

Projet ECMA 2022-2023

Rym Ben Maktouf

Ahmed Belmabrouk

17 février 2023

Table des matières

1	Contexte	2
2	Modélisation Papier	2
3	Résolution numérique	5
3.1	Résolution statique compacte	5
3.2	Résolution par dualisation	5
3.3	Résolution par les plans coupants	5
3.4	Résolution par la méthode de Branch and Cut	5
3.5	Résolution par une heuristique	6
3.6	Analyse de performance	6
4	Annexe	7

1 Contexte

Le but de ce projet est l'étude d'un problème de partitionnement robuste. L'objectif du problème statique consiste à partitionner les sommets du graphe en au plus K parties de poids inférieur ou égal à B, de façon à minimiser les distances des arêtes à l'intérieur des parties. Alors que pour le problème robuste les distances sont dotées d'incertitude ; On considère la possibilité que les valeurs choisies pour les distances soient sous-évaluées. On propose de résoudre ce par problème :

1. par un algorithme de plans coupants ;
2. par un algorithme de branch-and-cut (via un LazyCallback) ;
3. par dualisation ;
4. par une heuristique ou la résolution d'une formulation non compacte.

2 Modélisation Papier

1. Modélisation statique compacte :

$$(P_S) : \left\{ \begin{array}{ll} \min_{x,z} \sum_{ij \in E} l_{ij} x_{ij} & (1) \\ \text{s.c. } x_{ij} + z_i^k - z_j^k \leq 1 & \forall ij \in E, \forall k \in \{1, \dots, K_{max}\} \quad (2) \\ x_{ij} - z_i^k + z_j^k \leq 1 & \forall ij \in E, \forall k \in \{1, \dots, K_{max}\} \quad (3) \\ -x_{ij} + z_i^k + z_j^k \leq 1 & \forall ij \in E, \forall k \in \{1, \dots, K_{max}\} \quad (4) \\ \sum_{k=1}^{K_{max}} z_i^k = 1 & \forall i \in V \quad (5) \\ z_i^k = 0 & \forall i \in V, \forall k \in \{1, \dots, K_{max}\} \text{ tels que } k > i \quad (6) \\ \sum_{i \in V} z_i^k \geq 1 & \forall k \in \{1, \dots, K_{max}\} \quad (7) \\ \sum_{i \in V} w_i z_i^k \leq B & \forall k \in \{1, \dots, K_{max}\} \quad (8) \\ x_{ij} \in \{0, 1\} & \forall ij \in E \quad (9) \\ z_i^k \in \{0, 1\} & \forall i \in V, \forall k \in \{1, \dots, K_{max}\} \quad (10) \end{array} \right.$$

Les variables :

- x_{ij} est une variable binaire qui vaut 1 si l'arc (i, j) est choisi dans une des clusters(parties) et 0 sinon.
- z_i^k est une binaire qui vaut 1 si le noeud i est affecté au cluster k et 0 sinon pour tout noeud $i \in V$ et $k \in \{1 \dots K_{max}\}$.

Les contraintes :

- Les contraintes (2),(3) et (4) : contraintes liantes entre les variables x et z.
- Les contraintes (5) : chaque noeud i est dans exactement un cluster.
- Les contraintes (6) : chaque noeud ne peut pas être dans un cluster d'indice plus grand que i.
- Les contraintes (7) : chaque cluster est non vide.
- Les contraintes (8) : Contraintes de poids sur les sommets de chaque cluster .
- Les contraintes (9) et (10) : Contraintes de variables binaires

2. Modèle du problème robuste :

$$\min_{z,x} \max_{l_{ij}^1 \in U^1} \sum_{ij \in E} l_{ij}^1 x_{ij}$$

sous les contraintes :

$$x_{ij} + z_i^k - z_j^k \leq 1 \quad \forall ij \in E \quad \forall k \in \{1, \dots, K_{max}\} \quad (2a)$$

$$x_{ij} - z_i^k + z_j^k \leq 1 \quad \forall ij \in E \quad \forall k \in \{1, \dots, K_{max}\} \quad (2b)$$

$$-x_{ij} + z_i^k + z_j^k \leq 1 \quad \forall ij \in E \quad \forall k \in \{1, \dots, K_{max}\} \quad (2c)$$

$$\sum_{k=1}^{K_{max}} z_i^k = 1 \quad \forall i \in V \quad (2d)$$

$$z_i^k = 0 \quad \forall k > i, i \in V \quad \forall k \in \{1, \dots, K_{max}\} \quad (2e)$$

$$\sum_{i \in V} z_i^k \geq 1 \quad \forall k \in \{1, \dots, K_{max}\} \quad (2f)$$

$$\sum_{i \in V} w_i^k z_i^k \leq B \quad \forall k \in \{1, \dots, K_{max}\}, w_i^k \in U^k \quad (2g)$$

$$z_i^k \in \{0, 1\} \quad \forall i \in V \quad \forall k \in \{1, \dots, K_{max}\} \quad (2h)$$

$$x_{ij} \in \{0, 1\} \quad \forall ij \in E \quad (2i)$$

Toutes les contraintes sont statiques car elles ne font pas intervenir la donnée incertaine : le temps minimal de déplacement entre deux villes.

3. Résolution par plans coupants et LazyCallback :

(a) Le modèle du problème robuste est équivalent au modèle suivant :

$$\min_z z$$

sous les contraintes :

$$x_{ij} + z_i^k - z_j^k \leq 1 \quad \forall ij \in E \quad \forall k \in \{1, \dots, K_{max}\} \quad (3a)$$

$$x_{ij} - z_i^k + z_j^k \leq 1 \quad \forall ij \in E \quad \forall k \in \{1, \dots, K_{max}\} \quad (3b)$$

$$-x_{ij} + z_i^k + z_j^k \leq 1 \quad \forall ij \in E \quad \forall k \in \{1, \dots, K_{max}\} \quad (3c)$$

$$\sum_{k=1}^{K_{max}} z_i^k = 1 \quad \forall i \in V \quad (3d)$$

$$z_i^k = 0 \quad \forall k > i, i \in V \quad \forall k \in \{1, \dots, K_{max}\} \quad (3e)$$

$$\sum_{i \in V} z_i^k \geq 1 \quad \forall k \in \{1, \dots, K_{max}\} \quad (3f)$$

$$\sum_{i \in V} w_i^k z_i^k \leq B \quad \forall k \in \{1, \dots, K_{max}\}, w_i^2 \in U^2 \quad (3g)$$

$$z_i^k \in \{0, 1\} \quad \forall i \in V \quad \forall k \in \{1, \dots, K_{max}\} \quad (3h)$$

$$x_{ij} \in \{0, 1\} \quad \forall ij \in E \quad (3i)$$

$$z \geq \sum_{ij \in E} l_{ij}^1 x_{ij} \quad \forall ij \in E \quad (3j)$$

La robustesse n'apparaît plus dans l'objectif mais dans la contrainte (3j) dans ce modèle.

(b) On peut définir U^{0*} (resp U^{k*}) comme étant l'ensemble des distances initiales proposées (resp l'ensemble des poids initiaux) dans le modèle statique et on a $l_{ij}^0 = l_{ij}$ pour tout l_{ij}^0 dans U^{0*} (resp on a $w_i^k = w_i$ pour tout w_i^k dans U^{k*}) .

(c) Le sous-problème lié à U^{0*} :

$$\max_{l_{ij}^0 \in U^{0*}} \sum_{ij \in E} l_{ij}^0 x_{ij}^*$$

avec x^* la solution optimale du problème maître.

Ce qui est équivalent à :

$$\max_{\delta} \sum_{ij \in E} (l_{ij} + \delta_{ij}^0 (\hat{l}_i + \hat{l}_j)) x_{ij}^*$$

sous les contraintes :

$$\begin{aligned} \sum_{ij \in E} \delta_{ij}^0 &\leq L \\ \delta_{ij}^0 &\in [0, 3] \quad \forall ij \in E \end{aligned}$$

on a k_{max} sous-problèmes liés à U^{k*} :

$$\max_{w_i^k \in U^{k*}} \sum_{i \in V} w_i^k z_i^{k*}$$

avec z^* la solution optimale du problème maître.

Ce qui est équivalent à :

$$\max_{\delta} \sum_{i \in V} w_i^k (1 + \delta_i^k) z_i^{k*}$$

sous les contraintes :

$$\begin{aligned} \sum_{i \in V} \delta_i^k &\leq W \\ \delta_i^k &\in [0, W_i] \quad \forall i \in V \end{aligned}$$

(d) Les conditions à satisfaire pour qu'une solution soit optimale :

On trouve Z^* (valeur objectif de problème maître), x^* et z^* en résolvant le problème maître.

On trouve δ^{*0} en résolvant le sous-problème lié à U^{0*} et on trouve alors l^{0*} .

On trouve $\delta^{*1}, \dots, \delta^{*m}$ en résolvant les k_{max} sous-problèmes liés à U^{k*} et on trouve alors w^{1*}, \dots, w^{m*} .

Il faut que $\sum_{ij \in E} l_{ij}^{0*} x_{ij}^* \leq Z^*$ et $\sum_{i \in V} w_i^{k*} z_i^{k*} \leq B \quad \forall k \in \{1, \dots, K_{max}\}$ pour que la solution (x^*, z^*) soit optimale.

(e) Si on trouve que l^{0*} (et/ou w^{1*}, \dots, w^{m*}) est tel que $\sum_{ij \in E} l_{ij}^{0*} x_{ij}^* > Z^*$ (et/ou $\sum_{i \in V} w_i^{k*} z_i^{k*} > B$), on doit ajouter l^{0*} à U^{0*} (et/ou w^{k*} à U^{k*}) donc la coupe qui s'ajoute est la suivante :

$$l_{ij}^{0*} x_{ij} < z \quad (\text{et/ou } \sum_{i \in V} w_i^{k*} z_i^k \leq B \quad \forall k \in \{1, \dots, K_{max}\}),$$

4. Résolution par dualisation :

(a) Reformulation de l'objectif :

$$\begin{aligned} & \min_{x,z} \max_{l_{ij}^0 \in U^0} \sum_{ij \in E} l_{ij}^0 x_{ij} \\ & = \min_{x,z} \max_{\delta_{ij}^0} \sum_{ij \in E} (l_{ij} + \delta_{ij}^0 (\hat{l}_i + \hat{l}_j)) x_{ij} \\ & = \min_{x,z} \sum_{ij \in E} l_{ij} x_{ij} + \min_{x,z} \max_{\delta_{ij}^0} \sum_{ij \in E} \delta_{ij}^0 (\hat{l}_i + \hat{l}_j) x_{ij} \end{aligned}$$

(b) Le problème interne est :

$$\max_{\delta} \sum_{ij \in E} \delta_{ij}^0 (\hat{l}_i + \hat{l}_j) x_{ij}$$

sous les contraintes :

$$\begin{aligned} \sum_{ij \in E} \delta_{ij}^0 & \leq L \\ \delta_{ij}^0 & \in [0, 3] \quad \forall ij \in E \end{aligned}$$

(c) Dualisation :

$$\min \alpha^0 L + \sum_{ij \in E} 3\beta_{ij}^0$$

sous les contraintes :

$$\begin{aligned} \alpha^0 + \beta_{ij}^0 & \geq (\hat{l}_i + \hat{l}_j) x_{ij} \quad \forall ij \in E \\ \alpha^0 & \geq 0 \\ \beta_{ij}^0 & \geq 0 \end{aligned}$$

(d) Reformulation des contraintes :

$$\begin{aligned} & \min_{x,z} \max_{w_i^k \in U^k} \sum_{i \in V} w_i^k x_{ij} \\ & = \min_{x,z} \max_{\delta_i^k} \sum_{i \in V} w_i (1 + \delta_i^k) z_i^k \\ & = \min_{x,z} \sum_{i \in V} w_i z_i^k + \min_{x,z} \max_{\delta_i^k} \sum_{i \in V} w_i \delta_i^k z_i^k \end{aligned}$$

(e) Le problème interne est :

$$\max_{\delta_i^k} \sum_{i \in V} w_i \delta_i^k z_i^k$$

sous les contraintes :

$$\begin{aligned} \sum_{i \in V} \delta_i^k & \leq W \\ \delta_i^k & \in [0, W_i] \quad \forall i \in V \end{aligned}$$

(f) Dualisation :

$$\min \alpha^k W + \sum_{i \in V} W_i \beta_i^k$$

sous les contraintes :

$$\begin{aligned} \alpha^k + \beta_i^k & \geq w_i \delta_i^k z_i^k \quad \forall i \in V \\ \alpha^k & \geq 0 \\ \beta_i^k & \geq 0 \end{aligned}$$

(g) On remplace les 2 \min par un seul \min et on obtient le problème robuste dualisé suivant :

$$= \min_{x,z,\alpha,\beta} \sum_{ij \in E} l_{ij} x_{ij} + \alpha^0 L + \sum_{ij \in E} 3\beta_{ij}^0$$

sous les contraintes :

$$\begin{aligned} x_{ij} + z_i^k - z_j^k & \leq 1 \quad \forall ij \in E \quad \forall k \in \{1, \dots, K_{max}\} \\ x_{ij} - z_i^k + z_j^k & \leq 1 \quad \forall ij \in E \quad \forall k \in \{1, \dots, K_{max}\} \\ -x_{ij} + z_i^k + z_j^k & \leq 1 \quad \forall ij \in E \quad \forall k \in \{1, \dots, K_{max}\} \\ \sum_{k=1}^{K_{max}} z_i^k & = 1 \quad \forall i \in V \\ z_i^k & = 0 \quad \forall k > i, i \in V \quad \forall k \in \{1, \dots, K_{max}\} \\ \sum_{i \in V} z_i^k & \geq 1 \quad \forall k \in \{1, \dots, K_{max}\} \\ \sum_{i \in V} w_i z_i^k + \alpha^k W + \sum_{i \in V} W_i \beta_i^k & \leq B \quad \forall k \in \{1, \dots, K_{max}\} \\ \alpha^0 + \beta_{ij}^0 & \geq (\hat{l}_i + \hat{l}_j) x_{ij} \quad \forall ij \in E \\ \alpha^0 & \geq 0 \\ \beta_{ij}^0 & \geq 0 \end{aligned}$$

$$\begin{aligned}
\alpha^k + \beta_i^k &\geq w_i \delta_i^k z_i^k & \forall i \in V \quad \forall k \in \{1, \dots, K_{max}\} \\
\alpha^k &\geq 0 \\
\beta_i^k &\geq 0 \\
z_i^k &\in \{0, 1\} & \forall i \in V \quad \forall k \in \{1, \dots, K_{max}\} \\
x_{ij} &\in \{0, 1\} & \forall ij \in E
\end{aligned}$$

3 Résolution numérique

Le code de l'implémentation est dans le même répertoire que ce rapport. On a implémenté 4 algorithmes de résolution numérique du PLNE en question, à savoir l'algorithme de dualisation, une heuristique, l'algorithme de plans coupants, et l'algorithme de Branch & Cut.

3.1 Résolution statique compacte

Nous commençons par la résolution du problème statique en nombre entier afin de comparer l'efficacité et la robustesse des algorithmes implémentés. Notons que nous implémentons le PLNE donné dans la correction. (Voir fichier : *PbStatique.jl*)

Le problème statique s'écrit :

$$(P_S) : \begin{cases} \min_{x,z} \sum_{ij \in E} l_{ij} x_{ij} & (11) \\ \text{s.c. } x_{ij} + z_i^k - z_j^k \leq 1 & \forall ij \in E, \forall k \in \{1, \dots, K_{max}\} & (12) \\ x_{ij} - z_i^k + z_j^k \leq 1 & \forall ij \in E, \forall k \in \{1, \dots, K_{max}\} & (13) \\ -x_{ij} + z_i^k + z_j^k \leq 1 & \forall ij \in E, \forall k \in \{1, \dots, K_{max}\} & (14) \\ \sum_{k=1}^{K_{max}} z_i^k = 1 & \forall i \in V & (15) \\ \sum_{i \in V} w_i z_i^k \leq B & \forall k \in \{1, \dots, K_{max}\} & (16) \\ x_{ij} \in \{0, 1\} & \forall ij \in E & (17) \\ z_i^k \in \{0, 1\} & \forall i \in V, \forall k \in \{1, \dots, K_{max}\} & (18) \end{cases}$$

3.2 Résolution par dualisation

En ce qui concerne la dualisation, on a implémenté avec julia, le modèle du PLNE dualisé élaboré dans la question 4 de la partie modélisation sur papier. C'est une méthode exacte de résolution du problème. (Voir fichier : *Dualité.jl*)

3.3 Résolution par les plans coupants

- On a suivi la démarche élaborée dans la première question de la partie modélisation du projet.
- On part d'un problème appelé le problème maître tel que, dans la contrainte sur laquelle porte l'incertitude, l'ensemble des scénarios possibles U^{0*} est initialisé à $l_{ij}^0 = l_{ij}$ pour tout l_{ij}^0 dans U^{0*} .
- On a implémenté une boucle, qui résout à chaque itération le problème maître et le sous-problème associé puis génère la coupe associée (telle qu'elle est explicitée dans la question 3 de la partie modélisation), celle-ci sera enregistrée et dans le cas où elle est violée, elle sera alors ajoutée au problème maître et une autre itération de la boucle sera lancée.
- La boucle s'arrête dès qu'on trouve une coupe non violée.
- Ce qu'on doit noter, en ce qui concerne cet algorithme, est que l'arbre de résolution élaboré par cplex pour résoudre le problème maître (ainsi que la résolution de sous-problème associé) est refait à partir de sa racine à chaque fois qu'on trouve une coupe violée et donc les coupes sont ajoutées à chaque itération et pas au niveau des noeuds. (Voir fichier : *PLANSCOUPANTS.jl*)

3.4 Résolution par la méthode de Branch and Cut

Il s'agit d'un algorithme qui utilise les mêmes coupes que la méthode des plans coupants élaborées dans la question 3 dans la partie modélisation papier du rapport, sauf que cette fois-ci, on n'a plus à refaire tout l'arbre de branchement du problème maître en ajoutant les coupes violées à la racine comme on l'a fait pour les plans coupants, la résolution du sous-problème ainsi que l'ajout des coupes violées se fait à chaque noeud en appelant

la fonction implémentée dans le code *my_cb_function* à chaque noeud de résolution (à part les coupes déjà ajoutées par défaut par Cplex dans chaque noeud pour diminuer le nombre de noeuds de l'arbre de recherche). (Voir fichier : *B&C(lazycallback).jl*)

3.5 Résolution par une heuristique

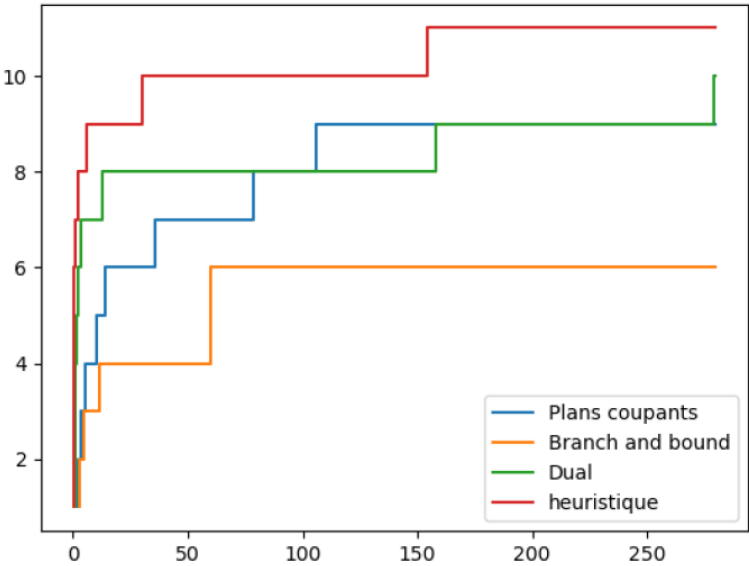
on se sert du problème maître de plans coupants, qu'on exécute sur le même problème, mais avec les poids fixés à leur maximum. On obtient ainsi une solution réalisable. Ensuite on résout le sous problème lie a l'incertitude sur les distances, et on renvoie son optimum. On obtient une borne supérieure de l'optimum, souvent très proche de l'optimum lui-même, compte tenu des données. (Voir fichier : *Heuristique.jl*)

3.6 Analyse de performance

On obtient les résultats suivants :

Instance	Opt robuste	Opt statique	Plans coupants		Brunch & Cut		Dualisation		Heuristique	
			Time	Gap	Time	Gap	Time	Gap	Time	Gap
10_ulysses_3	136.995	54.35	5.5	0%	4.95	0%	2.3	0%	0.04	0%
10_ulysses_6	55.1194	7.22	3.68	0%	11.2	0%	3.11	0%	0.05	0%
10_ulysses_9	33.29	0.72	13.8	0%	60	0%	0.756	0%	0.03	27%
14_burma_3	93.39	66.2137	2	0%	2.68	0%	0.87	0%	0.08	0%
14_burma_6	42.7406	17.9	9.96	0%	60	0%	1.06	0%	0.15	0%
14_burma_9	20.7624	4.72	35.92	0%			1.32	0%	0.10	0%
22_ulysses_3	358.637	284.21	78.53	0%	356	0%	13	0%	6.01	0.2%
22_ulysses_6	116.529	82.631	600	1%			279	0%	30	0%
22_ulysses_9	64.9736	26.21					158	0%	2	0%
26_eil_3	2227.63	1850					604	0%	154	0%

on obtient le diagramme de performance suivant :



Pour construire ce tableau, on a testé les 4 algorithmes sur les jeux de données fournis . Pour calculer la valeur de PR (en %) : $100 \frac{v_{robuste} - v_{statique}}{v_{robuste}}$ où $v_{robuste}$ est la valeur optimale du problème robuste et $v_{statique}$ est la valeur optimale du problème statique. (les cases vide correspondent aux instances où la résolution du problème robuste prend beaucoup de temps).

Pour calculer le gap (en %) : $100 \frac{v_{meilleure} - v_{relax}}{v_{meilleure}}$ où $v_{meilleure}$ est le meilleur résultat réalisable pour le problème robuste et v_{relax} est la valeur de la relaxation continue du problème dualisé. (dans notre cas la meilleure valeur réalisable est celle donnée par la dualisation)

Les case vides de la colonne de la méthode des plans coupants correspondent aux instances qui prennent beaucoup de temps pour la résolution optimale et qui n'ont pas une solution réalisable au bout du temps limite.

On remarque que, parmi les méthodes de résolution exactes (dualisation, plans coupants et branch and cut), la dualisation renvoie le plus grand nombre de solutions optimales dans les plus petits délais pour les instances données. Le tableau de valeurs ci dessus le montre clairement. Parmi les 10 instances testées , la dualisation trouve la solution optimale pour 7 en moins de 13 secondes, alors que l'algorithme de branch and cut et la méthode des plans coupants la trouvent pour 3 et 5 instances respectivement.

En plus, pour les instances où la valeur optimale n'a pas été trouvée au bout des 30 secondes par les 3 méthodes on a la dualisation qui donne la valeur réalisable la plus petite et donc la meilleure borne supérieure.

Donc, on note bien que la dualisation est la plus performante en terme de complexité temporelle à résoudre les instances qui ont été résolues optimalement par les 3 méthodes exactes.

L'algorithme de branch and cut a du mal à converger pour des instances relativement simple a résoudre par les autres méthodes. En fait, l'arbre de recherche est très grand parce qu'on ne trouve pas de bonne borne inférieur qui permettra soit de converger rapidement vers une solution entière (et donc de rajouter des coupes), soit de prouver l'optimalite de la dernière solution entière trouvée. En plus, par cette méthode on a plus de chance d'introduire des coupes inutiles qui vont alourdir le problème. Le mieux serait de trouver un compromis entre les différentes méthodes et de donner la possibilité d'utiliser les bornes supérieurs ou inférieurs trouvées par l'une d'elles pendant la résolution par une autre méthode.

En ce qui concerne l'heuristique, On remarque qu'elle est initialement bien meilleure que toutes les autres méthodes. Cependant avec l'augmentation du nombre de variables, il devient très long de la calculer à partir de 26 eil 3. Il semble au premier vu des résultats que l'heuristique soit plus performante, cependant elle donne ici des résultats très proches de l'optimal mais cela est du a la distribution des données, et pourrait changer dans une autre configuration.

4 Annexe

Les solutions optimales obtenues :

Instance : 10_ulysses_3

Valeur objectif : 136.99527629589417

Clusters : {1, 2, 3, 10}, {4, 6, 7, 8}, {5, 9}

Instance : 10_ulysses_6

Valeur objectif : 55.1194

Clusters : {2,3},{5},{6,9},{7},{10},{1,4,8}

Instance : 10_ulysses_9

Valeur objectif : 33,29

Clusters : {1},{2,3},{4},{5},{6},{7},{8},{9},{10}

Instance : 14_burma_3

Valeur objectif : 93.39

Clusters : {3,4,5,6,7}, {1,8,9,10,11}, {2,12,,13,14}

Instance : 14_burma_6

Valeur objectif : 42.7406

Clusters : {1,8,10},{2,13},{7},{9,10},{6,12,14},{3,4,5}

Instance : 14_burma_9

Valeur objectif : 20.7624

Clusters : {1,8},{2},{7},{6,19},{9,11},{3,4},{5},{13,14},{10}

Instance : 22_ulysses_3

Valeur objectif : 358.637

Clusters : {1, 2, 3, 4, 8,,16,17,18}, {5, 6, 9, 11,15,21}, {7,10,12,13,14,19}

Instance : 22_ulysses_6

Valeur objectif : 116.529

Clusters : {2,15,19,25},{5,14,20},{8,9,24,27},{7,11,16,18,26},{6,10,12,13,17},{3,4,21,22,23}

Instance : 22_ulysses_9

Valeur objectif : 64.9736

Clusters : {1,16,22},{5,9},{11,19},{7,13},{2,3,17},{6,10},{12,14,15},{4,8,18},{19,21}

Instance : 26_eil_3

Valeur objectif : 2227.63

Clusters : {1, 2, 3, 5,9,10,11,16,20,21}, {6, 7, 8, 14,22,23,24,25,26}, {4,12, 13,15,17,18,19}