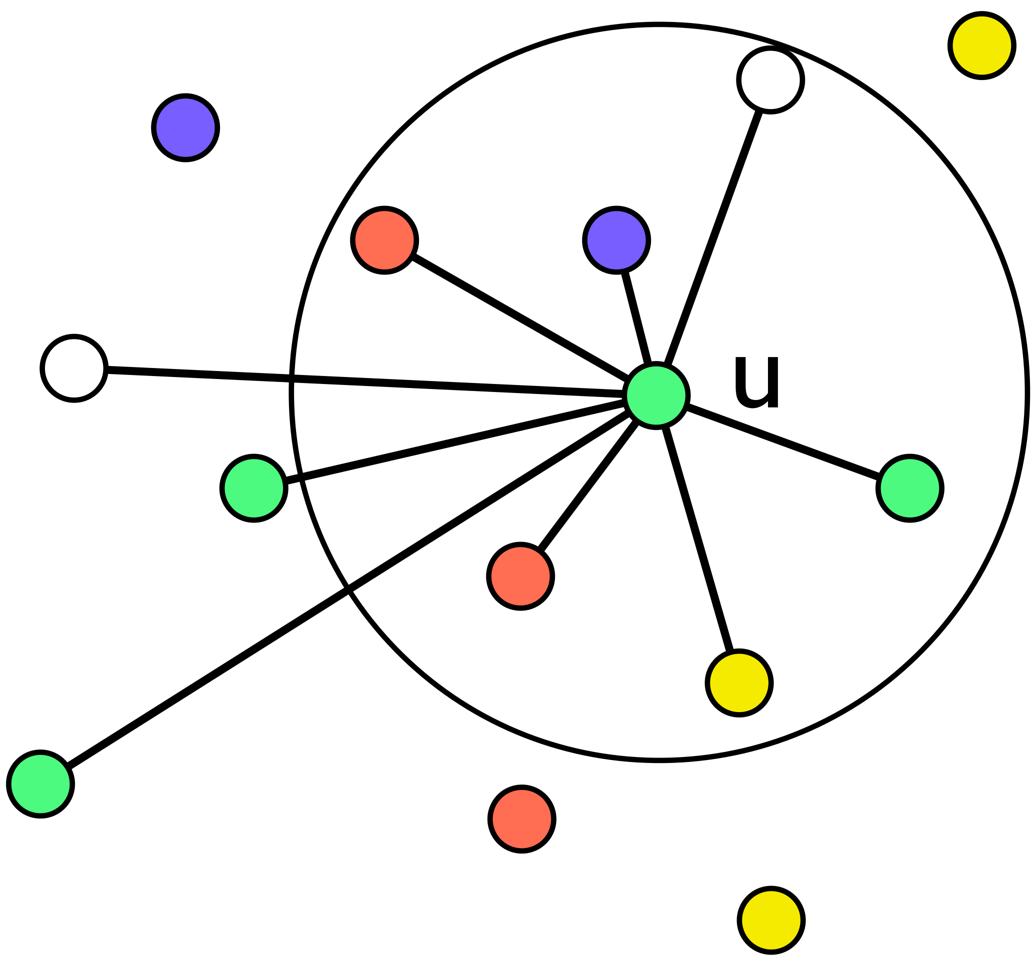


Tout sommet u doit connaître une route vers les sommets :

- B(u), sommets proches
- L, les landmarks
- C(u), les sommets de la même couleur

Connaissance

(Tables de routage)

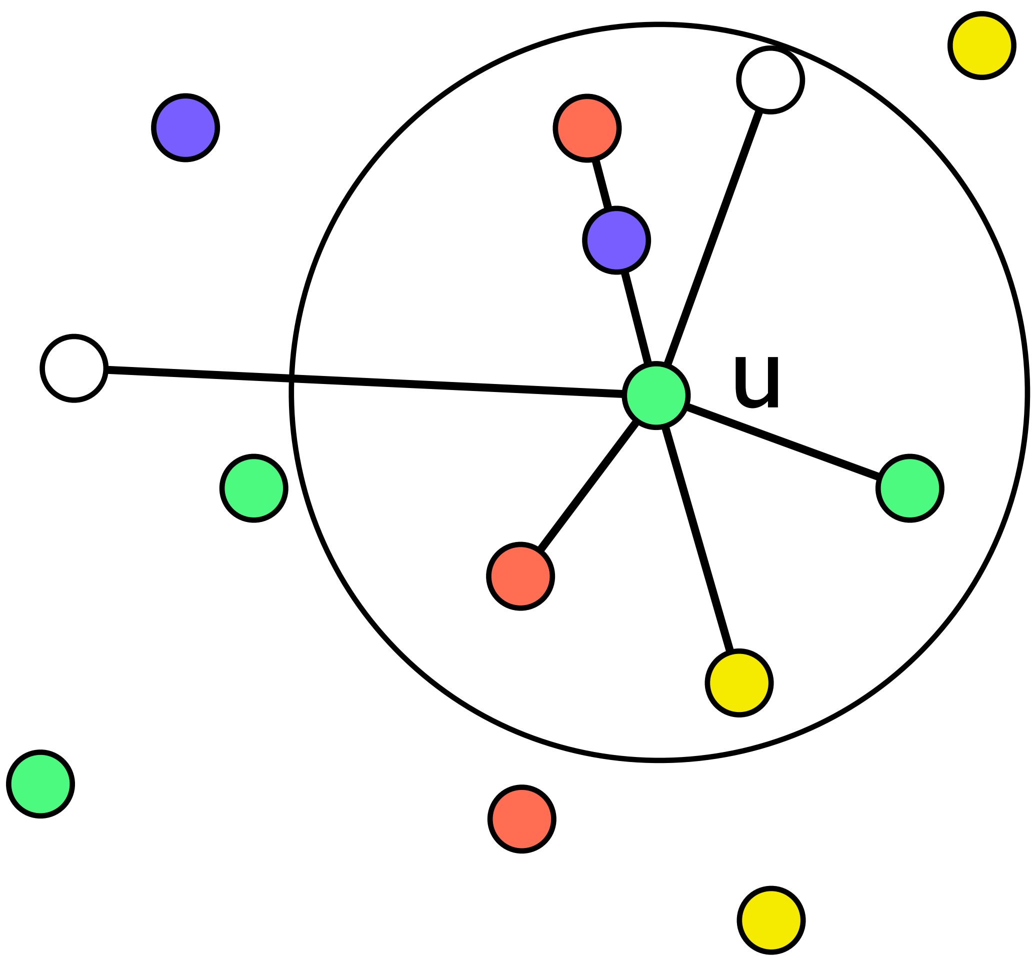


Tout sommet u doit connaître une route vers les sommets :

- B(u), sommets proches
- L, les landmarks
- C(u), les sommets de la même couleur

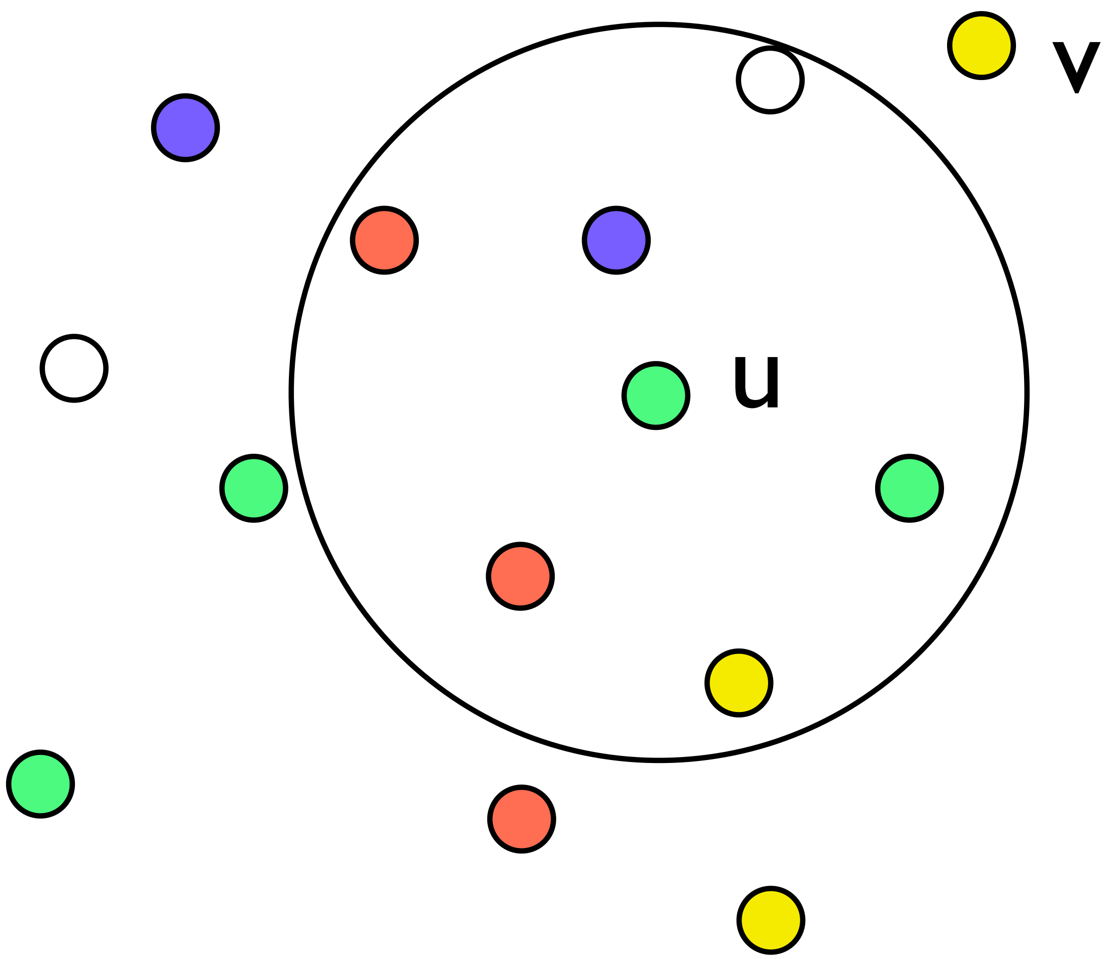
$O(k + 2n/k) = O(\sqrt{n})$ entrées

Routage

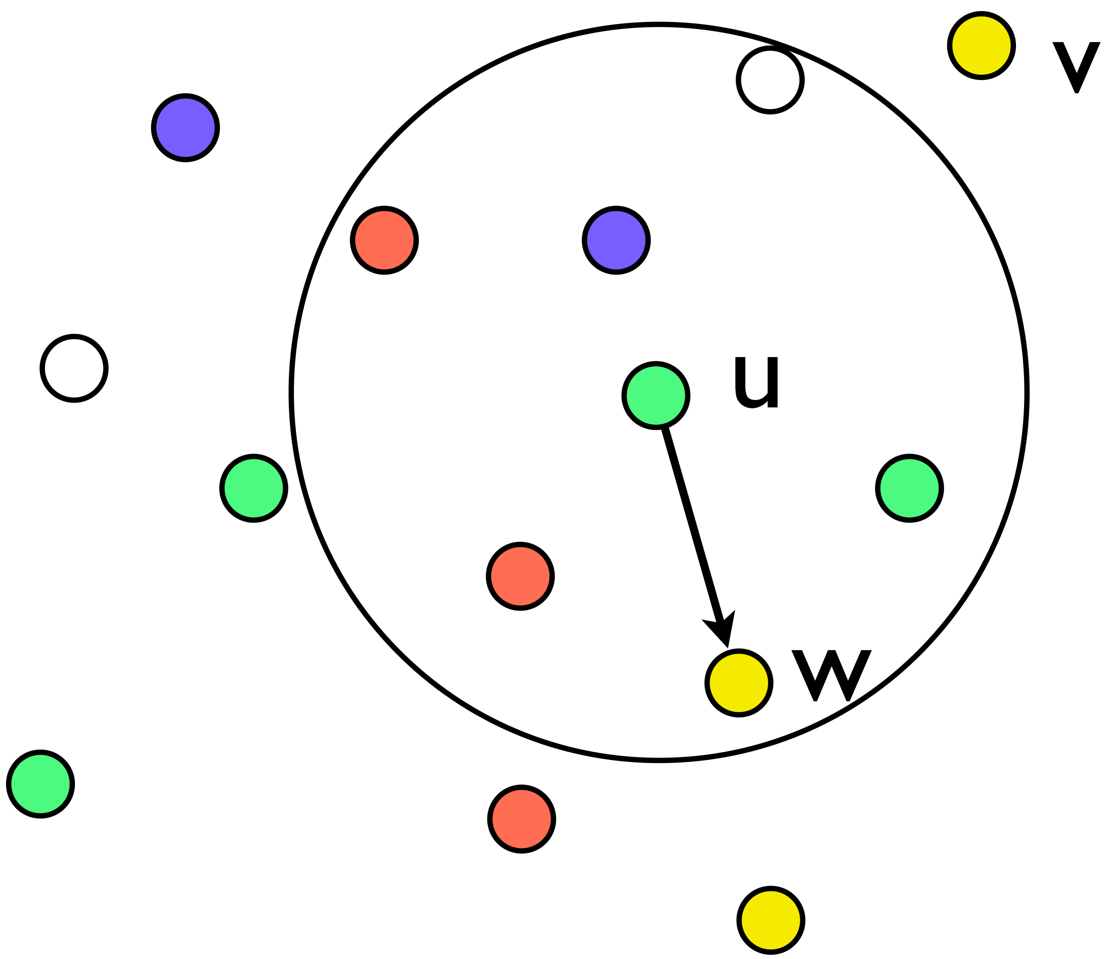


Le protocole de routage de u vers un sommet de B(u) ou L est du routage de plus court chemin.

Routage

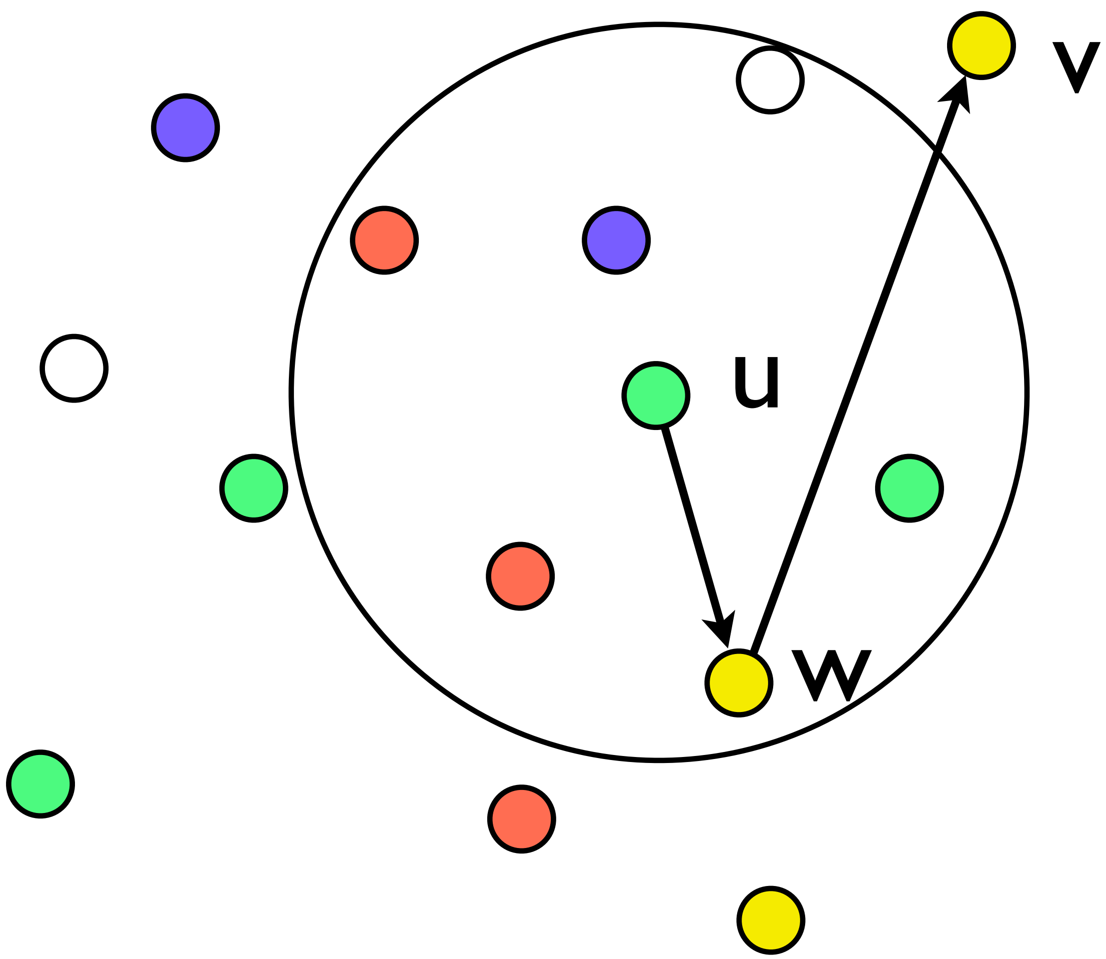


Routage



Pour router vers une autre destination v, u route via un sommet intermédiaire w de B(u) tel que : c(w) = c(v).

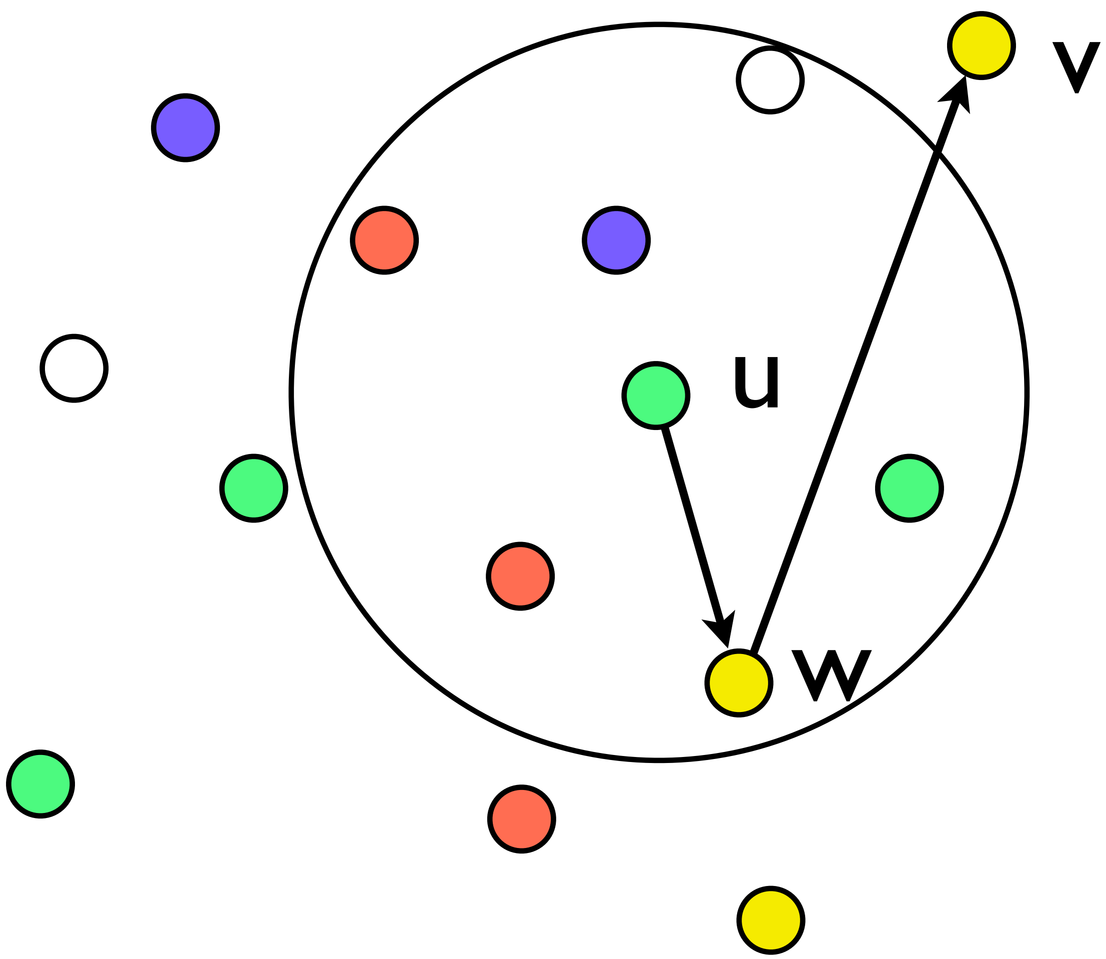
Routage



Pour router vers une autre destination v, u route via un sommet intermédiaire w de B(u) tel que : c(w) = c(v).

w connaît une route pour tous les sommets de sa couleur.

Routage



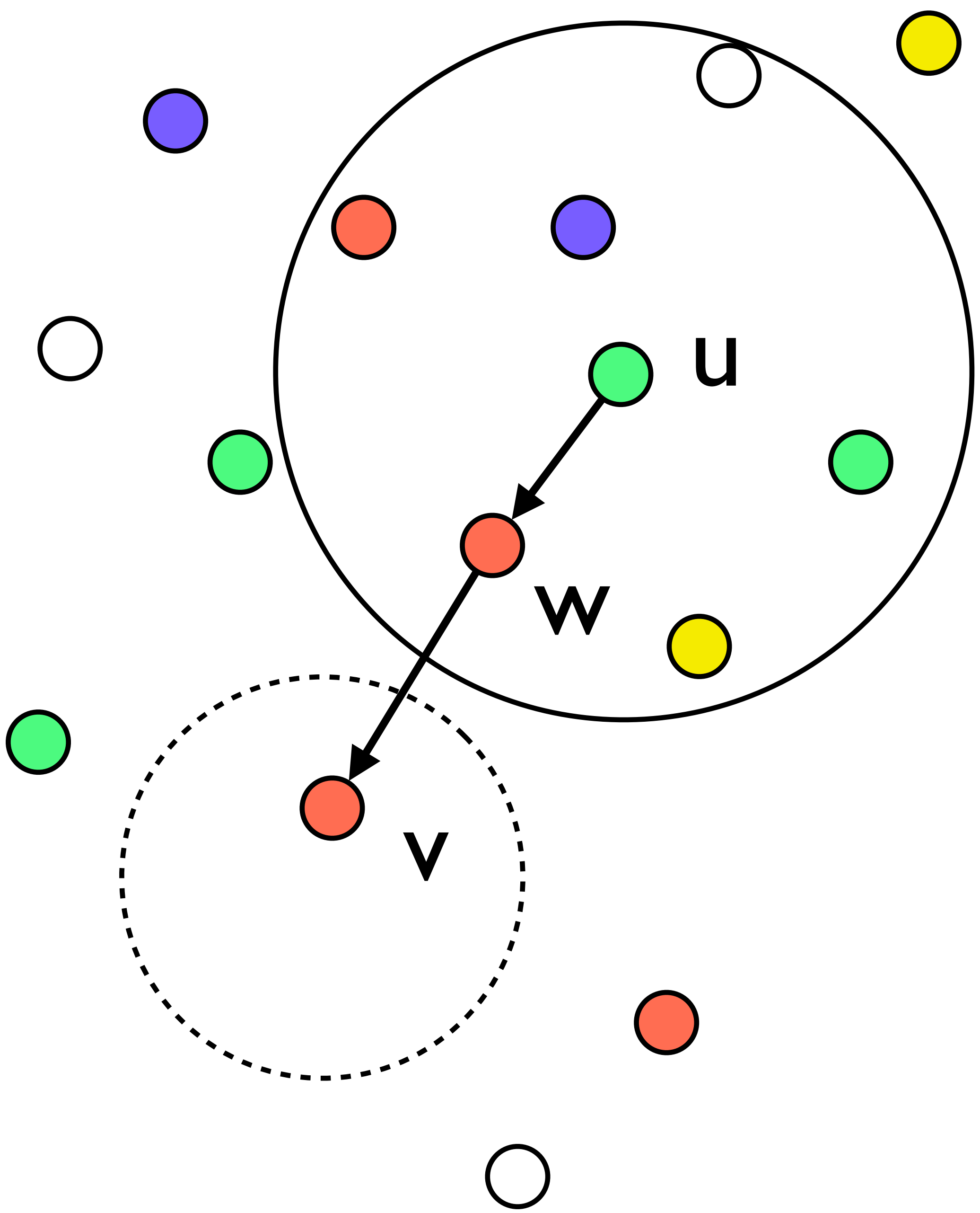
Pour router vers une autre destination v, u route via un sommet intermédiaire w de B(u) tel que : c(w) = c(v).

w connaît une route pour tous les sommets de sa couleur.

w est le représentant de la couleur c(v) pour u.

Routage

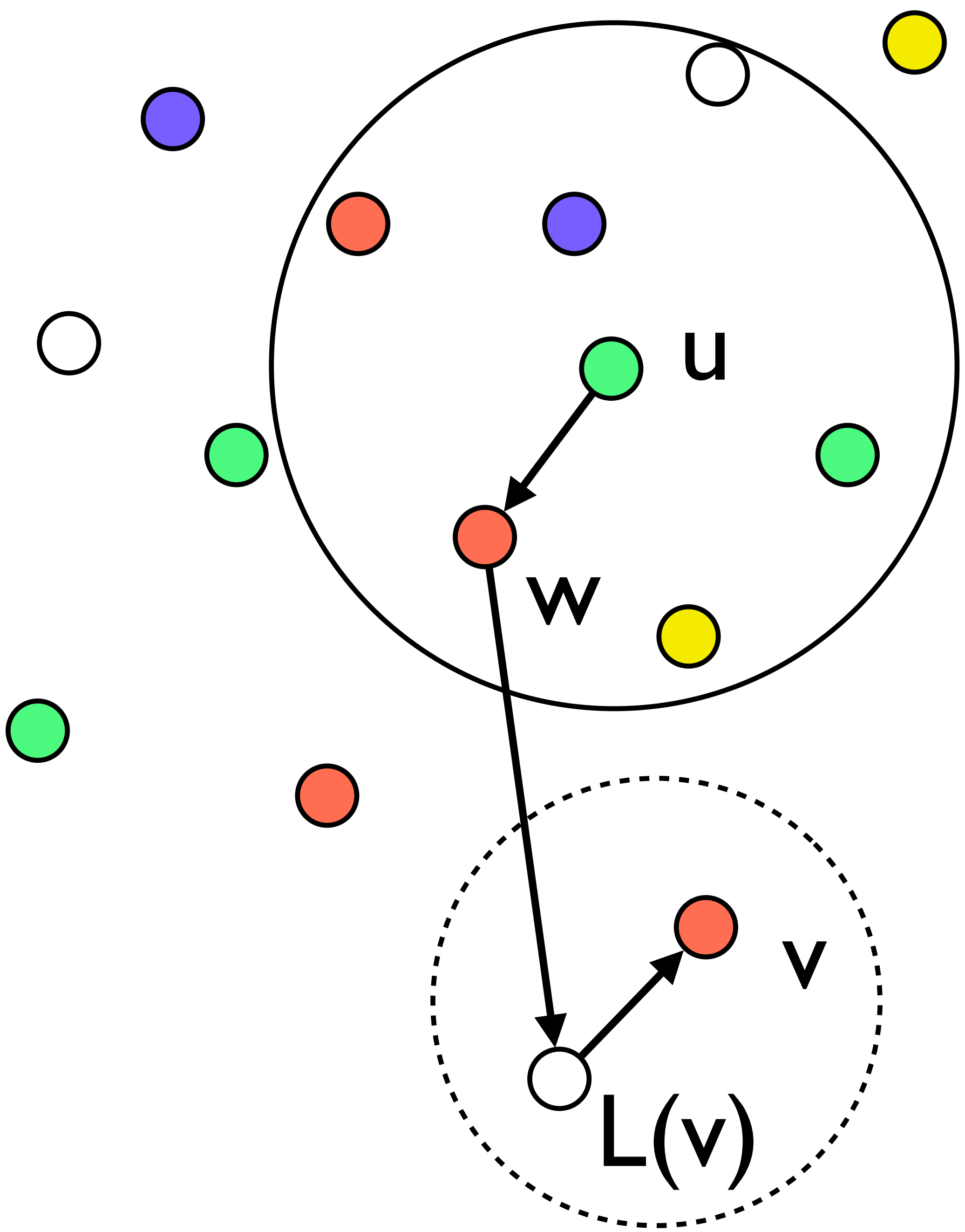
par intermédiaire



Si le représentant w choisit par u est ***suffisamment proche*** de la destination v, alors w stocke explicitement le chemin vers v.

Routage

par intermédiaire



Sinon, w stocke un chemin via le plus proche landmark de v, L(v).

Cela garanti un étirement ≤ 3 dans tous les cas.

Résumé

	construction	mise à jour
Boules	O	O
Landmarks	O	O
Boules Contigues	O	X
Couleurs	O	X

Résumé

	construction	mise à jour
Boules	O	O
Landmarks	O	O
Boules Contigues	O	X
Couleurs	O	X

Boules Contiguës

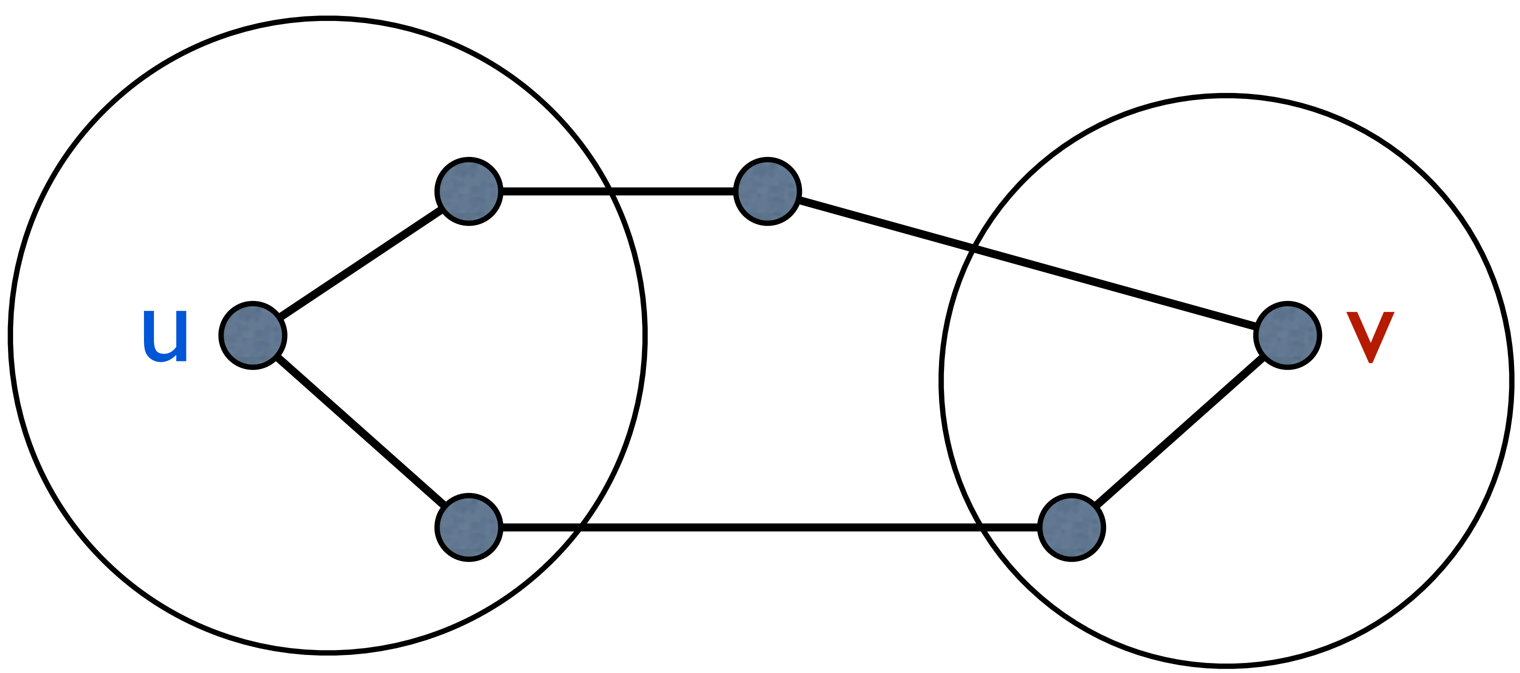
Construction

Distribuée des tables

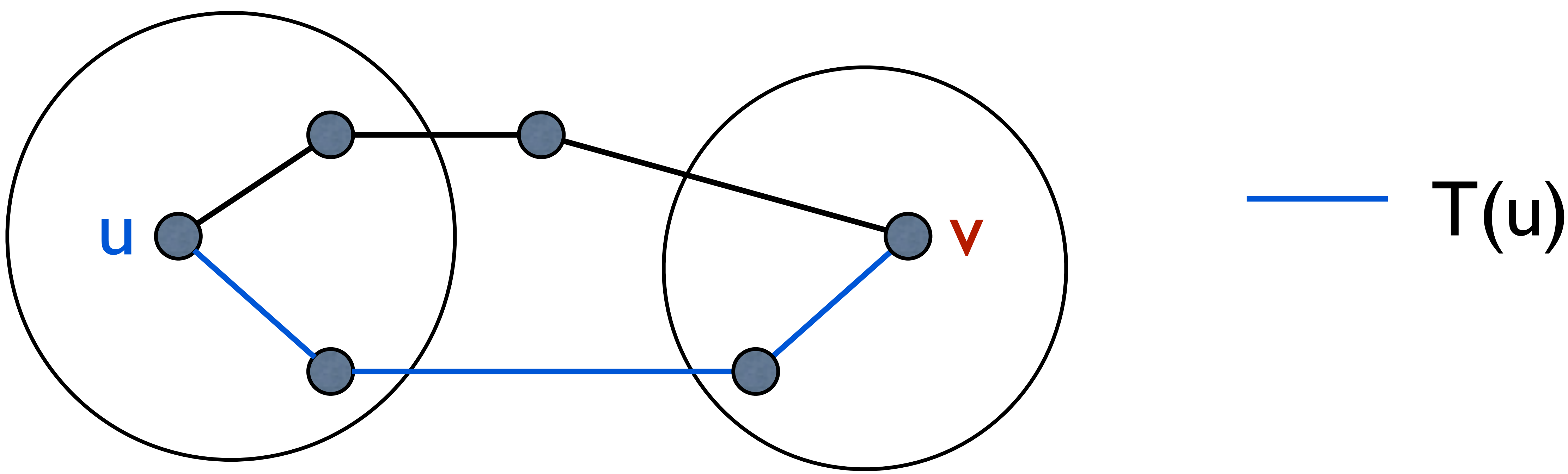
Modèle

- **Synchrone**
- **Passage de message**
- Mesures de performance :
  - **coût de communication** : nombre total d'entrées échangées
  - **étirement** : len(u → v)/d(u,v)
  - **mémoire :** nombre d'entrées (information Topologique ou de routage)

Boules contiguës

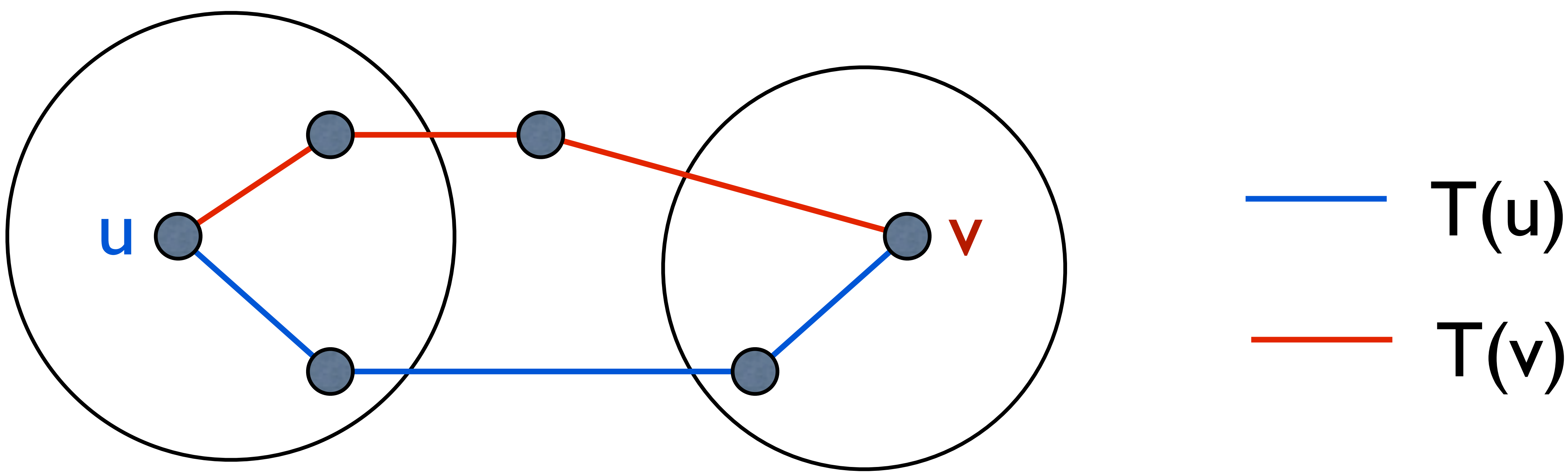


Boules contiguës



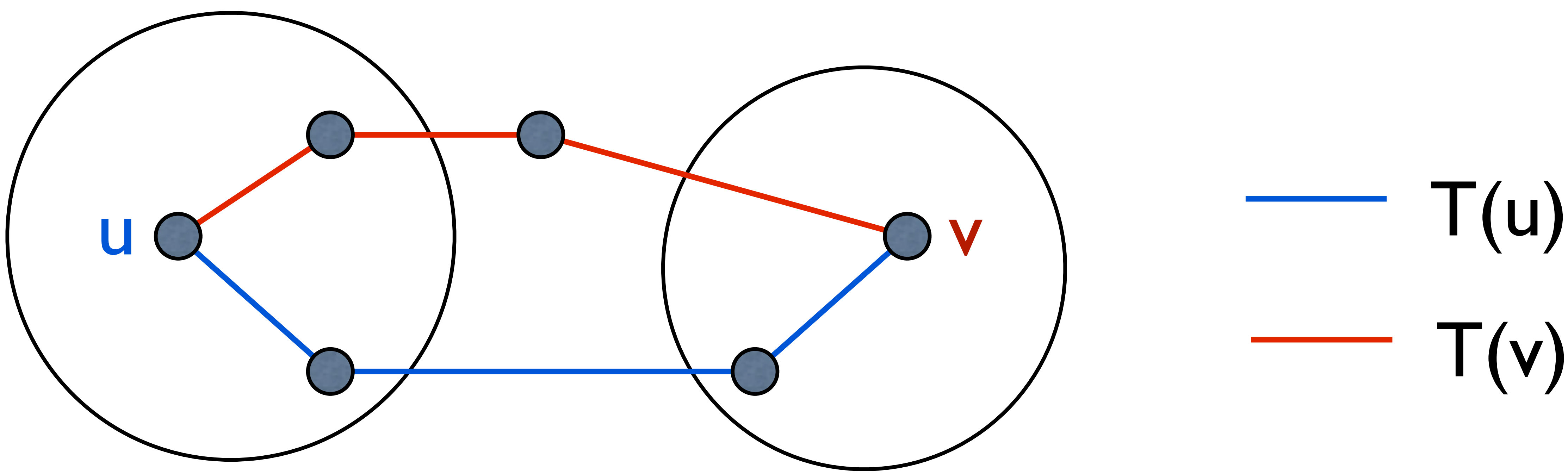
B(u) **EST** contiguë à B(v)

Boules contiguës



B(u) **EST** contiguë à B(v)

Boules contiguës



B(u) **EST** contiguë à B(v)

MAIS

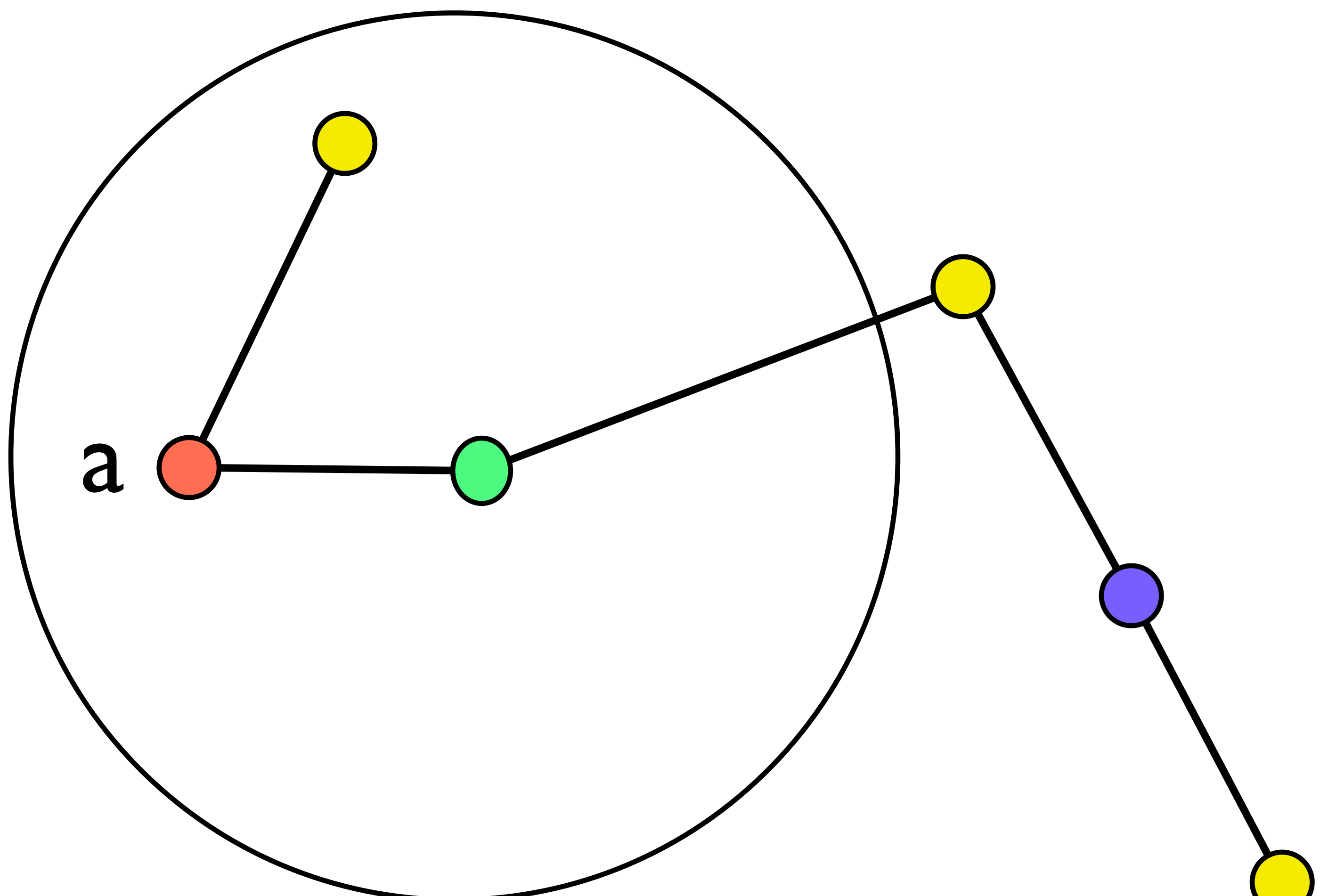
B(v) **N'EST PAS** contiguë à B(u)

Boules contiguës

- Tout sommet u contigu à v doit apprendre :
  - Sa distance au sommet v : d(u,v)
  - Le chemin dans B(v) jusqu'à v

Boules contiguës

messages : ID (distance, chemin)

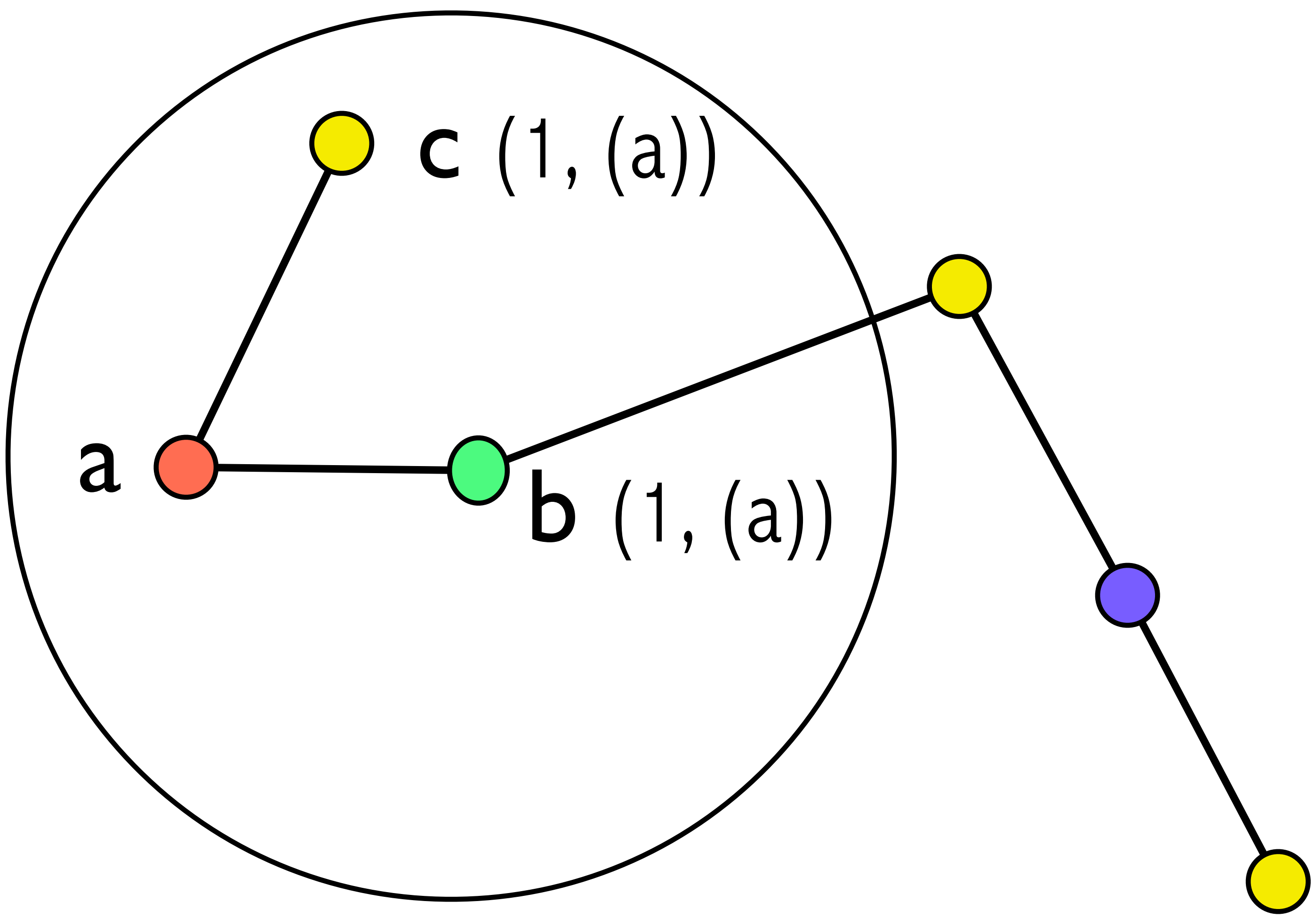


$d \notin B(f)$

$d \in B(e)$

Boules contiguës

messages : ID (distance, chemin)

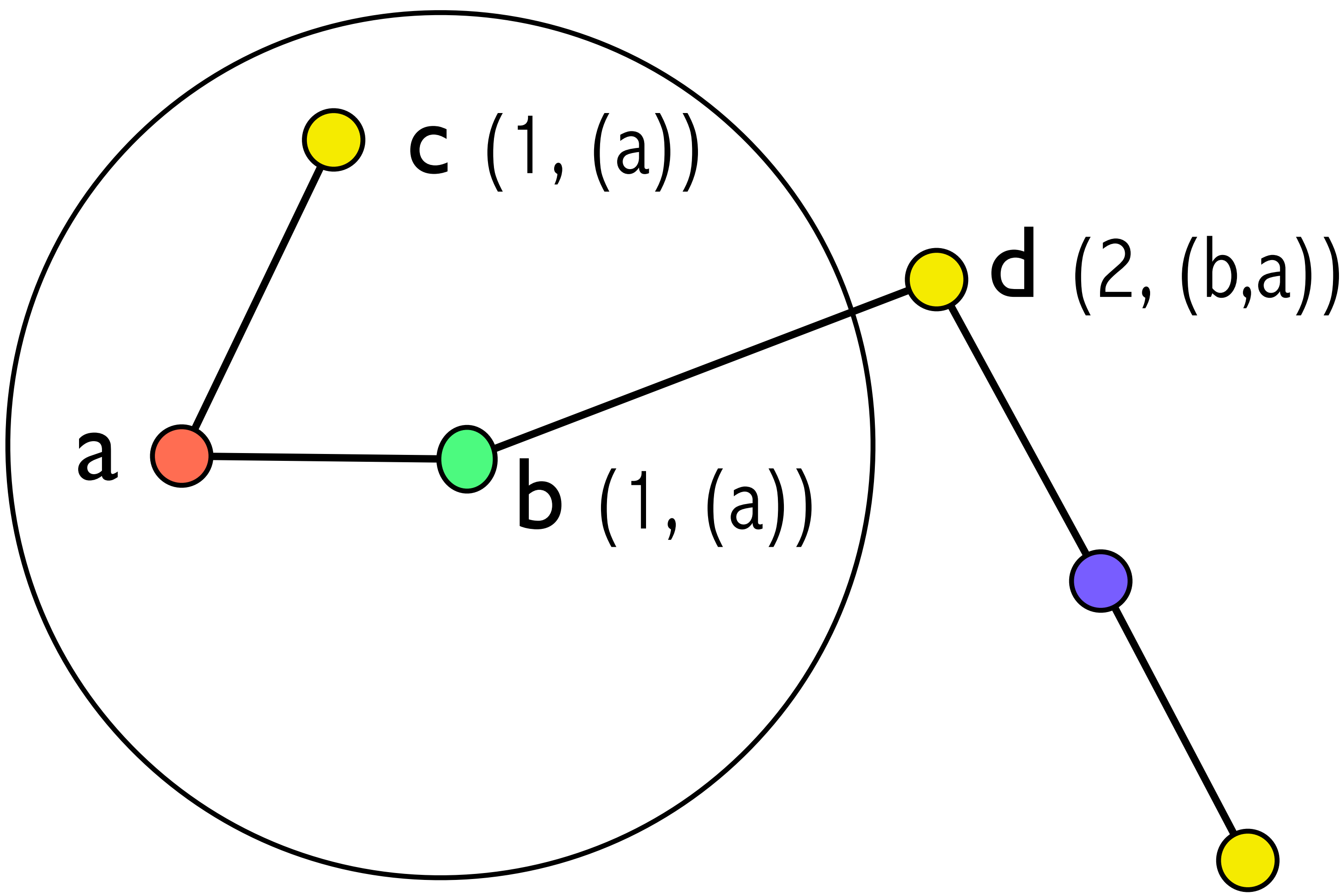


$d \notin B(f)$

$d \in B(e)$

Boules contiguës

messages : ID (distance, chemin)

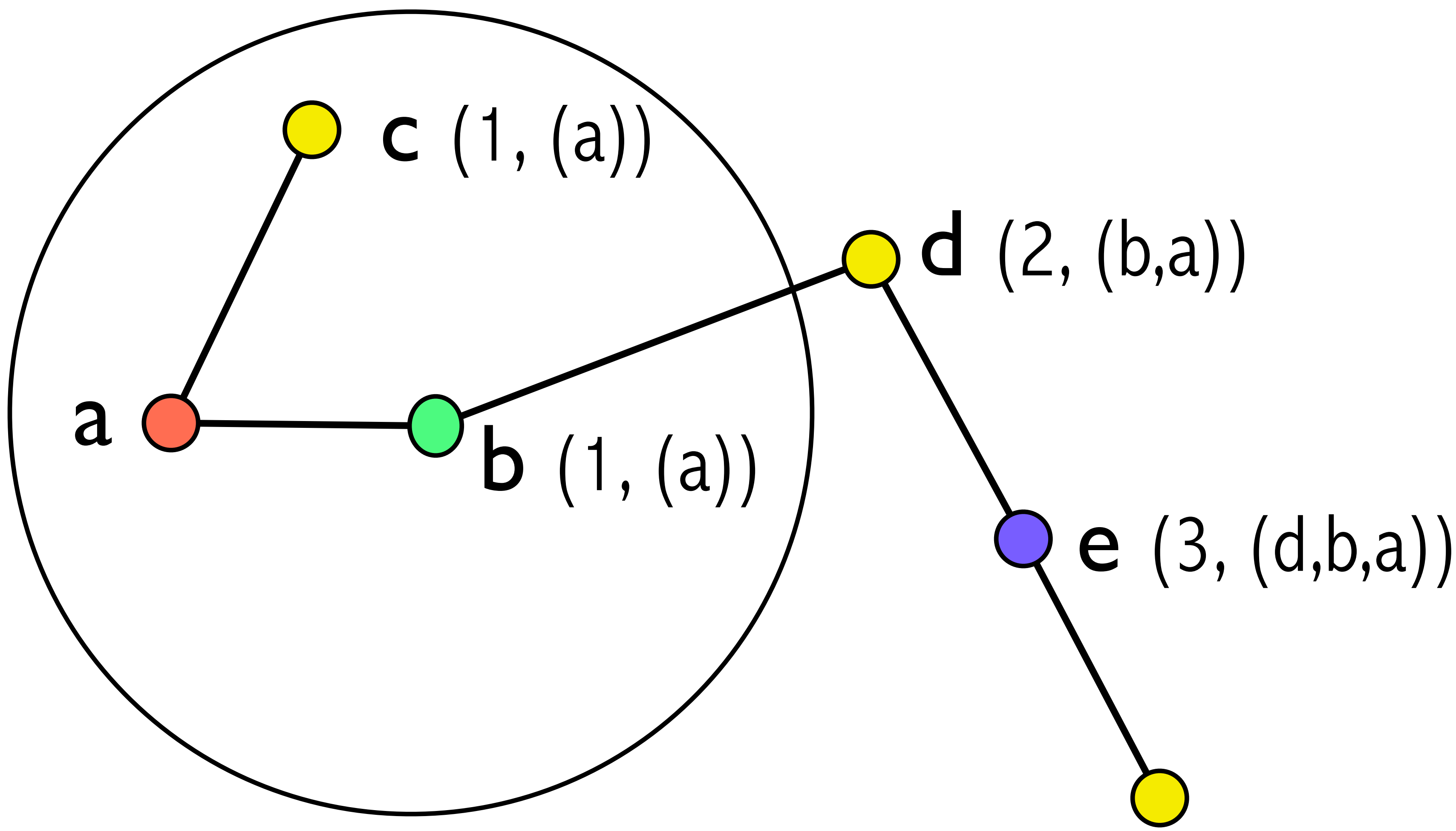


$d \notin B(f)$

$d \in B(e)$

Boules contiguës

messages : ID (distance, chemin)

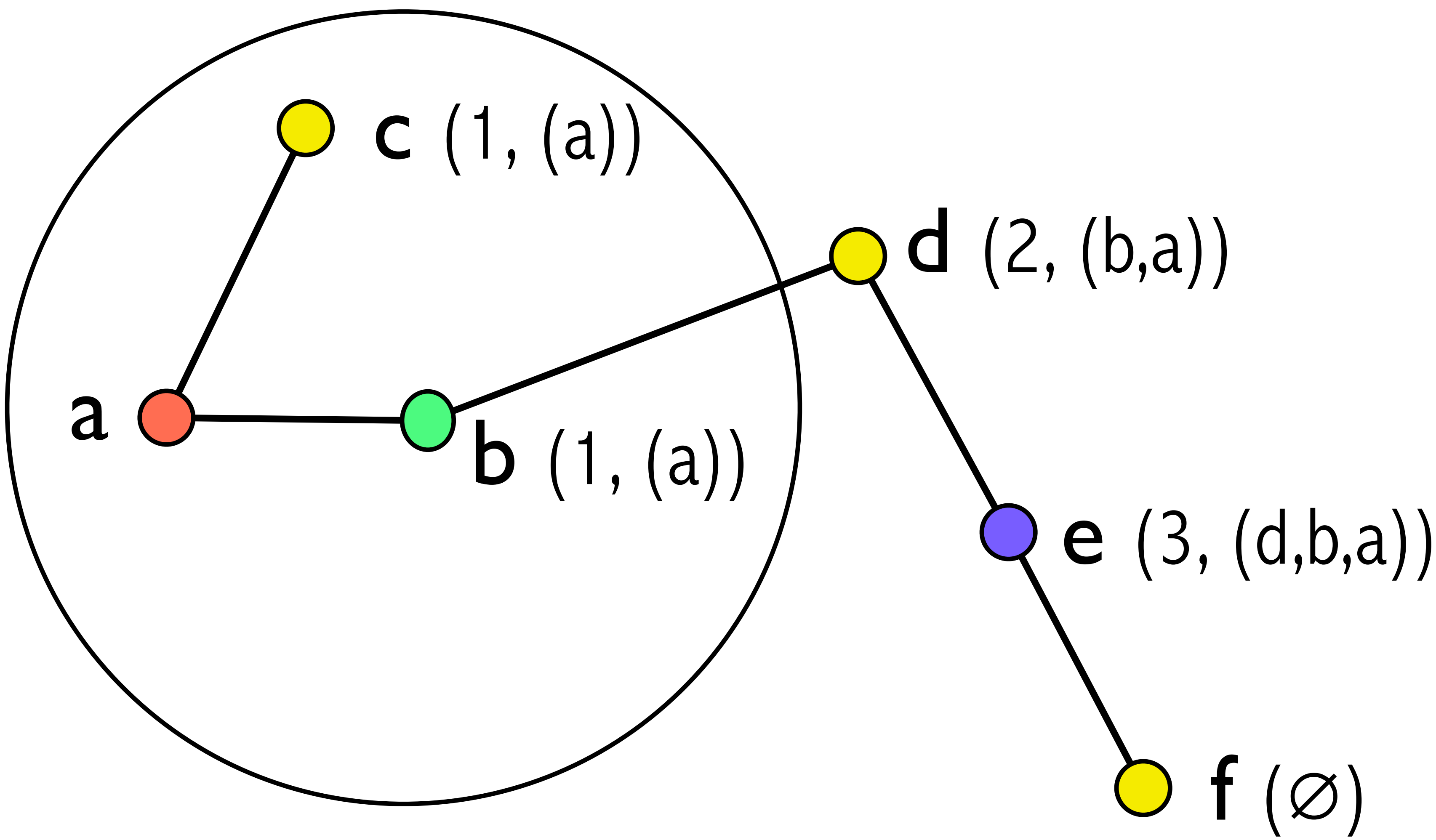


$d \notin B(f)$

$d \in B(e)$

Boules contiguës

messages : ID (distance, chemin)

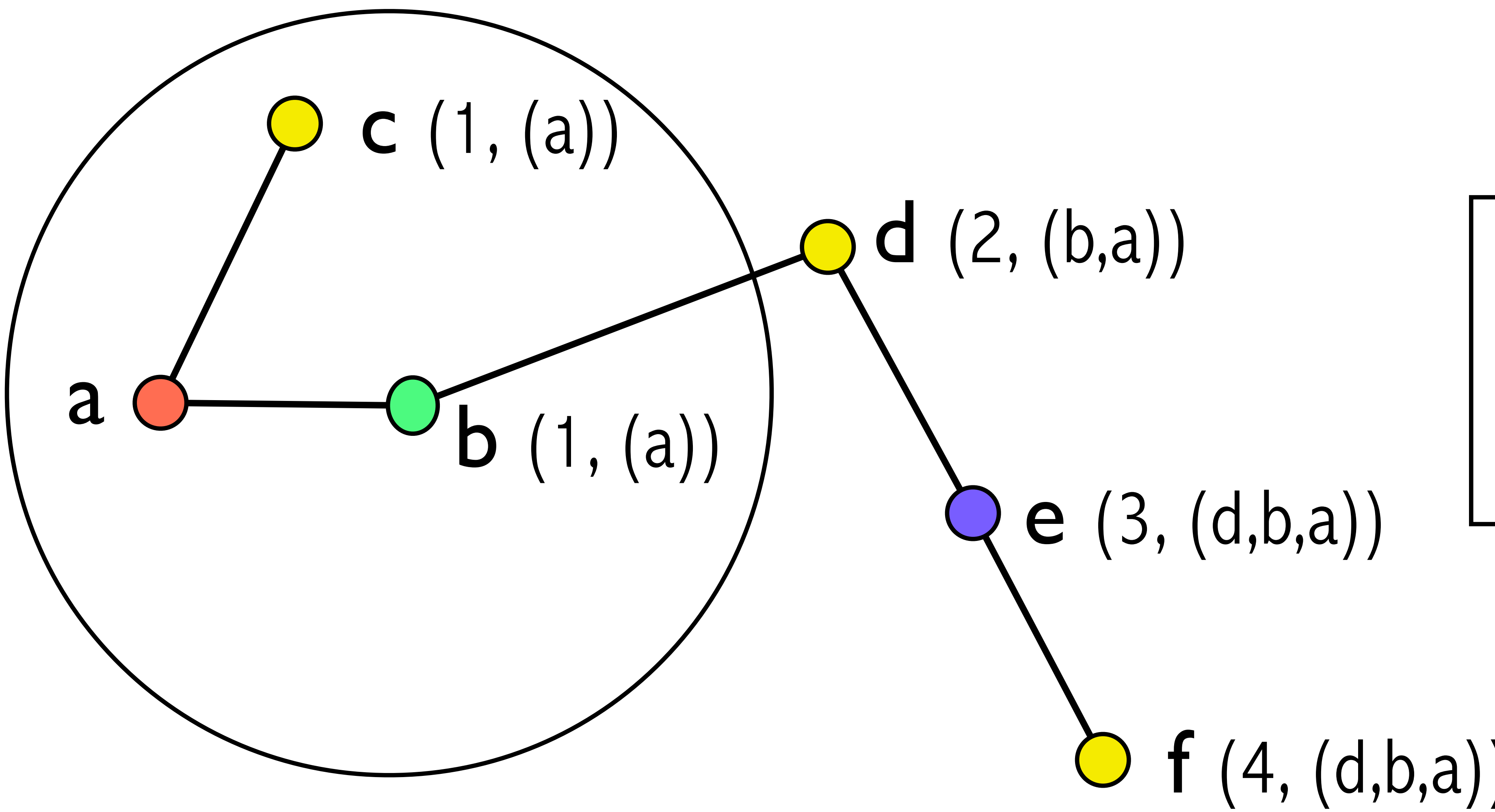


$d \notin B(f)$

$d \in B(e)$

Boules contiguës

messages : ID (distance, chemin)



$d \in B(f)$

$d \in B(e)$

“Suffisamment Proche” de la destination ?

On veut que : $\forall$ (w,v) $\in$ V

v contigu à w $\Rightarrow$ $\forall$ u $\in$ B(w) tq c(u) = c(v),

u utilise le routage par boules contiguës

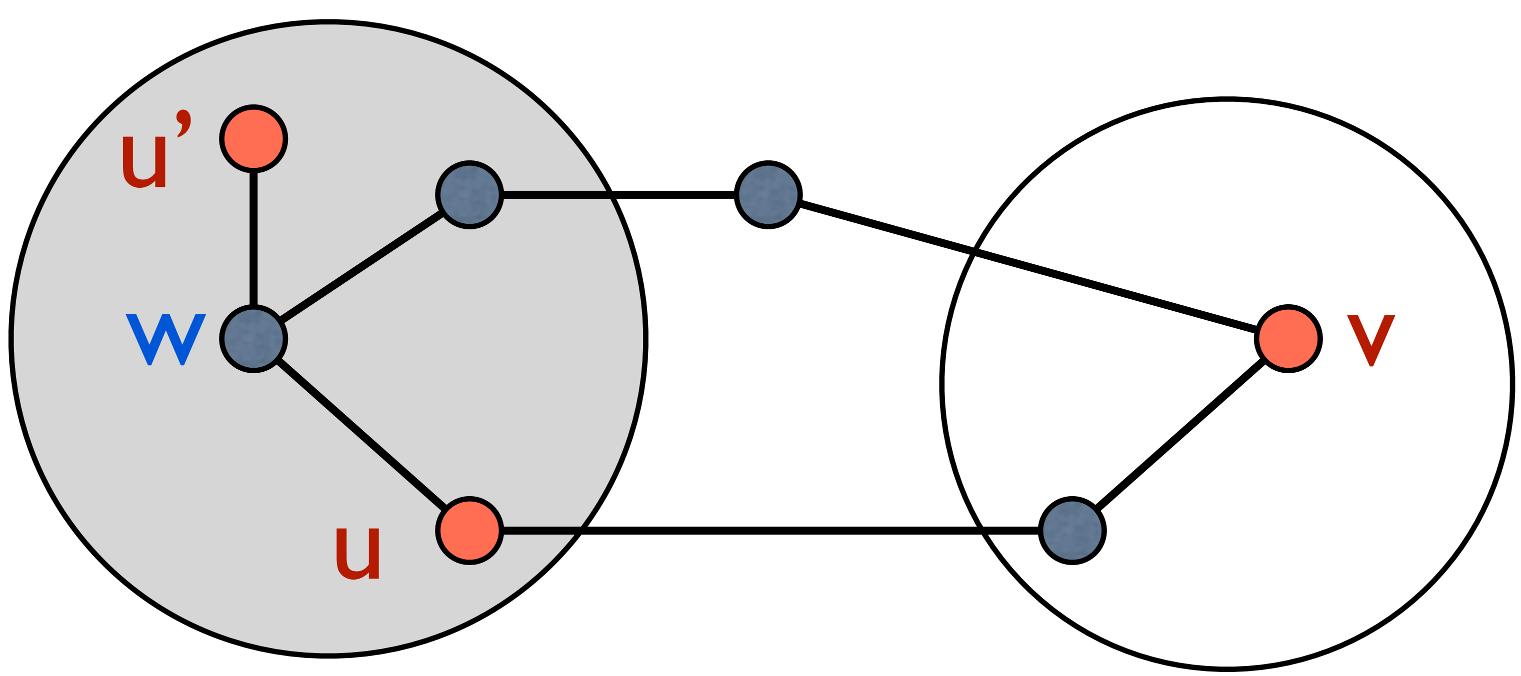


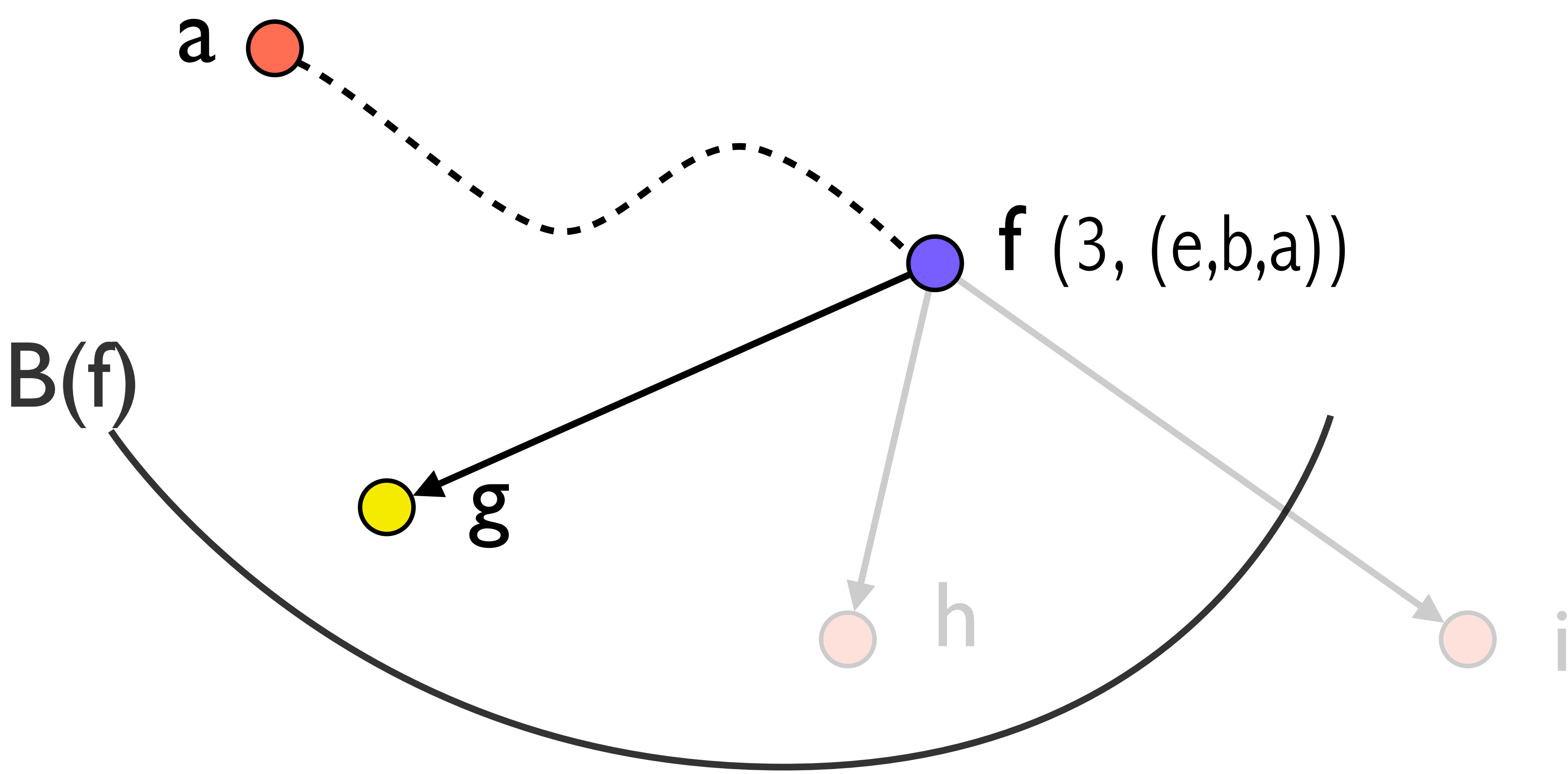
Table de couleur

Via boules contiguës

- Dès qu'un sommet prend connaissance d'un chemin via boules contiguës, il envoie à tous les sommets de sa boule cette information.

Table de couleur

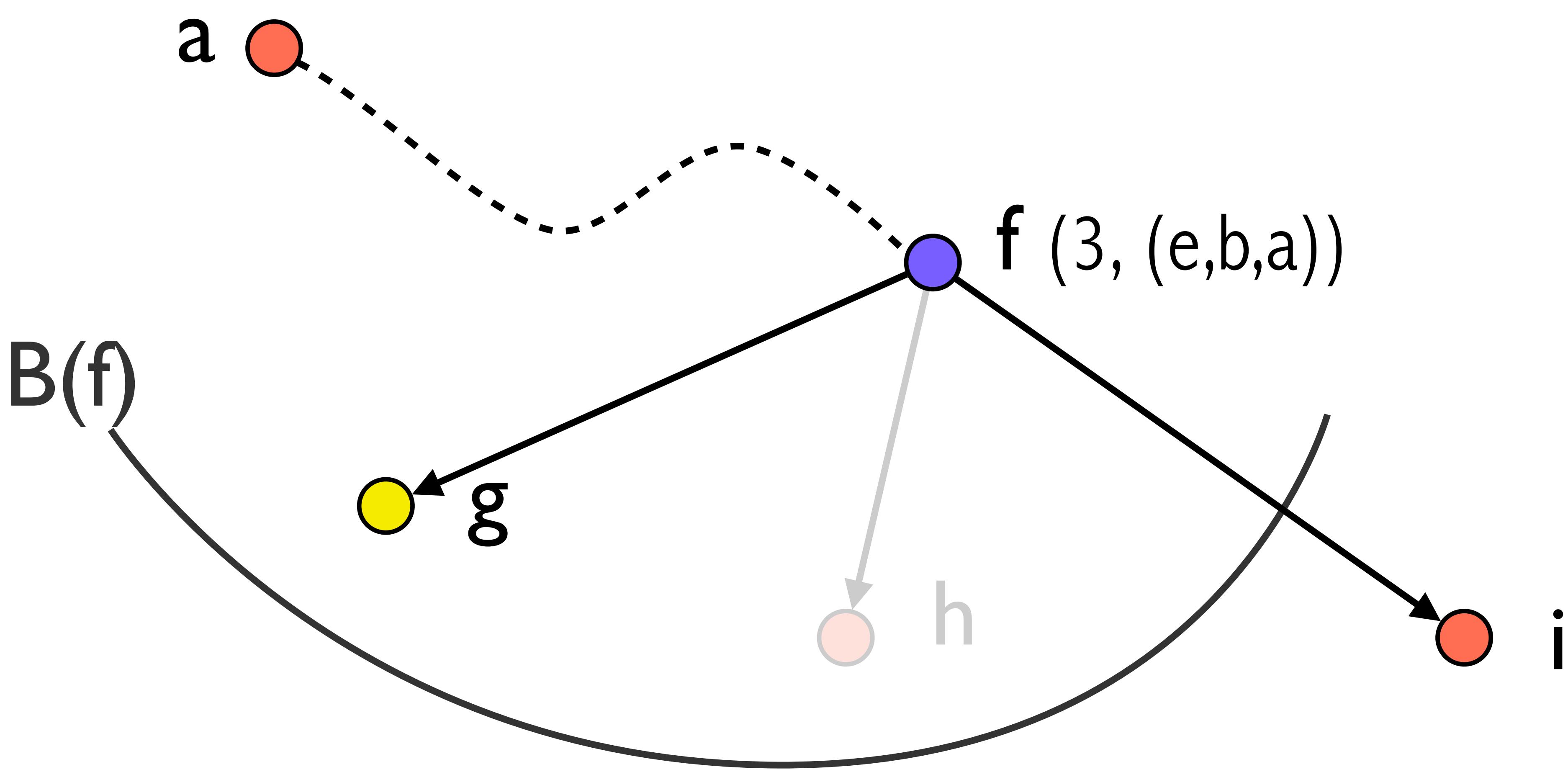
Via boules contiguës



g propage à son voisinage

Table de couleur

Via boules contiguës

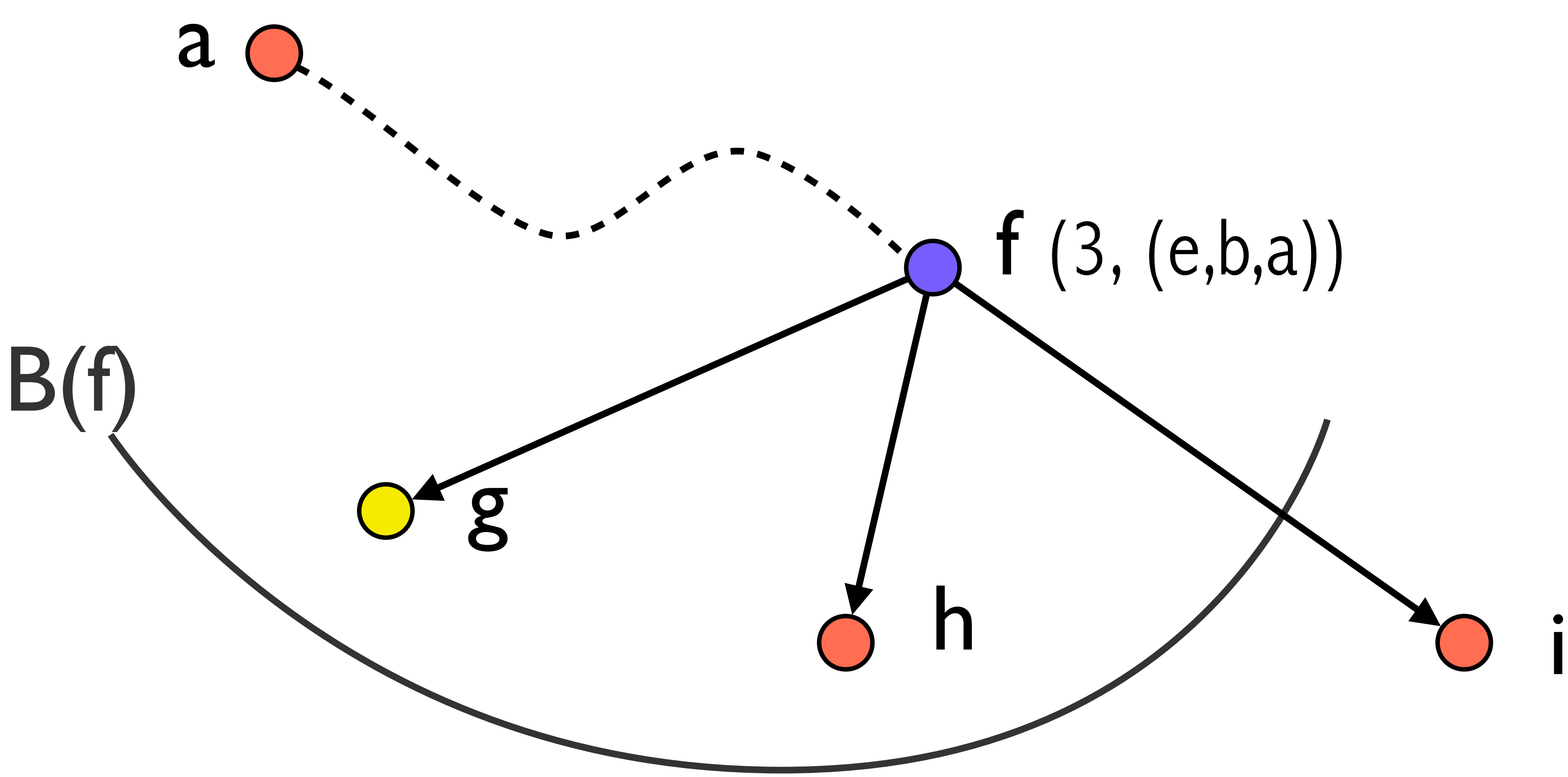


g propage à son voisinage

i ajoute une entrée dans sa table couleur

Table de couleur

Via boules contiguës



g propage à son voisinage

i ajoute une entrée dans sa table couleur

h fait les deux

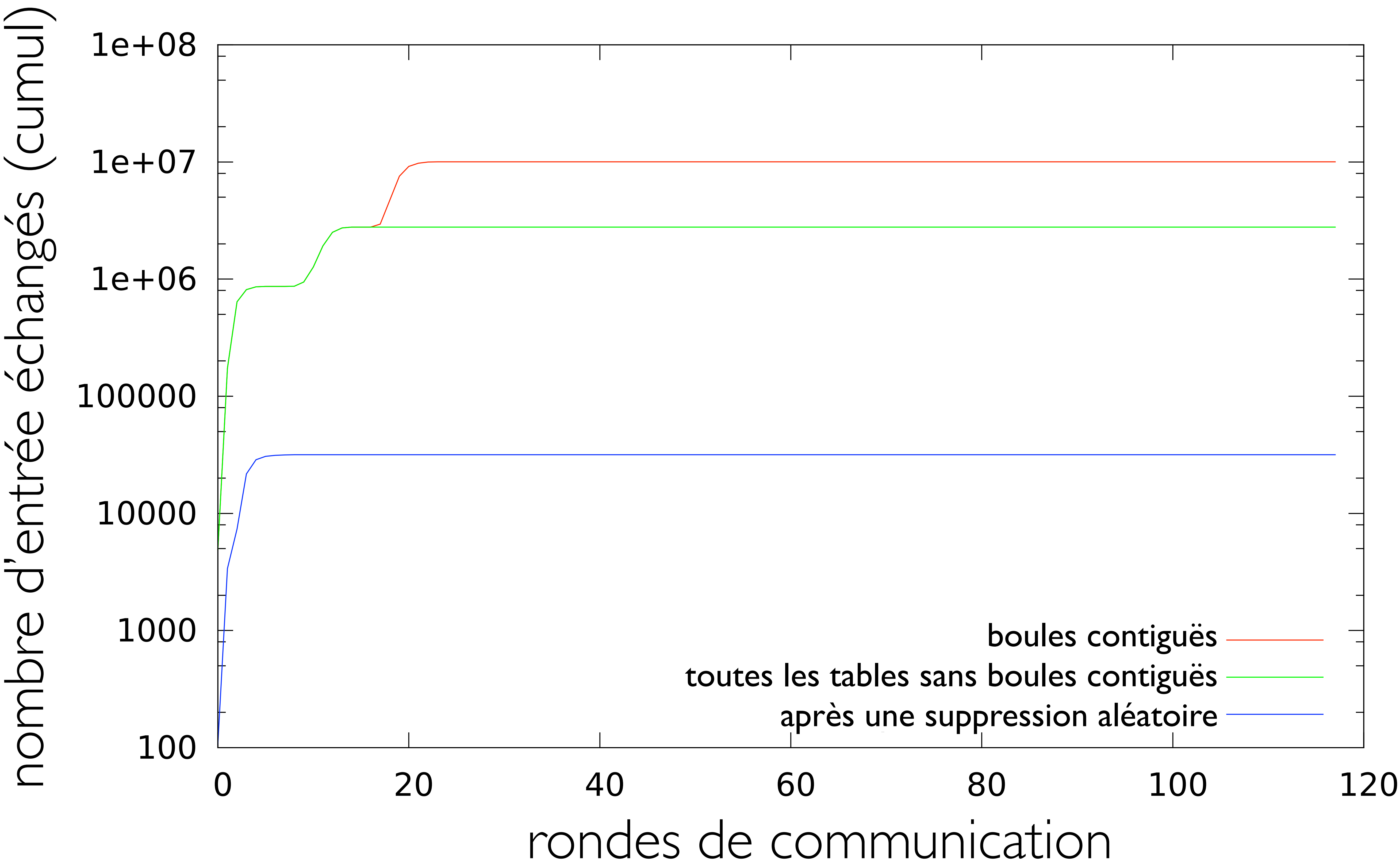
Expérimentations

Complexités, résultats expérimentaux

Algorithm	**Initialisation**		**Deletion**
	analyse, worst case	**experiments**	(average)
Vicinity Balls	$O(m \cdot k \cdot \log k)$	$32 \cdot 10^6$	12 776
Landmarks	$O(m \cdot \frac{n}{k})$	$8.5 \cdot 10^6$	437
C(u) without contiguous	$O(nm + n \cdot \min[(k \cdot \log k), D])$	$197 \cdot 10^6$	7 206
C(u) with contiguous	$O(nm + n^2 \cdot \log k \cdot \min[(k \cdot \log k), D])$	$652 \cdot 10^6$	$652 \cdot 10^6$

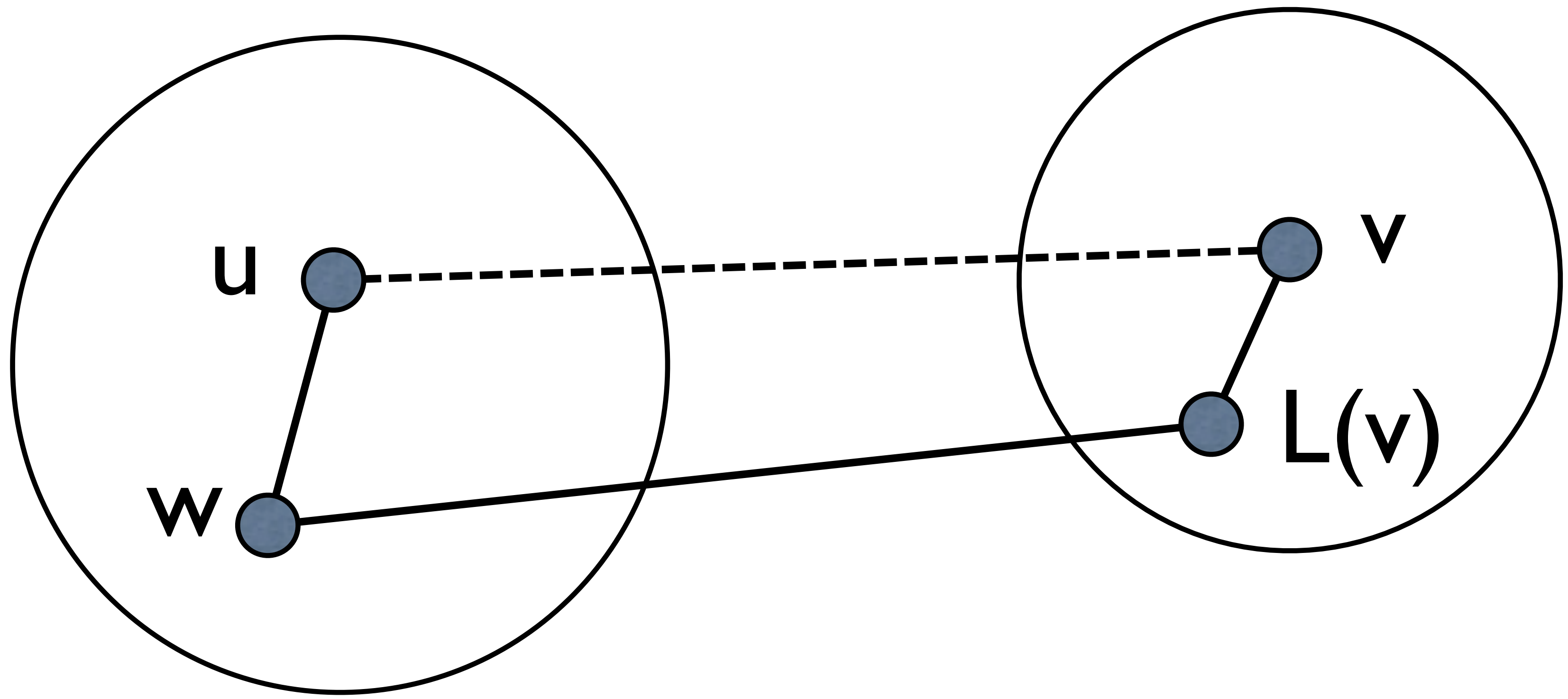
Graphe d'expérimentation : GLP, n=10K, m=26K

Nombre d’entrées échangés après une suppression ?

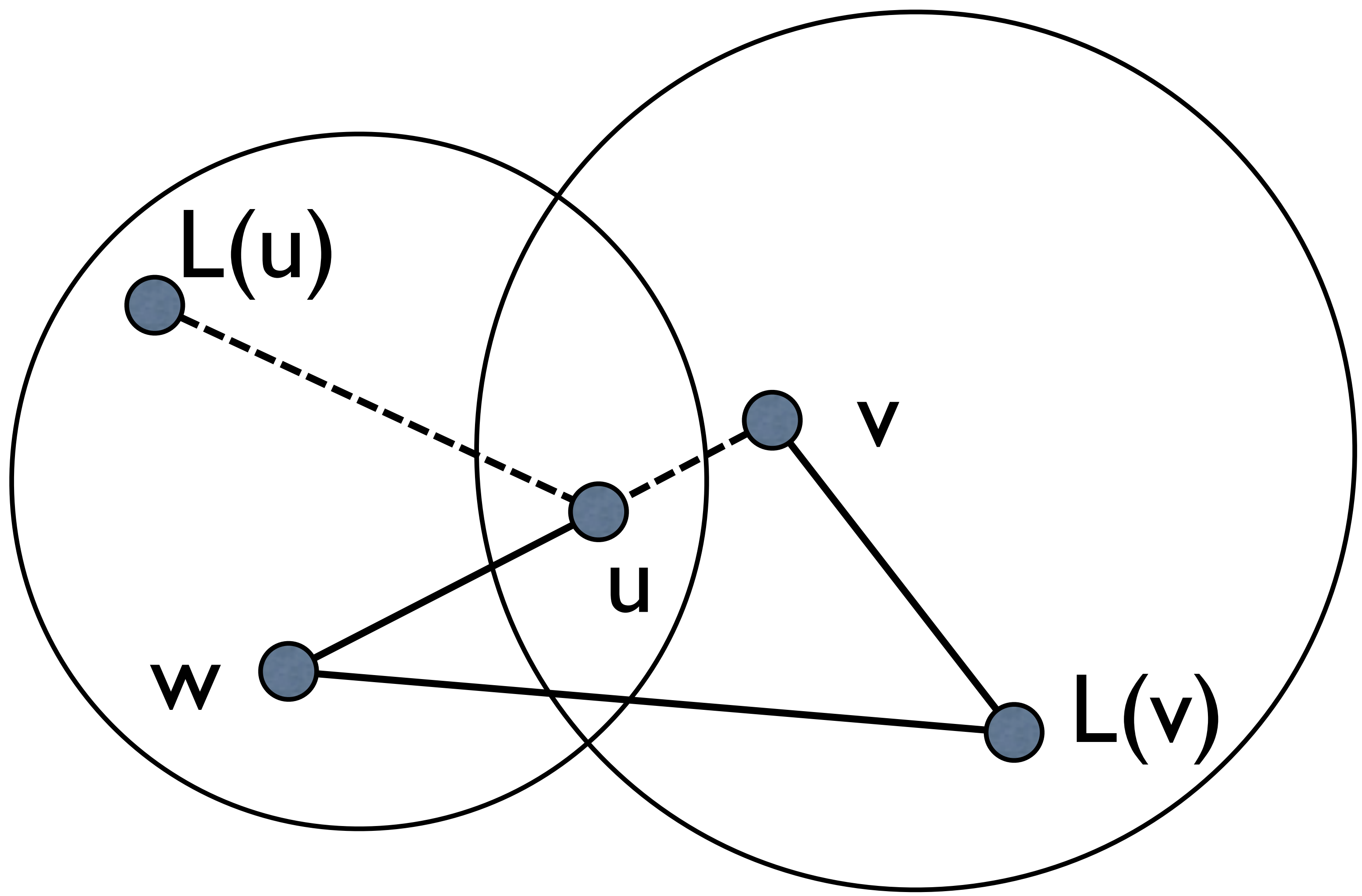


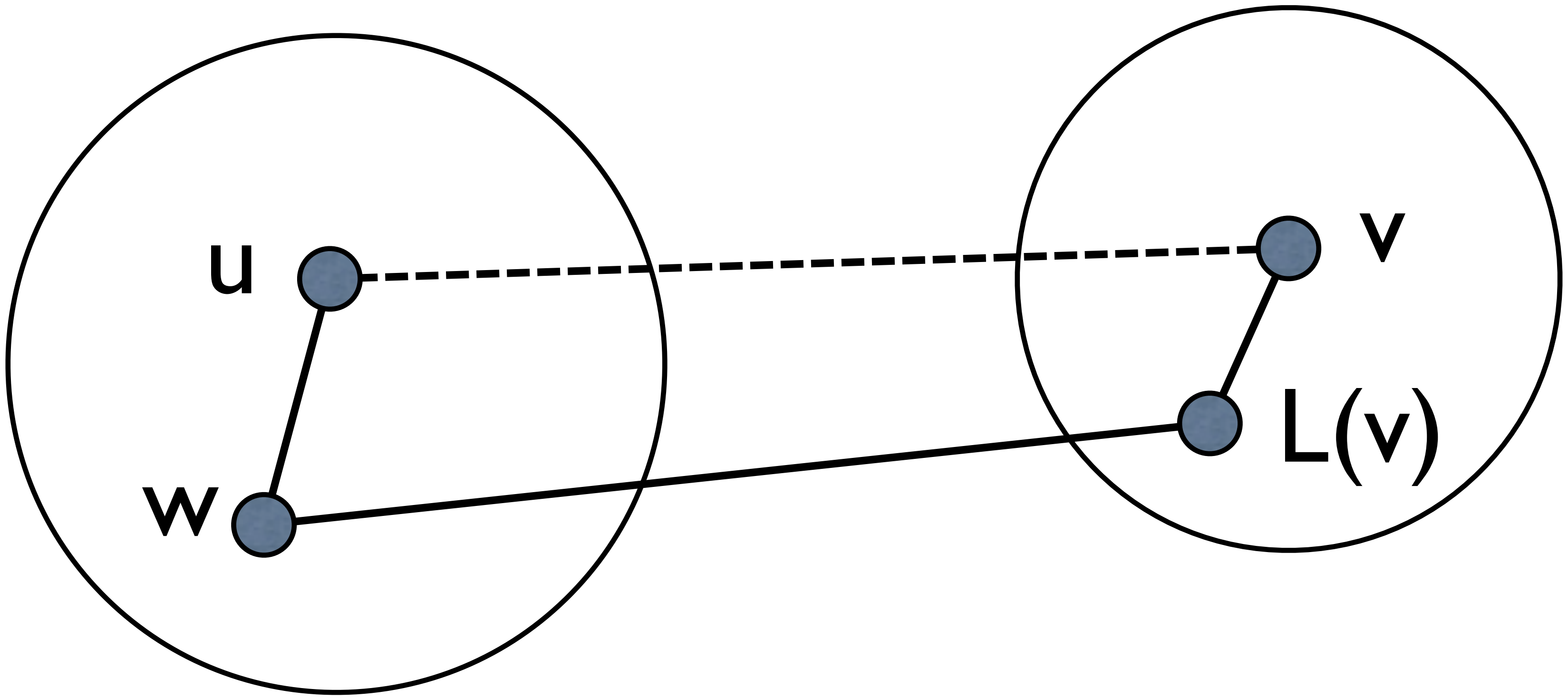
Etirement sans Boules Contiguës ?

$u \notin B(v)$



$u \in B(v)$

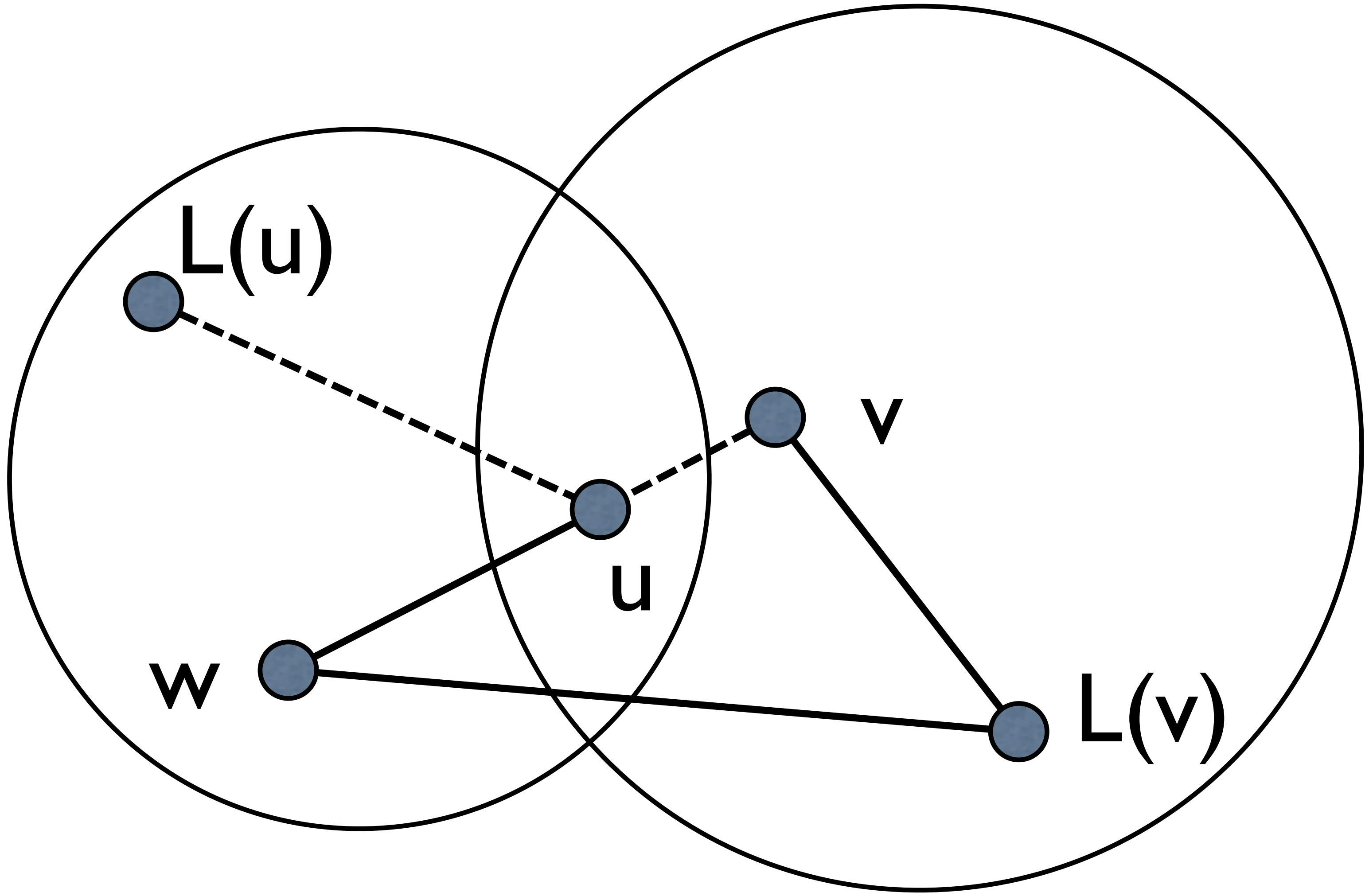




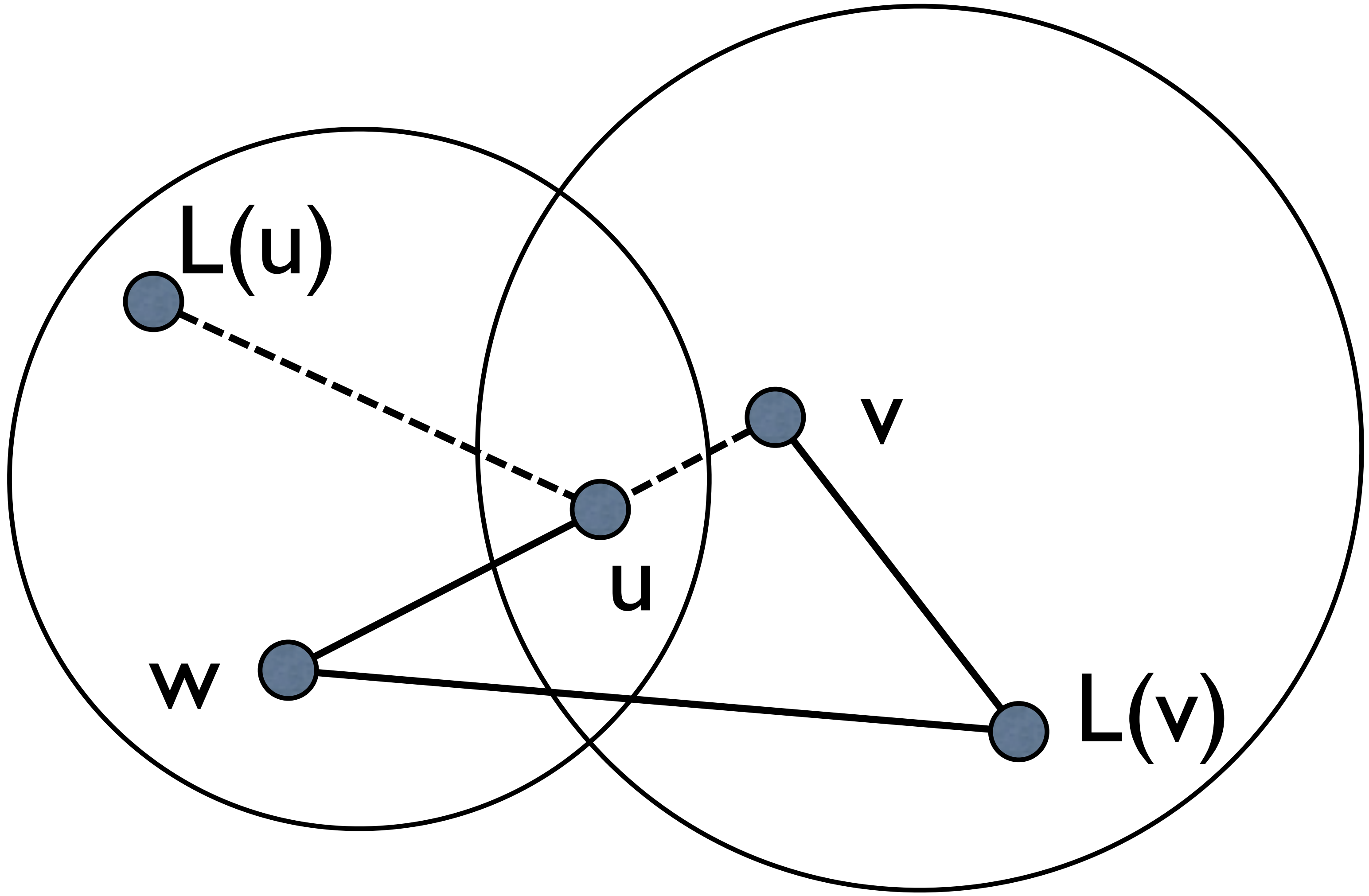
$$\mathrm{len}(u, v) \le d(u, w) + d(w, L(v)) + d(L(v), v)$$

$$\mathrm{len}(u, v) \le d(u,w) + d(w,u) + d(u,v) + d(v,L(v)) + d(L(v),v)$$

$$\mathrm{len}(u, v) \le 5d(u,v)$$

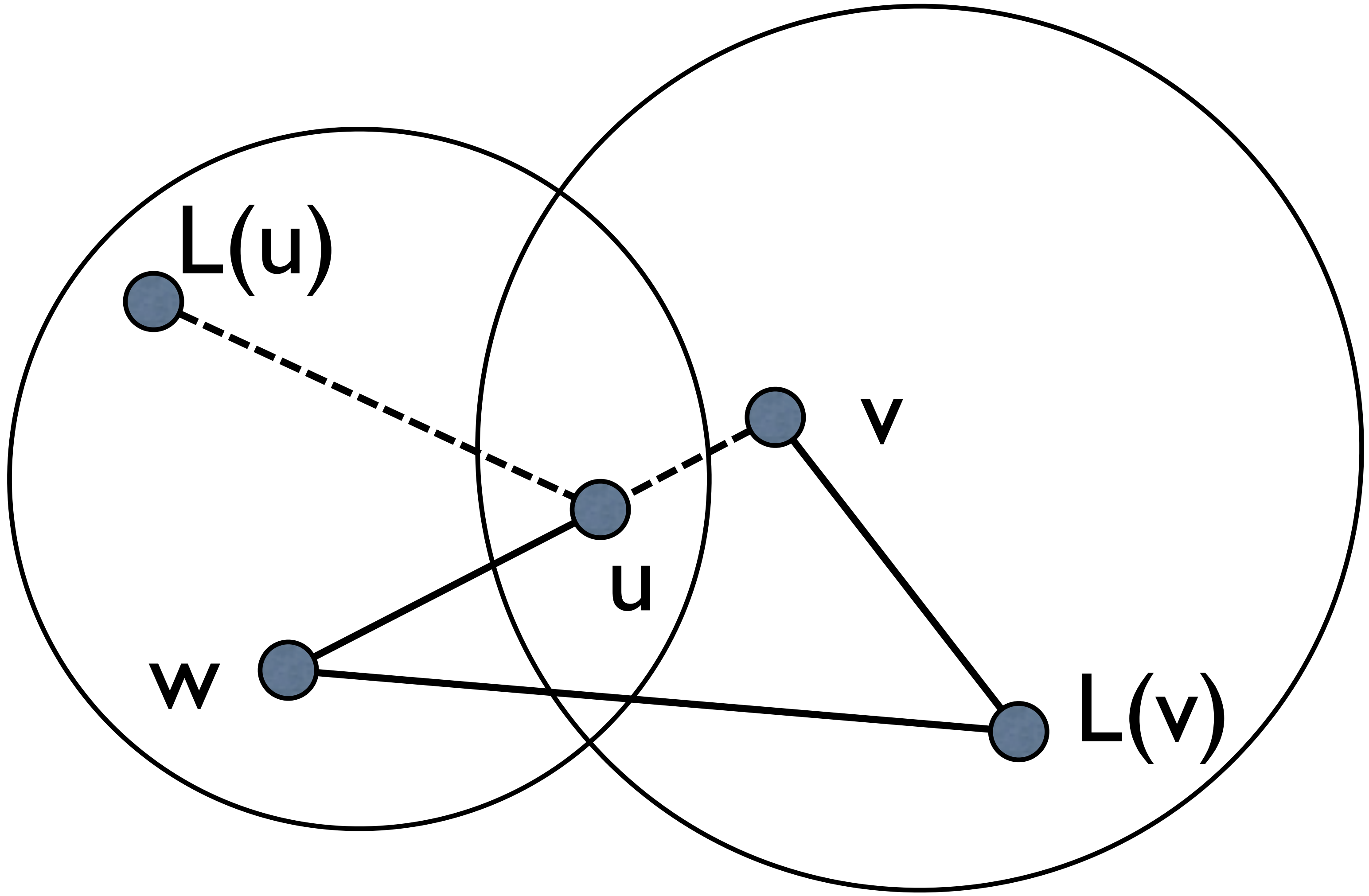


$$\mathrm{len}(u, v) \le d(u, w) + d(w, L(v)) + d(L(v), v)$$
$$\mathrm{len}(u, v) \le d(u,w) + d(w,u) + d(u,v) + d(v,L(v)) + d(L(v),v)$$



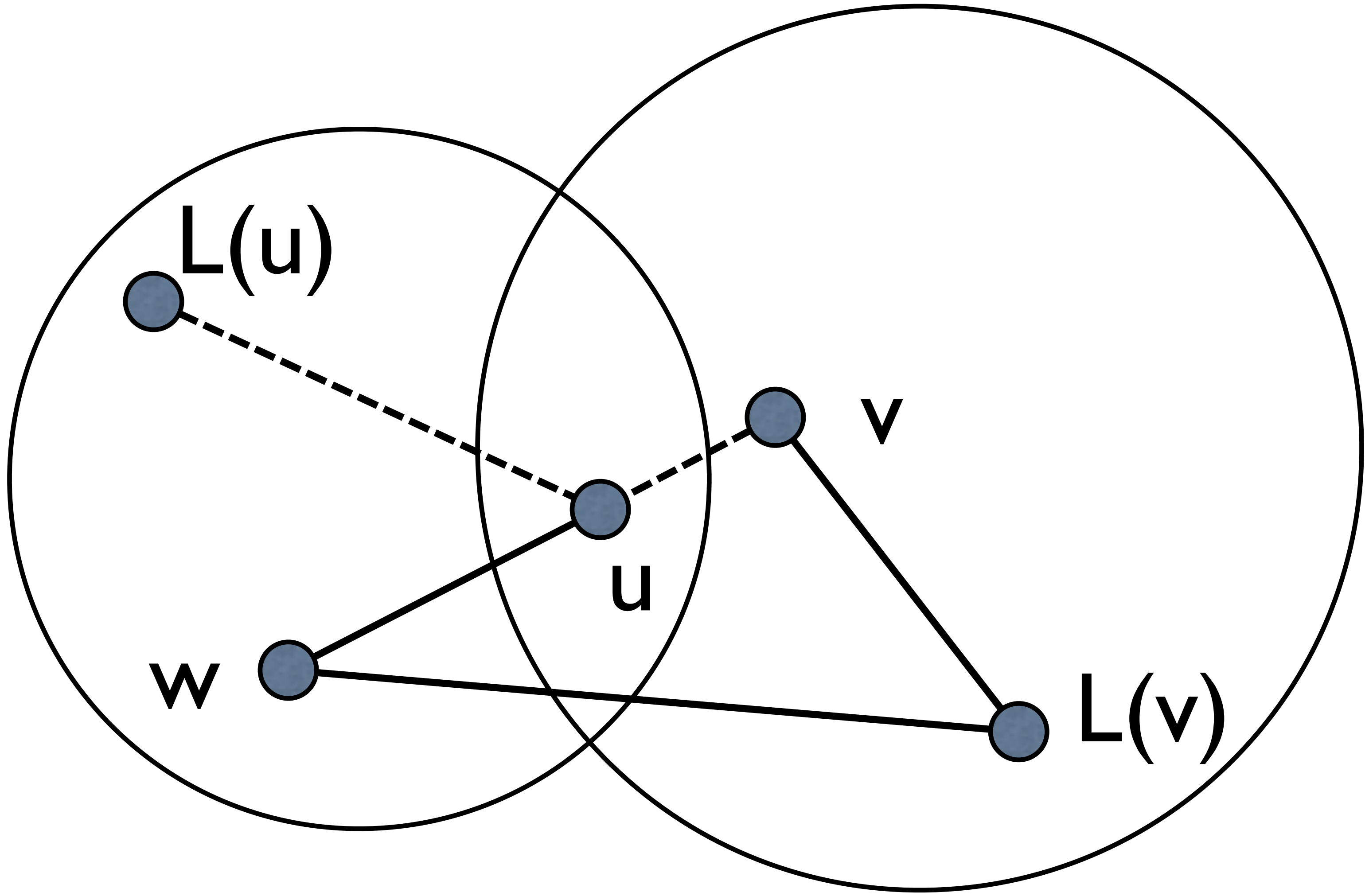
$$\text{len}(u, v) \leq d(u, w) + d(w, L(v)) + d(L(v), v)$$

$$\text{len}(u, v) \leq d(u,w) + d(w,u) + d(u,v) + \underline{d(v,L(v)) + d(L(v),v)}$$



$$\mathrm{len}(u, v) \leq d(u, w) + d(w, L(v)) + d(L(v), v)$$

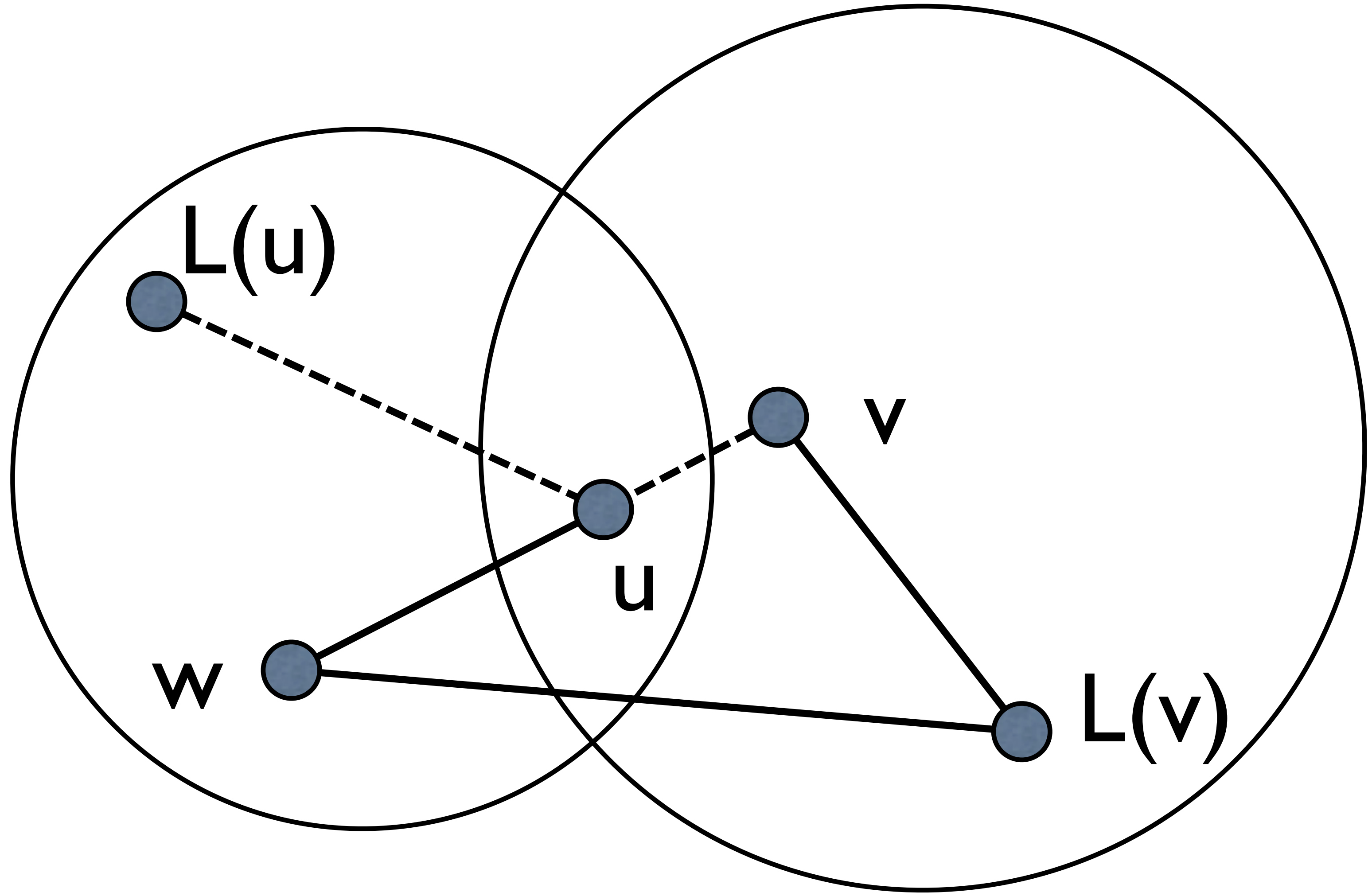
$$\mathrm{len}(u, v) \leq d(u,w) + d(w,u) + d(u,v) + d(v,L(v)) + d(L(v),v)$$



$$\mathrm{len}(u, v) \leq d(u, w) + d(w, L(v)) + d(L(v), v)$$

$$\mathrm{len}(u, v) \leq d(u,w) + d(w,u) + d(u,v) + d(v,L(v)) + d(L(v),v)$$

<u>remarque</u> : $d(L(v),v)) \leq d(L(u),v) \leq d(u,v) + d(u,L(u))$



$$\mathrm{len}(u, v) \le d(u, w) + d(w, L(v)) + d(L(v), v)$$
$$\mathrm{len}(u, v) \le d(u,w) + d(w,u) + d(u,v) + d(v,L(v)) + d(L(v),v)$$

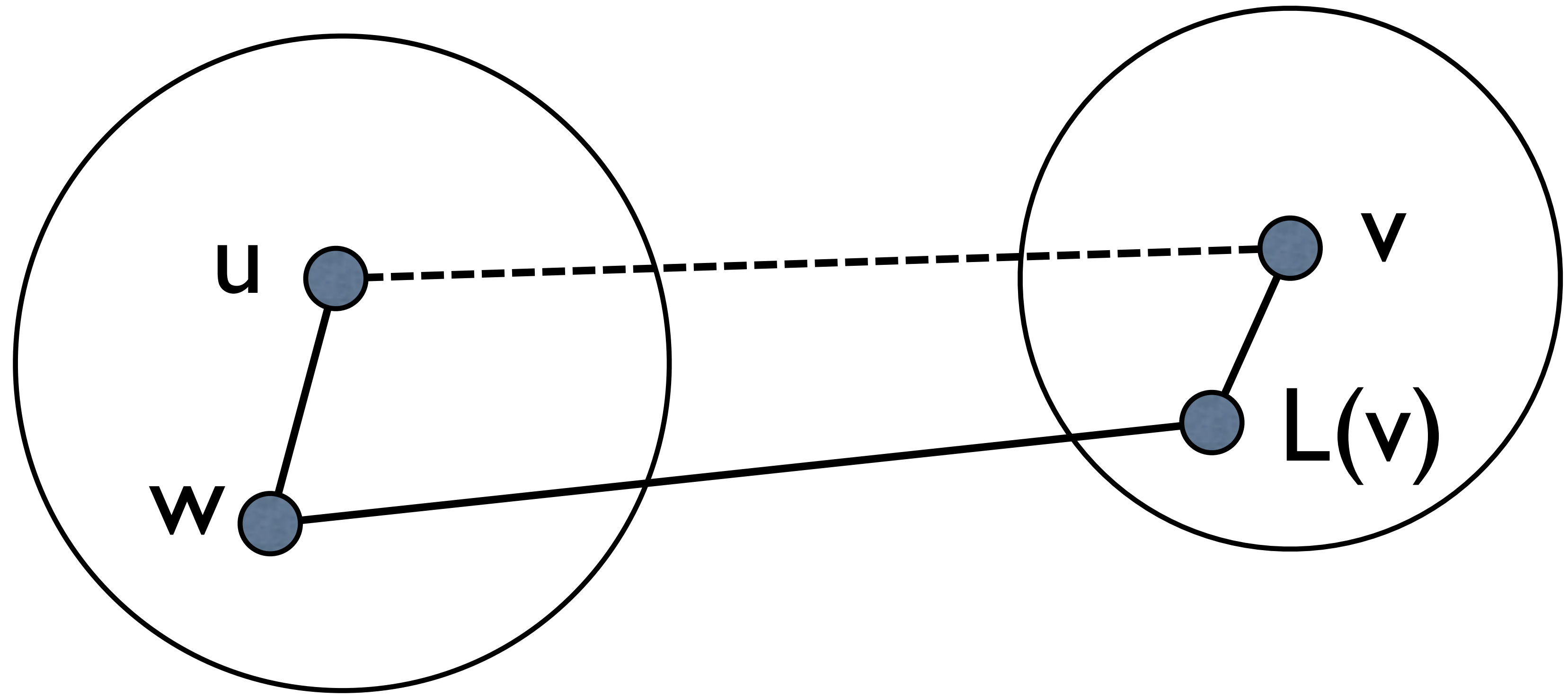
<u>remarque</u> : $d(L(v),v)) \le d(L(u),v) \le d(u,v) + d(u,L(u))$

$$\mathrm{len}(u, v) \le 2d(u,w) + d(u,v) + 2d(u,v) + 2d(u,L(u))$$
$$\mathrm{len}(u, v) \le 7d(u,v)$$

$u \notin B(v)$

Etirement : 5

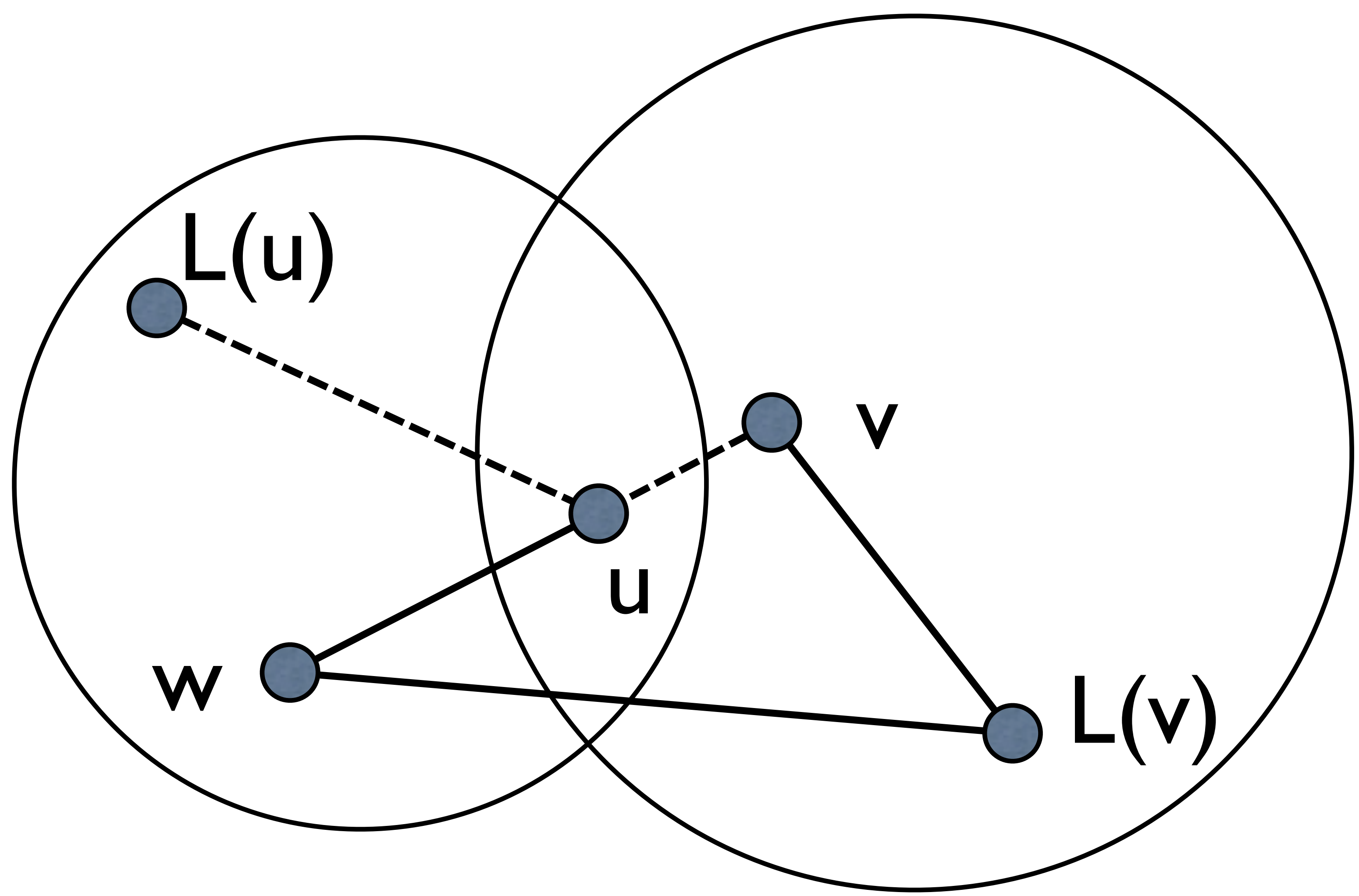
aller/retour : 5



$u \in B(v)$

Etirement : 7

aller/retour : 4



Conclusion

travaux en cours

- Une version asynchrone ?
- Tolérance aux pannes ? Aux déconnections ?