Programmation Semidéfinie et Optimisation Combinatoire

Frédéric Roupin

roupin@lipn.univ-paris13.fr

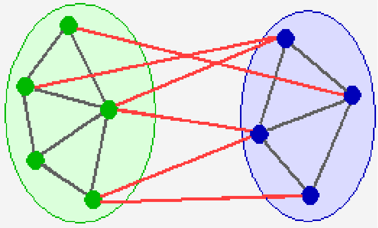
MPRO 2017-2018

Plan du cours

1. 1. Introduction

2. Application à l'optimisation combinatoire
   - Exemple introductif: le problème max-cut
   - Le problème du stable : nombre de Lovász
   - Relaxation SDP basique en $\{0, 1\}$
   - Relaxation SDP construite à partir d'une relaxation linéaire
   - Relaxation SDP en $\{-1, 1\}$
   - Liens avec l'approche Lagrangienne
   - Approche semidéfinie pour les problèmes en nombres entiers

Le problème max-cut



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

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

Le problème max-cut (2)

Modèle.

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

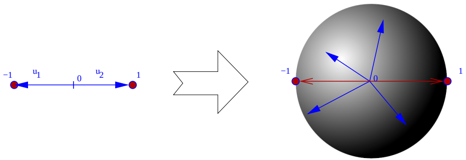
- Approche par programmation quadratique continue : un algorithme 1/2-approché très simple !
- Approche par Programmation linéaire standard : pas mieux...
- Approche par programmation semidéfinie : un algorithme approché avec une bien meilleure garantie !

Approche par PSD : reformulation

n variables de "décision" y_i $(i = 1, ..., n)$ à valeurs dans $\{-1, 1\}$.

$$(MC) \left\{ \begin{array}{l} \text{Maximiser } \frac{1}{2} \sum_{i<j} w_{ij} (1 - y_i y_j) \\ \text{Sous les contraintes } : \; y_i \in \{-1, 1\} \end{array} \right.$$

Relaxation. les variables y_i $(i = 1, ..., n)$ deviennent des vecteurs v_i de S^n. Il s'agit donc d'un relaxation sur la *dimension* des variables (initialement $y_i \in S^1 = \{-1, 1\}$).

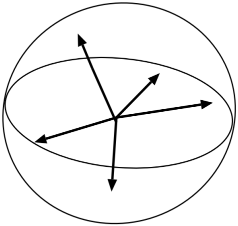


$$\left\{ \begin{array}{l} \text{Maximiser } \frac{1}{2} \sum_{i<j} w_{ij} \left(1 - v_i^T v_j\right) \\ \text{Sous les contraintes } : \; v_i \in S^n \; i = 1, ..., n \end{array} \right.$$

S^n : sphère unité en dimension n

Rappel : changement de variables "vecteurs unitaires" et "matrice positive à éléments diagonaux tous égaux à 1"

$A = VV^T$et $\forall i \in \{1, ..., n\}$, $A_{ii} = 1 \Leftrightarrow$ les vecteurs $V_1, ..., V_n \in S^n$.



Conséquence : se donner une matrice positive $n \times n$ dont les éléments diagonaux valent tous 1 équivaut à se donner un champ de vecteurs de la sphère unité de $\Re^n$

Formulation SDP de la relaxation

Conséquence :

$$\begin{cases} \text{Maximiser } \frac{1}{2}\sum_{i<j} w_{ij}\left(1 - v_i^T v_j\right) \\ \text{Sous les contraintes } : \; v_i \in S^n \; i = 1, ..., n \end{cases}$$

est équivalent à

$$(SDP)\begin{cases} \text{Maximiser } \frac{1}{2}\sum_{i<j} w_{ij}\,(1 - y_{ij}) = \frac{1}{2}W_{\text{tot}} - \frac{1}{4}W \bullet Y \\ \text{Sous les contraintes } : \; y_{ii} = 1 \,;\; i = 1, ..., n \\ \qquad\qquad Y \succcurlyeq 0 \end{cases}$$

- $Y = VV^T$: matrice de Gram des vecteurs $v_1, ..., v_n$.
- Relaxation sur le rang de Y : passage d'une matrice de rang 1 à une matrice de rang inférieur à n.

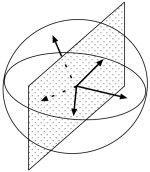
Max-cut : dual de la relaxation SDP

$$(DUAL)\left\{\begin{array}{l}\text{Minimiser } \frac{1}{2}W_{\text{tot}} + \sum_{i=1}^{n} x_i\\ \text{Sous la contrainte } : \ \mathbf{diag}(x) + \frac{1}{4}W \succcurlyeq 0\end{array}\right.$$

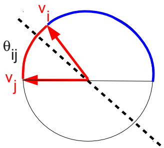
$$\begin{bmatrix} x_1 & 0 & \dots & 0 \\ 0 & x_2 & \dots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \dots & 0 & x_n \end{bmatrix} \succcurlyeq -\frac{1}{4} \begin{bmatrix} 0 & w_{12} & \dots & w_{1n} \\ w_{12} & 0 & \dots & \vdots \\ \vdots & \vdots & \ddots & w_{n(n-1)} \\ w_{1n} & \dots & w_{n(n-1)} & 0 \end{bmatrix}$$

Construction d'une solution pour le problème initial

1. Prendre un vecteur e_0 uniformément distribué sur S^n.
2. Pour chaque vecteur v_i : si $v_i^T e_0 \geqslant 0$ alors $y_i = 1$ sinon $y_i = -1$



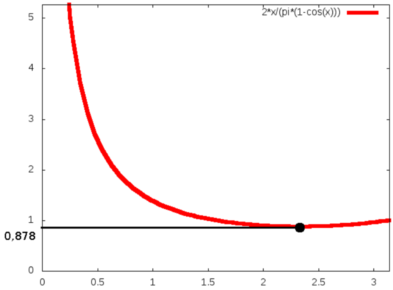
3. $v_i^T v_j = cos(\theta_{ij})||v_i||||v_j|| = cos(\theta_{ij})$



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

Construction d'une solution pour le problème initial (2)

- Donc l'espérance de la solution y ainsi obtenue vaut $E[W] = \sum_{i<j} w_{ij} Pr_{ij} = \frac{1}{\pi} \sum_{i<j} w_{ij} \arccos\left(v_i^T v_j\right)$
- La solution de la relaxation SDP vaut $\frac{1}{2} \sum_{i<j} w_{ij}(1 - v_i^T v_j)$
- Garantie obtenue : on étudie la fonction $\frac{2}{\pi} \frac{\theta}{1-cos\theta}$ pour $0 \leqslant \theta \leqslant \pi$. On a $\alpha = \min_{0 \leqslant \theta \leqslant \pi} \frac{2}{\pi} \frac{\theta}{1-cos\theta} > 0,878$



D'où $E[W] \geqslant \alpha \sum_{i<j} w_{ij} \left(1 - v_i^T v_j\right) \geqslant 0,878 val_{opt}(maxcut)$

Application dérivée : Max-2SAT

Données. Soit un ensemble de variables booléennes $X = \{x_1, ..., x_n\}$ et un ensemble de clauses $C = \{C_1, ..., C_m\}$ comportant au plus deux littéraux.
Question. Fixer les variables à “vrai” ou “faux” de façon à maximiser le nombre de clauses satisfaites.

Exemple. $X = \{x_1, x_2\}$, $C_1 = x_1 \vee \overline{x}_2$, $C_2 = x_2$, $C_3 = \overline{x}_1 \vee \overline{x}_2$. On fixe x_1 à “vrai” et x_2 à faux : C_1 et C_3 sont satisfaites, pas C_2.
Rappel : 2SAT$\in P$ mais Max-2SAT est NP-difficile !

Objectif. Afin d'utiliser la méthode appliquée à Max-Cut, écrire Max-2SAT sous la forme :

$$\begin{cases} \max \ \sum_{i<j} [a_{ij}(1 - y_i y_j) + b_{ij}(1 + y_i y_j)] \\ \text{Sous les contraintes} \ : \ y_i \in \{-1, 1\} \end{cases}$$

Max-2SAT (2)

Ajout d'une variable supplémentaire y_0 pour traiter les termes linéaires.

Evaluation d'une clause

$$\begin{aligned}
v(x_i \vee x_j) &= 1 - v(\overline{x}_i \wedge \overline{x}_j) \\
&= 1 - v(\overline{x}_i) \wedge (\overline{x}_j) \\
&= 1 - \frac{1 - y_0 y_i}{2} \frac{1 - y_0 y_j}{2} \\
&= \frac{1}{4}\left(3 + y_0 y_i + y_0 y_j - y_0^2 y_i y_j\right) \\
&= \frac{1 + y_0 y_i}{4} + \frac{1 + y_0 y_j}{4} + \frac{1 - y_i y_j}{4}
\end{aligned}$$

Pour les autres types de clauses :
même chose en remplaçant y_i par $-y_i$ si on a $\overline{x}_i$ à la place de x_i.

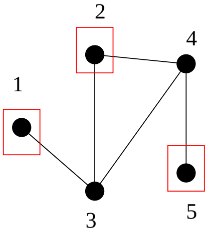
Le nombre de Lovász

Le problème du stable

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

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

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 $\lambda_{max}(A) \geqslant \lambda_{max}(J_k) = k$.
4. Donc pour tout A dans $\mathfrak{M}(G)$ $\lambda_{max}(A) \geqslant \alpha(G)$ (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 $\min\{\lambda : \lambda I - A \succcurlyeq 0\} = \lambda_{max}(A)$

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

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

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

Dual Semidéfini

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

dual :

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

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

Application dans les Graphes Parfaits

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

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

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

- 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 $v^r_j = 0 \ \forall r \neq i$
- $Z \succcurlyeq 0$ donc $\sum_{i=1}^k (kv^i - e_n)^T Z(kv^i - e_n) \geqslant 0$, où $e_n = (1, \ldots, 1)^T$
- $0 \leqslant k \left(\sum_{i=1}^k v^i (v^i)^T\right) \bullet Z - 2\left(\sum_{i=1}^k v^i\right)^T Ze_n + J_n \bullet Z$
- Or $\left(\sum_{i=1}^k v^i\right)^T Ze_n = e_n e_n^T \bullet Z$ (partition)
- Donc $0 \leqslant kI_n \bullet Z - J_n \bullet Z$ donc $J_n \bullet Z \leqslant \chi(\bar{G})$

Relaxation SDP en $\{0,1\}$: Problème Quadratique 0-1 sans autre contrainte

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

Relaxation "Classique" par programmation linéaire :

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

Relaxation basique SDP en $\{0,1\}$

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

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

RELAXATION

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

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

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

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

Exemple d'application : problème du stable

Modèle par programme linéaire en variables 0-1 :

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

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

Autre formulation semidéfinie du nombre de Lovász :

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

Relaxation SDP construite à partir d'une relaxation linéaire

Idée :

Partir d'une relaxation PL existante pour améliorer la relaxation SDP basique

Exemple : linéarisation "classique" d'un programme quadratique en 0-1 soumis à des contraintes linéaires

$$(LP)\left\{\begin{array}{ll} \max \frac{1}{2} C \bullet X + b^T x & \\ \text{s.c.} \quad 0 \leqslant x_i \leqslant 1 \ i \in \{1, ..., n\} & \\ \quad\quad 0 \leqslant X_{ij} \leqslant x_i \ i \neq j \in \{1, ..., n\} & \\ \quad\quad 0 \leqslant X_{ij} \leqslant x_j \ i \neq j \in \{1, ..., n\} & \\ \quad\quad x_i + x_j \leqslant 1 + X_{ij} \ i \neq j \in \{1, ..., n\} & \\ \quad\quad Ax \leqslant \ (\text{ou } =) d & \end{array}\right.$$

Démarche:

1. Conserver les contraintes quadratiques linéarisées (celles contenant X)
2. Renforcer la relaxation avec la contrainte non-linéaire $X - xx^T \succcurlyeq 0$
3. Traiter spécifiquement les contraintes linéaires $Ax \leqslant \ (\text{ou } =) d$

Relaxation SDP construite à partir d'une relaxation linéaire

Traitement des contraintes de bornes et celles contenant X

(P_L)	$(SDP\ \{0,1\})$
$x \in [0,1]^n$	$\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0$ $\mathbf{d}(X) = x$
$A \bullet X + c^T x = (\leqslant) d$ dont les inégalités de linéarisation: $\begin{cases} 0 \leqslant X_{ij} & \forall i < j \\ X_{ij} \leqslant x_i & \forall i < j \\ X_{ij} \leqslant x_j & \forall i < j \\ x_i + x_j - 1 \leqslant X_{ij} & \forall i < j \end{cases}$	$A \bullet X + c^T x = (\leqslant) d$ $\begin{cases} 0 \leqslant X_{ij} & \forall i < j \\ X_{ij} \leqslant x_i & \forall i < j \\ X_{ij} \leqslant x_j & \forall i < j \\ x_i + x_j - 1 \leqslant X_{ij} & \forall i < j \end{cases}$

Traitement amélioré des égalités linéaires du PL

(P_L)	$(SDP\{0,1\})$
$c^T x = d$	Règle **LE1** $\left\{\begin{array}{l} cc^T \bullet X = d^2 \\ c^T x = d \end{array}\right.$ $\Leftrightarrow cc^T \bullet X - 2dc^T x + d^2 = 0$
	Règle **LE2** $\sum_{j=1}^n c_j X_{ij} = dx_i \,\forall i \in \{1, ..., n\}$ $c^T x = d$

LE2 équivalente à LE1

Preuve de l'équivalence

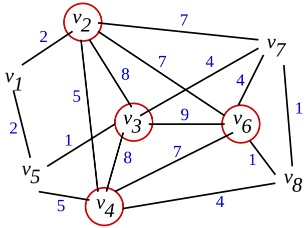
Proposition

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

Preuve.
$cc^T \bullet X = d^2 \Rightarrow cc^T \bullet (X - xx^T) + (c^T x)^2 = d^2$
Donc $cc^T \bullet (X - xx^T) = 0$
$\Rightarrow cc^T (X - xx^T) = 0$
$\forall k,j \in \{1,...,n\}\; c_k \left(\sum_{i=1}^n c_i (X_{ij} - x_i x_j)\right) = 0.$

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

Application : le Problème k-cluster



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

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

Relaxation semidéfinie "basique"

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

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

Construction d'une meilleure relaxation semidéfinie

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

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

Traitement des inégalités : knapsack quadratique

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

1. Que faire de la contrainte $a^T x \leqslant b$
2. rien : relaxation semidéfinie "basique"
3. Construire des contraintes quadratiques à partir de $a^T x \leqslant b$
4. Profiter ainsi de l'effet "semidéfini" pour obtenir des relaxations SDP plus fortes

Knapsack (sac-à-dos) quadratique

$$(KQ) \left\{ \begin{array}{ll} \text{Maximiser} & x^T C x = C \bullet xx^T \\ \text{s.}c.\ : & 0 \leqslant a^T x \leqslant b \\ & x \in \{0,1\}^n \end{array} \right.$$

Relaxation : $X - xx^T \succcurlyeq 0 \Leftrightarrow \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0$, et $\mathbf{d}(X) = x$

<u>Problème</u> : comment traiter l'inégalité linéaire ?

$$(SDP_1) \left\{ \begin{array}{ll} \text{Maximiser} & C \bullet X \\ \text{s.}c.\ : & 0 \leqslant a^T x \leqslant b \\ & X - xx^T \succcurlyeq 0 \\ & \mathbf{d}(X) = x \end{array} \right.$$

<u>1^{er} essai</u> :

$\left(b - a^T x\right)\left(b - a^T x\right) = b^2 - 2ba^T x + a^T xx^T a \geqslant 0$

Formulation SDP : $b^2 - 2ba^T x + aa^T \bullet X \geqslant 0$

Mais : $X - xx^T \succcurlyeq 0 \Rightarrow b^2 - 2ba^T x + aa^T \bullet X \geqslant 0$

Deuxième relaxation SDP du (KQ)

<u>2^{eme} essai</u> :

on met au carré l'inégalité $a^T x \leqslant b$: $aa^T \bullet xx^T \leqslant b^2$
Relaxation : on remplace xx^T par X.

$$(SDP_2) \begin{cases} \text{Maximiser} & C \bullet X \\ \text{s.c. :} & aa^T \bullet X \leqslant b^2 \\ & X - xx^T \succcurlyeq 0 \\ & \mathbf{d}(X) = x \end{cases}$$

(SDP_2) **est une meilleure relaxation que** (SDP_1)

$$\begin{aligned} aa^T \bullet X &\leqslant b^2 \\ aa^T \bullet X - aa^T \bullet xx^T + \left(a^T x\right)^2 &\leqslant b^2 \\ aa^T \bullet \left(X - xx^T\right) + \left(a^T x\right)^2 &\leqslant b^2 \end{aligned}$$

Puisque $X - xx^T \succcurlyeq 0$ on a $a^T x \leqslant b$

Troisième relaxation SDP du (KQ)

<u>Idée</u> : On multiplie l'inégalité $a^T x \leqslant b$ par x_i ou $(1 - x_i)$ pour la rendre quadratique.
Pour i fixé, on obtient :

$$\sum_{j=1}^n a_j X_{ij} \leqslant x_i \text{ et } \sum_{j=1}^n a_j \left(x_j - X_{ij}\right) \leqslant b \left(1 - x_i\right)$$

En sommant ces deux inégalités, on obtient : $a^T x \leqslant b$.

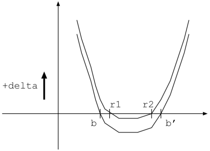
$$(SDP_3) \left\{ \begin{array}{ll} \mathrm{Max} & C \bullet X \\ \mathrm{s.c.} : & \sum_{j=1}^n a_j X_{ij} \leqslant b x_i \quad i = 1, ..., n \\ & \sum_{j=1}^n a_j \left(x_j - X_{1j}\right) \leqslant b \left(1 - x_1\right) \\ & X - x x^T \succcurlyeq 0 \\ & \mathbf{d}(X) = x \end{array} \right.$$

(SDP_3) **est une meilleure relaxation que** (SDP_2)**.**
On multiplie $\sum_{j=1}^n a_j X_{ij} \leqslant b x_i$ par a_i : $\sum_{j=1}^n a_i a_j X_{ij} \leqslant b a_i x_i$
On somme pour $i = 1, ..., n$ ces inégalités : $a^T a \bullet X \leqslant b a^T x \leqslant b^2$

Généralisation : traitements des inégalités linéaires

A : (coût : 2 contraintes), on garde $b \leqslant a^T x \leqslant b'$ => pas mieux que la PL
B : (coût : 1 contrainte), $\left(a^T x - b\right)\left(a^T x - b'\right) \leqslant 0$
i.e. $aa^T \bullet xx^T - (b + b')\, a^T x + bb' \leqslant 0$
Relaxation : $X \succcurlyeq xx^T$

$$\begin{aligned} aa^T \bullet X - (b+b')\, a^T x + bb' &\leqslant 0 \\ aa^T \bullet (X - xx^T) + \left(a^T x\right)^2 - (b+b')\, a^T x + bb' &\leqslant 0 \\ \delta + \left(a^T x\right)^2 - (b+b')\, a^T x + bb' &\leqslant 0 \end{aligned}$$



On pénalise les solutions telles que $aa^T \left(X - xx^T\right) \neq 0$ (toutes non booléennes car $X = xx^T \Rightarrow x_i^2 = x_i\, \forall i$)

Traitements possibles des inégalités linéaires

C : (coût : $n+3$ contraintes)
multiplier $b \leqslant a^T x \leqslant b'$ par x_i : $\quad bx_i \leqslant \sum_{j=1}^n a_j X_{ij} \leqslant b' x_i$
ou $(1-x_i)$: $b\,(1-x_i) \leqslant \sum_{j=1}^n a_j\,(x_j - X_{ij}) \leqslant b'\,(1-x_i)$

$$\sum_{j=1}^n a_j X_{ij} \leqslant b' x_i \;\forall i \quad [1]$$

$$bx_1 \leqslant \sum_{j=1}^n a_j X_{1j} \quad [2]$$

$$b\,(1-x_1) \leqslant \sum_{j=1}^n a_j\,(x_j - X_{1j}) \quad [3]$$

$$\sum_{j=1}^n a_j\,(x_j - X_{1j}) \leqslant b'\,(1-x_1) \quad [4]$$

C meilleure que A : $b \leqslant a^T x \leqslant b'$ ([1]+[4] et [2]+[3])

Traitement amélioré des inégalités linéaires

(P_L)	$(SDP\,\{0,1\})$
$b \leqslant a^T x \leqslant b'$	Règle **LI1** $aa^T \bullet X - (b + b')\, a^T x + bb' \leqslant 0$ Règle **LI2** $\sum_{j=1}^n a_j X_{ij} \leqslant b' x_i \ \forall i$ $b x_i \leqslant \sum_{j=1}^n a_j X_{ij} \ \forall i$ $b\,(1 - x_i) \leqslant \sum_{j=1}^n a_j\,(x_j - X_{ij}) \ \ \forall i$ $\sum_{j=1}^n a_j\,(x_j - X_{ij}) \leqslant b'\,(1 - x_i) \ \ \forall i$

Proposition

LI2 meilleure que LI1 si $a \geqslant 0$ et $b \leqslant 0$

Proof.

multiplier $\sum_{j=1}^n a_j X_{ij} \leqslant b' x_i$ par a_i :

$\sum_{j=1}^n a_i a_j X_{ij} \leqslant b' a_i x_i$,somme sur i : $aa^T \bullet X \leqslant b' a^T x$

$aa^T \bullet X - (b + b')\, a^T x + bb' \leqslant b' a^T x - (b + b') a^T x + bb' = b\left(b' - a^T x\right) \leqslant 0$

En résumé : contraintes linéaires et relaxations SDP

Approche générale

$\longrightarrow$ Utiliser la relaxation SDP basique ou une relaxation linéaire existante

$\longrightarrow$ Ces relaxations peuvent être renforcées par toute coupe/inégalité valide. En particulier, les contraintes de positivité "simples" $X_{ij} \geqslant 0$ et plus généralement les inégalités triangulaires sont efficaces.

$\longrightarrow$ Attention: une erreur "classique" est de laisser en l'état les contraintes linéaires $(c^T x = d)$ dans les relaxations SDP.

Remèdes

$\longrightarrow$ Remplacer $c^T x = d$ par (équivalent pour la SDP !)

1. $cc^T \bullet X - 2dc^T x + d^2 = 0$
2. ou $\sum_{j=1}^n c_j X_{ij} = dx_i \, \forall i \in \{1, \ldots, n\}$ et $c^T x = d$

$\longrightarrow$ Remplacer $b \leqslant a^T x \leqslant b'$ par

1. $aa^T \bullet X - (b + b')\, a^T x + bb' \leqslant 0$
2. ou $b\,(1 - x_i) \leqslant \sum_{j=1}^n a_j\,(x_j - X_{ij}) \;\; \forall i$,
   $\sum_{j=1}^n a_j\,(x_j - X_{ij}) \leqslant b'\,(1 - x_i) \;\; \forall i$

Le choix dépend du solver utilisé : Méthode de point-intérieur (ex. CSDP) : utiliser des relaxations avec peu de contraintes. Approches duales (ex. Conic Bundle, quasi-Newton) : Les contraintes "produits" sont généralement meilleures

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

Déjà vue pour max-cut !

Démarche géométrique

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

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

Relaxation sur la dimension

RELAXATION

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

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

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

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

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

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

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

Question : les deux SDPs obtenus sont-ils équivalents ?

Equivalence des relaxations SDP

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

$$(SDP)\left\{\begin{array}{l} \mathrm{Max}\ C \bullet X \\ \mathrm{s.c.} \quad A_i \bullet X = (\,ou\ \leqslant) b_i\,;\ i = 1, ..., k \\ \qquad X \succcurlyeq 0 \end{array}\right.$$

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

$$(SDP)_Q\left\{\begin{array}{l} \max\ \widetilde{C} \bullet W \\ \mathrm{s.c.} \quad \widetilde{A}_i \bullet W = (ou\ \leqslant)\, b_i\,;\ i = 1, ..., k \\ \qquad W \succcurlyeq 0 \end{array}\right.$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Améliorations semidéfinies de l'approche par PL

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

Est équivalent à

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

Lagrangien total d'un Programme Quadratique

Programme Quadratique quelconque :

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

Son programme dual est (Lagrangien total):

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

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

Proposition

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

Formulation semidéfinie du Lagrangien total

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

Proposition

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

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

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

Bidualiser $\Leftrightarrow$ convexifier en Programme Semidéfini

Le dual semidéfini de:

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

est:

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

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

Exemple : max-cut

$$(MC) \begin{cases} \text{Minimiser} & -\frac{1}{2}W_{\text{tot}} + \frac{1}{4}\sum_{i<j} W_{ij} y_i y_j \\ \text{Sous les contraintes} & : \ y_i^2 = 1 \,;\ i = 1, \ldots, n \end{cases}$$

$$\begin{aligned} L(y,\mu) &= \frac{1}{4} y^T W y + \sum_{i=1}^{n} \mu_i \left(y_i^2 - 1 \right) \\ &= y^T \left(\frac{1}{4} W + \textbf{diag}(\mu) \right) y - e_n^T \mu \end{aligned}$$

- Si $\frac{1}{4}W + \textbf{diag}(\mu) \succcurlyeq 0$ alors $\inf_y L(y,\mu) = -e_n^T \mu$ (atteint en 0)
- Sinon $\inf_y L(y,\mu) = -\infty$

On retrouve le dual de la relaxation SDP basique de max-cut !

$$\max_{\mu} \inf_{y} L(y,\mu) \Leftrightarrow \begin{cases} \min \frac{1}{2} W_{\text{tot}} + \sum_{i=1}^{n} \mu_i \\ \text{s.c. :} \ \textbf{diag}(\mu) + \frac{1}{4} W \succcurlyeq 0 \end{cases}$$

Lagrangien Partiel

Supposons que notre problème Quadratique contient des contraintes linéaires:

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

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

Proposition

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

Ajout de contraintes quadratiques redondantes avant d'effectuer la bidualisation

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

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

Proposition

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

On retrouve dans la formulation semidéfinie les contraintes quadratiques linéarisées :

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

Deux convexifications équivalentes dans le cas 0-1

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

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

Une deuxième convexification (valable dans le cas général)

Théorème

Soient A et Q respectivement une matrice $p \times n$ et une matrice $n \times n$. SI Q est positive sur $L = \mathrm{Ker}\,(A)$ 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.

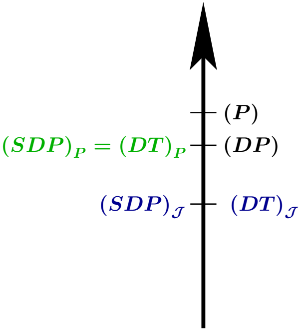
Formulation semidéfinie de la relaxation lagrangienne partielle (DP)

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

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

corollaire

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



En résumé : schéma Lagrangien pour obtenir une relaxation SDP

$$\boxed{\begin{array}{lll} \min & x^T Q_0 x + c_0^T x & \\ \text{s.c.} & x^T Q_i x + c_i^T x \leqslant a_i & i \in \{1, \dots, m\} \\ & x_j \in \{0, 1\} & j \in \{1, \dots, p\} \end{array}}$$

1. On réécrit les contraintes d'intégrité comme des contraintes quadratiques : $x_j \in \{0, 1\} \Leftrightarrow x_j^2 = x_j$
2. On ajoute des contraintes quadratiques redondantes pour fermer le plus possible le saut avant d'appliquer la dualité Lagrangienne.
3. On formule le dual Lagrangien comme un programme SDP.

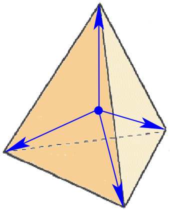
Cela explique également pourquoi les contraintes linéaires doivent être remplacées par des contraintes quadratiques dans les relaxations SDP : ici *le dual Lagrangien ignore les contraintes linéaires*.

Problèmes combinatoires avec des variables prenant k valeurs distinctes

Lemme

Soit un entier $0 < k \leqslant n$, alors il existe une famille de k vecteurs unitaires de $\Re^n$ tels que $u_i^T u_j \leqslant -\frac{1}{k-1}$ $(i, j = 1, \ldots, k\,;\ i \neq j)$. 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.
- 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
- 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

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.

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

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

max-k-cut (2)

Formulation semidéfinie "en nombres entiers" :

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

Relaxation semidéfinie :

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

Nombre Chromatique

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

Formulation PLNE

Variables :

y_k $(k = 1, ..., n)$. $y_k = 1 \Leftrightarrow$ on utilise la k^{ieme} couleur.

x_{ik} $(i, k = 1, ..., n)$. $x_{ik} = 1 \Leftrightarrow$ le sommet v_i est colorié avec la couleur k.

Contraintes :

- Chaque sommet doit être colorié avec une seule couleur.
- Les extrémités d'une arête doivent être coloriées de deux couleurs distinctes.
- Pour chaque couleur k, $y_k = \max_{1 \leqslant i \leqslant n} x_{ik}$

Nombre Chromatique : relaxation par PL

Formulation PL en variables 0-1 :

$$\begin{cases} \text{Minimiser} & \sum_{k=1}^{n} y_k \\ \text{s.c. :} & \sum_{k=1}^{n} x_{ik} = 1 \ \ ; \ i = 1, ..., n \\ & x_{ik} + x_{jk} \leqslant 1 \ \ ; \ (i,j) \in E \, ; \ k = 1, ..., n \\ & x_{ik} \leqslant y_k \ \ ; \ k = 1, ..., n \, ; \ i = 1, ..., n \\ & y \in \{0,1\}^n \ ; \ x \in \{0,1\}^{n^2} \end{cases}$$

Relaxation continue du PL :

Solution $x_{ik} = y_k = \frac{1}{n}(i = 1, ..., n\,;\ k = 1, ..., n)$ de valeur $\sum_{k=1}^{n} y_k = 1$!!

Nombre Chromatique : approche semidéfinie

Lemme

Soit un entier $0 < k \leqslant n$, 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 :

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

Nombre Chromatique : relaxation semidéfinie

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

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

Placement de tâches dans un système distribué

Données. N tâches communicantes et P processeurs. Une matrice C $N \times N$ représentant les coûts de communication, une matrice Q $N \times P$ représentant les coûts d'exécution.
Question. Trouver une affectation des tâches sur les processeurs minimisant le coût total (exécutions+communications).
Modélisation.

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

Placement de tâches dans un système distribué (2)

Relaxation LP.

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

Placement de tâches dans un système distribué (3)

Formulation pour relaxation SDP :

$$(SDP)\left\{\begin{array}{ll} \min & \frac{p-1}{p}\sum_{t=1}^{n}\sum_{k=1}^{p} Q_{tk}(x_t^T e_k + \frac{1}{p-1}) \\ & +\frac{p-1}{p}\sum_{t<t'} C_{tt'}(1 - x_t^T x_{t'}) \\ \text{s.c. :} & x_t^T x_t = 1 \ ; \ t = 1, ..., n \\ & e_k^T e_k = 1 \ ; \ k = 1, ..., p \\ & e_k^T e_{k'} = -\frac{1}{p-1} \ ; \ k \neq k' \\ & x_t \in \{e_1, \ldots, e_p\} \ t = 1, ..., n \end{array}\right.$$

Relaxation semidéfinie : remplacer $x_t \in \{e_1, \ldots, e_p\}$
par $x_t^T e_k \geqslant -\frac{1}{p-1}$ pour tout t, k.