Bicreteria path problem minimizing the cost and minimizing the number of labels

Master Parisien de Recherche Opérationnelle

Marta Pascoal - M. Eugénia Captivo - João Clímaco - Ana Laranjeira
Julien Khamphousone

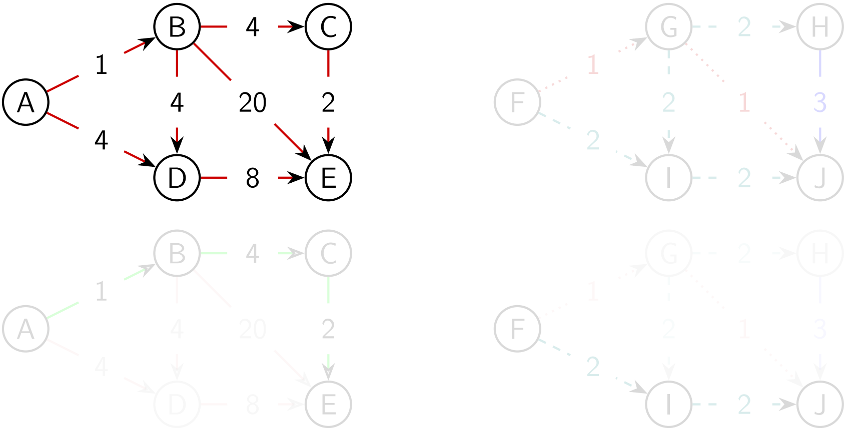
Published : 9 March 2013

Plan

Plan

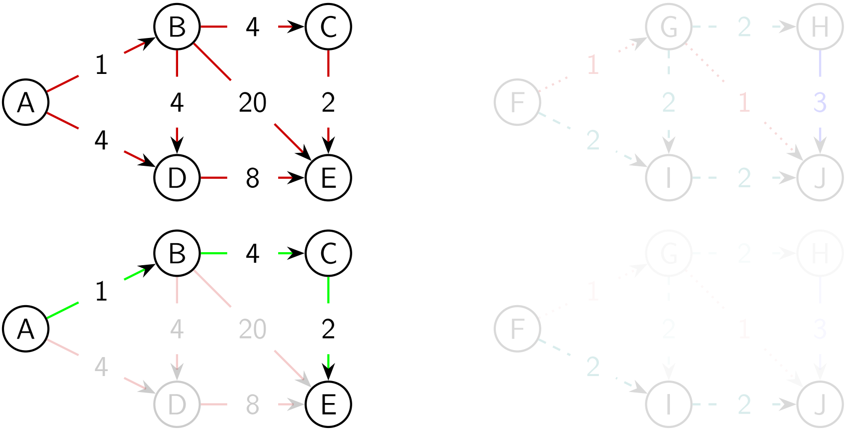
Problème étudié

- Constitué de deux sous problèmes



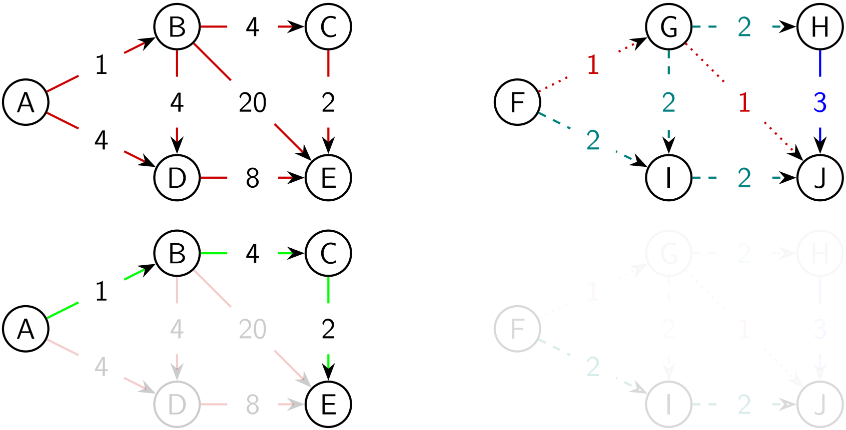
Problème étudié

- Constitué de deux sous problèmes



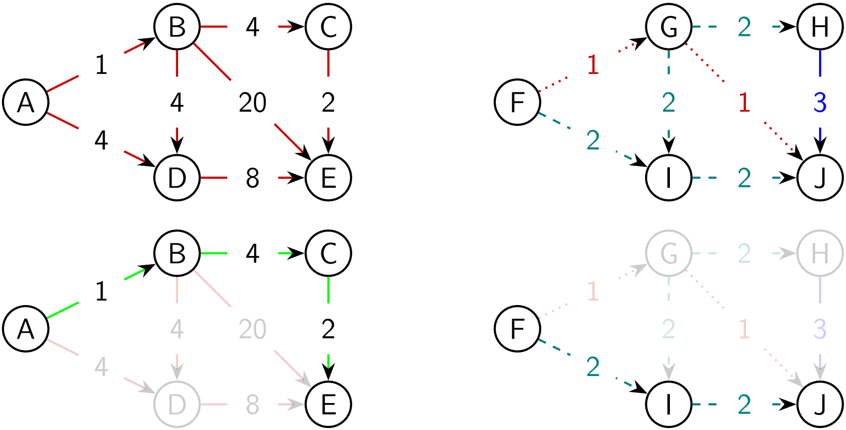
Problème étudié

- Constitué de deux sous problèmes



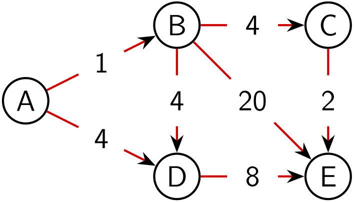
Problème étudié

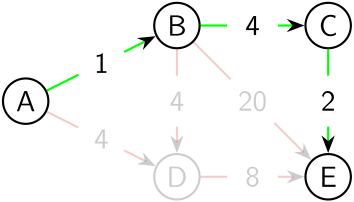
- Constitué de deux sous problèmes



Complexité 1er sous problème

- Problème du plus court chemin



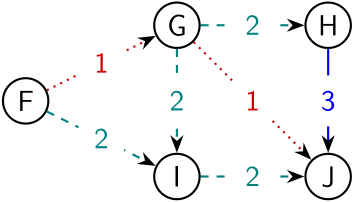


$\mathcal{P} = \{$chemin de s à $t\}$

- $c^* = \min_{p \in \mathcal{P}} c(p)$
- Polynomial : c^* peut être calculé en temps polynomial

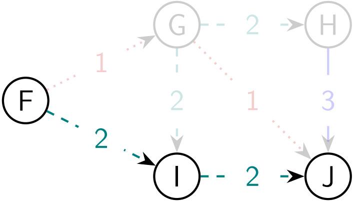
Complexité du 2nd sous problème

- Problème du chemin minimisant le nombre de labels utilisés



$\mathcal{P} = \{$chemin de s à $t\}$

- $l^* = \min_{p \in \mathcal{P}} l(p)$
- $NP - difficile$: l^* ne peut être calculé en temps polynomial [Wirth (2001)]



Problème étudié : MCLPP

- Problème bi-objectif de chemin de coût minimal et minimisant le nombre de labels

$\mathcal{P} = \{\text{chemin de } s \text{ à } t\}$

$$\min_{p \in \mathcal{P}} c(p)$$
$$\min_{p \in \mathcal{P}} l(p)$$

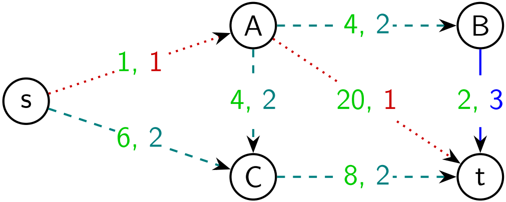
- Déjà étudié dans le cas des arbres couvrants [Clímaco et al. (2010)], on considère ici des chemins

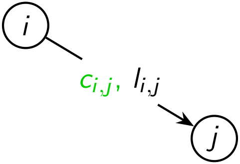
Problème étudié : MCLPP

- Problème bi-objectif de chemin de coût minimal et minimisant le nombre de labels

$\mathcal{P} = \{\text{chemin de } s \text{ à } t\}$

$$\min_{p \in \mathcal{P}} c(p)$$
$$\min_{p \in \mathcal{P}} l(p)$$





- Déjà étudié dans le cas des arbres couvrants [Clímaco et al. (2010)], on considère ici des chemins

Applications

Transports

- Itinéraire de plus petit coût entre deux points d'une carte ;
- Labels : différentes modes de transports (taxi, pédestre, bus, trains, etc.).

Télécommunications

- Plus court chemin entre deux nœuds d'un réseau (internet, électrique, etc.) ;
- Labels : différentes technologies ou opérateurs.

Applications

Transports

- Itinéraire de plus petit coût entre deux points d'une carte ;
- Labels : différentes modes de transports (taxi, pédestre, bus, trains, etc.).

Télécommunications

- Plus court chemin entre deux nœuds d'un réseau (internet, électrique, etc.) ;
- Labels : différentes technologies ou opérateurs.

Chemin efficace et dominance

Définition : chemin efficace

Un chemin $p \in \mathcal{P}$ est dit **efficace** $\iff \nexists p' \in \mathcal{P}$ tel que :

- $c(p') \leq c(p)$ et ;
- $l(p') \leq l(p)$;
- Au moins l'une des deux inégalités est stricte.

Définition : dominance

Pour $p \in \mathcal{P}$, s'il existe $p' \in \mathcal{P}$ tel que :

- $c(p') \leq c(p)$;
- $l(p') \leq l(p)$;
- Au moins l'une des deux inégalités est stricte ;

Nous dirons que $(c(p'), l(p'))$ **domine** $(c(p), l(p))$.

Chemin efficace et dominance

Définition : chemin efficace

Un chemin $p \in \mathcal{P}$ est dit **efficace** $\iff \nexists p' \in \mathcal{P}$ tel que :

- $c(p') \leq c(p)$ et ;
- $l(p') \leq l(p)$;
- Au moins l'une des deux inégalités est stricte.

Définition : dominance

Pour $p \in \mathcal{P}$, s'il existe $p' \in \mathcal{P}$ tel que :

- $c(p') \leq c(p)$;
- $l(p') \leq l(p)$;
- Au moins l'une des deux inégalités est stricte ;

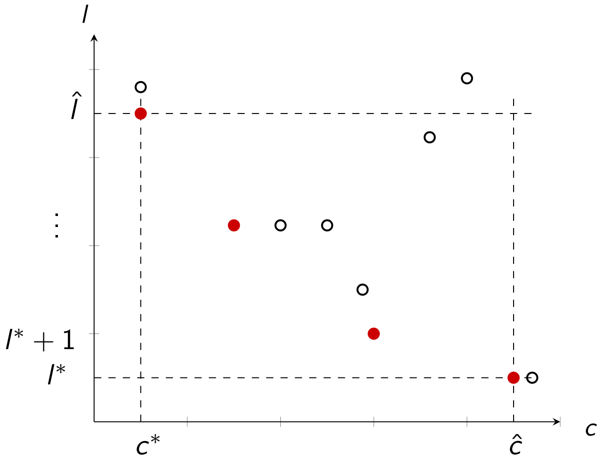
Nous dirons que $(c(p'), l(p'))$ **domine** $(c(p), l(p))$.

Objectif de l'article

Calculer un ensemble $\bar{\mathcal{P}}$ de chemins efficaces de cardinalité minimale

- Calculer $\bar{\mathcal{P}}$ tel que :
  - $\forall (p, p') \in \bar{\mathcal{P}}^2 : p \neq p'$, p et p' ont des images différentes, i.e. $c(p) \neq c(p')$ ou $l(p) \neq l(p')$;
  - Pour toute paire $(\bar{c}, \bar{l})$ non dominée, il existe un chemin $p \in \bar{\mathcal{P}}$ tel que $(c(p), l(p)) = (\bar{c}, \bar{l})$

Images possibles du MCLPP



- images de chemins efficaces
- images d'autres chemins

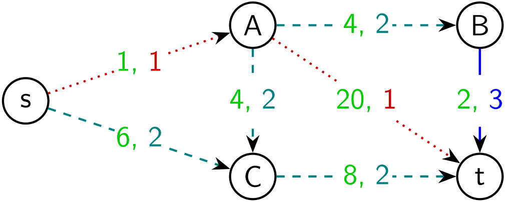
Plan

Algorithme d'énumération

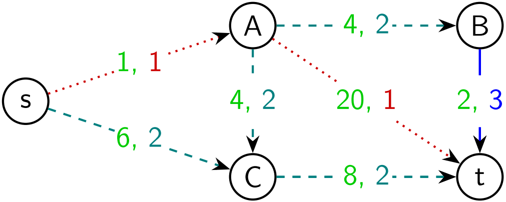
- **Algorithme pour construire $\bar{\mathcal{P}}$;**
- $\forall k = 1, \ldots, l(p^*)$, où p^* est le plus court chemin de notre graphe sans restriction sur les labels ;
  - Calculer le plus court chemin possédant k labels et l'ajouter à $\bar{\mathcal{P}}$;
  - Supprimer de $\bar{\mathcal{P}}$ les chemins dominés.

Algorithme d'énumération

- **Algorithme pour construire $\bar{\mathcal{P}}$**;
- $\forall k = 1, \ldots, l(p^*)$, où p^* est le plus court chemin de notre graphe sans restriction sur les labels;
  - Calculer le plus court chemin possédant k labels et l'ajouter à $\bar{\mathcal{P}}$;
  - Supprimer de $\bar{\mathcal{P}}$ les chemins dominés.

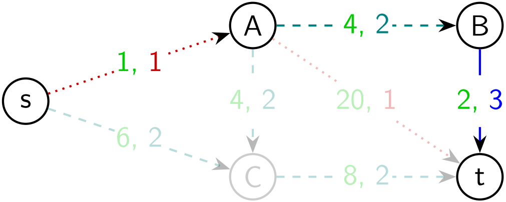


Algorithme d'énumération



- **Étape 1** : initialisation
- Calculer le plus court chemin sans restriction sur les labels

Algorithme d'énumération



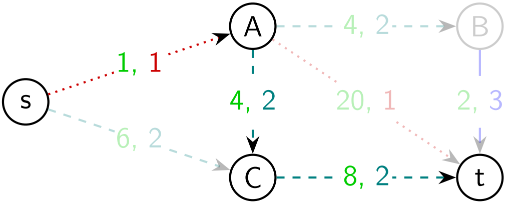
labels : $\{1, 2, 3\}$

$c(< s, A, B, t >) = 7$

- **Étape 1** : initialisation
- Calculer le plus court chemin sans restriction sur les labels $\longrightarrow l(p^*) = 3$

$\bar{\mathcal{P}} = \{< s, A, B, t >\}$

Algorithme d'énumération



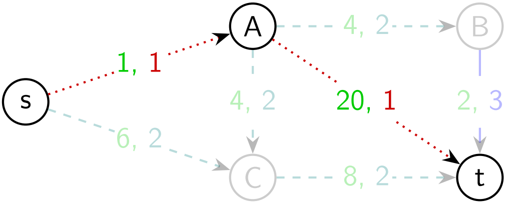
labels : {1, 2}

$c(< s, A, C, t >) = 13$

$\bar{\mathcal{P}} = \{< s, A, B, t >\}$

- **Étape 2** :
- Calculer les plus courts chemins avec au plus 2 labels

Algorithme d'énumération



labels : {1, 2}

$c(< s, A, C, t >) = 13$

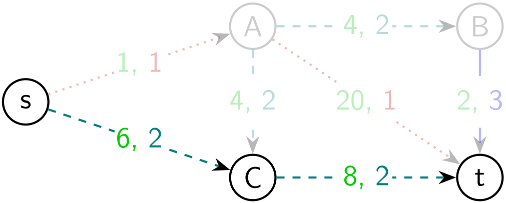
labels : {1, 3}

$c(< s, A, t >) = 21$

$\bar{\mathcal{P}} = \{< s, A, B, t >\}$

- **Étape 2** :
- Calculer les plus courts chemins avec au plus 2 labels

Algorithme d'énumération

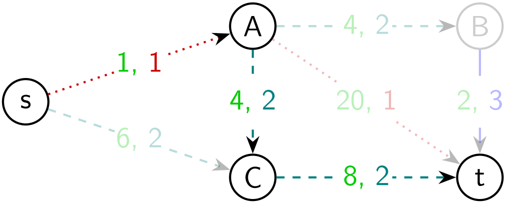


labels : {1, 2}
$c(< s, A, C, t >) = 13$
labels : {1, 3}
$c(< s, A, t >) = 21$
labels : {2, 3}
$c(< s, C, t >) = 14$

$\bar{\mathcal{P}} = \{< s, A, B, t >\}$

- **Étape 2** :
- Calculer les plus courts chemins avec au plus 2 labels

Algorithme d'énumération



labels : $\{1, 2\}$

$c(< s, A, C, t >) = 13$

labels : $\{1, 3\}$

$c(< s, A, t >) = 21$

labels : $\{2, 3\}$

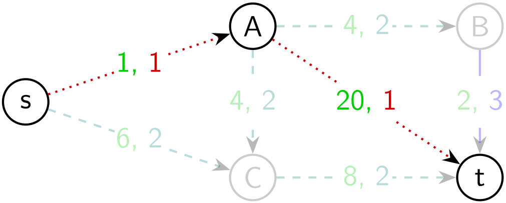
$c(< s, C, t >) = 14$

$\bar{\mathcal{P}} = \{< s, A, B, t >\}$

- **Étape 2** :
- Calculer les plus courts chemins avec au plus 2 labels

$\bar{\mathcal{P}} \leftarrow \bar{\mathcal{P}} \cup \{< s, A, C, t >\} = \{< s, A, B, t >, < s, A, C, t >\}$

Algorithme d'énumération

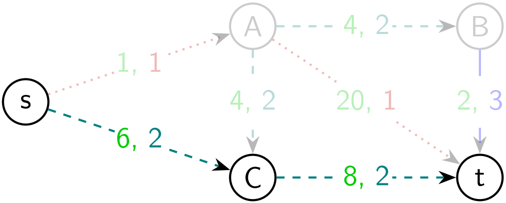


labels : $\{1\}$
$c(< s, A, t >) = 21$

$\bar{\mathcal{P}} = \{< s, A, B, t >, < s, A, C, t >\}$

- **Étape 3** :
- Calculer les plus courts chemins avec 1 label

Algorithme d'énumération



labels : $\{1\}$

$c(< s, A, t >) = 21$

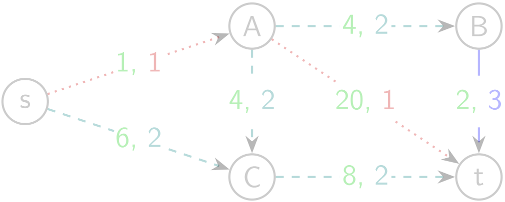
labels : $\{2\}$

$c(< s, C, t >) = 14$

$\bar{\mathcal{P}} = \{< s, A, B, t >, < s, A, C, t >\}$

- **Étape 3** :
- Calculer les plus courts chemins avec 1 label

Algorithme d'énumération



labels : $\{1\}$

$c(<s, A, t>) = 21$

labels : $\{2\}$

$c(<s, C, t>) = 14$

labels : $\{3\}$

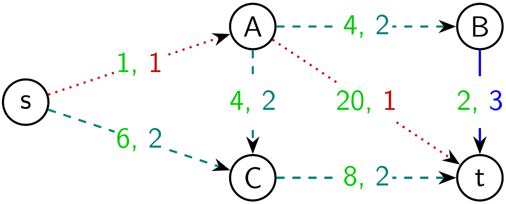
Pas de solutions

$\bar{\mathcal{P}} = \{<s, A, B, t>, <s, A, C, t>\}$

- **Étape 3** :
- Calculer les plus courts chemins avec 1 label

$\bar{\mathcal{P}} \leftarrow \bar{\mathcal{P}} \cup \{<s, , C, t>\} = \{<s, A, B, t>, <s, A, C, t>, <s, C, t>\}$

Algorithme d'énumération



- Finalement, l'algorithme retourne :

$$\begin{aligned}\bar{\mathcal{P}} = \{ &< s, A, B, t > \quad \textit{(3 labels)},\\ &< s, A, C, t > \quad \textit{(2 labels)},\\ &< s, C, t > \qquad \textit{(1 label)}\}\end{aligned}$$

Algorithmes proposés dans l'article

Construction de $\bar{\mathcal{P}}$ avec 5 algorithmes

- Algorithme `V1` d'énumération
  - Algorithme `V2` qui améliore `V1` en comparant les nouveaux plus courts chemins avec ceux calculés précédemment
  - Algorithme `V3` qui améliore encore `V2`
- Algorihtme `BFS1` avec un parcours en profondeur
- Algorithme `BFS2` variante de `BFS1` avec une réoptimisation des plus courts chemins

Plan

Algorithme d'énumération avec bornes supérieures : Principe

- `V1` + vecteur $CostUB_k$ de taille $1, \ldots, l(p^*)$;
- À une itération donnée, $\forall k = 1, \ldots, l(p^*)$
  $CostUB_k$: coût du meilleur plus court chemin ayant k labels parmi toutes les itérations précédentes ;
- Si lors du calcul du plus court chemin avec k labels, le coût intermédiaire à un nœud j compris sur le chemin de s à t est supérieur à $CostUB_k$, on stoppe le calcul.

Algorithme d'énumération avec bornes supérieures : Principe

- V1 + vecteur $CostUB_k$ de taille $1, \ldots, l(p^*)$;
- À une itération donnée, $\forall k = 1, \ldots, l(p^*)$

  $CostUB_k$: coût du meilleur plus court chemin ayant k labels parmi toutes les itérations précédentes ;
- Si lors du calcul du plus court chemin avec k labels, le coût intermédiaire à un nœud j compris sur le chemin de s à t est supérieur à $CostUB_k$, on stoppe le calcul.

Algorithme d'énumération avec bornes supérieures : Principe

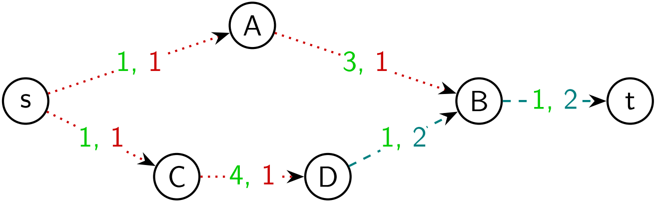
- `V1` + vecteur $CostUB_k$ de taille $1, \ldots, l(p^*)$;
- À une itération donnée, $\forall k = 1, \ldots, l(p^*)$

  $CostUB_k$: coût du meilleur plus court chemin ayant k labels parmi toutes les itérations précédentes ;
- Si lors du calcul du plus court chemin avec k labels, le coût intermédiaire à un nœud j compris sur le chemin de s à t est supérieur à $CostUB_k$, on stoppe le calcul.

Algorithme d'Énumération avec Bornes Supérieures

V2 : itération avec 2 labels

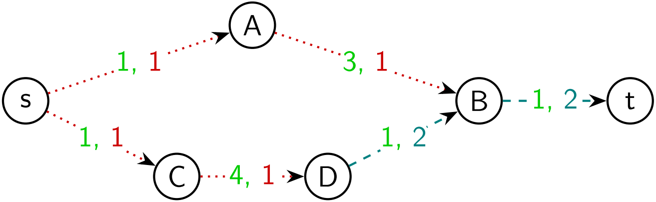
Nous nous plaçons en cours d'itération (étape 2) où :
$\bar{\mathcal{P}} = \{< s, A, B, t >, \quad \text{(2 labels)}\}$



Algorithme d'Énumération avec Bornes Supérieures

V2 : itération avec 2 labels

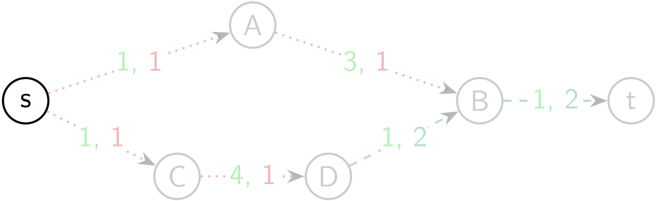
Nous nous plaçons en cours d'itération (étape 2) où :
$\bar{\mathcal{P}} = \{< s, A, B, t >, \quad \text{(2 labels)}\}$



- Nous allons maintenant sauvegarder dans $CostUB_2$, 2 car 2 labels, le coût de $\{< s, A, B, t >\}$ égal à 5

Algorithme d'Énumération avec Bornes Supérieures

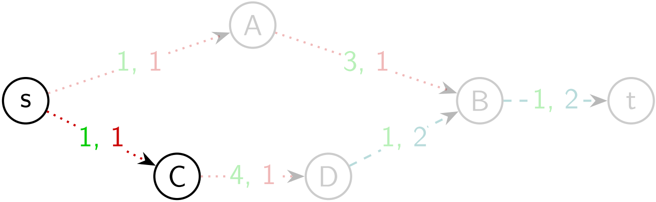
V2 : itération avec 2 labels



- L'algorithme doit désormais calculer l'autre plus court chemin avec 2 labels sachant que $CostUB_2 = 5$

Algorithme d'Énumération avec Bornes Supérieures

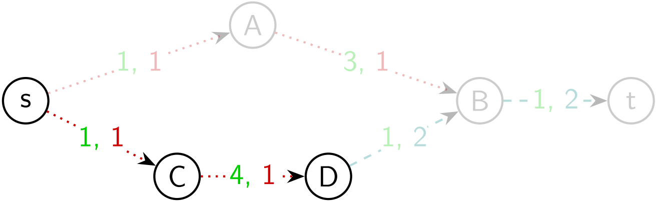
V2 : itération avec 2 labels



- L'algorithme doit désormais calculer l'autre plus court chemin avec 2 labels sachant que $CostUB_2 = 5$

Algorithme d'Énumération avec Bornes Supérieures

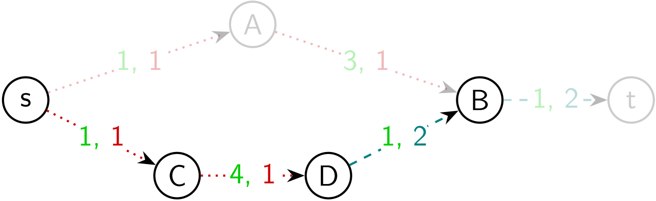
V2 : itération avec 2 labels



- L'algorithme doit désormais calculer l'autre plus court chemin avec 2 labels sachant que $CostUB_2 = 5$

Algorithme d'Énumération avec Bornes Supérieures

V2 : itération avec 2 labels



- Ici l'algorithme de labellisation qui calcule le plus court chemin s'arête car $\pi_D = 5$, coût du chemin de s à D est tel que $\pi_D + c_{D,B} = 6 > CostUB_2 = 5$.

Plan

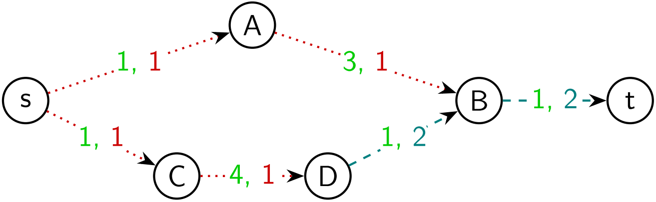
Algorithme avec pré-calcul de plus courts chemins

Afin d'améliorer `V2` :

- pour tout nœud v du graphe, calculer le plus court chemin de v au puits t noté π_v^t ;
- Utiliser l'arbre des coûts de ces plus courts chemins en plus de $CostUB$.

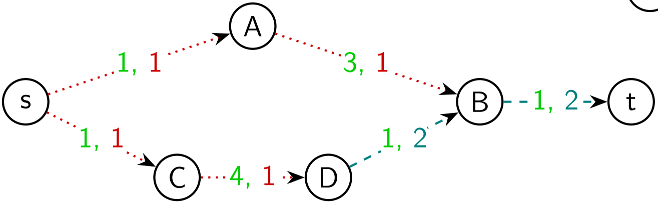
Algorithme avec pré-calcul de plus courts chemins

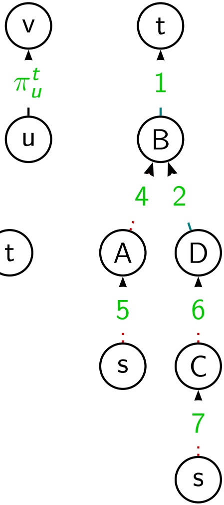
$\bar{\mathcal{P}} =< s, A, B, t >$
$CostUB_2 = 5$



Algorithme avec pré-calcul de plus courts chemins

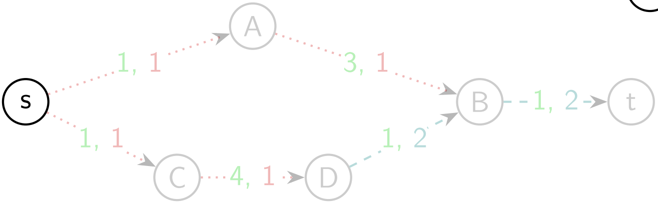
$$\bar{\mathcal{P}} = < s, A, B, t >$$
$$CostUB_2 = 5$$

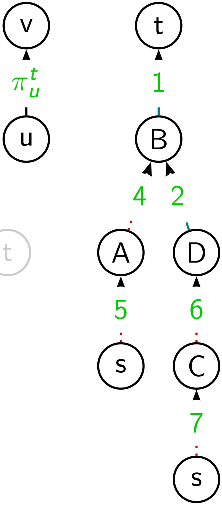




Algorithme avec pré-calcul de plus courts chemins

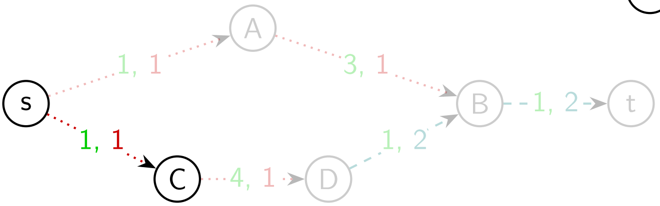
$$\bar{\mathcal{P}} = < s, A, B, t >$$
$$CostUB_2 = 5$$

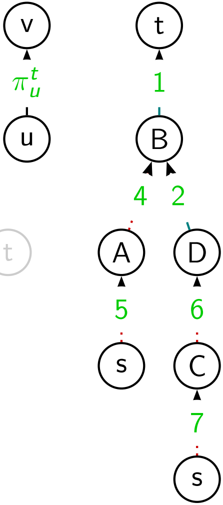




Algorithme avec pré-calcul de plus courts chemins

$$\bar{\mathcal{P}} = < s, A, B, t >$$
$$CostUB_2 = 5$$





- $\pi_s + c_{s,C} + \pi_j^t = 0 + 1 + 6$
- $\pi_s + c_{s,C} + \pi_j^t = 7 > CostUB_2$

Plan

Graphes testés

Graphes Aléatoires	Graphes de Grille
▶ $n \in \{100, 200, \dots, 1000\}$	▶ $[\![1; q]\!] \times [\![1; q]\!]$ ▶ $q \in \{10, 20, \dots, 50\}$ ▶ $n = q^2$
$c \in \{1, 2, \dots, 100\}$	labels $\in \{5, 10, 20\}$
▶ $m \in \{5n, 10n, 20n\}$	▶ $m = 4(2q-1)(q-1)$
▶ source et puits générés aléatoirements	▶ s (t) nœud coin inférieur gauche (supérieur droit) de la grille

Comparaison du temps d'exécution des algorithmes

Graphes Aléatoires		Graphes de Grille	
1er	V3	1er	V3
2e	V2	2e	V2
3e	BFS2	3e	V1
4e	V1	4e	BFS2
5e	BFS1	5e	BFS1

Perspective

- Ajouter une 3e fonction de coût qui pénaliserait un nombre trop important de transitions, i.e. changement de label entre deux arcs

Comparaison du temps d'exécution des algorithmes

Graphes Aléatoires		Graphes de Grille	
1er	V3	1er	V3
2e	V2	2e	V2
3e	BFS2	3e	V1
4e	V1	4e	BFS2
5e	BFS1	5e	BFS1

Perspective

- Ajouter une 3e fonction de coût qui pénaliserait un nombre trop important de transitions, i.e. changement de label entre deux arcs


QUESTIONS
SOURCES
PRÉCISIONS
PROBLÉMATIQUE
ANALYSE
DÉBAT
DISCUSSION
CONCLUSION
ÉVALUATION
BILAN
ARGUMENT
RAPPORT
PRÉSENTATION
DÉTAILS
MERCI
QUAND
EXPOSÉ
EXPLICATIONS
INFORMATIONS
DÉMONSTRATION
PÉDAGOGIQUE
QUANTIFICATION
COHÉRENCE
EXACTITUDE

Merci !