Programmation Quadratique en variables 0-1

Frédéric Roupin

roupin@ensiie.fr

Master M2 MMMEF

Plan du cours

1. 1. Introduction

Plan du cours

1. 1. Introduction
2. 2. Linéarisations

Plan du cours

1. 1. Introduction
2. 2. Linéarisations
3. 3. Approches Lagrangiennes

Plan du cours

1. 1. Introduction
2. 2. Linéarisations
3. 3. Approches Lagrangiennes
4. 4. Approche par Programmation Semidéfinie

Plan du cours

1. Introduction
   - a. Modèle Général étudié
   - b. Modéliser à l'aide de variables 0-1
   - c. Cas Quadratique 0-1 sans contrainte
   - d. Exemples concrets de formulations
2. Linéarisations
3. Approches Lagrangiennes
4. Approche par Programmation Semidéfinie

Plan du cours

1. Introduction
- a. Modèle Général étudié
- b. Modéliser à l'aide de variables 0-1
- c. Cas Quadratique 0-1 sans contrainte
- d. Exemples concrets de formulations

2. Linéarisations

3. Approches Lagrangiennes

4. Approche par Programmation Semidéfinie

I.a. Modèle Général étudié

Définition d'un programme Quadratique en variables 0-1

$$
(Q) \quad \begin{cases} \min & x^T Q x + b^T x = \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij} x_i x_j + \sum_{i=1}^{n} b_i x_i \\ \text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq)\, e_i \;\; i \in \{1, \ldots, m\} \\ & Ax = (\text{ou } \leq)\, b \\ & x \in \{0,1\}^n \end{cases}
$$

I.a. Modèle Général étudié

Définition d'un programme Quadratique en variables 0-1

$$
(Q)\quad \begin{cases} \min & x^TQx + b^Tx = \sum_{i=1}^n \sum_{j=1}^n Q_{ij}x_ix_j + \sum_{i=1}^n b_ix_i \\ \text{s.c.} & x^TB_ix + d_i^Tx = (\text{ou } \leq)\, e_i \;\; i \in \{1,\ldots,m\} \\ & Ax = (\text{ou } \leq)\, b \\ & x \in \{0,1\}^n \end{cases}
$$

- x : vecteur des variables de décision

I.a. Modèle Général étudié

Définition d'un programme Quadratique en variables 0-1

$$(Q)\quad \left\{\begin{array}{ll} \min & x^T Q x + b^T x = \sum_{i=1}^n \sum_{j=1}^n Q_{ij} x_i x_j + \sum_{i=1}^n b_i x_i \\ \text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq)\, e_i \;\; i \in \{1, \ldots, m\} \\ & Ax = (\text{ou } \leq)\, b \\ & x \in \{0,1\}^n \end{array}\right.$$

- x : vecteur des variables de décision
- $x \in \{0,1\}^n$ <=> $x_i^2 = x_i \quad \forall i \in \{1, \ldots, n\}$
  => problème continu !

I.a. Modèle Général étudié

Définition d'un programme Quadratique en variables 0-1

$$
(Q) \left\{ \begin{array}{ll} \min & x^T Q x + b^T x = \sum_{i=1}^n \sum_{j=1}^n Q_{ij} x_i x_j + \sum_{i=1}^n b_i x_i \\ \text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq) \, e_i \;\; i \in \{1, \ldots, m\} \\ & Ax = (\text{ou } \leq) \, b \\ & x \in \{0,1\}^n \end{array} \right.
$$

- x : vecteur des variables de décision
- $x \in \{0,1\}^n$ <=> $x_i^2 = x_i \quad \forall i \in \{1, \ldots, n\}$
  => problème continu !
- Cas particulier : programmation linéaire en 0-1

Problématique Générale

Problématique Générale

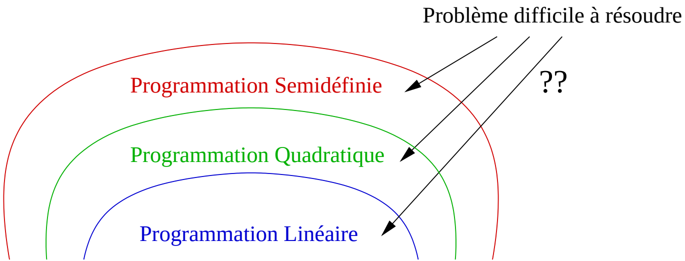
- Très large spectre d'application

Problématique Générale

- Très large spectre d'application
- Mais problèmes généralement NP-difficiles

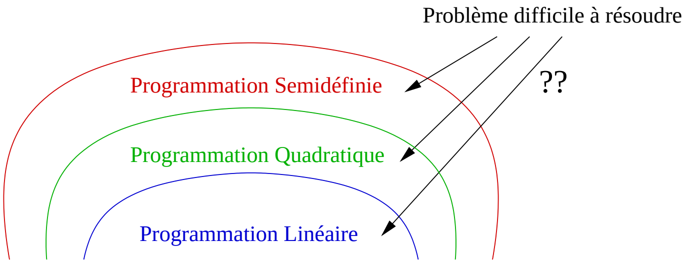
Problématique Générale

- Très large spectre d'application
- Mais problèmes généralement NP-difficiles
- Principe général : établir des relaxations convexes pour les cas difficiles pour obtenir des bornes (inférieures ou supérieures)



Problématique Générale

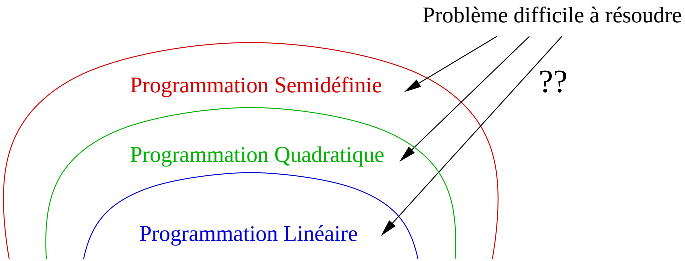
- Très large spectre d'application
- Mais problèmes généralement NP-difficiles
- Principe général : établir des relaxations convexes pour les cas difficiles pour obtenir des bornes (inférieures ou supérieures)



- méthodes de séparation/évaluation (Branch&Bound)

Problématique Générale

- Très large spectre d'application
- Mais problèmes généralement NP-difficiles
- Principe général : établir des relaxations convexes pour les cas difficiles pour obtenir des bornes (inférieures ou supérieures)



- méthodes de séparation/évaluation (Branch&Bound)
- méthodes de Branch&Cut

Plan du cours

1. Introduction
    - a. Modèle Général étudié
    - b. Modéliser à l'aide de variables 0-1
    - c. Cas Quadratique 0-1 sans contrainte
    - d. Exemples concrets de formulations

2. Linéarisations

3. Approches Lagrangiennes

4. Approche par Programmation Semidéfinie

I.b. Modéliser à l'aide de variables 0-1

- b.1 Variables prenant un ensemble fini de valeurs

I.b. Modéliser à l'aide de variables 0-1

- b.1 Variables prenant un ensemble fini de valeurs
- b.2 Contraintes alternatives/impliquées

I.b. Modéliser à l'aide de variables 0-1

- b.1 Variables prenant un ensemble fini de valeurs
- b.2 Contraintes alternatives/impliquées
- b.3 Minimum ou maximum de k fonctions bornées

I.b. Modéliser à l'aide de variables 0-1

- b.1 Variables prenant un ensemble fini de valeurs
- b.2 Contraintes alternatives/impliquées
- b.3 Minimum ou maximum de k fonctions bornées
- b.4 Fonctions linéaires par morceaux

I.b. Modéliser à l'aide de variables 0-1

- b.1 Variables prenant un ensemble fini de valeurs
- b.2 Contraintes alternatives/impliquées
- b.3 Minimum ou maximum de *k* fonctions bornées
- b.4 Fonctions linéaires par morceaux
- b.5 Fonctions en escalier

I.b. Modéliser à l'aide de variables 0-1

- b.1 Variables prenant un ensemble fini de valeurs
- b.2 Contraintes alternatives/impliquées
- b.3 Minimum ou maximum de k fonctions bornées
- b.4 Fonctions linéaires par morceaux
- b.5 Fonctions en escalier
- b.6 Ratio de fonctions pseudo-booléennes linéaires

Plan du cours

1. Introduction
- a. Modèle Général étudié
- b. Modéliser à l'aide de variables 0-1
- c. Cas Quadratique 0-1 sans contrainte
- d. Exemples concrets de formulations

2. Linéarisations

3. Approches Lagrangiennes

4. Approche par Programmation Semidéfinie

I.c. Cas Quadratique 0-1 sans contrainte

Maximiser $x^T Qx + b^T x$ avec $x \in \{0, 1\}^n$

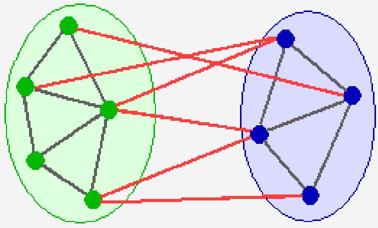
- I.c.1. Exemple de cas difficile : max-Cut $>_P$ NAE-3SAT

I.c. Cas Quadratique 0-1 sans contrainte

Maximiser $x^T Q x + b^T x$ avec $x \in \{0,1\}^n$

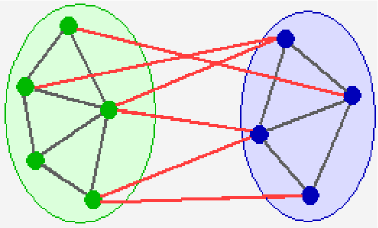
- I.c.1. Exemple de cas difficile : max-Cut $>_P$ NAE-3SAT
- I.c.2. Exemple de cas facile : maximiser avec $Q_{ij} \geq 0$ <=> problème de coupe minimale particulier (fonction sous-modulaire)

I.c.1. Problème max-cut



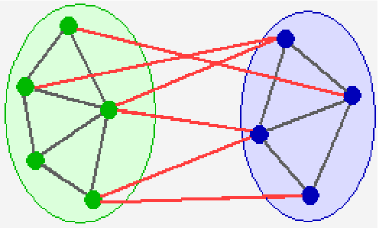
Données. Soit un graphe $G = (V, X)$ possédant n sommets et dont les arêtes $[v_i, v_j]$ sont valuées positivement par une matrice $W = \left(W_{ij}\right)$.

I.c.1. Problème max-cut



Données. Soit un graphe $G = (V, X)$ possédant n sommets et dont les arêtes $[v_i, v_j]$ sont valuées positivement par une matrice $W = \left(W_{ij}\right)$.
Question. Trouver une partition des sommets de V en deux sous-ensembles (V_1, V_2) telle que la somme des poids des arêtes ayant leurs extrémités dans des paquets différents soit maximale.

I.c.1. Problème max-cut



Données. Soit un graphe $G = (V, X)$ possédant n sommets et dont les arêtes $[v_i, v_j]$ sont valuées positivement par une matrice $W = \left(W_{ij}\right)$.
Question. Trouver une partition des sommets de V en deux sous-ensembles (V_1, V_2) telle que la somme des poids des arêtes ayant leurs extrémités dans des paquets différents soit maximale.
Modèle.

$$\begin{cases} \max \sum_{i<j} w_{ij} \left[x_i \left(1 - x_j\right) + x_j \left(1 - x_i\right)\right] \\ \text{Sous la contrainte : } x \in \{0, 1\}^n \end{cases}$$

I.c.2. Exemple de cas facile

Maximiser $x^T Qx + b^T x$ avec $x \in \{0,1\}^n$
avec :

- b_i de signe quelconque

I.c.2. Exemple de cas facile

Maximiser $x^T Q x + b^T x$ avec $x \in \{0, 1\}^n$
avec :

- b_i de signe quelconque
- Q_{ij} tous positifs

I.c.2. Exemple de cas facile

Maximiser $x^T Q x + b^T x$ avec $x \in \{0, 1\}^n$
avec :

- b_i de signe quelconque
- Q_{ij} tous positifs

I.c.2. Exemple de cas facile

Maximiser $x^T Q x + b^T x$ avec $x \in \{0,1\}^n$
avec :

- b_i de signe quelconque
- Q_{ij} tous positifs

=> Construction d'un problème équivalent flot maximal/coupe minimale

Plan du cours

1. 1. Introduction
   - a. Modèle Général étudié
   - b. Modéliser à l'aide de variables 0-1
   - c. Cas Quadratique 0-1 sans contrainte
   - d. Exemples concrets de formulations
2. 2. Linéarisations
3. 3. Approches Lagrangiennes
4. 4. Approche par Programmation Semidéfinie

I.d.1. Placement de tâches dans un système distribué

Données.

- n tâches communicantes et p processeurs

I.d.1. Placement de tâches dans un système distribué

Données.

- n tâches communicantes et p processeurs
- Une matrice C $n \times n$ représentant les coûts de communication si les tâches sont placées sur des processeurs différents. Coût de communication négligeable si les tâches sont sur le même processeur

I.d.1. Placement de tâches dans un système distribué

Données.

- n tâches communicantes et p processeurs
- Une matrice C $n \times n$ représentant les coûts de communication si les tâches sont placées sur des processeurs différents. Coût de communication négligeable si les tâches sont sur le même processeur
- une matrice Q $n \times p$ représentant les coûts d'exécution

I.d.1. Placement de tâches dans un système distribué

Données.

- n tâches communicantes et p processeurs
- Une matrice C $n \times n$ représentant les coûts de communication si les tâches sont placées sur des processeurs différents. Coût de communication négligeable si les tâches sont sur le même processeur
- une matrice Q $n \times p$ représentant les coûts d'exécution

I.d.1. Placement de tâches dans un système distribué

Données.

- n tâches communicantes et p processeurs
- Une matrice C $n \times n$ représentant les coûts de communication si les tâches sont placées sur des processeurs différents. Coût de communication négligeable si les tâches sont sur le même processeur
- une matrice Q $n \times p$ représentant les coûts d'exécution

Question. Trouver une affectation des tâches sur les processeurs minimisant le coût total (exécutions+communications)

I.d.1. Placement de tâches dans un système distribué

Modélisation.

$$(P)\left\{\begin{array}{ll} \min & \sum_{t=1}^{n}\sum_{k=1}^{p} Q_{tk}x_{tk} + \sum_{k=1}^{p}\sum_{t<t'} C_{tt'}x_{tk}(1-x_{t'k}) \\ \text{s.c.}: & \sum_{k=1}^{p} x_{tk} = 1 \ ; \ t = 1,\ldots,n \\ & x \in \{0,1\}^{np} \end{array}\right.$$

I.d.1. Placement de tâches dans un système distribué

Modélisation.

$$(P)\begin{cases} \min & \sum_{t=1}^{n}\sum_{k=1}^{p} Q_{tk}x_{tk} + \sum_{k=1}^{p}\sum_{t<t'} C_{tt'}x_{tk}(1-x_{t'k}) \\ \text{s.c.} : & \sum_{k=1}^{p} x_{tk} = 1 \; ; \; t = 1,\ldots,n \\ & x \in \{0,1\}^{np} \end{cases}$$

Cas $P = 2$: polynomial ! (NP-difficile pour $P \geq 3$)

I.d.2. Gestion de personnel

Objectif : mettre en place un planning hebdomadaire pour un ensemble de n personnes en maximisant le nombre de jours de congés consécutifs

I.d.2. Gestion de personnel

Objectif : mettre en place un planning hebdomadaire pour un ensemble de n personnes en maximisant le nombre de jours de congés consécutifs

Contraintes :

- n_j personnes sont nécessaires le jour j ($j = 1, \ldots, 7$).

I.d.2. Gestion de personnel

Objectif : mettre en place un planning hebdomadaire pour un ensemble de n personnes en maximisant le nombre de jours de congés consécutifs

Contraintes :

- n_j personnes sont nécessaires le jour j ($j = 1, \ldots, 7$).
- La personne i prend T_i jours de congés par semaine.

I.d.2. Gestion de personnel

$$\left\{\begin{array}{lll} \max & \sum_{i=1}^{n}\left(\sum_{j=1}^{6} x_{ij}x_{ij+1} + x_{i7}x_{i1}\right) & \\ \text{s.c.} & \sum_{i=1}^{n} x_{ij} = n - n_j & j = 1, \ldots, 7 \\ & \sum_{j=1}^{7} x_{ij} = T_i & i = 1, \ldots, n \\ & x_{ij} \in \{0, 1\} \;\; i = 1, \ldots, n, \; j = 1, \ldots, 7 & \end{array}\right.$$

I.d.3. Gestion de Stocks

- n type d'articles à réapprovisionner dont les demandes par unité de temps sont connues

I.d.3. Gestion de Stocks

- n type d'articles à réapprovisionner dont les demandes par unité de temps sont connues
- R : nombre maximal de réapprovisionnements par unité de temps

I.d.3. Gestion de Stocks

- n type d'articles à réapprovisionner dont les demandes par unité de temps sont connues
- R : nombre maximal de réapprovisionnements par unité de temps
- C_i : coût de stockage par unité de temps pour les articles de type i

I.d.3. Gestion de Stocks

- n type d'articles à réapprovisionner dont les demandes par unité de temps sont connues
- R : nombre maximal de réapprovisionnements par unité de temps
- C_i : coût de stockage par unité de temps pour les articles de type i
- $\Delta_1, \ldots, \Delta_m$: intervalles de temps possibles entre deux commandes successives d'un article de même type

I.d.3. Gestion de Stocks

- n type d'articles à réapprovisionner dont les demandes par unité de temps sont connues
- R : nombre maximal de réapprovisionnements par unité de temps
- C_i : coût de stockage par unité de temps pour les articles de type i
- $\Delta_1, \ldots, \Delta_m$: intervalles de temps possibles entre deux commandes successives d'un article de même type
- Choisir un intervalle de temps δ_i pour commander chaque type d'article

I.d.4. Gestion de Stocks

$$\begin{cases} \min & \sum_{i=1}^{n} C_i \delta_i \\ \text{s.c.} & \sum_{i=1}^{n} \frac{1}{\delta_i} \leq R \\ & \delta_i \in \{\Delta_1, \ldots, \Delta_m\} \quad i = 1, \ldots, n \end{cases}$$

I.d.4. Gestion de Stocks

$$\left\{ \begin{array}{lll} \min & \sum_{i=1}^{n} C_i \delta_i & \\ \text{s.c.} & \sum_{i=1}^{n} \frac{1}{\delta_i} \leq R & \\ & \delta_i \in \{\Delta_1, \ldots, \Delta_m\} & i = 1, \ldots, n \end{array} \right.$$

=> Programme non quadratique, et variables non 0-1 !

I.d.4. Gestion de Stocks

$$\left\{\begin{array}{ll} \min & \sum_{i=1}^{n} C_i \delta_i \\ \text{s.c.} & \sum_{i=1}^{n} \frac{1}{\delta_i} \leq R \\ & \delta_i \in \{\Delta_1, \dots, \Delta_m\} \quad i = 1, \dots, n \end{array}\right.$$

=> Programme non quadratique, et variables non 0-1 !

Réécriture :

$$\delta_i = \sum_{j=1}^{m} \Delta_i z_{ij} \;=>\; \frac{1}{\delta_i} = \sum_{j=1}^{m} \frac{1}{\Delta_i} z_{ij}$$

avec $z_{ij} \in \{0, 1\}$ et $\sum_{j=1}^{m} z_{ij} = 1$

I.d.4. Gestion de Stocks

$$\begin{cases} \min & \sum_{i=1}^{n} C_i\delta_i \\ \text{s.c.} & \sum_{i=1}^{n} \frac{1}{\delta_i} \leq R \\ & \delta_i \in \{\Delta_1, \ldots, \Delta_m\} \quad i = 1, \ldots, n \end{cases}$$

=> Programme non quadratique, et variables non 0-1 !

Réécriture :

$$\delta_i = \sum_{j=1}^{m} \Delta_i z_{ij} \;=>\; \frac{1}{\delta_i} = \sum_{j=1}^{m} \frac{1}{\Delta_i} z_{ij}$$

avec $z_{ij} \in \{0, 1\}$ et $\sum_{j=1}^{m} z_{ij} = 1$

=>

$$\begin{cases} \min & \sum_{i=1}^{n} C_i \sum_{j=1}^{m} \Delta_i z_{ij} \\ \text{s.c.} & \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{\Delta_i} z_{ij} \leq R \\ & \sum_{j=1}^{m} z_{ij} = 1 \\ & z_{ij} \in \{0, 1\} \;\; i = 1, \ldots, n j = 1, \ldots, m \end{cases}$$

I.d.5. Constitution d'un portefeuille d'actions

Choisir m actions parmi n en minimisant le risque et en garantissant un rendement global égal à une valeur fixée ρ

I.d.5. Constitution d'un portefeuille d'actions

Choisir m actions parmi n en minimisant le risque et en garantissant un rendement global égal à une valeur fixée ρ

- Γ : matrice des variances/co-variances

I.d.5. Constitution d'un portefeuille d'actions

Choisir m actions parmi n en minimisant le risque et en garantissant un rendement global égal à une valeur fixée ρ

- Γ : matrice des variances/co-variances
- r_i : espérance du rendement de l'action i

I.d.5. Constitution d'un portefeuille d'actions

Choisir m actions parmi n en minimisant le risque et en garantissant un rendement global égal à une valeur fixée ρ

- Γ : matrice des variances/co-variances
- r_i : espérance du rendement de l'action i
- x_i : proportion du titre i dans le portefeuille

I.d.5. Constitution d'un portefeuille d'actions

Choisir m actions parmi n en minimisant le risque et en garantissant un rendement global égal à une valeur fixée ρ

- Γ : matrice des variances/co-variances
- r_i : espérance du rendement de l'action i
- x_i : proportion du titre i dans le portefeuille

I.d.5. Constitution d'un portefeuille d'actions

Choisir m actions parmi n en minimisant le risque et en garantissant un rendement global égal à une valeur fixée ρ

- Γ : matrice des variances/co-variances
- r_i : espérance du rendement de l'action i
- x_i : proportion du titre i dans le portefeuille

$$
(\text{Markovitz}) \left\{ \begin{array}{ll} \min & x^T \Gamma x = \sum_{i=1}^n \sum_{j=1}^n \Gamma_{ij} x_i x_j \\ \text{s.c.} & \sum_{i=1}^n x_i = 1 \\ & \sum_{i=1}^n r_i x_i = \rho \\ & x \in [0,1]^n \end{array} \right.
$$

I.d.5. Constitution d'un portefeuille d'actions

Choisir m actions parmi n en minimisant le risque et en garantissant un rendement global égal à une valeur fixée ρ

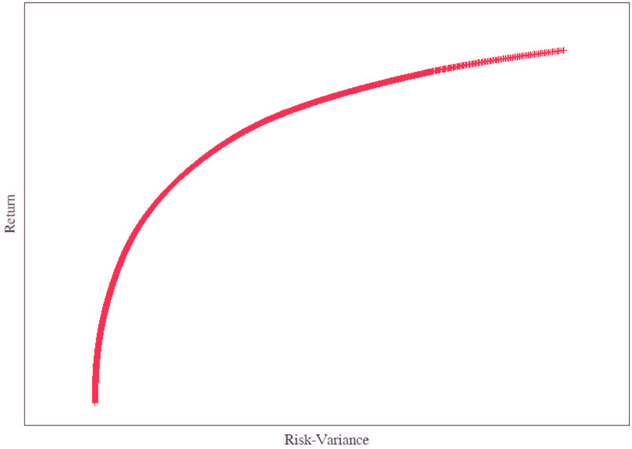
- Γ : matrice des variances/co-variances
- r_i : espérance du rendement de l'action i
- x_i : proportion du titre i dans le portefeuille

$$
\text{(Markovitz)} \quad \begin{cases} \min & x^T \Gamma x = \sum_{i=1}^n \sum_{j=1}^n \Gamma_{ij} x_i x_j \\ \text{s.c.} & \sum_{i=1}^n x_i = 1 \\ & \sum_{i=1}^n r_i x_i = \rho \\ & x \in [0,1]^n \end{cases}
$$

Problème quadratique convexe continu : facile à résoudre

Frontière d'efficience

En faisant varier la valeur du rendement souhaité, on obtient une frontière d'efficience continue et convexe !



I.d.6. Constitution d'un portefeuille d'actions (plus réaliste)

Soyons plus réalistes :

- Limitation du nombre de titres : $\sum_{i=1}^{n} y_i = m$ avec $y \in \{0, 1\}^n$, i.e. l'action i fait partie du portefeuille

I.d.6. Constitution d'un portefeuille d'actions (plus réaliste)

Soyons plus réalistes :

- Limitation du nombre de titres : $\sum_{i=1}^{n} y_i = m$ avec $y \in \{0,1\}^n$, i.e. l'action i fait partie du portefeuille
- Limitation des quantités de chaque action : $LB_i y_i \leq x_i \leq UB_i y_i$

I.d.6. Constitution d'un portefeuille d'actions (plus réaliste)

Soyons plus réalistes :

- Limitation du nombre de titres : $\sum_{i=1}^{n} y_i = m$ avec $y \in \{0,1\}^n$, i.e. l'action i fait partie du portefeuille
- Limitation des quantités de chaque action : $LB_i y_i \leq x_i \leq UB_i y_i$
- Achat des actions par fraction fixée f_i : $x_i = f_i z_i$, où $z_i \in \mathbb{N}$

I.d.6. Constitution d'un portefeuille d'actions (plus réaliste)

Soyons plus réalistes :

- Limitation du nombre de titres : $\sum_{i=1}^{n} y_i = m$ avec $y \in \{0,1\}^n$, i.e. l'action i fait partie du portefeuille
- Limitation des quantités de chaque action : $LB_i y_i \leq x_i \leq UB_i y_i$
- Achat des actions par fraction fixée f_i : $x_i = f_i z_i$, où $z_i \in \mathbb{N}$

I.d.6. Constitution d'un portefeuille d'actions (plus réaliste)

Soyons plus réalistes :

- Limitation du nombre de titres : $\sum_{i=1}^{n} y_i = m$ avec $y \in \{0,1\}^n$, i.e. l'action i fait partie du portefeuille
- Limitation des quantités de chaque action : $LB_i y_i \leq x_i \leq UB_i y_i$
- Achat des actions par fraction fixée f_i : $x_i = f_i z_i$, où $z_i \in \mathbb{N}$

=> Modèle discret : beaucoup plus difficile à résoudre !

I.d.6. Constitution d'un portefeuille d'actions (plus réaliste)

$$
\left\{
\begin{array}{lll}
\min & x^T \Gamma x = \sum_{i=1}^n \sum_{j=1}^n \Gamma_{ij} x_i x_j & \\
\text{s.c.} & \sum_{i=1}^n x_i = 1 & \\
& \sum_{i=1}^n r_i x_i = \rho & \\
& \sum_{i=1}^n y_i = m & \\
& LB_i y_i \leq x_i \leq UB_i y_i & i = 1, \ldots, n \\
& x \in [0,1]^n,\ y \in \{0,1\}^n &
\end{array}
\right.
$$

I.d.6. Constitution d'un portefeuille d'actions (plus réaliste)

$$\begin{cases} \min & x^T \Gamma x = \sum_{i=1}^n \sum_{j=1}^n \Gamma_{ij} x_i x_j \\ \text{s.c.} & \sum_{i=1}^n x_i = 1 \\ & \sum_{i=1}^n r_i x_i = \rho \\ & \sum_{i=1}^n y_i = m \\ & LB_i y_i \leq x_i \leq UB_i y_i \qquad\qquad i = 1, \ldots, n \\ & x \in [0,1]^n,\ y \in \{0,1\}^n \end{cases}$$

=> Idée : énumérer sur y pour retrouver un programme quadratique convexe continu !

Frontière d'efficience dans le cas discret (1)

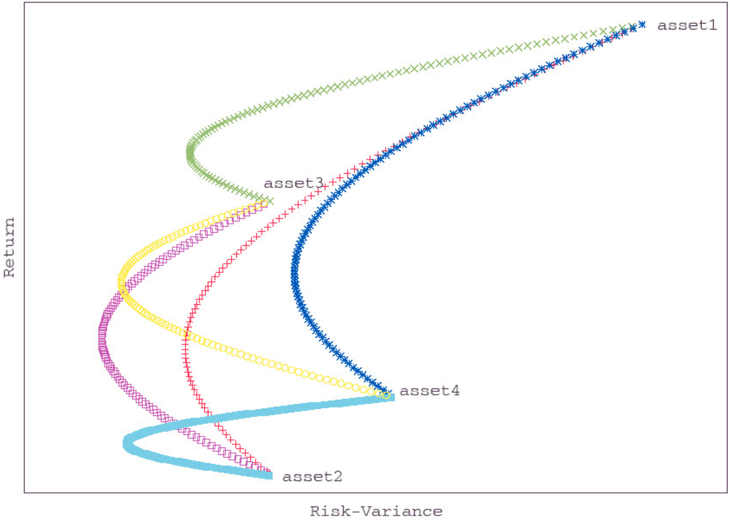
Exemple. Choisir 2 titres parmi 4 : $\sum_{i=1}^{4} y_i = 2$ avec $y \in \{0,1\}^4$

Titre	Espérance de rendement	Ecart-type
1	0.004798	0.046341
2	0.000659	0.035770
3	0.003174	0.030474
4	0.001377	0.035770

$$\begin{pmatrix} & 0.118368 & 0.143822 & 0.252213 \\ & & 0.164589 & 0.099763 \\ & & & 0.083122 \\ & & & \end{pmatrix}$$

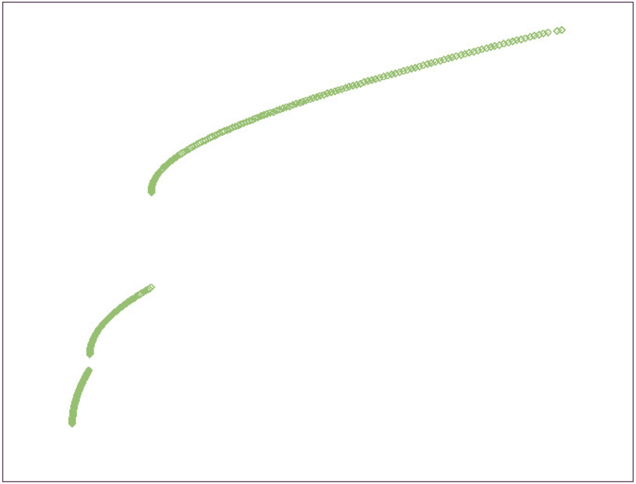
Frontière d'efficience dans le cas discret (2)

Pour chaque couple de titres parmi les quatre, on peut tracer la frontière d'efficience :



Frontière d'efficience dans le cas discret (3)

Pour un risque donné, on prend le meilleur rendement : frontière non convexe et même discontinue !



Risk-Variance

I.d.7. Problème du rééquilibrage d'un portefeuille

- x^0 : portefeuille initial, x : portefeuille final

I.d.7. Problème du rééquilibrage d'un portefeuille

- x^0 : portefeuille initial, x : portefeuille final
- Minimiser : le risque + $\alpha \times$ les coûts fixes et linéaires de transaction (commissions, courtage, taxes)

I.d.7. Problème du rééquilibrage d'un portefeuille

- x^0 : portefeuille initial, x : portefeuille final
- Minimiser : le risque + $\alpha\times$ les coûts fixes et linéaires de transaction (commissions, courtage, taxes)
- Coûts linéaires : a_i et v_i représentent respectivement les volumes de vente et d'achat de l'action i

I.d.7. Problème du rééquilibrage d'un portefeuille

- x^0 : portefeuille initial, x : portefeuille final
- Minimiser : le risque + $\alpha \times$ les coûts fixes et linéaires de transaction (commissions, courtage, taxes)
- Coûts linéaires : a_i et v_i représentent respectivement les volumes de vente et d'achat de l'action i
- Coûts fixes $c_i^F y_i$: $y_i = 1$ si $x_i \neq x_i^0$ et sinon $y_i = 0$

I.d.7. Problème du rééquilibrage d'un portefeuille

- x^0 : portefeuille initial, x : portefeuille final
- Minimiser : le risque + $\alpha\times$ les coûts fixes et linéaires de transaction (commissions, courtage, taxes)
- Coûts linéaires : a_i et v_i représentent respectivement les volumes de vente et d'achat de l'action i
- Coûts fixes $c_i^F y_i$: $y_i = 1$ si $x_i \neq x_i^0$ et sinon $y_i = 0$
- Pour l'action i : $C_i = c_i^F y_i + c_i^A a_i + c_i^V v_i$

I.d.7. Problème du rééquilibrage d'un portefeuille

- x^0 : portefeuille initial, x : portefeuille final
- Minimiser : le risque + $\alpha\times$ les coûts fixes et linéaires de transaction (commissions, courtage, taxes)
- Coûts linéaires : a_i et v_i représentent respectivement les volumes de vente et d'achat de l'action i
- Coûts fixes $c_i^F y_i$: $y_i = 1$ si $x_i \neq x_i^0$ et sinon $y_i = 0$
- Pour l'action i : $C_i = c_i^F y_i + c_i^A a_i + c_i^V v_i$

I.d.7. Problème du rééquilibrage d'un portefeuille

- x^0 : portefeuille initial, x : portefeuille final
- Minimiser : le risque + $\alpha \times$ les coûts fixes et linéaires de transaction (commissions, courtage, taxes)
- Coûts linéaires : a_i et v_i représentent respectivement les volumes de vente et d'achat de l'action i
- Coûts fixes $c_i^F y_i$: $y_i = 1$ si $x_i \neq x_i^0$ et sinon $y_i = 0$
- Pour l'action i : $C_i = c_i^F y_i + c_i^A a_i + c_i^V v_i$

$$
\left\{
\begin{array}{lll}
\min & \sum_{i=1}^n \sum_{j=1}^n \Gamma_{ij} x_i x_j + \alpha \sum_{i=1}^n C_i & \\
\text{s.c.} & \sum_{i=1}^n x_i = 1 & \\
& \sum_{i=1}^n r_i x_i \geq \rho & \\
& a_i \leq y_i(UB_i - x_i^0),\ a_i \geq x_i - x_i^0,\ a_i \geq 0 & i = 1, \ldots, n \\
& v_i \leq y_i(x_i^0 - LB_i),\ v_i \geq x_i^0 - x_i,\ v_i \geq 0 & i = 1, \ldots, n \\
& LB_i y_i \leq x_i \leq UB_i y_i & \\
& x \in [0,1]^n,\ y \in \{0,1\}^n &
\end{array}
\right.
$$

Plan du cours

1. Introduction

2. Linéarisations
- a. Linéarisation "Classique"
- b. Linéarisation "Produit"
- c. Linéarisations "Compactes"
- d. Application au problème du sac-à-dos quadratique

3. Approches Lagrangiennes

4. Approche par Programmation Semidéfinie

Plan du cours

1. 1. Introduction
2. 2. Linéarisations
   - a. Linéarisation "Classique"
   - b. Linéarisation "Produit"
   - c. Linéarisations "Compactes"
   - d. Application au problème du sac-à-dos quadratique
3. 3. Approches Lagrangiennes
4. 4. Approche par Programmation Semidéfinie

2.a. Linéarisation classique (Dantzig)

Programmation Quadratique en variables 0-1

$$
(Q) \quad \left\{
\begin{array}{lll}
\min_{x \in \Re^n} & x^T Q x = \sum_{i=1}^n \sum_{j=1}^n Q_{ij} x_i x_j & \\
\text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq) \, e_i & i \in \{1, \ldots, m\} \\
 & Ax = (\text{ou } \leq) \, b & \\
 & x \in \{0,1\}^n &
\end{array}
\right.
$$

2.a. Linéarisation classique (Dantzig)

Programmation Quadratique en variables 0-1

$$
(\mathrm{Q}) \quad \begin{cases} \min_{x\in\Re^n} & x^TQx = \sum_{i=1}^n \sum_{j=1}^n Q_{ij}x_ix_j \\ \text{s.c.} & x^TB_ix + d_i^Tx = (\text{ou } \leq)\, e_i \quad\quad i \in \{1,\ldots,m\} \\ & Ax = (\text{ou } \leq)\, b \\ & x \in \{0,1\}^n \end{cases}
$$

Linéarisation :

- Remplacer x_ix_j par Y_{ij} $\forall i < j$

2.a. Linéarisation classique (Dantzig)

Programmation Quadratique en variables 0-1

$$
(\mathrm{Q}) \quad \begin{cases}
\min_{x \in \Re^n} & x^T Q x = \sum_{i=1}^n \sum_{j=1}^n Q_{ij} x_i x_j \\
\text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq) \, e_i \quad i \in \{1, \ldots, m\} \\
& Ax = (\text{ou } \leq) \, b \\
& x \in \{0, 1\}^n
\end{cases}
$$

Linéarisation :

- Remplacer $x_i x_j$ par Y_{ij} $\forall i < j$
- Ajouter : $Y_{ij} \leq x_i$, $Y_{ij} \leq x_j$, et $Y_{ij} - x_i - x_j + 1 \geq 0$ $\forall i < j$

2.a. Linéarisation classique (Dantzig)

Programmation Quadratique en variables 0-1

$$
(\mathrm{Q}) \quad \left\{ \begin{array}{lll}
\min_{x \in \Re^n} & x^T Q x = \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij} x_i x_j & \\
\text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq)\, e_i & i \in \{1, \ldots, m\} \\
& Ax = (\text{ou } \leq)\, b & \\
& x \in \{0,1\}^n &
\end{array} \right.
$$

Linéarisation :

- Remplacer $x_i x_j$ par Y_{ij} $\forall i < j$
- Ajouter : $Y_{ij} \leq x_i$, $Y_{ij} \leq x_j$, et $Y_{ij} - x_i - x_j + 1 \geq 0$ $\forall i < j$
- Relaxation : Remplacer $x \in \{0,1\}^n$ par $0 \leq x_i \leq 1$ $\forall i$

2.a. Linéarisation classique (Dantzig)

Programmation Quadratique en variables 0-1

$$
(Q) \quad \begin{cases} \min_{x \in \Re^n} & x^T Q x = \sum_{i=1}^n \sum_{j=1}^n Q_{ij} x_i x_j \\ \text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq)\, e_i \quad i \in \{1, \ldots, m\} \\ & Ax = (\text{ou } \leq)\, b \\ & x \in \{0, 1\}^n \end{cases}
$$

Linéarisation :

- Remplacer $x_i x_j$ par Y_{ij} $\forall i < j$
- Ajouter : $Y_{ij} \leq x_i$, $Y_{ij} \leq x_j$, et $Y_{ij} - x_i - x_j + 1 \geq 0$ $\forall i < j$
- Relaxation : Remplacer $x \in \{0, 1\}^n$ par $0 \leq x_i \leq 1$ $\forall i$

2.a. Linéarisation classique (Dantzig)

Programmation Quadratique en variables 0-1

$$
(\mathrm{Q}) \left\{ \begin{array}{lll} \min_{x \in \Re^n} & x^T Q x = \sum_{i=1}^n \sum_{j=1}^n Q_{ij} x_i x_j & \\ \text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq)\, e_i & i \in \{1, \ldots, m\} \\ & Ax = (\text{ou } \leq)\, b & \\ & x \in \{0,1\}^n & \end{array} \right.
$$

Linéarisation :

- Remplacer $x_i x_j$ par Y_{ij} $\forall i < j$
- Ajouter : $Y_{ij} \leq x_i$, $Y_{ij} \leq x_j$, et $Y_{ij} - x_i - x_j + 1 \geq 0$ $\forall i < j$
- Relaxation : Remplacer $x \in \{0,1\}^n$ par $0 \leq x_i \leq 1$ $\forall i$

=> Programme linéaire continu

2.a. Linéarisation classique (Dantzig)

(PL_D)

$$\begin{cases} \min & \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij} Y_{ij} & \\ \text{s.c.} & \sum_{i=1}^{n}\sum_{j=1}^{n} B_{ij} Y_{ij} + d_i^T x = (\text{ou } \leq)\, e_i & i \in \{1,\ldots,m\} \\ & Y_{ij} \leq x_i,\, Y_{ij} \leq x_j & i < j \in \{1,\ldots,n\} \\ & Y_{ij} - x_i - x_j + 1 \geq 0 & i < j \in \{1,\ldots,n\} \\ & Ax = (\text{ou } \leq)\, b & \\ & x \in [0,1]^n \;\; Y_{ij} \geq 0\, \forall i < j \in \{1,\ldots,n\} & \end{cases}$$

2.a. Linéarisation classique (Dantzig)

(PL_D)

$$\left\{\begin{array}{lll} \min & \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij} Y_{ij} & \\ \text{s.c.} & \sum_{i=1}^{n}\sum_{j=1}^{n} B_{ij} Y_{ij} + d_i^T x = (\text{ou } \leq)\, e_i & i \in \{1,\ldots,m\} \\ & Y_{ij} \leq x_i,\, Y_{ij} \leq x_j & i < j \in \{1,\ldots,n\} \\ & Y_{ij} - x_i - x_j + 1 \geq 0 & i < j \in \{1,\ldots,n\} \\ & Ax = (\text{ou } \leq)\, b & \\ & x \in [0,1]^n \;\; Y_{ij} \geq 0\, \forall i < j \in \{1,\ldots,n\} & \end{array}\right.$$

Coût élevé : $O(n^2)$ variables et contraintes supplémentaires

Plan du cours

1. Introduction

2. Linéarisations
- a. Linéarisation "Classique"
- b. Linéarisation "Produit"
- c. Linéarisations "Compactes"
- d. Application au problème du sac-à-dos quadratique

3. Approches Lagrangiennes

4. Approche par Programmation Semidéfinie

2.b. Linéarisation "Produit" (Adams-Sherali)

On multiplie $Ax \leq b$ par x_k et par $1 - x_k$ pour tout k dans $\{1, \ldots, n\}$

2.b. Linéarisation "Produit" (Adams-Sherali)

On multiplie $Ax \leq b$ par x_k et par $1 - x_k$ pour tout k dans $\{1, \ldots, n\}$
(PL_AS)

$$\left\{\begin{array}{ll}
\min & \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij} \color{red}{Y_{ij}} \\
\text{s.c.} & \sum_{i=1}^{n} \sum_{j=1}^{n} B_{ij} Y_{ij} + d_i^T x = (\text{ou } \leq)\, e_i \qquad i \in \{1, \ldots, m\} \\
 & Y_{ij} \leq x_i,\, Y_{ij} \leq x_j \qquad i < j \in \{1, \ldots, n\} \\
 & Y_{ij} - x_i - x_j + 1 \geq 0 \qquad i < j \in \{1, \ldots, n\} \\
 & \sum_{j=1}^{k-1} A_j Y_{jk} + \sum_{j=k+1}^{n} A_j Y_{kj} (\text{ou } \leq)\, (b - A_k) x_k \; \forall k \\
 & \sum_{j=1}^{k-1} A_j (x_k - Y_{jk}) + \sum_{j=k+1}^{n} A_j (x_k - Y_{kj}) (\text{ou } \leq)\, b(1 - x_k) \; \forall k \\
 & x \in [0,1]^n \;\; Y_{ij} \geq 0
\end{array}\right.$$

Plan du cours

1. Introduction

2. Linéarisations
- a. Linéarisation "Classique"
- b. Linéarisation "Produit"
- c. Linéarisations "Compactes"
- d. Application au problème du sac-à-dos quadratique

3. Approches Lagrangiennes

4. Approche par Programmation Semidéfinie

2.c. Linéarisations "Compactes" (Glover)

- Facteur de x_i dans la fonction quadratique : $L_i(x) = \sum_{j=i+1}^{n} Q_{ij}x_j$

2.c. Linéarisations "Compactes" (Glover)

- Facteur de x_i dans la fonction quadratique : $L_i(x) = \sum_{j=i+1}^{n} Q_{ij} x_j$
- on détermine : $\alpha_i \leq L_i(x) \leq \beta_i$

2.c. Linéarisations "Compactes" (Glover)

- Facteur de x_i dans la fonction quadratique : $L_i(x) = \sum_{j=i+1}^{n} Q_{ij} x_j$
- on détermine : $\alpha_i \leq L_i(x) \leq \beta_i$
- Pour tout i dans $\{1, \ldots, n\}$, $x_i L_i(x)$ est représenté par une variable z_i

2.c. Linéarisations "Compactes" (Glover)

- Facteur de x_i dans la fonction quadratique : $L_i(x) = \sum_{j=i+1}^{n} Q_{ij} x_j$
- on détermine : $\alpha_i \leq L_i(x) \leq \beta_i$
- Pour tout i dans $\{1, \ldots, n\}$, $x_i L_i(x)$ est représenté par une variable z_i

2.c. Linéarisations "Compactes" (Glover)

- Facteur de x_i dans la fonction quadratique : $L_i(x) = \sum_{j=i+1}^{n} Q_{ij}x_j$
- on détermine : $\alpha_i \leq L_i(x) \leq \beta_i$
- Pour tout i dans $\{1, \ldots, n\}$, $x_i L_i(x)$ est représenté par une variable z_i

Contraintes :

- $\alpha_i x_i \leq z_i \leq \beta_i x_i$

2.c. Linéarisations "Compactes" (Glover)

- Facteur de x_i dans la fonction quadratique : $L_i(x) = \sum_{j=i+1}^{n} Q_{ij}x_j$
- on détermine : $\alpha_i \leq L_i(x) \leq \beta_i$
- Pour tout i dans $\{1, \ldots, n\}$, $x_i L_i(x)$ est représenté par une variable z_i

Contraintes :

- $\alpha_i x_i \leq z_i \leq \beta_i x_i$
- $L_i(x) - \beta_i(1 - x_i) \leq z_i \leq L_i(x) - \alpha_i(1 - x_i)$

Plan du cours

1. Introduction

2. Linéarisations
- a. Linéarisation "Classique"
- b. Linéarisation "Produit"
- c. Linéarisations "Compactes"
- d. Application au problème du sac-à-dos quadratique

3. Approches Lagrangiennes

4. Approche par Programmation Semidéfinie

Application au problème du sac-à-dos quadratique

- Ecriture de différentes relaxations linéaires

Application au problème du sac-à-dos quadratique

- Ecriture de différentes relaxations linéaires
- Comparaison des relaxations

Application au problème du sac-à-dos quadratique

- Ecriture de différentes relaxations linéaires
- Comparaison des relaxations
- Amélioration des relaxations compactes

Plan du cours

1. Introduction

2. Linéarisations

3. Approches Lagrangiennes
   - a. Relaxations Lagrangiennes
   - b. Méthodes de décomposition Lagrangienne

4. Approche par Programmation Semidéfinie

Plan du cours

1. Introduction

2. Linéarisations

3. Approches Lagrangiennes
- a. Relaxations Lagrangiennes
- b. Méthodes de décomposition Lagrangienne

4. Approche par Programmation Semidéfinie

Plan du cours

1. Introduction

2. Linéarisations

3. Approches Lagrangiennes
- a. Relaxations Lagrangiennes
- b. Méthodes de décomposition Lagrangienne

4. Approche par Programmation Semidéfinie

Plan du cours

1. 1. Introduction
2. 2. Linéarisations
3. 3. Approches Lagrangiennes
4. 4. Approche par Programmation Semidéfinie
   - a. Bases de la Programmation Semidéfinie
   - b. Application à la Programmation Quadratique en 0-1
     - 1. Le nombre de Lovász
     - 2. Relaxation SDP en $\{0, 1\}$
     - 3. Relaxation SDP en $\{-1, 1\}$
     - 4. Liens avec l'approche Lagrangienne
     - 5. Amélioration des relaxations Quadratiques Convexes
     - 6. Problèmes SDLS et contrainte sphérique

Plan du cours

1. Introduction

2. Linéarisations

3. Approches Lagrangiennes

4. Approche par Programmation Semidéfinie
- a. Bases de la Programmation Semidéfinie
- b. Application à la Programmation Quadratique en 0-1
  - 1. Le nombre de Lovász
  - 2. Relaxation SDP en $\{0, 1\}$
  - 3. Relaxation SDP en $\{-1, 1\}$
  - 4. Liens avec l'approche Lagrangienne
  - 5. Amélioration des relaxations Quadratiques Convexes
  - 6. Problèmes SDLS et contrainte sphérique

Généralités

- Utilisation récente en optimisation combinatoire (1993).

Généralités

- Utilisation récente en optimisation combinatoire (1993).
- Cas particulier de programmation convexe non-linéaire (fonction et contraintes convexes).

Généralités

- Utilisation récente en optimisation combinatoire (1993).
- Cas particulier de programmation convexe non-linéaire (fonction et contraintes convexes).
- Généralisation de la programmation linéaire (tout PL est un SDP particulier).

Généralités

- Utilisation récente en optimisation combinatoire (1993).
- Cas particulier de programmation convexe non-linéaire (fonction et contraintes convexes).
- Généralisation de la programmation linéaire (tout PL est un SDP particulier).
- *Avantage* : relaxation plus “fine” que la PL, bornes de très bonne qualité.

Généralités

- Utilisation récente en optimisation combinatoire (1993).
- Cas particulier de programmation convexe non-linéaire (fonction et contraintes convexes).
- Généralisation de la programmation linéaire (tout PL est un SDP particulier).
- *Avantage* : relaxation plus “fine” que la PL, bornes de très bonne qualité.
- *Inconvénient* : résolution numérique coûteuse.

Généralités

- Utilisation récente en optimisation combinatoire (1993).
- Cas particulier de programmation convexe non-linéaire (fonction et contraintes convexes).
- Généralisation de la programmation linéaire (tout PL est un SDP particulier).
- *Avantage* : relaxation plus “fine” que la PL, bornes de très bonne qualité.
- *Inconvénient* : résolution numérique coûteuse.
  - => A réserver aux problèmes très difficiles (lorsque l’approche par PL échoue).

Matrices (Semidéfinies) Positives

S_n : espace des matrices réelles symétriques $n \times n$

Matrices (Semidéfinies) Positives

S_n : espace des matrices réelles symétriques $n \times n$

Produit scalaire dans S_n :

$$(A, B) \in S_n^2 \quad A \bullet B = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} B_{ij}$$

Matrices (Semidéfinies) Positives

S_n : espace des matrices réelles symétriques $n \times n$

Produit scalaire dans S_n :

$$(A, B) \in S_n^2 \quad A \bullet B = \sum_{i=1}^n \sum_{j=1}^n A_{ij} B_{ij}$$

Definition

$A \in S_n$ est positive, $A \succcurlyeq 0$, si et seulement si pour tout $x \in \Re^n$
$x^T A x = \sum_{i=1}^n \sum_{j=1}^n A_{ij} x_i x_j = A \bullet xx^T \geq 0$

$$xx^T = \begin{bmatrix} x_1^2 & \cdots & x_1 x_i & \cdots & x_1 x_n \\ \vdots & \ddots & & & \vdots \\ x_i x_1 & & x_i^2 & & x_i x_n \\ \vdots & & & \ddots & \vdots \\ x_n x_1 & \cdots & x_n x_i & \cdots & x_n^2 \end{bmatrix}$$

Matrices (définies) positives : Définitions équivalentes

Matrices définies positives :

$A \succ 0$
$\forall z \neq 0 \in R^n \;\; z^T A z > 0$
Les valeurs propres de A sont > 0
Les mineurs principaux de A sont > 0

Matrices (définies) positives : Définitions équivalentes

Matrices définies positives :

$A \succ 0$
$\forall z \neq 0 \in R^n \quad z^T A z > 0$
Les valeurs propres de A sont > 0
Les mineurs principaux de A sont > 0

Matrices positives (semidéfinies positives)

$A \succeq 0$
$\forall z \in R^n \quad z^T A z \geq 0$
Les valeurs propres de A sont ≥ 0
Les mineurs symétriques de A sont ≥ 0

$\succeq$ définit un ordre partiel : $A \succeq B \Leftrightarrow A - B \succeq 0$.

Definition

Mineur symétrique d'une matrice :
déterminant d'une sous-matrice obtenue en choisissant un sous-ensemble de lignes et de colonnes de mêmes indices

Definition

Mineur symétrique d'une matrice :
déterminant d'une sous-matrice obtenue en choisissant un sous-ensemble de lignes et de colonnes de mêmes indices

Proposition

Une matrice est (semidéfinie) positive si et seulement si tous ses mineurs symétriques sont positifs ou nuls

Definition

Mineur symétrique d'une matrice :
déterminant d'une sous-matrice obtenue en choisissant un sous-ensemble de lignes et de colonnes de mêmes indices

Proposition

Une matrice est (semidéfinie) positive si et seulement si tous ses mineurs symétriques sont positifs ou nuls

Definition

Mineur symétrique d'une matrice :
déterminant d'une sous-matrice obtenue en choisissant un sous-ensemble de lignes et de colonnes de mêmes indices

Proposition

Une matrice est (semidéfinie) positive si et seulement si tous ses mineurs symétriques sont positifs ou nuls

Exemple : $A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 6 & 4 \\ 1 & 4 & 3 \end{bmatrix}$

- mineurs symétriques 1×1 : $1 \geq 0$, $6 \geq 0$, $3 \geq 0$

Definition

Mineur symétrique d'une matrice :
déterminant d'une sous-matrice obtenue en choisissant un sous-ensemble de lignes et de colonnes de mêmes indices

Proposition

Une matrice est (semidéfinie) positive si et seulement si tous ses mineurs symétriques sont positifs ou nuls

Exemple : $A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 6 & 4 \\ 1 & 4 & 3 \end{bmatrix}$

- mineurs symétriques 1×1 : $1 \geq 0$, $6 \geq 0$, $3 \geq 0$
- mineurs symétriques 2×2 : $6 - 4 \geq 0$, $18 - 16 \geq 0$, $3 - 1 \geq 0$

Definition

Mineur symétrique d'une matrice :
déterminant d'une sous-matrice obtenue en choisissant un sous-ensemble de lignes et de colonnes de mêmes indices

Proposition

Une matrice est (semidéfinie) positive si et seulement si tous ses mineurs symétriques sont positifs ou nuls

Exemple : $A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 6 & 4 \\ 1 & 4 & 3 \end{bmatrix}$

- mineurs symétriques 1×1 : $1 \geq 0$, $6 \geq 0$, $3 \geq 0$
- mineurs symétriques 2×2 : $6 - 4 \geq 0$, $18 - 16 \geq 0$, $3 - 1 \geq 0$
- mineur symétrique 3×3 : $\det(A) = 0 \geq 0$

Inégalités matricielles linéaires

Definition

On appelle **inégalité matricielle linéaire** toute expression de la forme $\sum_{i=1}^{n} A_i x_i \succeq A_0$, où $A_i \in S_n (i = 0, \ldots, n)$ et les x_i sont des variables réelles.

Inégalités matricielles linéaires

Definition

On appelle **inégalité matricielle linéaire** toute expression de la forme $\sum_{i=1}^{n} A_i x_i \succeq A_0$, où $A_i \in S_n (i = 0, \ldots, n)$ et les x_i sont des variables réelles.

Inégalités matricielles linéaires

Definition

On appelle **inégalité matricielle linéaire** toute expression de la forme $\sum_{i=1}^{n} A_i x_i \succeq A_0$, où $A_i \in S_n (i = 0, \ldots, n)$ et les x_i sont des variables réelles.

Exemple 1. Soit un réel t tel que $\begin{bmatrix} 1 & t & 1 \\ t & 2 & 1 \\ 1 & 1 & 4-t \end{bmatrix} \succeq 0$

Inégalités matricielles linéaires

Definition

On appelle **inégalité matricielle linéaire** toute expression de la forme $\sum_{i=1}^{n} A_i x_i \succeq A_0$, où $A_i \in S_n (i = 0, \ldots, n)$ et les x_i sont des variables réelles.

Exemple 1. Soit un réel t tel que $\begin{bmatrix} 1 & t & 1 \\ t & 2 & 1 \\ 1 & 1 & 4-t \end{bmatrix} \succeq 0$

ce qui est équivalent à l'*infinité* d'inégalités linéaires simples :

$\forall z \in R^n,\ z^T X z \geq 0$. avec $X = \begin{bmatrix} 1 & t & 1 \\ t & 2 & 1 \\ 1 & 1 & 4-t \end{bmatrix}$

Inégalités matricielles linéaires

Definition

On appelle **inégalité matricielle linéaire** toute expression de la forme $\sum_{i=1}^{n} A_i x_i \succeq A_0$, où $A_i \in S_n (i = 0, \ldots, n)$ et les x_i sont des variables réelles.

Exemple 1. Soit un réel t tel que $\begin{bmatrix} 1 & t & 1 \\ t & 2 & 1 \\ 1 & 1 & 4-t \end{bmatrix} \succeq 0$

ce qui est équivalent à l'*infinité* d'inégalités linéaires simples :

$\forall z \in R^n,\; z^T X z \geq 0$. avec $X = \begin{bmatrix} 1 & t & 1 \\ t & 2 & 1 \\ 1 & 1 & 4-t \end{bmatrix}$

ex. $\begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & t & 1 \\ t & 2 & 1 \\ 1 & 1 & 4-t \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = 7 - t \geq 0$

$$\begin{bmatrix} 1 & t & 1 \\ t & 2 & 1 \\ 1 & 1 & 4-t \end{bmatrix} \succeq 0$$

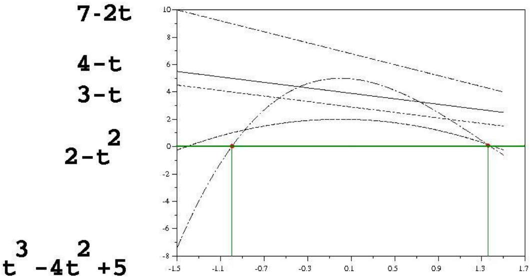
Cherchons t vérifiant l'inégalité matricielle, i.e. t vérifiant :

$4 - t \geq 0$; $2 - t^2 \geq 0$; $3 - t \geq 0$;

$7 - 2t \geq 0$; $t^3 - 4t^2 + 5 \geq 0$.

$$\begin{bmatrix} 1 & t & 1 \\ t & 2 & 1 \\ 1 & 1 & 4-t \end{bmatrix} \succeq 0$$

Cherchons t vérifiant l'inégalité matricielle, i.e. t vérifiant :
$4-t \geq 0$; $2-t^2 \geq 0$; $3-t \geq 0$;
$7-2t \geq 0$; $t^3-4t^2+5 \geq 0$.



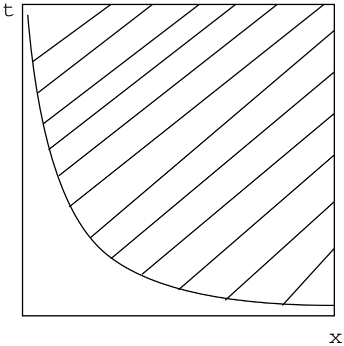
=> deux inégalités linéaires simples : $-1 \leq t \leq 1,381966$

Exemple 2. Soit x et t réels vérifiant $\begin{bmatrix} x & 1 \\ 1 & t \end{bmatrix} \succeq 0$

Exemple 2. Soit x et t réels vérifiant $\begin{bmatrix} x & 1 \\ 1 & t \end{bmatrix} \succeq 0$
$\Longleftrightarrow x \geq 0, \ t \geq 0, \ xt \geq 1$ (mineurs symétriques).

Exemple 2. Soit x et t réels vérifiant $\begin{bmatrix} x & 1 \\ 1 & t \end{bmatrix} \succeq 0$

$\Longleftrightarrow x \geq 0$, $t \geq 0$, $xt \geq 1$ (mineurs symétriques).



La région du plan obtenue est convexe mais n'est pas un polytope.

Inégalités matricielles linéaires (2)

- Soient $A_0, A_1, \ldots, A_m$ dans S_n, espace des matrices réelles symétriques $n \times n$

Inégalités matricielles linéaires (2)

- Soient $A_0, A_1, \ldots, A_m$ dans S_n, espace des matrices réelles symétriques $n \times n$
- Pour $x \in \Re^m$ on pose $F(x) = \sum_{i=1}^m x_i A_i - A_0$

Inégalités matricielles linéaires (2)

- Soient $A_0, A_1, \ldots, A_m$ dans S_n, espace des matrices réelles symétriques $n \times n$
- Pour $x \in \Re^m$ on pose $F(x) = \sum_{i=1}^{m} x_i A_i - A_0$

Inégalités matricielles linéaires (2)

- Soient $A_0, A_1, \ldots, A_m$ dans S_n, espace des matrices réelles symétriques $n \times n$
- Pour $x \in \Re^m$ on pose $F(x) = \sum_{i=1}^{m} x_i A_i - A_0$

$F(x)$ est une matrice appartenant
à l'espace affine défini par le repère :

- $-A_0$: origine

Inégalités matricielles linéaires (2)

- Soient $A_0, A_1, \ldots, A_m$ dans S_n, espace des matrices réelles symétriques $n \times n$
- Pour $x \in \Re^m$ on pose $F(x) = \sum_{i=1}^{m} x_i A_i - A_0$

$F(x)$ est une matrice appartenant
à l'espace affine défini par le repère :

- $-A_0$: origine
- $A_1, \ldots, A_m$ base

Inégalités matricielles linéaires (2)

- Soient $A_0, A_1, \ldots, A_m$ dans S_n, espace des matrices réelles symétriques $n \times n$
- Pour $x \in \Re^m$ on pose $F(x) = \sum_{i=1}^{m} x_i A_i - A_0$

$F(x)$ est une matrice appartenant
à l'espace affine défini par le repère :

- $-A_0$: origine
- $A_1, \ldots, A_m$ base
- x : coordonnées de $F(x)$ dans ce repère

Inégalités matricielles linéaires (3)

Etudions $S = \left\{ x \in \Re^m \;:\; F(x) = \sum_{i=1}^m x_i A_i - A_0 \succcurlyeq 0 \right\}$

Inégalités matricielles linéaires (3)

Etudions $S = \left\{ x \in \Re^m \,:\, F(x) = \sum_{i=1}^{m} x_i A_i - A_0 \succcurlyeq 0 \right\}$
S est défini par une inégalité matricielle linéaire

Inégalités matricielles linéaires (3)

Etudions $S = \{x \in \Re^m \,:\, F(x) = \sum_{i=1}^{m} x_i A_i - A_0 \succcurlyeq 0\}$
S est défini par une inégalité matricielle linéaire

- $F(x)$ est dans l'intersection d'un espace affine et du cône des matrices (semidéfinies) positives

Inégalités matricielles linéaires (3)

Etudions $S = \left\{ x \in \Re^m \; : \; F(x) = \sum_{i=1}^{m} x_i A_i - A_0 \succcurlyeq 0 \right\}$
S est défini par une inégalité matricielle linéaire

- $F(x)$ est dans l'intersection d'un espace affine et du cône des matrices (semidéfinies) positives
- La région recherchée est **convexe** car F est linéaire :
  $F\left(\lambda x + (1-\lambda)\, y\right) = \lambda F(x) + (1-\lambda)\, F(y) \succeq 0$
  $\forall\, \lambda \in [0,1]\,,\; x \in S,\;\; y \in S$

Inégalités matricielles linéaires (3)

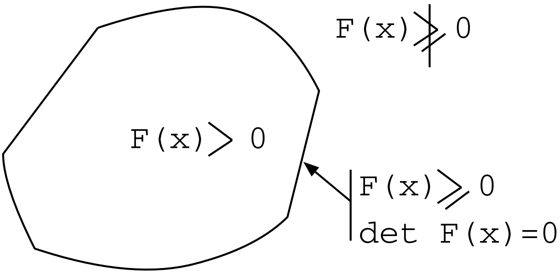
Etudions $S = \left\{x \in \Re^m \ : \ F(x) = \sum_{i=1}^m x_i A_i - A_0 \succcurlyeq 0\right\}$
S est défini par une inégalité matricielle linéaire

- $F(x)$ est dans l'intersection d'un espace affine et du cône des matrices (semidéfinies) positives
- La région recherchée est **convexe** car F est linéaire :
  $F\left(\lambda x + (1-\lambda)\, y\right) = \lambda F(x) + (1-\lambda)\, F(y) \succeq 0$
  $\forall\, \lambda \in [0,1]\,,\; x \in S,\;\; y \in S$
- mais n'est généralement pas un polytope pour $m \geq 2$:
  x et y réels vérifiant $\begin{bmatrix} x & 1 \\ 1 & y \end{bmatrix} \succeq 0$
  $\iff x \geq 0,\, y \geq 0,\, xy \geq 1$ (mineurs symétriques)

$$S = \{x \in \Re^m \; : \; \sum_{i=1}^m A_i x_i - A_0 \succcurlyeq 0\}$$

Théorème

$$\overset{\circ}{S} = \{x \mid F(x) \succ 0\} \quad et \quad \overline{S} = \{x \mid F(x) \succeq 0 \; et \; \det F(x) = 0\}$$



Nature de la frontière

det $F(x) = 0 \Leftrightarrow x$ annule un certain nombre de polynômes. En effet certains mineurs de $F(x)$ sont nuls (le nombre dépend du rang de $F(x)$).

=> On obtient des surfaces algèbriques (en PL : hyperplans)

Nature de la frontière

det $F(x) = 0 \Leftrightarrow x$ annule un certain nombre de polynômes. En effet certains mineurs de $F(x)$ sont nuls (le nombre dépend du rang de $F(x)$).

=> On obtient des surfaces algèbriques (en PL : hyperplans)

Exemple. $F(x) = \begin{bmatrix} x_1 & 1 & 0 \\ 1 & x_2+1 & 0 \\ 0 & 0 & x_2 \end{bmatrix} \succeq 0$

On cherche x tel que det $F(x) = 0$.

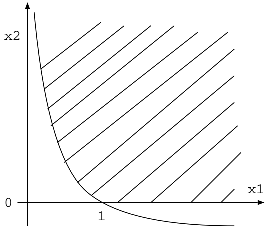
Nature de la frontière

det $F(x) = 0 \Leftrightarrow x$ annule un certain nombre de polynômes. En effet certains mineurs de $F(x)$ sont nuls (le nombre dépend du rang de $F(x)$).

=> On obtient des surfaces algèbriques (en PL : hyperplans)

Exemple. $F(x) = \begin{bmatrix} x_1 & 1 & 0 \\ 1 & x_2+1 & 0 \\ 0 & 0 & x_2 \end{bmatrix} \succeq 0$

On cherche x tel que det $F(x) = 0$.



Frontière : $\{x : x_1(x_2+1) - 1 = 0,\ 0 < x_1 < 1\} \cup \{x : x_2 = 0,\ x_1 \geq 1\}$

Programmes Semidéfinis

Definition

Un programme semidéfini est la minimisation d'une forme linéaire de $\Re^m$ soumise à une inégalité matricielle linéaire

$$(SDP)\left\{\begin{array}{ll} \min & c^T x \\ \text{s.c.} & F(x) = \sum_{i=1}^m x_i A_i - A_0 \succcurlyeq 0 \end{array}\right.$$

Programmes Semidéfinis

Definition

Un programme semidéfini est la minimisation d'une forme linéaire de $\Re^m$ soumise à une inégalité matricielle linéaire

$$(SDP) \begin{cases} \min & c^T x \\ \text{s.c.} & F(x) = \sum_{i=1}^m x_i A_i - A_0 \succcurlyeq 0 \end{cases}$$

Programmes Semidéfinis

Definition

Un programme semidéfini est la minimisation d'une forme linéaire de $\Re^m$ soumise à une inégalité matricielle linéaire

$$(SDP)\left\{\begin{array}{ll}\min & c^T x\\ \text{s.c.} & F(x) = \sum_{i=1}^m x_i A_i - A_0 \succcurlyeq 0\end{array}\right.$$

Proposition

La programmation linéaire est un cas particulier de la programmation semidéfinie :

$$(PL)\left\{\begin{array}{ll}\min & c^T x\\ \text{s.c.} & F(x) = \mathbf{diag}(Ax - b) \succeq 0 \Leftrightarrow Ax - b \geq 0\end{array}\right.$$

$A_i^T x - b_i$ est valeur propre positive de la matrice diagonale $F(x)$

Formulation d'un PL en PSD

$$(PL)\begin{cases}\text{Minimiser} & c^T x\\ \text{Sous contrainte} & F(x) = \mathbf{diag}(Ax-b) \succeq 0\end{cases}$$

Formulation d'un PL en PSD

$$(PL) \begin{cases} \text{Minimiser} & c^T x \\ \text{Sous contrainte} & F(x) = \textbf{diag}(Ax - b) \succeq 0 \end{cases}$$

Exemple.

$$(P) \begin{cases} \text{Minimiser} & x_1 + 3x_2 \\ \text{s.}c. \ : \ 2x_1 - x_2 \geq 1 \\ \quad 5x_1 + 7x_2 \geq -6 \end{cases}$$

Formulation d'un PL en PSD

$$(PL)\begin{cases} \text{Minimiser} & c^T x \\ \text{Sous contrainte} & F(x) = \mathbf{diag}(Ax - b) \succeq 0 \end{cases}$$

Exemple.

$$(P)\begin{cases} \text{Minimiser} & x_1 + 3x_2 \\ \text{s.}c. \ : \ 2x_1 - x_2 \geq 1 \\ \qquad 5x_1 + 7x_2 \geq -6 \end{cases}$$

$$\Leftrightarrow \quad (SDP)\begin{cases} \text{Minimiser} & x_1 + 3x_2 \\ \text{s.}c. \ : & \begin{bmatrix} 2x_1 - x_2 - 1 & 0 \\ 0 & 5x_1 + 7x_2 + 6 \end{bmatrix} \succeq 0 \end{cases}$$

Compléments de Schur

Théorème

Si A matrice $p \times p$ définie positive, $C \in S_n$, et B matrice $p \times n$, alors $\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succcurlyeq 0$ *est équivalent à* $C - B^T A^{-1} B \succcurlyeq 0$.

Compléments de Schur

Théorème

Si A matrice $p \times p$ définie positive, $C \in S_n$, et B matrice $p \times n$, alors
$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succcurlyeq 0$ *est équivalent à* $C - B^T A^{-1} B \succcurlyeq 0$.

Compléments de Schur

Théorème

Si A matrice $p \times p$ définie positive, $C \in S_n$, et B matrice $p \times n$, alors $\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succcurlyeq 0$ *est équivalent à* $C - B^T A^{-1} B \succcurlyeq 0$.

Preuve.

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} = \begin{bmatrix} I & 0 \\ B^T A^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & C - B^T A^{-1} B \end{bmatrix} \begin{bmatrix} I & A^{-1} B \\ 0 & I \end{bmatrix}$$

Compléments de Schur

Théorème

Si A matrice $p \times p$ définie positive, $C \in S_n$, et B matrice $p \times n$, alors $\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succcurlyeq 0$ *est équivalent à* $C - B^T A^{-1} B \succcurlyeq 0$.

Preuve.

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} = \begin{bmatrix} I & 0 \\ B^T A^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & C - B^T A^{-1} B \end{bmatrix} \begin{bmatrix} I & A^{-1}B \\ 0 & I \end{bmatrix}$$

Cas particuliers importants :

- $A = I_p$. On a $\begin{bmatrix} I_p & B \\ B^T & C \end{bmatrix} \succeq 0 \Leftrightarrow C - B^T B \succeq 0$

Compléments de Schur

Théorème

Si A matrice $p \times p$ définie positive, $C \in S_n$, et B matrice $p \times n$, alors $\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succ 0$ *est équivalent à* $C - B^T A^{-1} B \succ 0$.

Preuve.

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} = \begin{bmatrix} I & 0 \\ B^T A^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & C - B^T A^{-1} B \end{bmatrix} \begin{bmatrix} I & A^{-1}B \\ 0 & I \end{bmatrix}$$

Cas particuliers importants :

- $A = I_p$. On a $\begin{bmatrix} I_p & B \\ B^T & C \end{bmatrix} \succeq 0 \Leftrightarrow C - B^T B \succeq 0$
- $p = 1$, $C = X$, $B = x^T$. Soit $(X, x) \in S_n \times \Re^n$, on a

$$X - xx^T \succeq 0 \Leftrightarrow \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0$$

Exemple de PSD non linéaire non formulable en PL

$$(P)\begin{cases} \text{Minimiser} & \dfrac{(c^T x)^2}{d^T x} \\ \text{s.c.} & Ax + b \geq 0 \end{cases}$$

Hypothèse : $Ax + b \geq 0$ implique $d^T x > 0$.

Exemple de PSD non linéaire non formulable en PL

$$(P)\left\{\begin{array}{ll}\text{Minimiser} & \dfrac{(c^Tx)^2}{d^Tx} \\ \text{s.c.} & Ax + b \geq 0\end{array}\right.$$

Hypothèse : $Ax + b \geq 0$ implique $d^Tx > 0$.

$$\left\{\begin{array}{ll}\text{Minimiser} & t \\ \text{s.c.} & Ax + b \geq 0 \\ & \dfrac{(c^Tx)^2}{d^Tx} \leq t\end{array}\right.$$

Exemple de PSD non linéaire non formulable en PL

$$(P)\begin{cases} \text{Minimiser} & \dfrac{(c^T x)^2}{d^T x} \\ \text{s.c.} & Ax + b \geq 0 \end{cases}$$

Hypothèse : $Ax + b \geq 0$ implique $d^T x > 0$.

$$\begin{cases} \text{Minimiser} & t \\ \text{s.c.} & Ax + b \geq 0 \\ & \dfrac{(c^T x)^2}{d^T x} \leq t \end{cases}$$

t : complément de Schur

$$\begin{cases} \text{Minimiser} & t \\ \text{s.c.} & \begin{bmatrix} \mathbf{diag}(Ax+b) & 0 & 0 \\ 0 & t & c^T x \\ 0 & c^T x & d^T x \end{bmatrix} \succeq 0 \end{cases}$$

Exemple d'application : Résolution approchée d'un système linéaire

Soit $A \in \mathbb{R}^{p \times n}$, et $Ax = b$ n'admettant pas de solution.

Exemple d'application : Résolution approchée d'un système linéaire

Soit $A \in \mathbb{R}^{p \times n}$, et $Ax = b$ n'admettant pas de solution. On décide de minimiser :

$$\max_{i \in \{1,\ldots,p\}} \left| A_i^T x - b_i \right|$$

avec A_i : ième ligne ($i \in \{1,\ldots,p\}$) de A

1. Modéliser ce problème comme un programme linéaire.

Exemple d'application : Résolution approchée d'un système linéaire

Soit $A \in \mathbb{R}^{p \times n}$, et $Ax = b$ n'admettant pas de solution. On décide de minimiser :

$$\max_{i \in \{1, \ldots, p\}} \left| A_i^T x - b_i \right|$$

avec A_i : ième ligne ($i \in \{1, \ldots, p\}$) de A

1. Modéliser ce problème comme un programme linéaire.
2. Dans certaines applications les quantités b_i sont exprimées à l'aide d'une échelle logarithmique (car elles représentent par exemple une puissance ou une intensité). On souhaite alors minimiser

$$\max_{i \in \{1, ..., p\}} \left| log\left(A_i^T x\right) - log\left(b_i\right) \right|$$

modéliser ce nouveau problème comme un programme semidéfini.

Programmation Quadratique Convexe

Proposition

La programmation Quadratique Convexe est un cas particulier de la Programmation Semidéfinie

$\forall i \in \{0, \ldots, m\}$ $f_i(x) = x^T A_i^T A_i x + c_i^T x + d_i$ est convexe

$$(Q) \begin{cases} \min & f_0(x) \\ \text{s.c.} & f_i(x) \leq 0 \quad i = 1, \ldots, m \end{cases}$$

Programmation Quadratique Convexe

Proposition

La programmation Quadratique Convexe est un cas particulier de la Programmation Semidéfinie

$\forall i \in \{0, \ldots, m\}$ $f_i(x) = x^T A_i^T A_i x + c_i^T x + d_i$ est convexe

$$(Q) \begin{cases} \min & f_0(x) \\ \text{s.c.} & f_i(x) \leq 0 \quad i = 1, \ldots, m \end{cases}$$

Programmation Quadratique Convexe

Proposition

La programmation Quadratique Convexe est un cas particulier de la Programmation Semidéfinie

$\forall i \in \{0, \ldots, m\}$ $f_i(x) = x^T A_i^T A_i x + c_i^T x + d_i$ est convexe

$$(Q) \begin{cases} \min & f_0(x) \\ \text{s.c.} & f_i(x) \leq 0 \quad i = 1, \ldots, m \end{cases}$$

Schur : $f_i(x) \leq 0 \Leftrightarrow \begin{bmatrix} I_n & A_i x \\ x^T A_i^T & -c_i^T x - d_i \end{bmatrix} \succeq 0$

Programmation Quadratique Convexe

Proposition

La programmation Quadratique Convexe est un cas particulier de la Programmation Semidéfinie

$\forall i \in \{0,\ldots,m\}$ $f_i(x) = x^T A_i^T A_i x + c_i^T x + d_i$ est convexe

$$(Q)\begin{cases} \min & f_0(x) \\ \text{s.c.} & f_i(x) \leq 0 \quad i = 1,\ldots,m \end{cases}$$

Schur : $f_i(x) \leq 0 \Leftrightarrow \begin{bmatrix} I_n & A_i x \\ x^T A_i^T & -c_i^T x - d_i \end{bmatrix} \succeq 0$

$$(Q) \Leftrightarrow \begin{cases} \min & t \\ \text{s.c.} & \begin{bmatrix} I_n & A_0 x \\ x^T A_0^T & -c_i^T x - d_i + t \end{bmatrix} \succeq 0 \qquad f_0(x) - t \leq 0 \\ & \begin{bmatrix} I_n & A_i x \\ x^T A_i^T & -c_i^T x - d_i \end{bmatrix} \succeq 0 \;\; \forall i \neq 0 \end{cases}$$

Programmation Quadratique Convexe (2)

$$(Q)\left\{\begin{array}{lll}\min & x^TA_0^TA_0x + c_0^Tx + d_0 (\leq t) & \\ \text{s.c.} & x^TA_i^TA_ix + c_i^Tx + d_i \leq 0 & i = 1, \ldots, m\end{array}\right.$$

Utilisation d'une unique inégalité matricielle linéaire :

$$(Q) \Leftrightarrow \left\{\begin{array}{ll}\min & t \\ \text{s.c.} & \begin{bmatrix} d_0 - t & \frac{1}{2}c_0^T \\ \frac{1}{2}c_0 & A_0^TA_0 \end{bmatrix} \bullet \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \leq 0 \\ & \begin{bmatrix} d_i & \frac{1}{2}c_i^T \\ \frac{1}{2}c_i & A_i^TA_i \end{bmatrix} \bullet \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \leq 0 \\ & \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0 \end{array}\right.$$

$$\underbrace{A_i^TA_i \bullet (X - xx^T)}_{\geq 0} + x^TA_i^TA_ix + c_i^Tx + d_i \leq 0$$

Utilisation de la même idée en PL

$$
(PL)\left\{\begin{array}{ll}
\min & x_1 + 4x_2 \\
\text{Sous les contraintes} & x_1 - x_2 + x_3 = 2 \\
 & 3x_1 - 5x_3 = 6 \\
 & x_1 \geq 0\,,\ x_2 \geq 0\,,\ x_3 \geq 0
\end{array}\right.
$$

est équivalent à

$$
(SDP)\left\{\begin{array}{ll}
\text{Minimiser} & x_1 + 4x_2 \\
\text{Sous les contraintes} & \left(e_1 e_1^T + e_2 e_2^T\right) \bullet X - (4e_1 + 12e_2)^T x + 40 = 0 \\
 & \begin{bmatrix} \mathbf{diag}(x) & 0 & 0 \\ 0 & 1 & x^T \\ 0 & x & X \end{bmatrix} \succeq 0
\end{array}\right.
$$

où $e_1 = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$, $e_2 = \begin{pmatrix} 3 \\ 0 \\ -5 \end{pmatrix}$

Dualité

- Le programme dual de (SDP) est

$$\sup_{Z \succcurlyeq 0} \inf_{x \in \Re^m} L(x, Z)$$

avec $L(x, Z) = c^T x + Z \bullet \left(A_0 - \sum_{i=1}^{m} x_i A_i\right)$

Dualité

- Le programme dual de (SDP) est

$$\sup_{Z \succcurlyeq 0} \inf_{x \in \Re^m} L(x, Z)$$

avec $L(x, Z) = c^T x + Z \bullet \left(A_0 - \sum_{i=1}^m x_i A_i\right)$

  - Le multiplicateur de Lagrange Z associé à l'inégalité matricielle linéaire $F(x) \succcurlyeq 0$ est une matrice positive

Dualité

- Le programme dual de (SDP) est

$$\sup_{Z \succcurlyeq 0} \inf_{x \in \Re^m} L(x, Z)$$

avec $L(x, Z) = c^T x + Z \bullet \left(A_0 - \sum_{i=1}^m x_i A_i\right)$

  - Le multiplicateur de Lagrange Z associé à l'inégalité matricielle linéaire $F(x) \succcurlyeq 0$ est une matrice positive
  - A comparer avec le lagrangien d'une inégalité linéaire $a^T x - b \geq 0$: $L(x, \lambda) = c^T x + \lambda^T (b - a^T x)$ avec $\lambda \geq 0$

Dualité

- Le programme dual de (SDP) est

$$\sup_{Z \succcurlyeq 0} \inf_{x \in \Re^m} L(x, Z)$$

avec $L(x, Z) = c^T x + Z \bullet \left(A_0 - \sum_{i=1}^m x_i A_i\right)$

  - Le multiplicateur de Lagrange Z associé à l'inégalité matricielle linéaire $F(x) \succcurlyeq 0$ est une matrice positive
  - A comparer avec le lagrangien d'une inégalité linéaire $a^T x - b \geq 0$: $L(x, \lambda) = c^T x + \lambda^T (b - a^T x)$ avec $\lambda \geq 0$

- Fonction duale :

Dualité

- Le programme dual de (SDP) est

$$\sup_{Z \succcurlyeq 0} \inf_{x \in \Re^m} L(x, Z)$$

avec $L(x, Z) = c^T x + Z \bullet (A_0 - \sum_{i=1}^m x_i A_i)$

  - Le multiplicateur de Lagrange Z associé à l'inégalité matricielle linéaire $F(x) \succcurlyeq 0$ est une matrice positive
  - A comparer avec le lagrangien d'une inégalité linéaire $a^T x - b \geq 0$: $L(x, \lambda) = c^T x + \lambda^T (b - a^T x)$ avec $\lambda \geq 0$
- Fonction duale :
  - si $A_i \bullet Z = c_i \; \forall i \in \{1, \dots, m\}$ alors $\inf_{x \in \Re^m} L(x, Z) = A_0 \bullet Z$

Dualité

- Le programme dual de (SDP) est

$$\sup_{Z \succcurlyeq 0} \inf_{x \in \Re^m} L(x, Z)$$

avec $L(x, Z) = c^T x + Z \bullet \left(A_0 - \sum_{i=1}^m x_i A_i\right)$

  - Le multiplicateur de Lagrange Z associé à l'inégalité matricielle linéaire $F(x) \succcurlyeq 0$ est une matrice positive
  - A comparer avec le lagrangien d'une inégalité linéaire $a^T x - b \geq 0$: $L(x, \lambda) = c^T x + \lambda^T (b - a^T x)$ avec $\lambda \geq 0$

- Fonction duale :
  - si $A_i \bullet Z = c_i \ \forall i \in \{1, \ldots, m\}$ alors $\inf_{x \in \Re^m} L(x, Z) = A_0 \bullet Z$
  - Sinon : les termes en x_i ne sont pas tous annulés et $\inf_{x \in \Re^m} L(x, Z) = -\infty$

Dualité

- Le programme dual de (SDP) est

$$\sup_{Z \succcurlyeq 0} \inf_{x \in \Re^m} L(x, Z)$$

avec $L(x, Z) = c^T x + Z \bullet \left(A_0 - \sum_{i=1}^m x_i A_i\right)$

  - Le multiplicateur de Lagrange Z associé à l'inégalité matricielle linéaire $F(x) \succcurlyeq 0$ est une matrice positive
  - A comparer avec le lagrangien d'une inégalité linéaire $a^T x - b \geq 0$: $L(x, \lambda) = c^T x + \lambda^T (b - a^T x)$ avec $\lambda \geq 0$

- Fonction duale :
  - si $A_i \bullet Z = c_i \ \forall i \in \{1, \ldots, m\}$ alors $\inf_{x \in \Re^m} L(x, Z) = A_0 \bullet Z$
  - Sinon : les termes en x_i ne sont pas tous annulés et $\inf_{x \in \Re^m} L(x, Z) = -\infty$

Proposition

Le programme dual de (SDP) est un programme semidéfini :

$$(DSDP) \left\{ \begin{array}{lll} \sup & A_0 \bullet Z & \\ \text{s.c.} & A_i \bullet Z = c_i & i \in \{1, \ldots, m\} \\ & Z \succeq 0 & \end{array} \right.$$

Formes Duale et Primale

(DSDP) est-il bien un programme semidéfini ?

Formes Duale et Primale

(DSDP) est-il bien un programme semidéfini ?

- Dans le primal : variable $x \in \Re^m$

Formes Duale et Primale

(DSDP) est-il bien un programme semidéfini ?

- Dans le primal : variable $x \in \Re^m$
- Dans le dual : variable $Z \in S_n$

Formes Duale et Primale

(DSDP) est-il bien un programme semidéfini ?

- Dans le primal : variable $x \in \Re^m$
- Dans le dual : variable $Z \in S_n$
- $F(x)$ et Z appartiennent à l'intersection d'un espace affine (pas le même !) et du cône des matrices positives

Formes Duale et Primale

(DSDP) est-il bien un programme semidéfini ?

- Dans le primal : variable $x \in \Re^m$
- Dans le dual : variable $Z \in S_n$
- $F(x)$ et Z appartiennent à l'intersection d'un espace affine (pas le même !) et du cône des matrices positives
- Soit un repère $(-B_0, B_1, \ldots, B_p)$ (origine + base) de $\aleph_0 = \{Z\,;\, Z \in S_n,\ A_i \bullet Z = c_i,\ i = 1, \ldots, p\}$

Formes Duale et Primale

(DSDP) est-il bien un programme semidéfini ?

- Dans le primal : variable $x \in \Re^m$
- Dans le dual : variable $Z \in S_n$
- $F(x)$ et Z appartiennent à l'intersection d'un espace affine (pas le même !) et du cône des matrices positives
- Soit un repère $(-B_0, B_1, \ldots, B_p)$ (origine + base) de $\aleph_0 = \{Z\,;\, Z \in S_n,\ A_i \bullet Z = c_i,\ i = 1, \ldots, p\}$
- $Z = F(y) = -B_0 + \sum_{i=1}^{p} y_i B_i$

Formes Duale et Primale

(DSDP) est-il bien un programme semidéfini ?

- Dans le primal : variable $x \in \Re^m$
- Dans le dual : variable $Z \in S_n$
- $F(x)$ et Z appartiennent à l'intersection d'un espace affine (pas le même !) et du cône des matrices positives
- Soit un repère $(-B_0, B_1, \ldots, B_p)$ (origine + base) de $\aleph_0 = \{Z\,;\, Z \in S_n,\ A_i \bullet Z = c_i,\ i = 1, \ldots, p\}$
- $Z = F(y) = -B_0 + \sum_{i=1}^{p} y_i B_i$

Formes Duale et Primale

(DSDP) est-il bien un programme semidéfini ?

- Dans le primal : variable $x \in \Re^m$
- Dans le dual : variable $Z \in S_n$
- $F(x)$ et Z appartiennent à l'intersection d'un espace affine (pas le même !) et du cône des matrices positives
- Soit un repère $(-B_0, B_1, \ldots, B_p)$ (origine + base) de $\aleph_0 = \{Z\,;\, Z \in S_n,\ A_i \bullet Z = c_i,\ i = 1, \ldots, p\}$
- $Z = F(y) = -B_0 + \sum_{i=1}^{p} y_i B_i$

On peut réécrire $(DSDP)$ sous une forme primale :

$$(DSDP) \left\{ \begin{array}{ll} \text{sup} & A_0 \bullet Z = A_0 \bullet \left(-B_0 + \sum_{i=1}^{p} y_i B_i\right) \\ \text{s.c. :} & A_i \bullet Z = c_i \ \ i = 1, \ldots, m \\ & \Leftrightarrow Z = F(y) = -B_0 + \sum_{i=1}^{p} y_i B_i \succcurlyeq 0 \end{array} \right.$$

Théorèmes de Dualité

Lemme

Soit A et B dans S_n. Si $A \succcurlyeq 0$ et $B \succcurlyeq 0$ alors $A \bullet B \geq 0$ et l'égalité est atteinte si et seulement si $AB = 0$

Théorèmes de Dualité

Lemme

Soit A et B dans S_n. Si $A \succcurlyeq 0$ et $B \succcurlyeq 0$ alors $A \bullet B \geq 0$ et l'égalité est atteinte si et seulement si $AB = 0$

Théorèmes de Dualité

Lemme

Soit A et B dans S_n. Si $A \succcurlyeq 0$ et $B \succcurlyeq 0$ alors $A \bullet B \geq 0$ et l'égalité est atteinte si et seulement si $AB = 0$

Dualité Faible

Le saut de dualité entre (DSP) et (SDP)
vaut :

Théorèmes de Dualité

Lemme

Soit A et B dans S_n. Si $A \succcurlyeq 0$ et $B \succcurlyeq 0$ alors $A \bullet B \geq 0$ et l'égalité est atteinte si et seulement si $AB = 0$

Dualité Faible

Le saut de dualité entre (DSP) et (SDP)
vaut :$c^T x - A_0 \bullet Z = \sum_{i=1}^{m} (A_i \bullet Z) x_i - A_0 \bullet Z = F(x) \bullet Z \geq 0$

Théorèmes de Dualité

Lemme

Soit A et B dans S_n. Si $A \succcurlyeq 0$ et $B \succcurlyeq 0$ alors $A \bullet B \geq 0$ et l'égalité est atteinte si et seulement si $AB = 0$

Dualité Faible

Le saut de dualité entre (DSP) et (SDP)
vaut :$c^T x - A_0 \bullet Z = \sum_{i=1}^{m} (A_i \bullet Z) x_i - A_0 \bullet Z = F(x) \bullet Z \geq 0$

Dualité Forte (qualification de type "Slater")

- Le problème primal est strictement réalisable (il existe des points intérieurs : $\exists x \;/\; F(x) \succ 0$).

Théorèmes de Dualité

Lemme

Soit A et B dans S_n. Si $A \succcurlyeq 0$ et $B \succcurlyeq 0$ alors $A \bullet B \geq 0$ et l'égalité est atteinte si et seulement si $AB = 0$

Dualité Faible

Le saut de dualité entre (DSP) et (SDP)
vaut :$c^T x - A_0 \bullet Z = \sum_{i=1}^{m} (A_i \bullet Z) x_i - A_0 \bullet Z = F(x) \bullet Z \geq 0$

Dualité Forte (qualification de type "Slater")

- Le problème primal est strictement réalisable
  (il existe des points intérieurs : $\exists x \,/\, F(x) \succ 0$).
- Le problème dual est strictement réalisable
  $(\exists Z \,/\, Z = Z^T \succ 0,\ A_i \bullet Z = c_i,\ i = 1, \ldots, m)$

Théorèmes de Dualité

Lemme

Soit A et B dans S_n. Si $A \succcurlyeq 0$ et $B \succcurlyeq 0$ alors $A \bullet B \geq 0$ et l'égalité est atteinte si et seulement si $AB = 0$

Dualité Faible

Le saut de dualité entre (DSP) et (SDP)
vaut :$c^T x - A_0 \bullet Z = \sum_{i=1}^{m} (A_i \bullet Z) x_i - A_0 \bullet Z = F(x) \bullet Z \geq 0$

Dualité Forte (qualification de type "Slater")

- Le problème primal est strictement réalisable
  (il existe des points intérieurs : $\exists x \,/\, F(x) \succ 0$).
- Le problème dual est strictement réalisable
  $(\exists Z \,/\, Z = Z^T \succ 0,\ A_i \bullet Z = c_i,\ i = 1, \ldots, m)$
- Si les deux conditions sont remplies alors il n'y a pas de saut de dualité et il existe des solutions optimales pour le primal et le dual

Conditions des écarts complémentaires en PSD

Proposition

Supposons que le saut de dualité est nul ($F(x) \bullet Z = 0$). x et Z sont respectivement optimaux pour (SDP) et $(DSDP)$ si et seulement si :

$$\begin{array}{l} F(x) \succeq 0 \\ Z \succeq 0,\ A_i \bullet Z = c_i,\ \ i = 1, \ldots, m \\ ZF(x) = 0 \ \Leftrightarrow\ F(x) \bullet Z = 0 \end{array}$$

Conditions des écarts complémentaires en PSD

Proposition

Supposons que le saut de dualité est nul ($F(x) \bullet Z = 0$). x et Z sont respectivement optimaux pour (SDP) et $(DSDP)$ si et seulement si :

$$\begin{array}{l} F(x) \succeq 0 \\ Z \succeq 0,\ A_i \bullet Z = c_i,\ \ i = 1, \ldots, m \\ ZF(x) = 0 \ \Leftrightarrow\ F(x) \bullet Z = 0 \end{array}$$

Conditions des écarts complémentaires en PSD

Proposition

Supposons que le saut de dualité est nul ($F(x) \bullet Z = 0$). x et Z sont respectivement optimaux pour (SDP) et $(DSDP)$ si et seulement si :

$$\begin{array}{l} F(x) \succeq 0 \\ Z \succeq 0,\ A_i \bullet Z = c_i,\ \ i = 1, \ldots, m \\ ZF(x) = 0 \ \Leftrightarrow\ F(x) \bullet Z = 0 \end{array}$$

Généralisation du cas linéaire :

- En programmation linéaire $F(x) = \mathbf{diag}(Ax - b)$ et Z sont diagonales

Conditions des écarts complémentaires en PSD

Proposition

Supposons que le saut de dualité est nul ($F(x) \bullet Z = 0$). x et Z sont respectivement optimaux pour (SDP) et $(DSDP)$ si et seulement si :

$$\begin{array}{l} F(x) \succeq 0 \\ Z \succeq 0,\ A_i \bullet Z = c_i,\ \ i = 1, \ldots, m \\ ZF(x) = 0 \ \Leftrightarrow\ F(x) \bullet Z = 0 \end{array}$$

Généralisation du cas linéaire :

- En programmation linéaire $F(x) = \mathbf{diag}(Ax - b)$ et Z sont diagonales
- $ZF(x)$ s'écrit $Z_{ii}\mathbf{diag}(Ax - b)_{ii} = Z_{ii}\left(A_i^T x - b_i\right) = 0 \forall i$

Exemple d'application

$$(SDP)\begin{cases} \min_{z\in\Re} & z \\ \text{s.c.} & \begin{bmatrix} 3z & -z-1 \\ -z-1 & 3z \end{bmatrix} \succeq 0 \end{cases}$$

1. Résoudre (SDP), i.e. donner sa valeur optimale ainsi que l'ensemble des solutions optimales.

Exemple d'application

$$(SDP)\left\{\begin{array}{ll} \min_{z\in\Re} & z \\ \text{s.c.} & \begin{bmatrix} 3z & -z-1 \\ -z-1 & 3z \end{bmatrix} \succeq 0 \end{array}\right.$$

1. Résoudre (SDP), i.e. donner sa valeur optimale ainsi que l'ensemble des solutions optimales.
2. Ecrire ($DSDP$), le programme dual de (SDP).

Exemple d'application

$$(SDP)\left\{\begin{array}{ll}\min_{z\in\Re} & z\\ \text{s.c.} & \begin{bmatrix} 3z & -z-1\\ -z-1 & 3z\end{bmatrix}\succeq 0\end{array}\right.$$

1. Résoudre (SDP), i.e. donner sa valeur optimale ainsi que l'ensemble des solutions optimales.
2. Ecrire $(DSDP)$, le programme dual de (SDP).
3. Montrer qu'il n'y a pas de saut de dualité entre $(DSDP)$ et (SDP), et résoudre $(DSDP)$ en utilisant les conditions des écarts complémentaires.

Plan du cours

1. Introduction

2. Linéarisations

3. Approches Lagrangiennes

4. Approche par Programmation Semidéfinie
- a. Bases de la Programmation Semidéfinie
- b. Application à la Programmation Quadratique en 0-1
  - 1. Le nombre de Lovász
  - 2. Relaxation SDP en $\{0, 1\}$
  - 3. Relaxation SDP en $\{-1, 1\}$
  - 4. Liens avec l'approche Lagrangienne
  - 5. Amélioration des relaxations Quadratiques Convexes
  - 6. Problèmes SDLS et contrainte sphérique

Plan du cours

1. 1. Introduction
2. 2. Linéarisations
3. 3. Approches Lagrangiennes
4. 4. Approche par Programmation Semidéfinie
   - a. Bases de la Programmation Semidéfinie
   - b. Application à la Programmation Quadratique en 0-1
     - 1. Le nombre de Lovász
     - 2. Relaxation SDP en $\{0, 1\}$
     - 3. Relaxation SDP en $\{-1, 1\}$
     - 4. Liens avec l'approche Lagrangienne
     - 5. Amélioration des relaxations Quadratiques Convexes
     - 6. Problèmes SDLS et contrainte sphérique

4.b.1. Le nombre de Lovász

Le problème du STABLE
Données. Un $G = (V, E)$ un graphe non orienté possédant n sommets.
Question. Trouver un ensemble S de sommets de cardinalité maximale tel que le sous-graphe induit par S ne possède pas d'arête (stable de G).

4.b.1. Le nombre de Lovász

Le problème du STABLE
Données. Un $G = (V, E)$ un graphe non orienté possédant n sommets.
Question. Trouver un ensemble S de sommets de cardinalité maximale tel que le sous-graphe induit par S ne possède pas d'arête (stable de G).

- Soit $\mathfrak{M}(G) = \left\{A \in S_n \,:\, A_{ij} = 1 \,\mathrm{si}\, (i,j) \notin E \,\mathrm{ou}\, (i = j)\right\}$

4.b.1. Le nombre de Lovász

Le problème du STABLE
Données. Un $G = (V, E)$ un graphe non orienté possédant n sommets.
Question. Trouver un ensemble S de sommets de cardinalité maximale tel que le sous-graphe induit par S ne possède pas d'arête (stable de G).

- Soit $\mathfrak{M}(G) = \{A \in S_n \; : \; A_{ij} = 1 \,\mathrm{si}\, (i,j) \notin E \,\mathrm{ou}\, (i = j)\}$
- $\mathfrak{M}(G)$ est un espace affine

4.b.1. Le nombre de Lovász

Le problème du STABLE
Données. Un $G = (V, E)$ un graphe non orienté possédant n sommets.
Question. Trouver un ensemble S de sommets de cardinalité maximale tel que le sous-graphe induit par S ne possède pas d'arête (stable de G).

- Soit $\mathfrak{M}(G) = \{A \in S_n \,:\, A_{ij} = 1 \,\mathrm{si}\, (i,j) \notin E \,\mathrm{ou}\, (i = j)\}$
- $\mathfrak{M}(G)$ est un espace affine
- Pour tout $[v_i,\ v_j] \in E$ on définit E^{ij} matrice de S_n telle que $E^{ij}_{ij} = E^{ij}_{ji} = 1$ et sinon $E^{ij}_{kl} = 0$

4.b.1. Le nombre de Lovász

Le problème du STABLE
Données. Un $G = (V, E)$ un graphe non orienté possédant n sommets.
Question. Trouver un ensemble S de sommets de cardinalité maximale tel que le sous-graphe induit par S ne possède pas d'arête (stable de G).

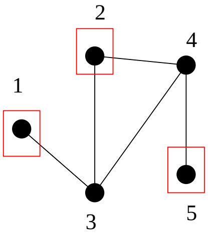
- Soit $\mathfrak{M}(G) = \{A \in S_n \,:\, A_{ij} = 1 \,\mathrm{si}\, (i, j) \notin E \,\mathrm{ou}\, (i = j)\}$
- $\mathfrak{M}(G)$ est un espace affine
- Pour tout $[v_i,\ v_j] \in E$ on définit E^{ij} matrice de S_n telle que $E^{ij}_{ij} = E^{ij}_{ji} = 1$ et sinon $E^{ij}_{kl} = 0$
- Repère de l'espace affine $\mathfrak{M}(G)$: origine J_n (matrice de "1"), base $\{\ldots, E_{ij}, \ldots\}$

Le problème du STABLE

Données. Un $G = (V, E)$ un graphe non orienté possédant n sommets.
Question. Trouver un ensemble S de sommets de cardinalité maximale tel que le sous-graphe induit par S ne possède pas d'arête (stable de G).

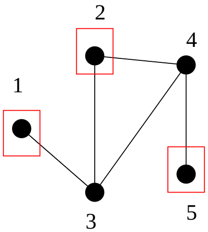
- Soit $\mathfrak{M}(G) = \{A \in S_n \,:\, A_{ij} = 1 \,\mathrm{si}\, (i, j) \notin E \,\mathrm{ou}\, (i = j)\}$
- $\mathfrak{M}(G)$ est un espace affine
- Pour tout $[v_i,\ v_j] \in E$ on définit E^{ij} matrice de S_n telle que $E^{ij}_{ij} = E^{ij}_{ji} = 1$ et sinon $E^{ij}_{kl} = 0$
- Repère de l'espace affine $\mathfrak{M}(G)$: origine J_n (matrice de "1"), base $\{\ldots, E_{ij}, \ldots\}$
- $A \in \mathfrak{M}(G)$ si et seulement si $A = J_n + \sum_{(i,j)\in E} x_{ij} E^{ij}$

Exemple :



$$\begin{bmatrix} \color{red}{\mathbf{1}} & \color{red}{\mathbf{1}} & -5 & 1 & \color{red}{\mathbf{1}} \\ \color{red}{\mathbf{1}} & \color{red}{\mathbf{1}} & 0 & 7 & \color{red}{\mathbf{1}} \\ -5 & 0 & 1 & 6 & \color{blue}{1} \\ 1 & 7 & 6 & 1 & -1 \\ \color{red}{\mathbf{1}} & \color{red}{\mathbf{1}} & \color{blue}{1} & -1 & \color{red}{\mathbf{1}} \end{bmatrix}$$

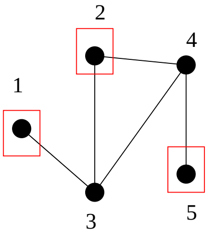
Exemple :



$$\begin{bmatrix} \mathbf{1} & \mathbf{1} & -5 & 1 & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & 0 & 7 & \mathbf{1} \\ -5 & 0 & 1 & 6 & 1 \\ 1 & 7 & 6 & 1 & -1 \\ \mathbf{1} & \mathbf{1} & 1 & -1 & \mathbf{1} \end{bmatrix}$$

1. $\exists$ stable de taille k dans G si et seulement si toute matrice A dans $\mathfrak{M}(G)$ contient la sous-matrice $J_k \in S_k$, matrice $k \times k$ uniquement constituée de 1

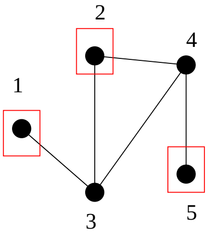
Exemple :



$$\begin{bmatrix} \mathbf{1} & \mathbf{1} & -5 & 1 & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & 0 & 7 & \mathbf{1} \\ -5 & 0 & 1 & 6 & 1 \\ 1 & 7 & 6 & 1 & -1 \\ \mathbf{1} & \mathbf{1} & 1 & -1 & \mathbf{1} \end{bmatrix}$$

1. $\exists$ stable de taille k dans G si et seulement si toute matrice A dans $\mathfrak{M}(G)$ contient la sous-matrice $J_k \in S_k$, matrice $k \times k$ uniquement constituée de 1
2. Les valeurs propres de J_k sont 0 et k (rang(J_k)$=$ 1)

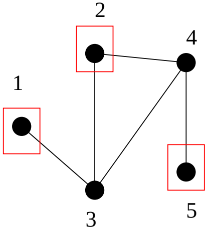
Exemple :



$$\begin{bmatrix} \mathbf{1} & \mathbf{1} & -5 & 1 & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & 0 & 7 & \mathbf{1} \\ -5 & 0 & 1 & 6 & 1 \\ 1 & 7 & 6 & 1 & -1 \\ \mathbf{1} & \mathbf{1} & 1 & -1 & \mathbf{1} \end{bmatrix}$$

1. $\exists$ stable de taille k dans G si et seulement si toute matrice A dans $\mathfrak{M}(G)$ contient la sous-matrice $J_k \in S_k$, matrice $k \times k$ uniquement constituée de 1
2. Les valeurs propres de J_k sont 0 et k (rang$(J_k)= 1$)
3. Soit $A \in \mathfrak{M}(G)$, on a $\lambda_{max}(A) \geq \lambda_{max}(J_k) = k$.

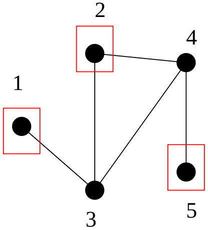
Exemple :



$$\begin{bmatrix} 1 & 1 & -5 & 1 & 1 \\ 1 & 1 & 0 & 7 & 1 \\ -5 & 0 & 1 & 6 & 1 \\ 1 & 7 & 6 & 1 & -1 \\ 1 & 1 & 1 & -1 & 1 \end{bmatrix}$$

1. $\exists$ stable de taille k dans G si et seulement si toute matrice A dans $\mathfrak{M}(G)$ contient la sous-matrice $J_k \in S_k$, matrice $k \times k$ uniquement constituée de 1
2. Les valeurs propres de J_k sont 0 et k (rang(J_k)$=$ 1)
3. Soit $A \in \mathfrak{M}(G)$, on a $\lambda_{max}(A) \geq \lambda_{max}(J_k) = k$.
4. Donc pour tout A dans $\mathfrak{M}(G)$ $\lambda_{max}(A) \geq \alpha(G)$ (nombre de stabilité : taille d'un plus grand stable dans G)

Exemple :



$$\begin{bmatrix} 1 & 1 & -5 & 1 & 1 \\ 1 & 1 & 0 & 7 & 1 \\ -5 & 0 & 1 & 6 & 1 \\ 1 & 7 & 6 & 1 & -1 \\ 1 & 1 & 1 & -1 & 1 \end{bmatrix}$$

1. $\exists$ stable de taille k dans G si et seulement si toute matrice A dans $\mathfrak{M}(G)$ contient la sous-matrice $J_k \in S_k$, matrice $k \times k$ uniquement constituée de 1
2. Les valeurs propres de J_k sont 0 et k (rang(J_k)$= 1$)
3. Soit $A \in \mathfrak{M}(G)$, on a $\lambda_{max}(A) \geq \lambda_{max}(J_k) = k$.
4. Donc pour tout A dans $\mathfrak{M}(G)$ $\lambda_{max}(A) \geq \alpha(G)$ (nombre de stabilité : taille d'un plus grand stable dans G)
5. Meilleure borne : $\min_{A \in \mathfrak{M}(G)} \lambda_{max}(A)$

Formulation du problème comme PSD

On a $\min\{\lambda : \lambda I - A \succeq 0\} = \lambda_{max}(A)$

Formulation du problème comme PSD

On a $\min\{\lambda : \lambda I - A \succeq 0\} = \lambda_{max}(A)$

- $\lambda I - A = \lambda I - MDM^T = M(\lambda I - D)M^T$ avec $D = \begin{bmatrix} \lambda_1 & \dots & 0 \\ 0 & \lambda_i & 0 \\ 0 & \dots & \lambda_n \end{bmatrix}$

Formulation du problème comme PSD

On a $\min\{\lambda : \lambda I - A \succeq 0\} = \lambda_{max}(A)$

- $\lambda I - A = \lambda I - MDM^T = M(\lambda I - D)M^T$ avec $D = \begin{bmatrix} \lambda_1 & \dots & 0 \\ 0 & \lambda_i & 0 \\ 0 & \dots & \lambda_n \end{bmatrix}$
- $\lambda I - D \succeq 0 \Leftrightarrow \lambda \geq \lambda_{max}(A)$

Formulation du problème comme PSD

On a $\min\{\lambda \,:\, \lambda I - A \succeq 0\} = \lambda_{max}(A)$

- $\lambda I - A = \lambda I - MDM^T = M(\lambda I - D)M^T$ avec $D = \begin{bmatrix} \lambda_1 & \dots & 0 \\ 0 & \lambda_i & 0 \\ 0 & \dots & \lambda_n \end{bmatrix}$
- $\lambda I - D \succeq 0 \Leftrightarrow \lambda \geq \lambda_{max}(A)$

Formulation du problème comme PSD

On a $\min\{\lambda \,:\, \lambda I - A \succeq 0\} = \lambda_{max}(A)$

- $\lambda I - A = \lambda I - MDM^T = M(\lambda I - D)M^T$ avec $D = \begin{bmatrix} \lambda_1 & \dots & 0 \\ 0 & \lambda_i & 0 \\ 0 & \dots & \lambda_n \end{bmatrix}$
- $\lambda I - D \succeq 0 \Leftrightarrow \lambda \geq \lambda_{max}(A)$

$\min_{A \in \mathfrak{M}(G)} \lambda_{max}(A)$ est donc équivalent à :

$$\theta(G) \begin{cases} \min \ \lambda \\ \text{s.c :} \ \lambda I_n - \underbrace{\left(J_n + \sum_{(i,j) \in E} x_{ij} E^{ij}\right)}_{A \in \mathfrak{M}(G)} \succeq 0 \end{cases}$$

Dual Semidéfini

$$\theta(G)\left\{\begin{array}{ll} \min & \lambda \\ \text{s.c :} & \lambda I_n - \underbrace{\left(J_n + \sum_{(i,j)\in E} x_{ij}E^{ij}\right)}_{A\in\mathfrak{M}(G)} \succeq 0 \end{array}\right.$$

Dual Semidéfini

$$\theta(G) \begin{cases} \min & \lambda \\ \text{s.c :} & \lambda I_n - \underbrace{\left(J_n + \sum_{(i,j)\in E} x_{ij} E^{ij} \right)}_{A \in \mathfrak{M}(G)} \succeq 0 \end{cases}$$

dual :

Dual Semidéfini

$$\theta(G)\begin{cases} \min & \lambda \\ \text{s.c}: & \lambda I_n - \underbrace{\left(J_n + \sum_{(i,j)\in E} x_{ij}E^{ij}\right)}_{A\in\mathfrak{M}(G)} \succeq 0 \end{cases}$$

dual :

$$D\theta(G)\begin{cases} \max & J_n \bullet Z = \sum_{i=1}^n \sum_{i=1}^n Z_{ij} \\ \text{s.c.}: & E^{ij} \bullet Z = Z_{ij} = 0 \qquad \forall (i,j)\in E \\ & I_n \bullet Z = Trace(Z) = 1 \\ & Z \succeq 0 \end{cases}$$

Dual Semidéfini

$$\theta(G) \left\{ \begin{array}{ll} \min & \lambda \\ \text{s.c :} & \lambda I_n - \underbrace{\left(J_n + \sum_{(i,j)\in E} x_{ij} E^{ij} \right)}_{A \in \mathfrak{M}(G)} \succeq 0 \\ & \end{array} \right.$$

dual :

$$D\theta(G) \left\{ \begin{array}{lll} \max & J_n \bullet Z = \sum_{i=1}^{n} \sum_{i=1}^{n} Z_{ij} & \\ \text{s.c. :} & E^{ij} \bullet Z = Z_{ij} = 0 & \forall (i,j) \in E \\ & I_n \bullet Z = Trace(Z) = 1 & \\ & Z \succeq 0 & \end{array} \right.$$

admet des points intérieurs : I_n/n est strictement admissible. Donc pas de saut de dualité

Application dans les Graphes Parfaits

- Dans un graphe parfait G, le nombre de stabilité (taille d'un plus grand stable) est égal au nombre chromatique du complémentaire de G par les arêtes : $\alpha(G) = \chi(\bar{G})$

Application dans les Graphes Parfaits

- Dans un graphe parfait G, le nombre de stabilité (taille d'un plus grand stable) est égal au nombre chromatique du complémentaire de G par les arêtes : $\alpha(G) = \chi(\bar{G})$
- $\chi(\bar{G})$: nombre minimal de couleurs pour colorier les sommets de $\bar{G}$ (deux sommets adjacents n'ont pas la même couleur)

Application dans les Graphes Parfaits

- Dans un graphe parfait G, le nombre de stabilité (taille d'un plus grand stable) est égal au nombre chromatique du complémentaire de G par les arêtes : $\alpha(G) = \chi(\bar{G})$
- $\chi(\bar{G})$: nombre minimal de couleurs pour colorier les sommets de $\bar{G}$ (deux sommets adjacents n'ont pas la même couleur)

Théorème

Pour tout graphe G on a $\alpha(G) \leq \theta(G) \leq \chi(\bar{G})$

Application dans les Graphes Parfaits

- Dans un graphe parfait G, le nombre de stabilité (taille d'un plus grand stable) est égal au nombre chromatique du complémentaire de G par les arêtes : $\alpha(G) = \chi(\bar{G})$
- $\chi(\bar{G})$: nombre minimal de couleurs pour colorier les sommets de $\bar{G}$ (deux sommets adjacents n'ont pas la même couleur)

Théorème

Pour tout graphe G on a $\alpha(G) \leq \theta(G) \leq \chi(\bar{G})$

- On a déjà vu que $\alpha(G) \leq \theta(G) = D\theta(G)$

Application dans les Graphes Parfaits

- Dans un graphe parfait G, le nombre de stabilité (taille d'un plus grand stable) est égal au nombre chromatique du complémentaire de G par les arêtes : $\alpha(G) = \chi(\bar{G})$
- $\chi(\bar{G})$: nombre minimal de couleurs pour colorier les sommets de $\bar{G}$ (deux sommets adjacents n'ont pas la même couleur)

Théorème

Pour tout graphe G on a $\alpha(G) \leq \theta(G) \leq \chi(\bar{G})$

- On a déjà vu que $\alpha(G) \leq \theta(G) = D\theta(G)$
- Donc si on montre que $D\theta(G) \leq \chi(\bar{G})$, on obtiendra $\alpha(G)$ dans les graphes parfaits !

Application dans les Graphes Parfaits

- Dans un graphe parfait G, le nombre de stabilité (taille d'un plus grand stable) est égal au nombre chromatique du complémentaire de G par les arêtes : $\alpha(G) = \chi(\bar{G})$
- $\chi(\bar{G})$: nombre minimal de couleurs pour colorier les sommets de $\bar{G}$ (deux sommets adjacents n'ont pas la même couleur)

Théorème

Pour tout graphe G on a $\alpha(G) \leq \theta(G) \leq \chi(\bar{G})$

- On a déjà vu que $\alpha(G) \leq \theta(G) = D\theta(G)$
- Donc si on montre que $D\theta(G) \leq \chi(\bar{G})$, on obtiendra $\alpha(G)$ dans les graphes parfaits !
- Et même $\chi(G)$ car le complément d'un graphe parfait est parfait ("weak perfect graph conjecture", Lovász)

$$D\theta(G)\left\{\begin{array}{lll} \max & J_n \bullet Z = \sum_{i=1}^{n}\sum_{i=1}^{n} Z_{ij} & \\ \text{s.c. :} & Z_{ij} = 0 & \forall (i,j) \in E \\ & I_n \bullet Z = Trace(Z) = 1 & \\ & Z \succeq 0 & \end{array}\right.$$

- Soit une partition $C_1, \ldots, C_k$ de G en k cliques (i.e. une coloration de $\bar{G}$), et Z admissible pour $D\theta(G)$

$$
D\theta(G)\left\{\begin{array}{lll}
\max & J_n \bullet Z = \sum_{i=1}^{n}\sum_{i=1}^{n} Z_{ij} & \\
\text{s.c. :} & Z_{ij} = 0 & \forall (i,j) \in E \\
 & I_n \bullet Z = Trace(Z) = 1 & \\
 & Z \succeq 0 &
\end{array}\right.
$$

- Soit une partition $C_1, \ldots, C_k$ de G en k cliques (i.e. une coloration de $\bar{G}$), et Z admissible pour $D\theta(G)$
- Soit v^i tel que $v^i_j = 1 \Leftrightarrow$ le sommet j est dans C_i
  donc $v^k_j = 0 \ \forall k \neq i$

$$D\theta(G)\begin{cases} \max & J_n \bullet Z = \sum_{i=1}^n \sum_{i=1}^n Z_{ij} \\ \text{s.c. :} & Z_{ij} = 0 \qquad\qquad\qquad\qquad \forall (i,j) \in E \\ & I_n \bullet Z = Trace(Z) = 1 \\ & Z \succeq 0 \end{cases}$$

- Soit une partition $C_1, \ldots, C_k$ de G en k cliques (i.e. une coloration de $\bar{G}$), et Z admissible pour $D\theta(G)$
- Soit v^i tel que $v^i_j = 1 \Leftrightarrow$ le sommet j est dans C_i
  donc $v^k_j = 0 \; \forall k \neq i$
- $Z \succcurlyeq 0$ donc $\sum_{i=1}^k (kv^i - e_n)^T Z (kv^i - e_n) \geq 0$, où $e_n = (1, \ldots, 1)^T$

$$D\theta(G)\left\{\begin{array}{lll} \max & J_n \bullet Z = \sum_{i=1}^n \sum_{i=1}^n Z_{ij} & \\ \text{s.c. :} & Z_{ij} = 0 & \forall (i,j) \in E \\ & I_n \bullet Z = Trace(Z) = 1 & \\ & Z \succeq 0 & \end{array}\right.$$

- Soit une partition $C_1, \ldots, C_k$ de G en k cliques (i.e. une coloration de $\bar{G}$), et Z admissible pour $D\theta(G)$
- Soit v^i tel que $v^i_j = 1 \Leftrightarrow$ le sommet j est dans C_i
  donc $v^k_j = 0 \ \forall k \neq i$
- $Z \succcurlyeq 0$ donc $\sum_{i=1}^k (kv^i - e_n)^T Z(kv^i - e_n) \geq 0$, où $e_n = (1, \ldots, 1)^T$
- $0 \leq k\left(\sum_{i=1}^k v^i(v^i)^T\right) \bullet Z - 2\left(\sum_{i=1}^k v^i\right)^T Ze_n + J_n \bullet Z$

$$D\theta(G)\left\{\begin{array}{lll} \max & J_n \bullet Z = \sum_{i=1}^n \sum_{i=1}^n Z_{ij} & \\ \text{s.c. :} & Z_{ij} = 0 & \forall (i,j) \in E \\ & I_n \bullet Z = Trace(Z) = 1 & \\ & Z \succeq 0 & \end{array}\right.$$

- Soit une partition $C_1, \ldots, C_k$ de G en k cliques (i.e. une coloration de $\bar{G}$), et Z admissible pour $D\theta(G)$
- Soit v^i tel que $v^i_j = 1 \Leftrightarrow$ le sommet j est dans C_i donc $v^k_j = 0\ \forall k \neq i$
- $Z \succcurlyeq 0$ donc $\sum_{i=1}^k (kv^i - e_n)^T Z(kv^i - e_n) \geq 0$, où $e_n = (1, \ldots, 1)^T$
- $0 \leq k\left(\sum_{i=1}^k v^i(v^i)^T\right) \bullet Z - 2\left(\sum_{i=1}^k v^i\right)^T Ze_n + J_n \bullet Z$
- Or $\left(\sum_{i=1}^k v^i\right)^T Ze_n = e_n e_n^T \bullet Z$ (partition)

$$D\theta(G)\begin{cases} \max & J_n \bullet Z = \sum_{i=1}^n \sum_{i=1}^n Z_{ij} & \\ \text{s.c. :} & Z_{ij} = 0 & \forall (i,j) \in E \\ & I_n \bullet Z = Trace(Z) = 1 & \\ & Z \succeq 0 & \end{cases}$$

- Soit une partition $C_1, \ldots, C_k$ de G en k cliques (i.e. une coloration de $\bar{G}$), et Z admissible pour $D\theta(G)$
- Soit v^i tel que $v^i_j = 1 \Leftrightarrow$ le sommet j est dans C_i
  donc $v^k_j = 0 \; \forall k \neq i$
- $Z \succcurlyeq 0$ donc $\sum_{i=1}^k (kv^i - e_n)^T Z (kv^i - e_n) \geq 0$, où $e_n = (1, \ldots, 1)^T$
- $0 \leq k \left(\sum_{i=1}^k v^i (v^i)^T\right) \bullet Z - 2 \left(\sum_{i=1}^k v^i\right)^T Z e_n + J_n \bullet Z$
- Or $\left(\sum_{i=1}^k v^i\right)^T Z e_n = e_n e_n^T \bullet Z$ (partition)
- Donc $0 \leq k I_n \bullet Z - J_n \bullet Z$ donc $J_n \bullet Z \leq \chi(\bar{G})$

Plan du cours

1. 1. Introduction
2. 2. Linéarisations
3. 3. Approches Lagrangiennes
4. 4. Approche par Programmation Semidéfinie
   - a. Bases de la Programmation Semidéfinie
   - b. Application à la Programmation Quadratique en 0-1
     - 1. Le nombre de Lovász
     - 2. Relaxation SDP en $\{0, 1\}$
     - 3. Relaxation SDP en $\{-1, 1\}$
     - 4. Liens avec l'approche Lagrangienne
     - 5. Amélioration des relaxations Quadratiques Convexes
     - 6. Problèmes SDLS et contrainte sphérique

Problème Quadratique 0-1 sans autre contrainte

$$(QP)_{\{0,1\}} \begin{cases} \max f(x) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j\neq i} C_{ij}x_i x_j + \sum_{i=1}^{n} b_i x_i \\ \qquad\quad = \frac{1}{2} C \bullet xx^T + b^T x \\ \text{s.}c. \quad \mathrm{x}_i \in \{0,1\} \; ; \; i = 1, ..., n \end{cases}$$

Problème Quadratique 0-1 sans autre contrainte

$$
(QP)_{\{0,1\}} \begin{cases} \max f(x) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j\neq i} C_{ij}x_ix_j + \sum_{i=1}^{n} b_ix_i \\ \qquad\quad = \frac{1}{2} C \bullet xx^T + b^T x \\ \text{s.}c. \quad \mathrm{x}_i \in \{0,1\} \ ; \ i = 1, ..., n \end{cases}
$$

Relaxation "Classique" par programmation linéaire :

$$
(LP)_{\{0,1\}} \begin{cases} \max \frac{1}{2} C \bullet X + b^T x \\ \text{s.}c. \quad 0 \leq x_i \leq 1 \ \ i \in \{1, ..., n\} \\ \qquad\quad 0 \leq X_{ij} \leq x_i \ \ i \neq j \in \{1, ..., n\} \\ \qquad\quad 0 \leq X_{ij} \leq x_j \ \ i \neq j \in \{1, ..., n\} \\ \qquad\quad x_i + x_j \leq 1 + X_{ij} \ \ i \neq j \in \{1, ..., n\} \end{cases}
$$

4.b.2. Relaxation SDP en $\{0,1\}$

Rappel. $(X, x) \in S_n \times \mathbb{R}^n$

on a $\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0 \Leftrightarrow X - xx^T \succeq 0.$

RELAXATION

$X = xx^T \quad \Longrightarrow$	$X \succeq xx^T$
$x_i \in \{0,1\} \quad \Longrightarrow$	x_i réel
	$\mathbf{d}(X) = x$

$\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0 => det \begin{bmatrix} 1 & x_i \\ x_i & x_i \end{bmatrix} \geq 0 \; \forall i$

i.e. $1 \geq x_i \geq x_i^2 \geq 0$

$$(SDP)_{QP\{0,1\}} \begin{cases} \max \frac{1}{2} C \bullet X + b^T x \\ \text{s.c.} \quad \mathbf{d}(X) = x \\ \qquad \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0 \end{cases}$$

Plan du cours

1. 1. Introduction
2. 2. Linéarisations
3. 3. Approches Lagrangiennes
4. 4. Approche par Programmation Semidéfinie
   - a. Bases de la Programmation Semidéfinie
   - b. Application à la Programmation Quadratique en 0-1
     - 1. Le nombre de Lovász
     - 2. Relaxation SDP en $\{0, 1\}$
     - 3. Relaxation SDP en $\{-1, 1\}$
     - 4. Liens avec l'approche Lagrangienne
     - 5. Amélioration des relaxations Quadratiques Convexes
     - 6. Problèmes SDLS et contrainte sphérique

4.b.3. Relaxation SDP en $\{-1, 1\}$

Démarche géométrique

$x_i = \frac{1+y_i}{2}$ et $y_i = 2x_i - 1$ $(i \in \{1, ..., n, \})$

$$(QP)_{\{-1,1\}} \begin{cases} \text{Max } f(y) = \frac{1}{8} \sum_{i=1}^{n} \sum_{j \neq i} C_{ij} (1 + y_i) \left(1 + y_j\right) \\ \quad + \frac{1}{2} \sum_{i=1}^{n} b_i (1 + y_i) \\ \quad = \frac{1}{8} \left[C \bullet yy^T + \sum_{i=1}^{n} \left(4b_i + 2 \sum_{j \neq i} C_{ij}\right) y_i \right] \\ \quad + \left(\frac{1}{4} C_{tot} + \frac{1}{2} \sum_{i=1}^{n} b_i\right) \\ \text{s.c.} \quad \mathrm{y}_i \in \{-1, 1\} \ ; \ i = 1, ..., n \end{cases}$$

Relaxation sur la dimension

RELAXATION

$y_i \in \{-1, 1\} \quad \Longrightarrow$	$v_i \in S^n$
$y_i y_j \quad \Longrightarrow$	$v_i^T v_j$
y_i	$v_i^T v_0$

Relaxation SDP en $\{-1, 1\}$

$$(P)_{QP\{-1,1\}} \left\{ \begin{array}{l} \max F(v_0, ..., v_n) = \frac{1}{8}\sum_{i=1}^n \sum_{j\neq i} C_{ij} v_i^T v_j \\ \qquad\qquad + \sum_{i=1}^n \left(4b_i + 2\sum_{j\neq i} C_{ij}\right) v_i^T v_0 \\ \qquad\qquad + \left(\frac{1}{4}C_{tot} + \frac{1}{2}\sum_{i=1}^n b_i\right) \\ \text{s.c. } : \; v_i \in S^n \; i = 0, ..., n \end{array} \right. \tag{1}$$

$Y = VV^T$ avec $Y_{ii} = 1$; $i = 0, ..., n$
<=> $v_i \in S^n \; i = 0, ..., n$ (Cholesky)

$$\Leftrightarrow \; (SDP)_{QP\{-1,1\}} \left\{ \begin{array}{l} \text{Max } \frac{1}{8} C' \bullet Y + \left(\frac{1}{4}C_{tot} + \frac{1}{2}\sum_{i=1}^n b_i\right) \\ \text{s.c. } \quad \mathbf{d}(Y) = e_{n+1} \\ \qquad\quad Y \in S_{n+1}^+ \end{array} \right.$$

où $C' = \begin{bmatrix} 0 & b'^T \\ b' & C \end{bmatrix}$
et $b' = 2b + \left(\sum_{j=1}^n C_{1j}, \cdots, \sum_{j=1}^n C_{ij}, \cdots, \sum_{j=1}^n C_{nj}\right)$
Question : les deux SDPs obtenus sont-ils équivalents ?

Equivalence des relaxations SDP

Q inversible, $M \in S_n$ on pose $\phi(M) = QMQ^T$. ϕ laisse stable S_n^+.
Changement de variable $W = QXQ^T$

$$(SDP) \begin{cases} \text{Max } C \bullet X \\ \text{s.c.} \quad A_i \bullet X = (\, ou \ \leq) b_i \,;\, i = 1, ..., k \\ \qquad\quad X \succeq 0 \end{cases}$$

$\widetilde{C} = \left(Q^{-1}\right)^T CQ^{-1}$, et $\widetilde{A}_i = \left(Q^{-1}\right)^T A_i Q^{-1}$ pour $i \in \{1, ..., k\}$

$$(SDP)_Q \begin{cases} \max \widetilde{C} \bullet W \\ \text{s.c.} \quad \widetilde{A}_i \bullet W = (ou \ \leq)\, b_i \,;\, i = 1, ..., k \\ \qquad\quad W \succeq 0 \end{cases}$$

Lemme. *X est une solution admissible de (SDP) si et seulement si $W = QXQ^T$ est une solution admissible de $(SDP)_Q$. De plus, on a $C \bullet X = \widetilde{C} \bullet W$*

Application aux cas $\{0,1\}$ et $\{-1,1\}$

Initialement : $x = \frac{1}{2}(y + e_n)$

i.e. $\begin{pmatrix} 1 \\ x \end{pmatrix} = Q \begin{pmatrix} 1 \\ y \end{pmatrix}$ avec $Q = \begin{bmatrix} 1 & 0 \\ \frac{1}{2}e & \frac{1}{2}I_n \end{bmatrix}$

On a $Q^{-1} = \begin{bmatrix} 1 & 0 \\ -e & 2I_n \end{bmatrix}$

SDP $\{-1,1\}$	SDP $\{0,1\}$
$Y_{ii} = E_i \bullet Y = 1 \; \forall i$	$\left(Q^{-1}\right)^T E_i Q^{-1} \Rightarrow \; X_{ii} = X_{0i}$
$Y \succeq 0$	$X \succeq 0$

$\left(Q^{-1}\right)^T E_i Q^{-1} = \begin{bmatrix} 1 & -e_n \\ 0 & 2I_n \end{bmatrix} E_i \begin{bmatrix} 1 & 0 \\ -e_n & 2I_n \end{bmatrix}$

implique $X_{00} - 4X_{0i} + 4X_{ii} = 1$

i.e. $X_{ii} = X_{0i} \; \forall i \in \{1, ..., n\} \; (X_{00} = 1)$

Calcul de la fonction Objectif en $\{0, 1\}$

En $\{-1, 1\}$ on a $f(Y) = \left(\frac{1}{4} C_{tot} + \frac{1}{2} \sum_{i=1}^{n} b_i\right) + \frac{1}{8} C' \bullet Y$

où $C' = \begin{bmatrix} 0 & b'^T \\ b' & C \end{bmatrix}$

et $b' = 2b + \left(\sum_{j=1}^{n} C_{1j}, \cdots, \sum_{j=1}^{n} C_{ij}, \cdots, \sum_{j=1}^{n} C_{nj}\right)$

En $\{0, 1\}$: $\left(Q^{-1}\right)^T C' Q^{-1} = \begin{bmatrix} -4 \sum_{i=1}^{n} b_i - 2C_{tot} & 4b^T \\ 4b & 4C \end{bmatrix}$

$$\begin{aligned}
g(X) &= \frac{1}{2} \sum_{i=1}^{n} b_i + \frac{1}{4} C_{tot} + \frac{1}{8} \phi^{-1}(C') \bullet X \\
&= \frac{1}{2} \sum_{i=1}^{n} b_i + \frac{1}{4} C_{tot} \\
&\quad + \frac{1}{8} \left(-4 \sum_{i=1}^{n} b_i - 2C_{tot} + 4 \sum_{i=1}^{n} \sum_{j \neq i} C_{ij} X_{ij} + 8b^T x \right) \\
&= \frac{1}{2} C \bullet X + b^T x
\end{aligned}$$

Améliorations semidéfinies de l'approche par PL

$$(SDP_2)_{\{0,1\}} \left\{ \begin{array}{l} \text{Max } \frac{1}{2} C \bullet X + b^T x \\ \text{s.}c. \quad \mathbf{d}(X) = x \\ \qquad \begin{bmatrix} 1 & x \\ x^T & X \end{bmatrix} \succeq 0 \\ \qquad 0 \leq X_{ij} \ i \neq j \in \{1, ..., n\} \\ \qquad X_{ij} \leq X_{ii} \ i \neq j \in \{1, ..., n\} \\ \qquad X_{ij} \leq X_{jj} \ i \neq j \in \{1, ..., n\} \\ \qquad X_{ii} + X_{jj} \leq 1 + X_{ij} \ i \neq j \in \{1, ..., n\} \end{array} \right.$$

Est équivalent à

$$(SDP_2)_{\{-1,1\}} \left\{ \begin{array}{l} \text{Max } \frac{1}{8} C' \bullet Y + \left(\frac{1}{4} C_{tot} + \frac{1}{2} \sum_{i=1}^{n} b_i \right) \\ \text{s.c. } Y_{0i} + Y_{0j} + Y_{ij} \geq -1 \ i \neq j \in \{1, ..., n\} \\ \quad Y_{ij} - Y_{i0} - Y_{j0} \geq -1 \ i \neq j \in \{1, ..., n\} \\ \quad Y_{i0} - Y_{j0} - Y_{ij} \geq -1 \ i \neq j \in \{1, ..., n\} \\ \quad -Y_{i0} + Y_{j0} - Y_{ij} \geq -1 \ i \neq j \in \{1, ..., n\} \\ \quad \mathbf{d}(Y) = e_{n+1} \\ \quad Y \in S_{n+1}^{+} \end{array} \right.$$

Application au problème du stable

Modèle par programme linéaire en variables 0-1 :

$$\left\{\begin{array}{lll} \max & \sum_{i=1}^{n} x_i & \\ \text{s.c. :} & (0 \leq) x_i + x_j \leq 1 & \forall (i,j) \in E \\ & x_i \in \{0,1\} & \forall i \in \{1,\ldots,n\} \end{array}\right.$$

- $\left(x_i + x_j - 1\right)\left(x_i + x_j\right) \leq 0 \; \forall (i,j) \in E$

Application au problème du stable

Modèle par programme linéaire en variables 0-1 :

$$\left\{\begin{array}{lll} \max & \sum_{i=1}^{n} x_i & \\ \text{s.c. :} & (0 \leq) x_i + x_j \leq 1 & \forall (i,j) \in E \\ & x_i \in \{0,1\} & \forall i \in \{1,\ldots,n\} \end{array}\right.$$

- $\left(x_i + x_j - 1\right)\left(x_i + x_j\right) \leq 0 \ \forall (i,j) \in E$
- Donc $0 \leq x_i x_j \leq 0$! => $X_{ij} = 0 \ \forall (i,j) \in E$

Application au problème du stable

Modèle par programme linéaire en variables 0-1 :

$$\begin{cases} \max & \sum_{i=1}^{n} x_i & \\ \text{s.c. :} & (0 \leq) x_i + x_j \leq 1 & \forall (i,j) \in E \\ & x_i \in \{0,1\} & \forall i \in \{1, \ldots, n\} \end{cases}$$

- $\left(x_i + x_j - 1\right)\left(x_i + x_j\right) \leq 0 \ \forall (i,j) \in E$
- Donc $0 \leq x_i x_j \leq 0$! => $X_{ij} = 0 \ \forall (i,j) \in E$

Application au problème du stable

Modèle par programme linéaire en variables 0-1 :

$$\begin{cases} \max & \sum_{i=1}^{n} x_i \\ \text{s.c. :} & (0 \leq) x_i + x_j \leq 1 \quad \forall (i,j) \in E \\ & x_i \in \{0,1\} \quad \forall i \in \{1,\ldots,n\} \end{cases}$$

- $\left(x_i + x_j - 1\right)\left(x_i + x_j\right) \leq 0 \; \forall (i,j) \in E$
- Donc $0 \leq x_i x_j \leq 0$! => $X_{ij} = 0 \; \forall (i,j) \in E$

Autre formulation semidéfinie du nombre de Lovász :

$$\theta_G = \begin{cases} \max & \sum_{i=1}^{n} x_i \\ \text{s.c. :} & X_{ij} = 0 \quad \forall (i,j) \in E \\ & X_{ii} = x_i \quad i = 1,\ldots,n \\ & X - xx^T \succcurlyeq 0 \end{cases}$$

Etude de cas

Problème du sac-à-dos (knapsack) quadratique (déjà étudié par l'approche linéaire)

$$(\text{KQP}) \begin{cases} \min & x^T Q x + c^T x \\ \text{s.c.} & a^T x \leq b \\ & x \in \{0,1\}^n \end{cases}$$

1. Relaxation semidéfinie "basique"

Etude de cas

Problème du sac-à-dos (knapsack) quadratique (déjà étudié par l'approche linéaire)

$$(\mathrm{KQP}) \begin{cases} \min & x^T Q x + c^T x \\ \text{s.c.} & a^T x \leq b \\ & x \in \{0,1\}^n \end{cases}$$

1. Relaxation semidéfinie "basique"
2. Premier traitement de la contrainte d'inégalité

Etude de cas

Problème du sac-à-dos (knapsack) quadratique (déjà étudié par l'approche linéaire)

$$
\text{(KQP)} \quad \begin{cases} \min & x^T Q x + c^T x \\ \text{s.c.} & a^T x \leq b \\ & x \in \{0,1\}^n \end{cases}
$$

1. Relaxation semidéfinie "basique"
2. Premier traitement de la contrainte d'inégalité
3. Deuxième traitement de la contrainte d'inégalité

Etude de cas

Problème du sac-à-dos (knapsack) quadratique (déjà étudié par l'approche linéaire)

$$(\text{KQP}) \left\{ \begin{array}{ll} \min & x^T Q x + c^T x \\ \text{s.c.} & a^T x \leq b \\ & x \in \{0,1\}^n \end{array} \right.$$

1. Relaxation semidéfinie "basique"
2. Premier traitement de la contrainte d'inégalité
3. Deuxième traitement de la contrainte d'inégalité
4. Comparaison des approches

Plan du cours

1. Introduction

2. Linéarisations

3. Approches Lagrangiennes

4. Approche par Programmation Semidéfinie
- a. Bases de la Programmation Semidéfinie
- b. Application à la Programmation Quadratique en 0-1
  - 1. Le nombre de Lovász
  - 2. Relaxation SDP en $\{0, 1\}$
  - 3. Relaxation SDP en $\{-1, 1\}$
  - 4. Liens avec l'approche Lagrangienne
  - 5. Amélioration des relaxations Quadratiques Convexes
  - 6. Problèmes SDLS et contrainte sphérique

Lagrangien total d'un Programme Quadratique

Programme Quadratique quelconque :

$$(Q)\left\{\begin{array}{ll} \min_{x\in\Re^n} & x^TA_0x + b^Tx \\ \text{s.c.} & x^TA_ix + c_i^Tx - d_i \leq 0 \quad \forall i \in \{1,\ldots,m\} \end{array}\right.$$

Lagrangien total d'un Programme Quadratique

Programme Quadratique quelconque :

$$(Q) \begin{cases} \min_{x \in \Re^n} & x^T A_0 x + b^T x \\ \text{s.c.} & x^T A_i x + c_i^T x - d_i \leq 0 \quad \forall i \in \{1, \ldots, m\} \end{cases}$$

Son programme dual est (Lagrangien total) :

$$(\text{DT}) \sup_{\lambda \geq 0} \Theta(\lambda) = \inf_{x \in \Re^n} x^T A(\lambda) x + b(\lambda)^T x - \lambda^T d$$

où $A(\lambda) = A_0 + \sum_{i=1}^m \lambda_i A_i$ et $b(\lambda) = b + \sum_{i=1}^m \lambda_i c_i$

Lagrangien total d'un Programme Quadratique

Programme Quadratique quelconque :

$$(Q) \begin{cases} \min_{x \in \Re^n} & x^T A_0 x + b^T x \\ \text{s.c.} & x^T A_i x + c_i^T x - d_i \leq 0 \quad \forall i \in \{1, \ldots, m\} \end{cases}$$

Son programme dual est (Lagrangien total) :

$$(\text{DT}) \sup_{\lambda \geq 0} \Theta(\lambda) = \inf_{x \in \Re^n} x^T A(\lambda) x + b(\lambda)^T x - \lambda^T d$$

où $A(\lambda) = A_0 + \sum_{i=1}^m \lambda_i A_i$ et $b(\lambda) = b + \sum_{i=1}^m \lambda_i c_i$

Proposition

$\Theta(\lambda)$ *est finie si et seulement si* $A(\lambda) \succcurlyeq 0$ *et* $\exists x_\lambda$ *tel que* $2A(\lambda)x_\lambda + b(\lambda) = 0$ *(point critique)*

$$(\mathrm{DT})\ \sup_{\lambda \geq 0} \Theta(\lambda) = \inf_{x \in \Re^n} x^T A(\lambda) x + b(\lambda)^T x - \lambda^T d$$

Proposition

(DT) peut se formuler comme le programme semidéfini suivant :

$$(SD) \left\{ \begin{array}{ll} \sup_{\lambda \geq 0} & r - \lambda^T d \\ s.c. & F(r, \lambda) = \begin{bmatrix} -r & \frac{1}{2} b(\lambda)^T \\ \frac{1}{2} b(\lambda) & A(\lambda) \end{bmatrix} \succcurlyeq 0 \end{array} \right.$$

$$\text{(DT)} \sup_{\lambda \geq 0} \Theta(\lambda) = \inf_{x \in \Re^n} x^T A(\lambda) x + b(\lambda)^T x - \lambda^T d$$

Proposition

(DT) peut se formuler comme le programme semidéfini suivant :

$$(SD) \begin{cases} \sup_{\lambda \geq 0} & r - \lambda^T d \\ s.c. & F(r,\lambda) = \begin{bmatrix} -r & \frac{1}{2} b(\lambda)^T \\ \frac{1}{2} b(\lambda) & A(\lambda) \end{bmatrix} \succcurlyeq 0 \end{cases}$$

- (r, λ) est admissible de (SD) $\Leftrightarrow \forall (\alpha, y) \in \Re \times \Re^n$ $q(\alpha, y) = -\alpha^2 r + \alpha b(\lambda)^T y + y^T A(\lambda) y$ est positive

$$\text{(DT)} \sup_{\lambda \geq 0} \Theta(\lambda) = \inf_{x \in \Re^n} x^T A(\lambda) x + b(\lambda)^T x - \lambda^T d$$

Proposition

(DT) peut se formuler comme le programme semidéfini suivant :

$$(SD) \left\{ \begin{array}{ll} \sup_{\lambda \geq 0} & r - \lambda^T d \\ s.c. & F(r, \lambda) = \begin{bmatrix} -r & \frac{1}{2} b(\lambda)^T \\ \frac{1}{2} b(\lambda) & A(\lambda) \end{bmatrix} \succcurlyeq 0 \end{array} \right.$$

- (r, λ) est admissible de (SD) $\Leftrightarrow \forall (\alpha, y) \in \Re \times \Re^n$ $q(\alpha, y) = -\alpha^2 r + \alpha b(\lambda)^T y + y^T A(\lambda) y$ est positive
- $\alpha = 0$: $A(\lambda) \succcurlyeq 0$

$$\text{(DT)} \sup_{\lambda \geq 0} \Theta(\lambda) = \inf_{x \in \Re^n} x^T A(\lambda) x + b(\lambda)^T x - \lambda^T d$$

Proposition

(DT) peut se formuler comme le programme semidéfini suivant :

$$(SD) \left\{ \begin{array}{ll} \sup_{\lambda \geq 0} & r - \lambda^T d \\ s.c. & F(r, \lambda) = \left[\begin{array}{cc} -r & \frac{1}{2} b(\lambda)^T \\ \frac{1}{2} b(\lambda) & A(\lambda) \end{array} \right] \succcurlyeq 0 \end{array} \right.$$

- (r, λ) est admissible de (SD) $\Leftrightarrow \forall (\alpha, y) \in \Re \times \Re^n$ $q(\alpha, y) = -\alpha^2 r + \alpha b(\lambda)^T y + y^T A(\lambda) y$ est positive
- $\alpha = 0$: $A(\lambda) \succcurlyeq 0$
- $\alpha \neq 0$: $q(1, x) = -r + b(\lambda)^T x + x^T A(\lambda) x \geq 0$ pour tout x dans $\Re^n$. Donc $r - \lambda^T d \leq x^T A(\lambda) x + b(\lambda)^T x - \lambda^T d$

Bidualiser $\Leftrightarrow$ convexifier en Programme Semidéfini

Le dual semidéfini de :

$$
(SD) \begin{cases} \sup_{\lambda \geq 0} & r - \lambda^T d \\ \text{s.c.} & \begin{bmatrix} -r & \frac{b^T + \sum_{i=1}^m \lambda_i c_i^T}{2} \\ \frac{b + \sum_{i=1}^m \lambda_i c_i}{2} & A_0 + \sum_{i=1}^m \lambda_i A_i \end{bmatrix} \succcurlyeq 0 \end{cases}
$$

Bidualiser $\Leftrightarrow$ convexifier en Programme Semidéfini

Le dual semidéfini de :

$$(SD) \left\{ \begin{array}{ll} \sup_{\lambda \geq 0} & r - \lambda^T d \\ \text{s.c.} & \begin{bmatrix} -r & \frac{b^T + \sum_{i=1}^m \lambda_i c_i^T}{2} \\ \frac{b + \sum_{i=1}^m \lambda_i c_i}{2} & A_0 + \sum_{i=1}^m \lambda_i A_i \end{bmatrix} \succcurlyeq 0 \end{array} \right.$$

est :

$$(SDP_Q) \left\{ \begin{array}{lll} \min & \begin{bmatrix} 0 & \frac{b^T}{2} \\ \frac{b}{2} & A_0 \end{bmatrix} \bullet \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} & \\ \text{s.c.} & A_i \bullet X + c_i^T x - d_i \leq 0 & \forall i \in \{1, \ldots, m\} \\ & \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0 & \end{array} \right.$$

Bidualiser ⇔ convexifier en Programme Semidéfini

Le dual semidéfini de :

$$(SD)\left\{\begin{array}{ll} \sup_{\lambda\geq 0} & r - \lambda^T d \\ \text{s.c.} & \begin{bmatrix} -r & \frac{b^T+\sum_{i=1}^m \lambda_i c_i^T}{2} \\ \frac{b+\sum_{i=1}^m \lambda_i c_i}{2} & A_0 + \sum_{i=1}^m \lambda_i A_i \end{bmatrix} \succcurlyeq 0 \end{array}\right.$$

est :

$$(SDP_Q)\left\{\begin{array}{ll} \min & \begin{bmatrix} 0 & \frac{b^T}{2} \\ \frac{b}{2} & A_0 \end{bmatrix} \bullet \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \\ \text{s.c.} & A_i \bullet X + c_i^T x - d_i \leq 0 \qquad \forall i \in \{1,\ldots,m\} \\ & \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0 \end{array}\right.$$

On retrouve la relaxation semidéfinie issue du modèle {0,1}

Lagrangien Partiel

Supposons que notre problème Quadratique contient des contraintes linéaires :

$$(\mathrm{P}) \begin{cases} \min_{x \in \Re^n} & x^T Q x + c^T x \\ \text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq) \, e_i \quad i \in I = \{1, \ldots, m\} \\ & Ax = b \end{cases}$$

Lagrangien Partiel

Supposons que notre problème Quadratique contient des contraintes linéaires :

$$(\mathrm{P}) \left\{ \begin{array}{lll} \min_{x \in \Re^n} & x^T Q x + c^T x & \\ \text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq) e_i & i \in I = \{1, \ldots, m\} \\ & Ax = b & \end{array} \right.$$

Lagrangien Partiel

Supposons que notre problème Quadratique contient des contraintes linéaires :

$$(\mathrm{P}) \begin{cases} \min_{x \in \Re^n} & x^T Q x + c^T x \\ \text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq)\, e_i \quad i \in I = \{1, \dots, m\} \\ & Ax = b \end{cases}$$

- On ne relâche pas les contraintes linéaires :
  $\mathfrak{L}_{DP}(x, \mu) = x^T \left(Q + \sum_{i \in I} \mu_i B_i\right) x + \left(c + \sum_{i \in I} \mu_i d_i\right)^T x - \mu^T e$

Lagrangien Partiel

Supposons que notre problème Quadratique contient des contraintes linéaires :

$$(P) \begin{cases} \min_{x \in \Re^n} & x^T Qx + c^T x \\ \text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq) \, e_i \quad i \in I = \{1, \ldots, m\} \\ & Ax = b \end{cases}$$

- On ne relâche pas les contraintes linéaires :
  $\mathfrak{L}_{DP}(x, \mu) = x^T \left(Q + \sum_{i \in I} \mu_i B_i\right) x + \left(c + \sum_{i \in I} \mu_i d_i\right)^T x - \mu^T e$
- Rappel : Dans la relaxation (totale) précédente :
  $\mathfrak{L}_{DT}(x, \mu, \lambda) = \mathfrak{L}_{DP}(x, \mu) + \lambda^T (Ax - b)$

Lagrangien Partiel

Supposons que notre problème Quadratique contient des contraintes linéaires :

$$
(\mathrm{P}) \begin{cases} \min_{x \in \Re^n} & x^T Q x + c^T x \\ \text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq)\, e_i \quad i \in I = \{1, \ldots, m\} \\ & Ax = b \end{cases}
$$

- On ne relâche pas les contraintes linéaires :
  $\mathfrak{L}_{DP}(x, \mu) = x^T \left(Q + \sum_{i \in I} \mu_i B_i\right) x + \left(c + \sum_{i \in I} \mu_i d_i\right)^T x - \mu^T e$
- Rappel : Dans la relaxation (totale) précédente :
  $\mathfrak{L}_{DT}(x, \mu, \lambda) = \mathfrak{L}_{DP}(x, \mu) + \lambda^T (Ax - b)$

Proposition

$(DT)\ \sup_{\mu, \lambda} \inf_x \mathfrak{L}_{DT}(x, \mu, \lambda) \leq (DP)\ \sup_{\mu} \inf_{x \mid Ax = b} \mathfrak{L}_{DP}(x, \mu)$

Ajout de contraintes quadratiques redondantes avant d'effectuer la bidualisation

$f_j(x) = 0$ sur $\{x : Ax = b\}$

$\mathfrak{I} = \left\{ f_j(x) = x^T C_j x + q_j^T x + \alpha_j \ : \ j \in J \right\}$

Ajout de contraintes quadratiques redondantes avant d'effectuer la bidualisation

$f_j(x) = 0$ sur $\{x : Ax = b\}$

$\mathfrak{J} = \left\{ f_j(x) = x^T C_j x + q_j^T x + \alpha_j \ : \ j \in J \right\}$

Ajouter $f_j(x) = 0 \forall j$ à (P) donne :

Ajout de contraintes quadratiques redondantes avant d'effectuer la bidualisation

$f_j(x) = 0$ sur $\{x : Ax = b\}$

$\mathfrak{I} = \left\{ f_j(x) = x^T C_j x + q_j^T x + \alpha_j \ : \ j \in J \right\}$

Ajouter $f_j(x) = 0 \forall j$ à (P) donne :

- $\mathfrak{L}_{DT_{\mathfrak{I}}}(x, \mu, \lambda, \omega) = \mathfrak{L}_{DT}(x, \mu, \lambda) + \sum_{j \in J} \omega_j f_j(x)$
  $\rightarrow$ meilleure borne !

Ajout de contraintes quadratiques redondantes avant d'effectuer la bidualisation

$f_j(x) = 0$ sur $\{x : Ax = b\}$
$\mathfrak{J} = \left\{ f_j(x) = x^T C_j x + q_j^T x + \alpha_j \ : \ j \in J \right\}$

Ajouter $f_j(x) = 0 \forall j$ à (P) donne :

- $\mathfrak{L}_{DT_{\mathfrak{J}}}(x, \mu, \lambda, \omega) = \mathfrak{L}_{DT}(x, \mu, \lambda) + \sum_{j \in J} \omega_j f_j(x)$
  $\rightarrow$ meilleure borne !
- $\mathfrak{L}_{DP_{\mathfrak{J}}}(x, \mu, \omega) = \mathfrak{L}_{DP}(x, \mu, \omega)$ $\quad f_j(x) = 0$ car $Ax = b$!

Ajout de contraintes quadratiques redondantes avant d'effectuer la bidualisation

$f_j(x) = 0$ sur $\{x : Ax = b\}$
$\mathfrak{J} = \left\{ f_j(x) = x^T C_j x + q_j^T x + \alpha_j \ : \ j \in J \right\}$

Ajouter $f_j(x) = 0 \forall j$ à (P) donne :

- $\mathfrak{L}_{DT_{\mathfrak{J}}}(x, \mu, \lambda, \omega) = \mathfrak{L}_{DT}(x, \mu, \lambda) + \sum_{j \in J} \omega_j f_j(x)$
  $\rightarrow$ meilleure borne !
- $\mathfrak{L}_{DP_{\mathfrak{J}}}(x, \mu, \omega) = \mathfrak{L}_{DP}(x, \mu, \omega)$ $\quad f_j(x) = 0$ car $Ax = b$!

Proposition

Pour tout ensemble $\mathfrak{J}$, $(DP)_{\mathfrak{J}}$ est équivalent à (DP), mais $(DT)_{\mathfrak{J}}$ est généralement meilleur que (DT)

Une condition suffisante pour avoir $(DT)_{\mathfrak{I}}$=(DP)

$\forall \mathfrak{I}$ on a : $\quad (DT)_{\mathfrak{I}} \leq (DP) \leq (P)$

Une condition suffisante pour avoir $(DT)_{\mathfrak{J}}$=(DP)

Si $\mathfrak{L}_{DT_{\mathfrak{J}}}$ est convexe alors $(DT)_{\mathfrak{J}}$=(DP)≤(P)

Proposition

Soit μ^* une solution optimale de (DP). S'il existe ω^* tel que $\mathfrak{L}_{DT_{\mathfrak{J}}}(x, \mu^*, \lambda, \omega^*)$ est convexe, alors les valeurs optimales de $(DT)_{\mathfrak{J}}$, $(SDP)_{\mathfrak{J}}$ et (DP) sont égales

Une condition suffisante pour avoir $(DT)_{\mathfrak{J}}$=(DP)

Si $\mathfrak{L}_{DT_{\mathfrak{J}}}$ est convexe alors $(DT)_{\mathfrak{J}}$=(DP)$\leq$(P)

Proposition

Soit μ^* une solution optimale de (DP). S'il existe ω^* tel que $\mathfrak{L}_{DT_{\mathfrak{J}}}(x, \mu^*, \lambda, \omega^*)$ est convexe, alors les valeurs optimales de $(DT)_{\mathfrak{J}}$, $(SDP)_{\mathfrak{J}}$ et (DP) sont égales

Une condition suffisante pour avoir $(DT)_{\mathfrak{J}}$=(DP)

Si $\mathfrak{L}_{DT_{\mathfrak{J}}}$ est convexe alors $(DT)_{\mathfrak{J}}$=(DP)≤(P)

Proposition

Soit μ^* une solution optimale de (DP). S'il existe ω^* tel que $\mathfrak{L}_{DT_{\mathfrak{J}}}(x, \mu^*, \lambda, \omega^*)$ est convexe, alors les valeurs optimales de $(DT)_{\mathfrak{J}}$, $(SDP)_{\mathfrak{J}}$ et (DP) sont égales

On retrouve dans la formulation semidéfinie les contraintes quadratiques linéarisées :

$$(DT)_{\mathfrak{J}} \Leftrightarrow (SDP)_{\mathfrak{J}} \left\{ \begin{array}{lll} \min & Q \bullet X + c^T x & \\ \text{s.c.} & Ax = b & \\ & B_i \bullet X + d_i^T x = (\text{ou } \leq)\, e_i & i \in I \\ & C_j \bullet X + q_j^T x + \alpha_j = 0 & j \in J \\ & \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0 & \end{array} \right.$$

Deux convexifications équivalentes dans le cas 0-1

$$(P)_{0-1} \quad \begin{cases} \min & x^T Q x + c^T x \\ \text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq)\, e_i \quad i \in I = \{1, \ldots, m\} \\ & Ax = b \\ & x_i^2 = x_i \Leftrightarrow x_i \in \{0,1\} \qquad \forall i \in \{1, \ldots, n\} \end{cases}$$

Deux convexifications équivalentes dans le cas 0-1

$$
(P)_{0-1} \left\{ \begin{array}{lll} \min & x^T Q x + c^T x & \\ \text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq)\, e_i & i \in I = \{1, \ldots, m\} \\ & Ax = b \text{ et } (Ax - b)^2 = 0 & \\ & x_i^2 = x_i \Leftrightarrow x_i \in \{0, 1\} & \forall i \in \{1, \ldots, n\} \end{array} \right.
$$

- $\mathfrak{L}_{DT_C}(x, \mu, \lambda, \omega) = \mathfrak{L}_{DT}(x, \mu, \lambda) + \omega\,(Ax - b)^2$

Deux convexifications équivalentes dans le cas 0-1

$$(P)_{0-1} \left\{ \begin{array}{lll} \min & x^T Q x + c^T x & \\ \text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq)\, e_i & i \in I = \{1, \ldots, m\} \\ & Ax = b \text{ et } (Ax - b)^2 = 0 & \\ & x_i^2 = x_i \Leftrightarrow x_i \in \{0, 1\} & \forall i \in \{1, \ldots, n\} \end{array} \right.$$

- $\mathfrak{L}_{DT_C}(x, \mu, \lambda, \omega) = \mathfrak{L}_{DT}(x, \mu, \lambda) + \omega\,(Ax - b)^2$
- Convexification réussie : (DP) peut être formulé comme un programme semidéfini

Deux convexifications équivalentes dans le cas 0-1

$$(P)_{0-1}\quad \left\{\begin{array}{lll} \min & x^TQx + c^Tx & \\ \text{s.c.} & x^TB_ix + d_i^Tx = (\text{ou } \leq)\ e_i & i \in I = \{1,\ldots,m\} \\ & Ax = b \text{ et } (Ax-b)^2 = 0 & \\ & x_i^2 = x_i \Leftrightarrow x_i \in \{0,1\} & \forall i \in \{1,\ldots,n\} \end{array}\right.$$

- $\mathfrak{L}_{DT_C}(x,\mu,\lambda,\omega) = \mathfrak{L}_{DT}(x,\mu,\lambda) + \omega\,(Ax-b)^2$
- Convexification réussie : (DP) peut être formulé comme un programme semidéfini
- Il suffit d'ajouter $A^TA \bullet X - 2b^TAx + b^2 = 0$ dans le PSD

Deux convexifications équivalentes dans le cas 0-1

$$
(P)_{0-1} \left\{ \begin{array}{lll} \min & x^T Q x + c^T x & \\ \text{s.c.} & x^T B_i x + d_i^T x = (\text{ou } \leq)\, e_i & i \in I = \{1, \ldots, m\} \\ & Ax = b \text{ et } (Ax - b)^2 = 0 & \\ & x_i^2 = x_i \Leftrightarrow x_i \in \{0, 1\} & \forall i \in \{1, \ldots, n\} \end{array} \right.
$$

- $\mathfrak{L}_{DT_C}(x, \mu, \lambda, \omega) = \mathfrak{L}_{DT}(x, \mu, \lambda) + \omega\,(Ax - b)^2$
- Convexification réussie : (DP) peut être formulé comme un programme semidéfini
- Il suffit d'ajouter $A^T A \bullet X - 2b^T Ax + b^2 = 0$ dans le PSD
- Dans le cas non-booléen, $(Ax - b)^2$ ne convexifie pas toujours le Lagrangien

Une deuxième convexification (valable dans le cas général)

Théorème

Soient A et Q respectivement une matrice $p \times n$ et une matrice $n \times n$. SI Q est positive sur $L = \ker(A)$ alors il existe une combinaison linéaire des fonctions quadratiques $q_{ij}(x) = x_i(a_j^T x - b_j)$ pour $i \in \{1, \ldots, n\}$ et $j \in \{1, \ldots, p\}$ qui convexifie la forme quadratique $x^T Qx$ sur $\Re^n$ tout entier.

Une deuxième convexification (valable dans le cas général)

Théorème

Soient A et Q respectivement une matrice $p \times n$ et une matrice $n \times n$. SI Q est positive sur $L = \ker(A)$ alors il existe une combinaison linéaire des fonctions quadratiques $q_{ij}(x) = x_i(a_j^T x - b_j)$ pour $i \in \{1, \ldots, n\}$ et $j \in \{1, \ldots, p\}$ qui convexifie la forme quadratique $x^T Qx$ sur $\Re^n$ tout entier.

Une deuxième convexification (valable dans le cas général)

Théorème

Soient A et Q respectivement une matrice $p \times n$ et une matrice $n \times n$. SI Q est positive sur $L = \ker(A)$ alors il existe une combinaison linéaire des fonctions quadratiques $q_{ij}(x) = x_i(a_j^T x - b_j)$ pour $i \in \{1, \ldots, n\}$ et $j \in \{1, \ldots, p\}$ qui convexifie la forme quadratique $x^T Qx$ sur $\Re^n$ tout entier.

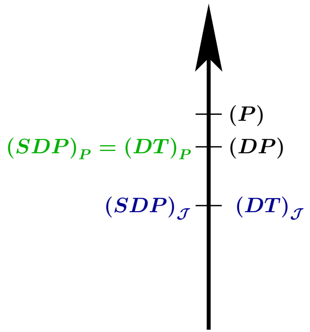
Formulation semidéfinie de la relaxation lagrangienne partielle (DP)

$$
(SDP_P) \left\{ \begin{array}{lll}
\min & Q \bullet X + c^T x & \\
\text{s.c.} & Ax = b & \\
& B_i \bullet X + d_i^T x = (\text{ou } \leq) \, e_i & i \in I \\
& \sum_{k=1}^{n} A_{jk} X_{ki} - b_j x_i = 0 & i \in \{1, \ldots, n\} \; j \in \{1, \ldots, p\} \\
& (X_{ii} = x_i) & (i \in \{1, \ldots, n\}) \\
& \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0 &
\end{array} \right.
$$

Ces deux relaxations semidéfinies permettent d'atteindre la valeur de (DP) (cas 0-1)

corollaire

(DP) est une limite pour les relaxations semidéfinies utilisant des contraintes redondantes construites à partir de $Ax = b$



Preuve directe de l'équivalence sans passer par la dualité lagrangienne

Ici cas simplifié : une seule contrainte linéaire

Proposition

Soit $(X, x) \in S_n \times \mathbb{R}$ *vérifiant* $cc^T \bullet X = d^2$, $c^T x = d$ *et* $\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0$, *alors* (X, x) *vérifie* $\sum_{j=1}^{n} c_j X_{ij} = dx_i \, \forall i \in \{1, ..., n\}$ *(réciproque évidente)*

Preuve directe de l'équivalence sans passer par la dualité lagrangienne

Ici cas simplifié : une seule contrainte linéaire

Proposition

Soit $(X, x) \in S_n \times \mathbb{R}$ *vérifiant* $cc^T \bullet X = d^2$, $c^T x = d$ *et* $\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0$, *alors* (X, x) *vérifie* $\sum_{j=1}^{n} c_j X_{ij} = dx_i \; \forall i \in \{1, ..., n\}$ *(réciproque évidente)*

Preuve directe de l'équivalence sans passer par la dualité lagrangienne

Ici cas simplifié : une seule contrainte linéaire

Proposition

Soit $(X, x) \in S_n \times \mathbb{R}$ *vérifiant* $cc^T \bullet X = d^2$, $c^T x = d$ *et* $\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0$, *alors* (X, x) *vérifie* $\sum_{j=1}^n c_j X_{ij} = dx_i \, \forall i \in \{1, ..., n\}$ *(réciproque évidente)*

Preuve.
$cc^T \bullet X = d^2 \Rightarrow cc^T \bullet (X - xx^T) + (c^T x)^2 = d^2$
Donc $cc^T \bullet (X - xx^T) = 0$
$\Rightarrow cc^T (X - xx^T) = 0$

Preuve directe de l'équivalence sans passer par la dualité lagrangienne

Ici cas simplifié : une seule contrainte linéaire

Proposition

Soit $(X, x) \in S_n \times \mathbb{R}$ *vérifiant* $cc^T \bullet X = d^2$, $c^T x = d$ *et* $\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0$, *alors* (X, x) *vérifie* $\sum_{j=1}^{n} c_j X_{ij} = dx_i \; \forall i \in \{1, ..., n\}$ *(réciproque évidente)*

Preuve.

$cc^T \bullet X = d^2 \Rightarrow cc^T \bullet \left(X - xx^T\right) + \left(c^T x\right)^2 = d^2$

Donc $cc^T \bullet \left(X - xx^T\right) = 0$

$\Rightarrow cc^T \left(X - xx^T\right) = 0$

$\forall k, j \in \{1, ..., n\}$ $c_k \left(\sum_{i=1}^{n} c_i \left(X_{ij} - x_i x_j\right)\right) = 0.$

- $c = 0$: résultat évident

Preuve directe de l'équivalence sans passer par la dualité lagrangienne

Ici cas simplifié : une seule contrainte linéaire

Proposition

Soit $(X,x) \in S_n \times \mathbb{R}$ *vérifiant* $cc^T \bullet X = d^2$, $c^T x = d$ *et* $\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0$, *alors* (X,x) *vérifie* $\sum_{j=1}^n c_j X_{ij} = dx_i \, \forall i \in \{1,...,n\}$ *(réciproque évidente)*

Preuve.

$cc^T \bullet X = d^2 \Rightarrow cc^T \bullet (X - xx^T) + \left(c^T x\right)^2 = d^2$

Donc $cc^T \bullet (X - xx^T) = 0$

$\Rightarrow cc^T (X - xx^T) = 0$

$\forall k, j \in \{1,...,n\} \; c_k \left(\sum_{i=1}^n c_i \left(X_{ij} - x_i x_j\right)\right) = 0.$

- $c = 0$: résultat évident
- Il existe k_0 tel que $c_{k0} \neq 0$.
  $\forall j \in \{1,...,n\} \sum_{i=1}^n c_i X_{ij} = x_j \sum_{i=1}^n c_i x_i = dx_j$.

Application : le Problème k-cluster

- $G = (V, E)$ un graphe non orienté de n sommets et un entier $1 \leq k \leq n$

Application : le Problème k-cluster

- $G = (V, E)$ un graphe non orienté de n sommets et un entier $1 \leq k \leq n$
- Chaque arête $[v_i, v_j]$ porte un poids W_{ij} (on a $W_{ii} = 0$ pour tout i dans $\{1, \ldots, n\}$)

Application : le Problème k-cluster

- $G = (V, E)$ un graphe non orienté de n sommets et un entier $1 \leq k \leq n$
- Chaque arête $[v_i, v_j]$ porte un poids W_{ij} (on a $W_{ii} = 0$ pour tout i dans $\{1, \ldots, n\}$)
- Trouver le sous-graphe de k sommets de poids maximal

Application : le Problème k-cluster

- $G = (V, E)$ un graphe non orienté de n sommets et un entier $1 \leq k \leq n$
- Chaque arête $[v_i, v_j]$ porte un poids W_{ij} (on a $W_{ii} = 0$ pour tout i dans $\{1, \ldots, n\}$)
- Trouver le sous-graphe de k sommets de poids maximal

Application : le Problème k-cluster

- $G = (V, E)$ un graphe non orienté de n sommets et un entier $1 \leq k \leq n$
- Chaque arête $[v_i, v_j]$ porte un poids W_{ij} (on a $W_{ii} = 0$ pour tout i dans $\{1, \ldots, n\}$)
- Trouver le sous-graphe de k sommets de poids maximal

$$(KC)_{\{0,1\}} \left\{ \begin{array}{ll} \max & \frac{1}{2} W \bullet X \\ \text{s.c.} & \sum_{i=1}^{n} x_i = k \\ & x \in \{0,1\}^n \\ & X = xx^T \end{array} \right.$$

Application : le Problème k-cluster

- $G = (V, E)$ un graphe non orienté de n sommets et un entier $1 \leq k \leq n$
- Chaque arête $[v_i, v_j]$ porte un poids W_{ij} (on a $W_{ii} = 0$ pour tout i dans $\{1, \ldots, n\}$)
- Trouver le sous-graphe de k sommets de poids maximal

$$(KC)_{\{0,1\}} \left\{ \begin{array}{ll} \max & \frac{1}{2} W \bullet X \\ \text{s.c.} & \sum_{i=1}^{n} x_i = k \\ & x \in \{0, 1\}^n \\ & X = xx^T \end{array} \right.$$

$$(KC)_{\{-1,1\}} \left\{ \begin{array}{ll} \max & \frac{1}{8} \sum_{i=1}^{n} \sum_{j \neq i} W_{ij} \left(1 + y_i y_j + y_i + y_j\right) \\ \text{s.c.} & \sum_{i=1}^{n} y_i = 2k - n \\ & y \in \{-1, 1\}^n \end{array} \right.$$

Relaxations semidéfinies "basiques"

Relaxations semidéfinies "basiques"

$$
(KC\,SDP)_{\{0,1\}} \left\{ \begin{array}{ll} \text{Maximiser} & \frac{1}{2} W \bullet X \\ \text{Sous contraintes} & e_n^T x = k \\ & \mathbf{d}(X) = x \\ & X' = \begin{bmatrix} 1 & x \\ x^T & X \end{bmatrix} \succeq 0 \end{array} \right.
$$

Relaxations semidéfinies "basiques"

$$(KC\,SDP)_{\{0,1\}} \left\{ \begin{array}{ll} \text{Maximiser} & \frac{1}{2} W \bullet X \\ \text{Sous contraintes} & e_n^T x = k \\ & \mathbf{d}(X) = x \\ & X' = \begin{bmatrix} 1 & x \\ x^T & X \end{bmatrix} \succeq 0 \end{array} \right.$$

$$(KC\,SDP)_{\{-1,1\}} \left\{ \begin{array}{ll} \max & \frac{1}{8} \sum_{i=1}^{n} \sum_{j \neq i} W_{ij} \left(1 + Y_{ij} + Y_{0i} + Y_{0j}\right) \\ \text{s.c.} & \sum_{i=1}^{n} Y_{0i} = 2k - n \\ & \mathbf{d}(Y) = e_{n+1} \\ & Y \succeq 0 \end{array} \right.$$

Construction d'une meilleure relaxation semidéfinie

$$(PL)_{KC}\left\{\begin{array}{lll} \max & \frac{1}{2}W \bullet X & \\ \text{s.c. :} & & \\ (a) & \sum_{i=1}^{n} x_i = e_n^T x = k & \\ (b) & \sum_{i<j} X_{ij} + \sum_{j>i} X_{ij} = (k-1)x_i & i \in \{1,\ldots,n\} \\ (c) & X_{ij} \geq 0 & i<j \notin E \\ (d) & X_{ij} \leq x_i \,;\, X_{ij} \leq x_j & i<j \notin E \\ & 0 \leq x_i \leq 1 & i \in \{1,\ldots,n\} \end{array}\right.$$

Construction d'une meilleure relaxation semidéfinie

$$(PL)_{KC}\left\{\begin{array}{lll}
\max & \frac{1}{2}W \bullet X & \\
\text{s.c. :} & & \\
(a) & \sum_{i=1}^{n} x_i = e_n^T x = k & \\
(b) & \sum_{i<j} X_{ij} + \sum_{j>i} X_{ij} = (k-1)x_i & i \in \{1,\ldots,n\} \\
(c) & X_{ij} \geq 0 & i<j \notin E \\
(d) & X_{ij} \leq x_i \,;\, X_{ij} \leq x_j & i<j \notin E \\
 & 0 \leq x_i \leq 1 & i \in \{1,\ldots,n\}
\end{array}\right.$$

$$(SDP)_{KC}\left\{\begin{array}{ll}
\max & \frac{1}{2}W \bullet X \\
\text{s.c. :} & \\
(a') & e_n^T x = k \,,\, (e_n e_n^T \bullet X - 2k e_n^T x + k^2 = 0) \\
(b') & \sum_{i=1}^{n} X_{ij} = k x_j \;\forall j \in \{1,\ldots,n\} \\
(c') & X_{ij} \geq 0 \;\, i<j \in \{1,\ldots,n\} \\
(d') & X_{ij} \leq x_i \,;\, X_{ij} \leq x_j \;\, i<j \in \{1,\ldots,n\} \\
 & \mathbf{d}(X) = x \,;\, X - xx^T \succeq 0
\end{array}\right.$$

k-cluster : Relaxations semidéfinie et Lagrangienne partielle

Formulation semidéfinie de la relaxation Lagrangienne partielle de :

k-cluster : Relaxations semidéfinie et Lagrangienne partielle

Formulation semidéfinie de la relaxation Lagrangienne partielle de :

$$\begin{cases} \max & \frac{1}{2}x^T W x & \\ \text{s.c.} & \sum_{i=1}^n x_i = k & \\ & x_i^2 = x_i & i = 1, \ldots, n \\ & x_i x_j \geq 0 & i < j \in \{1, \ldots, n\} \\ & x_i x_j \leq x_i \,;\, x_i x_j \leq x_j & i < j \in \{1, \ldots, n\} \end{cases}$$

Plan du cours

1. Introduction

2. Linéarisations

3. Approches Lagrangiennes

4. Approche par Programmation Semidéfinie
- a. Bases de la Programmation Semidéfinie
- b. Application à la Programmation Quadratique en 0-1
  - 1. Le nombre de Lovász
  - 2. Relaxation SDP en $\{0, 1\}$
  - 3. Relaxation SDP en $\{-1, 1\}$
  - 4. Liens avec l'approche Lagrangienne
  - 5. Amélioration des relaxations Quadratiques Convexes
  - 6. Problèmes SDLS et contrainte sphérique

4.b.5. Amélioration des relaxations Quadratiques Convexes

Programme Quadratique en variables 0-1 soumis à des contraintes linéaires :

$$(\mathrm{QL}) \begin{cases} \min & x^T Q x + c^T x \\ \text{s.c.} & Ax = b \\ & x \in \{0, 1\}^n \end{cases}$$

Q est quelconque donc problème non convexe même en relâchant les contraintes 0-1

4.b.5. Amélioration des relaxations Quadratiques Convexes

Programme Quadratique en variables 0-1 soumis à des contraintes linéaires :

$$(\mathrm{QL}) \left\{ \begin{array}{ll} \min & x^T Q x + c^T x \\ \text{s.c.} & Ax = b \\ & x \in \{0,1\}^n \end{array} \right.$$

Q est quelconque donc problème non convexe même en relâchant les contraintes 0-1

Idée de départ : "Capter" la qualité des bornes semidéfinies dans une relaxation quadratique convexe (très rapide à résoudre)

4.b.5. Amélioration des relaxations Quadratiques Convexes

Programme Quadratique en variables 0-1 soumis à des contraintes linéaires :

$$(\text{QL}) \begin{cases} \min & x^T Q x + c^T x \\ \text{s.c.} & Ax = b \\ & x \in \{0,1\}^n \end{cases}$$

Q est quelconque donc problème non convexe même en relâchant les contraintes 0-1

Idée de départ : "Capter" la qualité des bornes semidéfinies dans une relaxation quadratique convexe (très rapide à résoudre)

Principe de l'approche :

1. Ajouter un terme quadratique à la fonction pour la convexifier puis relaxation continue ($x \in [0,1]^n$)

4.b.5. Amélioration des relaxations Quadratiques Convexes

Programme Quadratique en variables 0-1 soumis à des contraintes linéaires :

$$(\text{QL}) \begin{cases} \min & x^T Q x + c^T x \\ \text{s.c.} & Ax = b \\ & x \in \{0,1\}^n \end{cases}$$

Q est quelconque donc problème non convexe même en relâchant les contraintes 0-1

Idée de départ : "Capter" la qualité des bornes semidéfinies dans une relaxation quadratique convexe (très rapide à résoudre)

Principe de l'approche :

1. Ajouter un terme quadratique à la fonction pour la convexifier puis relaxation continue ($x \in [0,1]^n$)
2. On peut obtenir le "meilleur" terme possible en résolvant un programme semidéfini

Convexification d'un programme quadratique (1)

Première idée : ajouter $K \sum_{i=1}^{n} \left(x_i^2 - x_i\right)$ à la fonction $x^T Qx + c^T x$ avec K "assez grand" : fonction nulle pour x admissible (donc vérifiant $x \in \{0, 1\}^n$)

Convexification d'un programme quadratique (1)

Première idée : ajouter $K \sum_{i=1}^{n} \left(x_i^2 - x_i\right)$ à la fonction $x^T Qx + c^T x$ avec K "assez grand" : fonction nulle pour x admissible (donc vérifiant $x \in \{0,1\}^n$)

- Fonction convexe : $Q + KI_n \succcurlyeq 0$

Convexification d'un programme quadratique (1)

Première idée : ajouter $K\sum_{i=1}^{n}\left(x_i^2 - x_i\right)$ à la fonction $x^TQx + c^Tx$ avec K "assez grand" : fonction nulle pour x admissible (donc vérifiant $x \in \{0,1\}^n$)

- Fonction convexe : $Q + KI_n \succcurlyeq 0$
- On prend $K \geq -\min_j \lambda_j(Q)$

Convexification d'un programme quadratique (1)

Première idée : ajouter $K \sum_{i=1}^{n} \left(x_i^2 - x_i\right)$ à la fonction $x^T Qx + c^T x$ avec K "assez grand" : fonction nulle pour x admissible (donc vérifiant $x \in \{0,1\}^n$)

- Fonction convexe : $Q + KI_n \succcurlyeq 0$
- On prend $K \geq -\min_j \lambda_j(Q)$
- On obtient une relaxation quadratique convexe en relâchant $x \in \{0,1\}^n$ en $x \in [0,1]^n$

Convexification d'un programme quadratique (1)

Première idée : ajouter $K\sum_{i=1}^{n}\left(x_i^2 - x_i\right)$ à la fonction $x^TQx + c^Tx$ avec K "assez grand" : fonction nulle pour x admissible (donc vérifiant $x \in \{0,1\}^n$)

- Fonction convexe : $Q + KI_n \succcurlyeq 0$
- On prend $K \geq -\min_j \lambda_j(Q)$
- On obtient une relaxation quadratique convexe en relâchant $x \in \{0,1\}^n$ en $x \in [0,1]^n$
- Amélioration possible : ajouter $\sum_{i=1}^{n} K_i\left(x_i^2 - x_i\right)$

Convexification d'un programme quadratique (1)

Première idée : ajouter $K \sum_{i=1}^{n} \left(x_i^2 - x_i\right)$ à la fonction $x^T Qx + c^T x$ avec K "assez grand" : fonction nulle pour x admissible (donc vérifiant $x \in \{0,1\}^n$)

- Fonction convexe : $Q + KI_n \succcurlyeq 0$
- On prend $K \geq -\min_j \lambda_j(Q)$
- On obtient une relaxation quadratique convexe en relâchant $x \in \{0,1\}^n$ en $x \in [0,1]^n$
- Amélioration possible : ajouter $\sum_{i=1}^{n} K_i \left(x_i^2 - x_i\right)$
- En fait, on peut considérer une approche encore plus générale en utilisant également les contraintes $Ax = b$!

Convexification d'un programme quadratique (2)

Amélioration de l'approche :

$\Rightarrow$ Ajouter une combinaison linéaire des fonctions $h_i(x) = x_i^2 - x_i$ ($i = 1, \ldots, n$) et $g_{ij}(x) = \sum_{k=1}^{n} A_{jk} x_k x_i - b_j x_i$ ($j = 1, \ldots, p$ et $i = 1, \ldots, n$) à la fonction $x^T Q x + c^T x$: fonction nulle pour x admissible (vérifiant $x \in \{0, 1\}^n$ et $Ax = b$)

Convexification d'un programme quadratique (2)

Amélioration de l'approche :

$\Rightarrow$ Ajouter une combinaison linéaire des fonctions $h_i(x) = x_i^2 - x_i$ ($i = 1, \ldots, n$) et $g_{ij}(x) = \sum_{k=1}^{n} A_{jk} x_k x_i - b_j x_i$ ($j = 1, \ldots, p$ et $i = 1, \ldots, n$) à la fonction $x^T Q x + c^T x$: fonction nulle pour x admissible (vérifiant $x \in \{0, 1\}^n$ et $Ax = b$)

$\Rightarrow$ On reconnaît les contraintes "produit" de l'approche linéaire d'Adams-Sharali et qui jouent également un rôle important dans la formulation semidéfinie de la relaxation Lagrangienne partielle de (QL) !

Convexification d'un programme quadratique (2)

Amélioration de l'approche :

$\Rightarrow$ Ajouter une combinaison linéaire des fonctions $h_i(x) = x_i^2 - x_i$ ($i = 1, \ldots, n$) et $g_{ij}(x) = \sum_{k=1}^{n} A_{jk} x_k x_i - b_j x_i$ ($j = 1, \ldots, p$ et $i = 1, \ldots, n$) à la fonction $x^T Qx + c^T x$: fonction nulle pour x admissible (vérifiant $x \in \{0, 1\}^n$ et $Ax = b$)

$\Rightarrow$ On reconnaît les contraintes "produit" de l'approche linéaire d'Adams-Sharali et qui jouent également un rôle important dans la formulation semidéfinie de la relaxation Lagrangienne partielle de (QL) !

Trouver la meilleure combinaison linéaire ?

Convexification d'un programme quadratique (2)

Amélioration de l'approche :

$\Rightarrow$ Ajouter une combinaison linéaire des fonctions $h_i(x) = x_i^2 - x_i$ ($i = 1, \ldots, n$) et $g_{ij}(x) = \sum_{k=1}^{n} A_{jk} x_k x_i - b_j x_i$ ($j = 1, \ldots, p$ et $i = 1, \ldots, n$) à la fonction $x^T Q x + c^T x$: fonction nulle pour x admissible (vérifiant $x \in \{0,1\}^n$ et $Ax = b$)

$\Rightarrow$ On reconnaît les contraintes "produit" de l'approche linéaire d'Adams-Sharali et qui jouent également un rôle important dans la formulation semidéfinie de la relaxation Lagrangienne partielle de (QL) !

Trouver la meilleure combinaison linéaire ?

$$\sup_{\alpha,\beta} \inf_{x \in [0,1]^n \cap \{x : Ax = b\}} \quad x^T Q x + c^T x + \sum_{i=1}^{n} \alpha_i \left(x_i^2 - x_i\right) + \sum_{j=1}^{p} \sum_{i=1}^{n} \beta_{ij} \left(\sum_{k=1}^{n} A_{jk} x_k x_i - b_j x_i\right)$$

Convexification d'un programme quadratique (2)

Amélioration de l'approche :

$\Rightarrow$ Ajouter une combinaison linéaire des fonctions $h_i(x) = x_i^2 - x_i$ ($i = 1, \dots, n$) et $g_{ij}(x) = \sum_{k=1}^n A_{jk} x_k x_i - b_j x_i$ ($j = 1, \dots, p$ et $i = 1, \dots, n$) à la fonction $x^T Q x + c^T x$: fonction nulle pour x admissible (vérifiant $x \in \{0,1\}^n$ et $Ax = b$)

$\Rightarrow$ On reconnaît les contraintes "produit" de l'approche linéaire d'Adams-Sharali et qui jouent également un rôle important dans la formulation semidéfinie de la relaxation Lagrangienne partielle de (QL) !

Trouver la meilleure combinaison linéaire ?

$$\sup_{\alpha,\beta} \inf_{x \in [0,1]^n \cap \{x : Ax = b\}} \quad x^T Q x + c^T x + \sum_{i=1}^n \alpha_i \left(x_i^2 - x_i\right) + \sum_{j=1}^p \sum_{i=1}^n \beta_{ij} \left(\sum_{k=1}^n A_{jk} x_k x_i - b_j x_i\right)$$

Le dual d'une formulation équivalente de ce problème peut s'interpréter comme un programme semidéfini !

Convexification d'un programme quadratique (3)

$$\sup_{\alpha,\beta} \inf_{x\in[0,1]^n\cap\{x:Ax=b\}} \quad x^TQx + c^Tx + \sum_{i=1}^n \alpha_i\left(x_i^2 - x_i\right) + \sum_{j=1}^p \sum_{i=1}^n \beta_{ij}\left(\sum_{k=1}^n A_{jk}x_kx_i - b_jx_i\right)$$

On remplace $x \in [0,1]^n$ par les contraintes convexes $x_i^2 - x_i \leq 0$

Convexification d'un programme quadratique (3)

$$\sup_{\alpha,\beta} \inf_{x\in[0,1]^n\cap\{x:Ax=b\}} \quad \begin{array}{l} x^TQx + c^Tx + \sum_{i=1}^n \alpha_i \left(x_i^2 - x_i\right) \\ +\sum_{j=1}^p \sum_{i=1}^n \beta_{ij} \left(\sum_{k=1}^n A_{jk}x_kx_i - b_jx_i\right) \end{array}$$

On remplace $x \in [0,1]^n$ par les contraintes convexes $x_i^2 - x_i \leq 0$

$$(SDP_P) \begin{cases} \min & Q \bullet X + c^Tx \\ \text{s.c.} & Ax = b \\ & \sum_{k=1}^n A_{jk}X_{ki} - b_jx_i = 0 \quad i \in \{1,\ldots,n\}\ j \in \{1,\ldots,p\} \\ & X_{ii} = x_i \quad (i \in \{1,\ldots,n\}) \\ & \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0 \end{cases}$$

Convexification d'un programme quadratique (4)

- Conséquence : Le programme quadratique convexe obtenu a la même valeur que le programme semidéfini (SDP_P)

Convexification d'un programme quadratique (4)

- Conséquence : Le programme quadratique convexe obtenu a la même valeur que le programme semidéfini (SDP_P)
- Intérêt : Utiliser ce programme quadratique convexe dans une méthode de séparation/évaluation (Branch&Bound) à la place de la PSD !

Convexification d’un programme quadratique (4)

- Conséquence : Le programme quadratique convexe obtenu a la même valeur que le programme semidéfini (SDP_P)
- Intérêt : Utiliser ce programme quadratique convexe dans une méthode de séparation/évaluation (Branch&Bound) à la place de la PSD !
- Résolution d’un seul PSD : on a "capté" la qualité de l’approche semidéfinie dans une approche quadratique convexe.

Plan du cours

1. Introduction

2. Linéarisations

3. Approches Lagrangiennes

4. Approche par Programmation Semidéfinie
- a. Bases de la Programmation Semidéfinie
- b. Application à la Programmation Quadratique en 0-1
  - 1. Le nombre de Lovász
  - 2. Relaxation SDP en $\{0, 1\}$
  - 3. Relaxation SDP en $\{-1, 1\}$
  - 4. Liens avec l'approche Lagrangienne
  - 5. Amélioration des relaxations Quadratiques Convexes
  - 6. Problèmes SDLS et contrainte sphérique

4.b.6. Problèmes SDLS et contrainte sphérique

SDLS : SemiDefinite Least Squares

Programme Quadratique en variables bivalentes :

$$
(Q) \quad \left\{ \begin{array}{lll} \max & Q \bullet X & \\ \text{s.c.} & Q_i \bullet X \geq 0 & i \in I = \{1, \ldots, m\} \\ & x_i \in \{-1, 1\} & \forall i \in \{1, \ldots, n\} \\ & X = xx^T & \end{array} \right.
$$

4.b.6. Problèmes SDLS et contrainte sphérique

SDLS : SemiDefinite Least Squares

Programme Quadratique en variables bivalentes :

$$(Q) \quad \begin{cases} \max & Q \bullet X \\ \text{s.c.} & Q_i \bullet X \geq 0 \quad\quad i \in I = \{1, \ldots, m\} \\ & x_i \in \{-1, 1\} \quad \forall i \in \{1, \ldots, n\} \\ & X = xx^T \end{cases}$$

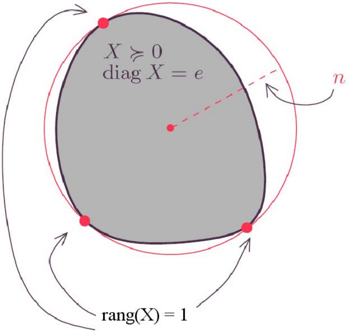
X est de rang 1 et appartient à

$\mathcal{C} = \left\{X \in S_n \,:\, X \succcurlyeq 0, X_{jj} = 1 \forall j \in \{1, \ldots, n\}\right\}$

La Contrainte Sphérique

Théorème

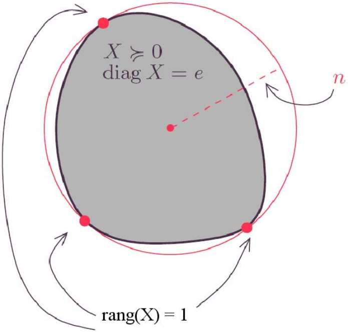
$X \in \mathcal{C}$ est de rang 1 est équivalent à
$\|X\|^2 = X \bullet X = \sum_{i=1}^{n} \sum_{j=1}^{n} X_{ij}^2 = n^2$



La Contrainte Sphérique

Théorème

$X \in \mathcal{C}$ est de rang 1 est équivalent à
$\|X\|^2 = X \bullet X = \sum_{i=1}^{n} \sum_{j=1}^{n} X_{ij}^2 = n^2$



- X appartient donc à la sphère de centre 0 et de rayon n (au sens de la norme associée à $\bullet$)

Dualiser la Contrainte Sphérique

- Idée : remplacer $X = xx^T$ par $\|X\|^2 = n^2$ dans (Q) puis dualiser cette dernière contrainte

Dualiser la Contrainte Sphérique

- Idée : remplacer $X = xx^T$ par $\|X\|^2 = n^2$ dans (Q) puis dualiser cette dernière contrainte
- α : le multiplicateur de Lagrange associé

Dualiser la Contrainte Sphérique

- Idée : remplacer $X = xx^T$ par $\|X\|^2 = n^2$ dans (Q) puis dualiser cette dernière contrainte
- α : le multiplicateur de Lagrange associé
- Pour $\alpha = 0$, on retrouve la relaxation semidéfinie de base

Dualiser la Contrainte Sphérique

- Idée : remplacer $X = xx^T$ par $\|X\|^2 = n^2$ dans (Q) puis dualiser cette dernière contrainte
- α : le multiplicateur de Lagrange associé
- Pour $\alpha = 0$, on retrouve la relaxation semidéfinie de base
- Pour $\alpha \neq 0$ Le Lagrangien vaut : $L(X,\alpha) = Q \bullet X + \frac{\alpha}{2}\left(n^2 - X \bullet X\right)$
$= \frac{\alpha}{2}n^2 + \frac{\alpha}{2}\left(\frac{Q \bullet Q}{\alpha^2} - \left(X - \frac{Q}{\alpha}\right) \bullet \left(X - \frac{Q}{\alpha}\right)\right)$
$= \frac{\alpha}{2}n^2 + \frac{1}{2\alpha}\|Q\|^2 - \frac{\alpha}{2}\|X - \frac{Q}{\alpha}\|^2$

Dualiser la Contrainte Sphérique

- Idée : remplacer $X = xx^T$ par $\|X\|^2 = n^2$ dans (Q) puis dualiser cette dernière contrainte
- α : le multiplicateur de Lagrange associé
- Pour $\alpha = 0$, on retrouve la relaxation semidéfinie de base
- Pour $\alpha \neq 0$ Le Lagrangien vaut : $L(X, \alpha) = Q \bullet X + \frac{\alpha}{2}\left(n^2 - X \bullet X\right)$
  $= \frac{\alpha}{2} n^2 + \frac{\alpha}{2}\left(\frac{Q \bullet Q}{\alpha^2} - \left(X - \frac{Q}{\alpha}\right) \bullet \left(X - \frac{Q}{\alpha}\right)\right)$
  $= \frac{\alpha}{2} n^2 + \frac{1}{2\alpha}\|Q\|^2 - \frac{\alpha}{2}\|X - \frac{Q}{\alpha}\|^2$
- Pour $\alpha < 0$ résoudre le problème dual est NP-difficile (concavité).

Dualiser la Contrainte Sphérique

- Idée : remplacer $X = xx^T$ par $\|X\|^2 = n^2$ dans (Q) puis dualiser cette dernière contrainte
- α : le multiplicateur de Lagrange associé
- Pour $\alpha = 0$, on retrouve la relaxation semidéfinie de base
- Pour $\alpha \neq 0$ Le Lagrangien vaut : $L(X, \alpha) = Q \bullet X + \frac{\alpha}{2}\left(n^2 - X \bullet X\right)$
$= \frac{\alpha}{2}n^2 + \frac{\alpha}{2}\left(\frac{Q \bullet Q}{\alpha^2} - \left(X - \frac{Q}{\alpha}\right) \bullet \left(X - \frac{Q}{\alpha}\right)\right)$
$= \frac{\alpha}{2}n^2 + \frac{1}{2\alpha}\|Q\|^2 - \frac{\alpha}{2}\|X - \frac{Q}{\alpha}\|^2$
- Pour $\alpha < 0$ résoudre le problème dual est NP-difficile (concavité).
- Pour $\alpha > 0$: Soit $\aleph = \{X \ : \ Q_i \bullet X \geq 0 \, \forall i\} \cap \left\{X \succcurlyeq 0, \ X_{jj} = 1 \, \forall j\right\}$.
Le programme dual $\inf_\alpha \sup_{X \in \aleph} L(X, \alpha)$ est équivalent à un Problème semidéfini de moindres carrés : projection de $\frac{Q}{\alpha}$ sur $\aleph$

Dualiser la Contrainte Sphérique

- Idée : remplacer $X = xx^T$ par $\|X\|^2 = n^2$ dans (Q) puis dualiser cette dernière contrainte
- α : le multiplicateur de Lagrange associé
- Pour $\alpha = 0$, on retrouve la relaxation semidéfinie de base
- Pour $\alpha \neq 0$ Le Lagrangien vaut : $L(X, \alpha) = Q \bullet X + \frac{\alpha}{2}\left(n^2 - X \bullet X\right)$
  $= \frac{\alpha}{2} n^2 + \frac{\alpha}{2}\left(\frac{Q \bullet Q}{\alpha^2} - \left(X - \frac{Q}{\alpha}\right) \bullet \left(X - \frac{Q}{\alpha}\right)\right)$
  $= \frac{\alpha}{2} n^2 + \frac{1}{2\alpha}\|Q\|^2 - \frac{\alpha}{2}\|X - \frac{Q}{\alpha}\|^2$
- Pour $\alpha < 0$ résoudre le problème dual est NP-difficile (concavité).
- Pour $\alpha > 0$: Soit $\aleph = \{X \,:\, Q_i \bullet X \geq 0 \, \forall i\} \cap \{X \succcurlyeq 0,\, X_{jj} = 1 \, \forall j\}$.
  Le programme dual $\inf_\alpha \sup_{X \in \aleph} L(X, \alpha)$ est équivalent à un Problème semidéfini de moindres carrés : projection de $\frac{Q}{\alpha}$ sur $\aleph$
- La borne obtenue est moins bonne que celle de la PSD standard, mais ces problèmes peuvent être résolus plus efficacement que les PSDs : échange temps de calcul contre qualité de la borne

Contrainte Sphérique : un exemple

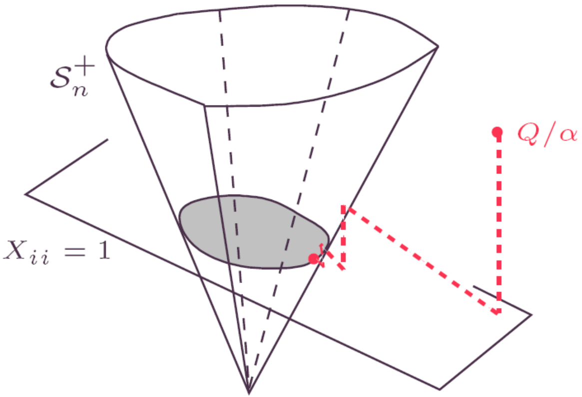
Pour le problème MAX-CUT :

$$(MC) \left\{ \begin{array}{lll} \max & Q \bullet X & \\ \text{s.c.} & X_{ii} = 1 & \forall i \in \{1, \ldots, n\} \\ & X = xx^T (\text{ou } X \bullet X = n^2) & \end{array} \right.$$

On obtient pour $\alpha > 0$:

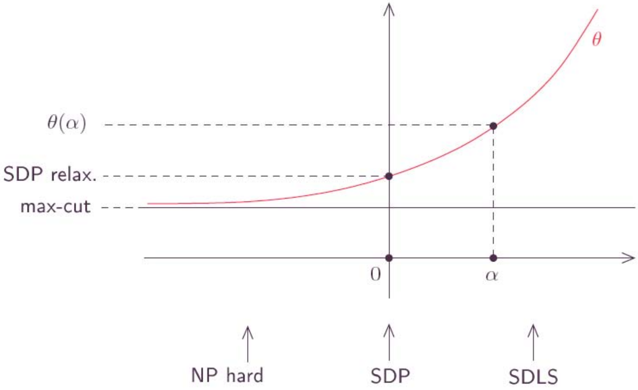
$$(SDPLS(\alpha)) \left\{ \begin{array}{lll} \min & \|X - \frac{Q}{\alpha}\| & \\ \text{s.c.} & X_{ii} = 1 & \forall i \in \{1, \ldots, n\} \\ & X \succcurlyeq 0 & \end{array} \right.$$

Problème semidéfini de moindres carrés



Projection de la matrice $\frac{Q}{\alpha}$ sur l'intersection du cône des matrices positives et d'un espace affine

Détériorer la qualité de la borne contre du temps de calcul



Lorsque α tend vers 0 on retrouve la valeur de la relaxation semidéfinie de base de max-cut