

Problème de partitionnement robuste

Nicolas BESSIN, Natello DESCORMIER

Table des matières

| | | |
|-----|---|----|
| 1 | Modélisation du problème statique | 2 |
| 2 | Modélisation du problème robuste | 4 |
| 2.1 | Fonction objectif | 4 |
| 2.2 | Contraintes | 5 |
| 2.3 | Synthèse de la modélisation | 5 |
| 3 | Résolution par plans coupants | 6 |
| 4 | Résolution par dualisation | 8 |
| 4.1 | Dualisation de l'objectif | 8 |
| 4.2 | Dualisation des contraintes | 9 |
| 4.3 | Synthèse | 10 |
| 5 | Formulation non compacte du problème | 12 |
| 5.1 | Problème statique | 12 |
| 5.2 | Problème robuste | 12 |
| 5.3 | Résolution par génération de colonnes | 16 |
| 5.4 | Heuristique gloutonne | 19 |
| 6 | Résultats | 21 |
| 6.1 | Diagramme de performance | 21 |
| 6.2 | Tableau des résultats | 22 |

1 Modélisation du problème statique

Le problème statique consiste à partitionner les sommets d'un graphe (V, E) en au plus K parties de poids inférieur ou égal à B , de façon à minimiser le poids des arêtes à l'intérieur des parties.

On définit deux familles de variables binaires de la manière suivante :

- x_{ij} indique si l'arête $ij \in E$ entre les sommets i et j est sélectionnée.
- y_{ik} indique si le sommet $i \in V$ est assigné à la partie $k \in [K]$.

Le programme doit assurer les contraintes suivantes :

- Chaque sommet $i \in V$ doit être assigné à une unique partie, i-e, un unique élément de $[K]$:

$$\forall i \in V, \sum_{k=1}^K y_{ik} = 1$$

- La somme des poids des sommets dans une partie ne doit pas dépasser un certain poids maximal (ou budget) B :

$$\forall k \in [K], \sum_{i \in V} w_i y_{ik}$$

- Les membres d'une même partie doivent être reliés par une arête sélectionnée. Pour tout $ij \in E$ et $k \in [K]$, on a $x_{ij} \geq y_{ik}y_{jk}$. Inversement, si une arête est sélectionnée et qu'une de ses extrémités est dans une partie, alors il en est de même pour l'autre. Ainsi pour tout $ij \in E$ et $k \in [K]$, $y_{jk} \geq x_{ij}y_{ik}$. Comme nous travaillons avec des variables binaires, on peut linéariser ces contraintes :

$$x_{ij} \geq y_{ik} + y_{jk} - 1$$

$$y_{jk} \geq x_{ij} + y_{ik} - 1$$

$$y_{ik} \geq x_{ij} + y_{jk} - 1$$

On peut d'ailleurs remarquer que si on a : $\forall ij \in E, l_{ij} \geq 0$, on peut alors se contenter de la première de ces trois contraintes, puisqu'on travaille en minimisation. En effet, on est assuré (si le problème est faisable) qu'il existe une solution optimale avec

$$\forall i, j \in V, x_{ij} = 1 \Leftrightarrow (\exists k_0 \in [K], y_{ik_0} = y_{jk_0} = 1)$$

(D'ailleurs toutes les solutions optimales vérifient cela lorsque $l_{ij} > 0$). Pour une telle solution, on a bien le caractère suivant : $x_{ij} = 1$ si et seulement si i et j sont dans la même sous-partie.

Enfin, on cherche à minimiser la somme des longueurs des arêtes sélectionnées.

On aboutit donc au programme linéaire en variables 0-1 suivant :

$$\left\{ \begin{array}{ll} \min_{x,y} & \sum_{ij \in E} l_{ij} x_{ij} \\ \text{s.c.} & x_{ij} - y_{ik} - y_{jk} \geq -1 \quad \forall ij \in E, \forall k \in [K] \\ & \sum_{k=1}^K y_{ik} = 1 \quad \forall i \in V \\ & \sum_{i \in V} w_i y_{ik} \leq B \quad \forall k \in [K] \\ & x_{ij} \in \{0, 1\} \quad \forall ij \in E \\ & y_{ik} \in \{0, 1\} \quad \forall i \in V, \forall k \in [K] \end{array} \right.$$

Ce programme contient $|E| + nK$ variables et $n + (|E| + 1)K$ contraintes, et puisqu'une partition de V contient au plus n sous-ensembles, on peut toujours considérer $K \leq n$, donc la taille du programme est polynomiale en la taille du graphe.

2 Modélisation du problème robuste

Pour modéliser le problème robuste sous la forme d'un PLNE, on modifie la fonction objectif et les contraintes.

2.1 Fonction objectif

La fonction objectif évolue à cause des incertitudes sur les longueurs. On cherche à minimiser la fonction objectif dans le pire cas, c'est-à-dire :

$$\max_{l^1 \in \mathcal{U}^1} \sum_{ij \in E} x_{ij} l_{ij}^1 = \max_{\delta^1 \in \mathcal{U}^1} \sum_{ij \in E} x_{ij} (l_{ij} + \delta_{ij}^1 (\hat{l}_i + \hat{l}_j)) = \sum_{ij \in E} x_{ij} l_{ij} + \max_{\delta^1 \in \Delta \mathcal{U}^1} \sum_{ij \in E} \delta_{ij}^1 x_{ij} (\hat{l}_i + \hat{l}_j)$$

$$\text{où } \Delta \mathcal{U}^1 = \left\{ \delta^1 \in \mathbb{R}^{|E|} : \sum_{ij \in E} \delta_{ij}^1 \leq L; \forall ij \in E, \delta_{ij}^1 \in [0, 3] \right\}.$$

On remarque que $\Delta \mathcal{U}^1$ est un polytope de $\mathbb{R}^{|E|}$ défini par la contrainte $A\delta^1 \leq \hat{L}$ où :

$$A = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ -1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -1 \end{pmatrix}; \hat{L} = \begin{pmatrix} L \\ 3 \\ 3 \\ \vdots \\ 3 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

A est une matrice totalement unimodulaire : pour s'en convaincre, il suffit de remarquer que la ligne de 1 est totalement unimodulaire, puisque toute sous-matrice carrée de cette ligne est une matrice d'un seul élément égal à 1. De plus, \hat{L} est entier, donc $\Delta \mathcal{U}^1$ est un polytope entier, c'est-à-dire une enveloppe convexe de points entiers. On peut alors poser $\hat{\mathcal{U}}^1$ les points entiers de $\Delta \mathcal{U}^1$, on a donc :

$$\hat{\mathcal{U}}^1 = \Delta \mathcal{U}^1 \cap \mathbb{Z}^E = \left\{ \delta^1 \in \mathbb{N}^E : \sum_{ij \in E} \delta_{ij}^1 \leq L; \forall ij \in E, \delta_{ij}^1 \leq 3 \right\}$$

$$\Delta \mathcal{U}^1 = \text{Conv}(\hat{\mathcal{U}}^1)$$

$$\max_{l^1 \in \mathcal{U}^1} \sum_{ij \in E} x_{ij} l_{ij}^1 = \sum_{ij \in E} x_{ij} l_{ij} + \max_{\delta^1 \in \hat{\mathcal{U}}^1} \sum_{ij \in E} \delta_{ij}^1 x_{ij} (\hat{l}_i + \hat{l}_j)$$

2.2 Contraintes

La seule contrainte concernée par des incertitudes est la contrainte de poids maximal dans chaque partie. Des valeurs de x, y sont admissibles si et seulement si cette contrainte est respectée quelle que soit la réalisation de l'incertitude sur les poids. Soit $k \in [K]$. La contrainte $\sum_{i \in V} w_i y_{ik} \leq B$ devient :

$$\bigwedge_{w^2 \in \hat{\mathcal{U}}^2} \left[\sum_{i \in V} w_i^2 y_{ik} \leq B \right]$$

i-e,

$$\bigwedge_{\delta^2 \in \hat{\mathcal{U}}^2} \left[\sum_{i \in V} w_i y_{ik} (1 + \delta_i^2) \leq B \right]$$

$$\text{où } \hat{\mathcal{U}}^2 = \left\{ \delta^2 \in \mathbb{R}^V : \sum_{i \in V} \delta_i^2 \leq W; \forall i \in V, \delta_i^2 \in [0, W_i] \right\}.$$

2.3 Synthèse de la modélisation

En utilisant les ensembles $\hat{\mathcal{U}}^1$ et $\hat{\mathcal{U}}^2$ qu'on vient de construire et qui sont de taille finie, on modélise le problème robuste de la manière suivante :

$$\left\{ \begin{array}{ll} \min_{x,y} & \sum_{ij \in E} x_{ij} l_{ij} + \max_{\delta^1 \in \hat{\mathcal{U}}^1} \sum_{ij \in E} \delta_{ij}^1 x_{ij} (\hat{l}_i + \hat{l}_j) \\ \text{s.c.} & x_{ij} - y_{ik} - y_{jk} \geq -1 \quad \forall ij \in E, \forall k \in [K] \\ & \sum_{k=1}^K y_{ik} = 1 \quad \forall i \in V \\ & \sum_{i \in V} w_i y_{ik} (1 + \delta_i^2) \leq B \quad \forall k \in [K], \forall \delta^2 \in \hat{\mathcal{U}}^2 \\ & x_{ij} \in \{0, 1\} \quad \forall ij \in E \\ & y_{ik} \in \{0, 1\} \quad \forall i \in V, \forall k \in [K] \end{array} \right.$$

3 Résolution par plans coupants

On peut ré-écrire le problème sous la forme suivante, en introduisant la variable z qui correspond à la "partie robuste" de l'objectif, et en exhibant la partie des contraintes sur le poids qui dépend de l'incertitude δ^2 .

$$\left\{ \begin{array}{ll} \min_{x,y,z} & \sum_{ij \in E} x_{ij} l_{ij} + z \\ \text{s.c.} & \begin{aligned} z &\geq \sum_{ij \in E} \delta_{ij}^1 x_{ij} (\hat{l}_i + \hat{l}_j) & \forall \delta^1 \in \hat{\mathcal{U}}^1 \\ \sum_{i \in V} w_i y_{ik} \delta_i^2 &\leq B - \sum_{i \in V} w_i y_{ik} & \forall k \in [K], \forall \delta^2 \in \hat{\mathcal{U}}^2 \\ x_{ij} - y_{ik} - y_{jk} &\geq -1 & \forall ij \in E, \forall k \in [K] \\ \sum_{k=1}^K y_{ik} &= 1 & \forall i \in V \\ x_{ij} &\in \{0, 1\} & \forall ij \in E \\ y_{ik} &\in \{0, 1\} & \forall i \in V, \forall k \in [K] \end{aligned} \end{array} \right.$$

Puisque $\hat{\mathcal{U}}^1$ et $\hat{\mathcal{U}}^2$ sont de taille exponentielle en L , on ne peut pas écrire toutes ces contraintes explicitement dans le modèle informatique. On peut alors utiliser une résolution par plans coupants, en ne considérant qu'une sous-partie $\mathcal{U}^{1*} \subset \hat{\mathcal{U}}^1$ et $\mathcal{U}^{2*} \subset \hat{\mathcal{U}}^2$ de ces contraintes, que l'on étend au cours de la résolution. Par exemple, on choisit initialement :

- $\mathcal{U}^{1*} = \{\min(3, L) \times e_{ij} : ij \in E\}$ l'ensemble des δ^1 qui mettent chacun un poids de 3 ou L (suivant quelle grandeur est la plus petite) sur exactement une des arêtes.
- $\mathcal{U}^{2*} = \{W_v \times e_v : v \in V\}$ l'ensemble des δ^2 qui mettent chacun un poids de W_v sur le sommet correspondant (on suppose pour cela $\forall v \quad W_v \leq W$)

Soit $(\tilde{x}, \tilde{y}, \tilde{z})$ la solution courante du problème maître. On définit deux types de sous-problèmes de séparation qui vont nous amener à ajouter des plans coupants (i.e. ajouter des contraintes au problème maître en étendant les ensembles \mathcal{U}^{1*} et \mathcal{U}^{2*}) :

$$\begin{aligned} V_l^{SP} := \max_{\delta^1} & \sum_{ij \in E} \delta_{ij}^1 \tilde{x}_{ij} (\hat{l}_i + \hat{l}_j) \\ \text{s.c.} & \sum_{ij \in E} \delta_{ij}^1 \leq L \\ & \delta_{ij}^1 \in [0, 3] \quad \forall ij \in E \end{aligned}$$

$$\begin{aligned}
V_w^{SP}(k) := \max_{\delta^2} \quad & \sum_{i \in V} w_i \tilde{y}_{ik} \delta_i^2 \\
\text{s.c.} \quad & \sum_{v \in V} \delta_v^2 \leq W \\
& \delta_v^2 \in [0, W_v] \quad \forall v \in V
\end{aligned}$$

Une solution $(\tilde{x}, \tilde{y}, \tilde{z})$ du problème maître est alors optimale si et seulement si $V_l^{SP} \leq \tilde{z}$ et $\forall k \in [K], V_w^{SP}(k) \leq B - \sum_{i \in V} w_i \tilde{y}_{ik}$, c'est-à-dire :

- On ne peut pas trouver de $\delta^1 \notin \mathcal{U}^{1*}$ qui empêche la solution courante
- La solution courante vérifie la contrainte de poids pour toutes les parties, même pour les $\delta^2 \notin \mathcal{U}^{2*}$.

Notons alors $\tilde{\delta}^1$ (resp. $\tilde{\delta}^2(k)$) une solution optimale de ce premier problème de séparation (resp. du problème de séparation sur le poids de la partie k). Lorsque les conditions d'optimalité ne sont pas remplies, les plans coupants ajoutés correspondent alors aux contraintes pour les nouveaux $\tilde{\delta}^1, (\tilde{\delta}^2(k))_{k \in [K]}$ renvoyés par les problèmes de séparation.

- Si $V_l^{SP} > \tilde{z}$, on ajoute au problème maître la coupe :

$$z \geq \sum_{ij \in E} x_{ij} \tilde{\delta}_{ij}^1 (\hat{l}_i + \hat{l}_j)$$

- S'il existe $k_0 \in [K]$ tel que $V_w^{SP}(k_0) > B - \sum_{i \in V} w_i \tilde{y}_{ik_0}$, on ajoute au problème maître toutes les coupes de la forme :

$$\sum_{i \in V} y_{ik} w_i (1 + \tilde{\delta}_i^2(k_0)) \leq B \quad \forall k \in [K]$$

(Puisque l'assignation des sommets aux parties est toujours à une permutation des parties près, il faut ajouter la coupe pour toutes les $k \in [K]$ afin d'éviter de juste "déplacer" l'infaisabilité vers un $k \neq k_0$ à l'itération suivante).

4 Résolution par dualisation

4.1 Dualisation de l'objectif

On rappelle le problème :

$$\min_{(x,y) \in \mathcal{X}} f(x) = \min_{(x,y) \in \mathcal{X}} \left[\sum_{ij \in E} x_{ij} l_{ij} + \max_{\delta^1 \in \hat{\mathcal{U}}^1} \sum_{ij \in E} \delta_{ij}^1 x_{ij} (\hat{l}_i + \hat{l}_j) \right]$$

où $\hat{\mathcal{U}}^1 = \Delta \mathcal{U}^1 \cap \mathbb{Z}^E = \left\{ \delta^1 \in \mathbb{N}^E : \sum_{ij \in E} \delta_{ij}^1 \leq L; \forall ij \in E, \delta_{ij}^1 \leq 3 \right\}$ et \mathcal{X} est l'ensemble des solutions (x, y) admissibles du problème robuste.

On a montré plus haut $\forall x, \max_{\delta^1 \in \hat{\mathcal{U}}^1} \sum_{ij \in E} \delta_{ij}^1 x_{ij} (\hat{l}_i + \hat{l}_j) = \max_{\delta^1 \in \Delta \mathcal{U}^1} \sum_{ij \in E} \delta_{ij}^1 x_{ij} (\hat{l}_i + \hat{l}_j)$, d'où

$\forall x, f(x) = \sum_{ij \in E} x_{ij} l_{ij} + V_P^1$ avec :

$$\begin{aligned} V_P^1 &:= \max_{\delta^1} \sum_{ij \in E} \delta_{ij}^1 x_{ij} (\hat{l}_i + \hat{l}_j) \\ \text{s.c. } &\sum_{ij \in E} \delta_{ij}^1 \leq L \\ &\delta_{ij}^1 \in [0, 3] \quad \forall ij \in E \end{aligned}$$

On dualise ce problème en associant la variable z à la contrainte $\sum_{ij \in E} \delta_{ij}^1 \leq L$ et les variables $(\eta_{ij})_{ij \in E}$ aux contraintes $\forall ij \in E, \delta_{ij}^1 \leq 3$. On obtient :

$$\begin{aligned} V_D^1 &:= \min_{z, \eta} Lz + \sum_{ij \in E} 3\eta_{ij} \\ \text{s.c. } &z + \eta_{ij} \geq x_{ij} (\hat{l}_i + \hat{l}_j) \quad \forall ij \in E \\ &\eta_{ij} \geq 0 \quad \forall ij \in E \\ &z \geq 0 \end{aligned}$$

Par dualité forte, $V_P^1 = V_D^1$. Le problème devient donc :

$$\min_{(x,y) \in \mathcal{X}} f(x) = \min_{(x,y) \in \mathcal{X}} \left[\sum_{ij \in E} x_{ij} l_{ij} + \min_{(z,\eta) \in \mathcal{Z}_x} Lz + \sum_{ij \in E} 3\eta_{ij} \right] = \min_{\substack{(x,y) \in \mathcal{X} \\ (z,\eta) \in \mathcal{Z}_x}} Lz + \sum_{ij \in E} (x_{ij} l_{ij} + 3\eta_{ij})$$

$$\text{avec } \mathcal{Z}_x = \left\{ (z, \eta) \in \mathbb{R}_+ \times (\mathbb{R}_+)^E : \forall ij \in E, z + \eta_{ij} \geq x_{ij}(\hat{l}_i + \hat{l}_j) \right\}$$

4.2 Dualisation des contraintes

On rappelle les contraintes :

$$\forall k \in [K], \forall \delta^2 \in \hat{\mathcal{U}}^2, \sum_{i \in V} w_i y_{ik} (1 + \delta_i^2) \leq B$$

$$\text{où } \hat{\mathcal{U}}^2 = \left\{ \delta^2 \in \mathbb{R}^V : \sum_{i \in V} \delta_i^2 \leq W; \forall i \in V, \delta_i^2 \in [0, W_i] \right\}.$$

Soit $k \in [K]$. Pour une solution y donnée, les contraintes sont satisfaites quel que soit $\delta^2 \in \hat{\mathcal{U}}^2$ si et seulement si :

$$\left[\max_{\delta^2 \in \hat{\mathcal{U}}^2} \sum_{i \in V} w_i y_{ik} (1 + \delta_i^2) \right] \leq B$$

qui équivaut à :

$$\sum_{i \in V} w_i y_{ik} + \max_{\delta^2 \in \hat{\mathcal{U}}^2} \sum_{i \in V} w_i y_{ik} \delta_i^2 \leq B$$

i-e $\sum_{i \in V} w_i y_{ik} + V_k^2 \leq B$ où :

$$\begin{aligned} V_k^2 &:= \max_{\delta^2} \quad \sum_{i \in V} w_i y_{ik} \delta_i^2 \\ \text{s.c.} \quad &\sum_{i \in V} \delta_i^2 \leq W \\ &\delta_i^2 \in [0, W_i] \quad \forall i \in V \end{aligned}$$

On dualise ce problème en associant la variable t à la contrainte $\sum_{i \in V} \delta_i^2 \leq W$ et les variables $(\xi_i)_{i \in V}$ aux contraintes $\forall i \in V, \delta_i^2 \leq W_i$. La matrice de contraintes étant la même que lors de la dualisation de la fonction objectif, on peut établir en utilisant les mêmes arguments :

$$\begin{aligned}
V_k^2 := \min_{t, \xi} \quad & Wt + \sum_{i \in V} W_i \xi_i \\
\text{s.c.} \quad & t + \xi_i \geq w_i y_{ik} \quad \forall i \in V \\
& \xi_i \geq 0 \quad \forall i \in V \\
& t \geq 0
\end{aligned}$$

D'où :

$$\sum_{i \in V} w_i y_{ik} + V_k^2 \leq B \iff \exists (t, \xi) \in \mathcal{T}_y^k, Wt + \sum_{i \in V} (w_i y_{ik} + W_i \xi_i) \leq B$$

où $\mathcal{T}_y^k = \{(t, \xi) \in \mathbb{R}_+ \times (\mathbb{R}_+)^V : \forall i \in V, t + \xi_i \geq w_i y_{ik}\}$.

La contrainte $\forall k \in [K], \forall \delta^2 \in \hat{\mathcal{U}}^2, \sum_{i \in V} w_i y_{ik} (1 + \delta_i^2) \leq B$ peut donc être remplacée par l'ajout de variables positives $(t^k, \xi^k)_{k \in [K]}$ et des contraintes :

- $\forall k \in [K], (t^k, \xi^k) \in \mathcal{T}_y^k$ (i.e., $\forall k \in [K], \forall i \in V, t^k + \xi_i^k \geq w_i y_{ik}$)
- $\forall k \in [K], Wt^k + \sum_{i \in V} (w_i y_{ik} + W_i \xi_i^k) \leq B$.

4.3 Synthèse

On rappelle la formulation du problème robuste :

$$\left\{
\begin{array}{ll}
\min_{x, y} & \sum_{ij \in E} x_{ij} l_{ij} + \max_{\delta^1 \in \hat{\mathcal{U}}^1} \sum_{ij \in E} \delta_{ij}^1 x_{ij} (\hat{l}_i + \hat{l}_j) \\
\text{s.c.} & x_{ij} - y_{ik} - y_{jk} \geq -1 \quad \forall ij \in E, \forall k \in [K] \\
& \sum_{k=1}^K y_{ik} = 1 \quad \forall i \in V \\
& \sum_{i \in V} w_i y_{ik} (1 + \delta_i^2) \leq B \quad \forall k \in [K], \forall \delta^2 \in \hat{\mathcal{U}}^2 \\
& x_{ij} \in \{0, 1\} \quad \forall ij \in E \\
& y_{ik} \in \{0, 1\} \quad \forall i \in V, \forall k \in [K]
\end{array}
\right.$$

Avec les résultats précédents, ce programme devient :

$$\left\{
 \begin{array}{ll}
 \min_{x,y,z,t,\eta,\xi} & Lz + \sum_{ij \in E} (x_{ij}l_{ij} + 3\eta_{ij}) \\
 \text{s.c.} & \\
 & x_{ij} - y_{ik} - y_{jk} \geq -1 \quad \forall ij \in E, \forall k \in [K] \\
 & \sum_{k=1}^K y_{ik} = 1 \quad \forall i \in V \\
 & z + \eta_{ij} \geq x_{ij}(\hat{l}_i + \hat{l}_j) \quad \forall ij \in E \\
 & Wt^k + \sum_{i \in V} (w_i y_{ik} + W_i \xi_i^k) \leq B \quad \forall k \in [K] \\
 & t^k + \xi_i^k \geq w_i y_{ik} \quad \forall i \in V, \forall k \in [K] \\
 & x_{ij} \in \{0, 1\} \quad \forall ij \in E \\
 & y_{ik} \in \{0, 1\} \quad \forall i \in V, \forall k \in [K] \\
 & z \geq 0 \\
 & \eta_{ij} \geq 0 \quad \forall ij \in E \\
 & t^k \geq 0 \quad \forall k \in [K] \\
 & \xi_i^k \geq 0 \quad \forall i \in V, \forall k \in [K]
 \end{array}
 \right.$$

5 Formulation non compacte du problème

5.1 Problème statique

On tente de construire une formulation non compacte du problème, en considérant des variables de décision correspondant à chaque partie de V possible.

On note $\mathcal{P} = \left\{ P \in \mathcal{P}(V) : \sum_{i \in P} w_i \leq B \right\}$. Pour chaque partie $P \in \mathcal{P}$, on note $E(P)$ l'ensemble des arêtes reliant deux sommets dans P .

On associe à chaque $P \in \mathcal{P}$ une variable binaire z_P indiquant si la partie P est utilisée lors de la partition. Ces variables sont soumises aux contraintes suivantes :

- Au plus K parties doivent être sélectionnées :

$$\sum_{P \in \mathcal{P}} z_P \leq K$$

- Chaque sommet $i \in V$ doit appartenir à une seule partie sélectionnée :

$$\forall i \in V, \sum_{P \ni i} z_P = 1$$

Chaque partie $P \in \mathcal{P}$ a un coût $l_P := \sum_{ij \in E(P)} l_{ij}$.

Le problème statique se modélise donc de la manière suivante :

$$\left\{ \begin{array}{ll} \min_z & \sum_{P \in \mathcal{P}} l_P z_P \\ \text{s.c.} & \sum_{P \in \mathcal{P}} z_P \leq K \\ & \sum_{P \ni i} z_P = 1 \quad \forall i \in V \\ & z_P \in \{0, 1\} \quad \forall P \in \mathcal{P} \end{array} \right.$$

Comme les l_P sont positifs, on peut relâcher

5.2 Problème robuste

L'incertitude sur les budgets réduit l'ensemble des parties possibles. Notons $\hat{\mathcal{P}}$ ce nouvel ensemble. Il va de soi qu'une partie admissible avec les incertitudes l'était nécessairement auparavant, donc $\hat{\mathcal{P}} \subset \mathcal{P}$. De plus, une partie $P \in \mathcal{P}$ est admissible si et seulement si :

$$\forall \delta^2 \in \Delta \mathcal{U}^2, \sum_{i \in P} w_i (1 + \delta_i^2) \leq B$$

où

$$\Delta\mathcal{U}^2 = \left\{ \delta^2 \in \mathbb{R}^V : \sum_{i \in V} \delta_i^2 \leq W; \forall i \in V, \delta_i^2 \in [0, W_i] \right\}$$

Notons pour $P \in \mathcal{P}$, $B(P) = B - \sum_{i \in P} w_i$. En remarquant que les incertitudes de poids peuvent être concentrées sur les sommets de P , on écrit :

$$P \in \hat{\mathcal{P}} \iff \max_{\delta_i^2 \in \Delta\mathcal{U}_P^2} \sum_{i \in P} w_i \delta_i^2 \leq B(P)$$

avec

$$\Delta\mathcal{U}_P^2 = \left\{ \delta^2 \in \mathbb{R}^P : \sum_{i \in P} \delta_i^2 \leq W; \forall i \in P, \delta_i^2 \in [0, W_i] \right\}$$

Soient $P \in \mathcal{P}$, puis $\delta^2 \in \Delta\mathcal{U}_P^2$ tel que $\sum_{i \in P} w_i \delta_i^2$ est maximal. Alors pour toute paire de sommets $j, k \in P$ tels que $w_j > w_k$, on a $\delta_j^2 = W_j$ ou $\delta_k^2 = 0$. Le corollaire de cette propriété est que si les w_i sont tous différents, alors au plus un des coefficients δ_i^2 est dans l'intervalle ouvert $]0, W_i[$.

Par conséquent, si les w_i sont triés, la valeur $V_P^2 := \max_{\delta_i^2 \in \Delta\mathcal{U}_P^2} \sum_{i \in P} w_i \delta_i^2$ est calculable en $O(|P|)$. Il est ensuite possible de vérifier si V_P^2 est inférieur à $B(P)$.

Algorithm 1: Vérifier si $P \in \hat{\mathcal{P}}$

Input: Un budget B , une incertitude maximale W , et une partie $P \in \mathcal{P}$ définie par les couples poids-incertitude (w_i, W_i) rangés par w_i décroissant.

Output: Si $P \in \hat{\mathcal{P}}$

$$B(P) \leftarrow B - \sum_{i \in P} w_i$$

$$V \leftarrow 0$$

$$\Delta \leftarrow 0$$

// On calcule V_P^2 .

if $\sum_{i \in P} W_i > W$ then

$i \leftarrow$ le premier indice $i \in P$

while $\Delta + W_i \leq W$ **do**

$V \leftarrow V + w_i W_i$

$\Delta \leftarrow \Delta + W_i$

$i \leftarrow$ l'indice suivant dans P

end

$V \leftarrow V + w_i(W - \Delta)$

else

$V \leftarrow \sum_{i \in P} w_i W_i$

end

end

return $V \leq B(P)$ // On vérifie si V_P^2 est inférieur à $B(P)$.

On se réduit à cet ensemble de parties par la suite.

Le reste de l'incertitude dans le problème robuste est définie par l'ensemble $\Delta\mathcal{U}^1$ d'incertitudes sur les longueurs :

$$\Delta\mathcal{U}^1 = \left\{ \delta^1 \in \mathbb{R}^{|E|} : \sum_{ij \in E} \delta_{ij}^1 \leq L; \forall ij \in E, \delta_{ij}^1 \in [0, 3] \right\}$$

Soit $P \in \hat{\mathcal{P}}$ et $\delta^1 \in \Delta\mathcal{U}^1$ une incertitude sur les longueurs. Calculons l'incertitude sur l_P :

$$\begin{aligned}
l_P^1 &= \sum_{ij \in E(P)} l_{ij}^1 \\
&= \sum_{ij \in E(P)} \left[l_{ij} + \delta_{ij}^1 (\hat{l}_i + \hat{l}_j) \right] \\
&= l_P + \sum_{ij \in E(P)} \delta_{ij}^1 (\hat{l}_i + \hat{l}_j)
\end{aligned}$$

On aboutit donc à une formulation suivante du problème robuste, non compacte :

$$\left\{
\begin{array}{ll}
\min_z & \sum_{P \in \hat{\mathcal{P}}} z_P l_P + \max_{\delta^1 \in \Delta \mathcal{U}^1} \sum_{P \in \hat{\mathcal{P}}} z_P \sum_{ij \in E(P)} \delta_{ij}^1 (\hat{l}_i + \hat{l}_j) \\
\text{s.c.} & \sum_{P \in \hat{\mathcal{P}}} z_P \leq K \\
& \sum_{P \in \hat{\mathcal{P}}: i \in P} z_P = 1 \quad \forall i \in V \\
& z_P \in \{0, 1\} \quad \forall P \in \hat{\mathcal{P}}
\end{array}
\right.$$

Soit z un vecteur binaire sur $\hat{\mathcal{P}}$ tel que $\forall i \in V, \sum_{P \in \hat{\mathcal{P}}: i \in P} z_P = 1$. On pose :

$$f(z) := \max_{\delta^1 \in \Delta \mathcal{U}^1} \sum_{P \in \hat{\mathcal{P}}} z_P \sum_{ij \in E(P)} \delta_{ij}^1 (\hat{l}_i + \hat{l}_j)$$

Les ensembles P tels que $z_P = 1$ sont disjoints. De même, les ensembles $E(P)$ tels que $z_P = 1$ sont aussi disjoints. C'est ce qui nous permet d'écrire :

$$\begin{aligned}
f(z) &= \max_{\delta^1 \in \Delta \mathcal{U}^1} \sum_{P \in \hat{\mathcal{P}}} \sum_{\substack{ij \in E(P) \\ z_P=1}} \delta_{ij}^1 (\hat{l}_i + \hat{l}_j) \\
&\quad \sum_{ij \in E} \delta_{ij}^1 \leq L \\
&\quad \delta_{ij}^1 \in [0, 3] \quad \forall P / z_P = 1, \forall ij \in E(P)
\end{aligned}$$

Par dualité forte, on obtient :

$$\begin{aligned}
f(z) = & \min_{\alpha, \beta} L\alpha + \sum_{P \in \hat{\mathcal{P}}} \sum_{\substack{ij \in E(P) \\ z_P=1}} 3\beta_{ij} \\
& \alpha + \beta_{ij} \geq (\hat{l}_i + \hat{l}_j) \quad \forall P / z_P = 1, \forall ij \in E(P) \\
& \alpha \geq 0, \beta_{ij} \geq 0
\end{aligned}$$

Puis ensuite, en ajoutant les contraintes pour les parties telles que $z_P = 0$:

$$\begin{aligned}
f(z) = & \min_{\alpha, \beta} L\alpha + \sum_{ij \in E} 3\beta_{ij} \\
& \alpha + \beta_{ij} \geq z_P (\hat{l}_i + \hat{l}_j) \quad \forall P, \forall ij \in E(P) \\
& \alpha \geq 0, \beta_{ij} \geq 0
\end{aligned}$$

On aboutit à la formulation suivante :

$$(P_{NC}) : \left\{
\begin{array}{ll}
\min_z & L\alpha + \sum_{P \in \hat{\mathcal{P}}} z_P l_P + \sum_{ij \in E} 3\beta_{ij} \\
\text{s.c.} & \sum_{P \in \hat{\mathcal{P}}} z_P \leq K \\
& \sum_{\substack{P \in \hat{\mathcal{P}} \\ i \in P}} z_P = 1 \quad \forall i \in V \\
& \alpha + \beta_{ij} \geq (\hat{l}_i + \hat{l}_j) \sum_{\substack{P \in \hat{\mathcal{P}} \\ ij \in E(P)}} z_P \quad \forall ij \in E \\
& \alpha \geq 0 \\
& \beta_{ij} \geq 0 \quad \forall ij \in E \\
& z_P \in \{0, 1\} \quad \forall P \in \hat{\mathcal{P}}
\end{array}
\right.$$

5.3 Résolution par génération de colonnes

Puisque l'ensemble $\hat{\mathcal{P}}$ est a priori de taille exponentielle en la taille de l'instance, on choisit de procéder à une résolution du problème par génération de colonnes.

Pour un ensemble de parties candidates $C \subset \hat{\mathcal{P}}$, on définit le problème maître :

$$(MP_C) : \left\{ \begin{array}{ll} \min_z & L\alpha + \sum_{P \in C} z_P l_P + \sum_{ij \in E} 3\beta_{ij} \\ \text{s.c.} & \sum_{P \in C} z_P \leq K \\ & \sum_{\substack{P \in C \\ i \in P}} z_P = 1 \quad \forall i \in V \\ & \alpha + \beta_{ij} \geq (\hat{l}_i + \hat{l}_j) \sum_{\substack{P \in C \\ ij \in E(P)}} z_P \quad \forall ij \in E \\ & \alpha \geq 0 \\ & \beta_{ij} \geq 0 \quad \forall ij \in E \\ & z_P \in \{0, 1\} \quad \forall P \in C \end{array} \right.$$

On cherche à déterminer le problème de *pricing*, qui va nous permettre d'ajouter des parties à C pour améliorer la solution. Pour cela, on dualise la relaxation continue de (P_{NC}) , ce qui donne :

$$(D_{NC}) : \left\{ \begin{array}{ll} \max_{\theta, \lambda, \mu} & K\theta + \sum_{i \in V} \lambda_i \\ \text{s.c.} & \sum_{P \in \hat{\mathcal{P}}} \sum_{ij \in E(P)} \mu_{ij} \leq L \\ & \sum_{\substack{P \in \hat{\mathcal{P}} \\ ij \in E(P)}} \mu_{ij} \leq 3 \quad \forall ij \in E \\ & \theta + \sum_{i \in P} \lambda_i \leq l_P + \sum_{ij \in E(P)} \mu_{ij} (\hat{l}_i + \hat{l}_j) \quad \forall P \in \hat{\mathcal{P}} \\ & \theta \leq 0 \\ & \lambda_i \in \mathbb{R} \quad \forall i \in V \\ & \mu_{ij} \geq 0 \quad \forall ij \in E \end{array} \right.$$

Soit C un ensemble de parties candidates, (P_C) le relâché continu de (MP_C) et (θ, λ, μ) la solution optimale du dual de (P_C) .

Le problème de *pricing* consiste à trouver la partie $P \notin C$ telle que la contrainte correspondante dans (D_{NC}) est la plus violée dans la solution courante, c'est-à-dire :

$$\operatorname{argmax}_{P \notin C} \theta + \sum_{i \in P} \lambda_i - \sum_{ij \in E(P)} \mu_{ij} (\hat{l}_i + \hat{l}_j) - l_P$$

On peut se débarrasser de θ , on cherche donc :

$$\operatorname{argmax}_{P \notin C} \sum_{i \in P} \lambda_i - \sum_{ij \in E(P)} (l_{ij} + \mu_{ij} (\hat{l}_i + \hat{l}_j))$$

On peut écrire ce problème de pricing comme un problème quadratique de sélection de sommets dans V :

$$(Q - Price) : \left\{ \begin{array}{ll} \max_x & \sum_{i \in V} x_i \lambda_i - \sum_{ij \in E} x_i x_j (l_{ij} + \mu_{ij} (\hat{l}_i + \hat{l}_j)) \\ \text{s.c.} & \quad \quad \quad \text{(contraintes de budget robuste)} \\ & x_i \in \{0, 1\} \end{array} \right. \quad \forall i \in V$$

La contrainte de budget s'écrit $\sum_{x \in V} w_i x_i (1 + \delta_i) \leq B, \forall \delta \in \hat{\mathcal{U}}^2$,

$$\text{où } \hat{\mathcal{U}}^2 = \left\{ \delta^2 \in \mathbb{N}^V : \sum_{i \in V} \delta_i^2 \leq W; \forall i \in V, \delta_i^2 \leq W_i \right\}$$

Comme lors de la dualisation des contraintes plus haut, ces contraintes indiquées sur $\hat{\mathcal{U}}^2$ peuvent être remplacées par l'ajout de variables positives $t, (\xi_i)_{i \in V}$ et des contraintes :

- $\forall i \in V, t + \xi_i \geq w_i x_i$.
- $Wt + \sum_{i \in V} (w_i x_i + W_i \xi_i) \leq B$.

D'où :

$$(Q - Price') : \left\{ \begin{array}{ll} \max_x & \sum_{i \in V} x_i \lambda_i - \sum_{ij \in E} x_i x_j (l_{ij} + \mu_{ij} (\hat{l}_i + \hat{l}_j)) \\ \text{s.c.} & \begin{array}{ll} w_i x_i \leq t + \xi_i & \forall i \in V \\ \sum_{i \in V} w_i x_i \leq B - Wt - \sum_{i \in V} W_i \xi_i & \\ \xi_i \geq 0 & \forall i \in V \\ t \geq 0 & \\ x_i \in \{0, 1\} & \forall i \in V \end{array} \end{array} \right.$$

ou encore, après linéarisation :

$$(L - Price) : \left\{ \begin{array}{ll} \max_x & \sum_{i \in V} x_i \lambda_i - \sum_{ij \in E} y_{ij} (l_{ij} + \mu_{ij} (\hat{l}_i + \hat{l}_j)) \\ \text{s.c.} & \begin{array}{ll} w_i x_i \leq t + \xi_i & \forall i \in V \\ \sum_{i \in V} w_i x_i \leq B - Wt - \sum_{i \in V} W_i \xi_i & \\ y_{ij} \geq x_i + x_j - 1 & \forall ij \in E \\ \xi_i \geq 0 & \forall i \in V \\ t \geq 0 & \\ x_i \in \{0, 1\} & \forall i \in V \\ y_{ij} \geq 0 & \forall ij \in E \end{array} \end{array} \right.$$

Une fois la génération de colonnes terminée, on résout la formulation intégrale de (MP_C) pour obtenir une solution admissible du problème. On a une borne inférieure de la valeur optimale du programme. En effet, à chaque étape de la génération de colonnes, notons :

$$R = \max_{P \notin C} \left[\theta + \sum_{i \in P} \lambda_i - \sum_{ij \in E(P)} \mu_{ij} (\hat{l}_i + \hat{l}_j) - l_P \right]$$

la valeur du coût réduit trouvé lors du pricing. Alors $(\theta - R, \lambda, \mu)$ est une solution admissible de (D_{NC}) .

Notre borne inférieure est donc $\text{val}(P_C) - KR$, où (P_C) est le relâché continu de (MP_C) .

5.4 Heuristique gloutonne

Afin de démarrer la résolution par génération de colonnes, il faut d'abord trouver une partition admissible en K parties de V , qui constitueront les premiers éléments de C .

On développe une heuristique gloutonne qui renvoie toujours une partition admissible. On démarre en triant les éléments de V dans l'ordre décroissant de poids w_i et en définissant K parties vides de V .

L'intuition est la suivante : dans chaque partie, l'incertitude sur les poids s'applique dans le pire cas aux plus gros éléments de la partie. Par conséquent, lorsqu'on traite les plus gros éléments de V , on cherche une partie qui peut l'accueillir sans dépasser sa contrainte de budget, mais qui a déjà épuisé son quota maximal d'incertitude W sur les éléments précédents.

On choisit donc la méthode suivante : on assigne chacun à leur tour les éléments de V à la partie non remplie (au sens robuste) avec le plus d'*incertitude utilisée*. Cette grandeur commence à 0 pour chaque partie, et augmente de W_i quand un élément de "quota d'incertitude" W_i est ajouté, sauf si le montant total dépasse W .

Algorithm 2: Construire une partition de V en parties admissibles pour le problème robuste

Input: Un budget B , une incertitude maximale W , un nombre de parties K et les éléments de V : v_1, v_2, \dots, v_n rangés par w_i décroissant.

Output: Une partition C de V constituée de K parties satisfaisant les contraintes robustes de poids.

```

 $C \leftarrow [\emptyset \text{ pour } i \text{ allant de } 1 \text{ à } K]$ 
 $B_u \leftarrow [0 \text{ pour } i \text{ allant de } 1 \text{ à } K] \text{ // Poids de chaque partie}$ 
 $I_u \leftarrow [0 \text{ pour } i \text{ allant de } 1 \text{ à } K] \text{ // Incertitude utilisée dans chaque partie}$ 
for  $i$  allant de 1 à  $K$  do
     $j \leftarrow$  la partie dans  $[1, K]$  telle que  $I_u[j]$  est maximale
     $\Delta \leftarrow \min(W_i, I_u[j]) \text{ // Combien d'incertitude va être dépensée sur ce sommet ?}$ 
    if  $B_u[j] + w_i(1 + \Delta) \leq W$  then
        // Ajouter le sommet à la partie  $j$  et mettre à jour
         $C[j] \leftarrow C[j] \cup i$ 
         $B_u[j] \leftarrow B_u[j] + w_i(1 + \Delta)$ 
         $I_u[j] \leftarrow I_u[j] + \Delta$ 
    else
        // Ajouter le sommet  $i$  ferait dépasser la contrainte de budget.
        Prendre l'indice  $j$  suivant, jusqu'à trouver une partie où on peut ajouter le sommet  $i$ 
    end
end
end
return  $C$ 
```

6 Résultats

Dans tout ce qui suit, nous nous intéresserons qu'aux instances que nous avons pu résoudre en moins de 10 minutes de temps de calcul, ce qui nous limite malheureusement à une assez petite fraction de toutes les instances proposées.

6.1 Diagramme de performance

Le diagramme de performance suivant nous montre plusieurs choses :

- Ajouter la robustesse au problème rend effectivement le problème plus difficile à résoudre, comme on s'y attend.
- Parmi les méthodes exactes, la plus performante semble être la méthode de traitement de la robustesse par dualisation, qui a également l'avantage de pouvoir parfois fournir une solution admissible même si la limite de temps est atteinte - là où les méthodes par plans coupants ne garantissent l'admissibilité qu'une fois la résolution terminée et optimale.
- Notre implémentation de la méthode dite "Branch & Cut" de plans coupants ajoutés par callbacks comporte très probablement encore un bug, car on s'attendrait à ce que ses performances soient meilleures que celle des plans coupants implémentés "naïvement"

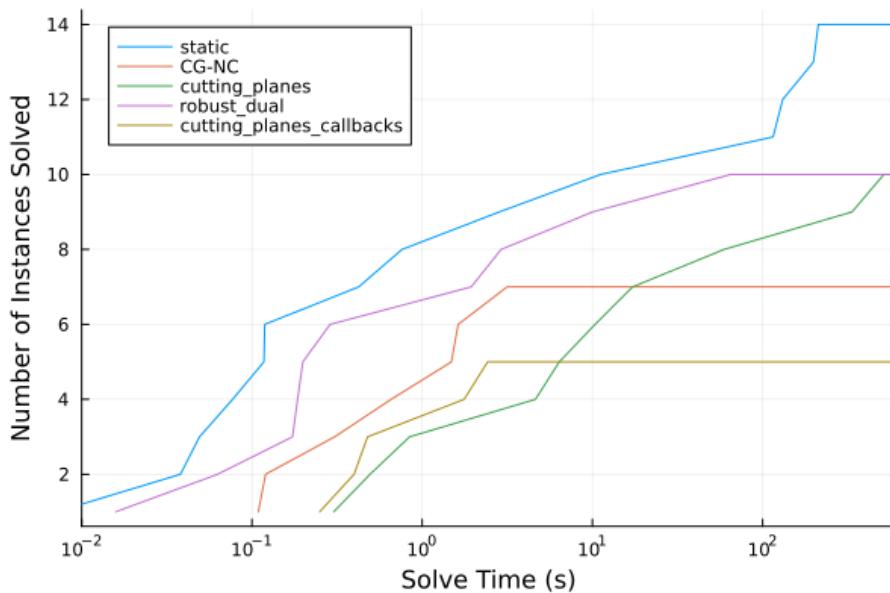


FIG. 1 – Diagramme de performance pour un temps de résolution maximal de 10 min.

static = Résolution du problème statique

cutting_planes = Algorithme de plans coupants "naïf"

cutting_planes_callbacks = Plans coupants + Callbacks

robust_dual = Dualisation

CG-NC = Formulation non-compacte et génération de colonnes

Finalement, on notera également que, pour la méthode heuristique basée sur la formulation non-compacte, nous ne comptons comme "résolue" une instance que si la solution renvoyée est effectivement équivalente aux solutions des autres méthodes. Cependant, comme on le verra dans le tableau complet, cette méthode passe un peu mieux à l'échelle, et permet donc d'obtenir des solutions admissibles (mais sans garanties d'optimalité) pour des tailles d'instance plus grandes que celles pour lesquelles les méthodes compactes convergent.

6.2 Tableau des résultats

| Instance | PR | Plans coupants | | + Callbacks | | Dualisation | | Non-compact | |
|------------------|---------|----------------|------|-------------|------|-------------|------|-------------|-------|
| | | Time | Gap | Time | Gap | Time | Gap | Time | Gap |
| 10_ulysses_3.tsp | 152.0% | 4.63 | 0.0% | 1.76 | 0.0% | 0.29 | 0.0% | 3.16 | 0.0% |
| 10_ulysses_6.tsp | 663.2% | 0.30 | 0.0% | 0.40 | 0.0% | 0.06 | 0.0% | 0.12 | 0.0% |
| 10_ulysses_9.tsp | 4522.1% | 0.49 | 0.0% | 0.25 | 0.0% | 0.02 | 0.0% | 0.11 | 0.0% |
| 14_burma_3.tsp | 41.0% | 0.84 | 0.0% | 0.48 | 0.0% | 0.19 | 0.0% | 1.49 | 0.0% |
| 14_burma_6.tsp | 137.9% | 6.39 | 0.0% | 2.43 | 0.0% | 0.17 | 0.0% | 0.66 | 0.0% |
| 14_burma_9.tsp | 339.4% | 59.32 | 0.0% | — | — | 0.20 | 0.0% | 0.31 | 0.0% |
| 22_ulysses_3.tsp | 26.2% | 10.37 | 0.0% | — | — | 1.94 | 0.0% | 21.04 | 70.9% |
| 22_ulysses_6.tsp | 41.0% | 335.08 | 0.0% | — | — | 10.03 | 0.0% | 3.71 | 19.1% |
| 22_ulysses_9.tsp | 147.8% | 17.23 | 0.0% | — | — | 2.91 | 0.0% | 1.63 | 0.0% |
| 26_eil_3.tsp | 24.2% | 515.30 | 0.0% | — | — | 65.08 | 0.0% | 116.47 | 5.5% |
| 26_eil_6.tsp | 8.9% | — | — | — | — | — | — | 13.58 | 25.7% |
| 26_eil_9.tsp | 143.2% | — | — | — | — | — | — | 6.29 | 50.1% |
| 30_eil_3.tsp | 12.2% | — | — | — | — | — | — | 283.66 | 0.6% |
| 30_eil_6.tsp | — | — | — | — | — | — | — | 25.42 | 22.5% |
| 30_eil_9.tsp | — | — | — | — | — | — | — | 7.80 | 9.1% |
| 34_pr_3.tsp | 132.5% | — | — | — | — | — | — | 605.85 | 97.2% |
| 34_pr_6.tsp | — | — | — | — | — | — | — | 126.05 | 36.4% |
| 34_pr_9.tsp | — | — | — | — | — | — | — | 19.93 | 58.8% |

TAB. 1 – Tableau complet des résultats obtenus en 10 minutes de calcul

On notera que dans ce tableau, le gap pour la formulation non compacte (qui est une heuristique puisque le Branch & Price n'est pas implémenté) est calculé au mieux possible : si l'on connaît la vraie valeur optimale du problème robuste via un autre algorithme, on calcule le gap entre la solution obtenue par génération de colonnes et la vraie valeur optimale. Sinon, on compare la valeur de la solution entière de la génération de colonnes à celle de son relâché, ce qui est alors une sur-estimation du vrai gap à l'optimal (puisque il est nécessairement plus grand que le gap d'intégrité de la formulation non-compacte).

Annexe

Ci-dessous, une liste des principaux résultats obtenus, pour les deux méthodes les plus performantes (Traitement de la robustesse par dualisation pour les valeurs exactes, & Génération de colonnes sur la formulation non compacte pour les grandes instances.)

- **Instance :** 10_ulysses_3.tsp
 - Method : robust_dual
 - Cost : 136.99527629589417
 - Time : 0.06363415718078613
 - Gap : 0.0
 - Solution : [[1, 2, 3, 10], [4, 6, 7, 8], [5, 9]]
- **Instance :** 10_ulysses_6.tsp
 - Method : robust_dual
 - Cost : 55.11939124322688
 - Time : 0.06000089645385742
 - Gap : 0.0
 - Solution : [[6, 9], [5], [2, 3], [7], [10], [1, 4, 8]]
- **Instance :** 10_ulysses_9.tsp
 - Method : robust_dual
 - Cost : 33.29189782877749
 - Time : 0.01447296142578125
 - Gap : 0.0
 - Solution : [[6], [10], [9], [4], [7], [1], [2, 3], [8], [5]]
- **Instance :** 14_burma_3.tsp
 - Method : robust_dual
 - Cost : 93.38998725996821
 - Time : 0.18731284141540527
 - Gap : 0.0
 - Solution : [[1, 8, 9, 10, 11], [2, 12, 13, 14], [3, 4, 5, 6, 7]]
- **Instance :** 14_burma_6.tsp
 - Method : robust_dual
 - Cost : 42.74062354260174
 - Time : 0.17052197456359863
 - Gap : 0.0
 - Solution : [[1, 8, 11], [2, 13], [7], [3, 4, 5], [9, 10], [6, 12, 14]]
- **Instance :** 14_burma_9.tsp
 - Method : robust_dual
 - Cost : 20.762438566071065
 - Time : 0.19952797889709473
 - Gap : 0.0
 - Solution : [[1, 8], [2], [7], [10], [6, 12], [3, 4], [13, 14], [9, 11], [5]]
- **Instance :** 22_ulysses_3.tsp
 - Method : robust_dual

- Cost : 358.6368286225183
- Time : 1.8745441436767578
- Gap : 0.0
- Solution : [[1, 2, 3, 4, 8, 16, 17, 18], [7, 10, 12, 13, 14, 19, 20, 22], [5, 6, 9, 11, 15, 21]]]
- **Instance :** 22_ulysses_6.tsp
 - Method : robust_dual
 - Cost : 116.52876945505506
 - Time : 7.648591041564941
 - Gap : 0.0
 - Solution : [[1, 8, 16, 22], [10, 12, 20, 21], [5, 6, 14, 15], [7, 11], [2, 3, 4, 17, 18], [9, 13, 19]]]
- **Instance :** 22_ulysses_9.tsp
 - Method : robust_dual
 - Cost : 64.9735924526909
 - Time : 2.176405906677246
 - Gap : 0.0
 - Solution : [[1, 16, 22], [6, 10], [5, 9], [12, 14, 15], [20, 21], [2, 3, 17], [7, 13], [11, 19], [4, 8, 18]]]
- **Instance :** 26_eil_3.tsp
 - Method : robust_dual
 - Cost : 2297.6295855710846
 - Time : 56.28470301628113
 - Gap : 0.0
 - Solution : [[1, 2, 3, 5, 9, 10, 11, 16, 20, 21], [4, 12, 13, 15, 17, 18, 19], [6, 7, 8, 14, 22, 23, 24, 25, 26]]]
- **Instance :** 26_eil_6.tsp
 - Method : CG-NC
 - Cost : 1015.8601313012331
 - Time : 12.717041015625
 - Gap : 0.25650605471139276
 - Solution : [[4, 10, 12, 15, 17], [6, 14, 23, 25], [7, 8, 24, 26], [13, 18, 19], [1, 2, 5, 11, 16], [3, 9, 20, 21, 22]]]
- **Instance :** 26_eil_6.tsp
 - Method : robust_dual
 - Cost : Inf
 - Time : 600.0072569847107
 - Gap : Inf
 - Solution : [Int64[]]
- **Instance :** 26_eil_9.tsp
 - Method : CG-NC
 - Cost : 772.6560624033953
 - Time : 5.189327955245972

- Gap : 0.5005060696184076
- Solution : [[18, 13], [19, 24], [5, 17, 15], [4, 10, 12], [1, 3, 22, 26], [6, 14, 25], [7, 8, 23], [9, 20, 21], [2, 11, 16]]
- **Instance :** 26_eil_9.tsp
 - Method : robust_dual
 - Cost : Inf
 - Time : 600.0098669528961
 - Gap : Inf
 - Solution : [Int64[]]
- **Instance :** 30_eil_3.tsp
 - Method : CG-NC
 - Cost : 3021.110276255874
 - Time : 203.8809576034546
 - Gap : 0.006355427056133514
 - Solution : [[4, 7, 8, 10, 11, 12, 13, 14, 17, 19, 26], [1, 2, 5, 6, 15, 20, 21, 22, 27, 28, 29, 30], [3, 9, 16, 18, 23, 24, 25]]
- **Instance :** 30_eil_3.tsp
 - Method : robust_dual
 - Cost : 3021.110276255874
 - Time : 320.30139112472534
 - Gap : 0.0
 - Solution : [[1, 2, 5, 6, 15, 20, 21, 22, 27, 28, 29, 30], [4, 7, 8, 10, 11, 12, 13, 14, 17, 19, 26], [3, 9, 16, 18, 23, 24, 25]]
- **Instance :** 30_eil_6.tsp
 - Method : CG-NC
 - Cost : 1360.907559536604
 - Time : 23.08368706703186
 - Gap : 0.22472678866130041
 - Solution : [[2, 5, 21, 22, 28, 29], [9, 10, 11, 14], [3, 7, 23, 25], [4, 6, 12, 17, 26, 30], [8, 13, 15, 19, 20, 27], [1, 16, 18, 24]]
- **Instance :** 30_eil_6.tsp
 - Method : robust_dual
 - Cost : Inf
 - Time : 600.0070669651031
 - Gap : Inf
 - Solution : [Int64[]]
- **Instance :** 30_eil_9.tsp
 - Method : CG-NC
 - Cost : 713.971761421444
 - Time : 6.4169557094573975
 - Gap : 0.09071368486652343
 - Solution : [[22, 23, 24], [3, 17, 26], [7, 11, 14, 19], [1, 2, 21, 28], [4, 6, 27, 30], [16, 18], [5, 15, 20, 29], [9, 10, 25], [8, 12, 13]]

- **Instance :** 34_pr_3.tsp
 - Method : CG-NC
 - Cost : 1.1472591721102013e6
 - Time : 601.1148326396942
 - Gap : 0.9951453843063541
 - Solution : [[23, 32, 2, 9, 25, 19], [21, 31, 1, 17, 22, 26, 30, 34], [20, 14, 3, 11, 29, 12, 18, 27, 13, 24, 28, 4, 5, 8, 16, 33, 6, 7, 10, 15]]
- **Instance :** 34_pr_6.tsp
 - Method : CG-NC
 - Cost : 314058.52583083964
 - Time : 99.4175136089325
 - Gap : 0.36440770068735834
 - Solution : [[11, 12, 13, 14, 15, 16, 17], [21, 22, 23, 24], [25, 26, 27, 28, 29], [1, 2, 3, 4, 5, 8], [18, 19, 30, 32, 33, 34], [6, 7, 8, 9, 10, 20, 31]]
- **Instance :** 34_pr_9.tsp
 - Method : CG-NC
 - Cost : 198030.4277218256
 - Time : 16.553050756454468
 - Gap : 0.5884294308486627
 - Solution : [[1, 2, 5], [21, 22, 24], [23, 25], [3, 6, 7, 8, 20], [28, 29, 32, 33, 34], [11, 12, 17, 18], [9, 13, 14, 15, 16], [26, 27, 30], [4, 10, 19, 31]]

Visualisation des solutions obtenues

