Programmation Semidéfinie en Optimisation Combinatoire

Frédéric Roupin

roupin@cnam.fr

Cédric

Conservatoire National des Arts et Métiers

Journées Franciliennes de Recherche Opérationnelle
15 Juin 2007

Plan de la Présentation

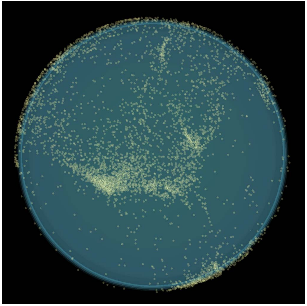
Plan de la Présentation

Plan de la Présentation

Plan de la Présentation

Plan de la Présentation

Plan de la Présentation



Utilisation de la Programmation Semidéfinie pour visualiser des groupes d'artistes semblables (Yahoo/MIT)

Un peu d'Histoire...

- Bellman et Fan, 1963 : "On systems of linear inequalities in Hermitian matrix variables". Optimisation sur un cône

Un peu d'Histoire...

- Bellman et Fan, 1963 : "On systems of linear inequalities in Hermitian matrix variables". Optimisation sur un cône
- Yakubovich, 1965 : "The method of matrix inequalities in the stability theory of nonlinear control systems"

Un peu d'Histoire...

- Bellman et Fan, 1963 : "On systems of linear inequalities in Hermitian matrix variables". Optimisation sur un cône
- Yakubovich, 1965 : "The method of matrix inequalities in the stability theory of nonlinear control systems"
- Lovász 1979 : "On the shannon capacity of a graph". Fonction $\Theta(G)$

Un peu d'Histoire...

- Bellman et Fan, 1963 : "On systems of linear inequalities in Hermitian matrix variables". Optimisation sur un cône
- Yakubovich, 1965 : "The method of matrix inequalities in the stability theory of nonlinear control systems"
- Lovász 1979 : "On the shannon capacity of a graph". Fonction $\Theta(G)$
- Shor 1987 : "Quadratic optimization problems"

Un peu d'Histoire...

- Bellman et Fan, 1963 : "On systems of linear inequalities in Hermitian matrix variables". Optimisation sur un cône
- Yakubovich, 1965 : "The method of matrix inequalities in the stability theory of nonlinear control systems"
- Lovász 1979 : "On the shannon capacity of a graph". Fonction $\Theta(G)$
- Shor 1987 : "Quadratic optimization problems"
- Lovász et Schrijver 1991 : Hiérarchie de relaxations linéaires et semidéfinies

Un peu d'Histoire...

- Bellman et Fan, 1963 : "On systems of linear inequalities in Hermitian matrix variables". Optimisation sur un cône
- Yakubovich, 1965 : "The method of matrix inequalities in the stability theory of nonlinear control systems"
- Lovász 1979 : "On the shannon capacity of a graph". Fonction $\Theta(G)$
- Shor 1987 : "Quadratic optimization problems"
- Lovász et Schrijver 1991 : Hiérarchie de relaxations linéaires et semidéfinies
- Goemans et Williamson 1995 : Approximation de MAX-CUT

Un peu d'Histoire...

- Bellman et Fan, 1963 : "On systems of linear inequalities in Hermitian matrix variables". Optimisation sur un cône
- Yakubovich, 1965 : "The method of matrix inequalities in the stability theory of nonlinear control systems"
- Lovász 1979 : "On the shannon capacity of a graph". Fonction $\Theta(G)$
- Shor 1987 : "Quadratic optimization problems"
- Lovász et Schrijver 1991 : Hiérarchie de relaxations linéaires et semidéfinies
- Goemans et Williamson 1995 : Approximation de MAX-CUT
- Poljak, Rendl et Wolkowicz 1995 : Recette pour la Programmation Quadratique en 0-1

Un peu d'Histoire...

- Bellman et Fan, 1963 : "On systems of linear inequalities in Hermitian matrix variables". Optimisation sur un cône
- Yakubovich, 1965 : "The method of matrix inequalities in the stability theory of nonlinear control systems"
- Lovász 1979 : "On the shannon capacity of a graph". Fonction $\Theta(G)$
- Shor 1987 : "Quadratic optimization problems"
- Lovász et Schrijver 1991 : Hiérarchie de relaxations linéaires et semidéfinies
- Goemans et Williamson 1995 : Approximation de MAX-CUT
- Poljak, Rendl et Wolkowicz 1995 : Recette pour la Programmation Quadratique en 0-1
- Lemaréchal et Oustry 1999 : Semidefinite relaxations and Lagrangian duality with application to combinatorial optimization

Outils de résolution numérique

- Nesterov Nemirosky 1994 : Méthodes de points intérieurs pour la programmation convexe, SIAM.

Outils de résolution numérique

- Nesterov Nemirosky 1994 : Méthodes de points intérieurs pour la programmation convexe, SIAM.
- Fujisawa et al 1998 : SDPA (point intérieur, primal-dual) SDPARA-C 2003 (parallèle)

Outils de résolution numérique

- Nesterov Nemirosky 1994 : Méthodes de points intérieurs pour la programmation convexe, SIAM.
- Fujisawa et al 1998 : SDPA (point intérieur, primal-dual) SDPARA-C 2003 (parallèle)
- Sturm 1999 : SeDuMi (point intérieur, primal-dual, MATLAB toolbox)

Outils de résolution numérique

- Nesterov Nemirosky 1994 : Méthodes de points intérieurs pour la programmation convexe, SIAM.
- Fujisawa et al 1998 : SDPA (point intérieur, primal-dual) SDPARA-C 2003 (parallèle)
- Sturm 1999 : SeDuMi (point intérieur, primal-dual, MATLAB toolbox)
- Borchers 1999 : CSDP (point intérieur, primal-dual)

Outils de résolution numérique

- Nesterov Nemirosky 1994 : Méthodes de points intérieurs pour la programmation convexe, SIAM.
- Fujisawa et al 1998 : SDPA (point intérieur, primal-dual) SDPARA-C 2003 (parallèle)
- Sturm 1999 : SeDuMi (point intérieur, primal-dual, MATLAB toolbox)
- Borchers 1999 : CSDP (point intérieur, primal-dual)
- Helmberg 1999 : SB (Spectral Bundle Method)

Outils de résolution numérique

- Nesterov Nemirosky 1994 : Méthodes de points intérieurs pour la programmation convexe, SIAM.
- Fujisawa et al 1998 : SDPA (point intérieur, primal-dual) SDPARA-C 2003 (parallèle)
- Sturm 1999 : SeDuMi (point intérieur, primal-dual, MATLAB toolbox)
- Borchers 1999 : CSDP (point intérieur, primal-dual)
- Helmberg 1999 : SB (Spectral Bundle Method)
- Burer 2003 : SDPLR (Low Rank Factorization)

Outils de résolution numérique

- Nesterov Nemirosky 1994 : Méthodes de points intérieurs pour la programmation convexe, SIAM.
- Fujisawa et al 1998 : SDPA (point intérieur, primal-dual) SDPARA-C 2003 (parallèle)
- Sturm 1999 : SeDuMi (point intérieur, primal-dual, MATLAB toolbox)
- Borchers 1999 : CSDP (point intérieur, primal-dual)
- Helmberg 1999 : SB (Spectral Bundle Method)
- Burer 2003 : SDPLR (Low Rank Factorization)
- Stingl et Kocvara 2005 : PENNON (Lagrangien augmenté)

Plan de la Présentation

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 $x \in \Re^n$
$x^T A x = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} x_i x_j = A \bullet xx^T \geq 0$

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

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 : $A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 5 & 0 \\ 1 & 0 & 3 \end{bmatrix}$

- mineurs symétriques 1×1 : $1 \geq 0$, $5 \geq 0$, $3 \geq 0$

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 : $A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 5 & 0 \\ 1 & 0 & 3 \end{bmatrix}$

- mineurs symétriques 1×1 : $1 \geq 0$, $5 \geq 0$, $3 \geq 0$
- mineurs symétriques 2×2 : $5 - 4 \geq 0$, $15 - 0 \geq 0$, $3 - 1 \geq 0$

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 : $A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 5 & 0 \\ 1 & 0 & 3 \end{bmatrix}$

- mineurs symétriques 1×1 : $1 \geq 0$, $5 \geq 0$, $3 \geq 0$
- mineurs symétriques 2×2 : $5 - 4 \geq 0$, $15 - 0 \geq 0$, $3 - 1 \geq 0$
- mineur symétrique 3×3 : $\det(A) = 5 \geq 0$

Programmation Semidéfinie

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

Programmation Semidéfinie

- 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 $F(x) = \sum_{i=1}^{m} x_i A_i - A_0$

Programmation Semidéfinie

- 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 $F(x) = \sum_{i=1}^{m} x_i A_i - A_0$

Programmation Semidéfinie

- 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

Programmation Semidéfinie

- 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

Programmation Semidéfinie

- 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

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

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

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

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 :
  $F\left(\lambda x + (1-\lambda)\,y\right) = \lambda F(x) + (1-\lambda)\,F(y) \succeq 0$
  $\forall\ \lambda \in [0,1]\,,\ x \in S,\ \ y \in S$

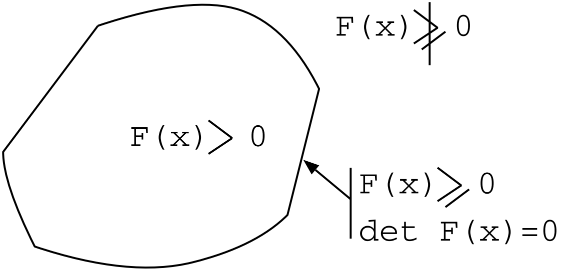
Etudions $S = \left\{ x \in \Re^m \;:\; F(x) = \sum_{i=1}^{m} x_i A_i - A_0 \succcurlyeq 0 \right\}$
S est défini par une inégalité matricielle linéaire

- $F(x)$ 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 $\begin{bmatrix} x & 1 \\ 1 & y \end{bmatrix} \succeq 0$
  $\Longleftrightarrow x \geq 0,\; y \geq 0,\; xy \geq 1$ (mineurs symétriques)

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

Théorème

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



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

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

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 $F(x)$

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* $C - B^T A^{-1} B \succcurlyeq 0$.

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

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* $C - B^T A^{-1} B \succcurlyeq 0$.

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

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 $C - B^T A^{-1} B \succcurlyeq 0$.

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

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 $\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succcurlyeq 0$ *est équivalent à* $C - B^T A^{-1} B \succcurlyeq 0$.

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

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

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

Programmation Quadratique Convexe

Proposition

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

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

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

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

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

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

Programmation Quadratique Convexe

$$(Q)\begin{cases} \min & x^T A_0^T A_0 x + c_0^T x + d_0 (\leq t) \\ \text{s.c.} & x^T A_i^T A_i x + c_i^T x + d_i \leq 0 \qquad i = 1, \ldots, m \end{cases}$$

Utilisation d'une unique inégalité matricielle linéaire :

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

Dualité

- Le programme dual de (SDP) est
$$\max_{Z \succcurlyeq 0} \inf_{x \in \Re^n} L(x, Z)$$
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

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

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

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

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 $Ax - b \geq 0$: $L(x, Z) = c^T x + \lambda^T (b - Ax)$ avec $\lambda \geq 0$

Dualité

- Le programme dual de (SDP) est
$$\max_{Z \succcurlyeq 0} \inf_{x \in \Re^n} L(x, Z)$$
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 $Ax - b \geq 0$: $L(x, Z) = c^T x + \lambda^T (b - Ax)$ avec $\lambda \geq 0$
- Fonction duale :

Dualité

- Le programme dual de (SDP) est

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

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 $Ax - b \geq 0$: $L(x, Z) = c^T x + \lambda^T (b - Ax)$ 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

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

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 $Ax - b \geq 0$: $L(x, Z) = c^T x + \lambda^T (b - Ax)$ 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
$$\max_{Z \succcurlyeq 0} \inf_{x \in \Re^n} L(x, Z)$$
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 $Ax - b \geq 0$: $L(x, Z) = c^T x + \lambda^T (b - Ax)$ 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 :
$$(DSDP) \left\{ \begin{array}{lll} \max & A_0 \bullet Z & \\ \text{s.c.} & A_i \bullet Z = c_i & i \in \{1, \ldots, m\} \\ & Z \succeq 0 & \end{array} \right.$$

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 :

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

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 :

$c^T x - A_0 \bullet Z = \sum_{i=1}^{m} (A_i \bullet Z) x_i - A_0 \bullet Z = F(x) \bullet Z \geq 0$

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

Plan de la Présentation

Plan de la Présentation

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).

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

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

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 $E^{ij}_{kl} = 0$

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 $\{\dots, E_{ij}, \dots\}$

Le nombre de Lovász

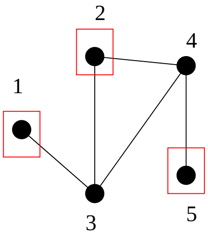
Le problème du STABLE

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).

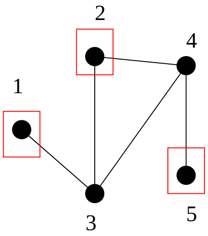
- Soit $\mathfrak{M}(G) = \{A \in S_n \ : \ A_{ij} = 1 \,\mathrm{si}\, (i,j) \notin E \,\mathrm{ou}\, (i = j)\}$
- $\mathfrak{M}(G)$ est un espace affine
- Pour tout $[v_i,\ v_j] \in E$ on définit E^{ij} matrice de S_n telle que $E^{ij}_{ij} = E^{ij}_{ji} = 1$ et sinon $E^{ij}_{kl} = 0$
- Repère de l'espace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

Exemple :



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

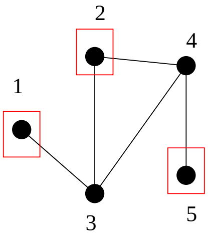
Exemple :



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

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

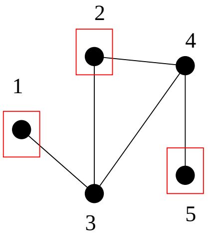
Exemple :



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

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)

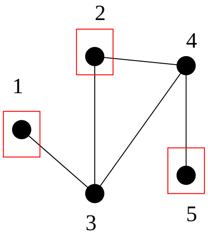
Exemple :



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

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 $\lambda_{max}(A) \geq \lambda_{max}(J_k) = k$.

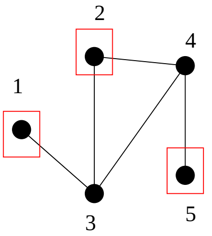
Exemple :



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

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 :



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

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 $\lambda_{max}(A) \geq \lambda_{max}(J_k) = k$.
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

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

Dual Semidéfini

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

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

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

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 pourra obtenir $\alpha(G)$ 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 pourra obtenir $\alpha(G)$ dans les graphes parfaits !
- Et même $\chi(G)$ car le complément d'un graphe parfait est parfait ("weak perfect graph conjecture", Lovász)

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

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

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

- 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 $v^k_j = 0\ \forall k \neq i$

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

- 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 $v^k_j = 0\ \forall k \neq i$
- $X \succcurlyeq 0$ 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 $v^k_j = 0\ \forall k \neq i$
- $X \succcurlyeq 0$ 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 $v^k_j = 0 \; \forall k \neq i$
- $X \succcurlyeq 0$ 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 $\left(\sum_{i=1}^k v^i\right)^T Ze_n = e_n e_n^T \bullet Z$

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

- 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 $v^k_j = 0 \ \forall k \neq i$
- $X \succcurlyeq 0$ 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 $\left(\sum_{i=1}^k v^i\right)^T Ze_n = e_n e_n^T \bullet Z$
- Donc $0 \leq k I_n \bullet Z - J_n \bullet Z$ donc $J_n \bullet Z \leq \chi(\bar{G})$

Plan de la Présentation

Questions :

- Comment concevoir une relaxation Semidéfinie ?

Questions :

- Comment concevoir une relaxation Semidéfinie ?
- Comment améliorer une relaxation Semidéfinie ?

Questions :

- Comment concevoir une relaxation Semidéfinie ?
- Comment améliorer une relaxation Semidéfinie ?
- Que deviennent les approches utilisées en Programmation Linéaire ?

Amélioration de la Programmation Linéaire

Programmation Quadratique en variables 0-1

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

Amélioration de la Programmation Linéaire

Programmation Quadratique en variables 0-1

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

Une linéarisation classique (Dantzig) :

- Remplacer $x \in \{0,1\}^n$ par $0 \leq x_i \leq 1\ \forall i$

Amélioration de la Programmation Linéaire

Programmation Quadratique en variables 0-1

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

Une linéarisation classique (Dantzig) :

- Remplacer $x \in \{0,1\}^n$ par $0 \leq x_i \leq 1\ \forall i$
- Remplacer x_ix_j par Y_{ij}

Amélioration de la Programmation Linéaire

Programmation Quadratique en variables 0-1

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

Une linéarisation classique (Dantzig) :

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

Chercheur (pas très) Opérationnel passant de la Programmation Linéaire à la Programmation Semidéfinie

Mettons un peu d'ordre : rangeons ces variables de linéarisation dans une matrice !

Passage de l'approche linéaire à l'approche semidéfinie

- Dans l'approche linéaire précédente :

Passage de l'approche linéaire à l'approche semidéfinie

- Dans l'approche linéaire précédente :
- Y représente xx^T si on pose $Y_{ii} = x_i$, car $x_i^2 = x_i \Leftrightarrow x \in \{0,1\}^n$ pour tout i. Rappel : $\left(xx^T\right)_{ij} = x_i x_j \ \forall i, j$

Passage de l'approche linéaire à l'approche semidéfinie

- Dans l'approche linéaire précédente :
- Y représente xx^T si on pose $Y_{ii} = x_i$, car $x_i^2 = x_i \Leftrightarrow x \in \{0,1\}^n$ pour tout i. Rappel : $\left(xx^T\right)_{ij} = x_i x_j \; \forall i,j$
- Idéal (problème initial) : $Y = xx^T$ (mais pas convexe !)

Passage de l'approche linéaire à l'approche semidéfinie

- Dans l'approche linéaire précédente :
- Y représente xx^T si on pose $Y_{ii} = x_i$, car $x_i^2 = x_i \Leftrightarrow x \in \{0,1\}^n$ pour tout i. Rappel : $\left(xx^T\right)_{ij} = x_i x_j \; \forall i, j$
- Idéal (problème initial) : $Y = xx^T$ (mais pas convexe !)
- Relaxation convexe : $Y - xx^T \succcurlyeq 0 \Leftrightarrow \begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} \succcurlyeq 0$

Passage de l'approche linéaire à l'approche semidéfinie

- Dans l'approche linéaire précédente :
- Y représente xx^T si on pose $Y_{ii} = x_i$, car $x_i^2 = x_i \Leftrightarrow x \in \{0,1\}^n$ pour tout i. Rappel : $\left(xx^T\right)_{ij} = x_i x_j \ \forall i,j$
- Idéal (problème initial) : $Y = xx^T$ (mais pas convexe !)
- Relaxation convexe : $Y - xx^T \succcurlyeq 0 \Leftrightarrow \begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} \succcurlyeq 0$

Passage de l'approche linéaire à l'approche semidéfinie

- Dans l'approche linéaire précédente :
- Y représente xx^T si on pose $Y_{ii} = x_i$, car $x_i^2 = x_i \Leftrightarrow x \in \{0,1\}^n$ pour tout i. Rappel : $\left(xx^T\right)_{ij} = x_i x_j \; \forall i,j$
- Idéal (problème initial) : $Y = xx^T$ (mais pas convexe !)
- Relaxation convexe : $Y - xx^T \succcurlyeq 0 \Leftrightarrow \begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} \succcurlyeq 0$

On peut :

- Remplacer les contraintes de linéarisation par $Y - xx^T \succcurlyeq 0$

Passage de l'approche linéaire à l'approche semidéfinie

- Dans l'approche linéaire précédente :
- Y représente xx^T si on pose $Y_{ii} = x_i$, car $x_i^2 = x_i \Leftrightarrow x \in \{0,1\}^n$ pour tout i. Rappel : $\left(xx^T\right)_{ij} = x_i x_j \ \forall i,j$
- Idéal (problème initial) : $Y = xx^T$ (mais pas convexe !)
- Relaxation convexe : $Y - xx^T \succcurlyeq 0 \Leftrightarrow \begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} \succcurlyeq 0$

On peut :

- Remplacer les contraintes de linéarisation par $Y - xx^T \succcurlyeq 0$
- Ajouter cette contrainte à la linéarisation précédente

Passage de l'approche linéaire à l'approche semidéfinie

- Dans l'approche linéaire précédente :
- Y représente xx^T si on pose $Y_{ii} = x_i$, car $x_i^2 = x_i \Leftrightarrow x \in \{0,1\}^n$ pour tout i. Rappel : $\left(xx^T\right)_{ij} = x_i x_j \ \forall i,j$
- Idéal (problème initial) : $Y = xx^T$ (mais pas convexe !)
- Relaxation convexe : $Y - xx^T \succcurlyeq 0 \Leftrightarrow \begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} \succcurlyeq 0$

On peut :

- Remplacer les contraintes de linéarisation par $Y - xx^T \succcurlyeq 0$
- Ajouter cette contrainte à la linéarisation précédente
- Les contraintes de linéarisation précédentes ne sont pas toujours satisfaites par (Y, x) tel que $Y - xx^T \succcurlyeq 0$ et $Y_{ii} = x_i \ \forall i \in \{1, \ldots, n\}$ [Laurent Poljak Rendl 1997]

Passage de l'approche linéaire à l'approche semidéfinie

- Dans l'approche linéaire précédente :
- Y représente xx^T si on pose $Y_{ii} = x_i$, car $x_i^2 = x_i \Leftrightarrow x \in \{0,1\}^n$ pour tout i. Rappel : $\left(xx^T\right)_{ij} = x_i x_j \; \forall i,j$
- Idéal (problème initial) : $Y = xx^T$ (mais pas convexe !)
- Relaxation convexe : $Y - xx^T \succcurlyeq 0 \Leftrightarrow \begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} \succcurlyeq 0$

On peut :

- Remplacer les contraintes de linéarisation par $Y - xx^T \succcurlyeq 0$
- Ajouter cette contrainte à la linéarisation précédente
- Les contraintes de linéarisation précédentes ne sont pas toujours satisfaites par (Y, x) tel que $Y - xx^T \succcurlyeq 0$ et $Y_{ii} = x_i \; \forall i \in \{1, \ldots, n\}$ [Laurent Poljak Rendl 1997]
- On peut donc partir de relaxations linéaires pour construire des relaxations semidéfinies [Roupin 2004]

Programme Semidéfini obtenu

Relaxation Semidéfinie = Relaxation Linéaire + contrainte semidéfinie

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

Les contraintes $0 \leq x_i \leq 1$ n'ont pas disparu car $\det \begin{bmatrix} 1 & x_i \\ x_i & x_i \end{bmatrix} \geq 0$ pour tout i dans $\{1, \ldots, n\}$ et donc $0 \leq x_i^2 \leq x_i \leq 1$

Hiérarchies de relaxations linéaires et semidéfinies

- Les méthodes de Lift and Project développées pour la programmation linéaire s'adaptent et s'étendent à la programmation semidéfinie

Hiérarchies de relaxations linéaires et semidéfinies

- Les méthodes de Lift and Project développées pour la programmation linéaire s'adaptent et s'étendent à la programmation semidéfinie
- [Adams Sherali 1990], [Lovász Schrijver 1991], [Lasserre 2001] ont proposé des hiérarchies de relaxations linéaires ou semidéfinies sur le domaine $\mathcal{P} = \{0,1\}^n \cap \{x \,:\, Ax \leq b\}$

Hiérarchies de relaxations linéaires et semidéfinies

- Les méthodes de Lift and Project développées pour la programmation linéaire s'adaptent et s'étendent à la programmation semidéfinie
- [Adams Sherali 1990], [Lovász Schrijver 1991], [Lasserre 2001] ont proposé des hiérarchies de relaxations linéaires ou semidéfinies sur le domaine $\mathcal{P} = \{0,1\}^n \cap \{x : Ax \leq b\}$
- Un cadre homogène de présentation et de comparaison de ces hiérarchies est donné dans [Laurent 2001]

Hiérarchies de relaxations linéaires et semidéfinies

- Les méthodes de Lift and Project développées pour la programmation linéaire s'adaptent et s'étendent à la programmation semidéfinie
- [Adams Sherali 1990], [Lovász Schrijver 1991], [Lasserre 2001] ont proposé des hiérarchies de relaxations linéaires ou semidéfinies sur le domaine $\mathcal{P} = \{0,1\}^n \cap \{x \,:\, Ax \leq b\}$
- Un cadre homogène de présentation et de comparaison de ces hiérarchies est donné dans [Laurent 2001]
- **Principe.** approcher l'enveloppe convexe de $\mathcal{P}$ par la projection d'un polytope de dimension supérieure

Hiérarchies de relaxations linéaires et semidéfinies

- Les méthodes de Lift and Project développées pour la programmation linéaire s'adaptent et s'étendent à la programmation semidéfinie
- [Adams Sherali 1990], [Lovász Schrijver 1991], [Lasserre 2001] ont proposé des hiérarchies de relaxations linéaires ou semidéfinies sur le domaine $\mathcal{P} = \{0,1\}^n \cap \{x \,:\, Ax \leq b\}$
- Un cadre homogène de présentation et de comparaison de ces hiérarchies est donné dans [Laurent 2001]
- **Principe.** approcher l'enveloppe convexe de $\mathcal{P}$ par la projection d'un polytope de dimension supérieure
- Généralisation des méthodes de "lift-and-project" proposées dans l'approche linéaire : coupes quadratiques convexes [Iyengar, 2001]

Retour au nombre de Lovász

Formulation du problème du stable en programme linéaire en variables 0-1 : $x_i = 1 \Leftrightarrow$ le sommet v_i est dans le stable

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

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

Retour au nombre de Lovász

Formulation du problème du stable en programme linéaire en variables 0-1 : $x_i = 1 \Leftrightarrow$ le sommet v_i est dans le stable

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

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

Retour au nombre de Lovász

Formulation du problème du stable en programme linéaire en variables 0-1 : $x_i = 1 \Leftrightarrow$ le sommet v_i est dans le stable

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

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

Retour au nombre de Lovász

Formulation du problème du stable en programme linéaire en variables 0-1 : $x_i = 1 \Leftrightarrow$ le sommet v_i est dans le stable

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

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

Autre formulation semidéfinie du nombre de Lovász :

$$\theta_G = \left\{\begin{array}{lll} \max & \sum_{i=1}^{n} x_i & \\ \text{s.c. :} & X_{ij} = 0 & \forall (i,j) \in E \\ & X_{ii} = x_i & i = 1,\dots,n \\ & X - xx^T \succcurlyeq 0 & \end{array}\right. \quad (2)$$

Plan de la Présentation

la PSD, instance de la dualité Lagrangienne

Notions développées dans cette partie :

- La dualité Lagrangienne permet de concevoir et de comparer des relaxations semidéfinies

la PSD, instance de la dualité Lagrangienne

Notions développées dans cette partie :

- La dualité Lagrangienne permet de concevoir et de comparer des relaxations semidéfinies
- En particulier pour la programmation quadratique (continue ou en variables 0-1) : les Lagrangiens total et partiel fournissent un cadre simple de comparaison

la PSD, instance de la dualité Lagrangienne

Notions développées dans cette partie :

- La dualité Lagrangienne permet de concevoir et de comparer des relaxations semidéfinies
- En particulier pour la programmation quadratique (continue ou en variables 0-1) : les Lagrangiens total et partiel fournissent un cadre simple de comparaison
- Effets spécifiques de la contrainte "semidéfinie"

Lagrangien total d'un Programme Quadratique

Programme Quadratique quelconque :

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

Lagrangien total d'un Programme Quadratique

Programme Quadratique quelconque :

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

Son programme dual est (Lagrangien total) :

$$\text{(DT)} \sup_{\lambda\geq 0}\Theta(\lambda)=\inf_{x\in\Re^n} x^TA(\lambda)x+b(\lambda)^Tx-\lambda^Td$$

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

Lagrangien total d'un Programme Quadratique

Programme Quadratique quelconque :

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

Son programme dual est (Lagrangien total) :

$$(\mathrm{DT})\ \sup_{\lambda\geq 0}\Theta(\lambda)=\inf_{x\in\Re^n} x^TA(\lambda)x+b(\lambda)^Tx-\lambda^Td$$

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

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)

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

Proposition

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

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

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

Proposition

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

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

- (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

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

Proposition

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

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

- (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
- $\alpha = 0 : A(\lambda) \succcurlyeq 0$

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

Proposition

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

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

- (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
- $\alpha = 0 : A(\lambda) \succcurlyeq 0$
- $\alpha \neq 0 : q(1, x) = -r + b(\lambda)^T x + x^T A(\lambda) x \geq 0$ 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 :

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

est :

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

Bidualiser ⇔ convexifier en Programme Semidéfini

Le dual semidéfini de :

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

est :

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

On retrouve la relaxation semidéfinie basique !

Lagrangien Partiel

Supposons que notre problème Quadratique contient des contraintes linéaires :

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

Lagrangien Partiel

Supposons que notre problème Quadratique contient des contraintes linéaires :

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

Lagrangien Partiel

Supposons que notre problème Quadratique contient des contraintes linéaires :

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

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

Lagrangien Partiel

Supposons que notre problème Quadratique contient des contraintes linéaires :

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

- 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 :

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

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

Proposition

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

Ajout de contraintes quadratiques redondantes avant d'effectuer la bidualisation

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

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

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 $\{x : Ax = b\}$

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

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 !

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 $(DT)_{\mathfrak{J}}=(DP)$

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

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

Si $\mathfrak{L}_{DT_{\mathfrak{J}}}$ est convexe alors (DT)$_{\mathfrak{J}}$=(DP)$\leq$(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 [Faye Roupin 2007]

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 [Faye Roupin 2007]

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 [Faye Roupin 2007]

On retrouve dans la formulation semidéfinie les contraintes quadratiques linéarisées :

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

La convexification dans le cas booléen

Cas particulier des Programmes Quadratiques en 0-1
[Poljak et al 1995, Lemaréchal Oustry 1999]

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

La convexification dans le cas booléen

Cas particulier des Programmes Quadratiques en 0-1
[Poljak et al 1995, Lemaréchal Oustry 1999]

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

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

La convexification dans le cas booléen

Cas particulier des Programmes Quadratiques en 0-1
[Poljak et al 1995, Lemaréchal Oustry 1999]

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

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

La convexification dans le cas booléen

Cas particulier des Programmes Quadratiques en 0-1
[Poljak et al 1995, Lemaréchal Oustry 1999]

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

- $\mathfrak{L}_{DT_C}(x, \mu, \lambda, \omega) = \mathfrak{L}_{DT}(x, \mu, \lambda) + \omega \, (Ax - b)^2$
- La convexification est faite ! : (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

La convexification dans le cas booléen

Cas particulier des Programmes Quadratiques en 0-1
[Poljak et al 1995, Lemaréchal Oustry 1999]

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

- $\mathfrak{L}_{DT_C}(x, \mu, \lambda, \omega) = \mathfrak{L}_{DT}(x, \mu, \lambda) + \omega\,(Ax - b)^2$
- La convexification est faite ! : (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
- Mais dans le cas non-booléen, $(Ax - b)^2$ ne convexifie pas toujours le Lagrangien : saut de dualité possible !

La convexification est possible dans le cas général !

[Faye Roupin 2007]

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.

La convexification est possible dans le cas général !

[Faye Roupin 2007]

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 $i \in \{1, \dots, n\}$ et $j \in \{1, \dots, p\}$ qui convexifie la forme quadratique $x^T Qx$ sur $\Re^n$ tout entier.

La convexification est possible dans le cas général !

[Faye Roupin 2007]

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.

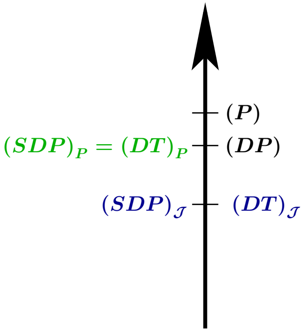
Formulation semidéfinie de la relaxation lagrangienne partielle (DP)

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

Les contraintes "produits" permettent d'atteindre la valeur de (DP)

corollaire

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



Dans le primal tous les chemins mènent à (DP) (mais pas dans le dual !)

Proposition

Si $X - xx^T \succcurlyeq 0$ *et* $rang(A) = p$ *alors il est équivalent d'utiliser*
[A] $A^T A \bullet X - 2b^T Ax + b^2 = 0$ *ou*
[B] $Ax = b$ *et* $\sum_{k=1}^{n} A_{jk}X_{ki} - b_j x_i = 0 \ \forall i \in \{1, \ldots, n\}$
$\forall j \in \{1, \ldots, p\}$.

Dans le primal tous les chemins mènent à (DP) (mais pas dans le dual !)

Proposition

Si $X - xx^T \succcurlyeq 0$ et $rang(A) = p$ alors il est équivalent d'utiliser
[A] $A^T A \bullet X - 2b^T Ax + b^2 = 0$ ou
[B] $Ax = b$ et $\sum_{k=1}^{n} A_{jk} X_{ki} - b_j x_i = 0$ $\forall i \in \{1, \ldots, n\}$
$\forall j \in \{1, \ldots, p\}$.

- Soit (X, x) vérifiant [B]. $\sum_j \sum_i A_{ji} \sum_k (A_{jk} X_{ki} - b_j x_i)$

Dans le primal tous les chemins mènent à (DP) (mais pas dans le dual !)

Proposition

Si $X - xx^T \succcurlyeq 0$ et $rang(A) = p$ alors il est équivalent d'utiliser
[A] $A^T A \bullet X - 2b^T Ax + b^2 = 0$ ou
[B] $Ax = b$ et $\sum_{k=1}^{n} A_{jk} X_{ki} - b_j x_i = 0 \; \forall i \in \{1, \ldots, n\}$
$\forall j \in \{1, \ldots, p\}$.

- Soit (X, x) vérifiant [B]. $\sum_j \sum_i A_{ji} \sum_k (A_{jk} X_{ki} - b_j x_i)$

Dans le primal tous les chemins mènent à (DP) (mais pas dans le dual !)

Proposition

Si $X - xx^T \succcurlyeq 0$ et $rang(A) = p$ alors il est équivalent d'utiliser
[A] $A^TA \bullet X - 2b^TAx + b^2 = 0$ ou
[B] $Ax = b$ et $\sum_{k=1}^{n} A_{jk}X_{ki} - b_jx_i = 0$ $\forall i \in \{1, \ldots, n\}$
$\forall j \in \{1, \ldots, p\}$.

- Soit (X, x) vérifiant [B]. $\sum_j \sum_i A_{ji} \sum_k (A_{jk}X_{ki} - b_jx_i) = A^TA \bullet X - b^TAx = 0 = A^TA \bullet X - 2b^TAx + b^2 = 0$ car $Ax = b$. Donc (X, x) vérifie [A].
- On pourrait faire la même chose en programmation linéaire...

Un effet (semidéfini) positif !

Réciproque :

- Soit (X, x) vérifiant $A^T A \bullet X - 2b^T Ax + b^2 = 0$. $A^T A \bullet (X - xx^T) + (Ax - b)^2 = 0$. Donc $A^T A \bullet (X - xx^T) = 0$ d'où $A^T A\,(X - xx^T) = 0$ car $A^T A \succcurlyeq 0$ et $X - xx^T \succcurlyeq 0$

Un effet (semidéfini) positif !

Réciproque :

- Soit (X, x) vérifiant $A^T A \bullet X - 2b^T Ax + b^2 = 0$.
  $A^T A \bullet (X - xx^T) + (Ax - b)^2 = 0$. Donc
  $A^T A \bullet (X - xx^T) = 0$ d'où $A^T A \left(X - xx^T\right) = 0$ car
  $A^T A \succcurlyeq 0$ et $X - xx^T \succcurlyeq 0$
- Donc $\forall (r, i) \in \{1, \dots, n\}^2$
  $\sum_{k=1}^{n} \sum_{j=1}^{p} A_{jr} A_{jk} X_{ki} - x_i \sum_{k=1}^{n} \sum_{j=1}^{p} A_{jr} A_{jk} x_k = 0$. Or
  $\sum_{j=1}^{p} A_{jr} \sum_{k=1}^{n} A_{jk} x_k = \sum_{j=1}^{p} A_{jr} b_j$, car $Ax = b$

Un effet (semidéfini) positif !

Réciproque :

- Soit (X, x) vérifiant $A^T A \bullet X - 2b^T Ax + b^2 = 0$.
  $A^T A \bullet (X - xx^T) + (Ax - b)^2 = 0$. Donc
  $A^T A \bullet (X - xx^T) = 0$ d'où $A^T A\,(X - xx^T) = 0$ car
  $A^T A \succcurlyeq 0$ et $X - xx^T \succcurlyeq 0$
- Donc $\forall (r, i) \in \{1, \ldots, n\}^2$
  $\sum_{k=1}^n \sum_{j=1}^p A_{jr} A_{jk} X_{ki} - x_i \sum_{k=1}^n \sum_{j=1}^p A_{jr} A_{jk} x_k = 0$. Or
  $\sum_{j=1}^p A_{jr} \sum_{k=1}^n A_{jk} x_k = \sum_{j=1}^p A_{jr} b_j$, car $Ax = b$
- On a donc une combinaison linéaire des p lignes de A
  $\sum_{j=1}^p A_{jr} \left(\sum_{k=1}^n A_{jk} X_{ki} - b_j x_i\right) = 0$.

Un effet (semidéfini) positif !

Réciproque :

- Soit (X, x) vérifiant $A^TA \bullet X - 2b^TAx + b^2 = 0$.
  $A^TA \bullet (X - xx^T) + (Ax - b)^2 = 0$. Donc
  $A^TA \bullet (X - xx^T) = 0$ d'où $A^TA\,(X - xx^T) = 0$ car
  $A^TA \succcurlyeq 0$ et $X - xx^T \succcurlyeq 0$
- Donc $\forall (r, i) \in \{1, \ldots, n\}^2$
  $\sum_{k=1}^n \sum_{j=1}^p A_{jr}A_{jk}X_{ki} - x_i \sum_{k=1}^n \sum_{j=1}^p A_{jr}A_{jk}x_k = 0$. Or
  $\sum_{j=1}^p A_{jr} \sum_{k=1}^n A_{jk}x_k = \sum_{j=1}^p A_{jr}b_j$, car $Ax = b$
- On a donc une combinaison linéaire des p lignes de A
  $\sum_{j=1}^p A_{jr}\left(\sum_{k=1}^n A_{jk}X_{ki} - b_jx_i\right) = 0$.
- $rang(A) = p$! donc $\sum_{k=1}^n A_{jk}X_{ki} - b_jx_i = 0\ \forall j \in \{1, \ldots, p\}$
  et $\forall i \in \{1, \ldots, n\}$. Donc (X, x) vérifie [B]

Cas des contraintes d'inégalité

- On considère une contrainte d'inégalité linéaire : $c^T x \leq d$

Cas des contraintes d'inégalité

- On considère une contrainte d'inégalité linéaire : $c^T x \leq d$
- [Adams Sherali 1990] une approche linéaire pour les programmes quadratiques : multiplication de l'inégalité par x_j : $x_j c^T x \leq dx_j$ puis linéarisation en $\sum_{i=1}^{n} c_i X_{ij} \leq dx_j$

Cas des contraintes d'inégalité

- On considère une contrainte d'inégalité linéaire : $c^T x \leq d$
- [Adams Sherali 1990] une approche linéaire pour les programmes quadratiques : multiplication de l'inégalité par x_j : $x_j c^T x \leq dx_j$ puis linéarisation en $\sum_{i=1}^{n} c_i X_{ij} \leq dx_j$
- Ajoutons une variable d'écart : $x_0 \geq 0$: $c^T x + x_0 = d$

Cas des contraintes d'inégalité

- On considère une contrainte d'inégalité linéaire : $c^T x \leq d$
- [Adams Sherali 1990] une approche linéaire pour les programmes quadratiques : multiplication de l'inégalité par x_j : $x_j c^T x \leq dx_j$ puis linéarisation en $\sum_{i=1}^{n} c_i X_{ij} \leq dx_j$
- Ajoutons une variable d'écart : $x_0 \geq 0$: $c^T x + x_0 = d$
- $c' = \begin{pmatrix} c \\ 1 \end{pmatrix}$, $x' = \begin{pmatrix} x \\ x_0 \end{pmatrix}$ et on impose $X' - x'x'^T \succcurlyeq 0$

Cas des contraintes d'inégalité

- On considère une contrainte d'inégalité linéaire : $c^T x \leq d$
- [Adams Sherali 1990] une approche linéaire pour les programmes quadratiques : multiplication de l'inégalité par x_j : $x_j c^T x \leq d x_j$ puis linéarisation en $\sum_{i=1}^{n} c_i X_{ij} \leq d x_j$
- Ajoutons une variable d'écart : $x_0 \geq 0$: $c^T x + x_0 = d$
- $c' = \begin{pmatrix} c \\ 1 \end{pmatrix}$, $x' = \begin{pmatrix} x \\ x_0 \end{pmatrix}$ et on impose $X' - x' x'^T \succcurlyeq 0$
- Appliquons l'approche précédente : $c' c'^T \bullet X' - 2(x_0 - d) c^T x + (x_0 - d)^2 = 0$

Cas des contraintes d'inégalité

- On considère une contrainte d'inégalité linéaire : $c^T x \leq d$
- [Adams Sherali 1990] une approche linéaire pour les programmes quadratiques : multiplication de l'inégalité par x_j : $x_j c^T x \leq dx_j$ puis linéarisation en $\sum_{i=1}^{n} c_i X_{ij} \leq dx_j$
- Ajoutons une variable d'écart : $x_0 \geq 0$: $c^T x + x_0 = d$
- $c' = \begin{pmatrix} c \\ 1 \end{pmatrix}$, $x' = \begin{pmatrix} x \\ x_0 \end{pmatrix}$ et on impose $X' - x'x'^T \succcurlyeq 0$
- Appliquons l'approche précédente : $c'c'^T \bullet X' - 2(x_0 - d)c^T x + (x_0 - d)^2 = 0$
- On a $c'c'^T \bullet \left(X' - x'x'^T\right) + \left(c^T x + x_0 - d\right)^2 = 0$

Cas des contraintes d'inégalité

- On considère une contrainte d'inégalité linéaire : $c^T x \leq d$
- [Adams Sherali 1990] une approche linéaire pour les programmes quadratiques : multiplication de l'inégalité par x_j : $x_j c^T x \leq dx_j$ puis linéarisation en $\sum_{i=1}^{n} c_i X_{ij} \leq dx_j$
- Ajoutons une variable d'écart : $x_0 \geq 0$: $c^T x + x_0 = d$
- $c' = \begin{pmatrix} c \\ 1 \end{pmatrix}$, $x' = \begin{pmatrix} x \\ x_0 \end{pmatrix}$ et on impose $X' - x'x'^T \succcurlyeq 0$
- Appliquons l'approche précédente : $c'c'^T \bullet X' - 2(x_0 - d)c^T x + (x_0 - d)^2 = 0$
- On a $c'c'^T \bullet \left(X' - x'x'^T\right) + \left(c^T x + x_0 - d\right)^2 = 0$
- Donc $c^T x \leq d$ et $c'c'^T \left(X' - x'x'^T\right) = 0$

Cas des contraintes d'inégalité

- On considère une contrainte d'inégalité linéaire : $c^T x \leq d$
- [Adams Sherali 1990] une approche linéaire pour les programmes quadratiques : multiplication de l'inégalité par x_j : $x_j c^T x \leq d x_j$ puis linéarisation en $\sum_{i=1}^n c_i X_{ij} \leq d x_j$
- Ajoutons une variable d'écart : $x_0 \geq 0$: $c^T x + x_0 = d$
- $c' = \begin{pmatrix} c \\ 1 \end{pmatrix}$, $x' = \begin{pmatrix} x \\ x_0 \end{pmatrix}$ et on impose $X' - x' x'^T \succcurlyeq 0$
- Appliquons l'approche précédente : $c' c'^T \bullet X' - 2(x_0 - d) c^T x + (x_0 - d)^2 = 0$
- On a $c' c'^T \bullet \left(X' - x' x'^T\right) + \left(c^T x + x_0 - d\right)^2 = 0$
- Donc $c^T x \leq d$ et $c' c'^T \left(X' - x' x'^T\right) = 0$
- Donc $\forall j$ dans $\{1, \ldots, n\}$ $\sum_{i=1}^n c_i \left(X_{ij} - x_j x_i\right) + X'_{j0} - x_j x_0 = 0$

Cas des contraintes d'inégalité

- On considère une contrainte d'inégalité linéaire : $c^T x \leq d$
- [Adams Sherali 1990] une approche linéaire pour les programmes quadratiques : multiplication de l'inégalité par x_j : $x_j c^T x \leq d x_j$ puis linéarisation en $\sum_{i=1}^n c_i X_{ij} \leq d x_j$
- Ajoutons une variable d'écart : $x_0 \geq 0$: $c^T x + x_0 = d$
- $c' = \begin{pmatrix} c \\ 1 \end{pmatrix}$, $x' = \begin{pmatrix} x \\ x_0 \end{pmatrix}$ et on impose $X' - x' x'^T \succcurlyeq 0$
- Appliquons l'approche précédente : $c' c'^T \bullet X' - 2(x_0 - d) c^T x + (x_0 - d)^2 = 0$
- On a $c' c'^T \bullet \left(X' - x' x'^T\right) + \left(c^T x + x_0 - d\right)^2 = 0$
- Donc $c^T x \leq d$ et $c' c'^T \left(X' - x' x'^T\right) = 0$
- Donc $\forall j$ dans $\{1, \ldots, n\}$ $\sum_{i=1}^n c_i \left(X_{ij} - x_j x_i\right) + X'_{j0} - x_j x_0 = 0$
- $\sum_{i=1}^n c_i X_{ij} - c^T x \left(x_j\right) + X'_{j0} - x_j x_0 = 0$

Cas des contraintes d'inégalité

- On considère une contrainte d'inégalité linéaire : $c^T x \leq d$
- [Adams Sherali 1990] une approche linéaire pour les programmes quadratiques : multiplication de l'inégalité par x_j : $x_j c^T x \leq dx_j$ puis linéarisation en $\sum_{i=1}^n c_i X_{ij} \leq dx_j$
- Ajoutons une variable d'écart : $x_0 \geq 0$: $c^T x + x_0 = d$
- $c' = \begin{pmatrix} c \\ 1 \end{pmatrix}$, $x' = \begin{pmatrix} x \\ x_0 \end{pmatrix}$ et on impose $X' - x' x'^T \succcurlyeq 0$
- Appliquons l'approche précédente : $c' c'^T \bullet X' - 2(x_0 - d) c^T x + (x_0 - d)^2 = 0$
- On a $c' c'^T \bullet \left(X' - x' x'^T\right) + \left(c^T x + x_0 - d\right)^2 = 0$
- Donc $c^T x \leq d$ et $c' c'^T \left(X' - x' x'^T\right) = 0$
- Donc $\forall j$ dans $\{1, \ldots, n\}$ $\sum_{i=1}^n c_i \left(X_{ij} - x_j x_i\right) + X'_{j0} - x_j x_0 = 0$
- $\sum_{i=1}^n c_i X_{ij} - c^T x \left(x_j\right) + X'_{j0} - x_j x_0 = 0$
- $c^T x = d - x_0$ implique $\sum_{i=1}^n c_i X_{ij} = dx_j - X'_{j0}$

Cas des contraintes d'inégalité

- On considère une contrainte d'inégalité linéaire : $c^T x \leq d$
- [Adams Sherali 1990] une approche linéaire pour les programmes quadratiques : multiplication de l'inégalité par x_j : $x_j c^T x \leq d x_j$ puis linéarisation en $\sum_{i=1}^{n} c_i X_{ij} \leq d x_j$
- Ajoutons une variable d'écart : $x_0 \geq 0$: $c^T x + x_0 = d$
- $c' = \begin{pmatrix} c \\ 1 \end{pmatrix}$, $x' = \begin{pmatrix} x \\ x_0 \end{pmatrix}$ et on impose $X' - x' x'^T \succcurlyeq 0$
- Appliquons l'approche précédente : $c' c'^T \bullet X' - 2(x_0 - d) c^T x + (x_0 - d)^2 = 0$
- On a $c' c'^T \bullet \left(X' - x' x'^T\right) + \left(c^T x + x_0 - d\right)^2 = 0$
- Donc $c^T x \leq d$ et $c' c'^T \left(X' - x' x'^T\right) = 0$
- Donc $\forall j$ dans $\{1, \ldots, n\}$ $\sum_{i=1}^{n} c_i \left(X_{ij} - x_j x_i\right) + X'_{j0} - x_j x_0 = 0$
- $\sum_{i=1}^{n} c_i X_{ij} - c^T x \left(x_j\right) + X'_{j0} - x_j x_0 = 0$
- $c^T x = d - x_0$ implique $\sum_{i=1}^{n} c_i X_{ij} = d x_j - X'_{j0}$
- Imposons $X'_{j0} \geq 0$ $\forall j \Rightarrow \sum_{i=1}^{n} c_i X_{ij} \leq d x_j$

Plan de la Présentation

Problèmes combinatoires avec des variables prenant k valeurs distinctes

Lemme

Soit un entier $0 < k \leq n$, alors il existe une famille de k vecteurs unitaires de $\Re^n$ tels que $u_i^T u_j \leq -\frac{1}{k-1}$ $(i, j = 1, \ldots, k\,;\; i \neq j)$. De plus, $-\frac{1}{k-1}$ est la valeur minimale pour laquelle il existe une telle famille.

Problèmes combinatoires avec des variables prenant k valeurs distinctes

Lemme

Soit un entier $0 < k \leq n$, alors il existe une famille de k vecteurs unitaires de $\Re^n$ tels que $u_i^T u_j \leq -\frac{1}{k-1}$ $(i, j = 1, \ldots, k\,;\; i \neq j)$. De plus, $-\frac{1}{k-1}$ est la valeur minimale pour laquelle il existe une telle famille.

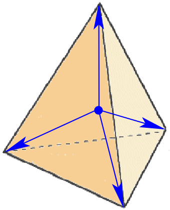
- pour $k = 2$: problèmes à variables bivalentes : $x_i \in \{-1, 1\}$

Problèmes combinatoires avec des variables prenant k valeurs distinctes

Lemme

Soit un entier $0 < k \leq n$, alors il existe une famille de k vecteurs unitaires de $\Re^n$ tels que $u_i^T u_j \leq -\frac{1}{k-1}$ $(i, j = 1, \ldots, k\,;\ i \neq j)$. De plus, $-\frac{1}{k-1}$ est la valeur minimale pour laquelle il existe une telle famille.

- pour $k = 2$: problèmes à variables bivalentes : $x_i \in \{-1, 1\}$
- pour $k = 4$: on obtient un tétraèdre régulier



Représentation du k-simplexe régulier par une contrainte semidéfinie

- Une matrice A symétrique est positive si et seulement si elle est factorisable sous la forme $A = U^T U$. $A_{ij} = u_i^T u_j$, où u_i est la ième colonne de U.

Représentation du k-simplexe régulier par une contrainte semidéfinie

- Une matrice A symétrique est positive si et seulement si elle est factorisable sous la forme $A = U^T U$. $A_{ij} = u_i^T u_j$, où u_i est la ième colonne de U.
- Conséquence : se donner une matrice positive à éléments diagonaux égaux à 1 c'est se donner un champ de vecteurs unitaires => changement de variable !

Représentation du k-simplexe régulier par une contrainte semidéfinie

- Une matrice A symétrique est positive si et seulement si elle est factorisable sous la forme $A = U^T U$. $A_{ij} = u_i^T u_j$, où u_i est la ième colonne de U.
- Conséquence : se donner une matrice positive à éléments diagonaux égaux à 1 c'est se donner un champ de vecteurs unitaires => changement de variable !
- $T = \begin{bmatrix} 1 & \frac{-1}{k-1} & \cdots & \frac{-1}{k-1} \\ \frac{-1}{k-1} & \ddots & \ddots & \vdots \\ \vdots & \frac{-1}{k-1} & 1 & \frac{-1}{k-1} \\ \frac{-1}{k-1} & \cdots & \frac{-1}{k-1} & 1 \end{bmatrix} \succcurlyeq 0$ définit un k-simplexe régulier !

Représentation du k-simplexe régulier par une contrainte semidéfinie

- Une matrice A symétrique est positive si et seulement si elle est factorisable sous la forme $A = U^T U$. $A_{ij} = u_i^T u_j$, où u_i est la ième colonne de U.
- Conséquence : se donner une matrice positive à éléments diagonaux égaux à 1 c'est se donner un champ de vecteurs unitaires => changement de variable !
- $T = \begin{bmatrix} 1 & \frac{-1}{k-1} & \cdots & \frac{-1}{k-1} \\ \frac{-1}{k-1} & \ddots & \ddots & \vdots \\ \vdots & \frac{-1}{k-1} & 1 & \frac{-1}{k-1} \\ \frac{-1}{k-1} & \cdots & \frac{-1}{k-1} & 1 \end{bmatrix} \succcurlyeq 0$ définit un k-simplexe régulier !
- $T_{ij} = u_i^T u_j$, et on sait construire facilement le champ de vecteurs (Décomposition de Cholesky) !

Programme Semidéfini en nombres entiers

- Soit un problème combinatoire à variables discrètes pouvant prendre k valeurs distinctes

Programme Semidéfini en nombres entiers

- Soit un problème combinatoire à variables discrètes pouvant prendre k valeurs distinctes
- Modèle :

Programme Semidéfini en nombres entiers

- Soit un problème combinatoire à variables discrètes pouvant prendre k valeurs distinctes
- Modèle :
  - Chaque sommet d'un k-simplexe régulier représente une de ces valeurs : $u_i^T u_j = \frac{-1}{k-1}$ pour $i \neq j$ et $u_i^2 = 1$ $\forall i \in \{1, \ldots, k\}$

Programme Semidéfini en nombres entiers

- Soit un problème combinatoire à variables discrètes pouvant prendre k valeurs distinctes
- Modèle :
  - Chaque sommet d'un k-simplexe régulier représente une de ces valeurs : $u_i^T u_j = \frac{-1}{k-1}$ pour $i \neq j$ et $u_i^2 = 1$ $\forall i \in \{1, \ldots, k\}$
  - Chaque variable est représenté par un vecteur unitaire tel que : $v_i \in \{u_1, \ldots, u_k\}$

Programme Semidéfini en nombres entiers

- Soit un problème combinatoire à variables discrètes pouvant prendre k valeurs distinctes
- Modèle :
  - Chaque sommet d'un k-simplexe régulier représente une de ces valeurs : $u_i^T u_j = \frac{-1}{k-1}$ pour $i \neq j$ et $u_i^2 = 1$ $\forall i \in \{1, \dots, k\}$
  - Chaque variable est représenté par un vecteur unitaire tel que : $v_i \in \{u_1, \dots, u_k\}$
- Relaxation : Autoriser les vecteurs à prendre une position quelconque sur la sphère unité

Programme Semidéfini en nombres entiers

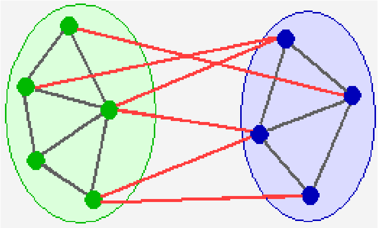
- Soit un problème combinatoire à variables discrètes pouvant prendre k valeurs distinctes
- Modèle :
  - Chaque sommet d'un k-simplexe régulier représente une de ces valeurs : $u_i^T u_j = \frac{-1}{k-1}$ pour $i \neq j$ et $u_i^2 = 1$ $\forall i \in \{1, \ldots, k\}$
  - Chaque variable est représenté par un vecteur unitaire tel que : $v_i \in \{u_1, \ldots, u_k\}$
- Relaxation : Autoriser les vecteurs à prendre une position quelconque sur la sphère unité
- Pour les problèmes purement quadratiques uniformes : pas besoin de garder le k-simplexe régulier fixe !

Application au problème MAX-CUT

[Goemans Williamson 1995]

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

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.

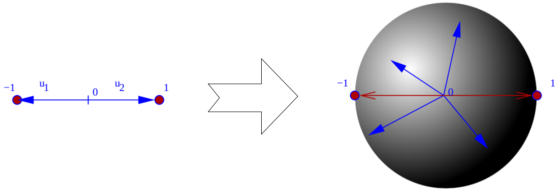


Relaxation Semidéfinie

$$(max - Cut) \begin{cases} \max & \frac{1}{2}\sum_{i<j} W_{ij}\left(1 - v_i^T v_j\right) \\ \text{s.c. :} & v_i \in \{u_1, u_2\} \end{cases}$$

Relaxation Semidéfinie

$$(max - Cut)\left\{\begin{array}{ll} \max & \frac{1}{2}\sum_{i<j} W_{ij}\left(1 - v_i^T v_j\right) \\ \text{s.c. :} & v_i \in \{u_1, u_2\} \end{array}\right.$$



Relaxation sur la dimension i.e. le rang de Y

$$(PSD)\left\{\begin{array}{ll} \max & \frac{1}{2}\sum_{i<j} W_{ij}\left(1 - v_i^T v_j\right) = \frac{1}{2}\sum_{i<j} W_{ij}\left(1 - Y_{ij}\right) \\ \text{s.c. :} & v_i^T v_i = Y_{ii} = 1\ ;\ i = 1, \ldots, n \\ & V^T V = Y \succeq 0 \end{array}\right.$$

Un algorithme aléatoire 0,878 approché

1. Prendre un vecteur $e_0 \in \Re^n$ uniformément distribué sur la sphère unité

Un algorithme aléatoire 0,878 approché

1. Prendre un vecteur $e_0 \in \Re^n$ uniformément distribué sur la sphère unité
2. Pour chaque vecteur v_i ($i = 1, \ldots, n$), si $v_i^T e_0 \geq 0$ alors $y_i = 1$ sinon $y_i = -1$

Un algorithme aléatoire 0,878 approché

1. Prendre un vecteur $e_0 \in \Re^n$ uniformément distribué sur la sphère unité
2. Pour chaque vecteur v_i ($i = 1, \ldots, n$), si $v_i^T e_0 \geq 0$ alors $y_i = 1$ sinon $y_i = -1$

- La probabilité de séparer deux sommets x_i et x_j est proportionnelle à l'angle $\theta_{ij} = \arccos\left(v_i^T v_j\right)$. Elle vaut : $\frac{1}{\pi}\theta_{ij}$

Un algorithme aléatoire 0,878 approché

1. Prendre un vecteur $e_0 \in \Re^n$ uniformément distribué sur la sphère unité
2. Pour chaque vecteur v_i ($i = 1, \dots, n$), si $v_i^T e_0 \geq 0$ alors $y_i = 1$ sinon $y_i = -1$

- La probabilité de séparer deux sommets x_i et x_j est proportionnelle à l'angle $\theta_{ij} = \arccos\left(v_i^T v_j\right)$. Elle vaut : $\frac{1}{\pi}\theta_{ij}$
- Donc l'espérance de y vaut : $E\left(y\right) = \frac{1}{\pi}\sum_{i<j} W_{ij}\theta_{ij}$

Un algorithme aléatoire 0,878 approché

1. Prendre un vecteur $e_0 \in \Re^n$ uniformément distribué sur la sphère unité
2. Pour chaque vecteur v_i $(i = 1, \ldots, n)$, si $v_i^T e_0 \geq 0$ alors $y_i = 1$ sinon $y_i = -1$

- La probabilité de séparer deux sommets x_i et x_j est proportionnelle à l'angle $\theta_{ij} = \arccos\left(v_i^T v_j\right)$. Elle vaut : $\frac{1}{\pi}\theta_{ij}$
- Donc l'espérance de y vaut : $E\left(y\right) = \frac{1}{\pi}\sum_{i<j} W_{ij}\theta_{ij}$
- La valeur de la solution semidéfinie vaut : $\sum_{i<j} W_{ij}\left(1 - v_i^T v_j\right) = \sum_{i<j} W_{ij}\left(1 - cos(\theta_{ij})\right)$

Un algorithme aléatoire 0,878 approché

1. Prendre un vecteur $e_0 \in \Re^n$ uniformément distribué sur la sphère unité
2. Pour chaque vecteur v_i ($i = 1, \ldots, n$), si $v_i^T e_0 \geq 0$ alors $y_i = 1$ sinon $y_i = -1$

- La probabilité de séparer deux sommets x_i et x_j est proportionnelle à l'angle $\theta_{ij} = \text{arccos}\left(v_i^T v_j\right)$. Elle vaut : $\frac{1}{\pi}\theta_{ij}$
- Donc l'espérance de y vaut : $E\left(y\right) = \frac{1}{\pi}\sum_{i<j} W_{ij}\theta_{ij}$
- La valeur de la solution semidéfinie vaut : $\sum_{i<j} W_{ij}\left(1 - v_i^T v_j\right) = \sum_{i<j} W_{ij}\left(1 - cos(\theta_{ij})\right)$
- Garantie : On a $\min_{0\leq\theta\leq\pi} \frac{1}{\pi}\frac{\theta}{1-cos\theta} > 0,878$

Un algorithme aléatoire 0,878 approché

1. Prendre un vecteur $e_0 \in \Re^n$ uniformément distribué sur la sphère unité
2. Pour chaque vecteur v_i ($i = 1, \ldots, n$), si $v_i^T e_0 \geq 0$ alors $y_i = 1$ sinon $y_i = -1$

- La probabilité de séparer deux sommets x_i et x_j est proportionnelle à l'angle $\theta_{ij} = \arccos\left(v_i^T v_j\right)$. Elle vaut : $\frac{1}{\pi}\theta_{ij}$
- Donc l'espérance de y vaut : $E(y) = \frac{1}{\pi}\sum_{i<j} W_{ij}\theta_{ij}$
- La valeur de la solution semidéfinie vaut : $\sum_{i<j} W_{ij}\left(1 - v_i^T v_j\right) = \sum_{i<j} W_{ij}\left(1 - cos(\theta_{ij})\right)$
- Garantie : On a $\min_{0\leq\theta\leq\pi} \frac{1}{\pi}\frac{\theta}{1-cos\theta} > 0,878$
- D'où $E(y) \geq 0,878 \sum_{i<j} W_{ij}\left(1 - v_i^T v_j\right) \geq 0,878\, val\, (\text{max-cut})$

MAX-K-CUT

- **Données.** Un graphe $G = (V, X)$ de n sommets et dont les arêtes $[v_i, v_j]$ sont valuées positivement par une matrice $W = (W_{ij})$.

MAX-K-CUT

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

MAX-K-CUT

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

MAX-K-CUT

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

$$(max - k - Cut) \begin{cases} \max & \frac{k-1}{k} \sum_{i<j} W_{ij} \left(1 - y_i^T y_j\right) \\ \text{s.c :} & y_i \in \{u_1, \ldots, u_k\} \end{cases}$$

[Freize Jerrum 1995]

$$(PSD)\left\{\begin{array}{ll} \max & \frac{k-1}{k}\sum_{i<j} W_{ij}\left(1-Y_{ij}\right) \\ \text{s.c. :} & Y_{ii}=1\ ;\ i=1,\ldots,n \\ & Y_{ij}\geq -\frac{1}{k-1}\ ;\ i\neq j;\ i,j=1,\ldots,n \\ & Y\succcurlyeq 0 \end{array}\right.$$

[Freize Jerrum 1995]

$$(PSD)\left\{\begin{array}{ll}\max & \frac{k-1}{k}\sum_{i<j} W_{ij}\left(1-Y_{ij}\right)\\ \text{s.c. :} & Y_{ii}=1\ ;\ i=1,\ldots,n\\ & Y_{ij}\geq -\frac{1}{k-1}\ ;\ i\neq j\,;\ i,j=1,\ldots,n\\ & Y\succcurlyeq 0\end{array}\right.$$

- Tirage aléatoire de k vecteurs de la sphère unité, puis affectation des vecteurs au plus proche

[Freize Jerrum 1995]

$$
(PSD) \begin{cases} \max & \frac{k-1}{k} \sum_{i<j} W_{ij} \left(1 - Y_{ij}\right) \\ \text{s.c. :} & Y_{ii} = 1 \; ; \; i = 1, \ldots, n \\ & Y_{ij} \geq -\frac{1}{k-1} \; ; \; i \neq j \, ; \; i, j = 1, \ldots, n \\ & Y \succcurlyeq 0 \end{cases}
$$

- Tirage aléatoire de k vecteurs de la sphère unité, puis affectation des vecteurs au plus proche
- => Algorithme $\left(1 - \frac{1}{k} - \frac{2}{k^2 \ln\, k}\right)$-approché

Nombre Chromatique

Colorier les sommets d'un graphe avec le minimum de couleurs (deux sommets adjacents n'ont pas la même couleur)

Nombre Chromatique

Colorier les sommets d'un graphe avec le minimum de couleurs (deux sommets adjacents n'ont pas la même couleur)

Lemme

Soit un entier $0 < k \leq n$, et $u_1, \ldots, u_n$ une famille de vecteurs unitaires de $\Re^n$ telle que $u_i^T u_j \in \left\{-\frac{1}{k-1}, 1\right\}$ $(i, j = 1, \ldots, n)$, alors il y a au plus k vecteurs distincts parmi les n

Nombre Chromatique

Colorier les sommets d'un graphe avec le minimum de couleurs (deux sommets adjacents n'ont pas la même couleur)

Lemme

Soit un entier $0 < k \leq n$, et $u_1, \ldots, u_n$ une famille de vecteurs unitaires de $\Re^n$ telle que $u_i^T u_j \in \left\{-\frac{1}{k-1}, 1\right\}$ $(i, j = 1, \ldots, n)$, alors il y a au plus k vecteurs distincts parmi les n

Nombre Chromatique

Colorier les sommets d'un graphe avec le minimum de couleurs (deux sommets adjacents n'ont pas la même couleur)

Lemme

Soit un entier $0 < k \leq n$, et $u_1, \ldots, u_n$ une famille de vecteurs unitaires de $\Re^n$ telle que $u_i^T u_j \in \left\{ -\frac{1}{k-1}, 1 \right\}$ $(i, j = 1, \ldots, n)$, alors il y a au plus k vecteurs distincts parmi les n

Programme semidéfini en nombres entiers :

$$
(\chi(G)) \left\{ \begin{array}{ll} \min & k \\ \text{s.c :} & Y_{ij} = -\frac{1}{k-1} \; ; \; \forall (i,j) \in E \\ & Y_{ii} = 1 \; ; \; i = 1, \ldots, n \\ & Y_{ij} \in \left\{ -\frac{1}{k-1}, 1 \right\} \; ; \; i, j = 1, \ldots, n \\ & Y \succeq 0 \\ & k \in N^* \end{array} \right.
$$

Relaxation semidéfinie

$$(\chi PSD) \begin{cases} \min & k \\ \text{s.c. :} & Y_{ij} = -\frac{1}{k-1} \; ; \forall (i,j) \in E \\ & Y_{ii} = 1 \; ; \; i = 1, \ldots, n \\ & Y \succeq 0 \\ & k \geq 0 \end{cases}$$

- On retrouve (encore !) une autre formulation du nombre de Lovász

Relaxation semidéfinie

$$(\chi PSD)\left\{\begin{array}{ll}\min & k\\ \text{s.c. :} & Y_{ij}=-\frac{1}{k-1}\ ;\ \forall(i,j)\in E\\ & Y_{ii}=1\ ;\ i=1,\ldots,n\\ & Y\succeq 0\\ & k\geq 0\end{array}\right.$$

- On retrouve (encore !) une autre formulation du nombre de Lovász
- On peut l'améliorer en ajoutant : $Y_{ij}\geq -\frac{1}{k-1}\ \forall(i,j)\notin E$

Amélioration de l'Approche

[Thèse Andersson 2000]

- Dans les exemples précédents : pas de structure "linéaire", le k-simplexe régulier de référence n'est pas explicitement présent

Amélioration de l'Approche

[Thèse Andersson 2000]

- Dans les exemples précédents : pas de structure "linéaire", le k-simplexe régulier de référence n'est pas explicitement présent
- Idée : associer un k-simplexe régulier à chaque variable x_i

Amélioration de l'Approche

[Thèse Andersson 2000]

- Dans les exemples précédents : pas de structure "linéaire", le k-simplexe régulier de référence n'est pas explicitement présent
- Idée : associer un k-simplexe régulier à chaque variable x_i
- Option coûteuse : k fois plus de variables dans le PSD !

Amélioration de l'Approche

[Thèse Andersson 2000]

- Dans les exemples précédents : pas de structure "linéaire", le k-simplexe régulier de référence n'est pas explicitement présent
- Idée : associer un k-simplexe régulier à chaque variable x_i
- Option coûteuse : k fois plus de variables dans le PSD !
- $\forall i, j, p : (u_p^i)^2 = 1$, $(u_p^i)^T u_{p'}^i = \frac{-1}{k-1}$

Amélioration de l'Approche

[Thèse Andersson 2000]

- Dans les exemples précédents : pas de structure "linéaire", le k-simplexe régulier de référence n'est pas explicitement présent
- Idée : associer un k-simplexe régulier à chaque variable x_i
- Option coûteuse : k fois plus de variables dans le PSD !
- $\forall i, j, p$: $(u_p^i)^2 = 1$, $(u_p^i)^T u_{p'}^i = \frac{-1}{k-1}$
- Liaisons entre les k-simplexes $\forall i, i', j, j', p$: $(u_j^i)^T u_{j+p}^{i'} = (u_{j'}^i)^T u_{j'+p}^{i'}$ et $(u_p^i)^T u_{p'}^{i'} \geq \frac{-1}{k-1}$ $\forall i \neq j$ $\forall p, p'$

Amélioration de l'Approche

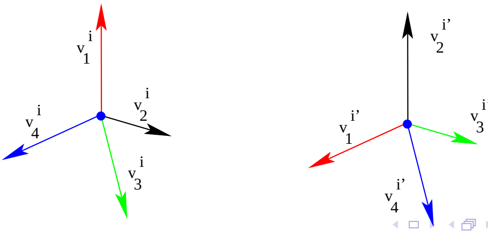
[Thèse Andersson 2000]

- Dans les exemples précédents : pas de structure "linéaire", le k-simplexe régulier de référence n'est pas explicitement présent
- Idée : associer un k-simplexe régulier à chaque variable x_i
- Option coûteuse : k fois plus de variables dans le PSD !
- $\forall i, j, p : (u_p^i)^2 = 1$, $(u_p^i)^T u_{p'}^i = \frac{-1}{k-1}$
- Liaisons entre les k-simplexes $\forall i, i', j, j', p$: $(u_j^i)^T u_{j+p}^{i'} = (u_{j'}^i)^T u_{j'+p}^{i'}$ et $(u_p^i)^T u_{p'}^{i'} \geq \frac{-1}{k-1}$ $\forall i \neq j \; \forall p, p'$

Amélioration de l'Approche

[Thèse Andersson 2000]

- Dans les exemples précédents : pas de structure "linéaire", le k-simplexe régulier de référence n'est pas explicitement présent
- Idée : associer un k-simplexe régulier à chaque variable x_i
- Option coûteuse : k fois plus de variables dans le PSD !
- $\forall i, j, p : (u_p^i)^2 = 1$, $(u_p^i)^T u_{p'}^i = \frac{-1}{k-1}$
- Liaisons entre les k-simplexes $\forall i, i', j, j', p$:
  $(u_j^i)^T u_{j+p}^{i'} = (u_{j'}^i)^T u_{j'+p}^{i'}$ et $(u_p^i)^T u_{p'}^{i'} \geq \frac{-1}{k-1}$ $\forall i \neq j \; \forall p, p'$



Application de cette approche au problème MAX-K-SECTION

Problème MAX-K-CUT avec une contrainte de cardinalité supplémentaire : chaque paquet doit contenir le même nombre de sommets [Andersson 1999]

$$
\left\{\begin{array}{lll}
\max & \frac{p-1}{p}\sum_{i<i'} W_{ii'}\left(1-\frac{1}{k}\sum_{p=0}^{k-1}(u_p^i)^T u_p^{i'}\right) & \\
\text{s.c. :} & (u_p^i)^2 = 1 & \forall i, p \\
 & (u_p^i)^T u_{p'}^i = \frac{-1}{k-1} & \forall i \, \forall p \neq p' \\
 & (u_p^i)^T u_{p'}^{i'} \geq \frac{-1}{k-1} & \forall i \neq i' \, \forall p, p' \\
 & (u_j^i)^T u_{j+p}^{i'} = (u_{j'}^i)^T u_{j'+p}^{i'} & \forall i, i', j, j', p \\
 & \sum_i v_p^i = 0 & \forall p
\end{array}\right.
$$

- Tirage aléatoire d'un seul vecteur unitaire e_0

Application de cette approche au problème MAX-K-SECTION

Problème MAX-K-CUT avec une contrainte de cardinalité supplémentaire : chaque paquet doit contenir le même nombre de sommets [Andersson 1999]

$$\begin{cases} \max & \frac{p-1}{p} \sum_{i<i'} W_{ii'} \left(1 - \frac{1}{k} \sum_{p=0}^{k-1} (u_p^i)^T u_p^{i'}\right) & \\ \text{s.c. :} & (u_p^i)^2 = 1 & \forall i, p \\ & (u_p^i)^T u_{p'}^i = \frac{-1}{k-1} & \forall i \, \forall p \neq p' \\ & (u_p^i)^T u_{p'}^{i'} \geq \frac{-1}{k-1} & \forall i \neq i' \, \forall p, p' \\ & (u_j^i)^T u_{j+p}^{i'} = (u_{j'}^i)^T u_{j'+p}^{i'} & \forall i, i', j, j', p \\ & \sum_i v_p^i = 0 & \forall p \end{cases}$$

- Tirage aléatoire d'un seul vecteur unitaire e_0
- Puis pour chaque i on prend le plus grand produit scalaire parmi les $(u_p^i)^T e_0$

Plan de la Présentation

Contrainte Sphérique

[Malick 2005]

Programmation Quadratique en variables bivalentes

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

Contrainte Sphérique

[Malick 2005]
Programmation Quadratique en variables bivalentes

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

X est de rang 1 et appartient à
$\mathcal{C} = \left\{X \in S_n \,:\, X \succcurlyeq 0, X_{jj} = 1 \forall j \in \{1, \dots, n\}\right\}$

Contrainte Sphérique

[Malick 2005]
Programmation 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
$\mathcal{C} = \left\{X \in S_n \,:\, X \succcurlyeq 0, X_{jj} = 1 \forall j \in \{1, \ldots, n\}\right\}$

Théorème

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

Contrainte Sphérique

[Malick 2005]
Programmation Quadratique en variables bivalentes

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

X est de rang 1 et appartient à
$\mathcal{C} = \{X \in S_n \,:\, X \succcurlyeq 0, X_{jj} = 1 \forall j \in \{1, \dots, n\}\}$

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 $\bullet$)

Contrainte Sphérique

[Malick 2005]

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

Contrainte Sphérique

[Malick 2005]

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

Contrainte Sphérique

[Malick 2005]

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

Contrainte Sphérique

[Malick 2005]

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

Contrainte Sphérique

[Malick 2005]

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

Contrainte Sphérique

[Malick 2005]

- Idée : remplacer $X = xx^T$ par $\|X\|^2 = n^2$ dans (Q) puis dualiser cette dernière contrainte
- Soit α le multiplicateur de Lagrange associé
- Pour $\alpha = 0$, on retrouve la relaxation semidéfinie basique !
- Pour $\alpha \neq 0$ Le Lagrangien est :
  $L(\alpha, X) = \left(\frac{\alpha}{2}n^2 + \frac{1}{2\alpha}\|Q\|^2\right) - \frac{\alpha}{2}\|X - \frac{Q}{\alpha}\|^2$
- Pour $\alpha < 0$ résoudre le problème dual est NP-difficile (concavité).
- Pour $\alpha > 0$ Problème semidéfini de moindres carrés : projection de $\frac{Q}{\alpha}$ sur un compact convexe !

Contrainte Sphérique

[Malick 2005]

- Idée : remplacer $X = xx^T$ par $\|X\|^2 = n^2$ dans (Q) puis dualiser cette dernière contrainte
- Soit α le multiplicateur de Lagrange associé
- Pour $\alpha = 0$, on retrouve la relaxation semidéfinie basique !
- Pour $\alpha \neq 0$ Le Lagrangien est :
  $L(\alpha, X) = \left(\frac{\alpha}{2} n^2 + \frac{1}{2\alpha}\|Q\|^2\right) - \frac{\alpha}{2}\|X - \frac{Q}{\alpha}\|^2$
- Pour $\alpha < 0$ résoudre le problème dual est NP-difficile (concavité).
- Pour $\alpha > 0$ Problème semidéfini de moindres carrés : projection de $\frac{Q}{\alpha}$ sur un compact convexe !
- La borne obtenue est moins bonne que celle de la PSD standard, mais ces problèmes peuvent être résolus plus efficacement que les PSDs ! [Malick 2004]

Contrainte Sphérique : un exemple

[Malick 2005]
Pour le problème MAX-CUT :

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

On obtient pour $\alpha > 0$:

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

Programmation co-positive

- Ensemble des matrices co-positives :
  $C_n = \{X \in S_n \ : \ y^T X y \geq 0, \forall y \in \Re^n_+\}$

Programmation co-positive

- Ensemble des matrices co-positives :
  $C_n = \{X \in S_n \ : \ y^T X y \geq 0, \ \forall y \in \Re^n_+\}$
- Ensemble des matrices complètement positives :
  $C_n^* = \left\{X = \sum_{i=1}^k y_i y_i^T, k \geq 1, y \in \Re^n_+, \ \forall i \in \{1, \ldots, k\}\right\}$

Programmation co-positive

- Ensemble des matrices co-positives :
  $C_n = \{X \in S_n \;:\; y^T X y \geq 0, \forall y \in \Re^n_+\}$
- Ensemble des matrices complètement positives :
  $C^*_n = \left\{X = \sum_{i=1}^k y_i y_i^T, k \geq 1, y \in \Re^n_+, \forall i \in \{1, \ldots, k\}\right\}$
- Un programme copositif est un programme linéaire sur C^*_n ou sur C_n : généralement NP-difficile !

Programmation co-positive

- Ensemble des matrices co-positives :
  $C_n = \{X \in S_n \; : \; y^T X y \geq 0, \, \forall y \in \Re_+^n\}$
- Ensemble des matrices complètement positives :
  $C_n^* = \left\{X = \sum_{i=1}^k y_i y_i^T, k \geq 1, y \in \Re_+^n, \forall i \in \{1, \ldots, k\}\right\}$
- Un programme copositif est un programme linéaire sur C_n^* ou sur C_n : généralement NP-difficile !
- Mais alors quel est l'intérêt ? ?

Programmation co-positive

- Ensemble des matrices co-positives :
  $C_n = \{X \in S_n \ : \ y^T X y \geq 0, \, \forall y \in \Re^n_+\}$
- Ensemble des matrices complètement positives :
  $C_n^* = \left\{X = \sum_{i=1}^k y_i y_i^T, k \geq 1, y \in \Re^n_+, \, \forall i \in \{1, \dots, k\}\right\}$
- Un programme copositif est un programme linéaire sur C_n^* ou sur C_n : généralement NP-difficile !
- Mais alors quel est l'intérêt ? ?
- Relâcher les programmes co-positifs en programmes linéaires ou en PSD !

Programmation co-positive

- Ensemble des matrices co-positives :
  $C_n = \{X \in S_n \;:\; y^T X y \geq 0, \; \forall y \in \Re^n_+\}$
- Ensemble des matrices complètement positives :
  $C_n^* = \left\{X = \sum_{i=1}^k y_i y_i^T, k \geq 1, y \in \Re^n_+, \; \forall i \in \{1, \ldots, k\}\right\}$
- Un programme copositif est un programme linéaire sur C_n^* ou sur C_n : généralement NP-difficile !
- Mais alors quel est l'intérêt ? ?
- Relâcher les programmes co-positifs en programmes linéaires ou en PSD !
  - [Klerk et Pasechnik 2002] le nombre de stabilité d'un graphe peut être obtenu en résolvant un programme co-positif : hiérarchie de relaxations linéaires et semidéfinies du problème

Programmation co-positive

- Ensemble des matrices co-positives :
  $C_n = \{X \in S_n \ : \ y^T X y \geq 0, \ \forall y \in \Re^n_+\}$
- Ensemble des matrices complètement positives :
  $C_n^* = \left\{X = \sum_{i=1}^k y_i y_i^T, k \geq 1, y \in \Re^n_+, \ \forall i \in \{1, \ldots, k\}\right\}$
- Un programme copositif est un programme linéaire sur C_n^* ou sur C_n : généralement NP-difficile !
- Mais alors quel est l'intérêt ? ?
- Relâcher les programmes co-positifs en programmes linéaires ou en PSD !
  - [Klerk et Pasechnik 2002] le nombre de stabilité d'un graphe peut être obtenu en résolvant un programme co-positif : hiérarchie de relaxations linéaires et semidéfinies du problème
  - [Povh et Rendl 2005] modélisation du problème de la 3-partition d'un graphe par un programme co-positif : obtention de bornes par programmation semidéfinie

Programmation co-positive pour le QAP

[Povh et Rendl 2006]
Problème de l'affectation quadratique (QAP) :

$$(QAP)\left\{\begin{array}{lll} \min & \sum_{i,j,k,l} C_{ijkl}\, x_{ij} x_{kl} & \\ \text{s.c.} & \sum_{i=1}^{n} x_{ij} = 1 & j = 1, \ldots, n \\ & \sum_{j=1}^{n} x_{ij} = 1 & i = 1, \ldots, n \\ & x \in \{0,1\}^{n^2} & \end{array}\right.$$

- Affectation de n objets à n emplacements : $x_{ij} = 1$ l'objet i est affecté à l'emplacement j

Programmation co-positive pour le QAP

[Povh et Rendl 2006]
Problème de l'affectation quadratique (QAP) :

$$(QAP)\left\{\begin{array}{lll} \min & \sum_{i,j,k,l} C_{ijkl}\, x_{ij} x_{kl} & \\ \text{s.c.} & \sum_{i=1}^{n} x_{ij} = 1 & j = 1,\ldots,n \\ & \sum_{j=1}^{n} x_{ij} = 1 & i = 1,\ldots,n \\ & x \in \{0,1\}^{n^2} & \end{array}\right.$$

- Affectation de n objets à n emplacements : $x_{ij} = 1$ l'objet i est affecté à l'emplacement j
- C_{ijkl} : coût lorsque l'objet i est en j et l'objet k est en l

Modélisation du QAP par un programme co-positif

[Povh et Rendl 2006]

$$(QAP_{CP}) \left\{ \begin{array}{lll} \min & \sum_{i,j,k,l} C_{ijkl} \, Y_{ijkl} & \\ \text{s.c.} & \sum_{i=1}^{n} Y^{ii} = I_n & i = 1, \dots, n \\ & I_n \bullet Y^{ij} = \delta_{ij} & \forall i, j \\ & \sum_{i,j,k,l} Y_{ijkl} = n^2 & \\ & Y \in C^*_{n^2} & \end{array} \right.$$

Y^{ij} : matrice $n \times n$ telle que $Y^{ij}_{kl} = Y_{ijkl}$

Modélisation du QAP par un programme co-positif

[Povh et Rendl 2006]

$$
(QAP_{CP}) \left\{
\begin{array}{lll}
\min & \sum_{i,j,k,l} C_{ijkl}\, Y_{ijkl} & \\
\text{s.c.} & \sum_{i=1}^{n} Y^{ii} = I_n & i = 1, \ldots, n \\
 & I_n \bullet Y^{ij} = \delta_{ij} & \forall i, j \\
 & \sum_{i,j,k,l} Y_{ijkl} = n^2 & \\
 & Y \in C^*_{n^2} &
\end{array}
\right.
$$

Y^{ij} : matrice $n \times n$ telle que $Y^{ij}_{kl} = Y_{ijkl}$

Théorème

Y est admissible pour (QAP_{CP}) si seulement si Y est dans l'enveloppe convexe de $\{xx^T \; : \; x \text{ admissible pour } (QAP)\}$

Modélisation du QAP par un programme co-positif

[Povh et Rendl 2006]

$$(QAP_{CP})\left\{\begin{array}{lll} \min & \sum_{i,j,k,l} C_{ijkl}\, Y_{ijkl} & \\ \text{s.c.} & \sum_{i=1}^{n} Y^{ii} = I_n & i = 1, \ldots, n \\ & I_n \bullet Y^{ij} = \delta_{ij} & \forall i, j \\ & \sum_{i,j,k,l} Y_{ijkl} = n^2 & \\ & Y \in C^*_{n^2} & \end{array}\right.$$

Y^{ij} : matrice $n \times n$ telle que $Y^{ij}_{kl} = Y_{ijkl}$

Théorème

Y est admissible pour (QAP_{CP}) si seulement si Y est dans l'enveloppe convexe de $\{xx^T \ : \ x \text{ admissible pour } (QAP)\}$

Modélisation du QAP par un programme co-positif

[Povh et Rendl 2006]

$$(QAP_{CP}) \left\{ \begin{array}{lll} \min & \sum_{i,j,k,l} C_{ijkl}\, Y_{ijkl} & \\ \text{s.c.} & \sum_{i=1}^{n} Y^{ii} = I_n & i = 1, \ldots, n \\ & I_n \bullet Y^{ij} = \delta_{ij} & \forall i, j \\ & \sum_{i,j,k,l} Y_{ijkl} = n^2 & \\ & Y \in C^*_{n^2} & \end{array} \right.$$

Y^{ij} : matrice $n \times n$ telle que $Y^{ij}_{kl} = Y_{ijkl}$

Théorème

Y est admissible pour (QAP_{CP}) si seulement si Y est dans l'enveloppe convexe de $\left\{ xx^T \ : \ x \text{ admissible pour } (QAP) \right\}$

Relaxation semidéfinie : remplacer $Y \in C^*_{n^2}$ par $Y \succcurlyeq 0$

Programmation Semidéfinie pour visualiser des groupes d'artistes semblables (Yahoo/MIT)

[Gleich, Rasmussen, Lang, Zhukov]

- **Données.** Un graphe dont les sommets sont les artistes et qui sont connectés par une arête si au moins un des deux a été classé parmi les 20 "plus proches" de l'autre selon les utilisateurs (échelle de 1 à 100)

Programmation Semidéfinie pour visualiser des groupes d'artistes semblables (Yahoo/MIT)

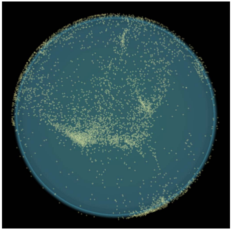
[Gleich, Rasmussen, Lang, Zhukov]

- **Données.** Un graphe dont les sommets sont les artistes et qui sont connectés par une arête si au moins un des deux a été classé parmi les 20 "plus proches" de l'autre selon les utilisateurs (échelle de 1 à 100)
- **L'objectif** est de visualiser ces similarités entre artistes sur une sphère (puis dans un plan)

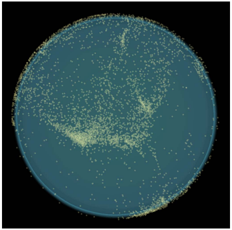
Programmation Semidéfinie pour visualiser des groupes d'artistes semblables (Yahoo/MIT)

[Gleich, Rasmussen, Lang, Zhukov]

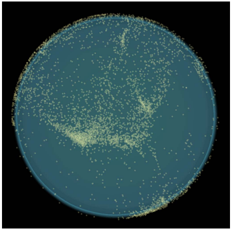
- **Données.** Un graphe dont les sommets sont les artistes et qui sont connectés par une arête si au moins un des deux a été classé parmi les 20 "plus proches" de l'autre selon les utilisateurs (échelle de 1 à 100)
- **L'objectif** est de visualiser ces similarités entre artistes sur une sphère (puis dans un plan)
- **Intérêt** : pouvoir parcourir plus facilement ces données (regroupement des artistes "semblables")



- Problème combinatoire sousjacent : minimisation de la somme des carrés des valeurs des arêtes sur une hypersphère (PSD !). Il s'agit donc de la relaxation continue d'un problème quadratique en 0-1.



- Problème combinatoire sousjacent : minimisation de la somme des carrés des valeurs des arêtes sur une hypersphère (PSD !). Il s'agit donc de la relaxation continue d'un problème quadratique en 0-1.
- Les données représentent près de 10000 artistes, mais le graphe est peu dense



- Problème combinatoire sousjacent : minimisation de la somme des carrés des valeurs des arêtes sur une hypersphère (PSD !). Il s'agit donc de la relaxation continue d'un problème quadratique en 0-1.
- Les données représentent près de 10000 artistes, mais le graphe est peu dense
- Utilisation de l'algorithme de Burer et Monteiro (SDPLR)