La programmation quadratique en variables entières

Amélie Lambert

Cnam

ECE 2016-2017

1. Les formes quadratiques
2. Exemple de modélisation par de la programmation quadratique
3. Résolution des programmes quadratiques

Les formes quadratiques

Définition 1 : On appelle *forme quadratique* une expression de la forme

$$f(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij} x_i x_j$$

avex $x = (x_1, x_2, \ldots, x_n)$ un vecteur de variables réelles.

Les formes quadratiques

Définition 1 : On appelle *forme quadratique* une expression de la forme

$$f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j$$

avex $x = (x_1, x_2, \ldots, x_n)$ un vecteur de variables réelles.

Notation : Une forme quadratique peut s'écrire sous la forme matricielle, et on a :

$$f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j = x^T Q x$$

On dit que *la matrice Q représente la forme quadratique $f(x)$.*

Les formes quadratiques

Remarque : Pour une même forme quadratique $f(x)$, il peut exister plusieurs matrices Q qui la représentent.

Les formes quadratiques

Remarque : Pour une même forme quadratique $f(x)$, il peut exister plusieurs matrices Q qui la représentent.

Exemple 1 : La forme quadratique $x_1^2 + 4x_1x_2 + 3x_2^2$ peut s'écrire par exemple de plusieurs façons :

Les formes quadratiques

Remarque : Pour une même forme quadratique $f(x)$, il peut exister plusieurs matrices Q qui la représentent.

Exemple 1 : La forme quadratique $x_1^2 + 4x_1x_2 + 3x_2^2$ peut s'écrire par exemple de plusieurs façons :

- $x_1^2 + 4x_1x_2 + 3x_2^2 = x_1^2 + 2x_1x_2 + 2x_2x_1 + 3x_2^2$, sa représentation matricielle est alors : $Q_1 = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$

Les formes quadratiques

Remarque : Pour une même forme quadratique $f(x)$, il peut exister plusieurs matrices Q qui la représentent.

Exemple 1 : La forme quadratique $x_1^2 + 4x_1x_2 + 3x_2^2$ peut s'écrire par exemple de plusieurs façons :

- $x_1^2 + 4x_1x_2 + 3x_2^2 = x_1^2 + 2x_1x_2 + 2x_2x_1 + 3x_2^2$, sa représentation matricielle est alors : $Q_1 = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$
- $x_1^2 + 4x_1x_2 + 3x_2^2 = x_1^2 + 4x_1x_2 + 0x_2x_1 + 3x_2^2$, sa représentation matricielle est alors : $Q_2 = \begin{pmatrix} 1 & 4 \\ 0 & 3 \end{pmatrix}$

Les formes quadratiques

Remarque : Pour une même forme quadratique $f(x)$, il peut exister plusieurs matrices Q qui la représentent.

Exemple 1 : La forme quadratique $x_1^2 + 4x_1x_2 + 3x_2^2$ peut s'écrire par exemple de plusieurs façons :

- $x_1^2 + 4x_1x_2 + 3x_2^2 = x_1^2 + 2x_1x_2 + 2x_2x_1 + 3x_2^2$, sa représentation matricielle est alors : $Q_1 = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$
- $x_1^2 + 4x_1x_2 + 3x_2^2 = x_1^2 + 4x_1x_2 + 0x_2x_1 + 3x_2^2$, sa représentation matricielle est alors : $Q_2 = \begin{pmatrix} 1 & 4 \\ 0 & 3 \end{pmatrix}$
- $x_1^2 + 4x_1x_2 + 3x_2^2 = x_1^2 + 6x_1x_2 - 2x_2x_1 + 3x_2^2$, sa représentation matricielle est alors : $Q_3 = \begin{pmatrix} 1 & 6 \\ -2 & 3 \end{pmatrix}$

Les formes quadratiques

Propriété 1 : Une forme quadratique peut toujours être représentée par une matrice symétrique, puisque $f(x) = \frac{1}{2}x^T(Q + Q^T)x$. Cette représentation par une matrice symétrique est unique.

Les formes quadratiques

Propriété 1 : Une forme quadratique peut toujours être représentée par une matrice symétrique, puisque $f(x) = \frac{1}{2}x^T(Q + Q^T)x$. Cette représentation par une matrice symétrique est unique.

Dans notre la représentation matricielle symétrique de $x_1^2 + 4x_1x_2 + 3x_2^2$ est

$$Q_1 = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$$

Les formes quadratiques

Propriété 1 : Une forme quadratique peut toujours être représentée par une matrice symétrique, puisque $f(x) = \frac{1}{2}x^T(Q + Q^T)x$. Cette représentation par une matrice symétrique est unique.

Dans notre la représentation matricielle symétrique de $x_1^2 + 4x_1x_2 + 3x_2^2$ est
$$Q_1 = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$$

Dans la suite on considère que les formes quadratiques sont représentées par des matrices symétriques.

Notion de convexité

Définition 2 : On dit qu'une matrice symétrique est semi-définie positive si $\forall x \in \mathbb{R}^n$, on a $x^T Q x \geq 0$.

Notion de convexité

Définition 2 : On dit qu'une matrice symétrique est semi-définie positive si $\forall x \in \mathbb{R}^n$, on a $x^T Qx \geq 0$.

Propriété 2 : On dit qu'une forme quadratique est *convexe* lorsque sa matrice symétrique associée est semi-définie positive.

Définitions

Definition 3 : f possède un *minimum local* u si

$$\exists \delta > 0 \text{ tel que } f(u) \leq f(x), \forall x, ||x - u|| \leq \delta$$

Definition 4 : f possède un *minimum global* u si

$$\forall x, \quad f(u) \leq f(x)$$

Conditions nécessaires et suffisantes d'optimalité globale

Conditions nécessaires suffisantes d'optimalité globale Soit $f : \mathbb{R}^n \to \mathbb{R}$ deux fois différentiable dans $\mathbb{R}^n$, et soit u un minimum local de f, si f est une fonction convexe, alors u est un minimum global de f.

Exemple : un problème de planning de personnel

Données :

Exemple : un problème de planning de personnel

Données :

- Une entreprise emploie N personnes,

Exemple : un problème de planning de personnel

Données :

- Une entreprise emploie N personnes,
- Le nombre de personnes nécéssaires pour le jour $j = 1, \ldots, 7$ vaut n_j.

Exemple : un problème de planning de personnel

Données :

- Une entreprise emploie N personnes,
- Le nombre de personnes nécéssaires pour le jour $j = 1, \ldots, 7$ vaut n_j.
- Chaque jour : n_j personnes travaillent et $N - n_j$ sont en congé.

Exemple : un problème de planning de personnel

Données :

- Une entreprise emploie N personnes,
- Le nombre de personnes nécéssaires pour le jour $j = 1, \ldots, 7$ vaut n_j.
- Chaque jour : n_j personnes travaillent et $N - n_j$ sont en congé.
- Chaque personne doit bénéficier de 2 jours de congé.

Exemple : un problème de planning de personnel

Données :

- Une entreprise emploie N personnes,
- Le nombre de personnes nécéssaires pour le jour $j = 1, \dots, 7$ vaut n_j.
- Chaque jour : n_j personnes travaillent et $N - n_j$ sont en congé.
- Chaque personne doit bénéficier de 2 jours de congé.
- On suppose que $\sum_{j=1}^{7}(N - n_j) = 2N$.

Exemple : un problème de planning de personnel

Données :

- Une entreprise emploie N personnes,
- Le nombre de personnes nécéssaires pour le jour $j = 1, \ldots, 7$ vaut n_j.
- Chaque jour : n_j personnes travaillent et $N - n_j$ sont en congé.
- Chaque personne doit bénéficier de 2 jours de congé.
- On suppose que $\displaystyle\sum_{j=1}^{7}(N - n_j) = 2N$.

Problème : Affecter les jours de congé aux personnes de façon à maximiser les jours de congé consécutifs

Exemple : un problème de planning de personnel

Données :

- Une entreprise emploie N personnes,
- Le nombre de personnes nécéssaires pour le jour $j = 1, \ldots, 7$ vaut n_j.
- Chaque jour : n_j personnes travaillent et $N - n_j$ sont en congé.
- Chaque personne doit bénéficier de 2 jours de congé.
- On suppose que $\sum_{j=1}^{7}(N - n_j) = 2N$.

Problème : Affecter les jours de congé aux personnes de façon à maximiser les jours de congé consécutifs

Variables : $x_{ij} \in \{0, 1\}$

$$x_{ij} = \begin{cases} 1 \text{ si la personne i est en congé le jour j} \\ \text{sinon} \end{cases} \tag{1}$$

Exemple : un problème de planning de personnel

Une solution réalisable du problème :

jour	lundi	mardi	mercredi	jeudi	vendredi	samedi	dimanche
n_j	5	7	4	5	7	6	6
Pers. 1	congé	x	congé	x	x	x	x
Pers. 2	x	x	x	congé	x	congé	x
Pers. 3	x	congé	x	x	x	congé	x
Pers. 4	congé	x	congé	x	x	x	x
Pers. 5	congé	x	congé	x	x	x	x
Pers. 6	x	x	congé	congé	x	x	x
Pers. 7	x	x	x	congé	x	x	congé
Pers. 8	x	x	x	x	congé	x	congé

Modélisation : contraintes

Il faut s'assurer que chaque jour, le nombre de personnes en congé respecte les besoins de l'entreprise

$$\sum_{i=1}^{N} x_{ij} = N - n_j \qquad \forall j = 1, \ldots, 7$$

Modélisation : contraintes

Il faut s'assurer que chaque jour, le nombre de personnes en congé respecte les besoins de l'entreprise

$$\sum_{i=1}^{N} x_{ij} = N - n_j \qquad \forall j = 1, \dots, 7$$

Il faut s'assurer que chaque personne a exactement 2 jours de congé

$$\sum_{j=1}^{7} x_{ij} = 2 \qquad \forall i = 1, \dots, N$$

Modélisation : fonction objectif

Pour chaque personne il faut maximiser le nombre de jours de congés consécutifs qui est égal à :

$$\sum_{j=1}^{6} x_{ij}x_{ij+1} + x_{i7}x_{i1}$$

Modélisation : fonction objectif

Pour chaque personne il faut maximiser le nombre de jours de congés consécutifs qui est égal à :

$$\sum_{j=1}^{6} x_{ij}x_{ij+1} + x_{i7}x_{i1}$$

On obtient donc :

$$\max \sum_{i=1}^{N} \left(\sum_{j=1}^{6} x_{ij}x_{ij+1} + x_{i7}x_{i1}\right)$$

Problème obtenu

$$
(PQ)\begin{cases}
\max \quad \displaystyle\sum_{i=1}^{N}\Big(\sum_{j=1}^{6} x_{ij}x_{ij+1} + x_{i7}x_{i1}\Big) & \\
\displaystyle\sum_{i=1}^{N} x_{ij} = N - n_j & \forall j = 1, \ldots, 7 \\
\displaystyle\sum_{j=1}^{7} x_{ij} = 2 & \forall i = 1, \ldots, N \\
x_{ij} \in \{0, 1\} & \forall i = 1, \ldots, N, \quad \forall j = 1, \ldots, 7
\end{cases}
$$

Ce problème est un problème quadratique en variables entières et binaires.

Algorithme générique de résolution par reformulation linéaire

Problème quadratique considéré

$$
(P)\begin{cases}
\min \quad f_0(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{0ij}x_ix_j \\
\quad f_r(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{rij}x_ix_j \leq b_r & \forall r = 1,\ldots,m \\
\quad x_i \in \{0,1\} & \forall i = 1,\ldots,n
\end{cases}
$$

Problème quadratique considéré

$$
(P)\begin{cases}
\min \quad f_0(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{0ij}x_ix_j & \\[2ex]
\quad f_r(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{rij}x_ix_j \leq b_r & \forall r = 1, \ldots, m \\[2ex]
\quad x_i \in \{0, 1\} & \forall i = 1, \ldots, n
\end{cases}
$$

Avec les formes quadratiques $f_r(x)$ qui sont quelconques

Problème quadratique considéré

$$(P)\begin{cases} \min \quad f_0(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{0ij}x_ix_j \\ f_r(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{rij}x_ix_j \leq b_r & \forall r = 1,\ldots,m \\ x_i \in \{0,1\} & \forall i = 1,\ldots,n \end{cases}$$

Avec les formes quadratiques $f_r(x)$ qui sont quelconques

$\Rightarrow$ leur représentation matricielle n'est pas nécéssairement semi-définie positive et donc elles ne sont pas nécessairement convexes.

Problème quadratique considéré

$$
(P) \begin{cases}
\min \quad f_0(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} q_{0ij} x_i x_j \\
f_r(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} q_{rij} x_i x_j \leq b_r & \forall r = 1, \ldots, m \\
x_i \in \{0, 1\} & \forall i = 1, \ldots, n
\end{cases}
$$

Avec les formes quadratiques $f_r(x)$ qui sont quelconques

$\Rightarrow$ leur représentation matricielle n'est pas nécéssairement semi-définie positive et donc elles ne sont pas nécessairement convexes.

Ce problème est assez général pour modéliser de nombreux problème réels

Algorithme générique de résolution

Les algorithmes que nous connaissons pour résoudre des programmes mathématiques fonctionnent si le problème considéré est convexe

Algorithme générique de résolution

Les algorithmes que nous connaissons pour résoudre des programmes mathématiques fonctionnent si le problème considéré est convexe

Principe de l'algorithme : 2 étapes

Etape 1 : Reformuler (P) en un problème équivalent et convexe, ici linéaire.
i.e. Reformuler (P) en un problème équivalent dont les fonctions sont linéaires

Algorithme générique de résolution

Les algorithmes que nous connaissons pour résoudre des programmes mathématiques fonctionnent si le problème considéré est convexe

Principe de l'algorithme : 2 étapes

Etape 1 : Reformuler (P) en un problème équivalent et convexe, ici linéaire.
i.e. Reformuler (P) en un problème équivalent dont les fonctions sont linéaires

Etape 2 : Appliquer sur le problème reformulé un des algorithmes de résolution dédié à la programmation convexe.

Linéarisation

Rappel : Le problème considéré est le suivant :

$$(P)\begin{cases} \min \quad f_0(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{0ij}x_ix_j \\ \\ \quad f_r(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{rij}x_ix_j \leq b_r & \forall r = 1,\ldots,m \\ \quad x_i \in \{0,1\} & \forall i = 1,\ldots,n \end{cases}$$

Linéarisation

Rappel : Le problème considéré est le suivant :

$$(P) \begin{cases} \min \quad f_0(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{0ij} x_i x_j & \\ \quad f_r(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{rij} x_i x_j \leq b_r & \forall r = 1, \ldots, m \\ \quad x_i \in \{0, 1\} & \forall i = 1, \ldots, n \end{cases}$$

Le problème vient des produits de variables $x_i x_j$

Linéarisation

Rappel : Le problème considéré est le suivant :

$$(P)\begin{cases} \min \quad f_0(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{0ij}x_ix_j & \\ \quad f_r(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{rij}x_ix_j \leq b_r & \forall r = 1,\ldots,m \\ \quad x_i \in \{0,1\} & \forall i = 1,\ldots,n \end{cases}$$

Le problème vient des produits de variables x_ix_j

Idée :

Linéarisation

Rappel : Le problème considéré est le suivant :

$$(P) \begin{cases} \min \quad f_0(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} q_{0ij} x_i x_j & \\ f_r(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} q_{rij} x_i x_j \leq b_r & \forall r = 1, \ldots, m \\ x_i \in \{0, 1\} & \forall i = 1, \ldots, n \end{cases}$$

Le problème vient des produits de variables $x_i x_j$

Idée :

1. Remplacer chaque produit $x_i x_j$ par une nouvelle variable y_{ij}

Linéarisation

Rappel : Le problème considéré est le suivant :

$$(P)\begin{cases} \min \quad f_0(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{0ij}x_ix_j \\ f_r(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{rij}x_ix_j \leq b_r & \forall r = 1, \ldots, m \\ x_i \in \{0,1\} & \forall i = 1, \ldots, n \end{cases}$$

Le problème vient des produits de variables x_ix_j

Idée :

1. Remplacer chaque produit x_ix_j par une nouvelle variable y_{ij}
2. Ajouter la contrainte $y_{ij} = x_ix_j$, pour forcer l'équivalence avec le problème de départ.

Linéarisation

On obtient alors le problème suivant :

$$
(P') \begin{cases}
\min \quad f_0(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{0ij}y_{ij} & \\
f_r(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{rij}y_{ij} \leq b_r & \forall r = 1,\dots,m \\
y_{ij} = x_i x_j & \forall i = 1,\dots,n, \quad j = 1,\dots,n \\
x_i \in \{0,1\} & \forall i = 1,\dots,n
\end{cases}
$$

Linéarisation

On obtient alors le problème suivant :

$$(P') \begin{cases} \min \quad f_0(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} q_{0ij} y_{ij} & \\ \quad f_r(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} q_{rij} y_{ij} \leq b_r & \forall r = 1, \dots, m \\ \quad y_{ij} = x_i x_j & \forall i = 1, \dots, n, \quad j = 1, \dots, n \\ \quad x_i \in \{0, 1\} & \forall i = 1, \dots, n \end{cases}$$

- La nouvelle fonction objectif est linéaire

Linéarisation

On obtient alors le problème suivant :

$$(P')\begin{cases} \min \quad f_0(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{0ij}y_{ij} & \\ f_r(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{rij}y_{ij} \leq b_r & \forall r = 1,\ldots,m \\ y_{ij} = x_i x_j & \forall i = 1,\ldots,n, \quad j = 1,\ldots,n \\ x_i \in \{0,1\} & \forall i = 1,\ldots,n \end{cases}$$

- La nouvelle fonction objectif est linéaire
- les nouvelles contraintes sont linéaires

Linéarisation

On obtient alors le problème suivant :

$$(P')\begin{cases} \min \quad f_0(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{0ij}y_{ij} & \\ f_r(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{rij}y_{ij} \leq b_r & \forall r = 1,\ldots,m \\ y_{ij} = x_i x_j & \forall i = 1,\ldots,n, \quad j = 1,\ldots,n \\ x_i \in \{0,1\} & \forall i = 1,\ldots,n \end{cases}$$

- La nouvelle fonction objectif est linéaire
- les nouvelles contraintes sont linéaires
- **Mais la nouvelle contrainte $y_{ij} = x_i x_j$ est quadratique et non convexe**

Linéarisation

On obtient alors le problème suivant :

$$(P')\begin{cases} \min \quad f_0(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} q_{0ij}y_{ij} & \\ f_r(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} q_{rij}y_{ij} \leq b_r & \forall r = 1,\ldots,m \\ y_{ij} = x_i x_j & \forall i = 1,\ldots,n, \quad j = 1,\ldots,n \\ x_i \in \{0,1\} & \forall i = 1,\ldots,n \end{cases}$$

- La nouvelle fonction objectif est linéaire
- les nouvelles contraintes sont linéaires
- **Mais la nouvelle contrainte $y_{ij} = x_i x_j$ est quadratique et non convexe**

Question : Peut-on remplacer la contrainte $y_{ij} = x_i x_j$ par un ensemble de contraintes linéaires et équivalentes ?

Reformulation de $y_{ij} = x_i x_j$

Théorème 1 : On a $(E) \Longleftrightarrow y_{ij} = x_i x_j$, avec :

$$(E) \left\{ \begin{array}{l} y_{ij} \leq x_i \\ y_{ij} \leq x_j \\ y_{ij} \geq x_i + x_j - 1 \\ y_{ij} \geq 0 \\ x_i \in \{0, 1\} \end{array} \right.$$

Reformulation de $y_{ij} = x_i x_j$

Théorème 1 : On a $(E) \iff y_{ij} = x_i x_j$, avec :

$$(E) \begin{cases} y_{ij} \leq x_i \\ y_{ij} \leq x_j \\ y_{ij} \geq x_i + x_j - 1 \\ y_{ij} \geq 0 \\ x_i \in \{0, 1\} \end{cases}$$

Preuve :

- si $x_i = 0$ et/ou $x_j = 0$, on a $y_{ij} \leq x_i = 0$ et $y_{ij} \geq 0 \iff y_{ij} = 0$.

Reformulation de $y_{ij} = x_i x_j$

Théorème 1 : On a $(E) \iff y_{ij} = x_i x_j$, avec :

$$(E) \begin{cases} y_{ij} \leq x_i \\ y_{ij} \leq x_j \\ y_{ij} \geq x_i + x_j - 1 \\ y_{ij} \geq 0 \\ x_i \in \{0, 1\} \end{cases}$$

Preuve :

- si $x_i = 0$ et/ou $x_j = 0$, on a $y_{ij} \leq x_i = 0$ et $y_{ij} \geq 0 \iff y_{ij} = 0$.
- si $x_i = x_j = 1$, on a $y_{ij} \leq x_i = 1$ et $y_{ij} \geq x_i + x_j - 1 = 1 \iff y_{ij} = 1$.

Linéarisation

On obtient alors le problème suivant :

$$(PL)\left\{\begin{array}{lll}
\min \quad f_0(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{0ij} y_{ij} & & \\
\quad f_r(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{rij} y_{ij} \leq b_r & \forall r = 1, \ldots, m & \\
\quad y_{ij} \leq x_i & \forall i = 1, \ldots, n, & j = 1, \ldots, n \\
\quad y_{ij} \leq x_j & \forall i = 1, \ldots, n, & j = 1, \ldots, n \\
\quad y_{ij} \geq x_i + x_j - 1 & \forall i = 1, \ldots, n, & j = 1, \ldots, n \\
\quad y_{ij} \geq 0 & \forall i = 1, \ldots, n, & j = 1, \ldots, n \\
\quad x_i \in \{0, 1\} & \forall i = 1, \ldots, n &
\end{array}\right.$$

Linéarisation

On obtient alors le problème suivant :

$$
(PL) \begin{cases}
\min \quad f_0(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{0ij} y_{ij} & \\
f_r(x) = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} q_{rij} y_{ij} \leq b_r & \forall r = 1, \ldots, m \\
y_{ij} \leq x_i & \forall i = 1, \ldots, n, \quad j = 1, \ldots, n \\
y_{ij} \leq x_j & \forall i = 1, \ldots, n, \quad j = 1, \ldots, n \\
y_{ij} \geq x_i + x_j - 1 & \forall i = 1, \ldots, n, \quad j = 1, \ldots, n \\
y_{ij} \geq 0 & \forall i = 1, \ldots, n, \quad j = 1, \ldots, n \\
x_i \in \{0, 1\} & \forall i = 1, \ldots, n
\end{cases}
$$

Ce problème est un programme linéaire, on peut le soumettre à un algorithme de résolution pour les problèmes linéaires

Linéarisation : Illustration graphique

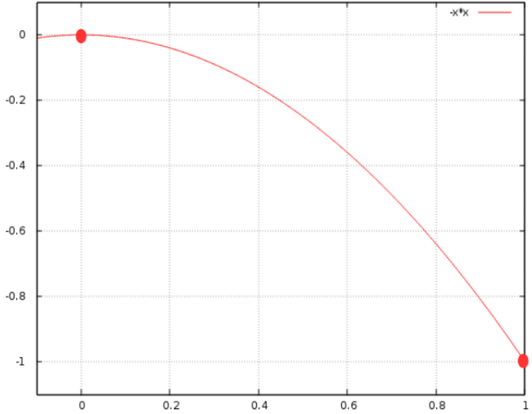
Soit le problème quadratique non convexe suivant :

$$(P)\left\{\begin{array}{ll} \min & f_0(x) = -x^2 \\ & x \in \{0,1\} \end{array}\right.$$

Linéarisation : Illustration graphique

Soit le problème quadratique non convexe suivant :

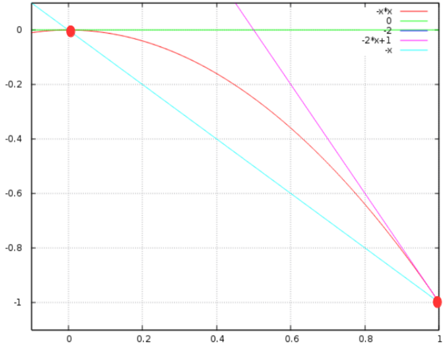
$$(P)\left\{\begin{array}{ll} \min & f_0(x) = -x^2 \\ & x \in \{0,1\} \end{array}\right.$$



Linéarisation : Illustration graphique

On reformule (P) en un programme linéaire grâce à la linéarisation, on obtient (PL) :

$$(PL)\begin{cases} \min \quad f_0(x) = -y \\ \quad y \leq x \\ \quad y \geq 2x - 1 \\ \quad y \geq 0 \\ \quad x \in \{0, 1\} \end{cases}$$



Reformulation linéaire spécifique au problème

Exemple : Localisation et transport

Données :

Exemple : Localisation et transport

Données :

- Une entreprise livre avec des camions de capacité Cm^3.

Exemple : Localisation et transport

Données :

- Une entreprise livre avec des camions de capacité Cm^3.
- Elle cherche à louer n entrepots $E_1, E_2, \ldots, E_n$. Le prix de la location de E_i est p_i et sa capacité de stockage est $c_i \times C$.

Exemple : Localisation et transport

Données :

- Une entreprise livre avec des camions de capacité Cm^3.
- Elle cherche à louer n entrepots $E_1, E_2, \ldots, E_n$. Le prix de la location de E_i est p_i et sa capacité de stockage est $c_i \times C$.
- La région est découpée en p zones $Z_1, Z_2, \ldots, Z_p$, et l'entreprise doit livrer dans ces zones avec une demande d_j.

Exemple : Localisation et transport

Données :

- Une entreprise livre avec des camions de capacité Cm^3.
- Elle cherche à louer n entrepots $E_1, E_2, \ldots, E_n$. Le prix de la location de E_i est p_i et sa capacité de stockage est $c_i \times C$.
- La région est découpée en p zones $Z_1, Z_2, \ldots, Z_p$, et l'entreprise doit livrer dans ces zones avec une demande d_j.
- Le coût de livraison depuis l'entrepot E_i vers la zone Z_j vaut l_{ij}

Exemple : Localisation et transport

Données :

- Une entreprise livre avec des camions de capacité Cm^3.
- Elle cherche à louer n entrepots $E_1, E_2, \ldots, E_n$. Le prix de la location de E_i est p_i et sa capacité de stockage est $c_i \times C$.
- La région est découpée en p zones $Z_1, Z_2, \ldots, Z_p$, et l'entreprise doit livrer dans ces zones avec une demande d_j.
- Le coût de livraison depuis l'entrepot E_i vers la zone Z_j vaut l_{ij}

Problème : Determiner les entrepôts qu'il faut louer de façon à minimiser la somme des coûts de transport

Exemple : Localisation et transport

Données :

- Une entreprise livre avec des camions de capacité Cm^3.
- Elle cherche à louer n entrepots $E_1, E_2, \ldots, E_n$. Le prix de la location de E_i est p_i et sa capacité de stockage est $c_i \times C$.
- La région est découpée en p zones $Z_1, Z_2, \ldots, Z_p$, et l'entreprise doit livrer dans ces zones avec une demande d_j.
- Le coût de livraison depuis l'entrepot E_i vers la zone Z_j vaut l_{ij}

Problème : Determiner les entrepôts qu'il faut louer de façon à minimiser la somme des coûts de transport

Variables :

- $y_i \in \{0, 1\}$ qui vaut 1 si l'entrepot E_i est loué
- $x_{ij} \in \mathbb{N}^*$ qui représente le nombre de camions qui effectueront des livraisons depuis l'entrepot E_i vers la zone Z_j.

Modélisation : contraintes

Pour chaque entrepot E_i, il faut limiter la quantité livrée à partir de l'entrepot E_i à la capacité de cet entrepot :

$$\sum_{j=1}^{p} x_{ij} \leq c_i \qquad \forall i = 1, \ldots, n$$

Modélisation : contraintes

Pour chaque entrepot E_i, il faut limiter la quantité livrée à partir de l'entrepot E_i à la capacité de cet entrepot :

$$\sum_{j=1}^{p} x_{ij} \leq c_i \qquad \forall i = 1, \ldots, n$$

Pour chaque zone Z_j, il faut satisfaire la demande :

$$\sum_{i=1}^{n} x_{ij} = d_j \qquad \forall j = 1, \ldots, p$$

Modélisation : contraintes

Pour chaque livraison entre l'entrepot E_i et la zone Z_j, il faut forcer la location d'un entrepot si une quantité est livrée depuis cet entrepot :

$$(1 - y_i)x_{ij} = 0 \qquad \forall i = 1, \dots, n, \quad \forall j = 1, \dots, p$$

Modélisation : contraintes

Pour chaque livraison entre l'entrepot E_i et la zone Z_j, il faut forcer la location d'un entrepot si une quantité est livrée depuis cet entrepot :

$$(1 - y_i)x_{ij} = 0 \qquad \forall i = 1, \dots, n, \quad \forall j = 1, \dots, p$$

Les variables y_i sont binaires et x_{ij} entières :

$$y_i \in \{0, 1\} \qquad \forall i = 1, \dots, n$$

$$x_{ij} \in \mathbb{N}^* \qquad \forall i = 1, \dots, n, \quad \forall j = 1, \dots, p$$

Modélisation : la fonction objectif

On souhaite minimiser :

Modélisation : la fonction objectif

On souhaite minimiser :

- le coût de location :

$$\sum_{i=1}^{n} p_i y_i$$

Modélisation : la fonction objectif

On souhaite minimiser :

- le coût de location :

$$\sum_{i=1}^{n} p_i y_i$$

- le coût de transport pour chaque livraison entre l'entrepot E_i et la zone Z_j

$$\sum_{i=1}^{n} \sum_{j=1}^{p} l_{ij} x_{ij}$$

Modélisation : la fonction objectif

On souhaite minimiser :

- le coût de location :

$$\sum_{i=1}^{n} p_i y_i$$

- le coût de transport pour chaque livraison entre l'entrepot E_i et la zone Z_j

$$\sum_{i=1}^{n} \sum_{j=1}^{p} l_{ij} x_{ij}$$

On obtient alors :

$$\min \sum_{i=1}^{n} p_i y_i + \sum_{i=1}^{n} \sum_{j=1}^{p} l_{ij} x_{ij}$$

Problème obtenu

$$
(PQ) \begin{cases}
\min \quad \sum_{i=1}^{n} p_i y_i + \sum_{i=1}^{n} \sum_{j=1}^{p} l_{ij} x_{ij} & \\
\sum_{j=1}^{p} x_{ij} \leq c_i & \forall i = 1, \ldots, n \\
\sum_{i=1}^{n} x_{ij} = d_j & \forall j = 1, \ldots, p \\
(1 - y_i) x_{ij} = 0 & \forall i = 1, \ldots, n, \quad \forall j = 1, \ldots, p \\
x_{ij} \in \mathbb{N}^* & \forall i = 1, \ldots, n, \quad \forall j = 1, \ldots, p \\
y_i \in \{0, 1\} & \forall i = 1, \ldots, n
\end{cases}
$$

Problème obtenu

$$
(PQ) \begin{cases}
\min \quad \sum_{i=1}^{n} p_i y_i + \sum_{i=1}^{n}\sum_{j=1}^{p} l_{ij} x_{ij} & \\
\sum_{j=1}^{p} x_{ij} \le c_i & \forall i = 1,\dots,n \\
\sum_{i=1}^{n} x_{ij} = d_j & \forall j = 1,\dots,p \\
(1 - y_i) x_{ij} = 0 & \forall i = 1,\dots,n, \quad \forall j = 1,\dots,p \\
x_{ij} \in \mathbb{N}^* & \forall i = 1,\dots,n, \quad \forall j = 1,\dots,p \\
y_i \in \{0,1\} & \forall i = 1,\dots,n
\end{cases}
$$

La contrainte $(1 - y_i)x_{ij} = 0$ est quadratique

Reformulation par un programme linéaire

On a :

- si le site E_i n'est pas ouvert ($\Longrightarrow y_i = 0$), alors pour tout $j = 1, \ldots, p$, il faut $x_{ij} = 0$,
- si le site E_i est ouvert ($\Longrightarrow y_i = 1$), il faut une contrainte qui limite seulement à c_i la quantité totale livrée en E_i.

Reformulation par un programme linéaire

On a :

- si le site E_i n'est pas ouvert ($\Longrightarrow y_i = 0$), alors pour tout $j = 1, \ldots, p$, il faut $x_{ij} = 0$,
- si le site E_i est ouvert ($\Longrightarrow y_i = 1$), il faut une contrainte qui limite seulement à c_i la quantité totale livrée en E_i.

Les contraintes

$$\sum_{j=1}^{p} x_{ij} \leq c_i \text{ et } (1 - y_i)x_{ij} = 0$$

peuvent donc être remplacées par la contrainte

$$\sum_{j=1}^{p} x_{ij} \leq c_i y_i$$

Reformulation par un programme linéaire

$$
(LP) \begin{cases}
\min \quad \displaystyle\sum_{i=1}^{n} p_i y_i + \sum_{i=1}^{n} \sum_{j=1}^{p} l_{ij} x_{ij} & \\
\displaystyle\sum_{j=1}^{p} x_{ij} \leq c_i y_i & \forall i = 1, \ldots, n \\
\displaystyle\sum_{i=1}^{n} x_{ij} = d_j & \forall j = 1, \ldots, p \\
x_{ij} \in \mathbb{N}^* & \forall i = 1, \ldots, n, \quad \forall j = 1, \ldots, p \\
y_i \in \{0, 1\} & \forall i = 1, \ldots, n
\end{cases}
$$

Ce problème est un problème linéaire en variables entières et binaires.