PROJET SEMESTRIEL : DÉCOUPAGE ÉLECTORAL

ISRAA BEN SASSI, CAMILLE GOUJET ET VICTOR LESUR
À L'ATTENTION DE ARNAUD KNIPPEL

FIGURE 0.0.1. Illustration du découpage électoral en France

Contents

Abstract. Le découpage électoral est le mécanisme par lequel le territoire national est subdivisé en circonscriptions électorales, destinées à permettre et favoriser l'expression des citoyens par le vote.
Ce découpage est l'objet récurrent de polémiques, sinon d'actions en justice. Car il est orchestré, par exemple aux Etats Unis, au niveau de chaque État par les gouverneurs et ceux-ci ont la fâcheuse tendance à faire en sorte de favoriser leur parti. Par exemple, en façonnant des circonscriptions qui fragmentent l'électorat de leur adversaire, pour limiter son nombre d'élus. Ou, au contraire, en rassemblant dans une seule et même circonscription un maximum de votes adverses : ainsi, même avec 80% des voix, cela ne fera jamais qu'un seul élu dans la circonscription. C'est ce que l'on appelle gâcher des voix. Il arrive ainsi qu'à l'échelle d'un pays, un parti a moins d'élus qu'un autre alors qu'il a eu plus de voix. Un exemple? Une certaine Hillary Clinton a perdu la présidentielle en 2016 avec plus de 2 millions de votes d'avance sur son adversaire.

Ce projet consiste à étudier une ou plusieurs méthodes de découpage électoral, et à essayer de comprendre et démontrer dans quelle mesure on peut influencer le résultat d'élections par ce découpage.

Dans un premier temps, nous nous intéresserons à l'histoire de ce mécanisme et redéfinirons quelques notions de base pour que la compréhension du lecteur pour le reste du projet soit optimale.
Nous étudierons ensuite à partir de situations simples et faciles à modéliser de petites bases de données. À partir de celles-ci, nous modéliserons des programmes linéaires en nombres entiers que nous tenterons de résoudre pour en mesurer l'écart des résultats possibles et donc la manipulation partisane que l'on peut en faire pour un camp ou un autre..

Nous généraliserons nos modèles à des bases de données de plus en plus conséquentes et complexes et en étudierons les aspects.

Enfin, nous tenterons de remédier au problème d'équité qui pèse sur les partis à chaque nouvelle élection en élaborant une modélisation la plus équitable et représentative possible.

Afin de mener à bien ce projet, nous nous munirons d'un solveur tel que CPLEX pour nous assister dans la résolution de problèmes d'optimisation linéaire.

Part 1

PARTIE THÉORIQUE

CHAPTER 1

Introduction

1.1. Histoire

Le découpage électoral partisan des circonscriptions, aussi connu sous le nom de « *gerrymandering* », est une pratique qui n'est pas récente et qui est sûrement concomitante d'élections de représentants dans ces circonscriptions. Néanmoins, c'est avec l'apparition des démocraties représentatives modernes, d'abord les États-Unis puis la France, que la question est mis sur le devant de la scène. Ainsi, le terme gerrymandering nous vient des USA et fait référence à un découpage effectué par le gouverneur Elbridge Gerry. Ce découpage était si bizarre qu'il rappelait la forme d'une salamandre, d'où le mot-valise gerrymandering (composé de Gerry et de salamander du français salamandre).

De nos jours avec des outils de recensement d'une plus grande précision et des outils informatiques puissants, le découpage partisan fait toujours débat. En France, par exemple, il y a eu trois découpages électoraux sous la cinquième République (soit en 1958, 1986 et 2010). Dans ces trois cas, le gouvernement n'a nécessité uniquement d'une majorité absolue au parlement et non d'une majorité qualifiée (celle-ci est nécessaire pour un changement de constitution par exemple)[1]. Ainsi, il est plus facile au gouvernement de procéder au gerrymandering. Néanmoins, le gouvernement ne peut faire ce qu'il veut. En France, tout découpage électoral doit être sur une base essentiellement démographique, comme l'indique le conseil constitutionnel[2]. De fait, comme nous le verrons plus tard, le nombre d'habitants d'une circonscription doit être compris entre 80 et 120% de la moyenne des circonscriptions de la même région.

1.2. Quelques définitions importantes

Définition 1. **Découpage électoral**
Le découpage électoral est le mécanisme par lequel le territoire national est subdivisé en *circonscriptions électorales*, destinées à permettre et favoriser l'expression des citoyens par le vote.

Remarque. En France, le mécanisme du découpage électoral est utilisé pour les élections législatives, les élections sénatoriales et les élections au Conseil départemental.

Définition 2. **Circonscription électorale**
La circonscription électorale est une division du territoire effectuée dans le cadre d'une élection.
Chaque citoyen est rattaché à une circonscription et à une seule dans le cadre d'un vote. La circonscription peut être utilisée dans le cadre des scrutins majoritaires ou dans le cadre des scrutins de liste (scrutins proportionnels).

EXEMPLE. 10 000 électeurs, 10 circonscriptions de 1 000 électeurs, deux partis A et B, le premier recueillant 4 000 suffrages et le second 6 000.

- 6 circonscriptions gagnées par le parti A avec 53,3% des voix soit 3200 voix pour A et 2800 voix pour B
- 4 circonscriptions gagnées par le parti B avec 80% des voix, soit 800 voix pour A et 3200 voix pour B.

Le parti B perd les élections bien qu'il soit largement majoritaire en nombre de voix.

CHAPTER 2

Modéliser le découpage

2.1. Application sur l'exemple donné

Grâce à l'exemple ci-dessus, on remarque que le découpage électoral peut grandement influencer le résultat d'une élection. Bien modéliser ce découpage est alors essentiel; que notre but soit de favoriser un camp ou d'essayer de créer des conditions les plus équitables possible. Pour commencer, nous allons vous proposer la modélisation suivante :

Chaque circonscription se décompose en "brique" élémentaire que sont les cantons. Ces cantons seront dans notre modèle assimilés à des nœuds d'un graphe. Chaque canton est relié par des arêtes aux autres cantons limitrophes. Pour simplifier, nous allons considérer seulement deux partis A et B. Ainsi, on attribua un poids à chaque nœud correspondant à un réel entre 0 et 1 qui traduit le pourcentage de votants qui soutiennent le parti A. Donc si le poids est de 0, tous les votants soutiennent le parti B et si le poids est de 1, tous les votants soutiennent le parti A.

Créer les circonscriptions à partir du graphe revient à résoudre un problème de classification non supervisée. En effet, chaque circonscription ayant un seul élu, on peut influencer le nombre d'élus obtenu par un parti en découpant différemment les circonscriptions.

Pour rajouter un minimum de réalisme, on peut exiger que les circonscriptions soit d'un seul tenant, c'est-à-dire rajouter une **contrainte de connexité entre les nœuds d'une même classe.**

Si l'on reprend l'exemple précédent de **10 000 électeurs** à répartir sur **10 circonscriptions de 1 000 électeurs**, on peut imaginer que les électeurs se répartissent sur des **cantons de 200 électeurs** où 16 cantons soutiennent le parti B (poids 0), 28 soutiennent le parti A (poids 1) et 6 sont des cantons de poids 0,33.

Alors on peut répartir ces cantons de façon à obtenir :
- 4 circonscriptions avec un canton B et 4 cantons A
- 6 circonscriptions avec 2 cantons A, 2 cantons B et un canton de poids 0,33.

Avec cette répartition, le parti B remporte une majorité de circonscriptions sans pour autant remporter le plus grand nombre de voix. Cette répartition est représentée par le graphe ci-dessous :

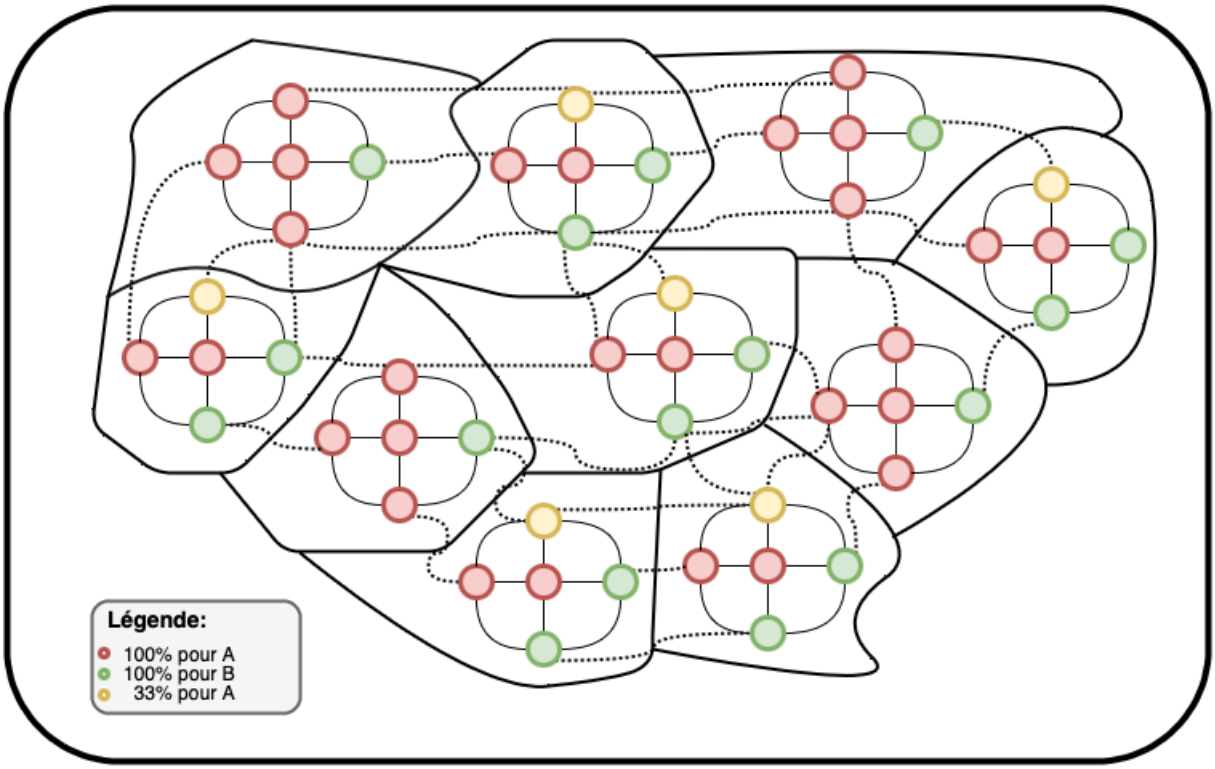

FIGURE 2.1.1. Graphe modélisant la situation

2.2. Modélisation du découpage

2.2.1. Problème d'optimisation.

DÉFINITION 3. Un problème d'optimisation combinatoire est défini par :

- un énoncé
- une fonction économique
- un sens.

2.2.1.1. *Énoncé:*

Dans notre cas, l'énoncé, c'est-à-dire les données, correspond au graphe $G = (X, U)$ décrit ci-dessus. Soit $X = \{x_1, x_2, \ldots, x_n\}$ l'ensemble des nœuds de notre graphe, tel que chaque x_i correspond à un canton. On lui associe les variables nbV_i, pV_i qui sont respectivement le **nombre de votants du canton** et le **pourcentage/proportion de votants qui soutiennent le parti A**.

. **Notre objectif est de définir un nombre k de circonscriptions tel que $\forall j \in \{1, \ldots, k\}$, K_j est un ensemble connexe de X, où un élément de X n'appartient qu'à un seul K_j.**

Plaçons nous dans un premier temps dans la situation où notre objectif est de **faire gagner le parti A** :

Nous pouvons commencer par définir la fonction p qui calcule le **poids du parti A dans une circonscription donnée, soit la proportion de votants pour le parti A dans cette circonscription.**

Le résultat $p(K_j)$ nous donne en fait le rapport du nombre de personnes votant pour le parti A dans la circonscription K_j sur le nombre total de votants dans cette même circonscription :

$$\begin{aligned} p :& X \to \mathbb{R}^+ \\ & K_j \mapsto p(K_j) \end{aligned}$$

$$p(K_j) = \frac{\sum_{i=1, x_i \in K_j}^{n} nbV_i \times pV_i}{\sum_{i=1, x_i \in K_j}^{n} nbV_i}$$

2.2.1.2. *La fonction économique.*

Ceci nous permet de définir la fonction économique f suivante :

$$\begin{aligned} f :& X \to \mathbb{R}^+ \\ & K_j \mapsto f(K_j) \end{aligned}$$

$$f(K_j) = \begin{cases} 1 & \text{si } p(K_j) \geq 0,5 \\ 0 & \text{sinon} \end{cases}$$

Ainsi la fonction f nous dit si c'est le parti A remporte la circonscription K_j ($f(K_j) = 1$) ou si c'est le parti B qui remporte la circonscription ($f(K_j) = 0$).

2.2.1.3. *Sens.* Si notre objectif est de faire gagner le plus de circonscriptions au parti A, on peut définir le problème suivant :

$$max \sum_{j=1}^{k} f(K_j)$$

Inversement, si notre objectif est de faire gagner le plus de circonscriptions au parti B, il suffit de poser :

$$min \sum_{j=1}^{k} f(K_j)$$

Remarquons que dans notre formulation, nous avons posé une contrainte de façon à ce que les circonscriptions soient d'un seul tenant mais nous n'avons pas imposé de contrainte de taille sur les circonscriptions.
Pour cela, nous pouvons nous inspirer des contraintes qui existent sur la taille des circonscriptions en France. Celles-ci sont un peu abstraites mais recommande des tailles de circonscriptions (en nombre de votants) significativement égales.

La règle que nous nous fixons alors est que le nombre de votants d'une circonscription doit être compris entre 80% **et** 120% **du nombre moyen de votants dans une circonscription**.

On peut ainsi définir la contrainte suivante pour un K_j donné :

$$\frac{0,8}{k}\sum_{i=1}^{n} nbV_i \leq \sum_{i=1, x_i \in K_j}^{n} nbV_i \leq \frac{1,20}{k}\sum_{i=1}^{n} nbV_i$$

Notons aussi que si notre objectif n'est pas de favoriser un camp ou un autre, alors l'utilisation de la fonction f n'a pas d'intérêt.
Il faudrait définir une nouvelle fonction économique tout en gardant la contrainte sur la taille que l'on vient d'énoncer.

2.2.2. Le choix de variables

. Afin de pouvoir formuler le problème sous forme d'un problème mathématique, nous avons besoin de définir les variables que nous allons utiliser.
Définissons simplement les variables binaires x_{ij} tel que pour le $i^{ème}$ canton et la $j^{ème}$ circonscription, on ait :

$$x_{ij} = \begin{cases} 1 & \text{si } x_i \in K_j \\ 0 & \text{sinon} \end{cases}$$

On peut remarquer que cette variable traduit l'appartenance d'un canton à une circonscription. Ce qui implique de devoir énoncer ces deux contraintes suivantes :

$$\forall i \in \{1, \ldots, n\}, \sum_{j=1}^{k} x_{ij} = 1$$
$$\forall j \in \{1, \ldots, k\}, \sum_{i=1}^{n} x_{ij} \geq 1$$

La première imposant à chaque canton d'appartenir à une et une seule circonscription et la seconde obligeant chaque circonscription à contenir au-moins un canton.
Cette dernière contrainte peut par ailleurs être modifiée pour forcer les circonscriptions à chacune contenir, un nombre de cantons que l'on pourra définir.

En explicitant les variables de la sorte, on peut alors redéfinir la fonction $p(K_j)$ précédente :

$$p(K_j) = \frac{\sum_{i=1}^{n} x_{ij} \times nbV_i \times pV_i}{\sum_{i=1}^{n} x_{ij} \times nbV_i}$$

Et ainsi reformuler notre problème mathématique sous la forme suivante :

$$\max(z) = \sum_{j=1}^{k} \lfloor 2 \left(\frac{\sum_{i=1}^{n} x_{ij} \times nbV_i \times pV_i}{\sum_{i=1}^{n} x_{ij} \times nbV_i} \right) \rfloor$$

$$\begin{cases} \sum_{j=1}^{k} x_{ij} = 1 & \forall i \in \{1, \ldots, n\} \\ \sum_{i=1}^{n} x_{ij} \geq 1 & \forall j \in \{1, \ldots, k\} \\ \sum_{i=1}^{n} x_{ij} \times nbV_i \leq \frac{1,20}{k} \sum_{i=1}^{n} nbV_i & \forall j \in \{1, \ldots, k\} \\ \sum_{i=1}^{n} x_{ij} \times nbV_i \geq \frac{0,8}{k} \sum_{i=1}^{n} nbV_i & \forall j \in \{1, \ldots, k\} \\ x_{ij} \in \{0, 1\} & \forall (i,j) \in \{1, \ldots, n\} \times \{1, \ldots, k\} \end{cases}$$

Afin d'appliquer des algorithmes rapides d'optimisation (par exemple ceux du logiciel CPLEX), il nous faut maintenant linéariser la formule du problème.

Pour cela, nous voulons prioritairement changer le *quotient* de la formule en *produit* (plus gérable pour la résolution de programmations linéaires) :

1ère étape :

Posons tout d'abord $z_j = \begin{cases} 1 & si \ \frac{\sum_{i=1}^{n} x_{ij} \times nbV_i \times pV_i}{\sum_{i=1}^{n} x_{ij} \times nbV_i} \geq 0.5 \\ 0 & si \ \frac{\sum_{i=1}^{n} x_{ij} \times nbV_i \times pV_i}{\sum_{i=1}^{n} x_{ij} \times nbV_i} < 0.5 \end{cases}$,

On peut remarquer que $\forall j \in \{1, \ldots, k\}, z_j \leq \frac{\sum_{i=1}^{n} x_{ij} \times nbV_i \times pV_i}{\sum_{i=1}^{n} x_{ij} \times nbV_i} + \frac{1}{2}$,

Multiplions maintenant notre inégalité par $\sum_{i=1}^{n} x_{ij} \times nbV_i$:

$$z_j \times \sum_{i=1}^{n} x_{ij} \times nbV_i \leq \sum_{i=1}^{n} x_{ij} \times nbV_i \times pV_i + \frac{1}{2} \sum_{i=1}^{n} x_{ij} \times nbV_i$$

Ainsi, nous obtenons une condition quadratique.

2ème étape :

La deuxième étape consiste à transformer notre condition quadratique en condition linéaire.
Pour cela, on introduit 3 nouvelles contraintes impliquant une nouvelle variable y_{ij} .
L'objectif de l'introduction de cette variable est d'obtenir le même comportement

que si on gardait la formule avec le produit des $z_j$avec les x_{ij} mais dans une formulation linéaire.

Posons alors $\begin{cases} y_{ij} \leq \frac{1}{2}(z_j + x_{ij}) \\ y_{ij} \geq z_j + x_{ij} - 1 \end{cases}$,

Ainsi, le problème devient :

$$\max(z) = \sum_{j=1}^{k} z_j$$

$$\begin{cases} \sum_{j=1}^{k} x_{ij} = 1 & \forall i \in \{1, \ldots, n\} \\ \sum_{i=1}^{n} x_{ij} \geq 1 & \forall j \in \{1, \ldots, k\} \\ \sum_{i=1}^{n} x_{ij} \times nbV_i \leq \frac{1,20}{k} \sum_{i=1}^{n} nbV_i & \forall j \in \{1, \ldots, k\} \\ \sum_{i=1}^{n} x_{ij} \times nbV_i \geq \frac{0,8}{k} \sum_{i=1}^{n} nbV_i & \forall j \in \{1, \ldots, k\} \\ x_{ij} \in \{0, 1\} & \forall (i,j) \in \{1, \ldots, n\} \times \{1, \ldots, k\} \\ z_j \in \{0, 1\} & \forall j \in \{1, \ldots, k\} \\ y_{ij} \in \{0, 1\} & \forall (i,j) \in \{1, \ldots, n\} \times \{1, \ldots, k\} \\ \sum_{i=1}^{n} y_{ij} \times nbV_i \leq \sum_{i=1}^{n} x_{ij} \times nbV_i \times pV_i + \frac{1}{2} \sum_{i=1}^{n} x_{ij} \times nbV_i & \forall (i,j) \in \{1, \ldots, n\} \times \{1, \ldots, k\} \\ y_{ij} \leq \frac{1}{2}(z_j + x_{ij}) & \forall (i,j) \in \{1, \ldots, n\} \times \{1, \ldots, k\} \\ y_{ij} \geq z_j + x_{ij} - 1 & \forall (i,j) \in \{1, \ldots, n\} \times \{1, \ldots, k\} \end{cases}$$

Finalement, on obtient une formulation mathématique de notre problème sous forme linéaire.

La dernière contrainte qu'il nous manque est celle de la connexité. C'est-à-dire que pour tout couple de sommets appartenant à une même classe, il existe un chemin pour aller de l'un à l'autre ne contenant que des sommets de leur classe d'appartenance.
Pour expliciter cette notion, posons

$$G_j = (K_j, U_j)$$

où G_j le sous-graphe représentant la circonscription j,
K_j l'ensemble des sommets d'une circonscription
U_j l'ensemble des arcs reliant les sommets de K_j

Nous voulons que $\forall j \in \{1, \ldots, k\}$ et $\forall (x_i, x_{i'}) \in K_j^2$ $x_i \longleftrightarrow x_{i'}$ dans G_j.

Pour commencer, on se contentera de calculer la distance entre 2 cantons (2 noeuds du graphe associé au problème)

$$x_{ij} \times x_{kj} \times d_{ik} \leq D$$

où D = distance maximale que l'on se fixe pour la résolution de notre problème
où d_{ik}= distance entre les sommets i et k du graphe

Les distances d_{ik}seront des données du problème.

CHAPTER 3

La classification des données

3.1. Intérêt

3.2. Classification non supervisée

3.2.1. Principe.

3.2.2. Application.

Part 2

PARTIE PRATIQUE

CHAPTER 4

Utilisation d'un solveur

Nous décidons d'utiliser CPLEX via l'API Python.
Afin de prendre en main ce solveur, nous allons tout d'abord essayer de résoudre le problème sur un *toy example.*
On décide de le construire nous-mêmes pour pouvoir vérifier les résultats en introduisant peu de valeurs et que ces valeurs soient simples.

On rappelle que l'on a $n\,cantons$ et $k\,circonscriptions$.
Les données nbV_i, pV_i appartenant au canton i respectivement le nombre de votants du canton et le pourcentage/proportion de votants qui soutiennent le parti A.

Ici, on choisit donc :

$$\begin{cases} 2\,circonscriptions & ==> k = 2 \\ 5\,cantons & ==> n = 5 \end{cases}$$

Pour les nbV_i, pV_i,on choisit arbitrairement :

$$\begin{cases} nbV_1 = 5 & pV_1 = 0.4 \\ nbV_2 = 6 & pV_2 = 0.6 \\ nbV_3 = 5 & pV_3 = 0.7 \\ nbV_4 = 6 & pV_4 = 0.2 \\ nbV_5 = 7 & pV_5 = 0.9 \end{cases}$$

On a donc :

$$\max(z) = z_1 + z_2$$

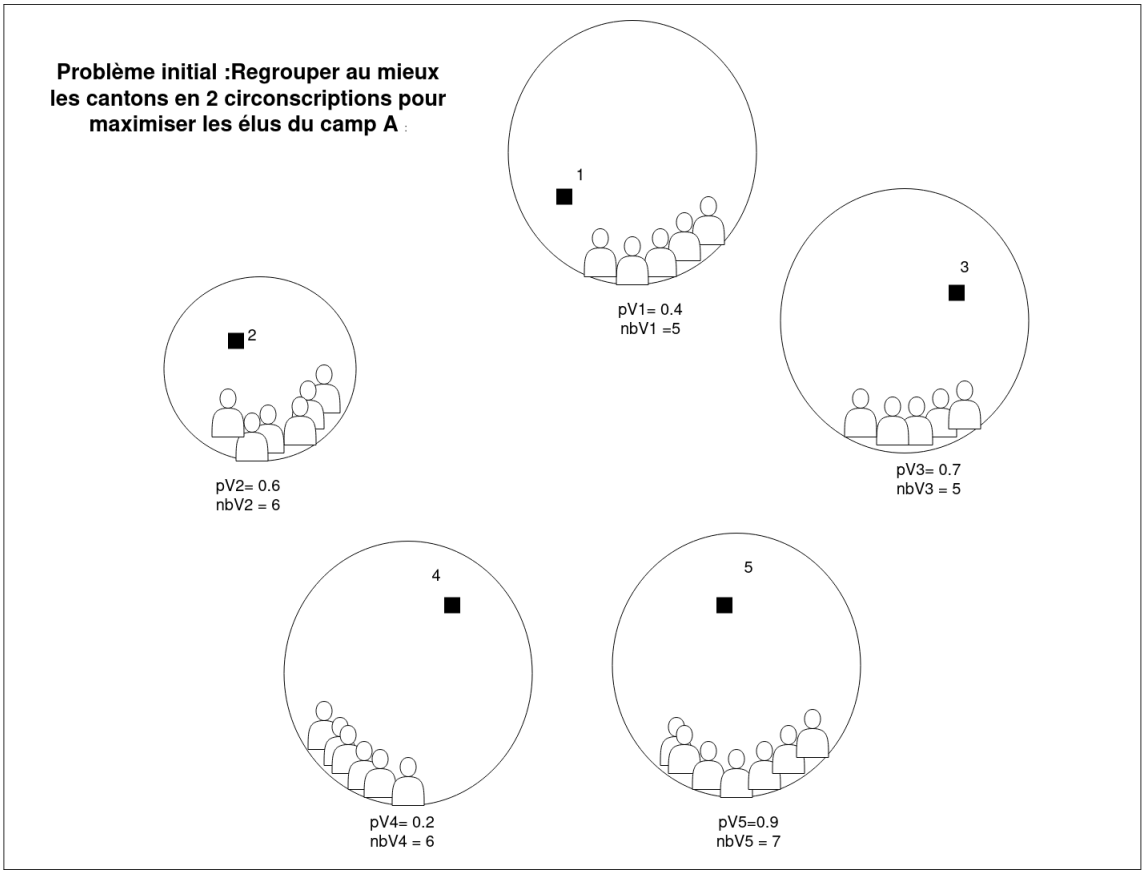

FIGURE 4.0.1. Situation initiale à résoudre

$$
\begin{cases}
x_{i1} + x_{i2} = 1 & \forall i \in \{1, \ldots, 5\} \\
x_{1j} + x_{2j} + x_{3j} + x_{4j} + x_{5j} \geq 1 & \forall j \in \{1, \ldots, 2\} \\
x_{1j}nbV_1 + x_{2j}nbV_2 + x_{3j}nbV_3 + x_{4j}nbV_4 + x_{5j}nbV_5 \ \leq \ \frac{1,20}{2} \sum_{i=1}^{n} nbV_i & \forall j \in \{1, \ldots, 2\} \\
x_{1j}nbV_1 + x_{2j}nbV_2 + x_{3j}nbV_3 + x_{4j}nbV_4 + x_{5j}nbV_5 \ \geq \ \frac{0,8}{2} \sum_{i=1}^{n} nbV_i & \forall j \in \{1, \ldots, 2\} \\
x_{ij} \in \{0, 1\} & \forall (i, j) \in \{1, \ldots, n\} \times \{1, \ldots, 2\} \\
z_j \in \{0, 1\} & \forall j \in \{1, \ldots, 2\} \\
y_{ij} \in \{0, 1\} & \forall (i, j) \in \{1, \ldots, 5\} \times \{1, \ldots, 2\} \\
\sum_{i=1}^{5} y_{ij} \times nbV_i \leq \sum_{i=1}^{5} x_{ij} \times nbV_i \times pV_i + \frac{1}{2} \sum_{i=1}^{5} x_{ij} \times nbV_i & \forall (i, j) \in \{1, \ldots, 5\} \times \{1, \ldots, 2\} \\
y_{ij} \leq \frac{1}{2}(z_j + x_{ij}) & \forall (i, j) \in \{1, \ldots, 5\} \times \{1, \ldots, 2\} \\
y_{ij} \geq z_j + x_{ij} - 1 & \forall (i, j) \in \{1, \ldots, 5\} \times \{1, \ldots, 2\}
\end{cases}
$$

L'objectif est à présent de mettre ce programme linéaire sous la forme d'un problème résoluble en machine.

CHAPTER 5

Algorithme de classification non supervisée