Modélisation, Optimisation, Graphes, Programmation Linéaire : Compte rendu de Projet

CONTESENNE Sami et KADHI Youssef

1 Répartition de patients dans les unités de soin

1.1

On cherche à affecter chaque ville à un des k secteurs permettant de minimiser la distance moyenne de chaque habitant d'une ville a son unité de soin.
On définit donc des variables de décision booléennes x_{ij} qui déterminent si la ville i est affectée au secteur j (valeur 1) ou non (valeur 0).
On définit également les constantes suivantes :
$\gamma = \frac{1+\alpha}{k} \sum_{i=1}^{n} v_i$ (comme dans l'énoncé).
$p = \sum_{i=1}^{n} v_i$ (somme des populations de toutes les villes).

La fonction à minimiser pour résoudre ce problème est alors : Min $z = \displaystyle\sum_{i=1}^{n} \sum_{j=1}^{k} d_{ij} x_{ij} \frac{v_i}{p}$

On a d'abord des contraintes portants sur les capacités des secteurs. Pour chaque secteur j, le total des populations affectées à ce secteur s'écrit $\sum_{i=1}^{n} v_i x_{ij}$. On obtient donc k contraintes de la forme : $\sum_{i=1}^{n} v_i x_{ij} \leq \gamma \quad j \in \{1 \ldots k\}$ dans le programme linéaire (PL).

Ensuite pour s'assurer qu'une ville est bien affectée à un unique secteur, on a besoin de nouvelles contraintes. Pour chaque ville i, le nombre de secteurs auxquels cette ville est affectée s'écrit $\sum_{j=1}^{k} x_{ij}$. On a donc n contraintes de la forme $\sum_{j=1}^{k} x_{ij} = 1 \quad i \in \{1 \ldots n\}$ dans le PL.
Ces contraintes sont utiles non pas parce qu'il est possible qu'une ville soit affectée à plusieurs secteurs (en effet cela augmenterait la fonction objectif et génèrerait donc une solution non optimale), mais plutôt pour ne pas avoir de ville affectée à aucun secteur.

Finalement le PL modélisant le problème est le suivant :

$$\text{Min } z = \sum_{i=1}^{n} \sum_{j=1}^{k} d_{ij} x_{ij} \frac{v_i}{p}$$
$$\begin{cases} \sum_{i=1}^{n} v_i x_{ij} \leq \gamma \quad j \in \{1 \ldots k\} \\ \sum_{j=1}^{k} x_{ij} = 1 \quad i \in \{1 \ldots n\} \\ x_{ij} \in \{0, 1\} \end{cases}$$

On ne peut pas modéliser ce problème avec un algorithme de flot maximum à coût minimum. En effet, une première idée consisterait à prendre pour flot les populations, pour coût les distances d'une ville à l'autre, mais dans ce cas il n'y a plus de possibilité pour représenter la contrainte portant sur les capacités de secteur (avec γ).

1.2

En implémentant ce programme linéaire et en prenant aléatoirement les villes de Strasbourg, Reims et Dijon on obtient comme résultat selon la valeur de α :

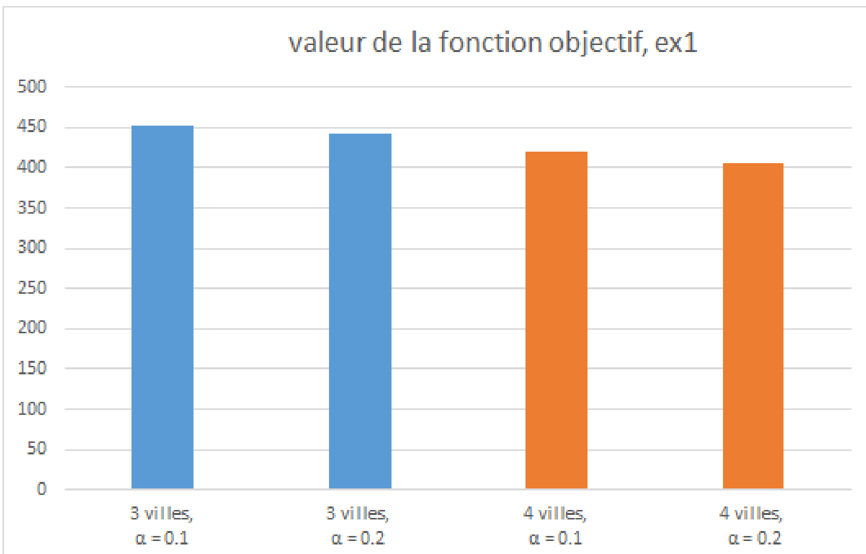
```
alpha = 0.1
Valeur de la fonction objectif : 453.6907869266377
------------------------------------

Les villes affectées au secteur Strasbourg sont :
['Nice' 'Strasbourg' 'Bordeaux' 'Grenoble']

Les villes affectées au secteur Reims sont :
['Nantes' 'Lille' 'Rennes' 'Reims' 'Le Havre' 'Angers']

Les villes affectées au secteur Dijon sont :
['Toulouse' 'Montpellier' 'Saint-Etienne' 'Toulon' 'Dijon']
```

```
alpha = 0.2
Valeur de la fonction objectif : 442.2120869687194
------------------------------------

Les villes affectées au secteur Strasbourg sont :
['Nice' 'Strasbourg' 'Angers']

Les villes affectées au secteur Reims sont :
['Nantes' 'Bordeaux' 'Lille' 'Rennes' 'Reims' 'Le Havre']

Les villes affectées au secteur Dijon sont :
['Toulouse' 'Montpellier' 'Saint-Etienne' 'Toulon' 'Grenoble' 'Dijon']
```

On ajoute a la liste la ville de Le Havre :

```
alpha = 0.1
Valeur de la fonction objectif : 419.2856888764203
------------------------------------

Les villes affectées au secteur Strasbourg sont :
['Nice' 'Strasbourg' 'Toulon' 'Grenoble']

Les villes affectées au secteur Reims sont :
['Bordeaux' 'Lille' 'Reims' 'Dijon']

Les villes affectées au secteur Le Havre sont :
['Nantes' 'Rennes' 'Le Havre' 'Angers']

Les villes affectées au secteur Dijon sont :
['Toulouse' 'Montpellier' 'Saint-Etienne']
```

```
alpha = 0.2
Valeur de la fonction objectif : 405.20424547622383
-----------------------------------

Les villes affectées au secteur Strasbourg sont :
['Nice' 'Strasbourg']

Les villes affectées au secteur Reims sont :
['Toulouse' 'Lille' 'Reims' 'Angers']

Les villes affectées au secteur Le Havre sont :
['Nantes' 'Bordeaux' 'Rennes' 'Le Havre']

Les villes affectées au secteur Dijon sont :
['Montpellier' 'Saint-Etienne' 'Toulon' 'Grenoble' 'Dijon']
```



On observe que quand α augmente, la fonction objectif s'améliore. En effet augmenter α rend les k premières contraintes plus permissives puisque γ augmente et il y a donc une plus grande capacité pour chaque secteur. On observe de plus que lorsqu'on ajoute de nouveaux secteurs la fonction objectif s'améliore. En effet ajouter de nouveaux secteurs permet de donner plus de choix d'affectation et donc potentiellement d'avoir un choix meilleur que celui avec moins de secteurs.

2 Localisation optimale des unités de soin

2.1

Les secteurs ne sont maintenant plus définis à l'avance. On repart du PL précédent auquel on va ajouter de nouvelles variables et contraintes. On a besoin de nouvelles variables de décision qui détermineront quelle ville est secteur.
On définit donc n variables $S_i = \begin{cases} 1 & \text{si la ville } i \text{ est secteur} \\ 0 & \textit{sinon} \end{cases}$
On veut avoir exactement k secteurs, on ajoute donc la nouvelle contrainte : $\sum_{i=1}^{n} S_i = k$.
De plus, on ne souhaite pas affecter de ville à d'autres villes non secteur. Un moyen de s'en assurer est d'avoir la propriété suivante : $\forall i$, $\forall j$, $S_j = 0 \Rightarrow x_{ij} = 0$.
La contrainte suivante $x_{ij} \leq S_j \quad j \in \{1 \dots n\} i \in \{1 \dots n\}$ permet de s'assurer que la propriété soit toujours satisfaite.
On peut noter que cette contrainte est inutile si $S_j = 1$ puisqu'on a déjà les autres contraintes assurant que tous les x_{ij} soient booléens et donc ≤ 1.

$$\text{Min } z = \sum_{i=1}^{n} \sum_{j=1}^{k} d_{ij} x_{ij} \frac{v_i}{p}$$

$$\begin{cases} \sum_{i=1}^{n} v_i x_{ij} \leq \gamma \quad j \in \{1 \dots k\} \\ \sum_{j=1}^{k} x_{ij} = 1 \quad i \in \{1 \dots n\} \\ \sum_{i=1}^{n} S_i = k \\ x_{ij} \leq S_j \quad j \in \{1 \dots n\} \quad i \in \{1 \dots n\} \\ x_{ij} \in \{0, 1\} \end{cases}$$

$k = 3$, $\alpha = 0.1$

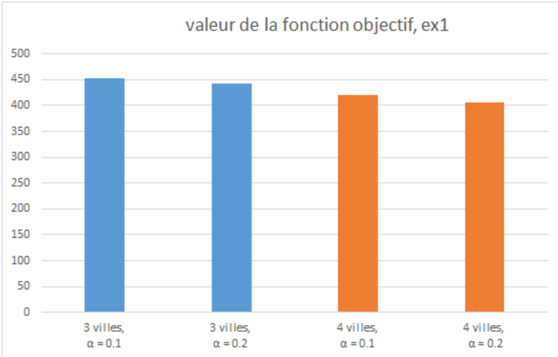
```
Valeur de la fonction objectif : 228.0857769673166
------------------------------------

Les secteurs sélectionnés sont :
['Nantes', 'Montpellier', 'Dijon']

Les villes affectées au secteur  Nantes sont :
['Nantes' 'Bordeaux' 'Rennes' 'Le Havre' 'Angers']

Les villes affectées au secteur  Montpellier sont :
['Toulouse' 'Nice' 'Montpellier' 'Toulon']

Les villes affectées au secteur  Dijon sont :
['Strasbourg' 'Lille' 'Reims' 'Saint-Étienne' 'Grenoble' 'Dijon']
```

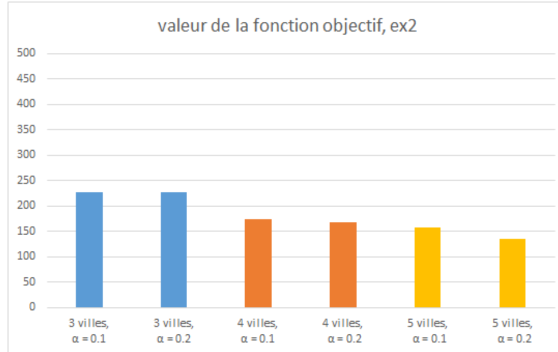
$k = 4$, $\alpha = 0.1$

```
Valeur de la fonction objectif : 174.34938729134524
------------------------------------

Les secteurs sélectionnés sont :
['Toulouse', 'Rennes', 'Reims', 'Toulon']

Les villes affectées au secteur  Toulouse sont :
['Toulouse' 'Bordeaux' 'Saint-Étienne']

Les villes affectées au secteur  Rennes sont :
['Nantes' 'Rennes' 'Le Havre' 'Angers']

Les villes affectées au secteur  Reims sont :
['Strasbourg' 'Lille' 'Reims' 'Dijon']

Les villes affectées au secteur  Toulon sont :
['Nice' 'Montpellier' 'Toulon' 'Grenoble']
```

$k = 5$, $\alpha = 0.1$

```
Valeur de la fonction objectif : 156.85562379015292
------------------------------------

Les secteurs sélectionnés sont :
['Toulouse', 'Nice', 'Nantes', 'Lille', 'Dijon']

Les villes affectées au secteur  Toulouse sont :
['Toulouse' 'Montpellier']

Les villes affectées au secteur  Nice sont :
['Nice' 'Toulon']

Les villes affectées au secteur  Nantes sont :
['Nantes' 'Bordeaux' 'Rennes']

Les villes affectées au secteur  Lille sont :
['Lille' 'Reims' 'Le Havre' 'Angers']

Les villes affectées au secteur  Dijon sont :
['Strasbourg' 'Saint-Étienne' 'Grenoble' 'Dijon']
```





On a les mêmes observations que dans l'exercice 1 en faisant varier α et le nombre de secteurs.
On observe que la nouvelle modélisation donne un bien meilleur résultat que la précédente avec une fonction objectif deux fois meilleure. En effet, choisir les villes secteurs de manière optimale est forcément meilleur que de les choisir au hasard.

2.2

Pour assurer l'accès au soin à tous, nous allons changer la fonction objectif du PL précédent pour minimiser la distance maximum d'un malade à son unité de soin.
On définit donc cette distance par une variable M comme fonction objectif et on ajoute en condition que M doit être supérieur ou égale à toutes les distances entre les villes et leurs secteurs.
On obtient le PL suivant :

$$\text{Min } z = M$$
$$\begin{cases} \sum_{i=1}^{n} v_i x_{ij} \leq \gamma \quad j \in \{1 \dots k\} \\ \sum_{j=1}^{k} x_{1j} = 1 \quad i \in \{1 \dots n\} \\ \sum_{i=1}^{n} S_i = k \\ x_{ij} \leq S_j \quad j \in \{1 \dots n\} i \in \{1 \dots n\} \\ \sum_{j=1}^{k} d_{ij} x_{ij} \leq M \quad i \in \{1 \dots n\} \\ x_{ij} \in \{0, 1\}, \quad M \in \mathbb{N} \end{cases}$$

$k = 3$, $\alpha = 0.1$

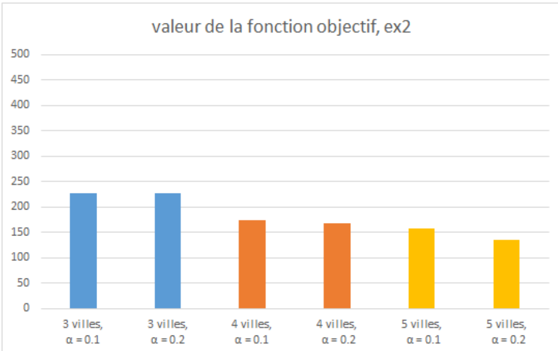
```
Valeur de la fonction objectif : 458.0
-------------------------------------
Les secteurs sélectionnés sont :
['Bordeaux', 'Reims', 'Toulon']

Les villes affectées au secteur  Bordeaux sont :
['Toulouse' 'Nantes' 'Bordeaux' 'Rennes']

Les villes affectées au secteur  Reims sont :
['Strasbourg' 'Lille' 'Reims' 'Le Havre' 'Dijon' 'Angers']

Les villes affectées au secteur  Toulon sont :
['Nice' 'Montpellier' 'Saint-Étienne' 'Toulon' 'Grenoble']
```

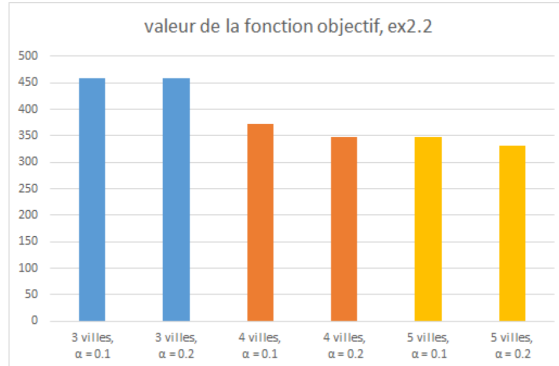
$k = 4$, $\alpha = 0.1$

```
Valeur de la fonction objectif : 372.0
-------------------------------------
Les secteurs sélectionnés sont :
['Nantes', 'Montpellier', 'Reims', 'Grenoble']

Les villes affectées au secteur  Nantes sont :
['Nantes' 'Bordeaux' 'Rennes' 'Angers']

Les villes affectées au secteur  Montpellier sont :
['Toulouse' 'Nice']

Les villes affectées au secteur  Reims sont :
['Strasbourg' 'Lille' 'Reims' 'Le Havre']

Les villes affectées au secteur  Grenoble sont :
['Montpellier' 'Saint-Étienne' 'Toulon' 'Grenoble' 'Dijon']
```

$k = 5$, $\alpha = 0.1$

```
Valeur de la fonction objectif : 347.0
-------------------------------------
Les secteurs sélectionnés sont :
['Toulouse', 'Nantes', 'Montpellier', 'Le Havre', 'Dijon']

Les villes affectées au secteur  Toulouse sont :
['Toulouse' 'Montpellier']
Les villes affectées au secteur  Nantes sont :
['Nantes' 'Bordeaux' 'Angers']
Les villes affectées au secteur  Montpellier sont :
['Nice' 'Saint-Étienne' 'Toulon']
Les villes affectées au secteur  Le Havre sont :
['Lille' 'Rennes' 'Le Havre']
Les villes affectées au secteur  Dijon sont :
['Strasbourg' 'Reims' 'Grenoble' 'Dijon']
```





On obtient cette fois-ci de moins bonnes solutions que dans la modélisation de 2-1. C'est normal puisqu'on ajoute une condition supplémentaire au PL, cependant cette solution a le mérite de considérer toutes les villes avec le même poids et ne pas avantager une plus qu'une autre.

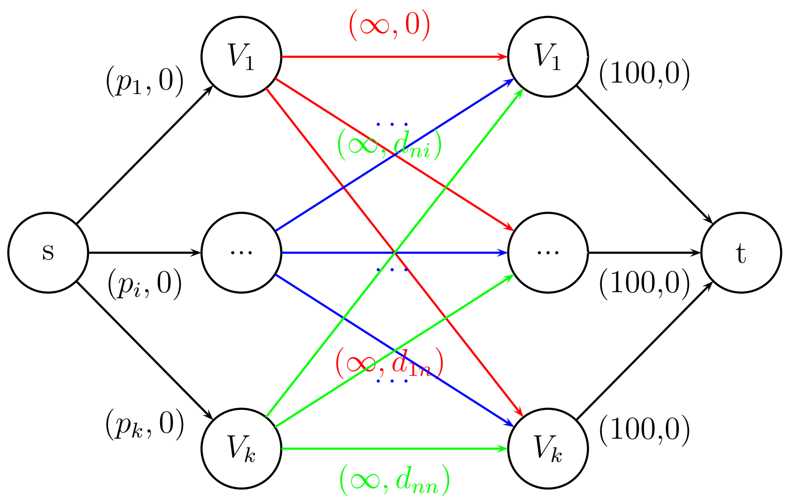
3 Équilibrage des charges des unités de soin

3.1

On souhaite équilibrer le nombre de patients dans chaque unité de soin de sorte qu'il n'y ait pas plus de 100 patients par unité, pour cela on déplace les patients d'une unité à l'autre tout en minimisant le coût du déplacement.
On peut modéliser ce problème comme un problème de transport comme suit : on définit n^2 variables de décisions x_{ij} qui correspondent aux nombres de personnes déplacées de l'unité i à l'unité j.
On souhaite qu'il y ait au maximum 100 personnes par unité, on peut modéliser cela par les contraintes suivantes : $\sum_{i=1}^{k} x_{ij} \leq 100 \quad j \in \{1 \dots k\}$.
On souhaite également avoir p_i personne dans l'unité de soin i à l'état initial, cela peut être modélisé par les contraintes : $\sum_{j=1}^{k} x_{ij} = p_i \quad i \in \{1 \dots k\}$.

$$\text{Min } z = \sum_{i=1}^{k} \sum_{j=1}^{k} d_{ij} x_{ij}$$
$$\begin{cases} \sum_{i=1}^{k} x_{ij} \leq 100 \quad j \in \{1 \dots k\} \\ \sum_{j=1}^{k} x_{ij} = p_i \quad i \in \{1 \dots k\} \\ x_{ij} \in \mathbb{N} \end{cases}$$

Le graphe du problème :



3.2

En prenant les 5 villes résultat de la question précédente c'est a dire : Toulouse, Nantes, Montpellier, Le Havre, Dijon.
Et $p = [150, 25, 125, 50, 75]$

```
Solution optimale:
[[100.  50.   0.  -0.  -0.]
 [ -0.  25.  -0.  -0.  -0.]
 [  0.  -0. 100.  -0.  25.]
 [ -0.  -0.  -0.  50.  -0.]
 [ -0.  -0.  -0.  -0.  75.]]

La ville de Toulouse envoie 50 patients à la ville de Nantes
La ville de Montpellier envoie 25 patients à la ville de Dijon

p en sortie:
[100.  75. 100.  50. 100.]

Valeur de la fonction objectif : 41550.0
```

$p = [38, 67, 127, 212, 56]$

```
Solution optimale:
[[ 38.  -0.  -0.  -0.  -0.]
 [ -0.  67.  -0.   0.  -0.]
 [ 27.  -0. 100.  -0.  -0.]
 [ 35.  33.  -0. 100.  44.]
 [ -0.  -0.  -0.  -0.  56.]]

La ville de  Montpellier envoie 27 patients à la ville de  Toulouse
La ville de  Le Havre envoie 35 patients à la ville de  Toulouse
La ville de  Le Havre envoie 33 patients à la ville de  Nantes
La ville de  Le Havre envoie 44 patients à la ville de  Dijon

p en sortie:
[100. 100. 100. 100. 100.]

Valeur de la fonction objectif : 71150.0
```

$p = [87, 100, 12, 43, 51]$

```
Solution optimale:
[[ 87.  -0.  -0.  -0.  -0.]
 [ -0. 100.  -0.  -0.  -0.]
 [ -0.  -0.  12.  -0.  -0.]
 [ -0.  -0.  -0.  43.  -0.]
 [ -0.  -0.  -0.  -0.  51.]]

p en sortie:
[ 87. 100.  12.  43.  51.]

Valeur de la fonction objectif : 0.0
```