ECMA

Justine De Sousa   Natalia Jorquera

Le CNAM
Institute Polytechnique de Paris

11 février 2022

$$x_{ij} = 1 \text{ ssi l'arc (ij) est emprunté}$$

$$y_i = 1 \text{ ssi le sommet i est emprunté}$$

$$\left\{\begin{array}{l}
\min\limits_{x} \sum\limits_{ij \in A} d_{ij} \cdot x_{ij} \\
\quad s.c. : \\
\qquad \sum\limits_{i \in V} p_i \cdot y_i \leqslant S \\
\qquad \sum\limits_{i:ij \in A} x_{ij} = \sum\limits_{i:ji \in A} x_{ji} \ \forall\ j \in V \backslash \{s,t\} \\
\qquad \sum\limits_{j:sj \in A} x_{s,j} = 1 = \sum\limits_{i:it \in A} x_{i,t} \\
\qquad \sum\limits_{j:ij \in A} x_{ij} = y_i \ \forall\ i \in V \backslash \{s\} \\
\qquad \sum\limits_{j:ji \in A} x_{ji} = y_i \ \forall\ i \in V \backslash \{t\} \\
\qquad x_{ij} \in \{0,1\} \ \forall\ ij \in A \\
\qquad y_i \in \{0,1\} \ \forall\ i \in V
\end{array}\right.$$

$$
\left\{
\begin{array}{l}
\displaystyle \min_{(x,y)} \max_{\delta^1_{ij}} \sum_{ij\in A} d_{ij} \cdot (1+\delta^1_{ij}) \cdot x_{ij} \\
\qquad s.c. : \\
\qquad\qquad \left.\begin{array}{l}
\displaystyle \sum_{i:ij\in A} x_{ij} = \sum_{i:ji\in A} x_{ji} \ \forall\, j \in V\backslash\{s,t\}, \\
\displaystyle \sum_{j\in V} x_{sj} = \sum_{i\in V} x_{it} = 1, \\
\displaystyle \sum_{j:ij\in A} x_{ij} = y_i \ \forall\, i \in V\backslash\{s\}, \\
\displaystyle \sum_{j:ji\in A} x_{ji} = y_i \ \forall\, i \in V\backslash\{t\}, \\
x_{ij} \in \{0,1\} \ \forall\, (ij) \in A,\ y_i \in \{0,1\} \ \forall\, i \in V,
\end{array}\right\} \mathcal{X}^{comb} \\
\qquad\qquad \left.\begin{array}{l}
\left.\begin{array}{l}
\displaystyle \sum_{ij\in A} \delta^1_{ij} \leqslant d_1, \\
\delta^1_{ij} \in [0, D_{ij}] \ \forall\, (i,j) \in A,
\end{array}\right\} \mathcal{U}^1 \\
\left.\begin{array}{l}
\displaystyle \sum_{i\in V} (p_i + \delta^2_i \cdot \hat{p}_i) \cdot y_i \leqslant S, \\
\displaystyle \sum_{i\in V} \delta^2_i \leqslant d_2, \\
\delta^2_i \in [0,2] \ \forall\, i \in V,
\end{array}\right\} \mathcal{U}^2
\end{array}\right\} \mathcal{X}^{num}
\end{array}
\right.
$$

$$
\left\{
\begin{array}{l}
\displaystyle \min_{(x,y),(\alpha,\beta)} \sum_{ij\in A} d_{ij}\cdot x_{ij} + \sum_{ij\in A} D_{ij}\cdot\beta_{ij} + d_1\cdot\alpha \\
\quad s.c.: \\
\quad\quad \left.\begin{array}{ll}
\displaystyle \sum_{i:ij\in A} x_{ij} = \sum_{i:ji\in A} x_{ji} & \forall\, j\in V\backslash\{s,t\}, \\
\displaystyle \sum_{j\in V} x_{sj} = 1 = \sum_{i\in V} x_{it}, & \\
\displaystyle \sum_{j:ij\in A} x_{ij} = y_i & \forall\, i\in V\backslash\{s\}, \\
\displaystyle \sum_{j:ji\in A} x_{ji} = y_i & \forall\, i\in V\backslash\{t\},
\end{array}\right\} \mathcal{X}^{comb} \\
\quad\quad \left.\begin{array}{l}
\left.\begin{array}{ll}
\alpha+\beta_{ij} \geqslant d_{ij}\cdot x_{ij} & \forall\, ij\in A \\
\alpha\geqslant 0,\ \beta_{ij}\geqslant 0 & \forall\, ij\in A
\end{array}\right\}\mathcal{U}^1 \\
\left.\begin{array}{ll}
\displaystyle \sum_{i\in V} p_i\cdot y_i + \sum_{i\in V} 2\varepsilon_i + d_2\cdot\gamma \leqslant S, & \\
\gamma+\varepsilon_i \geqslant \hat{p}_i\cdot y_i & \forall\, i\in V \\
\gamma\geqslant 0,\ \varepsilon_i\geqslant 0 & \forall\, i\in V
\end{array}\right\}\mathcal{U}^2
\end{array}\right\} \mathcal{X}^{num}, \\
\quad\quad x_{ij}\in\{0,1\} \quad \forall\,(ij)\in A, \\
\quad\quad y_i\in\{0,1\} \quad \forall\, i\in V,
\end{array}
\right.
$$

Algorithme de plans coupants

Entrées : Graphe G, Sommet départ s, Sommet arrivée t, problème maître (MP), sous-problème (SPo,SP2), données de robustesse

Initialiser $\mathcal{U}^{1*} = \varnothing \quad \mathcal{U}^{2*} = \varnothing$, SPoValue=0, SP2=0, robustValue=0, $\hat{x}_{ij} = 0 \; \forall i,j \in A, \hat{y}_i = 0 \; \forall i \in V$, $\hat{\delta}^1_{ij} = 0 \; \forall i,j \in A, \hat{\delta}^2_i = 0 \; \forall i \in V$

tant que *SPoValue>robustValue || SP2Value>S* **faire**

  si *SPoValue>robustValue* **alors**

    $(MP) \leftarrow \sum_{(ij) \in A} d_{ij} \cdot (1 + \hat{\delta}^1_{ij}) \cdot x_{ij} \leqslant z$

  si *SP2Value>S* **alors**

    $(MP) \leftarrow \sum_{i \in V} (p_i + \hat{\delta}^2_i \cdot \hat{p}_i) \cdot y_i \leqslant S$

  Résoudre (MP)

  robustValue = z

  $\hat{x}_{ij} = x^\star_{ij} \; \forall i,j \in A$

  $\hat{y}_i = y^\star_i \; \forall i \in V$

  Résoudre (SPo)

  SPoValue = $\sum_{(ij) \in A} d_{ij} \cdot (1 + \delta^1_{ij}) \cdot x^\star_{ij}$

  $\hat{\delta}^1_{ij} = \delta^{1\star}_{ij} \; \forall i,j \in A$

  Résoudre (SP2)

  SP2Value = $\sum_{i \in V} (p_i + \delta^2_i \cdot \hat{p}_i) \cdot y^\star_i$

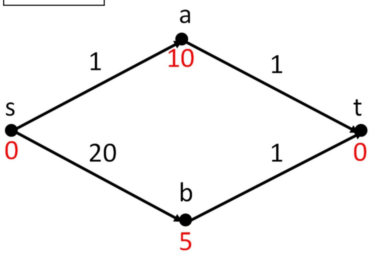
  $\hat{\delta}^2_i = \delta^{2\star}_i \; \forall i \in V$

- à chaque nœud de la résolution de branch & Bound, on vérifie si les conditions d'optimalité sont remplies :
  - $\sum_{(ij)\in A} d_{ij} \cdot (1 + \delta_{ij}^{1\star}) \cdot x_{ij}^{\star} \leqslant z^{\star}$
  - $\sum_{i\in V} (p_i + \delta_i^{2\star} \cdot \hat{p}_i) \cdot y_i^{\star} \leqslant S$
- Si les conditions d'optimalité ne sont pas remplies, les contraintes suivantes sont ajoutées, et la résolution se poursuit :
  - $\sum_{(ij)\in A} d_{ij} \cdot (1 + \hat{\delta}_{ij}^{1}) \cdot x_{ij} \leqslant z$
  - $\sum_{i\in V} (p_i + \hat{\delta}_i^{2} \cdot \hat{p}_i) \cdot y_i \leqslant S$
- Si les conditions d'optimalité sont remplies, il s'arrête.

Heuristique : l'algorithme $A^\star$

```
Initialiser closedList à vide;
Initialiser openList à vide;
Ajouter s à openList;
tant que openList non vide faire
    Retirer le sommet u de openList tel que f(u) soit le plus petit;
    si u == t alors
        Retourner le chemin;
    sinon
        Ajouter u à closedList;
    pour chaque successeur i de u faire
        h(i) : estimation de la distance de i à t;
        g_tmp ← g(u) + arc(u,i);
        f_tmp ← g(u) + h(u);
        si i est déjà dans openList avec un meilleur f(i) alors
            On passe au voisin suivant;
        si i est déjà dans closedList avec un meilleur f(i) alors
            On passe au voisin suivant;
        parent(i) ← u;
        g(i) ← g_tmp;
        g(i) ← f_tmp;
        openList ← openList ∪ i;
        closedList ← closedList \ i;
    closedList ← closedList \ u;
```

$\boxed{S = 10}$



Reconstitution du chemin : [s, a, t] de valeur 12

`openList = [ (s,0) ]`

- itération 1 : `currNode = s`
  - `openList` = [(a,11), (b,20)]
  - `closedList` = [s]
  - `parents` = [(a,s), (b,s)]
- itération 2 : `currNode = a`
  - `openList` = [(b,20), (t,12)]
  - `closedList` = [s, a]
  - `parents` = [(a,s), (b,s), (t,a)]
- itération 3 : `currNode = t`
  - `STOP`

$\boxed{S = 9}$

a non ajouté à `openList`

chemin [s, b, t] de valeur 26

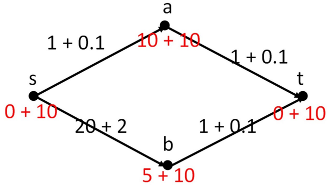
$\boxed{S = 4}$

a,b non ajoutés à `openList`

`openList` vide : l'algo s'arrête

Heuristique : ajout des poids et durées maximales

$$S = 10$$
$$\hat{p} = (5, 5, 5, 5)$$
$$D_{ij} = (10, 10, 10, 10)$$



Reconstitution du chemin : [s, a, t]

`openList = [ (s,0) ]`

- <u>itération 1 :</u> `currNode = s`
  - `openList` = [(a,31.1), (b,42)]
  - `closedList` = [s]
  - `parents` = [(a,s), (b,s)]
- <u>itération 2 :</u> `currNode = a`
  - `openList` = [(b,42), (t,41.2)]
  - `closedList` = [s, a]
  - `parents` = [(a,s), (b,s), (t,a)]
- <u>itération 3 :</u> `currNode = t`
  - `STOP`

Algorithme 1 : Post-traitement

si $\sum_{i \in P} 2 * \hat{p}_i > d_2$ **alors**

  Trouver $(\delta_i)^2_{i \in P} = \arg\max \left\{ \sum_{i \in P} p_i + \delta_i^2 \hat{p}_i \,\middle|\, \sum_{i \in P} \delta_i^2 \leqslant d_2, \forall i \in P \; \delta_i^2 \leqslant 2 \right\}$

fin

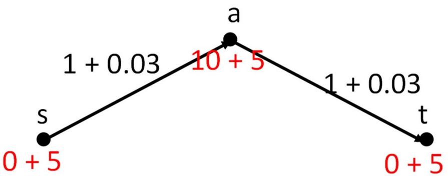
si $\sum_{(ij) \in P} D_{ij} > d_1$ **alors**

  Trouver

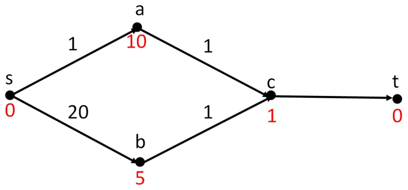
  $(\delta_{ij})^1_{(ij) \in P} = \arg\max \left\{ \sum_{(ij) \in P} d_{ij}(1 + \delta_{ij}^1) \,\middle|\, \sum_{ij} \delta_{ij}^1 \leqslant d_1, \forall (ij) \in P \; \delta_{ij}^1 \leqslant D_{ij} \right\}$

fin

$d_1 = 10$
$d_2 = 6$



- Si solution non réalisable : l′algorithme s′arrête



- Recherche de voisinages ?

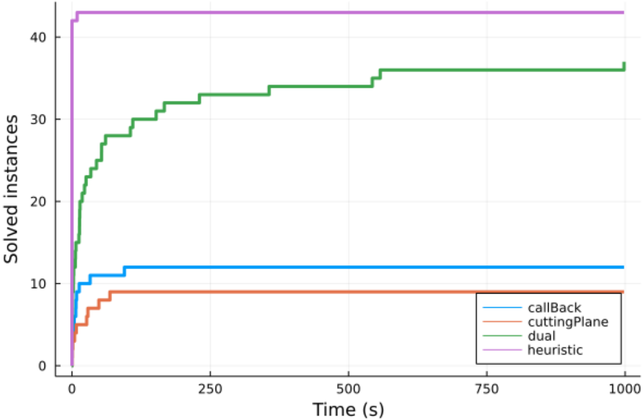


Figure 1 – Diagramme de performance des 4 méthodes utilisées

	callBack		cuttingPlane		dual		heuristic		
Instance	Temps (s)	GAP	Temps (s)	GAP	Temps (s)	GAP	Temps (s)	GAP	PR
20_BAY	0.39	0.0%	1.4	0.0%	0.08	0.0%	0.0	6.66%	0.27%
20_COL	0.18	0.0%	0.7	0.0%	0.09	0.0%	0.0	0.52%	0.45%
20_NY	0.1	0.0%	0.44	0.0%	0.08	0.0%	0.0	0.48%	1.73%
40_BAY	2.05	0.0%	7.86	0.0%	0.32	0.0%	0.0	1.69%	1.02%
40_NY	3.44	0.0%	68.62	0.0%	0.4	0.0%	0.0	0.4%	1.83%
40_COL	7.31	6.65%	100.42	0.0%	0.62	0.0%	0.0	1.66%	0.28%
60_COL	4.01	0.0%	4.71	0.0%	0.8	0.0%	0.0	11.22%	3.61%
60_NY	8.43	0.0%	48.63	0.0%	0.93	0.0%	0.0	1.09%	2.59%
60_BAY	6.9	0.0%	28.83	0.0%	0.88	0.0%	0.0	18.64%	3.64%
80_NY	94.88	5.91%	101.45	0.0%	3.15	0.0%	0.0	7.11%	2.39%
80_BAY	32.81	3.56%	100.34	0.0%	2.72	0.0%	0.0	4.56%	1.14%
80_COL	13.08	0.0%	26.59	0.0%	2.54	0.0%	0.0	1.45%	1.55%
100_NY	100.43	100%	104.11	0.0%	6.02	0.0%	0.0	100%	4.21%
100_BAY	100.87	100%	101.41	0.0%	6.32	0.0%	9.32	10.67%	1.02%
100_COL	100.42	100%	103.92	0.0%	7.15	0.0%	0.0	6.9%	4.3%

- Le problème du plus court chemin robuste a été modélisé mathématiquement et quatre méthodes de résolution ont été implémentées
- Nous soulignons la rapidité des heuristiques utilisées, ainsi que l'efficacité dans de petites instances des méthodes exactes.
- Reste ouvert : utiliser un mélange de ces méthodes pour augmenter la vitesse, l'étude de la modélisation mathématique, entre autres.

Merci de votre attention !