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

Article published: 9 March 2013 - https://doi.org/10.1007/

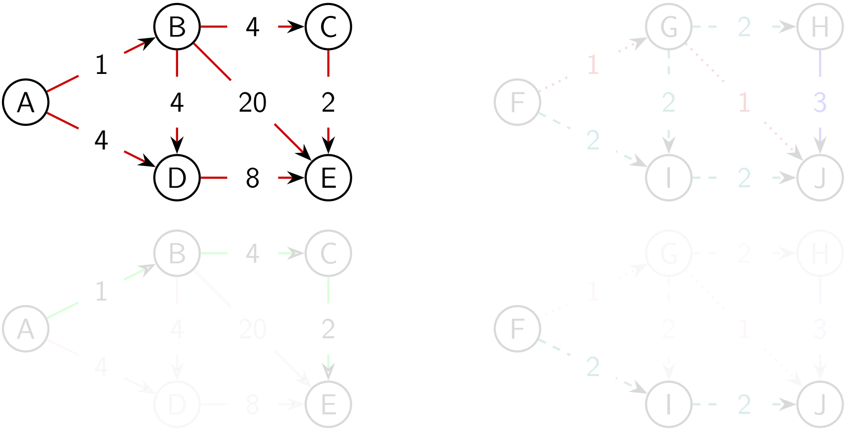
Translated from French to English in July 2024

Table of Contents

Table of Contents

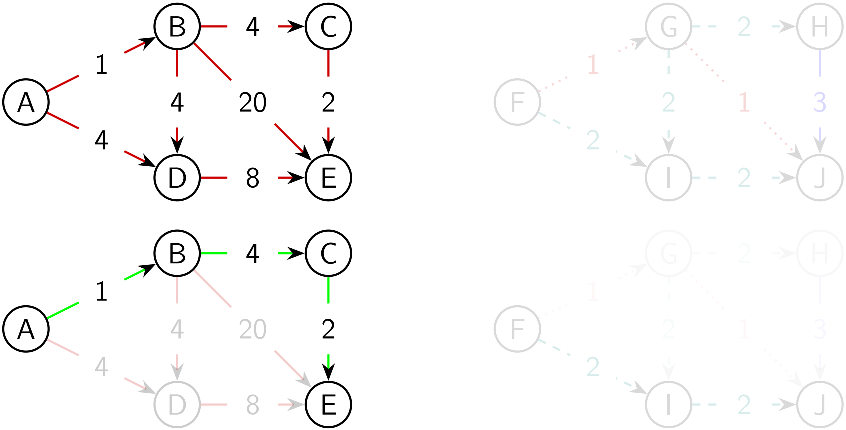
Problem studied

- Composed by two sub-problems



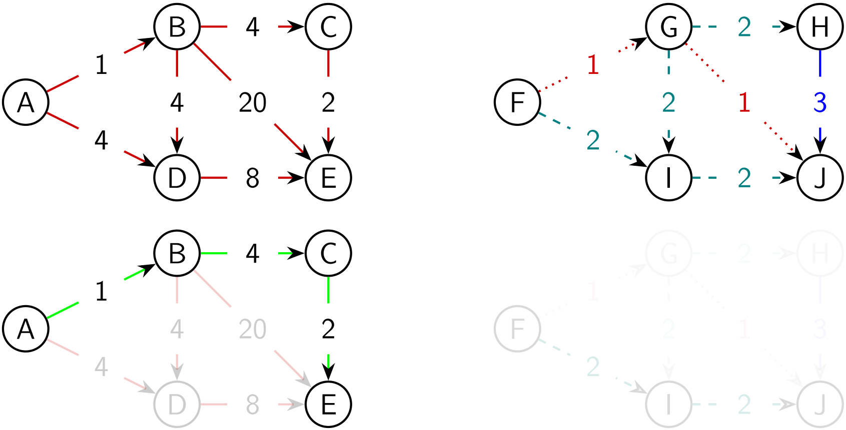
Problem studied

- Composed by two sub-problems



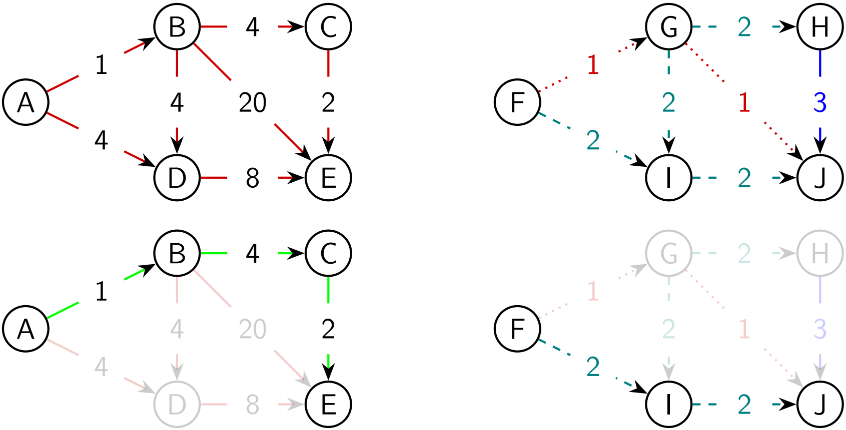
Problem studied

- Composed by two sub-problems



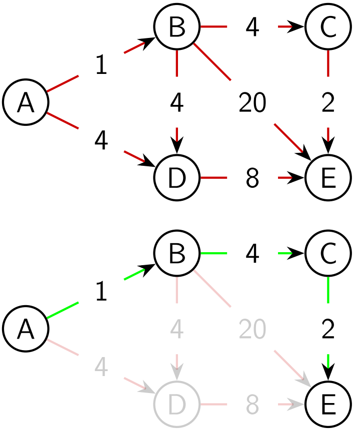
Problem studied

- Composed by two sub-problems



Complexity 1st subproblem

- Shortest path problem

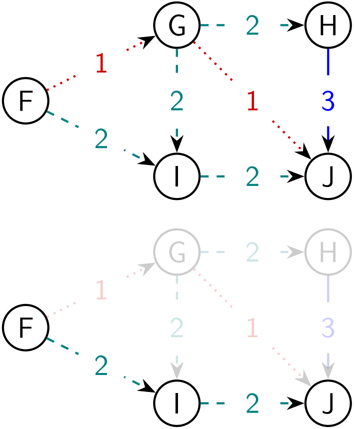


$\mathcal{P} = \{\text{path from } s \text{ to } t\}$

- $c^* = \min_{p \in \mathcal{P}} c(p)$
- Polynomial: c^* can be calculated in polynomial time

Complexity of the 2nd subproblem

- Path problem minimizing the number of labels used



$\mathcal{P} = \{\text{path from } s \text{ to } t\}$

- $l^* = \min_{p \in \mathcal{P}} l(p)$
- $NP - hard$: l^* cannot be calculated in polynomial time [Wirth (2001)]

Problem studied: MCLPP

- Bi-objective path problem with minimum cost and minimizing the number of labels

$\mathcal{P} = \{\text{path from } s \text{ to } t\}$

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

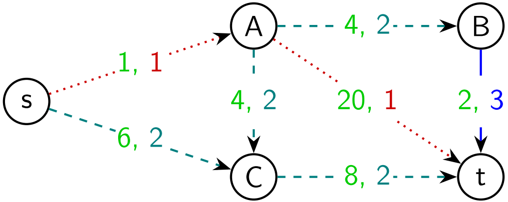
- Already studied in the case of spanning trees [Clímaco et al. (2010)], we consider here paths

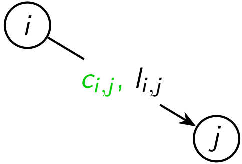
Problem studied: MCLPP

- Bi-objective path problem with minimum cost and minimizing the number of labels

$\mathcal{P} = \{\text{path from } s \text{ to } t\}$

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





- Already studied in the case of spanning trees [Clímaco et al. (2010)], we consider here paths

Applications

Transport

- Least cost route between two points on a map;
- Labels: different modes of transport (taxi, pedestrian, bus, trains, etc.).

Telecommunications

- Shortest path between two nodes of a network (internet, electricity, etc.);
- Labels: different technologies or operators.

Applications

Transport

- Least cost route between two points on a map;
- Labels: different modes of transport (taxi, pedestrian, bus, trains, etc.).

Telecommunications

- Shortest path between two nodes of a network (internet, electricity, etc.);
- Labels: different technologies or operators.

Efficient path and dominance

Definition: efficient path

A path $p \in \mathcal{P}$ is said **efficient** $\iff \nexists p' \in \mathcal{P}$ such that:

- $c(p') \leq c(p)$ and ;
- $l(p') \leq l(p)$;
- At least one of the two inequalities is strict.

Definition: dominance

For $p \in \mathcal{P}$, if there exists $p' \in \mathcal{P}$ such that:

- $c(p') \leq c(p)$;
- $l(p') \leq l(p)$;
- At least one of the two inequalities is strict;

We will say that $(c(p'), l(p'))$ **dominates** $(c(p), l(p))$.

Efficient path and dominance

Definition: efficient path

A path $p \in \mathcal{P}$ is said **efficient** $\iff \nexists p' \in \mathcal{P}$ such that:

- $c(p') \leq c(p)$ and ;
- $l(p') \leq l(p)$;
- At least one of the two inequalities is strict.

Definition: dominance

For $p \in \mathcal{P}$, if there exists $p' \in \mathcal{P}$ such that:

- $c(p') \leq c(p)$;
- $l(p') \leq l(p)$;
- At least one of the two inequalities is strict;

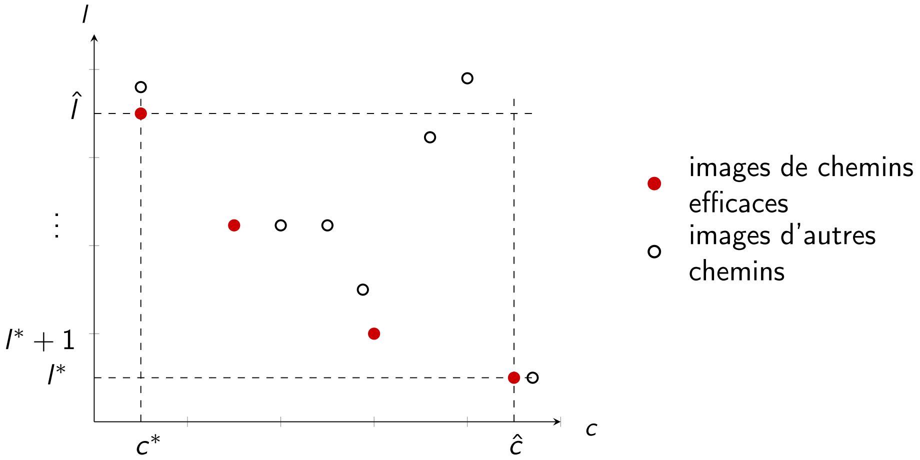
We will say that $(c(p'), l(p'))$ **dominates** $(c(p), l(p))$.

Purpose of the article

Compute a set $\bar{\mathcal{P}}$ of efficient paths of minimum cardinality

- Calculate $\bar{\mathcal{P}}$ such that:
  - $\forall (p, p') \in \bar{\mathcal{P}}^2 : p \neq p'$, p and p' have different images, i.e. $c(p) \neq c(p')$ or $l(p) \neq l(p')$;
  - For any non-dominated pair $(\bar{c}, \bar{l})$, there exists a path $p \in \bar{\mathcal{P}}$ such that $(c(p), l(p)) = (\bar{c}, \bar{l})$

Possible MCLPP images



Plan

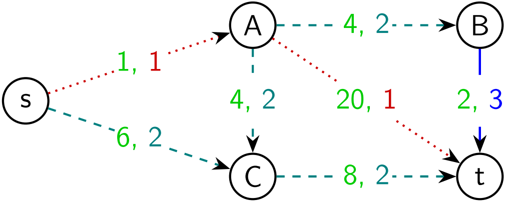
1. Problem Overview
2. Enumeration Algorithm
3. Enumeration algorithm with upper bounds
4. Algorithm with pre-calculation of shortest paths
5. Performance

Enumeration algorithm

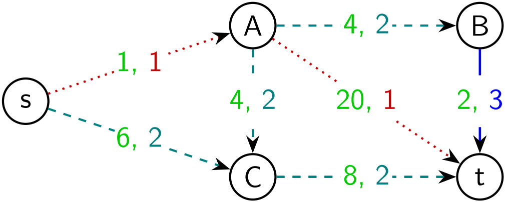
- **Algorithm to construct $\bar{\mathcal{P}}$;**
- $\forall k = 1, \ldots, l(p^*)$, where p^* is the shortest path of our graph without restriction on labels;
  - Calculate the shortest path having k labels and add it to $\bar{\mathcal{P}}$;
  - Remove dominated paths from $\bar{\mathcal{P}}$.

Enumeration algorithm

- **Algorithm to construct $\bar{\mathcal{P}}$** ;
- $\forall k = 1, \ldots, l(p^*)$, where p^* is the shortest path of our graph without restriction on labels;
  - Calculate the shortest path having k labels and add it to $\bar{\mathcal{P}}$;
  - Remove dominated paths from $\bar{\mathcal{P}}$.

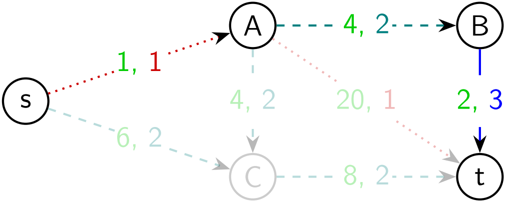


Enumeration algorithm



- **Step 1**: initialization
- Calculate the shortest path without restriction on labels

Enumeration algorithm



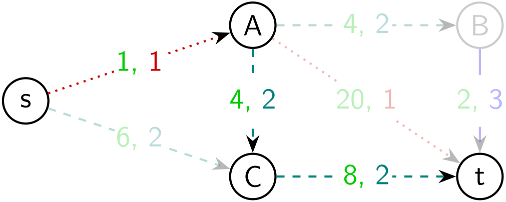
labels : $\{1, 2, 3\}$

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

- **Step 1**: initialization
- Calculate the shortest path without restriction on labels $\longrightarrow l(p^*) = 3$

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

Enumeration algorithm



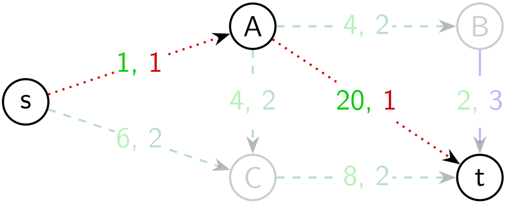
labels : {1, 2}

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

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

- **Step 2**:
- Calculate shortest paths with at most 2 labels

Enumeration algorithm



labels : {1, 2}

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

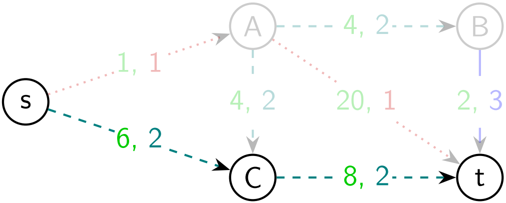
labels : {1, 3}

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

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

- **Step 2**:
- Calculate shortest paths with at most 2 labels

Enumeration algorithm



labels : $\{1, 2\}$

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

labels : $\{1, 3\}$

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

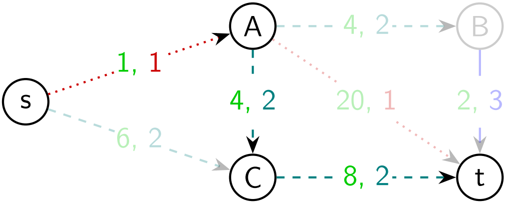
labels : $\{2, 3\}$

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

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

- **Step 2**:
- Calculate shortest paths with at most 2 labels

Enumeration algorithm



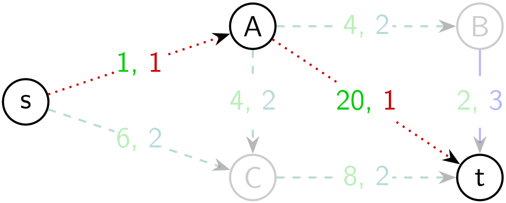
labels : $\{1, 2\}$
$c(< s, A, C, t >) = 13$
labels : $\{1, 3\}$
$c(< s, A, t >) = 21$
labels : $\{2, 3\}$
$c(< s, C, t >) = 14$

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

- **Step 2**:
- Calculate shortest paths with at most 2 labels

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

Enumeration algorithm



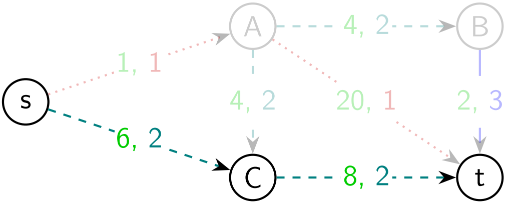
labels : $\{1\}$

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

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

- **Step 3**:
- Calculate shortest paths with 1 label

Enumeration algorithm

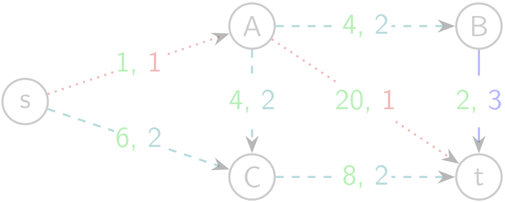


labels : $\{1\}$
$c(< s, A, t >) = 21$
labels : $\{2\}$
$c(< s, C, t >) = 14$

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

- **Step 3**:
- Calculate shortest paths with 1 label

Enumeration algorithm



labels : $\{1\}$

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

labels : $\{2\}$

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

labels : $\{3\}$

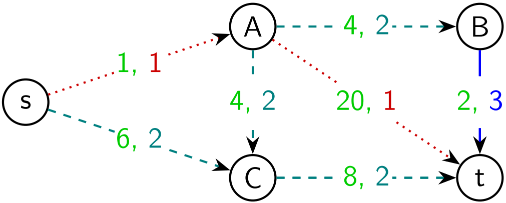
Pas de solutions

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

- **Step 3**:
- Calculate shortest paths with 1 label

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

Enumeration algorithm



- Finally, the algorithm returns:

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

Algorithms proposed in the article

Construction of $\bar{\mathcal{P}}$ with 5 algorithms

- Enumeration algorithm `V1`
  - Algorithm `V2` which improves `V1` by comparing the new shortest paths with those previously calculated
  - Algorithm `V3` which further improves `V2`
- Algorihtm `BFS1` with a depth scan
- Algorithm `BFS2` variant of `BFS1` with reoptimization of shortest paths

Plan

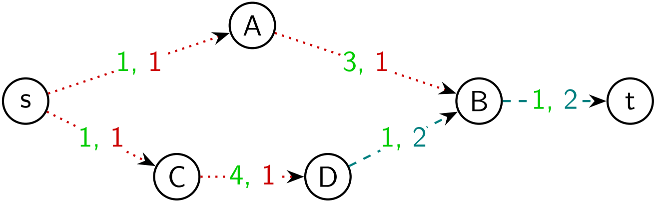
Enumeration algorithm with upper bounds: Principle

- `V1` + vector $CostUB_k$ of size $1, \ldots, l(p^*)$;
- At a given iteration, $\forall k = 1, \ldots, l(p^*)$
  $CostUB_k$: cost of the best shortest path having k labels among all previous iterations;
- If when calculating the shortest path with k labels, the intermediate cost at a node j included on the path from s to t is greater than $CostUB_k$, we stop the calculation.

Enumeration Algorithm with Upper Bounds

V2: iteration with 2 labels

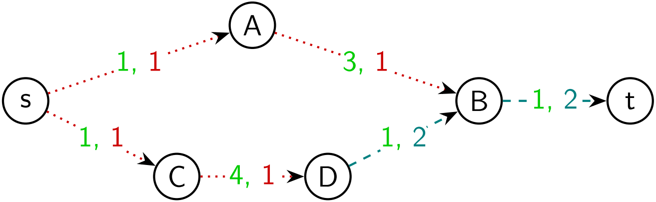
We place ourselves during iteration (step 2) where:
$\bar{\mathcal{P}} = \{< s, A, B, t >, \quad \text{(2 labels)}\}$



Enumeration Algorithm with Upper Bounds

V2: iteration with 2 labels

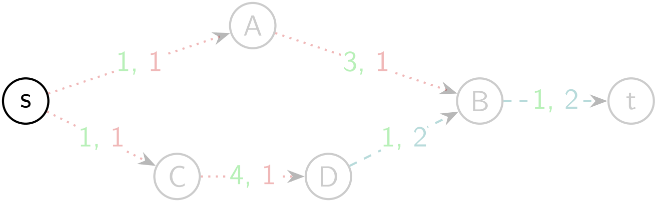
We place ourselves during iteration (step 2) where:
$\bar{\mathcal{P}} = \{< s, A, B, t >, \quad \text{(2 labels)}\}$



- We will now save in $CostUB_2$, 2 because 2 labels, the cost of $\{< s, A, B, t >\}$ equals to 5

Enumeration Algorithm with Upper Bounds

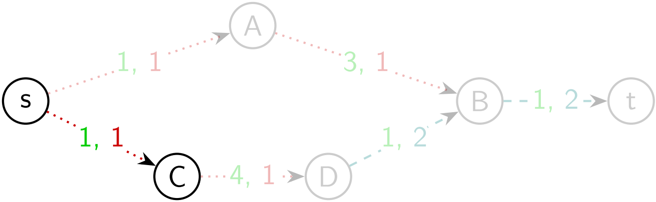
V2: iteration with 2 labels



- The algorithm must now calculate the other shortest path with 2 labels knowing that $CostUB_2 = 5$

Enumeration Algorithm with Upper Bounds

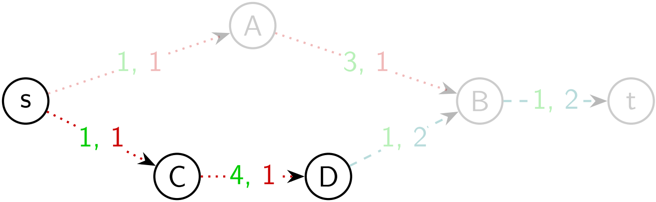
V2: iteration with 2 labels



- The algorithm must now calculate the other shortest path with 2 labels knowing that $CostUB_2 = 5$

Enumeration Algorithm with Upper Bounds

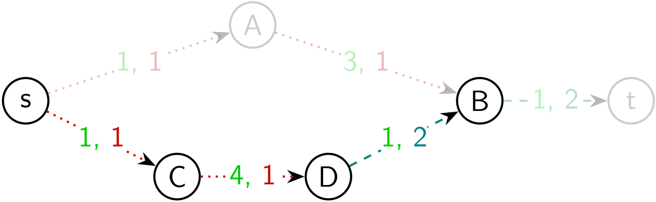
V2: iteration with 2 labels



- The algorithm must now calculate the other shortest path with 2 labels knowing that $CostUB_2 = 5$

Enumeration Algorithm with Upper Bounds

V2: iteration with 2 labels



- Here the labeling algorithm which calculates the shortest path stops because $\pi_D = 5$, cost of the path from s to D is such that $\pi_D + c_{D,B} = 6 > CostUB_2$ =5.

Plan

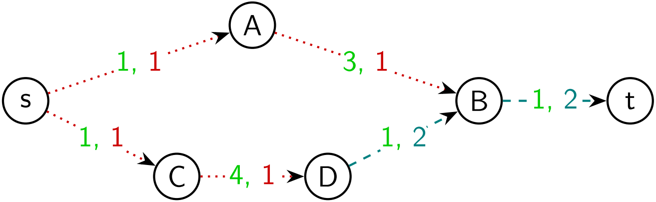
Algorithm with pre-calculation of shortest paths

In order to improve `V2`:

- for any node v of the graph, calculate the shortest path from v to the well t denoted π_v^t;
- Use the cost tree of these shortest paths in addition to $CostUB$.

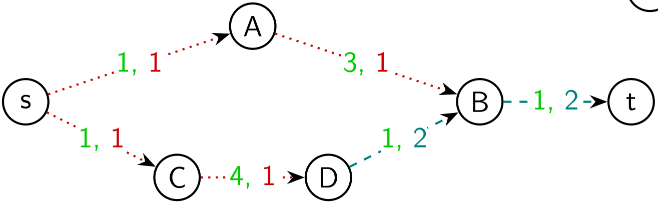
Algorithm with pre-calculation of shortest paths

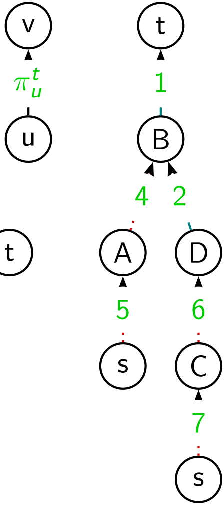
$\bar{\mathcal{P}} =< s, A, B, t >$
$CostUB_2 = 5$



Algorithm with pre-calculation of shortest paths

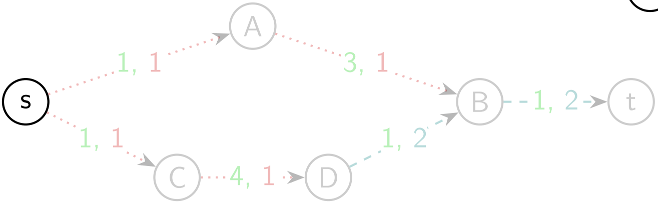
$$\bar{\mathcal{P}} =< s, A, B, t >$$
$$CostUB_2 = 5$$

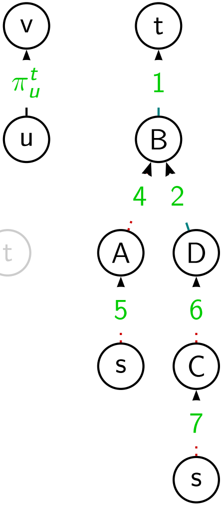




Algorithm with pre-calculation of shortest paths

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

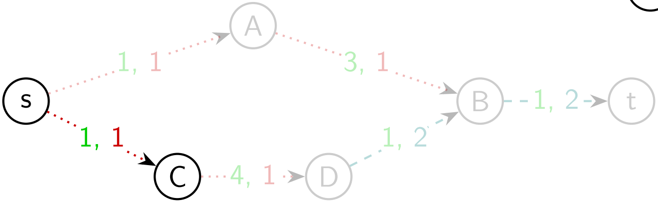


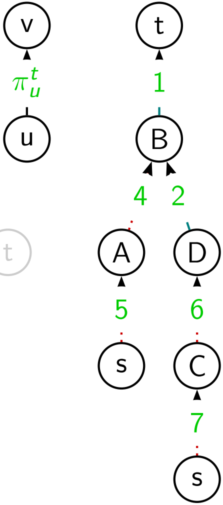


Algorithm with pre-calculation of shortest paths

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

$CostUB_2 = 5$





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

Plan

Graphs tested

Random Graphs	Grid Graphs
▶ $n \in \{100, 200, \ldots, 1000\}$	▶ $[\![1; q]\!] \times [\![1; q]\!]$ ▶ $q \in \{10, 20, \ldots, 50\}$ ▶ $n = q^2$
$c \in \{1, 2, \ldots, 100\}$	labels $\in \{5, 10, 20\}$
▶ $m \in \{5n, 10n, 20n\}$	▶ $m = 4(2q-1)(q-1)$
▶ source and sink generated randomly	▶ s (t respectively) node lower left (upper right) corner of the grid

Comparison of algorithm execution times

Random Graphs		Grid Graphs	
1st	V3	1st	V3
2e	V2	2e	V2
3e	BFS2	3e	V1
4e	V1	4e	BFS2
5e	BFS1	5e	BFS1

Perspective

- Add a 3rd cost function which would penalize too many transitions, i.e. change of label between two arcs


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

Thanks !