Chapitre 1

Modélisation

1. **Introduction**
2. **Modélisation à l'aide de variables entières**
3. **Formulations classiques**

1.1 Introduction

Un programme mathématique (PM) est de la forme

$$\begin{array}{rl} \min\limits_{\boldsymbol{x}} & \boldsymbol{f}(\boldsymbol{x}) \\ \textbf{sujet à :} & \boldsymbol{x} \in \boldsymbol{X} \end{array}$$

où

$\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$ **est le vecteur des variables de décision**

$\boldsymbol{X} \subseteq \mathbb{R}^{\boldsymbol{N}}$ **est le domaine réalisable**

$\boldsymbol{f} : \boldsymbol{X} \to \mathbb{R}$ **est la fonction objectif.**

Il n'existe pas d'algorithme capable de résoudre un problème aussi général. Il existe des algorithmes spécialisés applicables à des classes de programmes mathématiques définies par les propriétés de $\boldsymbol{f}$ et $\boldsymbol{X}$.

Exemples :

- **Algorithme du simplexe pour les programmes linéaires**
- **Algorithme du gradient pour les programmes non linéaires tels que $\boldsymbol{X} = \mathbb{R}^n$ et $\boldsymbol{f}$ est convexe et différentiable**

Souvent la modélisation d'un problème requiert des variables qui ne peuvent prendre que des valeurs entières (appelées *variables entières*) pour déterminer un nombre de véhicules requis pour accomplir des tâches, un ordre dans lequel une série d'opérations doit être effectuée, un nombre d'unités d'un produit à fabriquer, etc. Dans bien des cas, l'utilisation de variables continues est inacceptable.

Dans ce cours, nous étudierons des programmes linéaires en nombres entiers

$$
(\boldsymbol{PNE}) \qquad \begin{aligned} \min_{\boldsymbol{x}} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{sujet à :} \quad & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \in \mathbb{N}^{\boldsymbol{n}} \end{aligned}
$$

avec $\boldsymbol{c} \in \mathbb{R}^{\boldsymbol{n}}$, $\boldsymbol{A} \in \mathbb{R}^{\boldsymbol{m}\times\boldsymbol{n}}$, $\boldsymbol{b} \in \mathbb{R}^{\boldsymbol{m}}$ **et** $\mathbb{N} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots\}$,

ou des programmes linéaires mixtes

$$
(\boldsymbol{PMX}) \qquad \begin{aligned} \min_{\boldsymbol{x},\boldsymbol{y}} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{h}^{\mathrm{T}}\boldsymbol{y} \\ \text{sujet à :} \quad & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{G}\boldsymbol{y} \leq \boldsymbol{b} \\ & \boldsymbol{x} \in \mathbb{N}^{\boldsymbol{n}},\ \boldsymbol{y} \geq \mathbf{0} \end{aligned}
$$

avec $\boldsymbol{h}, \boldsymbol{y} \in \mathbb{R}^{\boldsymbol{p}}$ **et** $\boldsymbol{G} \in \mathbb{R}^{\boldsymbol{m}\times\boldsymbol{p}}$.

Souvent les variables sont restreintes à prendre des valeurs dans $\{0, 1\}$ (appelées *variables binaires*), ce qui engendre les problèmes suivants :

$$
\begin{array}{llrl}
(\boldsymbol{PNE_{0-1}}) & & \min\limits_{\boldsymbol{x}} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\
& & \text{sujet à :} & \boldsymbol{Ax} \leq \boldsymbol{b} \\
& & & \boldsymbol{x} \in \mathbb{B}^n = \{0, 1\}^n
\end{array}
$$

$$
\begin{array}{llrl}
(\boldsymbol{PMX_{0-1}}) & & \min\limits_{\boldsymbol{x},\boldsymbol{y}} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{h}^{\mathrm{T}}\boldsymbol{y} \\
& & \text{sujet à :} & \boldsymbol{Ax} + \boldsymbol{Gy} \leq \boldsymbol{b} \\
& & & \boldsymbol{x} \in \mathbb{B}^n,\ \boldsymbol{y} \geq 0.
\end{array}
$$

On obtient les relations suivantes entre les classes de problèmes :

$$\boldsymbol{PNE_{0-1}} \subset \boldsymbol{PNE} \subset \boldsymbol{PMX} \subset \boldsymbol{PM}$$

et

$$\boldsymbol{PNE_{0-1}} \subset \boldsymbol{PMX_{0-1}} \subset \boldsymbol{PMX} \subset \boldsymbol{PM}$$

Nous étudierons principalement les classes $\boldsymbol{PNE}$ et $\boldsymbol{PNE_{0-1}}$ puisqu'il est facile d'étendre les résultats à $\boldsymbol{PMX}$ et $\boldsymbol{PMX_{0-1}}$.

Hypothèse de base : Tous les coefficients de c et A dans PNE sont entiers.

Sinon il peut ne pas exister de solution optimale comme dans l'exemple suivant

$$\begin{array}{rl}\min\limits_{x,y} & \sqrt{2}x - y \\ \text{sujet à :} & \sqrt{2}x - y \geq 0 \\ & x \geq 1 \\ & x, y \in \mathbb{N}\end{array}$$

Preuve : Supposons qu'il existe une solution optimale (x^*, y^*) de valeur z^*. Observons d'abord que

$$0 < z^* \leq \sqrt{2} - 1 < 1.$$

En effet, $z^* \geq 0$ car $\sqrt{2}x - y \geq 0$ pour toute solution réalisable. De plus, si $z^* = 0$, alors $\sqrt{2} = \frac{y^*}{x^*}$ ce qui est impossible pour $x^*, y^* \in \mathbb{N}$. Aussi, puisque $x = y = 1$ est une solution réalisable, alors $z^* \leq \sqrt{2} - 1$. Par conséquent, $0 < z^* < 1$, $x^* \geq 1$ et $z^* = \sqrt{2}x^* - y^*$ est irrationnel.

Il existe donc un entier $n \geq 1$ tel que

$$\begin{aligned} & \frac{x^*}{n+1} < z^* = \sqrt{2}x^* - y^* < \frac{x^*}{n} \\ \Leftrightarrow \quad & \frac{x^*}{n+1} + y^* < \sqrt{2}x^* < \frac{x^*}{n} + y^* \\ \Leftrightarrow \quad & \frac{1}{n+1} + \frac{y^*}{x^*} < \sqrt{2} < \frac{1}{n} + \frac{y^*}{x^*}. \end{aligned}$$

Montrons que $(\bar{x}, \bar{y}) = ((n+1)x^*, (n+1)y^* + x^*)$ est réalisable et de coût inférieur à z^*.

$(\bar{x}, \bar{y})$ est réalisable car

$$\begin{aligned} \sqrt{2}\bar{x} - \bar{y} &= \sqrt{2}(n+1)x^* - (n+1)y^* - x^* \\ &> \left(\frac{1}{n+1} + \frac{y^*}{x^*}\right)(n+1)x^* - (n+1)y^* - x^* = 0 \end{aligned}$$

$\bar{x} = (n+1)x^* \geq x^* \geq 1$

$\bar{x}$ et $\bar{y}$ sont des entiers non négatifs.

La valeur $\bar{z}$ de $(\bar{x}, \bar{y})$ est

$$\begin{aligned} \bar{z} = \sqrt{2}\bar{x} - \bar{y} &= \sqrt{2}(n+1)x^* - (n+1)y^* - x^* \\ &= (n+1)z^* - x^* = z^* + nz^* - x^* \\ &< z^* + n\left(\frac{x^*}{n}\right) - x^* = z^* \end{aligned}$$

D'où il y a contradiction. $\square$

1.2 Modélisation à l'aide de variables entières

1.2.1. Valeurs discrètes

Voyons trois façons de modéliser $x \in \{1.5, 2, 2.5\}$.

i) avec une variable entière :

$$y \in \mathbb{N}, \qquad 3 \leq y \leq 5, \qquad 2x - y = 0.$$

ii) avec 3 variables binaires :

$$\begin{aligned} x - 1.5y_1 - 2y_2 - 2.5y_3 &= 0 \\ y_1 + y_2 + y_3 &= 1 \\ y_1, y_2, y_3 &\in \{0, 1\}. \end{aligned}$$

iii) avec 2 variables binaires :

$$\begin{aligned} x - 0.5y_2 - y_3 &= 1.5 \\ y_2 + y_3 &\leq 1 \\ y_2, y_3 &\in \{0, 1\}. \end{aligned}$$

1.2.2. Expressions logiques

Considérons n options $o_1, o_2, \ldots, o_n$. Associons une variable binaire x_i à chaque option o_i qui vaut 1 si l'option est choisie et 0 sinon.

Comment modélise-t-on les conditions suivantes ?

i) Si o_1 et o_2 sont choisies, alors o_3 ne peut l'être.

$$\begin{aligned} x_1 + x_2 + x_3 &\leq 2 \\ x_1, x_2, x_3 &\in \{0, 1\}. \end{aligned}$$

ii) Si o_1 ou o_2 est choisie, alors o_3 doit l'être aussi.

$$\begin{aligned} x_1 + x_2 &\leq 2x_3 \\ x_1, x_2, x_3 &\in \{0, 1\}. \end{aligned}$$

ou

$$\begin{aligned} x_1 &\leq x_3 \\ x_2 &\leq x_3 \\ x_1, x_2, x_3 &\in \{0, 1\}. \end{aligned}$$

1.2.3. Contraintes disjonctives

Soit d_1 et d_2 les durées de deux tâches qui doivent être réalisées sur la même machine. Comment modéliser la contrainte qui interdit de faire les tâches en même temps ?

t_1, t_2 : variables indiquant l'heure de début des tâches.

y_{12} : variable binaire indiquant si la tâche 1 débute avant la tâche 2 ($= 1$ si oui).

M : constante suffisamment grande.

$$\begin{aligned} t_1 + d_1 &\leq t_2 + M(1 - y_{12}) \\ t_2 + d_2 &\leq t_1 + M y_{12} \\ y_{12} &\in \{0, 1\}. \end{aligned}$$

Comment modéliser que $x \in \mathbb{R}^n$ doit appartenir à au moins k des q polytopes

$$P^i = \{x \in \mathbb{R}^n \ : \ A^i x \leq b^i\}, \qquad i = 1, \ldots, q \ ?$$

(A^i et b^i sont de dimensions $m_i \times n$ et $m_i \times 1$, resp.)

Comme les P^i sont des polytopes, il existe $u \in \mathbb{R}^n_+$ tel que, si $x \in \bigcup_{i=1}^{q} P^i$, alors $-u \leq x \leq u$.

Pour tout $i = 1, 2, \ldots, q$, il existe $w^i \in \mathbb{R}^{m_i}$ tel que $A^i x \leq b^i + w^i$, $\forall x \in [-u, u]$.

Associons une variable binaire y_i à chaque polytope qui indique si $x \in P^i$ ou non.

Le modèle est :

$$\begin{array}{rcll} A^i x & \leq & b^i + w^i(1 - y_i) & \forall i = 1, 2, \ldots, q \\ \sum_{i=1}^{q} y_i & \geq & k & \\ y_i & \in & \{0, 1\} & \forall i = 1, 2, \ldots, q. \end{array}$$

Note : y_i peut être égale à 0 même si $x \in P^i$.

1.2.4 Coûts fixes

Considérons la fonction de coût

$$h(x) = \begin{cases} 0 & \text{si } x = 0 \\ f + px & \text{si } x > 0 \end{cases}$$

où f et p sont deux constantes positives.

Soit y une variable binaire qui vaut 1 si $x > 0$ et 0 si $x = 0$. Soit M une constante suffisamment grande.

Alors :

$$\begin{aligned} h(x) &= fy + px \\ x &\leq My \\ y &\in \{0, 1\}. \end{aligned}$$

Remarque : $x = 0 \not\Rightarrow y = 0$. En général, on minimise les coûts ce qui forcera $y = 0$ dans ce cas.

1.2.5 Exemples sur les échecs

Quel est le plus grand nombre de fous que l'on peut placer sur un échiquier (8×8) de sorte qu'aucun fou ne soit menacé par un autre ?

Variables :

$$x_{ij} = \begin{cases} 1 & \text{s'il y a un fou sur la case } (i, j) \\ 0 & \text{sinon} \end{cases}$$

Formulation PNE :

$$z = \max_{x} \qquad \sum_{i=1}^{8} \sum_{j=1}^{8} x_{ij} \tag{1.1}$$

sujet à :

$$\sum_{i+j=p} x_{ij} \leq 1, \quad \forall p \in \{2, 3, \ldots, 16\} \tag{1.2}$$

$$\sum_{i-j=q} x_{ij} \leq 1, \quad \forall q \in \{-7, -6, \ldots, 7\} \tag{1.3}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j = 1, \ldots, 8 \tag{1.4}$$

Il est souvent utile de trouver des bornes inférieure ($\underline{z}$) et supérieure ($\bar{z}$) sur la valeur optimale z.

Pour une maximisation, une borne inférieure est donnée par une solution réalisable.

Un fou par case de la ligne $i = 1$ implique $\underline{z} = 8$. On peut faire mieux en ajoutant un fou par case $j = 2, \ldots, 7$ de la ligne $i = 8$ pour obtenir $\underline{z} = 14$.

Le nombre de diagonales sur l'échiquier donne une borne supérieure $\bar{z} = 15$.

D'où $14 \leq z \leq 15$.

Comme il ne peut y avoir un fou sur la première et la dernière diagonale en même temps, on peut revoir la borne supérieure : $\bar{z} = 14$, ce qui indique que la valeur optimale est $z = 14$.

Quel est le plus petit nombre de fous requis pour occuper ou menacer toutes les cases ?

Formulation PNE :

$$z = \min_{x} \qquad \sum_{i=1}^{8} \sum_{j=1}^{8} x_{ij} \tag{1.5}$$

sujet à :

$$\sum_{\substack{(k,\ell) \text{ tels que} \\ k+\ell=i+j \text{ ou } k-\ell=i-j}} x_{k\ell} \geq 1, \quad \forall i,j = 1,\ldots,8 \tag{1.6}$$

$$x_{ij} \in \{0,1\}, \quad \forall i,j = 1,\ldots,8 \tag{1.7}$$

Borne supérieure (solution réalisable (3,3), (3,4), (3,6), (4,6), (5,3), (6,3), (6,5), (6,6)) : $\bar{z} = 8$.

Borne inférieure : un fou couvre au plus 14 cases (en (4,4)), ce qui implique $\underline{z} = \left\lceil \frac{64}{14} \right\rceil = 5$.

On peut resserrer cette borne en observant que le problème est décomposable en cases noires et en cases blanches : $\underline{z} = 2 \left\lceil \frac{32}{14} \right\rceil = 6$.

D'où $6 \leq z \leq 8$.

1.3 Formulations classiques

1.3.1. Problème de sac à dos binaire

b : volume d'un sac à dos

m items pouvant être apportés

a_j : volume de l'item j

c_j : utilité de l'item j

Quels items doivent être apportés pour maximiser l'utilité ?

Variables :

$$x_j = \begin{cases} 1 & \text{si l'item } j \text{ est choisi} \\ 0 & \text{sinon} \end{cases}$$

Formulation PNE :

$$\max_{x} \qquad \sum_{j=1}^{m} c_j x_j \tag{1.8}$$

sujet à :

$$\sum_{j=1}^{m} a_j x_j \leq b \tag{1.9}$$

$$x \in \mathbb{B}^m \tag{1.10}$$

Note : On peut supposer que $a_j > 0$ et $c_j > 0$. En effet,

- **Si $a_j \geq 0$ et $c_j \leq 0$, on peut poser $x_j = 0$.**
- **Si $a_j \leq 0$ et $c_j \geq 0$, on peut poser $x_j = 1$.**
- **Si $a_j < 0$ et $c_j < 0$, on peut poser $x'_j = 1 - x_j$, $a'_j = -a_j > 0$, $c'_j = -c_j > 0$ et $b' = b - a_j$ pour retomber sur un cas équivalent avec volume et utilité positives.**

Variantes :

entier et borné : **certains items peuvent être choisis plus qu'une fois.**

Multidimensionnel : **en considérant une contrainte de type (1.9) pour chaque dimension.**

Sacs à dos multiples : **les items peuvent être répartis dans plusieurs sacs à dos.**

1.3.2. Problème d'affectation

n tâches à réaliser

n employés pour les réaliser (une tâche par employé)

c_{ij} : rendement si l'employé i fait la tâche j

Il faut déterminer quel employé affecter à chaque tâche de façon à maximiser le rendement total.

Variables : $x_{ij} = \begin{cases} 1 & \text{si l'employé } i \text{ est affecté à la tâche } j \\ 0 & \text{sinon} \end{cases}$

Formulation PNE :

$$\max_x \quad \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{1.11}$$

sujet à :

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i \in \{1,\dots,n\} \tag{1.12}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \in \{1,\dots,n\} \tag{1.13}$$

$$x \in \mathbb{B}^{n\times n} \tag{1.14}$$

Généralisation : plus de tâches que d'employés, durée pour chaque tâche, temps disponible par employé.

1.3.3 Problème de couplage parfait

$2n$ personnes

n équipes de 2 personnes à former

c_{ij} : rendement d'une équipe composée des personnes i et j

Il faut construire les n équipes de façon à maximiser le rendement total. Chaque personne doit faire partie d'une équipe.

Variables :

$$x_{ij} = \begin{cases} 1 & \text{si } i \text{ et } j \text{ forment une équipe } (i < j) \\ 0 & \text{sinon} \end{cases}$$

Formulation PNE :

$$\max_{x} \qquad \sum_{i=1}^{2n-1} \sum_{j=i+1}^{2n} c_{ij} x_{ij} \tag{1.15}$$

sujet à :

$$\sum_{i<\ell} x_{i\ell} + \sum_{\ell<j} x_{\ell j} = 1, \quad \forall \ell \in \{1, \ldots, 2n\} \tag{1.16}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j = 1, \ldots, 2n, \ \ i < j \tag{1.17}$$

Variante : Problème de couplage en remplaçant = par ≤ dans (1.16).

1.3.4 Recouvrement d'ensemble

$M = \{1, 2, \ldots, m\}$: ensemble de régions à couvrir par des antennes

$N = \{1, 2, \ldots, n\}$: ensemble de sites potentiels pour ériger les antennes

$S_j \subseteq M$: sous-ensemble des régions couvertes par une antenne sur le site j

$$a_{ij} = \begin{cases} 1 & \text{si la région } i \in S_j \\ 0 & \text{sinon} \end{cases}$$

c_j : coût de construction d'une antenne sur le site j

Il faut déterminer sur quels sites ériger les antennes pour couvrir toutes les régions à coût minimal.

Variables :

$$x_j = \begin{cases} 1 & \text{si une antenne est érigée sur le site } j \\ 0 & \text{sinon} \end{cases}$$

Formulation PNE :

$$\min_{\boldsymbol{x}} \qquad \sum_{j \in N} c_j x_j \tag{1.18}$$

sujet à :

$$\sum_{j \in N} a_{ij} x_j \geq 1, \quad \forall i \in M \tag{1.19}$$

$$x \in \mathbb{B}^n \tag{1.20}$$

Variantes :

Partitionnement d'ensemble : en remplaçant $\geq$ par $=$ dans (1.19).

Empaquetage d'ensemble : en remplaçant $\geq$ par $\leq$ dans (1.19) et min par max dans (1.18) (c_j : profit).

1.3.5 Localisation d'entrepôts sans capacité

$M = \{1, 2, \ldots, m\}$: ensemble des clients

$N = \{1, 2, \ldots, n\}$: ensemble des sites pour construire un entrepôt

f_j : coût fixe pour construire sur le site j

c_{ij} : coût pour desservir le client i à partir du site j

Il faut déterminer sur quels sites construire un entrepôt et quel entrepôt doit desservir chaque client de façon à minimiser les coûts totaux. Chaque client doit être affecté à un seul entrepôt.

Variables :

$$y_j = \begin{cases} 1 & \text{si un entrepôt est construit sur le site } j \\ 0 & \text{sinon} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{si entrepôt } j \text{ dessert client } i \\ 0 & \text{sinon} \end{cases}$$

Formulation PNE :

$$\min_{\boldsymbol{x},\boldsymbol{y}} \qquad \sum_{i \in M} \sum_{j \in N} c_{ij} x_{ij} + \sum_{j \in N} f_j y_j \tag{1.21}$$

sujet à :

$$\sum_{j \in N} x_{ij} = 1, \quad \forall i \in M \tag{1.22}$$

$$\sum_{i \in M} x_{ij} \leq m y_j, \quad \forall j \in N \tag{1.23}$$

$$x_{ij}, y_j \in \{0,1\}, \quad \forall i \in M, j \in N \tag{1.24}$$

Variante : avec capacité q_j à l'entrepôt j et demande a_i pour client i.

On remplace alors les contraintes (1.23) par

$$\sum_{i \in M} a_i x_{ij} \leq q_j y_j, \qquad \forall j \in N \tag{1.25}$$

1.3.6 Taille des lots sans capacité

n périodes

un seul produit

d_t : demande à la période t

f_t : coût fixe de production à la période t

p_t : coût unitaire de production à la période t

h_t : coût d'entreposage du stock à la fin de la période t

Il faut minimiser les coûts totaux pour satisfaire la demande.

Variables :

x_t : quantité produite à la période t

s_t : quantité en stock à la fin de la période t

$$y_t = \begin{cases} 1 & \text{s'il y a production à la période } t \\ 0 & \text{sinon} \end{cases}$$

Formulation PNE :

$$\min_{x,y,s} \quad \sum_{t=1}^{n} \left(f_t y_t + p_t x_t + h_t s_t \right) \tag{1.26}$$

sujet à :

$$s_{t-1} + x_t = d_t + s_t, \quad \forall t \in \{1, 2, \ldots, n\} \tag{1.27}$$

$$x_t \leq M_t y_t, \quad \forall t \in \{1, 2, \ldots, n\} \tag{1.28}$$

$$s_0 = 0, \tag{1.29}$$

$$x_t, s_t \geq 0, \quad \forall t \in \{1, 2, \ldots, n\} \tag{1.30}$$

$$y_t \in \{0, 1\}, \quad \forall t \in \{1, 2, \ldots, n\} \tag{1.31}$$

En fixant $s_n = 0$, on peut utiliser $M_t = \sum_{i=t}^{n} d_i$.

Variantes : avec capacité de production et capacité d'entreposage

1.3.7 Commis voyageur asymétrique

n villes à visiter $\qquad$ $N = \{1, \ldots, n\}$

un seul commis qui doit revenir à la ville de départ

c_{ij} : distance de la ville i à la ville j

Il faut trouver un circuit qui visite toutes les villes et minimise la distance totale.

Variables :

$$x_{ij} = \begin{cases} 1 & \text{si la ville } j \text{ suit la ville } i \text{ dans le circuit} \\ 0 & \text{sinon} \end{cases}$$

Formulation PNE :

$$\min_{\boldsymbol{x}} \qquad \sum_{i \in N} \sum_{j \in N, j \neq i} \boldsymbol{c_{ij} x_{ij}} \tag{1.32}$$

sujet à :

$$\sum_{j \in N, j \neq i} \boldsymbol{x_{ij} = 1}, \quad \forall \boldsymbol{i \in N} \tag{1.33}$$

$$\sum_{i \in N, i \neq j} \boldsymbol{x_{ij} = 1}, \quad \forall \boldsymbol{j \in N} \tag{1.34}$$

$$\sum_{i \in S} \sum_{j \notin S} \boldsymbol{x_{ij} \geq 1}, \quad \forall \boldsymbol{S \subset N, S \neq \emptyset} \tag{1.35}$$

$$\boldsymbol{x_{ij} \in \{0, 1\}}, \quad \forall \boldsymbol{i, j \in N, i \neq j} \tag{1.36}$$

Les contraintes (1.35) d'élimination de sous-tours peuvent être remplacées par

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j \neq i}} \boldsymbol{x_{ij} \leq |S| - 1}, \qquad \forall \boldsymbol{S \subset N, 2 \leq |S| \leq n - 1.} \tag{1.37}$$

Variante : symétrique ($\boldsymbol{c_{ij} = c_{ji}, \forall i, j}$)

1.3.8 Tournées de véhicules

n clients à visiter avec demande d_i $\qquad$ $N = \{1, \ldots, n\}$

K : ensemble de véhicules identiques disponibles

Q : capacité d'un véhicule

o : dépôt d'où partent et où reviennent les véhicules

N^o : $N \cup \{o\}$

c_{ij} : distance du lieu i au lieu j (lieu = client ou dépôt)

Il faut trouver les routes des véhicules de distance totale minimale afin de visiter tous les clients une fois tout en respectant la capacité des véhicules.

Variables :

$$x_{ij}^k = \begin{cases} 1 & \text{si le véhicule } k \text{ visite le lieu } j \text{ immédiatement} \\ & \text{après le lieu } i \\ 0 & \text{sinon} \end{cases}$$

Formulation PNE :

$$\min_{x} \quad \sum_{k\in K}\sum_{i\in N^o}\sum_{j\in N^o, j\neq i} c_{ij}x_{ij}^k \tag{1.38}$$

sujet à :

$$\sum_{k\in K}\sum_{j\in N^o, j\neq i} x_{ij}^k = 1, \quad \forall i \in N \tag{1.39}$$

$$\sum_{j\in N} x_{oj}^k \leq 1, \quad \forall k \in K \tag{1.40}$$

$$\sum_{j\in N^o, j\neq i} x_{ij}^k - \sum_{j\in N^o, j\neq i} x_{ji}^k = 0, \quad \forall k \in K, i \in N \tag{1.41}$$

$$\sum_{i\in N}\sum_{j\in N^o, j\neq i} d_i x_{ij}^k \leq Q, \quad \forall k \in K \tag{1.42}$$

$$\sum_{k\in K}\sum_{i\in S}\sum_{j\in S} x_{ij}^k \leq |S| - 1, \quad \forall S \subseteq N, \; 2 \leq |S| \leq n \tag{1.43}$$

$$x_{ij}^k \in \{0,1\}, \quad \forall k \in K, i,j \in N^o, \; i \neq j. \tag{1.44}$$

Remarque : Beaucoup de contraintes d'élimination de sous-tours (1.43) et de symétrie car les véhicules sont identiques.

Peut-on renforcir les contraintes (1.43) et éliminer la symétrie ?

$q(S)$: nombre minimum de véhicules requis pour desservir les clients de $S \subseteq N$

Variables :

$$x_{ij} = \begin{cases} 1 & \text{si un véhicule visite le lieu } j \text{ immédiatement} \\ & \text{après le lieu } i \\ 0 & \text{sinon} \end{cases}$$

Nouvelle formulation PNE :

$$\min_{\boldsymbol{x}} \quad \sum_{i\in N^o} \sum_{j\in N^o, j\neq i} c_{ij} x_{ij} \tag{1.45}$$

sujet à :

$$\sum_{j\in N^o, j\neq i} x_{ij} = 1, \quad \forall i \in N \tag{1.46}$$

$$\sum_{j\in N} x_{oj} \leq |K| \tag{1.47}$$

$$\sum_{j\in N^o, j\neq i} x_{ij} - \sum_{j\in N^o, j\neq i} x_{ji} = 0, \quad \forall i \in N \tag{1.48}$$

$$\sum_{i\in S} \sum_{j\in S} x_{ij} \leq |S| - q(S), \quad \forall S \subseteq N, \; 2 \leq |S| \leq n \tag{1.49}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in N^o, \; i \neq j. \tag{1.50}$$

Remarque : Pas de symétrie mais beaucoup de contraintes d'élimination de sous-tours et de capacité (1.49).

Autre formulation PNE :

$\boldsymbol{R}$ **: ensemble des routes réalisables**

$\boldsymbol{c_r}$ **: coût de la route** $\boldsymbol{r}$

$$\boldsymbol{a_{ir}} = \begin{cases} \mathbf{1} & \textbf{si la route } \boldsymbol{r} \textbf{ visite le client } \boldsymbol{i} \\ \mathbf{0} & \textbf{sinon} \end{cases}$$

Variables :

$$\boldsymbol{y_r} = \begin{cases} \mathbf{1} & \textbf{si la route } \boldsymbol{r} \textbf{ est choisie} \\ \mathbf{0} & \textbf{sinon} \end{cases}$$

$$\min_{\boldsymbol{y}} \quad \sum_{\boldsymbol{r \in R}} \boldsymbol{c_r y_r} \tag{1.51}$$

sujet à :

$$\sum_{\boldsymbol{r \in R}} \boldsymbol{a_{ir} y_r} = \mathbf{1}, \quad \forall \boldsymbol{i} \in \boldsymbol{N} \tag{1.52}$$

$$\sum_{\boldsymbol{r \in R}} \boldsymbol{y_r} \leq |\boldsymbol{K}|, \tag{1.53}$$

$$\boldsymbol{y_r} \in \{\mathbf{0}, \mathbf{1}\}, \quad \forall \boldsymbol{r} \in \boldsymbol{R}. \tag{1.54}$$

Remarques :

- **modèle de partitionnement d'ensemble**
- **pas de symétrie**
- **très grand nombre de variables**

Généralisations :

- **cas symétrique**
- **plusieurs dépôts**
- **avec fenêtres de temps et durées de déplacement**
- **plusieurs visites possibles par client**