




Graphes, Jeux et R.O.

Soutenance RO203



Julien Segonne
Baptiste Montagnes



1

Bridges

Résolution CPLEX



Génération des instances

On fixe une taille de grille n et une densité.

1. **Création de deux matrices : t et occupied_cells**
2. **Création d'un premier pont simple ou double**
3. **Tant que la densité n'est pas atteinte :**
 - ❑ Sélection d'un sommet non connecté à l'ensemble déjà créé,
 - ❑ Vérification que le sommet peut être relié à l'ensemble de sommets existants,
 - ❑ Vérification que le nouveau pont ne chevauche pas ou ne croise pas d'autres ponts déjà existants.

```
data > ≡ instance_t7_d0.2_1.txt
1      ,3, ,1, , ,
2      ,5, , ,3, ,
3      , , , , , ,
4      , ,1, ,2, ,
5      ,3,3, , , ,
6      2, ,3, , , ,
7      , , , , , ,
8      , , , , , ,
```

Choix de modélisation

$\forall i, V_i$ représente le nombre de ponts liés au sommet i .

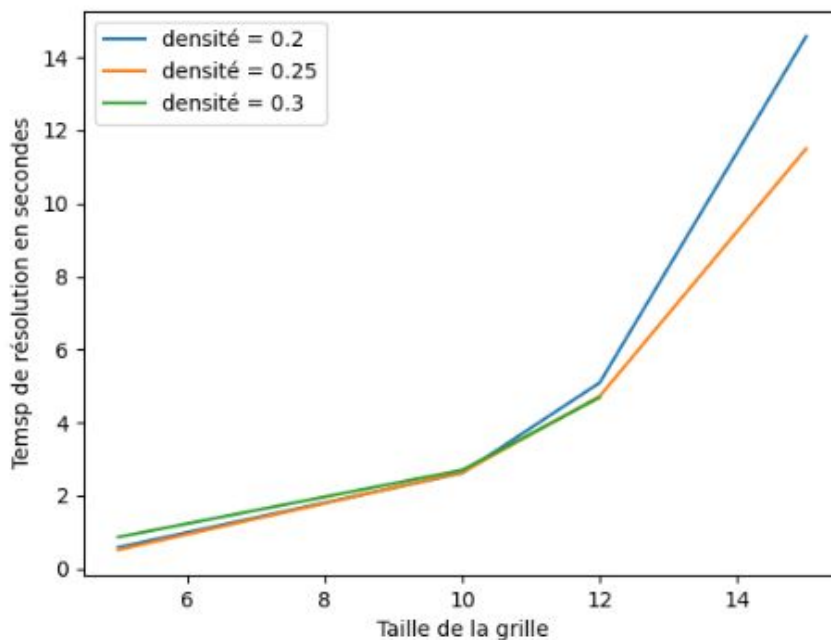
$\forall i, j \in \llbracket 1, n \rrbracket, x_{ij}$ représente le nombre de ponts reliant le sommet i au sommet j .

$$(P_1) \left\{ \begin{array}{l} \text{Maximiser } B \\ \text{s.c. :} \\ \forall i \in \llbracket 1, n \rrbracket, \sum_{j \neq i} x_{ij} = V_i \\ \forall i, j \in \llbracket 1, n \rrbracket, x_{ij} = x_{ji} \\ \forall i, j \in \llbracket 1, n \rrbracket, 0 \leq x_{ij} \leq 2 \\ \forall i, j, k, l \in \llbracket 1, n \rrbracket \text{ tels qu'un pont reliant } i \text{ à } j \text{ et un pont reliant } k \text{ à } l \text{ se croisent,} \\ \quad x_{ij} = 0 \text{ ou } x_{lk} = 0 \end{array} \right.$$

Résultats obtenus

Instance	cplex	
	Temps (s)	Optimal ?
instance_t5_d0.25_1.txt	0.131	×
instance_t5_d0.25_2.txt	0.059	×
instance_t5_d0.2_1.txt	0.053	×
instance_t5_d0.2_2.txt	0.066	×
instance_t5_d0.3_1.txt	0.048	×
instance_t5_d0.3_2.txt	0.05	×
instance_t7_d0.25_1.txt	0.097	×
instance_t7_d0.25_2.txt	0.097	×
instance_t7_d0.2_1.txt	0.084	×
instance_t7_d0.2_2.txt	0.131	×
instance_t7_d0.3_1.txt	0.085	×
instance_t7_d0.3_2.txt	0.083	×
instance_t9_d0.25_1.txt	0.167	×
instance_t9_d0.25_2.txt	0.17	×
instance_t9_d0.2_1.txt	0.557	×
instance_t9_d0.2_2.txt	0.209	×
instance_t9_d0.3_1.txt	0.199	×
instance_t9_d0.3_2.txt	0.16	×

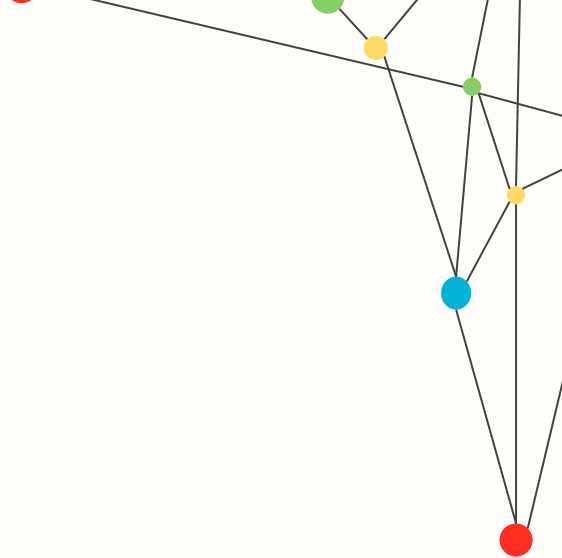
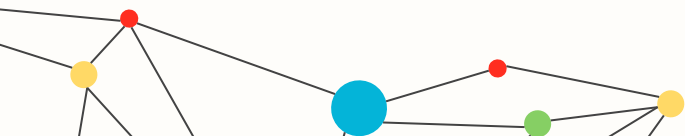
Pour une densité de 0.2, les méthodes de génération et résolution d'instances fonctionnent jusqu'à une taille de 21.





Difficultés rencontrées

- Nous n'avons pas réussi à résoudre des instances pour lesquelles la solution comporte une boucle,
- Création d'une liste contenant les voisins d'un sommet,
- Affichage graphique de la solution dans la console.





2

Singles

Résolution CPLEX et heuristique



Génération des instances

On fixe une taille de grille n et une densité d .

1. Création d'une matrice comme ci-contre
2. Echange aléatoire de colonnes et de lignes 2 à 2
3. Suppression aléatoire de cases de la grille (tant que **densité > d**) : en vérifiant qu'une case adjacente ne soit pas déjà supprimée et que cela ne crée pas de formes géométriques particulières
4. Remplissage des cases supprimées par des nombres aléatoires entre 1 et n

1	2	3	4	...
n	1	2	3	...
$n-1$	n	1	2	...
$n-2$	$n-1$	n	1	...
\vdots	\vdots	\vdots	\vdots	\ddots

1	4,3,1,2,5
2	1,5,3,4,2
3	5,4,4,3,1
4	2,1,4,5,3
5	3,2,5,1,1

Choix de modélisation

$\forall i, j \in \llbracket 1, n \rrbracket, \forall k \in \llbracket 0, n \rrbracket, x_{ijk}$ vaut 1 si la valeur se trouvant (i,j) vaut k et 0 sinon.

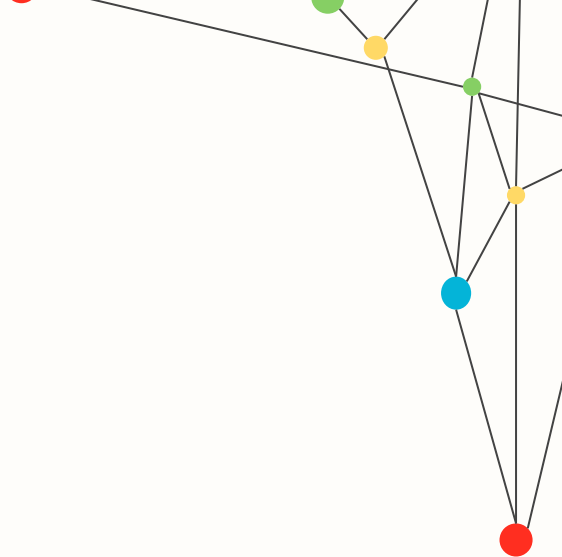
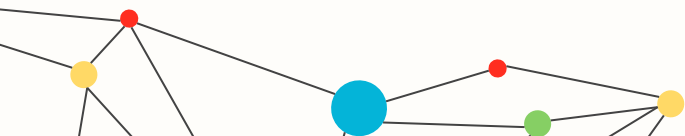
$$(P_2) \left\{ \begin{array}{l} \text{Maximiser } B \\ \text{s.c. :} \\ \forall j, k \in \llbracket 1, n \rrbracket, \sum_{i=1}^n x_{ijk} \leq 1 \\ \forall i, k \in \llbracket 1, n \rrbracket, \sum_{j=1}^n x_{ijk} \leq 1 \\ \forall i \in \llbracket 1, n-1 \rrbracket, \forall j \in \llbracket 1, n \rrbracket, x_{i,j,0} + x_{i+1,j,0} \leq 1 \\ \forall i \in \llbracket 1, n \rrbracket, \forall j \in \llbracket 1, n-1 \rrbracket, x_{i,j,0} + x_{i,j+1,0} \leq 1 \\ \text{L'ensemble des cases blanches est connexe} \end{array} \right.$$

PLNE Singles



Résolution heuristique

1. On parcourt de la grille pour compter les nombres qui **apparaissent le plus souvent** : ils seront masqués prioritairement
2. On masque les coins si besoin en premier
3. On parcourt le reste des points **par ordre de fréquence** : si un point apparaît déjà sur une ligne et une colonne et n'a pas de case noire adjacente, on le masque



Résultats obtenus

Instance	cplex		heuristique	
	Temps (s)	Optimal ?	Temps (s)	Optimal ?
instance.t10.1.txt	0.08	×	0.08	×
instance.t10.10.txt	0.08	×	0.08	×
instance.t10.2.txt	0.08	×	0.08	×
instance.t10.3.txt	0.08	×	-	-
instance.t10.4.txt	0.08	×	-	-
instance.t10.5.txt	0.1	×	-	-
instance.t10.6.txt	0.12	×	-	-
instance.t10.7.txt	0.12	×	-	-
instance.t10.8.txt	0.12	×	-	-
instance.t10.9.txt	0.12	×	-	-
instance.t4.1.txt	0.04	×	-	-
instance.t4.10.txt	0.04	×	-	-
instance.t4.2.txt	0.12	×	-	-
instance.t4.3.txt	0.03	×	-	-
instance.t4.4.txt	0.03	×	-	-
instance.t4.5.txt	0.04	×	-	-
instance.t4.6.txt	0.04	×	-	-
instance.t4.7.txt	0.05	×	-	-
instance.t4.8.txt	0.04	×	-	-
instance.t4.9.txt	0.04	×	-	-
instance.t5.1.txt	0.05	×	-	-
instance.t5.10.txt	0.04	×	-	-
instance.t5.2.txt	0.04	×	-	-
instance.t5.3.txt	0.04	×	-	-
instance.t5.4.txt	0.05	×	-	-
instance.t5.5.txt	0.05	×	-	-
instance.t5.6.txt	0.06	×	-	-
instance.t5.7.txt	0.12	×	-	-
instance.t5.8.txt	0.07	×	-	-

Le temps augmente avec la dimension de la grille.

Le temps de résolution devient trop long à partir d'une grille de taille 42 x 42.

Difficultés

- Décomposer la contrainte de connexité : possibilité d'implémenter toutes les formes possibles mais le nombre est exponentielle avec la dimension
- Trouver une méthode heuristique fonctionnelle
- Générer des instances sans créer une grille et tester si elle est résolvable

6	12	11		7	10	2	2	8	12	5	10
4	2		3		6	10	8	1	12	2	1
1		2	10	6		2	7	4	7	1	6
	10	1	10		8	7	9	2	7	3	2
2		11		12	2		11	1	4	12	10
5	4		12	11		8		5	8	6	2
8	4	5	8		10	6	4		2	12	9
5	6	4		3	1	12	12	7		8	1
12	2	1	7		3	3	1	9	9		12
1	12	6	9	5		7	12	8		9	6
7	1	12	8	11	3		5		3	10	6
10	10	3	9	3	4	1		12	4	11	9



**Merci pour votre
attention !**