Contraction Hierarchies et les réseaux de transport en commun

Olivier Marty
Sous la direction de Laurent Viennot, IRIF, Inria

École Normale Supérieure de Cachan

Mercredi 7 septembre 2016

Université Paris Diderot

- Séminaires hebdomadaires
- Environ une rencontre hebdomadaire avec Laurent
- Présentation de mes travaux au séminaire des thésards

Plan

Plan

Contexte

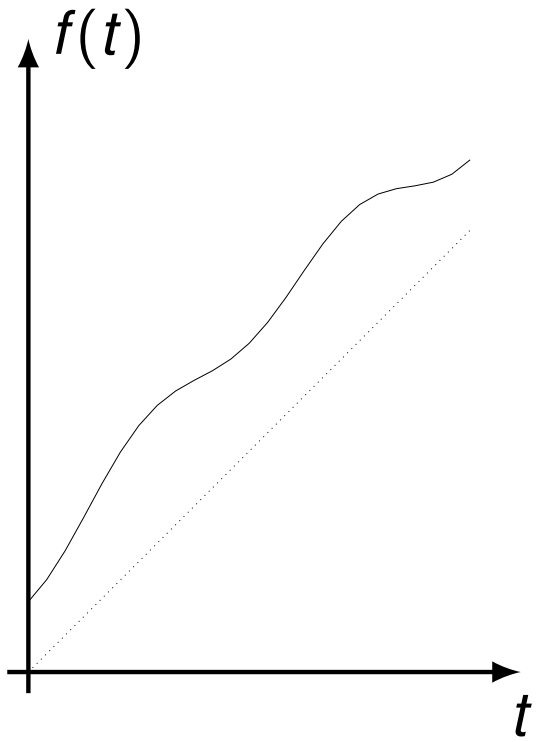
Précalcul pour accélérer des requêtes de plus courts chemins.

État de l'art :

- Contraction Hierarchies pour les réseaux routiers [Geisberger 08]
- CH avec trafic [Batz 13] (fonctions linéaires par morceaux)
- CH pour les trains [Geisberger 09] (temps de transfert approximatifs)
- Transports en commun sans précalcul [Delling 15] (pas de time profile)
- Transports en commun avec précalcul [Bast 10] (300h et au plus deux changements)

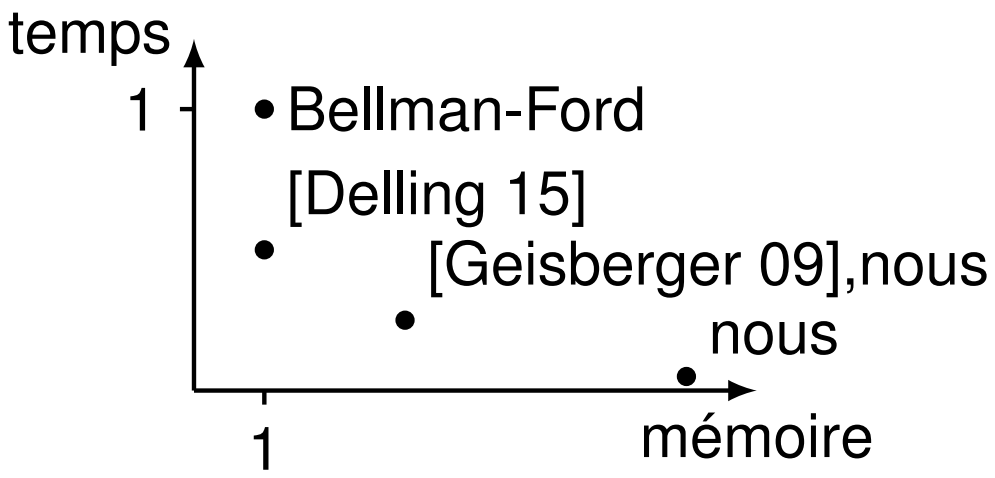
Notre but

Calculer les *time profiles* dans un réseau de *transport en commun*.



Plan

Nos résultats

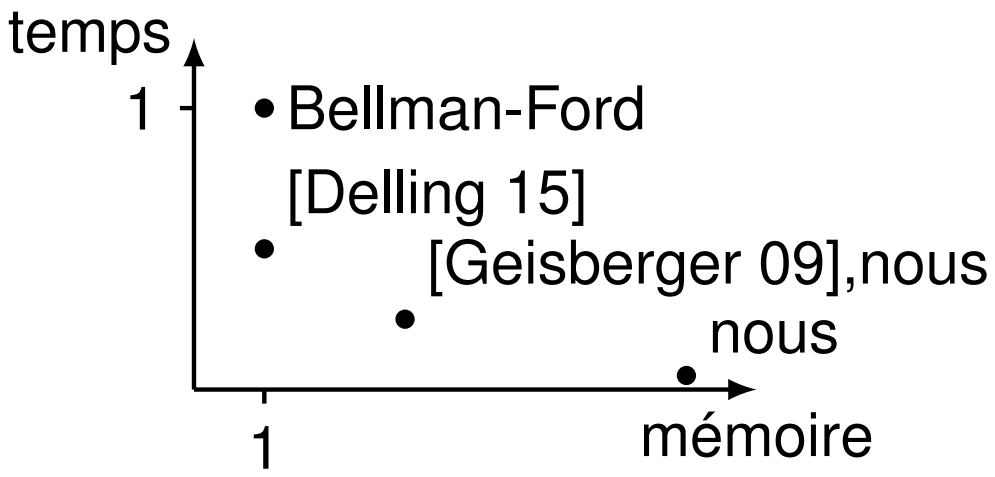


Calculer les *time profiles* sur un graphe de transport en commun avec

speed-up requête: $\times 40$

mémoire prétraitement: $\times 20 - 150$

Nos résultats



Calculer les *time profiles* sur un graphe de transport en commun avec
- speed-up requête: $\times 40$
- mémoire prétraitement: $\times 20 - 150$

Time profile labelling :
- speed-up requête: $\times 20000$
- mémoire prétraitement: $\times 500$

Plan

Idées

1. L'heuristique [Schild 15] pour trouver des petits séparateurs fonctionne sur ces graphes
2. Permet CH sans l'heuristique des raccourcis inutiles
3. Permet time profile labelling

Plan

Contraction Hierarchies

1. Ordonner les sommets
2. Contracter les sommets du graphe un à un (créer des raccourcis entre ses voisins)
3. Requêtes: utiliser deux Dijkstras partiels

1: Ordonner les sommets

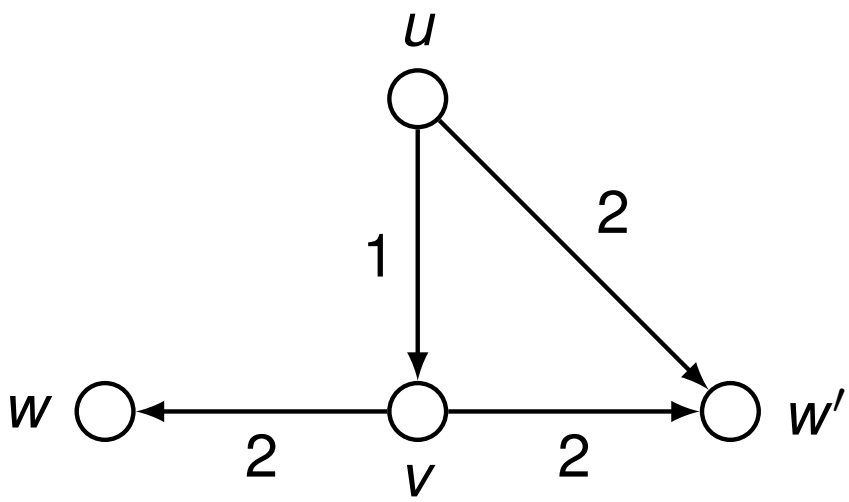
Heuristiques:

- Par degrés croissant
- Par nombre de raccourcis ajoutés croissant
- Uniformément

Pour chaque contraction le meilleur sommet est selectionné.

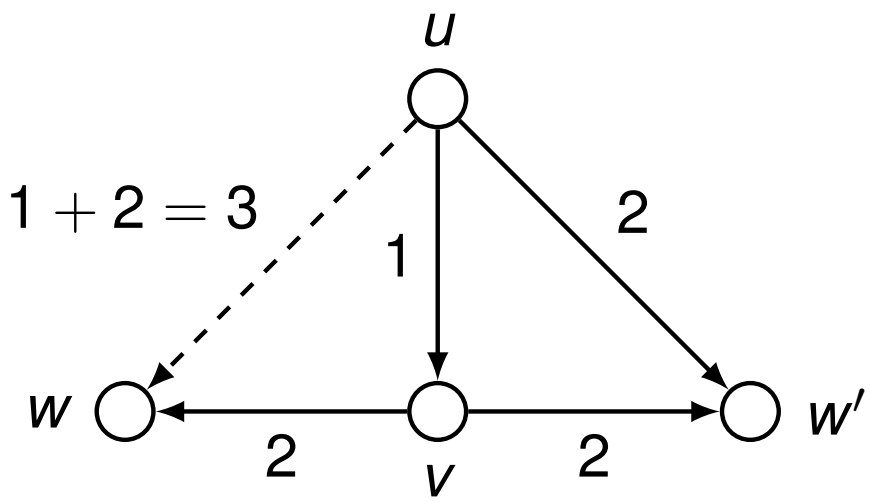
2: Contracter les sommets

But: oublier v mais préserver les distances.



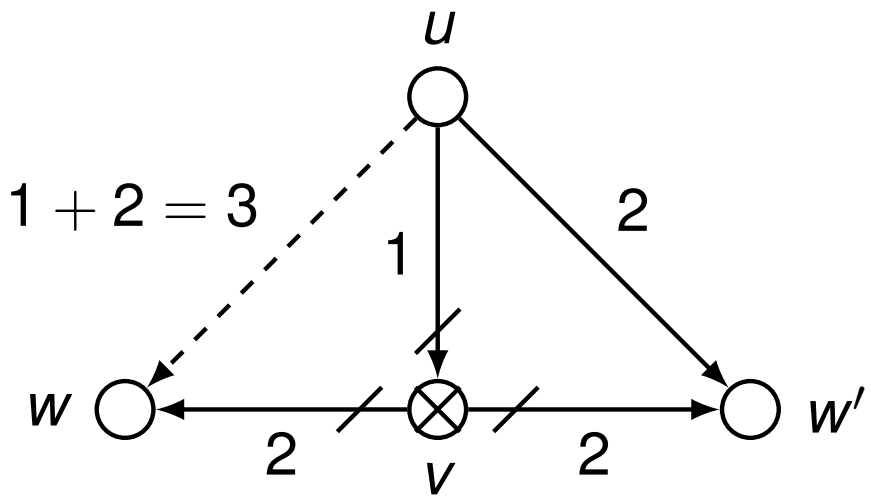
2: Contracter les sommets

But: oublier v mais préserver les distances.



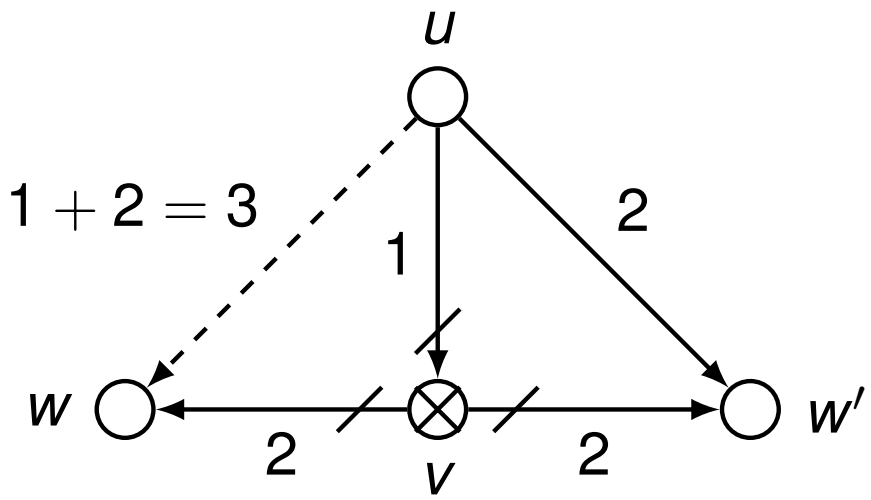
2: Contracter les sommets

But: oublier v mais préserver les distances.



2: Contracter les sommets

But: oublier v mais préserver les distances.



On obient G^+ : graphe avec les raccourcis.

3: Requête distance de s à t

On définit:

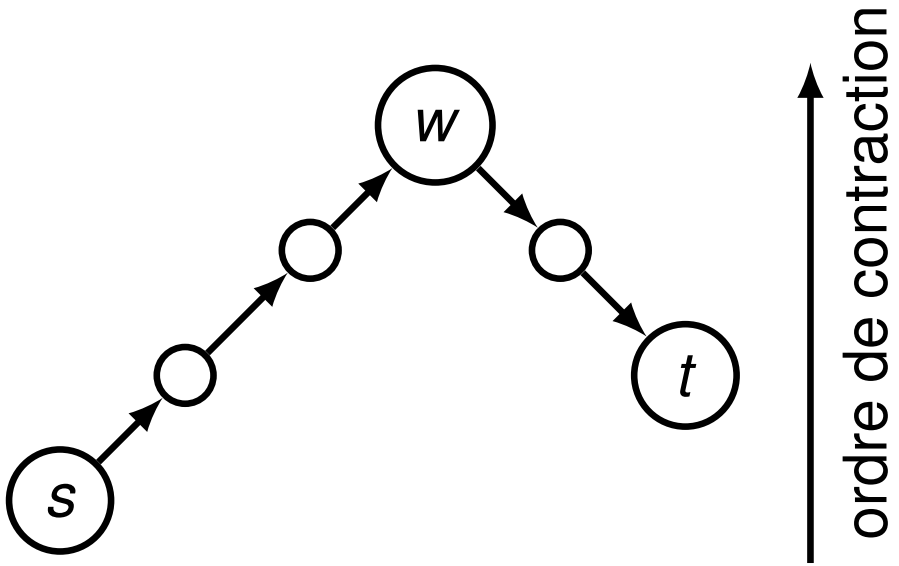
- $G^{+\uparrow}$: graphe avec les arêtes et les raccourcis montant (vers un sommet contracté après)
- $G^{+\downarrow}$: graphe avec les arêtes descendantes

3: Requête distance de s à t

On définit:

- $G^{+\uparrow}$: graphe avec les arêtes et les raccourcis montant (vers un sommet contracté après)
- $G^{+\downarrow}$: graphe avec les arêtes descendantes

Deux dijkstras partiels : à partir de s dans $G^{+\uparrow}$, et de t dans $G^{+\downarrow}$ inversé

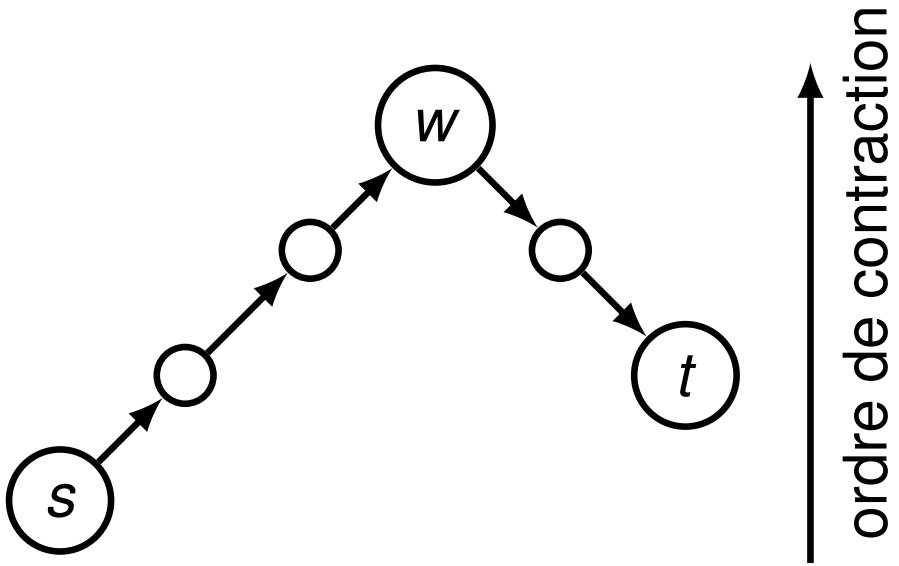


3: Requête distance de s à t

On définit:

- $G^{+\uparrow}$: graphe avec les arêtes et les raccourcis montant (vers un sommet contracté après)
- $G^{+\downarrow}$: graphe avec les arêtes descendantes

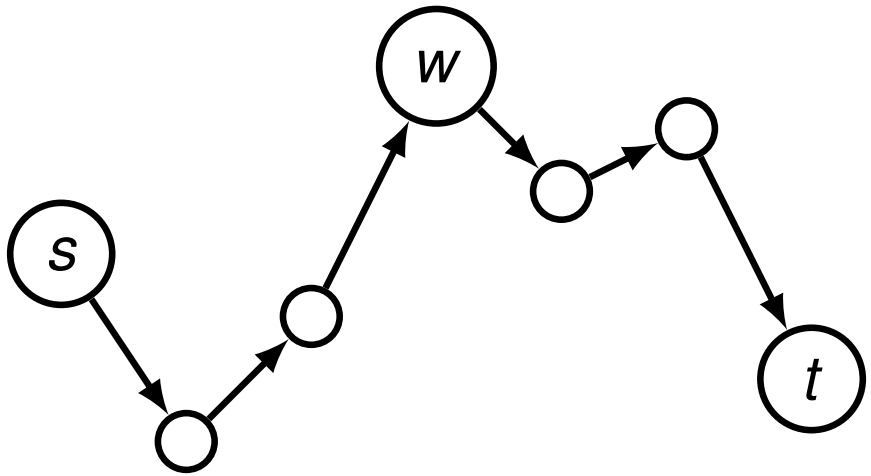
Deux dijkstras partiels : à partir de s dans $G^{+\uparrow}$, et de t dans $G^{+\downarrow}$ inversé



$$d_G(s,t) = \min_{w \in V}(\mathrm{d}_{G^{+\uparrow}}(s,w) + \mathrm{d}_{G^{+\downarrow}}(w,t))$$

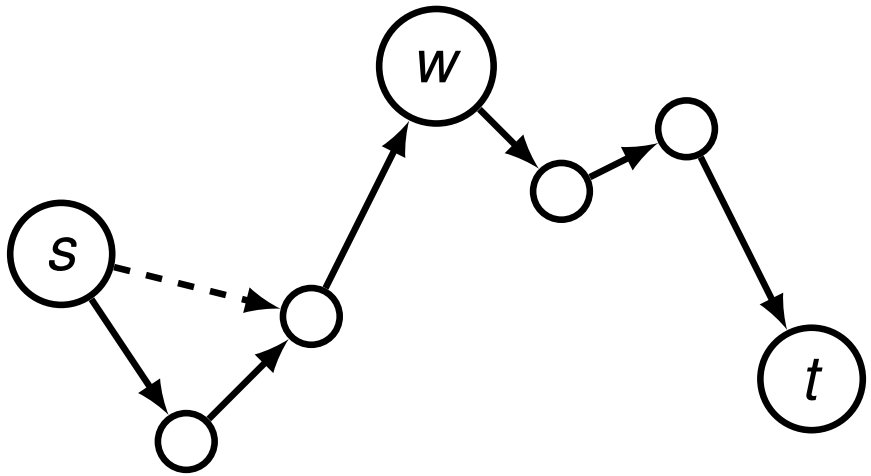
Correction

Plus court chemin dans G :



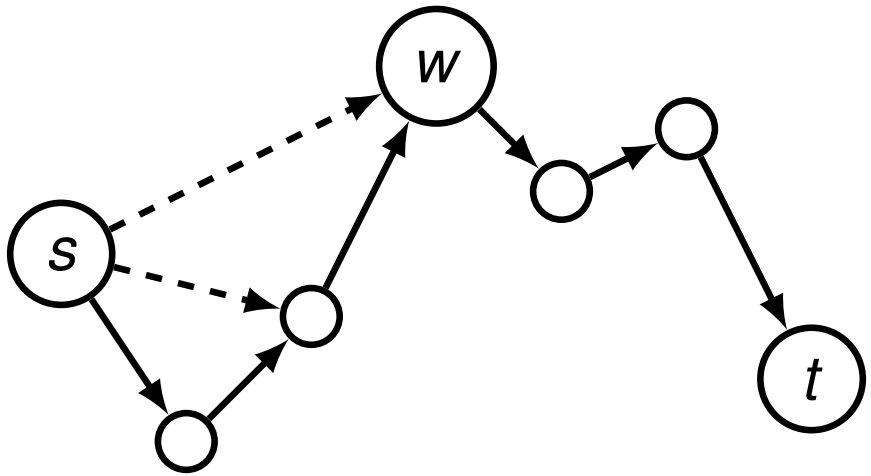
Correction

Plus court chemin dans G :



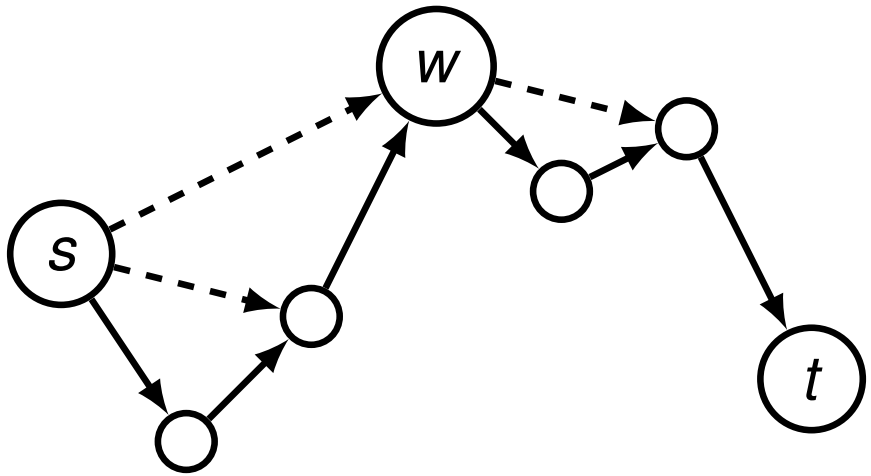
Correction

Plus court chemin dans G :



Correction

Plus court chemin dans G :



Plan

Petits séparateurs équilibrés

Définition : sous-ensemble de sommets $S \subset V$

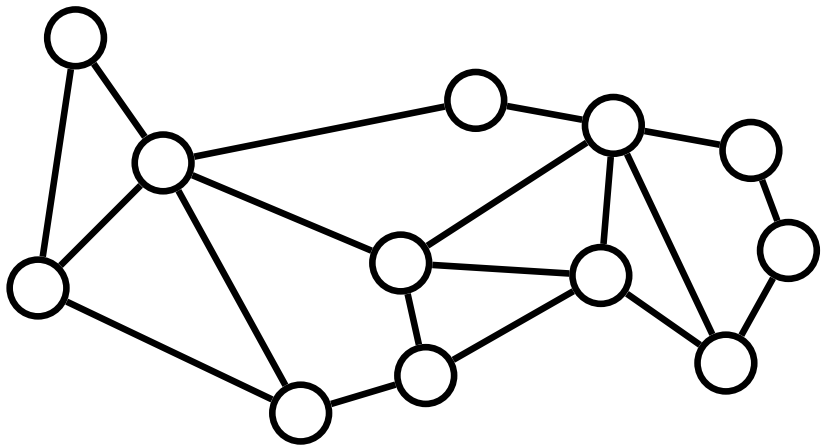
- Séparateur : sépare G en plusieurs composantes connexes si retiré
- Équilibré : les composantes sont petites (au plus $\alpha|V|$ sommets)
- Petit : $|S| \leq |V|^{\varepsilon}, \varepsilon \leq \frac{1}{2}$ (comme les graphes planaires)

Petits séparateurs équilibrés

Définition : sous-ensemble de sommets $S \subset V$

- Séparateur : sépare G en plusieurs composantes connexes si retiré
- Équilibré : les composantes sont petites (au plus $\alpha|V|$ sommets)
- Petit : $|S| \leq |V|^{\varepsilon}, \varepsilon \leq \frac{1}{2}$ (comme les graphes planaires)

Heuristique [Schild 15] :

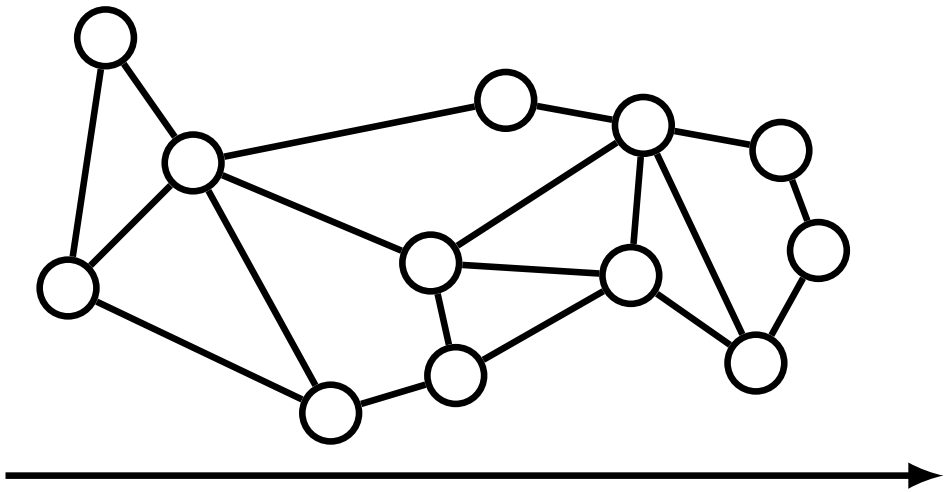


Petits séparateurs équilibrés

Définition : sous-ensemble de sommets $S \subset V$

- Séparateur : sépare G en plusieurs composantes connexes si retiré
- Équilibré : les composantes sont petites (au plus $\alpha|V|$ sommets)
- Petit : $|S| \leq |V|^{\varepsilon}, \varepsilon \leq \frac{1}{2}$ (comme les graphes planaires)

Heuristique [Schild 15] :

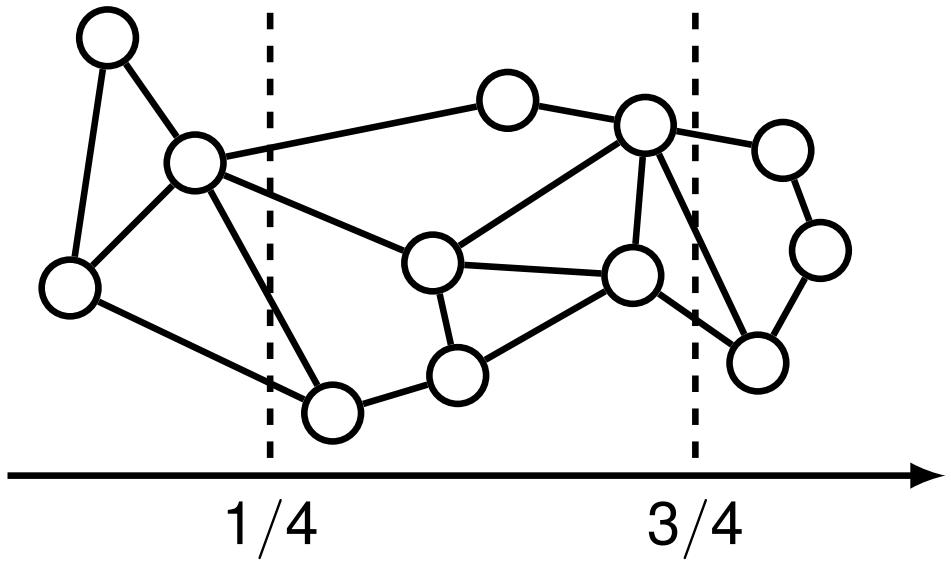


Petits séparateurs équilibrés

Définition : sous-ensemble de sommets $S \subset V$

- Séparateur : sépare G en plusieurs composantes connexes si retiré
- Équilibré : les composantes sont petites (au plus $\alpha|V|$ sommets)
- Petit : $|S| \leq |V|^{\varepsilon}, \varepsilon \leq \frac{1}{2}$ (comme les graphes planaires)

Heuristique [Schild 15] :

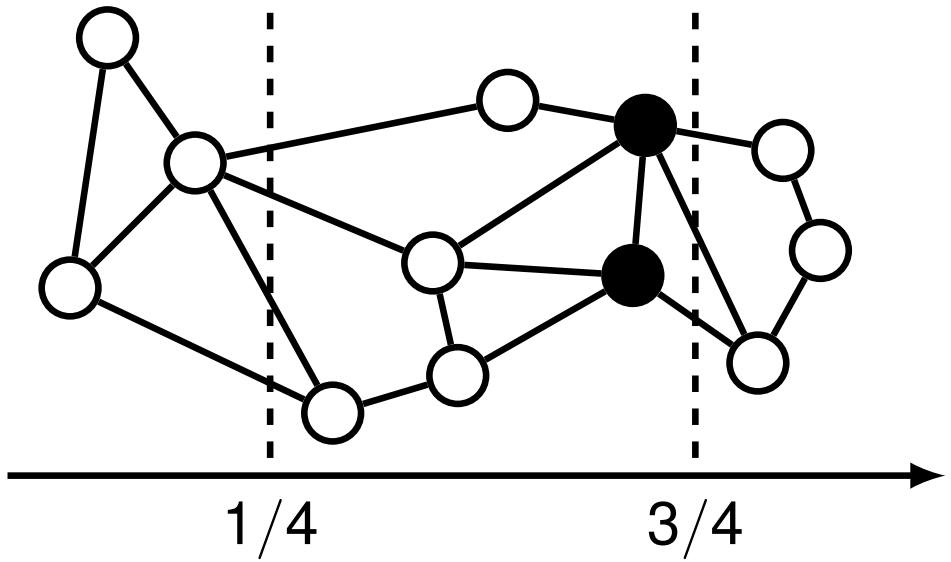


Petits séparateurs équilibrés

Définition : sous-ensemble de sommets $S \subset V$

- Séparateur : sépare G en plusieurs composantes connexes si retiré
- Équilibré : les composantes sont petites (au plus $\alpha|V|$ sommets)
- Petit : $|S| \leq |V|^{\varepsilon}, \varepsilon \leq \frac{1}{2}$ (comme les graphes planaires)

Heuristique [Schild 15] :



Ordre de contraction

Étant donné un séparateur :

1. Obtenir récursivement un ordre pour les composantes
2. Concaténer ces ordres
3. Ajouter les sommets du séparateur à la fin

Ordre de contraction

Étant donné un séparateur :

1. Obtenir récursivement un ordre pour les composantes
2. Concaténer ces ordres
3. Ajouter les sommets du séparateur à la fin

Le nombre de raccourcis ajoutés est

$$\begin{aligned} C(n) &= 2C(n/2) + O(n^{2\varepsilon}) \\ &= \underbrace{O(n) + \cdots + O(n)}_{\log n \text{ fois}} \\ &= O(n \log n) \end{aligned}$$

Plan

Poids des arêtes

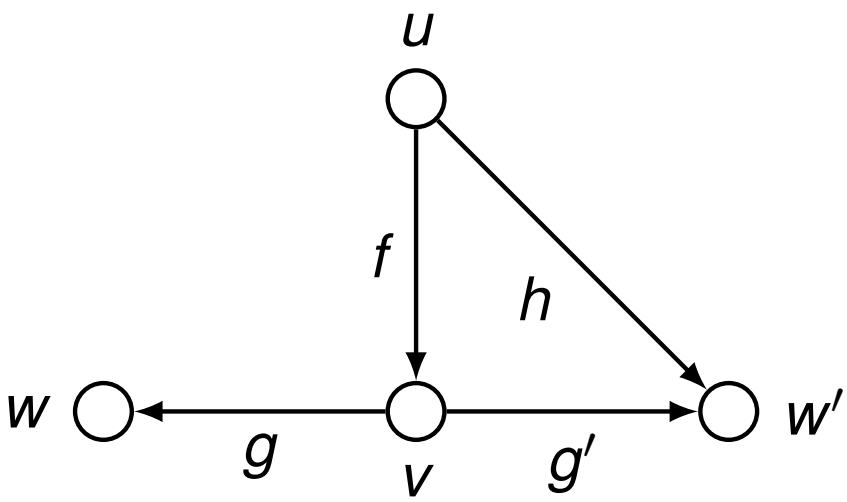
Fonctions linéaires par morceaux, associe l'heure de départ à l'heure d'arrivée

- croissante (FIFO)
- $f(x) \geq x$ (pas de retour dans le temps)

Poids des arêtes

Fonctions linéaires par morceaux, associe l'heure de départ à l'heure d'arrivée

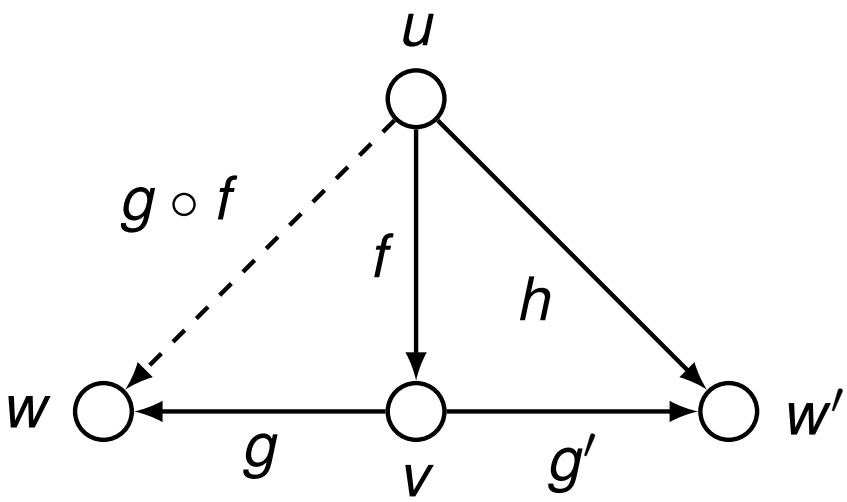
- croissante (FIFO)
- $f(x) \geq x$ (pas de retour dans le temps)



Poids des arêtes

Fonctions linéaires par morceaux, associe l'heure de départ à l'heure d'arrivée

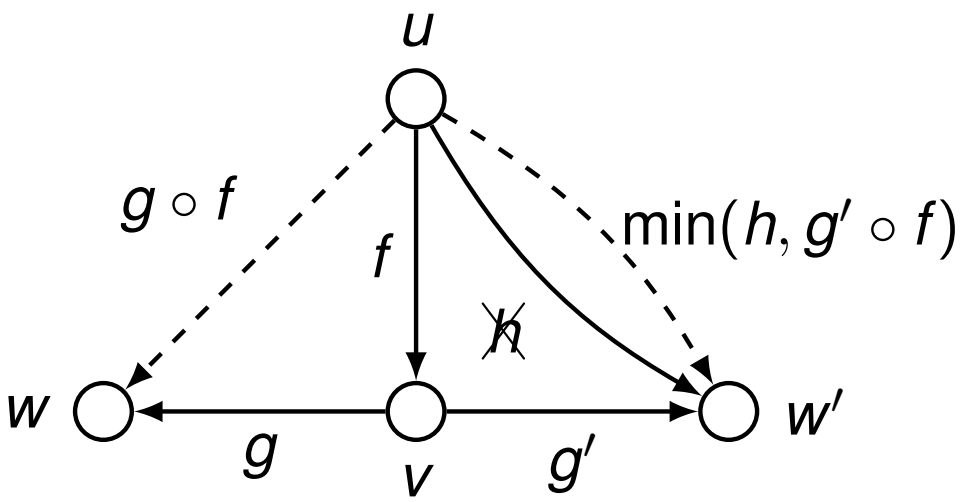
- croissante (FIFO)
- $f(x) \geq x$ (pas de retour dans le temps)



Poids des arêtes

Fonctions linéaires par morceaux, associe l'heure de départ à l'heure d'arrivée

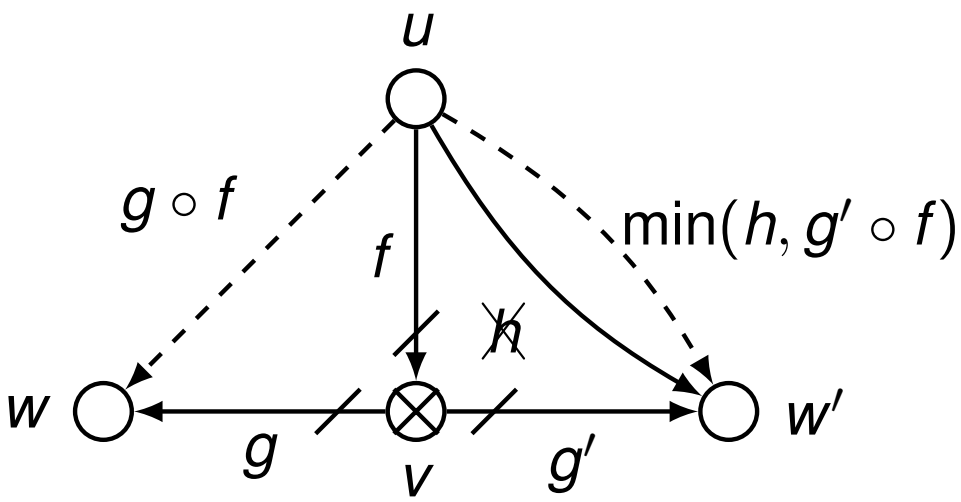
- croissante (FIFO)
- $f(x) \geq x$ (pas de retour dans le temps)



Poids des arêtes

Fonctions linéaires par morceaux, associe l'heure de départ à l'heure d'arrivée

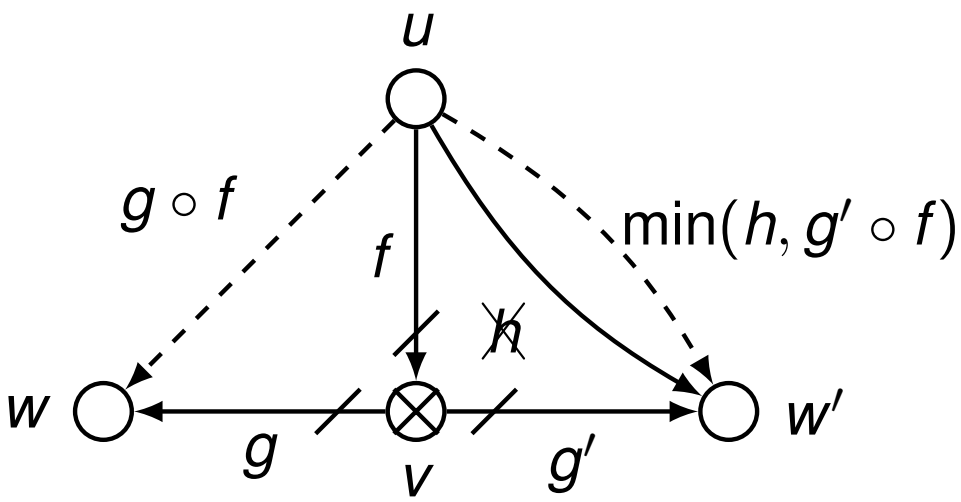
- croissante (FIFO)
- $f(x) \geq x$ (pas de retour dans le temps)



Poids des arêtes

Fonctions linéaires par morceaux, associe l'heure de départ à l'heure d'arrivée

- croissante (FIFO)
- $f(x) \geq x$ (pas de retour dans le temps)



Ensemble stable par composition et minimum.

Requêtes

Time profile entre s et t ?

Classiquement : pas de dijkstra (pas d'ordre total sur les poids) mais Bellman-Ford.

Requêtes

Time profile entre s et t ?

Classiquement : pas de dijkstra (pas d'ordre total sur les poids) mais Bellman-Ford.

Contribution : $G^{+\uparrow}$ et $G^{+\downarrow}$ sont des DAGs, programmation dynamique pour calculer $f_{G^{+\uparrow}}(s, w)$ et $f_{G^{+\downarrow}}(w, t)$, puis

$$f_G(s, t) = \min_{w \in V}(f_{G^{+\downarrow}}(w, t) \circ f_{G^{+\uparrow}}(s, w))$$

Time profile labelling

Label de u, précalculé pour tout u :

$$\{f_{G^{+\uparrow}}(u,v) \mid v \text{ accessible dans } G^{+\uparrow}\}$$

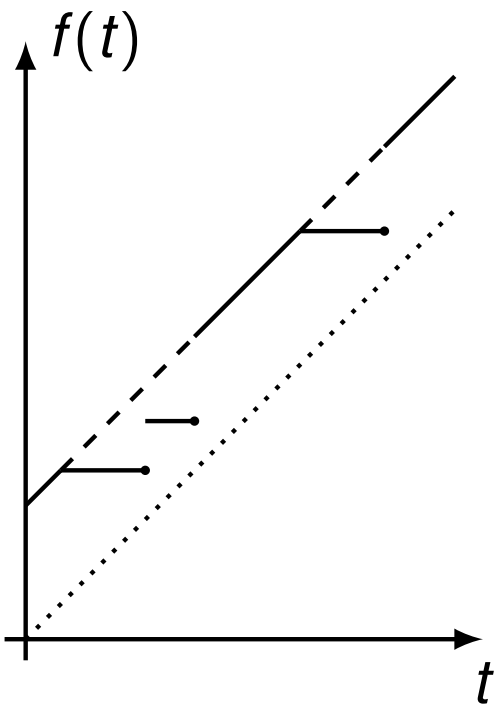
$$\cup$$

$$\{f_{G^{+\downarrow}}(u,v) \mid v \text{ coaccessible dans } G^{+\downarrow}\}$$

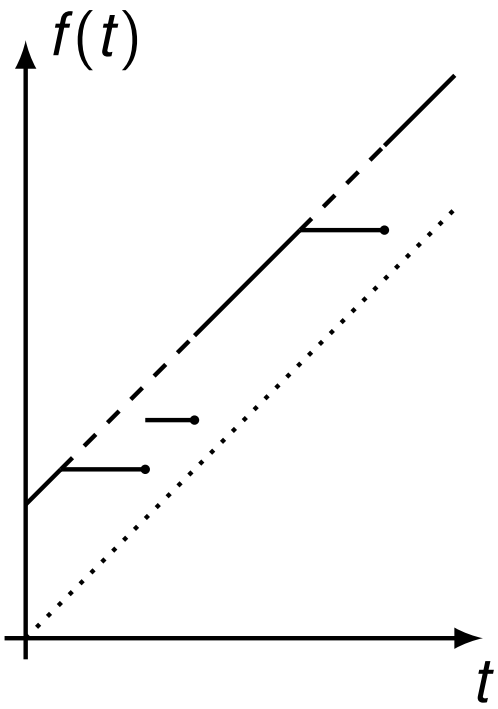
Les labels de s et de t donnent le time profile de s à t.

Plan

Walk or bus profiles



Walk or bus profiles



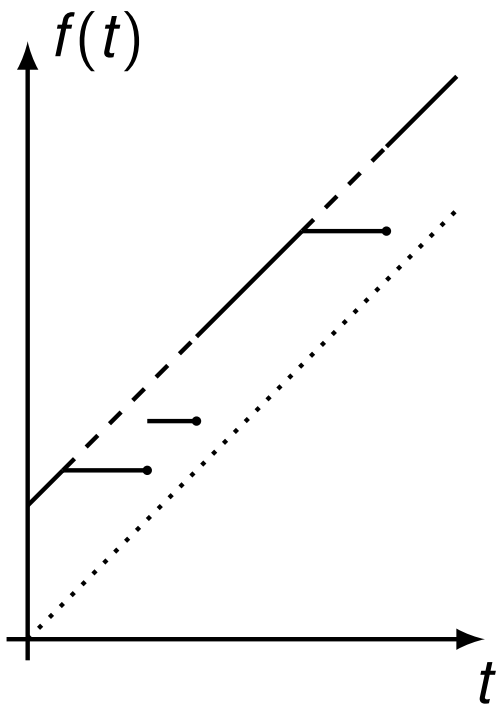
Walk profile : entier w ou $+\infty$

$$t \mapsto t + w$$

Bus profile : tableau trié C de connections

$$t \mapsto \min\{a \mid \exists (d, a) \in C, d \geq t\}$$

Walk or bus profiles



Walk profile : entier w ou $+\infty$

$$t \mapsto t + w$$

Bus profile : tableau trié C de connections

$$t \mapsto \min\{a \mid \exists (d, a) \in C, d \geq t\}$$

Évaluation en $O(\log |C|)$, composition et minimum en $O(|C| + |C'|)$.

Instances

Quatres instances:

- RATP (Paris)
- STIF (Île-de-France)
- VBB (Berlin)
- RATP + OpenStreetMap

Instances

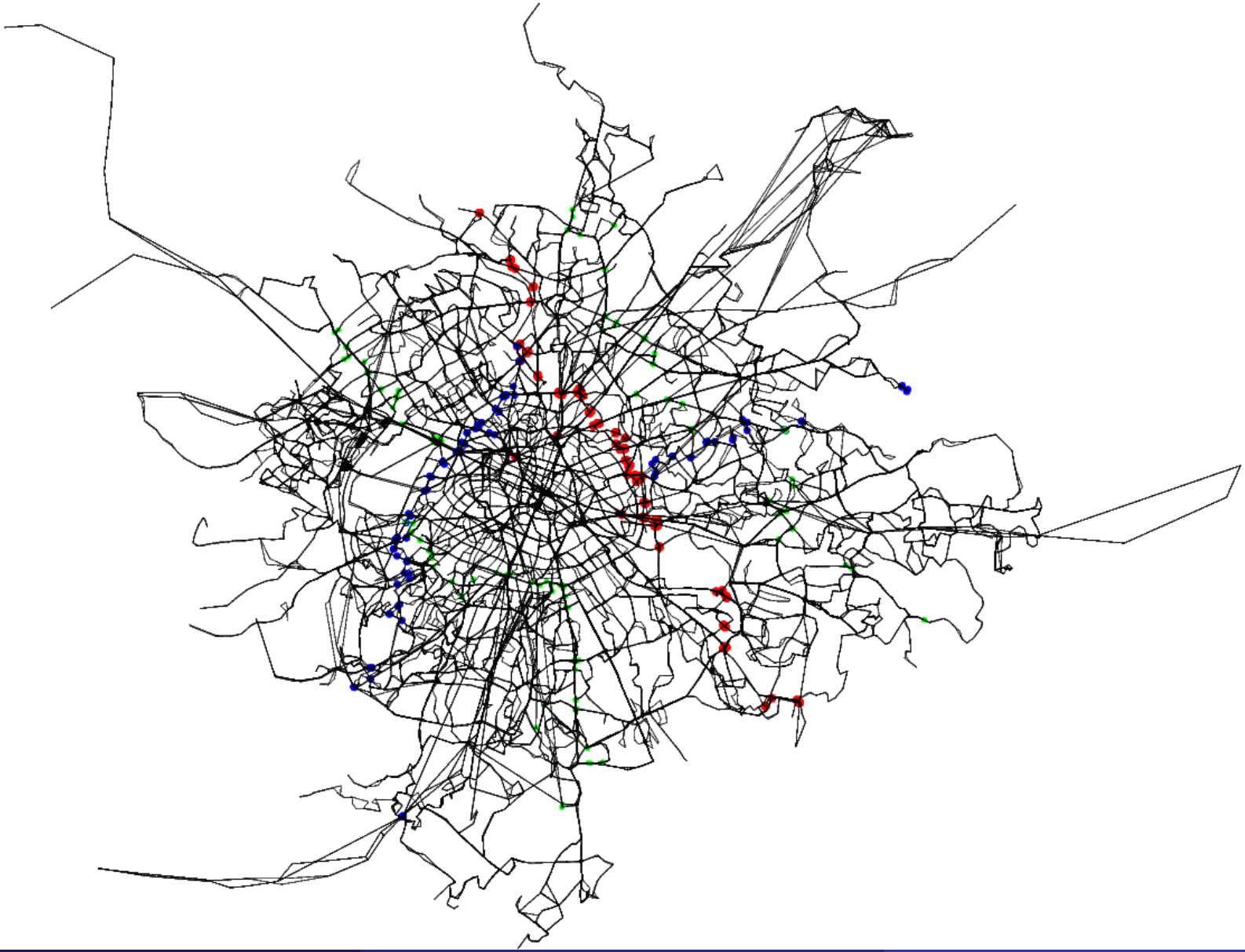
Quatres instances:

- RATP (Paris)
- STIF (Île-de-France)
- VBB (Berlin)
- RATP + OpenStreetMap

Taille des graphes :

	RATP	STIF	VBB	RATP+OSM
$\|V\|$	22 k	35 k	11 k	555 k
$\|E\|$	185 k	205 k	41 k	1405 k
degrés moyen	8	5	3	2
degrés max	128	85	21	128
$\|C\|$ (moyen / max)	60 / 574	47 / 615	42 / 1372	60 / 574
taille (Mo)	17	24	12	42

Graphe de la RATP



Prétraitement

	RATP	STIF	VBB	RATP+OSM
Graphes				
$\|V\|$	22 k	35 k	11 k	555 k
$\|E\|$	185 k	205 k	41 k	1405 k
Calcul				
separateurs	5 min 30	6 min	30 s	1 h 30
contraction (s)	2 h	30 min	30 s	30 h
Graphes contractés				
|E|	1062 k	814 k	97 k	4610 k
degrés moyen	47	23	8	34
degrés max	1029	1001	219	1792
$\|C\|$ (moyen / max)	100 / 847	70 / 1008	46 / 1372	170 / 885
size (Mo)	791	423	37	6306

Requêtes

	RATP	STIF	VBB	RATP+OSM
Queries mean/max (ms)				
BF G	55028/281829	16725/262257	799/2896	101435/395141
DP G^+	2690/10835	1044/4380	44/247	5338/19793

$\times$25 à $\times$40 en moyenne, jusqu'à $\times$65 pour certaines requêtes.

Time profile labelling

	STIF	VBB
Labels		
calcul	3h	3min
size	11Go	500Mo
requêtes mean/max (ms)	3/13	1/4

$\times 20000$

Plan

Conclusion

Contraction Hierarchies fonctionne avec des graphes de transport en commun.
Code sur `http://github.com/OlivierMarty/ch-transport/`.

Généralisation : Contraction Hierarchies pour tout graphe avec des petits séparateurs équilibrés et pondéré dans un semi-anneau sans *cycle négatif*.

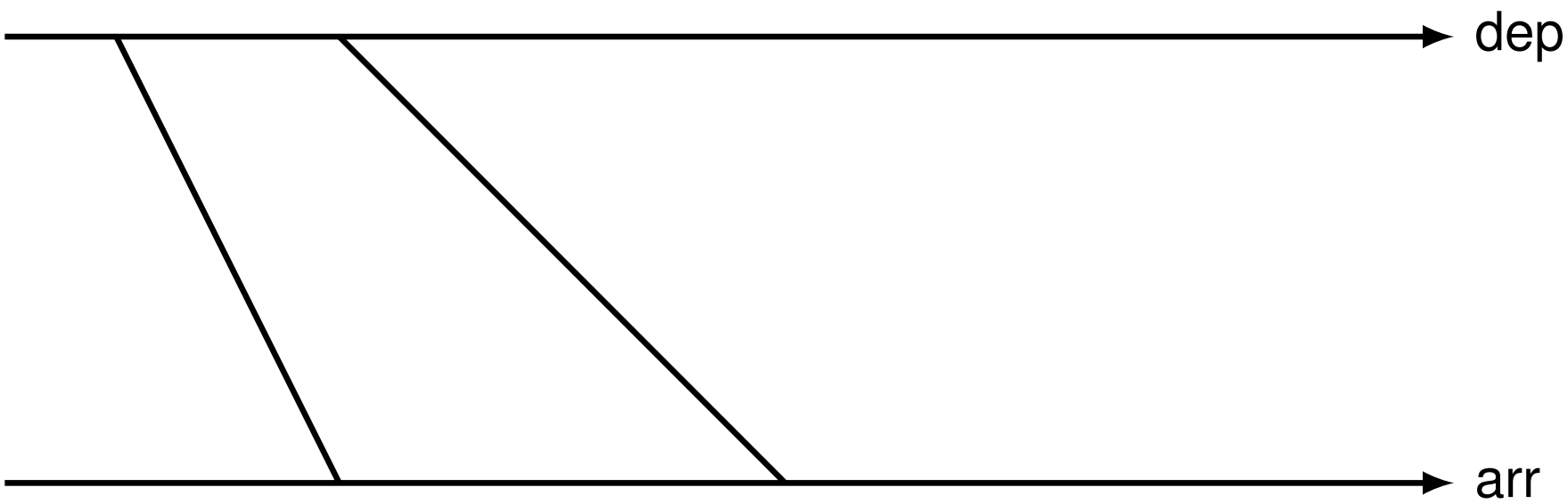
Perspectives :

- vérifier transfert pattern (peu de stations où changer)
- graphe dynamique (réutiliser les mêmes chemins ?)
- walk or bus intéressant ? (taille des étiquettes)
- labels pour RATP+OSM
- requêtes multi-critère (nombre de changements, prix, marche limité)

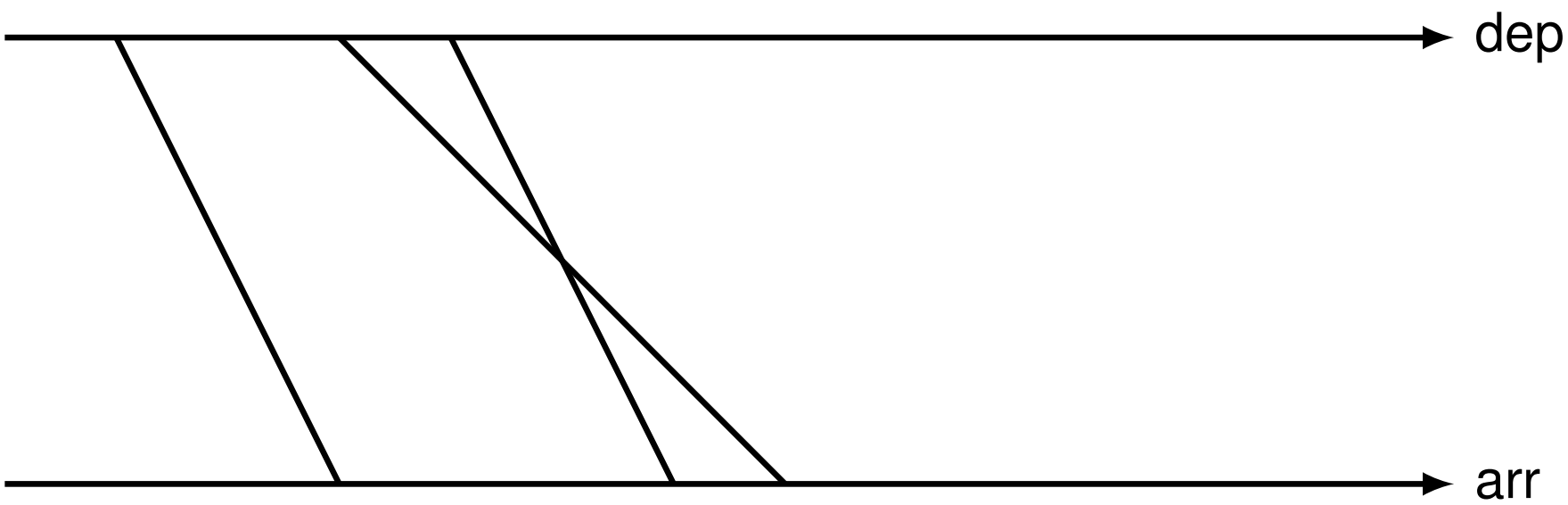
Filtrer les connexions dominées

Filtrer les connexions dominées

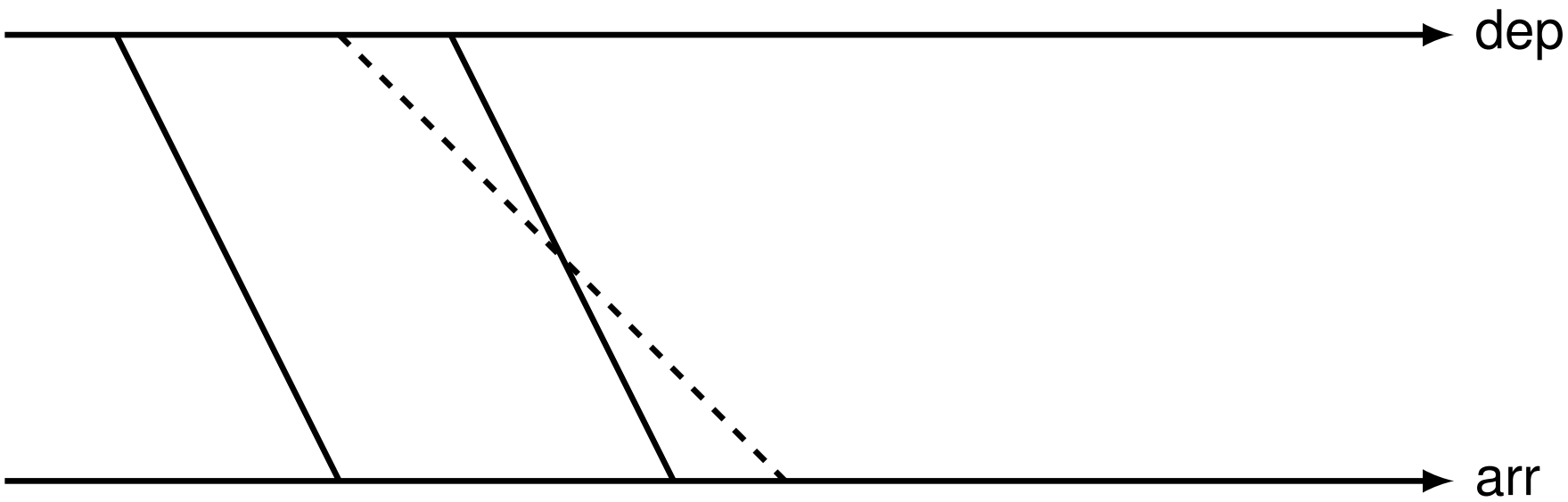
Filtrer les connexions dominées



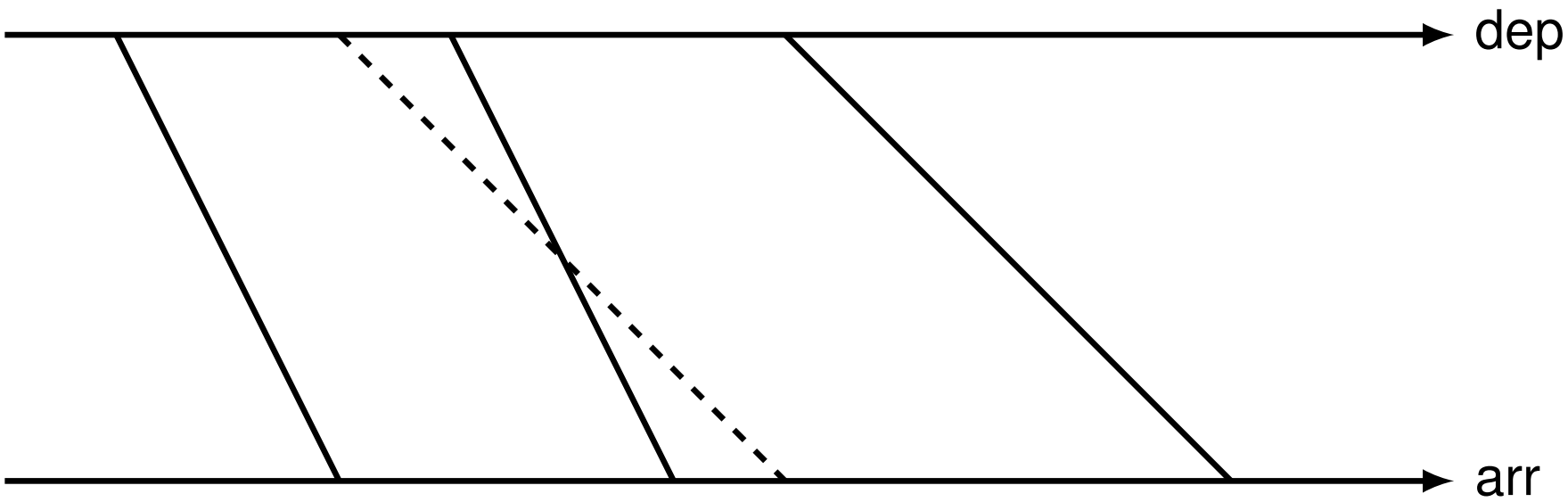
Filtrer les connexions dominées



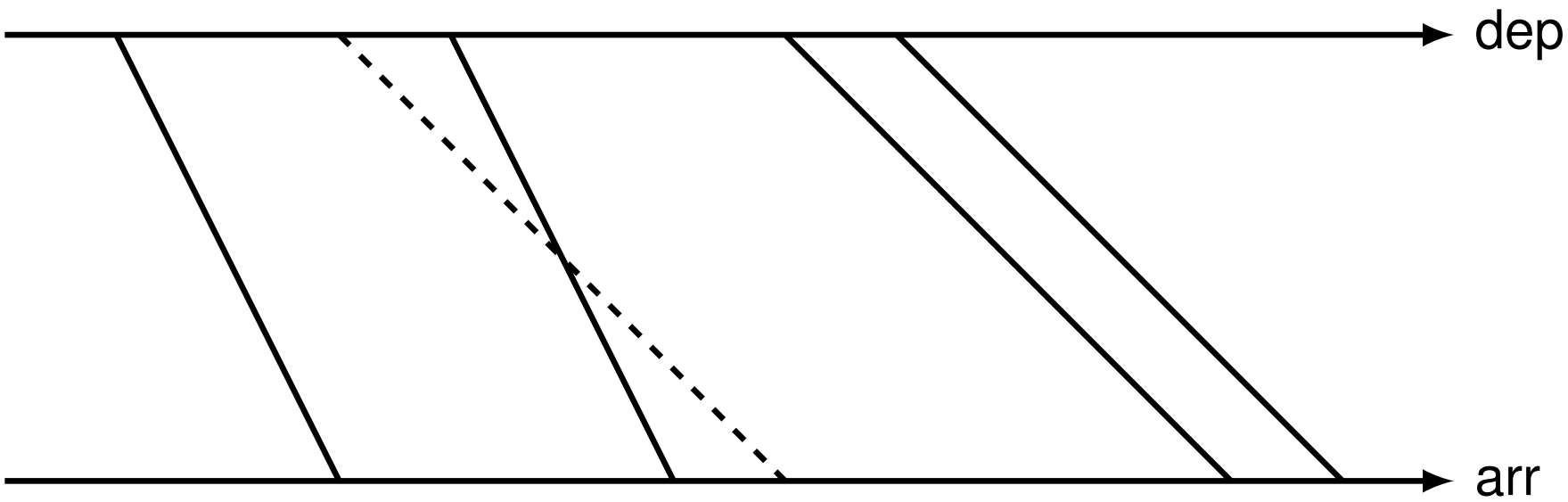
Filtrer les connexions dominées



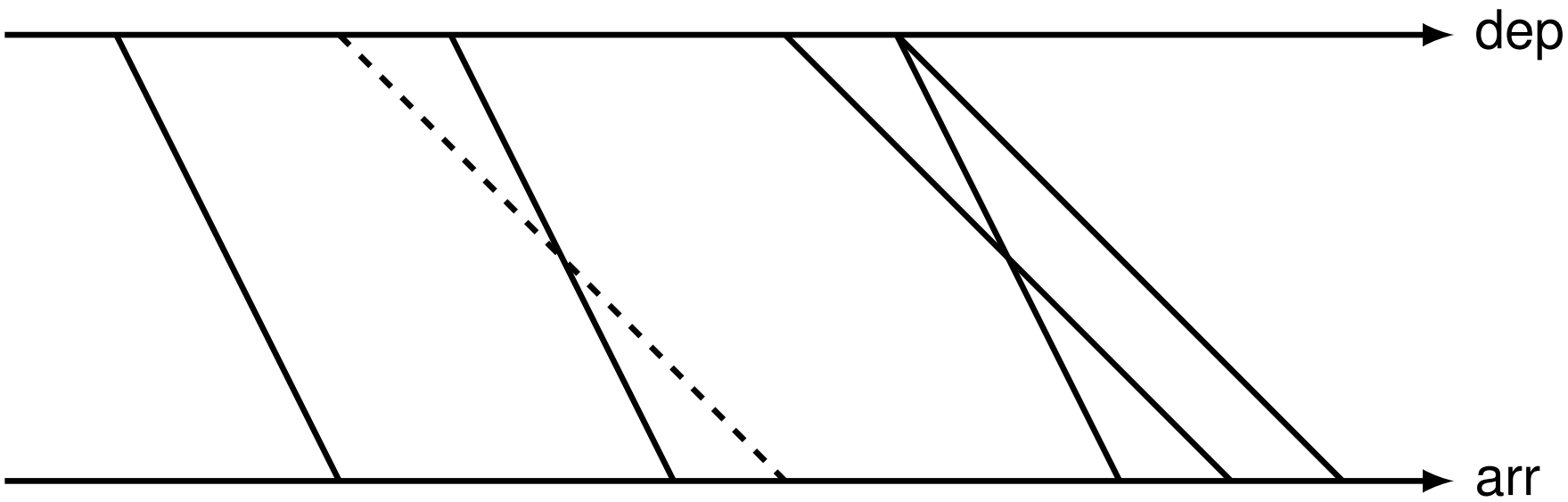
Filtrer les connexions dominées



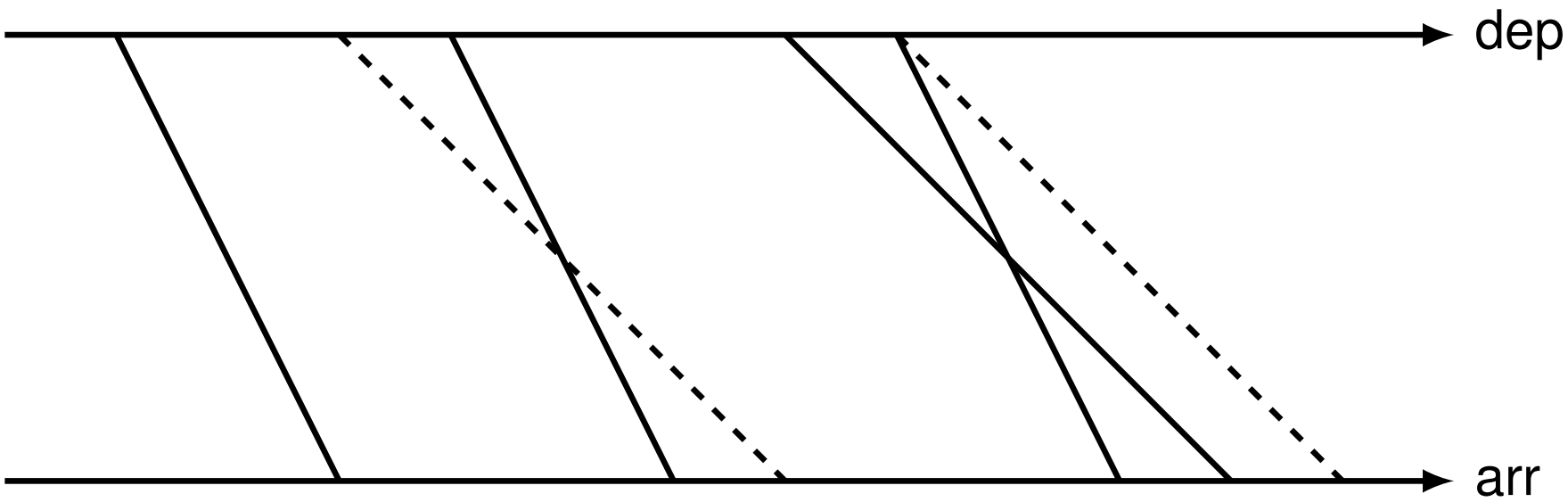
Filtrer les connexions dominées



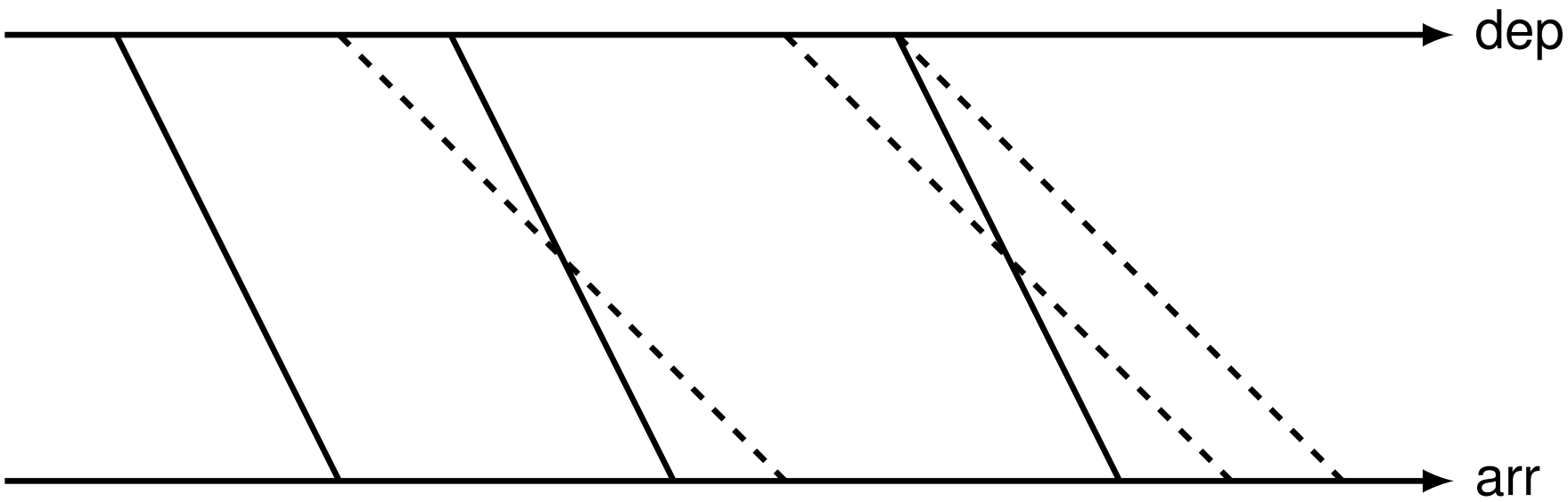
Filtrer les connexions dominées



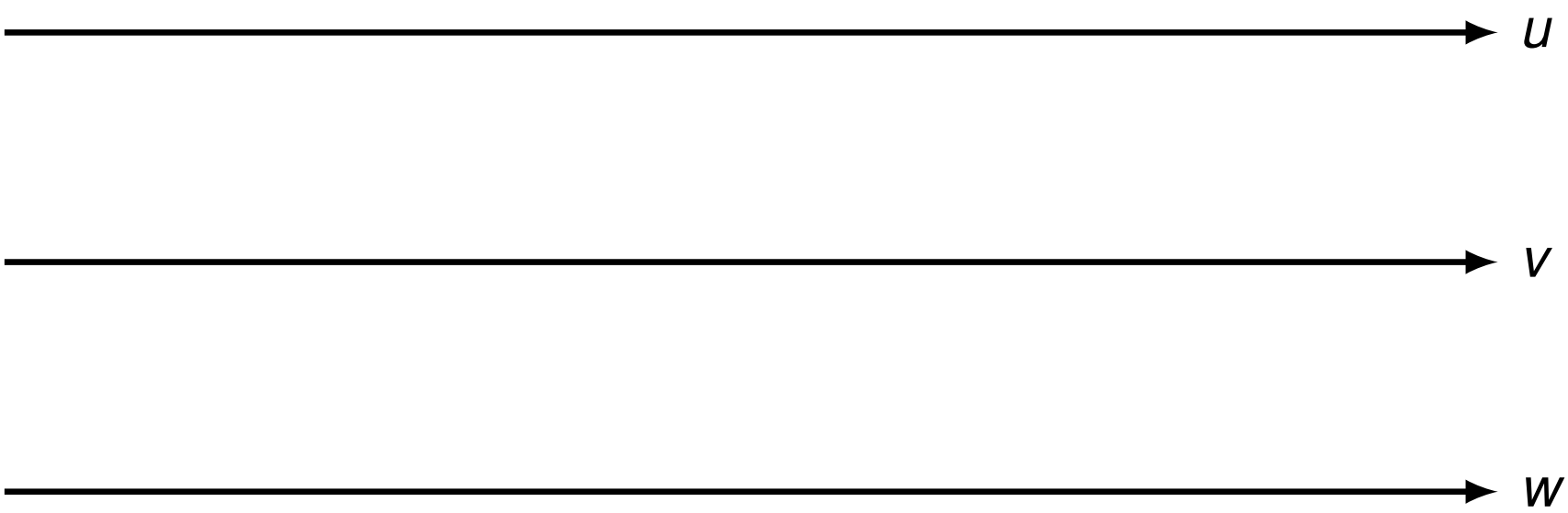
Filtrer les connexions dominées



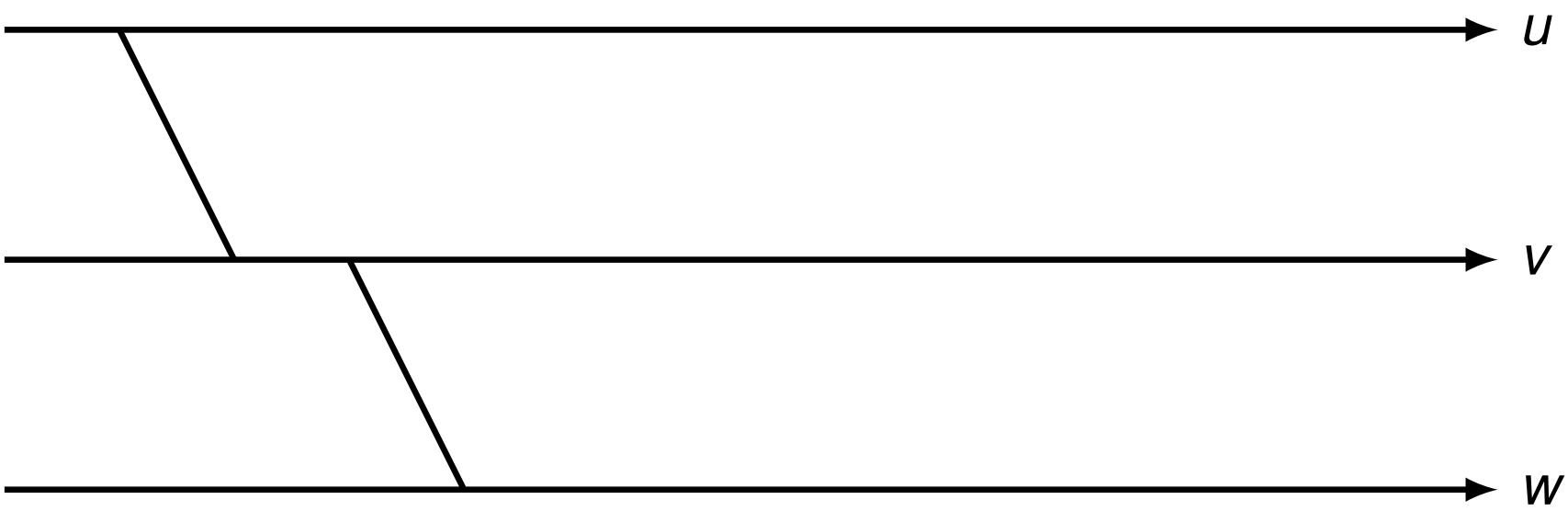
Filtrer les connexions dominées



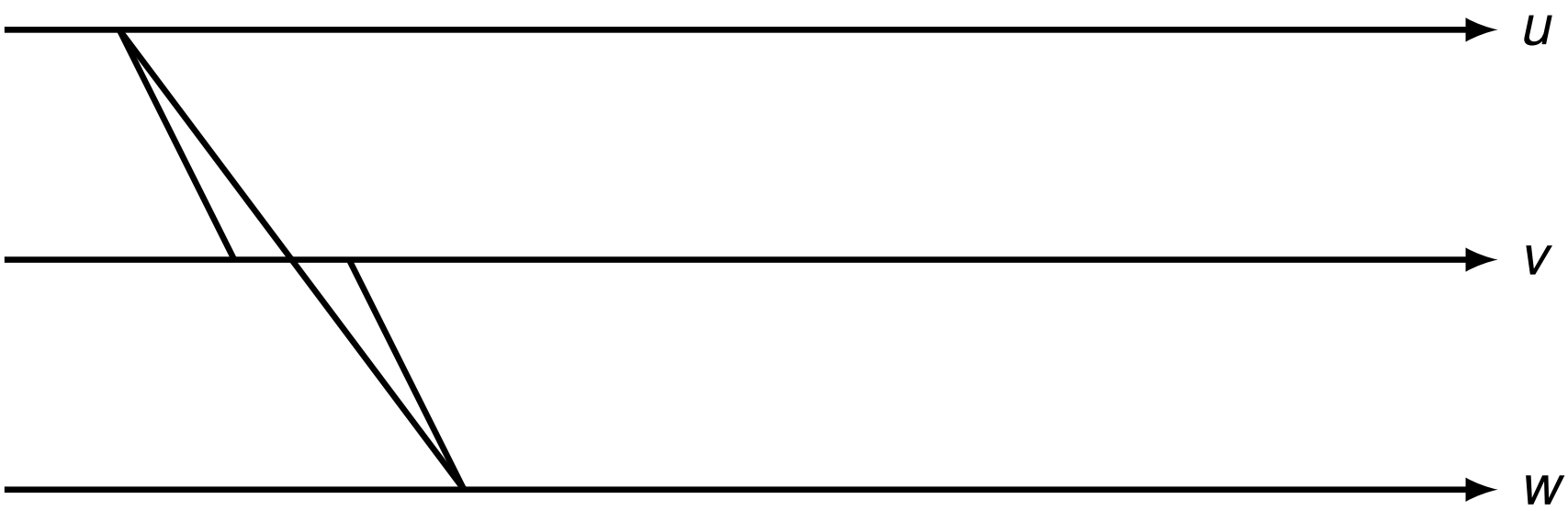
Composer deux ATFs



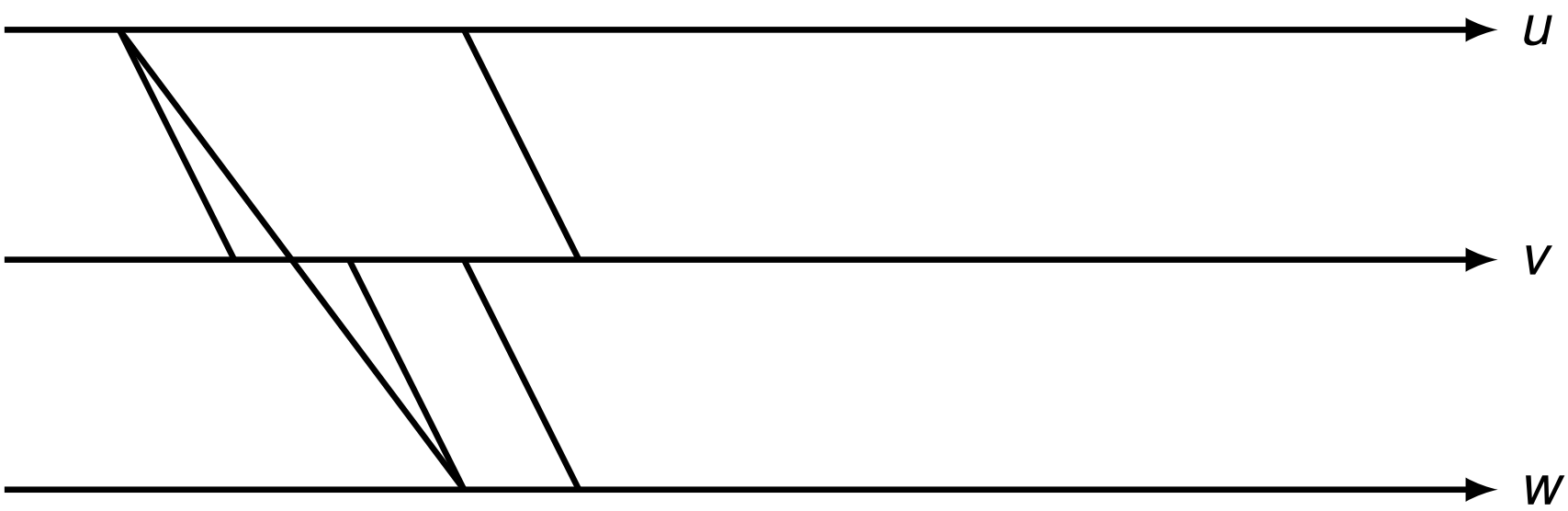
Composer deux ATFs



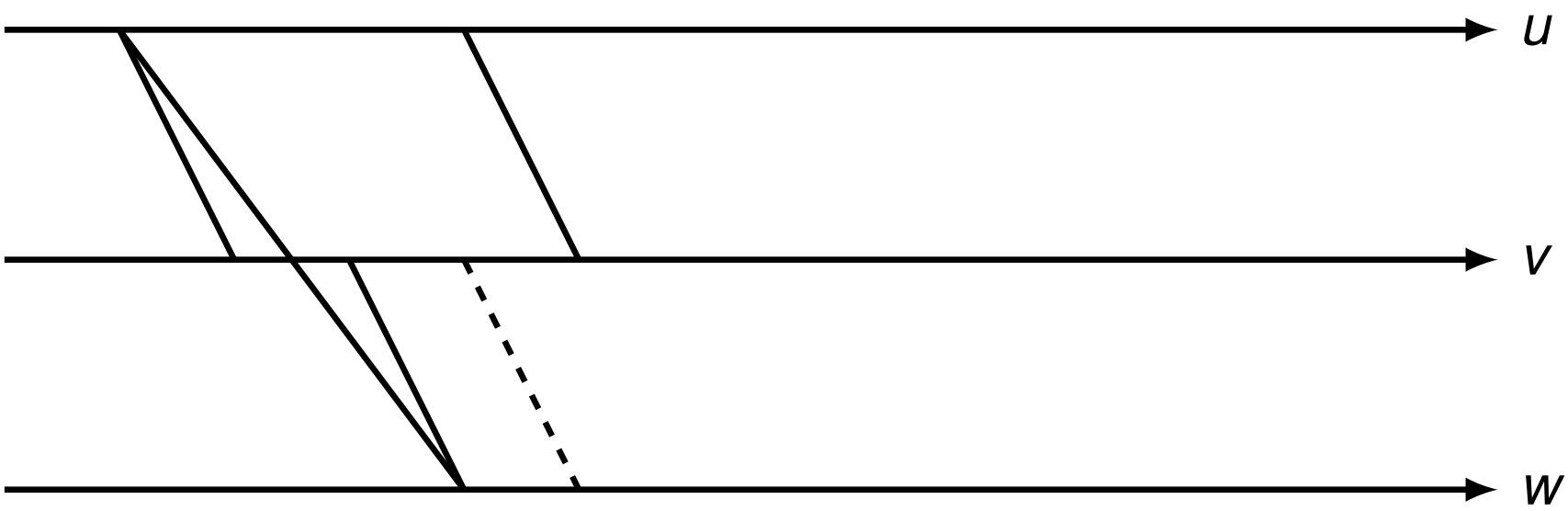
Composer deux ATFs



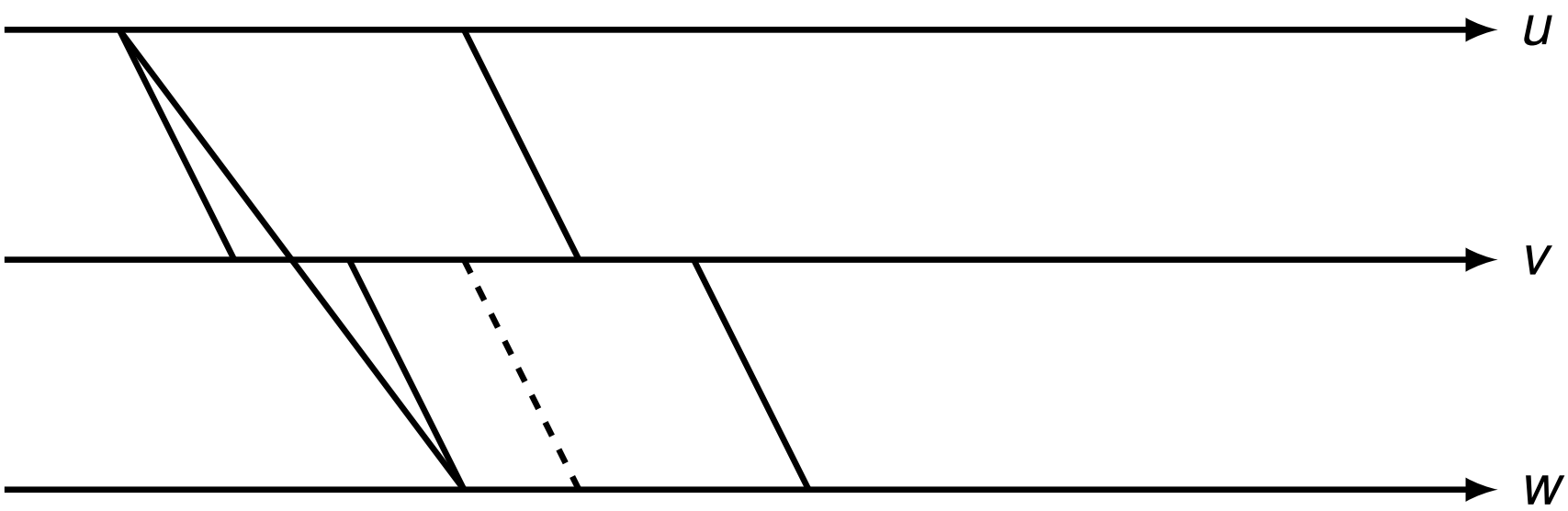
Composer deux ATFs



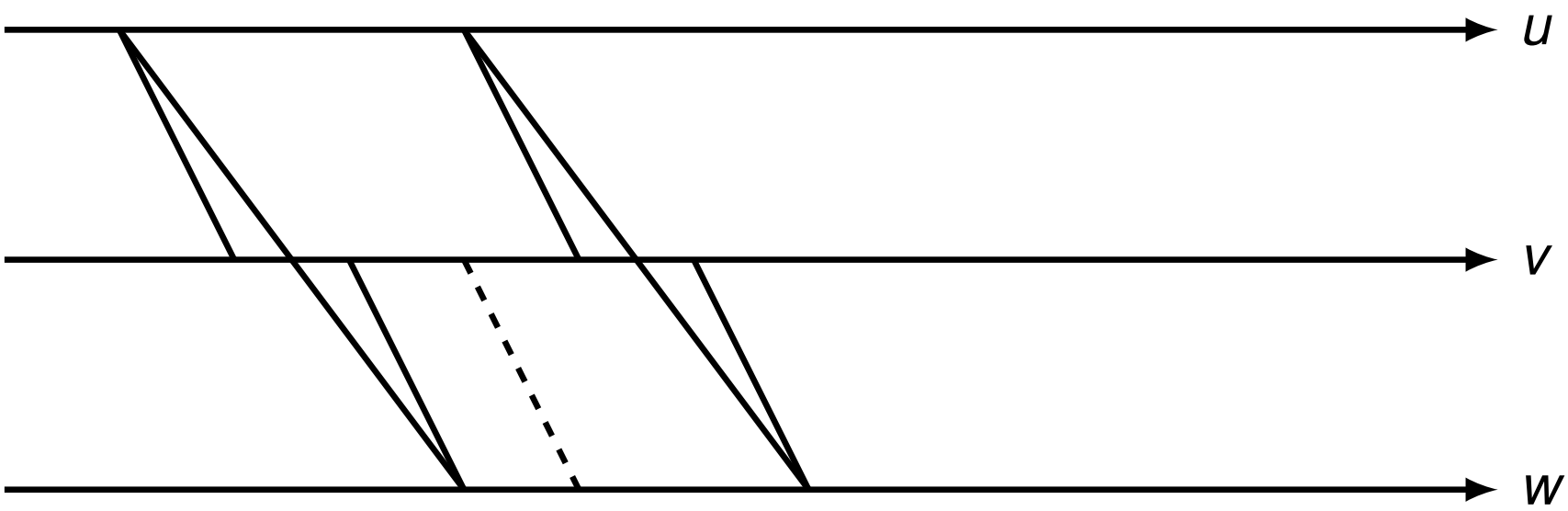
Composer deux ATFs



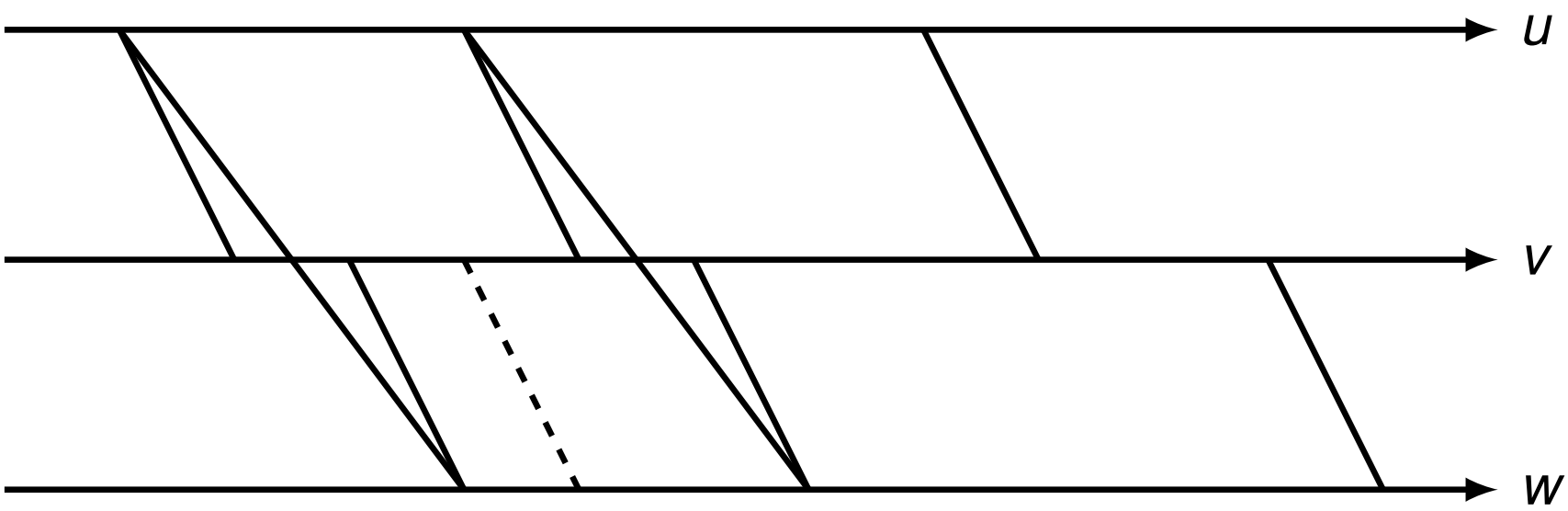
Composer deux ATFs



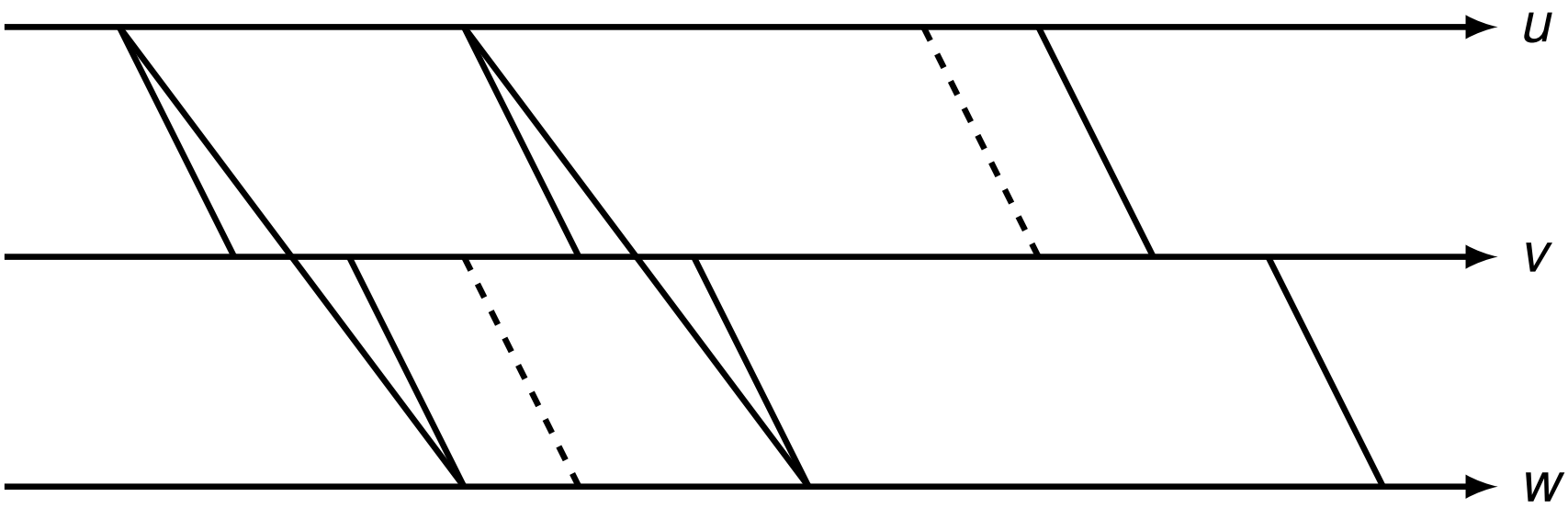
Composer deux ATFs



Composer deux ATFs



Composer deux ATFs



Composer deux ATFs