

ECMA

Projet d'optimisation robuste

Louise Eymery
Hannes Van Overloop

Février 2023

Table des matières

1	Modélisation	2
2	Résolution exacte	7
2.1	Dualisation	7
2.2	Branch-and-Cut	7
2.3	Plans coupants	7
2.4	Bornes	7
3	Résolution heuristique	8
3.1	Solution initiale	8
3.2	Voisinages	8
3.3	Descente locale	8
3.4	Discussion	8
4	Résultats	9
5	Annexes	10

1 Modélisation

Question 1

On se place dans un graphe $G = (V, E)$.
Le problème statique s'écrit :

$$\begin{aligned}
 \min_x \quad & \sum_{e \in E} l_e \widehat{x}_e \\
 \text{s.t.} \quad & \sum_{v \in V} w_v y_{k,v} \leq B & \forall k \in \llbracket 1, K \rrbracket \quad (1) \\
 & \sum_{k=1}^K y_{k,v} = 1 & \forall v \in V \quad (2) \\
 & x_{k,e} \geq (y_{k,u} + y_{k,v}) - 1 & \forall e = (u, v) \in E, \forall k \in \llbracket 1, K \rrbracket \quad (3) \\
 & x_{k,e} \leq \frac{1}{2}(y_{k,u} + y_{k,v}) & \forall e = (u, v) \in E, \forall k \in \llbracket 1, K \rrbracket \quad (4) \\
 & \widehat{x}_e = \sum_{k=1}^K x_{k,e} & \forall e \in E \quad (5) \\
 & y_{k,v} \in \{0, 1\} & \forall v \in V, \forall k \in \llbracket 1, K \rrbracket \\
 & x_{k,e} \in \{0, 1\} & \forall e \in E, \forall k \in \llbracket 1, K \rrbracket \\
 & \widehat{x}_e \in \{0, 1\} & \forall e \in E
 \end{aligned}$$

$y_{k,v}$ vaut un 1 si v est dans le cluster k et 0 sinon.

$x_{k,e}$ vaut 1 si l'arête e est entre deux sommets du cluster k et 0 sinon.

La première contrainte traduit le respect de la limite de poids des arêtes dans chacun des clusters, la deuxième qu'un sommet est dans exactement un cluster, et les deux suivantes sont une linéarisation de la contrainte suivante : $\forall (uv) = e \in E, x_{k,e} = 1 \Leftrightarrow y_{k,u} = y_{k,v} = 1$.

Enfin, avec la cinquième contrainte, \widehat{x}_e vaut 1 si l'arête e est au sein d'une partie, 0 sinon.

Question 2

Passons à la modélisation robuste. Pour l'objectif,

On note $x = (\widehat{x}_e)_{e \in E} = (\sum_{k=1}^K x_{k,e})_{e \in E}$. Alors $l^T x$ devient :

$$\begin{aligned}
 & \max_{l^1 \in \mathcal{U}^1} (l^1)^T x \\
 & \Leftrightarrow \max_{\delta^1 \in [0, 3]^{|E|}} \sum_{(i,j) \in E} (l_{ij} + \delta_{ij}^1 (\widehat{l}_i + \widehat{l}_j)) \widehat{x}_{ij} \\
 & \text{s.t.} \quad \sum_{(i,j) \in E} \delta_{ij}^1 \leq L
 \end{aligned}$$

Pour les contraintes, on a pour tout $k \in \llbracket 1, K \rrbracket$:

$$\begin{aligned}
 \sum_{v \in V} w_v y_{k,v} \leq B, \forall w \in \mathcal{U}^2 & \Leftrightarrow \max_{w^2 \in \mathcal{U}^2} \sum_{v \in V} w_v^2 y_{k,v} \leq B \\
 & \Leftrightarrow \max_{\delta^2 \in [0, W_v]^{|V|}} \sum_{v \in V} w_v (1 + \delta_v^2) y_{k,v} \leq B \\
 \text{s.t.} \quad & \sum_{v \in V} \delta_v^2 \leq W
 \end{aligned}$$

On réécrit alors le problème :

$$\begin{aligned}
\min_x \max_{\delta^1} \quad & \sum_{(i,j) \in E} (l_{ij} + \delta_{ij}^1 (\widehat{l}_i + \widehat{l}_j)) \widehat{x}_{ij} \\
\text{s.t.} \quad & \sum_{(i,j) \in E} \delta_{ij}^1 \leq L \\
& \delta_{ij}^1 \in [0, 3] \quad \forall (i, j) \in E \\
& \begin{cases} \max_{\delta^2} \sum_{v \in V} w_v (1 + \delta_v^2) y_{k,v} \leq B \\ \sum_{v \in V} \delta_v^2 \leq W \\ \delta_v^2 \in [0, W_v], \forall v \in V \end{cases} \quad \forall k \in \llbracket 1, K \rrbracket \\
& \sum_{k=1}^K y_{k,v} = 1 \quad \forall v \in V \\
& x_{k,e} \geq (y_{k,u} + y_{k,v}) - 1 \quad \forall e = (u, v) \in E, \forall k \in \llbracket 1, K \rrbracket \\
& x_{k,e} \leq \frac{1}{2} (y_{k,u} + y_{k,v}) \quad \forall e = (u, v) \in E, \forall k \in \llbracket 1, K \rrbracket \\
& \sum_{k=1}^K x_{k,e} = \widehat{x}_e \quad \forall e \in E \\
& y_{k,v} \in \{0, 1\} \quad \forall v \in V, \forall k \in \llbracket 1, K \rrbracket \\
& x_{k,e} \in \{0, 1\} \quad \forall e \in E, \forall k \in \llbracket 1, K \rrbracket \\
& \widehat{x}_e \in \{0, 1\} \quad \forall e \in E
\end{aligned}$$

Question 3

a) Le problème devient :

$$\begin{aligned}
\min_{x,t} \quad & t \\
\text{s.t.} \quad & \sum_{(i,j) \in E} (l_{ij} + \delta_{ij}^1 (\widehat{l}_i + \widehat{l}_j)) \widehat{x}_{ij} \leq t \\
& \sum_{(i,j) \in E} \delta_{ij}^1 \leq L \\
& \delta_{ij}^1 \in [0, 3] \quad \forall (i, j) \in E \\
& \begin{cases} \max_{\delta^2} \sum_{v \in V} w_v (1 + \delta_v^2) y_{k,v} \leq B \\ \sum_{v \in V} \delta_v^2 \leq W \\ \delta_v^2 \in [0, W_v], \forall v \in V \end{cases} \quad \forall k \in \llbracket 1, K \rrbracket \\
& \sum_{k=1}^K y_{k,v} = 1 \quad \forall v \in V \\
& x_{k,e} \geq (y_{k,u} + y_{k,v}) - 1 \quad \forall e = (u, v) \in E, \forall k \in \llbracket 1, K \rrbracket \\
& x_{k,e} \leq \frac{1}{2} (y_{k,u} + y_{k,v}) \quad \forall e = (u, v) \in E, \forall k \in \llbracket 1, K \rrbracket \\
& \sum_{k=1}^K x_{k,e} = \widehat{x}_e \quad \forall e \in E \\
& y_{k,v} \in \{0, 1\} \quad \forall v \in V, \forall k \in \llbracket 1, K \rrbracket \\
& x_{k,e} \in \{0, 1\} \quad \forall e \in E, \forall k \in \llbracket 1, K \rrbracket \\
& \widehat{x}_e \in \{0, 1\} \quad \forall e \in E
\end{aligned}$$

b) On pose :

$$\mathcal{U}^{1*} = \begin{cases} \{l_{ij} + \frac{L}{|E|}(\widehat{l}_i + \widehat{l}_j), \forall (i, j) \in E\} & \text{si } \frac{L}{|E|} \leq 3 \\ \{l_{ij}, \forall (i, j) \in E\} & \text{sinon} \end{cases}$$

et

$$\mathcal{U}^{2*} = \begin{cases} \{w_v(1 + \frac{W}{|E|}), \forall v \in V\} & \text{si } \frac{W}{|V|} \leq W_v \\ \{w_v, \forall v \in V\} & \text{sinon} \end{cases}$$

c) On a les sous-problèmes suivants.

Pour l'objectif :

$$\begin{aligned} \max_{\delta^1} \quad & \sum_{(i,j) \in E} (l_{ij} + \delta_{ij}^1(\widehat{l}_i + \widehat{l}_j))\widehat{x}_{ij}^* \\ \text{s.t.} \quad & \sum_{(i,j) \in E} \delta_{ij}^1 \leq L \\ & \delta_{ij}^1 \in [0, 3] \quad \forall (i, j) \in E \end{aligned}$$

Pour la contrainte, pour tout $k \in \llbracket 1, K \rrbracket$,

$$\begin{aligned} \max_{\delta^2} \quad & \sum_{v \in V} w_v(1 + \delta_v^2) y_{k,v}^* \\ \text{s.t.} \quad & \sum_{v \in V} \delta_v^2 \leq W \\ & \delta_v^2 \in [0, W_v] \quad \forall v \in V \end{aligned}$$

d) On souhaite que :

$$l^* x^* \leq t^*$$

Et que pour tout $k \in \llbracket 1, K \rrbracket$:

$$w^{2*} y_k^* \leq B$$

e) Ce qui nous donne les coupes suivantes à ajouter.

Pour le premier sous-problème :

$$t \geq \sum_{(i,j) \in E} (l_{ij} + \delta_{ij}^{1*}(\widehat{l}_i + \widehat{l}_j))\widehat{x}_{ij}$$

Et pour ceux liés à la contrainte :

Pour tout $k \in \llbracket 1, K \rrbracket$:

$$\sum_{v \in V} w_v(1 + \delta_v^{2*}) y_{k,v} \leq B$$

Question 4

a) La fonction objectif se réécrit comme suit :

$$\max_{\delta^1} \sum_{(i,j) \in E} (\widehat{x}_{ij} l_{ij} + \delta_{ij}^1(\widehat{l}_i + \widehat{l}_j)\widehat{x}_{ij}) \Leftrightarrow \sum_{(i,j) \in E} \widehat{x}_{ij} l_{ij} + \max_{\delta^1} \sum_{(i,j) \in E} \delta_{ij}^1(\widehat{l}_i + \widehat{l}_j)\widehat{x}_{ij}$$

b) Le problème interne lié aux variables δ_{ij}^1 est donc :

$$\begin{aligned}
& \max_{\delta^1} \sum_{(i,j) \in E} \delta_{ij}^1 (\widehat{l}_i + \widehat{l}_j) \widehat{x}_{ij} \\
& \text{s.t.} \quad \sum_{ij \in E} \delta_{ij}^1 \leq L \\
& \quad \delta_{ij}^1 \leq 3 \quad \forall (i,j) \in E \\
& \quad \delta_{ij}^1 \geq 0 \quad \forall (i,j) \in E
\end{aligned}$$

c) Que l'on dualise comme suit :

$$\begin{aligned}
& \min_{\alpha, \beta} L\alpha + 3 \sum_{(i,j) \in E} \beta_{ij} \\
& \text{s.t.} \quad \alpha + \beta_{ij} \geq (\widehat{l}_i + \widehat{l}_j) \widehat{x}_{ij} \quad \forall (i,j) \in E \\
& \quad \alpha \geq 0 \\
& \quad \beta_{ij} \geq 0 \quad \forall (i,j) \in E
\end{aligned}$$

d) Les contraintes robustes, de la forme :

Pour tout $k \in \llbracket 1, K \rrbracket$,

$$\begin{aligned}
& \max_{\delta^2 \in [0, W_v]^{|V|}} \sum_{v \in V} w_v (1 + \delta_v^2) y_{k,v} \leq B \\
& \text{s.t.} \quad \sum_{v \in V} \delta_v^2 \leq W
\end{aligned}$$

peuvent aussi s'écrire :

Pour tout $k \in \llbracket 1, K \rrbracket$,

$$\begin{aligned}
& \left(\sum_{v \in V} w_v y_{k,v} + \max_{\delta^2 \in [0, W_v]^{|V|}} \sum_{v \in V} w_v \delta_v^2 y_{k,v} \right) \leq B \\
& \text{s.t.} \quad \sum_{v \in V} \delta_v^2 \leq W
\end{aligned}$$

e) Les problèmes internes liés aux variables δ_v^2 sont donc :

Pour tout $k \in \llbracket 1, K \rrbracket$,

$$\begin{aligned}
& \max_{\delta^2} \sum_{v \in V} \delta_v^2 w_v y_{k,v} \\
& \text{s.t.} \quad \sum_{v \in V} \delta_v^2 \leq W \\
& \quad \delta_v^2 \leq W_v \quad \forall v \in V \\
& \quad \delta_v^2 \geq 0 \quad \forall v \in V
\end{aligned}$$

f) Soit $k \in \llbracket 1, K \rrbracket$. Le dual du problème robuste (R_k) est :

$$\begin{aligned}
& \min_{\zeta, \gamma} W\zeta + \sum_{v \in V} W_v \gamma_v \\
& \text{s.t.} \quad \zeta + \gamma_v \geq w_v y_{k,v} \quad \forall v \in V \\
& \quad \zeta \geq 0 \\
& \quad \gamma_v \geq 0 \quad \forall v \in V
\end{aligned}$$

g) En conclusion, on obtient le problème ci-dessous :

$$\begin{aligned}
\min_{x, \alpha, \beta} \quad & \sum_{(i,j) \in E} l_{ij} \widehat{x}_{ij} + L\alpha + 3 \sum_{(i,j) \in E} \beta_{ij} \\
\text{s.t.} \quad & \alpha + \beta_{ij} \geq x_{ij}(\widehat{l}_i + \widehat{l}_j) & \forall (i,j) \in E \\
& \alpha \geq 0 \\
& \beta_{ij} \geq 0 & \forall (i,j) \in E \\
& \sum_{v \in V} w_v y_{k,v} + W \zeta_k + \sum_{v \in V} W_v \gamma_{kv} \leq B & \forall k \in \llbracket 1, K \rrbracket \\
& \zeta_k + \gamma_{kv} \geq w_v y_{k,v} & \forall v \in V, \forall k \in \llbracket 1, K \rrbracket \\
& \zeta_k \geq 0 & \forall k \in \llbracket 1, K \rrbracket \\
& \gamma_{kv} \geq 0 & \forall v \in V, \forall k \in \llbracket 1, K \rrbracket \\
& \sum_{k=1}^K y_{k,v} = 1 & \forall v \in V \\
& x_{k,e} \geq (y_{k,u} + y_{k,v}) - 1 & \forall e = (u,v) \in E, \forall k \in \llbracket 1, K \rrbracket \\
& x_{k,e} \leq \frac{1}{2}(y_{k,u} + y_{k,v}) & \forall e = (u,v) \in E, \forall k \in \llbracket 1, K \rrbracket \\
& \widehat{x}_e = \sum_{k=1}^K x_{k,e} & \forall e \in E \\
& y_{k,v} \in \{0, 1\} & \forall v \in V, \forall k \in \llbracket 1, K \rrbracket \\
& x_{k,e} \in \{0, 1\} & \forall e \in E, \forall k \in \llbracket 1, K \rrbracket \\
& \widehat{x}_e \in \{0, 1\} & \forall e \in E
\end{aligned}$$

2 Résolution exacte

Trois résolutions exactes sont implémentées en utilisant le solveur CPLEX :

- Par dualisation
- Par plans coupants
- Par branch-and-cut

2.1 Dualisation

La résolution par dualisation permet de résoudre les instances de 10 et 14 noeuds en moins d'une seconde. Au-delà, le temps de résolution croît très rapidement pour atteindre presque 2h sur des instances de 30 noeuds et au plus 6 ou 9 clusters. La résolution exacte semble donc compliquée pour des grandes instances.

2.2 Branch-and-Cut

La résolution par Branch-and-Cut est moins performante que celle par dualisation. En effet, les instances de 10 et 14 noeuds nécessitent jusqu'à 10 minutes pour être résolues. De même, au-delà, le temps de résolution croît très rapidement et atteint presque 2h sur des instances dès 22 noeuds.

2.3 Plans coupants

La résolution par plans coupants est encore moins performante. En effet, elle est clairement une forme moins efficace du Branch-and-Cut et n'est, en plus, pas implémenté directement dans CPLEX.

2.4 Bornes

La résolution de la relaxation continue du modèle par dualisation nous permet d'obtenir des bornes inférieures pour les différentes instances. Ces dernières nous permettent d'avoir une idée de la performance de l'heuristique détaillée ci-après (3).

3 Résolution heuristique

Étant donné les performances décevantes des résolutions exactes sur des instances plus grandes, nous implémentons une heuristique avec pour objectif d'être en capacité de générer rapidement une solution admissible, que nous espérons être de bonne qualité.

Nous optons pour une descente locale à voisinages multiples. Pour cela nous générons tout d'abord une solution initiale, puis nous cherchons des solutions voisines améliorantes dans différentes structures de voisinage.

3.1 Solution initiale

Vu la complexité du problème, la mise en place d'une solution initiale par une heuristique gourmande s'avère compliquée voire impossible (plusieurs échecs ont été rencontrés). Nous avons donc fait le choix d'utiliser CPLEX pour générer une solution admissible. Pour cela nous résolvons la formulation par dualisation du problème, avec pour différence que l'objectif est remplacé par une constante. Ainsi, le solveur renvoie une solution admissible dès qu'il la trouve. Cette dernière est notre solution initiale.

3.2 Voisinages

Pour ensuite améliorer la solution initiale, nous mettons en place trois structures de voisinage :

- L'**échange simple** consiste à changer un noeud de cluster.
- L'**échange de couples** consiste à échanger de cluster des sommets A, B vérifiant $w_v(A) < w_v^2(B, k_2)$ et $w_v(B) < w_v^2(A, k_1)$.
- Le voisinage w_v **similaire** consiste à échanger un ensemble de sommets $\{A_1, \dots, A_p\}$ de même w_v à ε près, comme suit : A_1 va dans le cluster de A_2 , A_2 dans celui de A_3 , ..., A_p dans celui de A_1 .

Pour chacun de ces voisinages on ne tient évidemment compte que des voisins qui sont admissibles. On vérifie cela en résolvant les sous-problèmes calculant le poids des clusters concernés et en s'assurant que ces derniers restent inférieurs ou égaux à B .

3.3 Descente locale

Finalement, dans la recherche locale, nous considérons successivement chaque voisinage. La procédure est détaillée ci-après :

- Nous regardons pour tous les noeuds (parcourus dans un ordre aléatoire) son meilleur voisin dans le voisinage *échange simple*. S'il est améliorant nous y allons.
- Nous regardons dans un ordre aléatoire, pour l'ensemble des couples vérifiant la condition du voisinage *échange de couples*, son voisin dans ce voisinage. S'il est améliorant nous y allons.
- Nous regardons dans un ordre aléatoire, pour tous les ensembles de sommets vérifiant la condition du voisinage w_v *similaire*, son voisin dans ce voisinage. S'il est améliorant nous y allons.

Cette procédure est répétée tant que le temps limite, qui est notre condition d'arrêt, n'est pas atteint. Précisons que la recherche de la solution initiale n'est pas prise en compte dans le temps de calcul.

Précisons également que, sur les instances de 100 noeuds ou plus, nous ne regardons pas l'ensemble des noeuds, l'ensemble des couples ou l'intégralité des ensembles de sommets mais seulement un pourcentage de la totalité. Nous faisons ce choix afin d'accroître la diversification de la solution et donc d'étendre le champs de la recherche locale.

3.4 Discussion

Sur les instances de plus de cent noeuds la recherche d'une solution initiale peut prendre un temps important (atteignant parfois 10min). La limite de temps n'est donc pas tout à fait représentative du temps réellement passé pour atteindre la solution obtenue. Il limite cependant la durée passé à effectuer la descente locale.

4 Résultats

Voici tout d'abord en figure 1 les performances de résolution des trois méthodes exactes. L'abscisse représente le temps de résolution, en secondes. L'ordonnée donne le nombre d'instances résolues. Nous retrouvons bien ce qui est écrit dans la partie résolution exacte (2).

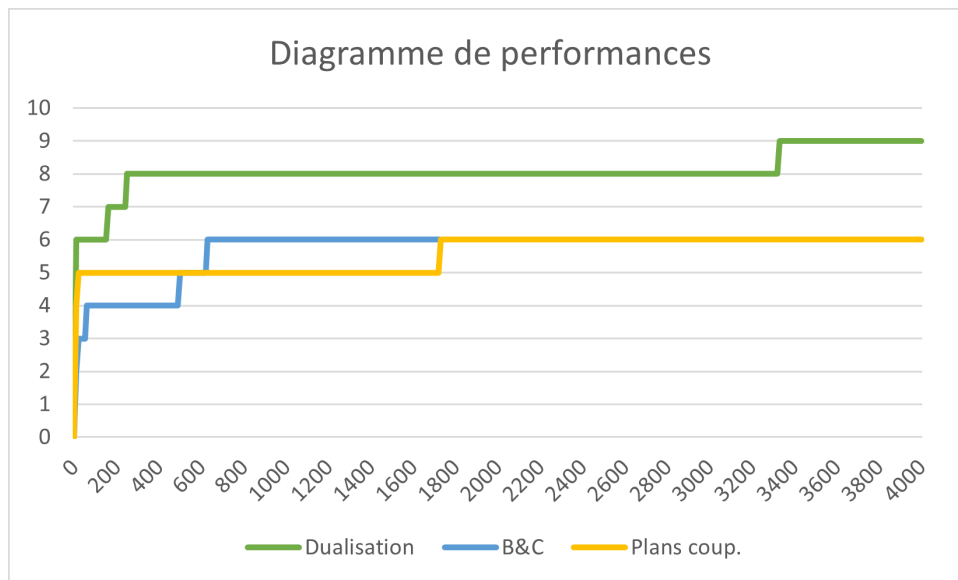


FIGURE 1 – Diagramme de performance de méthodes de résolution exacte

Nous comparons ensuite les résultats et performances de chacune des quatre méthodes de résolution, sur les instances pas uniquement résolues par l'heuristique. Ces données sont regroupées dans le tableau 1.

Les résolutions par méthode exacte sont arrêtées au bout de $2h$ ($7200s$). Pour les résolutions par heuristique nous imposons une limite de temps de $60s$. Un *Gap* de 0% correspond à une solution exacte, à contrario, un *Gap* de 99% signifie que la meilleure solution trouvée a une valeur très éloignée de la borne inférieure calculée par le solveur.

Nous avons constaté que les *gap* fournis par CPLEX lors du Branch-and-Cut étaient très importants, et ce même quand les solutions étaient proches de l'optimum trouvé par la méthode de dualisation. Les bornes trouvées par CPLEX avec cette méthode ne sont donc pas très bonnes, *ie.* éloignées de la solution optimale. C'est pourquoi nous avons rajouté une colonne "Best gap (if different)", qui correspond à l'écart entre la meilleure solution trouvée avec la méthode Branch-and-Cut, et la meilleure borne que nous avons calculée (qui est ici obtenue par la modélisation par dualisation).

Finalement, le tableau 2 regroupe les résultats obtenus à l'aide de l'heuristique sur les instances plus grandes.

On trouvera également en annexe un tableau contenant la meilleure solution obtenue pour chaque instance. Lorsqu'une solution exacte a été obtenue par résolution exacte, c'est cette solution, sinon c'est la solution obtenue par heuristique.

5 Annexes

Instance	PR	Plans coupants		Branch-and-cut			Dualisation		Heuristique	
		Temps (s)	Gap	Temps (s)	Gap (Cplex)	Best gap (if diff.)	Temps (s)	Gap	Temps	Gap
10_ulysses_3	152%	18.7	0%	1.2	0%		0.38	0%	120	2%
10_ulysses_6	663%	1.7	0%	47	0%		0.19	0%	120	4%
10_ulysses_9	4524%	4.5	0%	7.5	0%		0.16	0%	120	3%
14_burma_3	41%	7.6	0%	14	0%		0.84	0%	120	19%
14_burma_6	138%	9.4	0%	624	0%		0.86	0%	120	5%
14_burma_9				7234	0%		7230	0%	120	19%
22_ulysses_3	26%	1715	0%	489	0%		6.4	0%	120	22%
22_ulysses_6	41%			900	32%		157	0%	120	29%
22_ulysses_9				7213	99%	55%	7242	53%	120	213%
26_eil_3				7200	19%	5%	250	0%	120	26%
26_eil_6				7200	90%	68%	7789	65%	120	58%
26_eil_9				7203	99%		7206	99.2%	120	111%
30_eil_3				7202	38%	5%	3295	0%	120	15%
30_eil_6				7203	98%	79%	7348	78%	120	115%
30_eil_9				6524	99%		7259	99.7%	120	102%
34_pr_3							5111	0%	120	34%
34_pr_6							7300	74%	120	66%

TABLE 1 – Résultats et performances des différentes méthodes de résolution

Instance	Gap	Valeur	TimeLimit (s)	Solution initiale (s)
38_rat_3	41%	10920	120	1.1
38_rat_6	82%	4977	120	1.0
38_rat_9	109%	3405	120	2.1
40_eil_3	41%	7940	120	1.3
40_eil_6	82%	3690	120	3.1
40_eil_9	103%	2143	120	30
44_lin_3	63%	200481	120	1.7
44_lin_6	134%	96946	120	9.0
44_lin_9	140%	54969	120	1.8
48_att_3	64%	900134	120	0.8
48_att_6	205%	439320	120	46
48_att_9	131%	203934	120	2.9
52_berlin_3	76%	200370	120	4.1
52_berlin_6	109%	76841	120	7.7
52_berlin_9	289%	58201	120	112
70_st_3	38%	30003	120	6.1
70_st_6	96%	14449	120	31
70_st_9	113%	8354	120	68
80_gr_3	52%	25454	120	27
80_gr_6	74%	9644	120	32
80_gr_9	167%	6977	120	249
100_kroA_3	61%	2.08×10^6	120	0.9
100_kroA_6	102%	802227	120	1.9
100_kroA_9	163%	530466	120	212
202_gr_3	118%	90873	120	2.7

TABLE 2 – Résultats avec l'heuristique

Instance	Meilleure solution obtenue
10_ulysses_3	[1 1 1 2 3 1 1 1 3 2]
10_ulysses_6	[1 2 2 1 6 3 4 1 3 5]
10_ulysses_9	[7 2 2 8 6 9 4 1 5 3]
14_burma_3	[1 2 3 3 3 3 3 1 1 1 2 2 2]
14_burma_6	[5 3 1 1 1 2 6 5 4 5 2 3 2]
14_burma_9	[6 3 1 1 4 5 2 6 8 7 8 5 9 9]
22_ulysses_3	[2 2 2 2 1 1 3 2 1 3 1 3 3 1 2 2 2 3 3 1 3]
22_ulysses_6	[2 4 4 4 6 6 1 2 5 3 1 3 5 6 6 2 4 4 5 3 3 2]
22_ulysses_9	[1 6 6 3 5 8 9 3 5 8 7 4 9 4 4 1 6 3 7 2 2 1]
26_eil_3	[2 2 2 3 2 1 1 1 2 2 2 3 3 1 3 2 3 3 3 2 2 1 1 1 1]
26_eil_6	[6 3 6 5 1 2 6 6 3 1 1 4 2 1 1 5 4 5 3 3 6 2 4 2 6]
26_eil_9	[8 8 5 9 1 2 4 4 1 9 8 7 6 2 1 5 7 6 7 5 5 3 3 4 2 3]
30_eil_3	[1 1 2 3 2 2 3 3 1 3 3 3 3 2 1 3 1 3 2 2 2 1 1 1 3 2 2 2 2]
30_eil_6	[6 6 4 5 2 5 2 1 4 5 1 2 2 3 6 4 1 2 3 3 6 3 1 1 5 2 6 3 3]
30_eil_9	[7 7 3 8 4 8 9 2 5 5 9 2 2 9 4 6 3 6 9 4 7 1 1 1 5 3 8 7 4 8]
34_pr_3	[3 3 3 3 1 3 3 3 2 2 2 3 3 2 2 2 1 3 1 1 3 3 1 1 2 1 2 2 1 1 1]
34_pr_6	[4 4 4 3 1 4 4 4 2 2 2 2 2 2 2 1 1 1 3 6 6 3 6 3 6 5 6 5 5 1 5 5 5]
34_pr_9	[2 7 1 1 1 1 7 2 8 8 6 8 3 8 3 3 3 3 4 6 7 6 2 4 4 9 5 5 5 5 1 9 9 9]
38_rat_3	[3 3 3 3 3 3 1 1 1 3 1 3 3 3 1 2 1 1 3 3 2 2 1 2 3 2 1 3 1 1 3 2 2 2 2 1 2 2]
38_rat_6	[5 5 5 5 1 1 4 1 5 3 5 5 3 1 1 4 4 3 3 1 2 4 4 2 4 4 6 3 6 2 2 2 2 2 2 6 6]
38_rat_9	[3 2 3 2 2 4 4 9 3 3 3 8 4 5 6 6 6 8 7 2 2 4 5 5 6 9 7 7 8 8 5 5 9 6 9 1 1]
40_eil_3	[3 1 1 2 3 1 3 3 1 3 1 2 2 1 1 1 3 3 2 1 1 2 2 2 3 2 2 3 3 2 2 3 3 1 1 2 1]
40_eil_6	[1 2 2 6 4 3 5 5 6 6 5 2 4 4 3 4 3 5 5 1 3 2 2 3 2 6 2 3 5 1 1 3 1 1 5 6 4 2 6]
40_eil_9	[3 2 1 5 4 6 9 4 1 9 9 8 6 6 2 2 4 4 9 1 6 2 5 3 5 8 7 8 8 3 7 1 3 3 1 7 5 2 6 8]
44_lin_3	[2 2 2 2 2 2 2 2 1 1 3 2 2 1 3 3 2 3 1 1 1 3 2 2 3 1 1 1 1 3 1 1 2 1 3 1 3 3 3 3 3]
44_lin_6	[4 4 3 3 3 3 4 2 3 2 4 2 2 4 5 2 5 5 6 5 4 6 1 5 5 6 3 4 6 6 4 6 6 2 6 5 1 1 1 5 1 5 1]
44_lin_9	[1 1 6 2 2 1 1 2 2 1 1 6 2 6 9 7 6 4 7 9 7 9 3 5 6 7 3 3 4 4 9 3 3 5 7 8 5 8 4 8 8 8 4]
48_att_3	[2 2 3 2 2 1 1 2 3 2 1 3 3 3 1 1 1 1 3 2 3 2 2 3 3 1 3 1 1 2 1 2 2 1 1 3 1 1 2]
48_att_6	[5 3 5 3 3 2 4 4 5 6 6 2 2 2 6 5 6 4 2 1 1 5 1 3 3 1 1 4 3 1 4 6 1 5 3 4 2 5 5 5 3 1 4 1 2 6 6]
48_att_9	[5 8 9 8 1 3 4 2 3 8 7 7 6 5 7 2 3 4 3 7 7 9 2 6 1 8 4 1 5 5 1 7 9 8 5 3 2 1 2 9 9 3 4 6 7 6 9]
52_berlin_3	[2 3 3 1 1 2 3 3 1 3 1 1 1 1 2 2 3 2 3 2 3 2 3 1 1 1 3 3 3 3 2 1 2 3 2 2 2 2 3 3 1 2 2 1 2 2 2 3 1 1]
52_berlin_6	[5 1 3 5 2 6 1 6 6 6 4 2 4 2 2 3 3 3 1 1 5 1 5 2 4 4 1 1 1 5 6 5 1 3 5 5 6 3 1 6 5 3 2 2 3 3 2 4]
52_berlin_9	[1 7 7 1 1 5 7 6 6 9 3 5 1 3 4 8 8 2 6 8 7 7 8 2 1 5 3 1 7 8 8 4 9 2 2 2 4 9 6 6 7 4 8 2 5 1 4 9 8 3 3]
70_st_3	[3 3 3 2 2 3 2 3 1 2 1 1 3 3 2 2 3 1 3 2 1 3 2 3 2 3 2 1 1 3 3 1 1 1 2 2 2 2 1 1 3 1 3 1 2 1 2 3 3 1 1 1 3 1 3 3 1 2 3 3 1]
70_st_6	[6 5 2 4 6 2 2 4 5 3 3 6 1 4 4 1 5 1 4 2 6 4 1 4 4 2 3 2 1 4 1 2 6 3 1 1 6 1 3 5 5 2 2 2 5 3 6 3 2 6 6 3 4 6 2 6 4 5 5 3 1 3 5 1 3 5 4 1]
70_st_9	[9 1 1 4 9 7 1 1 6 8 8 2 3 7 5 9 4 4 5 7 2 5 3 7 2 1 6 7 3 4 3 1 2 4 3 9 4 3 6 6 7 5 8 4 6 6 3 8 1 9 2 9 2 8 1 8 1 7 5 2 6 2 8 9 8 5 8 4 5 3]
80_gr_3	[3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 2 2 2 3 3 3 3 1 1 3 1 1 1 2 1 2 2 2 1 1 2 1 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1]
80_gr_6	[5 3 3 3 3 3 3 3 3 1 3 3 3 5 6 6 5 6 6 2 5 6 6 4 5 5 5 3 1 1 5 5 1 1 2 1 5 1 1 1 2 6 1 2 1 2 2 2 2 2 2 4 2 4 4 6 4 6 4 4 4 4 4 1 2]
80_gr_9	[2 9 9 9 4 9 9 9 9 4 3 4 9 4 4 6 4 4 4 6 1 1 4 6 5 1 1 1 9 2 2 2 3 2 3 2 8 8 3 6 3 3 6 3 5 5 6 3 8 5 5 6 5 5 6 5 5 6 1 7 1 1 1 7 7 8 7 8 8 8 8 7 7 8 2]