Programmation Mathématique Avancée - Programmation quadratique - relaxations SDP et Quadratiques Convexes

Amélie Lambert

Cnam

MPRO

Problème quadratique cas pure binaire ($I = J$, $u_i = 1$)

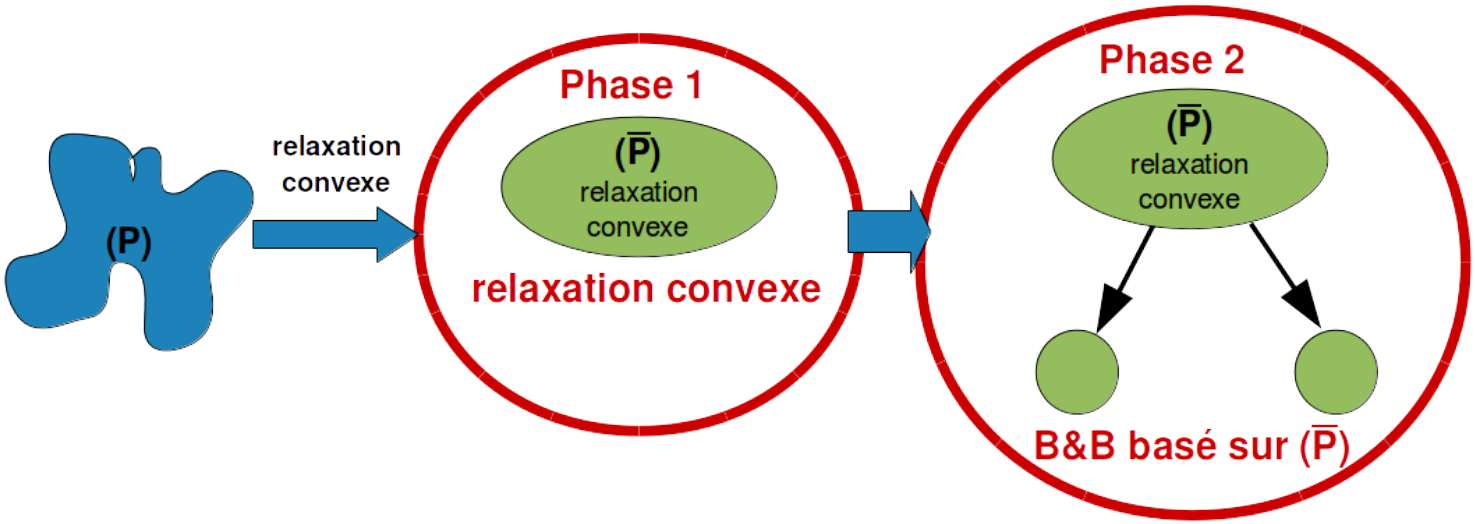
$$(P_{0,1}) \begin{cases} \min & f(x) = \langle Q, xx^T \rangle + c^T x \\ \text{s.t.} & \\ & Ax \leq b \\ & x_i \in \{0,1\} \qquad\qquad i \in J \end{cases}$$

Etat de l'art : Il existe des algorithmes "efficaces" pour résoudre des programmes en variables binaires dont la fonction objectif $f(x)$ est convexe : Branch-and-bound (b&b) basés sur la relaxation continue.

Mais la fonction $f(x)$ est quelconque et donc non nécessairement convexe

Algorithmes génériques de résolution : 2 directions (1/3)

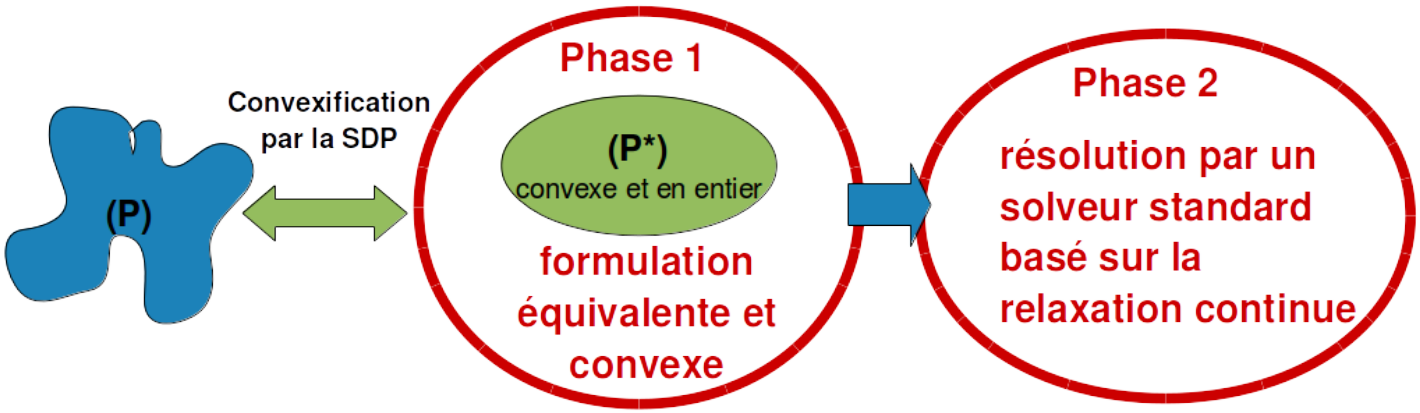
Algorithme basé sur une relaxation :



1. Construire une relaxation convexe du problème (P)
2. Développer un algorithme de b&b basé sur cette relaxation convexe
   $\Longrightarrow$ A chaque nœud de l'arbre la borne inférieure est calculée en résolvant la relaxation convexe considérée.

Algorithmes génériques de résolution : 2 directions (2/3)

Algorithme basé sur une reformulation :



1. Reformuler (P) en un problème équivalent dont l'objectif est convexe. On a deux choix :
   - i) Linéariser la fonction objectif.
   - ii) Réécrire $f(x)$ en une fonction quadratique et convexe.
2. Appliquer sur le problème reformulé un b&b dédié à la programmation convexe en variables binaires.

Algorithme générique de résolution : 2 directions (3/3)

Pour les deux catégories d'algorithme, le comportement du b&b est très dépendant de la qualité de la borne à la racine de l'arbre de recherche.

But : Trouver une relaxation (ou reformulation) :

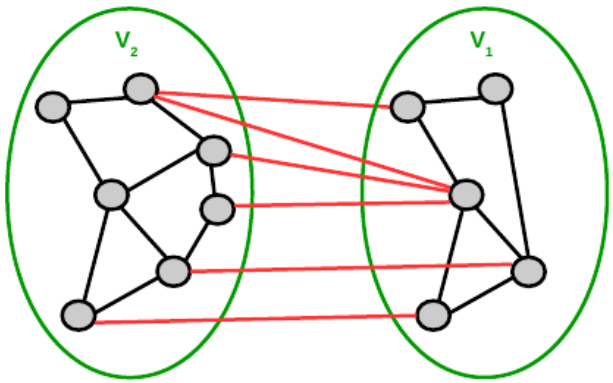
1. qui soit rapide à calculer
2. qui fournisse une borne inférieure de bonne qualité, i.e. la plus grande possible

Le problème de la coupe maximale (MaxCut)

Soit un graphe $G = (V, E)$:

- n sommets,
- les arêtes $[e_i, e_j]$ sont valuées positivement par une matrice W : w_{ij} est le poids de l'arête $[e_i, e_j]$.

Problème : Trouver une 2-partition des sommets de V : (V_1, V_2), telle que la somme des poids des arêtes ayant leurs extrémités dans des partitions différentes soit maximale.



Formulation en $0-1$:
$z_i = 1$ si i est dans V_1 et $z_i = 0$ sinon.

$$(MC_{0,1}) \begin{cases} \max \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{ij} \left[z_i(1-z_j) + z_j(1-z_i) \right] \\ z \in \{0,1\}^n \end{cases}$$

On compte w_{ij} si z_i est dans V_1 et que z_j ne l'est pas $(z_i(1-z_j))$ ou l'inverse $(z_j(1-z_i))$.

MaxCut - formulation $\{-1, 1\}$

On considère maintenant les variables $x_i = -1$ pour tous les sommets appartenant à V_1 et 1 pour ceux appartenant à V_2.

Exercice : effectuez le changement de variable, quelle formulation obtient-on ?

MaxCut - formulation $\{-1, 1\}$

On considère maintenant les variables $x_i = -1$ pour tous les sommets appartenant à V_1 et 1 pour ceux appartenant à V_2.

Exercice : effectuez le changement de variable, quelle formulation obtient-on ?

On obtient la formulation suivante :

$$(MC_{-1,1}) \begin{cases} \max & \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{1}{2} w_{ij}(1 - x_i x_j) \\ & x \in \{-1, 1\}^n \end{cases}$$

MaxCut - formulation avec la matrice Laplacienne

Matrice Laplacienne d'un graphe non orienté :

Cas non pondéré :

$$L_{ij}\begin{cases} deg(i) & \text{si } i = j \\ -1 & \text{si } i \neq j \text{ et } [e_i, e_j] \in E \\ 0 & \text{sinon} \end{cases}$$

Cas pondéré :

$$L_{ij}\begin{cases} \sum_{k=1}^{n} w_{ik} & \text{si } i = j \\ -w_{ij} & \text{si } i \neq j \text{ et } [e_i, e_j] \in E \\ 0 & \text{sinon} \end{cases}$$

Si on note M la matrice d'adjacence pondérée du graphe et et D sa matrice des degrés pondérée, on a $L = D - M$

Exercice : Montrer que le problème MaxCut peut se formuler à l'aide de la matrice Laplacienne.

$$(MC_{-1,1})\begin{cases} \max & \sum_{i=1}^{n}\sum_{j=i+1}^{n} \frac{1}{2} w_{ij}(1 - x_i x_j) \\ & x \in \{-1,1\}^n \end{cases} \qquad (MC_L)\begin{cases} \max & \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{1}{4} L_{ij} x_i x_j \\ & x \in \{-1,1\}^n \end{cases}$$

Construction d'une relaxation SDP

La programmation semi-définie

- La **programmation semi-définie (SDP)** est une extension de la programmation linéaire (LP).
- La SDP a été étudiée pour la première fois dans les années 60.
- Un problème SDP est un programme dont l'objectif est linéaire en les termes de la matrice variable X et dont les contraintes sont également linéaires en les termes de cette matrice.

Inégalités matricielles linéaires (LMI)

Définition : Soient $A_i \in \mathcal{S}^n$ et $x \in \mathbb{R}^n$, une **inegalité matricielle linéaire** est de la forme : $g(x) = \sum_{i=1}^{m} x_i A_i \succeq A_0$

Exemple : Soient 2 réels x_1 et $x_2 \in \mathbb{R}$, et l'inégalité matricielle suivante :

$$x_1 \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + x_2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \succeq \begin{pmatrix} -1 & 0 & -1 \\ 0 & -2 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

est équivalent à : $\begin{pmatrix} x_2+1 & x_1 & 1 \\ x_1 & 2 & 0 \\ 1 & 0 & x_2 \end{pmatrix} \succeq 0$

ou bien encore à l'infinité d'inégalités : $\forall v \in \mathbb{R}^3$,

$$v^T \begin{pmatrix} x_2+1 & x_1 & 1 \\ x_1 & 2 & 0 \\ 1 & 0 & x_2 \end{pmatrix} v \geq 0$$

Inégalités matricielles linéaires (LMI)

$$g(x) = \sum_{i=1}^{m} x_i A_i - A_0 \succeq 0$$

$g(x)$ caractérise un ensemble convexe, qui est l'ensemble $\mathcal{C}$:

$$\mathcal{C} = \{x \in \mathbb{R}^n \ : \ g(x) \succeq 0\}$$

$\mathcal{C}$ est dans l'intersection d'un espace affine et du cône des matrices semi-définies positives.

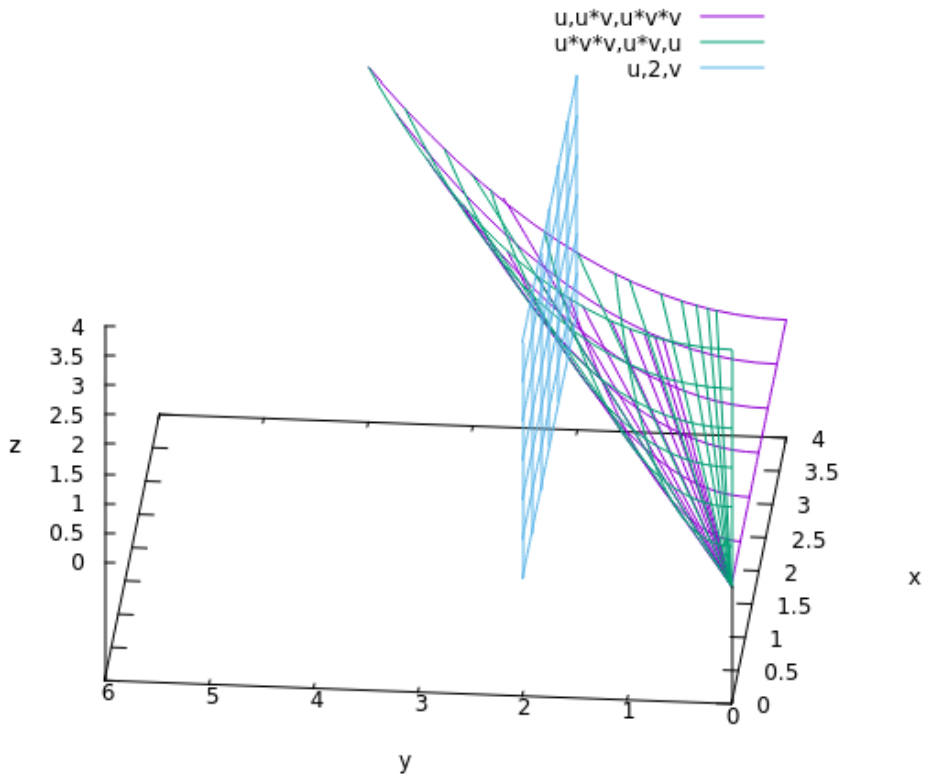
$\mathcal{C}$ n'est en général pas un polytope pour $n \geq 2$

Inégalités matricielles linéaires (LMI) : exemple

Dans $\mathcal{S}_+^2$, on cherche les matrices $\begin{pmatrix} x & y \\ y & z \end{pmatrix} \succeq 0$ telles que $y = 2$, i.e. :

$$\begin{pmatrix} x & 2 \\ 2 & z \end{pmatrix} \succeq 0 \iff x \geq 0, z \geq 0, xz \geq 4$$

Ce sont celles qui sont à l'intersection du cône $\mathcal{S}_+^2$ et de l'hyperplan $y = 2$.



Le problème primal semi-défini

Soient $Q, A_1, \ldots, A_m \in \mathcal{S}^n$, et $b \in \mathbb{R}^m$, le primal (SDP) s'écrit :

$$(SDP)\begin{cases} \max f(X) = \langle Q, X \rangle \\ \text{s.t.} \\ \quad g_r(X) = \langle A_r, X \rangle = b_r \quad \forall r = 1, \ldots, m \\ \quad X \succeq 0 \end{cases}$$

Les contraintes $g_r(x)$ indiquent la structure de la matrice X.

rappel : $\forall A, B \in \mathcal{S}^n$, $\langle A, B \rangle = \sum_{i=1}^{n} \sum_{i=1}^{n} a_{ij} b_{ij}$

Exemple : un problème primal SDP (1/2)

$$(Ex_1)\begin{cases} \min \left\langle \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, X \right\rangle \\ \text{s.t.} \\ \quad \left\langle \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, X \right\rangle = 2 \\ \quad X \succeq 0 \end{cases}$$

$$(Ex_1)\begin{cases} \min X_{11} + X_{22} \\ \text{s.t.} \\ \quad X_{12} + X_{21} = 2 \\ \quad X \succeq 0 \end{cases}$$

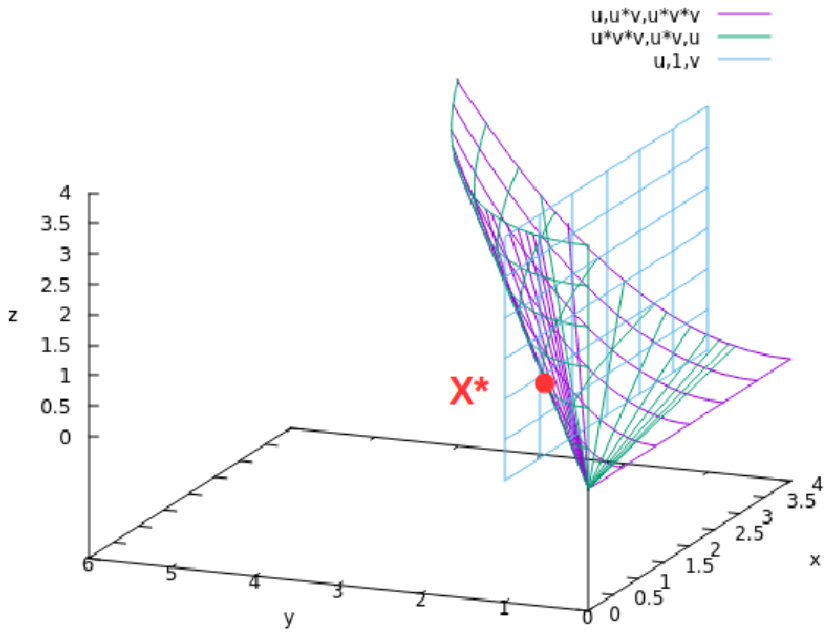
Ce qui est équivalent au problème :

$$(Ex_1)\begin{cases} \min X_{11} + X_{22} \\ \text{s.t.} \\ \quad \begin{pmatrix} X_{11} & 1 \\ 1 & X_{22} \end{pmatrix} \succeq 0 \end{cases}$$

Exemple : un problème primal SDP (2/2)

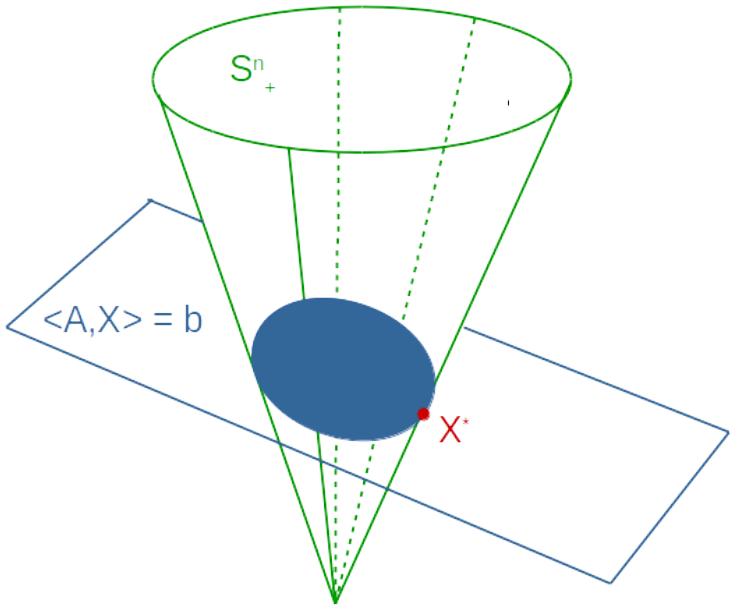
$$(Ex_1)\begin{cases} \min X_{11} + X_{22} \\ \text{s.t.} \\ \begin{pmatrix} X_{11} & 1 \\ 1 & X_{22} \end{pmatrix} \succeq 0 \end{cases}$$

La solution optimale est la matrice $X^* = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ de valeur 2.



Le problème primal SDP : géométrie

La matrice optimale X^* se trouve à l'intersection de l'hyperplan $\langle A, X \rangle = b$ et du cône $\mathcal{S}_+^n$ des matrices SDP



Exemple de région admissible pour un programme SDP

Programmation SDP et Programmation Linéaire

Considérons un programme SDP où :

- $Q, A_1, \ldots, A_m \in \mathcal{S}^n$ sont des matrices diagonales, dont les éléments diagonaux sont les vecteurs q et a_r $(r = 1, \ldots, m)$, $b \in \mathbb{R}^m$,
- la matrice variable X est diagonale, et ses termes diagonaux sont les éléments du vecteur x

Le problème SDP s'écrit

$$(SDP_d)\begin{cases} \max\langle q, x\rangle \\ \text{s.t.} \\ \quad \langle a_r, x\rangle = b_r \\ \quad x \succeq 0 \end{cases} \iff (PL)\begin{cases} \max q^T x \\ \text{s.t.} \\ \quad a_r^T x = b_r \\ \quad x \geq 0 \end{cases}$$

La programmation SDP inclut donc la programmation linéaire.

Relaxation semi-définie : exemple cas $\{-1, 1\}$

$$\begin{cases} \min\limits_{(x,y)\in\{-1,1\}} -2x^2 + xy \end{cases}$$

```
valeur opt : -3
sol opt : (-1,1)
```

Relaxation semi-définie : exemple cas $\{-1, 1\}$

$$\left\{ \min_{(x,y)\in\{-1,1\}} -2x^2 + xy \right. \iff \begin{cases} \min & -2Z_{11} + 0.5(Z_{12} + Z_{21}) \\ \text{s.t.} & Z = \begin{pmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{pmatrix} = \begin{pmatrix} x^2 & xy \\ yx & y^2 \end{pmatrix} \\ & (x, y) \in \{-1, 1\}^2 \\ & Z \in \mathcal{S}^n \end{cases}$$

- On introduit Z une matrice variable qui modélise les produits xy^T.
  $\rightarrow$ On linéarise la fonction objectif.

Relaxation semi-définie : exemple cas $\{-1, 1\}$

$$\left\{ \min_{(x,y)\in\{-1,1\}} -2x^2 + xy \right. \iff \begin{cases} \min & -2Z_{11} + 0.5(Z_{12} + Z_{21}) \\ \text{s.t.} & \operatorname{rang}(Z) = 1 \\ & (x, y) \in \{-1, 1\}^2 \\ & Z \in \mathcal{S}_+^n \end{cases}$$

- On introduit Z une matrice variable qui modélise les produits xy^T.
  $\rightarrow$ On linéarise la fonction objectif.
- On réécrit la contrainte $Z = \begin{pmatrix} x \\ y \end{pmatrix} (x\ y)$ en $\operatorname{rang}(Z) = 1$ et $Z \succeq 0$

Relaxation semi-définie : exemple cas $\{-1, 1\}$

$$\left\{ \min_{(x,y)\in\{-1,1\}} -2x^2 + xy \right. \iff \begin{cases} \min & -2Z_{11} + 0.5(Z_{12} + Z_{21}) \\ \text{s.t.} & \text{rang}(Z) = 1 \\ & Z_{ii} = 1 \\ & Z \in \mathcal{S}_+^n \end{cases}$$

- On introduit Z une matrice variable qui modélise les produits xy^T.
  $\rightarrow$ On linéarise la fonction objectif.
- On réécrit la contrainte $Z = \begin{pmatrix} x \\ y \end{pmatrix} (x\ y)$ en $\text{rang}(Z) = 1$ et $Z \succeq 0$
- On ré-écrit les contraintes $\{-1, 1\}$ en $x^2 = 1$ et $y^2 = 1$.

Relaxation semi-définie : exemple cas $\{-1, 1\}$

$$\left\{ \min_{(x,y)\in\{-1,1\}} -2x^2 + xy \right. \xrightarrow{relax} \begin{cases} \min & -2Z_{11} + 0.5(Z_{12} + Z_{21}) \\ \text{s.t.} & Z_{ii} = 1 \\ & Z \in \mathcal{S}_+^n \end{cases}$$

- On introduit Z une matrice variable qui modélise les produits xy^T.
  $\rightarrow$ On linéarise la fonction objectif.
- On réécrit la contrainte $Z = \begin{pmatrix} x \\ y \end{pmatrix} (x\ y)$ en $\text{rang}(Z) = 1$ et $Z \succeq 0$
- On ré-écrit les contraintes $\{-1, 1\}$ en $x^2 = 1$ et $y^2 = 1$.
- Pour obtenir une relaxation, on relâche la contrainte $\text{rang}(Z) = 1$

Formulation SDP pour MaxCut en $\{-1, 1\}$

Il est possible de formuler le problème MaxCut avec un programme SDP :

$$(MC_L)\begin{cases} \max & \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{4}L_{ij}x_i x_j \\ & x \in \{-1,1\}^n \end{cases} \iff (F-SDP_{MC})\begin{cases} \max & \frac{1}{4}\langle L, X\rangle \\ & X_{ii} = 1 \\ & \mathrm{rang(X)} = 1 \\ & X \succeq 0 \end{cases}$$

Preuve : Exercice

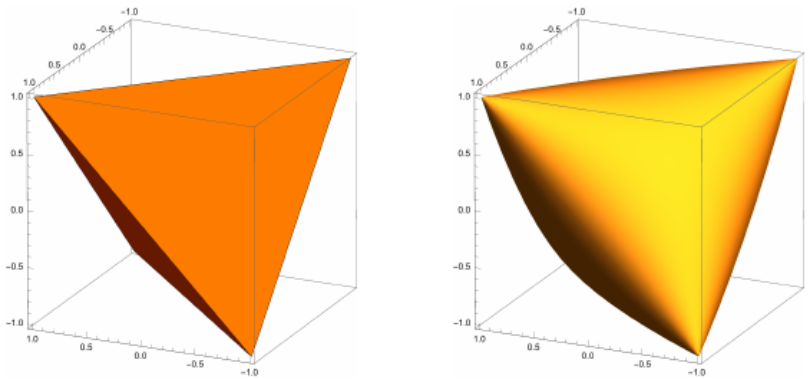
Relaxation SDP pour MaxCut en $\{-1, 1\}$

En relâchant la contrainte de rang, on obtient la relaxation

$$(MC_L)\begin{cases} \max & \frac{1}{4}\sum_{i=1}^{n}\sum_{j=i+1}^{n} L_{ij}x_i x_j \\ & x \in \{-1,1\}^n \end{cases} \xrightarrow{relax} (R-SDP_{MC})\begin{cases} \max & \frac{1}{4}\langle L, X\rangle \\ & X_{ii} = 1 \\ & X \succeq 0 \end{cases}$$

Preuve : Si x est réalisable pour (MC_L) alors $X = xx^T$ est réalisable pour $(R - SDP_{MC})$ donc $v(R - SDP_{MC}) \geq v(MC_L)$

Illustration : Graphe de 3 sommets :



Source "Topics in Convex Optimisation (Michaelmas 2018)"

b&b basé sur la relaxation $(R - SDP)_{MC}$

Mécanisme de séparation :

On sélectionne deux sommets i et j et on considère les deux sous-problèmes :

1. $S_1 := \{x \in \{-1,1\}^n : x_i - x_j = 0\}$ (i.e. i et j sont dans la même partition)
2. $S_2 := \{x \in \{-1,1\}^n : x_i + x_j = 0\}$ (i.e. i et j sont dans des partitions différentes)

On développe un b&b basé sur la borne obtenu par la résolution de $(R - SDP_{MC})$.

- Résolution efficace de $(R - SDP_{MC})$ par les algorithmes de points intérieurs
- Mais la borne à la racine n'est pas assez serrée

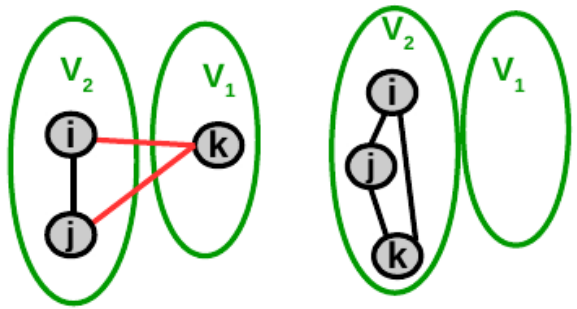
$\Longrightarrow$ Pas assez efficace pour des graphes ayant plus de 50 nœuds

Renforcement de $(R - SDP_{MC})$: triangulaires

Soit les variables $s_{ij} = 1$ si l'arête $[e_i, e_j]$ est considérée dans la coupe maximum, 0 sinon

On considère les inégalités triangulaires, $\forall\ 1 \leq i < j < k \leq n$:

$$\mathcal{T}_{0,1} \begin{cases} s_{ij} + s_{ik} + s_{jk} \leq 2 & (i) \\ s_{ij} - s_{ik} - s_{jk} \leq 0 & (ii) \\ -s_{ij} + s_{ik} - s_{jk} \leq 0 & (iii) \\ -s_{ij} - s_{ik} + s_{jk} \leq 0 & (iv) \end{cases}$$

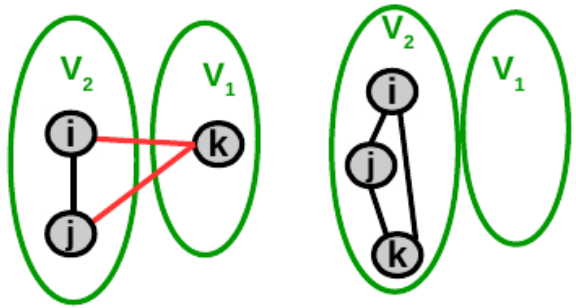


- (i) si $[e_i, e_j]$ et $[e_i, e_k]$ sont dans la coupe max, $[e_j, e_k]$ ne peut pas y être (et symmétriquement).
- (ii) si $[e_i, e_j]$ est dans la coupe au moins une des arêtes $[e_i, e_k]$ ou $[e_j, e_k]$ devront y être.
- (iii) si $[e_i, e_k]$ est dans la coupe au moins une des arêtes $[e_i, e_j]$ ou $[e_j, e_k]$ devront y être.
- (iv) si $[e_j, e_k]$ est dans la coupe au moins une des arêtes $[e_i, e_k]$ ou $[e_i, e_j]$ devront y être.

Renforcement de $(R - SDP_{MC})$: triangulaires

Soit les variables $s_{ij} = 1$ si l'arête $[e_i, e_j]$ est considérée dans la coupe maximum, 0 sinon
On considère les inégalités triangulaires, $\forall\ 1 \leq i < j < k \leq n$:

$$\mathcal{T}_{0,1} \begin{cases} s_{ij} + s_{ik} + s_{jk} \leq 2 \\ s_{ij} - s_{ik} - s_{jk} \leq 0 \\ -s_{ij} + s_{ik} - s_{jk} \leq 0 \\ -s_{ij} - s_{ik} + s_{jk} \leq 0 \end{cases}$$

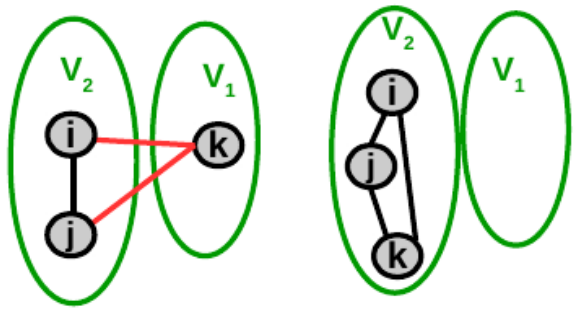


- Dans notre modélisation les variables sont en $\{-1, 1\}$, et on a $x_i = 1$ si i est dans la partition V_1 et $x_i = -1$ si i est dans la partition V_2.
- Si $s_{ij} = 1$, c'est quand $x_i = 1$ et $x_j = -1$ (ou l'inverse) et donc quand $x_i x_j = X_{ij} = -1$.
- On peut effectuer le changement de variable suivant : $X_{ij} = 1 - 2s_{ij}$. En effet, si $s_{ij} = 1 \Longrightarrow X_{ij} = -1$ et si si $s_{ij} = 0 \Longrightarrow X_{ij} = 1$

Renforcement de $(R - SDP_{MC})$: triangulaires

On considère maintenant les inégalités triangulaires, $\forall\ 1 \leq i < j < k \leq n$:

$$\mathcal{T}_{0,1}\begin{cases} s_{ij} + s_{ik} + s_{jk} \leq 2 \\ s_{ij} - s_{ik} - s_{jk} \leq 0 \\ -s_{ij} + s_{ik} - s_{jk} \leq 0 \\ -s_{ij} - s_{ik} + s_{jk} \leq 0 \end{cases}$$



avec $s_{ij} = 1$ si l'arête $[e_i, e_j]$ est considérée dans la coupe maximum, 0 sinon.

On effectue le changement de variable : $X_{ij} = 1 - 2s_{ij}$ on obtient les inégalités suivantes :

$$\mathcal{T}_{-1,1}\begin{cases} X_{ij} + X_{ik} + X_{jk} \geq -1 \quad (i) \\ X_{ij} - X_{ik} - X_{jk} \geq -1 \quad (ii) \\ -X_{ij} + X_{ik} - X_{jk} \geq -1 \quad (iii) \\ -X_{ij} - X_{ik} + X_{jk} \geq -1 \quad (iv) \end{cases} \quad (SDP - T_{MC})\begin{cases} \max \quad \frac{1}{4}\langle L, X\rangle \\ X_{ii} = 1 \\ X \in \mathcal{T}_{-1,1} \\ X \succeq 0 \end{cases}$$

On a $v(MC_L) \leq v(SDP - T_{MC}) \leq v(R - SDP_{MC})$

b&b basé sur la relaxation $(SDP - T_{MC})$ s

- Difficulté : la résolution de $(R - SDP_{MC})$ à chaque nœud car les triangulaires sont très nombreuses : $4\binom{n}{3}$
  $\Longrightarrow$ Nécessite un algorithme de séparation pour déterminer les inégalités triangulaires actives

- Utilisation des mêmes règles de séparation que pour le b&b précédent

Implémentation de cet algorithme : `BiqMac` (Rinaldi, Rendl, Wiegele)
Actuellement un des plus efficaces pour résoudre MaxCut.

Le problème de l'affectation quadratique (QAP)

Données :

- n équipements et n sites, on note $\mathcal{I} = \{1, \ldots, n\}$;
- une matrice $A = (a_{ij})$, où a_{ij} est le flux entre les équipements i et j ;
- une matrice $B = (b_{kl})$, où b_{kl} est la distance entre les sites k et l ;
- une matrice $C = (c_{ik})$, où c_{ik} est le coût d'affectation de l'équipement i sur le site k.

But :

Affecter chaque équipement à un site tel que la somme des flux par les distances soit minimisé.

QAP - Exemple

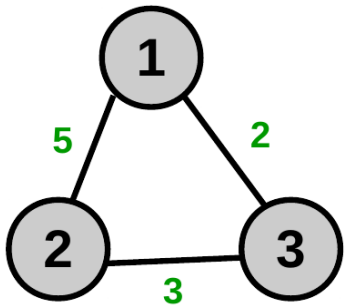
Flux entre les équipements i et j :

$$A = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

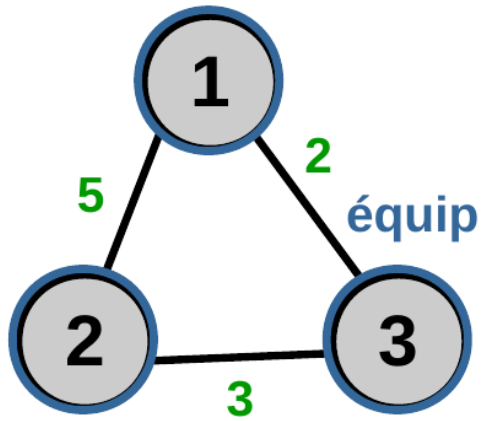
Distance entre les sites k et l :

$$B = \begin{pmatrix} 0 & 5 & 2 \\ 5 & 0 & 3 \\ 2 & 3 & 0 \end{pmatrix}$$

Coût d'affectation nul : $C = 0$



équip 1 : 5 * 1 + 2 * 2 = 9



équip 3 : 2 * 2 + 3 * 1 = 7

équip 2 : 5 * 1 + 3 * 1 = 8

Sol opt de valeur 24

QAP : Formulation de Lawler (1963)

- Variables de décision : $x_{ij} = 1$ si l'equipement i est affecté au site j
- Données : on considère la matrice $Q = (q_{ijkl})$

$$q_{ijkl} \begin{cases} q_{ijkl} = c_{ij} & \text{si } i = k \text{ et } j = l \\ q_{ijkl} = a_{ik} b_{jl} & \text{sinon} \end{cases}$$

- Formulation de Lawler

$$(QAP) \begin{cases} \min f(x) = \displaystyle\sum_{(i,j,k,l) \in \mathcal{I}^4} q_{ijkl} x_{ij} x_{kl} \\ \text{s.t.} \displaystyle\sum_{i=1}^{n} x_{ij} = 1 \quad j \in \mathcal{I} \quad \leftarrow 1 \text{ seul équipement par site} \\ \displaystyle\sum_{j=1}^{n} x_{ij} = 1 \quad i \in \mathcal{I} \quad \leftarrow 1 \text{ seul site par équipement} \\ x_{ij} \in \{0,1\} \quad (i,j) \in \mathcal{I}^2 \end{cases}$$

Linéarisation du QAP (1/2)

i. On ajoute les variables y_{ijkl} pour modéliser les produits $x_{ij}x_{kl}$

ii. On force l'égalité $y_{ijkl} = x_{ij}x_{kl}$ par les inégalités de Fortet.

$$(QAP)\begin{cases} \min f(x) = \displaystyle\sum_{(i,j,k,l)\in\mathcal{I}^4} q_{ijkl}x_{ij}x_{kl} \\ \text{s.t.} \displaystyle\sum_{i=1}^{n} x_{ij} = 1 \quad j \in \mathcal{I} \\ \displaystyle\sum_{j=1}^{n} x_{ij} = 1 \quad i \in \mathcal{I} \\ x_{ij} \in \{0,1\} \quad (i,j) \in \mathcal{I}^2 \end{cases} \iff (QAP_L)\begin{cases} \min f(x) = \displaystyle\sum_{(i,j,k,l)\in\mathcal{I}^4} q_{ijkl}y_{ijkl} \\ \text{s.t.} \displaystyle\sum_{i=1}^{n} x_{ij} = 1 \quad j \in \mathcal{I} \\ \displaystyle\sum_{j=1}^{n} x_{ij} = 1 \quad i \in \mathcal{I} \\ x_{ij} \in \{0,1\} \quad (i,j) \in \mathcal{I}^2 \\ y_{ijkl} \leq x_{ij} \quad (i,j,k,l) \in \mathcal{I}^4 \\ y_{ijkl} \leq x_{kl} \quad (i,j,k,l) \in \mathcal{I}^4 \\ y_{ijkl} \geq x_{ij} + x_{kl} - 1 \quad (i,j,k,l) \in \mathcal{I}^4 \\ y_{ijkl} \geq 0 \quad (i,j,k,l) \in \mathcal{I}^4 \end{cases}$$

Linéarisation du QAP (2/2)

Observation : $q_{ijkl} \geq 0 \Rightarrow$ les variables y vont prendre leur valeur minimale.

$\Rightarrow$ Suppression des contraintes de Fortet redondantes.

$$(QAP) \begin{cases} \min f(x) = \sum_{(i,j,k,l) \in \mathcal{I}^4} q_{ijkl} x_{ij} x_{kl} \\ \text{s.t.} \sum_{i=1}^{n} x_{ij} = 1 \quad j \in \mathcal{I} \\ \sum_{j=1}^{n} x_{ij} = 1 \quad i \in \mathcal{I} \\ x_{ij} \in \{0,1\} \quad (i,j) \in \mathcal{I}^2 \end{cases} \Longleftrightarrow (QAP_L) \begin{cases} \min f(x) = \sum_{(i,j,k,l) \in \mathcal{I}^4} q_{ijkl} y_{ijkl} \\ \text{s.t.} \sum_{i=1}^{n} x_{ij} = 1 \quad j \in \mathcal{I} \\ \sum_{j=1}^{n} x_{ij} = 1 \quad i \in \mathcal{I} \\ x_{ij} \in \{0,1\} \quad (i,j) \in \mathcal{I}^2 \\ y_{ijkl} \geq x_{ij} + x_{kl} - 1 \quad (i,j,k,l) \in \mathcal{I}^4 \\ y_{ijkl} \geq 0 \quad (i,j,k,l) \in \mathcal{I}^4 \end{cases}$$

Linéarisation du QAP (2/2)

Observation : $q_{ijkl} \geq 0 \Rightarrow$ les variables y vont prendre leur valeur minimale.

$\Rightarrow$ Suppression des contraintes de Fortet redondantes.

$$(QAP) \begin{cases} \min f(x) = \sum_{(i,j,k,l) \in \mathcal{I}^4} q_{ijkl} x_{ij} x_{kl} \\ \text{s.t.} \sum_{i=1}^{n} x_{ij} = 1 \quad j \in \mathcal{I} \\ \sum_{j=1}^{n} x_{ij} = 1 \quad i \in \mathcal{I} \\ x_{ij} \in \{0,1\} \quad (i,j) \in \mathcal{I}^2 \end{cases} \iff (QAP_L) \begin{cases} \min f(x) = \sum_{(i,j,k,l) \in \mathcal{I}^4} q_{ijkl} y_{ijkl} \\ \text{s.t.} \sum_{i=1}^{n} x_{ij} = 1 \quad j \in \mathcal{I} \\ \sum_{j=1}^{n} x_{ij} = 1 \quad i \in \mathcal{I} \\ x_{ij} \in \{0,1\} \quad (i,j) \in \mathcal{I}^2 \\ y_{ijkl} \geq x_{ij} + x_{kl} - 1 \quad (i,j,k,l) \in \mathcal{I}^4 \\ y_{ijkl} \geq 0 \quad (i,j,k,l) \in \mathcal{I}^4 \end{cases}$$

Mais la valeur de la borne par relaxation continue est toujours 0 !

Renforcement : RLT (Sherali et Adams 1990)

Principe : La Reformulation Linearization Techniques (RLT) consiste à :

1. multiplier les égalités par les variables
2. multiplier les inégalités par les variables et par leurs complémentaires.
3. linéariser le problème obtenu avec les variables y_{ijkl}.

Si on a n variables, m égalités et p inégalités, on obtient un PL avec :

- $\underbrace{n}_{\text{var x}} + \underbrace{\frac{n(n-1)}{2}}_{\text{var y}}$ variables
- $\underbrace{m}_{\text{égalités}} + \underbrace{mn}_{\text{RLT}}$ contraintes d'égalité
- $\underbrace{p}_{\text{inégalités}} + \underbrace{4\frac{n(n-1)}{2}}_{\mathcal{F}} + \underbrace{2pn}_{\text{RLT}}$ contraintes d'inégalité

Propriété : En réitérant RLT n fois, on obtient un PL dont la relaxation continue est égale à la solution optimale du problème initial.

Exemple : RLT pour le QAP

- $\forall j \in \mathcal{I}, \sum_{i=1}^{n} x_{ij} = 1$, multipliées par x_{kl} pour tout $(k,l) \in \mathcal{I}^2$

$$x_{kl}(\sum_{i=1}^{n} x_{ij}) = 1(x_{kl}) \Rightarrow (\sum_{i=1}^{n} x_{ij}x_{kl}) = 1(x_{kl}) \Rightarrow \sum_{i=1}^{n} y_{ijkl} = x_{kl} \quad \forall (j,k,l) \in \mathcal{I}^3$$

- $\forall i \in \mathcal{I}, \sum_{j=1}^{n} x_{ij} = 1$, multipliées par x_{kl} pour tout $(k,l) \in \mathcal{I}^2$

$$x_{kl}(\sum_{j=1}^{n} x_{ij}) = 1(x_{kl}) \Rightarrow (\sum_{j=1}^{n} x_{ij}x_{kl}) = 1(x_{kl}) \Rightarrow \sum_{j=1}^{n} y_{ijkl} = x_{kl} \quad \forall (i,k,l) \in \mathcal{I}^3$$

RLT pour le QAP : Redondances de certaines inégalités

$$(QAP_{RLT1}) \begin{cases} \min f(x) = \sum_{(i,j,k,l)\in\mathcal{I}^4} q_{ijkl} y_{ijkl} \\ \text{s.t.} \sum_{i=1}^{n} x_{ij} = 1 \quad j \in \mathcal{I} \\ \sum_{i=1}^{n} y_{ijkl} = x_{kl} \quad (j,k,l) \in \mathcal{I}^3 \\ \sum_{j=1}^{n} x_{ij} = 1 \quad i \in \mathcal{I} \\ \sum_{j=1}^{n} y_{ijkl} = x_{kl} \quad (i,k,l) \in \mathcal{I}^3 \\ x_{ij} \in \{0,1\} \quad (i,j) \in \mathcal{I}^2 \\ y_{ijkl} \leq x_{ij} \quad (i,j,k,l) \in \mathcal{I}^4 \\ y_{ijkl} \leq x_{kl} \quad (i,j,k,l) \in \mathcal{I}^4 \\ y_{ijkl} \geq x_{ij} + x_{kl} - 1 \quad (i,j,k,l) \in \mathcal{I}^4 \\ y_{ijkl} \geq 0 \quad (i,j,k,l) \in \mathcal{I}^4 \end{cases} \iff (QAP_{RLT2}) \begin{cases} \min f(x) = \sum_{(i,j,k,l)\in\mathcal{I}^4} q_{ijkl} y_{ijkl} \\ \text{s.t.} \sum_{i=1}^{n} x_{ij} = 1 \quad j \in \mathcal{I} \\ \sum_{i=1}^{n} y_{ijkl} = x_{kl} \quad (j,k,l) \in \mathcal{I}^3 \\ \sum_{j=1}^{n} x_{ij} = 1 \quad i \in \mathcal{I} \\ \sum_{j=1}^{n} y_{ijkl} = x_{kl} \quad (i,k,l) \in \mathcal{I}^3 \\ x_{ij} \in \{0,1\} \quad (i,j) \in \mathcal{I}^2 \\ y_{ijkl} \geq 0 \quad (i,j,k,l) \in \mathcal{I}^4 \end{cases}$$

Exercice : preuve

Comparaison des bornes pour le QAP

Instance	opt	Lin	RLT
nug06	86	0	86
nug12	578	0	522.89
nug15	1150	0	1040.99

Convexification

Convexification : principe général

Idée : Modifier $f(x)$ pour la rendre convexe.

Méthode : Ajouter à $f(x)$ des fonctions qui s'annulent sur le domaine.

Convexification : principe général

Idée : Modifier $f(x)$ pour la rendre convexe.

Méthode : Ajouter à $f(x)$ des fonctions qui s'annulent sur le domaine.

Exemple : si variables sont binaires ($x_i \in \{0,1\}$), on a $(x_i^2 - x_i) = 0$

Soient $\rho \in \mathbb{R}$ et $f(x) = \langle Q, xx^T \rangle$, on construit :

$$f_\rho(x) = \underbrace{\langle Q, xx^T \rangle}_{f(x)} + \rho \sum_{i=1}^{n} \underbrace{(x_i^2 - x_i)}_{0 \text{ si } x_i \in \{0,1\}}$$

On a bien $\forall \rho \in \mathbb{R}$, $f_\rho(x) = f(x)$ si $x \in \{0,1\}^n$.

Convexification : principe général

Idée : Modifier $f(x)$ pour la rendre convexe.

Méthode : Ajouter à $f(x)$ des fonctions qui s'annulent sur le domaine.

Exemple : si variables sont binaires ($x_i \in \{0,1\}$), on a $(x_i^2 - x_i) = 0$

Soient $\rho \in \mathbb{R}$ et $f(x) = \langle Q, xx^T \rangle$, on construit :

$$f_\rho(x) = \underbrace{\langle Q, xx^T \rangle}_{f(x)} + \rho \sum_{i=1}^{n} \underbrace{(x_i^2 - x_i)}_{0 \text{ si } x_i \in \{0,1\}}$$

On a bien $\forall \rho \in \mathbb{R}$, $f_\rho(x) = f(x)$ si $x \in \{0,1\}^n$.

$$Q = \begin{pmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{nn} \end{pmatrix} \not\succeq 0 \quad \text{devient} \quad Q_\rho = \begin{pmatrix} q_{11} + \rho & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{nn} + \rho \end{pmatrix}$$

Si ρ suffisamment grand, on a bien $Q_\rho \succeq 0$

Exemple : convexification du QAP

Ainsi pour le QAP, on obtient une famille de reformulation quadratiques convexes notée(QAP_ρ), qui dépend d'un paramètre scalaire ρ :

$$(QAP_\rho)\begin{cases} \min f_\rho(x) = \displaystyle\sum_{(i,j,k,l)\in\mathcal{I}^4} q_{ijkl}x_{ij}x_{kl} + \sum_{(i,j)\in\mathcal{I}^2} \rho(x_{ij}^2 - x_{ij}) \\ \text{s.t.} \displaystyle\sum_{i=1}^{n} x_{ij} = 1 \quad j \in \mathcal{I} \\ \displaystyle\sum_{j=1}^{n} x_{ij} = 1 \quad i \in \mathcal{I} \\ x_{ij} \in \{0,1\} \quad (i,j) \in \mathcal{I}^2 \end{cases}$$

Convexification : méthode de la ppvp (Hammer et Rubin)

Il faut maintenant choisir $\rho \in \mathbb{R}$ tel que :

1. La fonction $f_\rho(x)$ soit convexe (son Hessien Q_ρ soit SDP)
2. La valeur optimale de la relaxation continue de (QAP_ρ) (notée $(\overline{QAP}_\rho)$) soit la plus grande possible
   $\Longrightarrow$ on débute le b&b avec une bonne borne.

$$(VP)\begin{cases} \max\limits_{\rho\in\mathbb{R}} & v(\overline{QAP}_\rho) \\ \text{s.t.} & Q_\rho \succeq 0 \end{cases}$$

Convexification : méthode de la ppvp (Hammer et Rubin)

Il faut maintenant choisir $\rho \in \mathbb{R}$ tel que :

1. La fonction $f_\rho(x)$ soit convexe (son Hessien Q_ρ soit SDP)
2. La valeur optimale de la relaxation continue de (QAP_ρ) (notée $(\overline{QAP}_\rho)$) soit la plus grande possible
   $\Longrightarrow$ on débute le b&b avec une bonne borne.

$$(VP)\begin{cases} \max\limits_{\rho\in\mathbb{R}} & v(\overline{QAP}_\rho) \\ \text{s.t.} & Q_\rho \succeq 0 \end{cases}$$

Théorème (Hammer et Rubin) : Le scalaire $\rho^* \in \mathbb{R}$ qui maximise la valeur de $(\overline{QAP}_\rho)$ tout en rendant la matrice Q_ρ semi-définie positive est l'opposé de la plus petite valeur propre de Q, ici $\rho^* = -\lambda_{min}(Q)$.

Preuve : exercice

Convexification : méthode de la ppvp (Hammer et Rubin)

Il faut maintenant choisir $\rho \in \mathbb{R}$ tel que :

1. La fonction $f_\rho(x)$ soit convexe (son Hessien Q_ρ soit SDP)
2. La valeur optimale de la relaxation continue de (QAP_ρ) (notée $(\overline{QAP}_\rho)$) soit la plus grande possible
   $\Longrightarrow$ on débute le b&b avec une bonne borne.

$$(VP)\begin{cases} \max\limits_{\rho\in\mathbb{R}} & v(\overline{QAP}_\rho) \\ \text{s.t.} & Q_\rho \succeq 0 \end{cases}$$

Théorème (Hammer et Rubin) : Le scalaire $\rho^* \in \mathbb{R}$ qui maximise la valeur de $(\overline{QAP}_\rho)$ tout en rendant la matrice Q_ρ semi-définie positive est l'opposé de la plus petite valeur propre de Q, ici $\rho^* = -\lambda_{min}(Q)$.

Preuve : exercice

Avantage : la dimension du problème reformulé est la même que celle du problème initial

Convexification par la ppvp : Illustration graphique

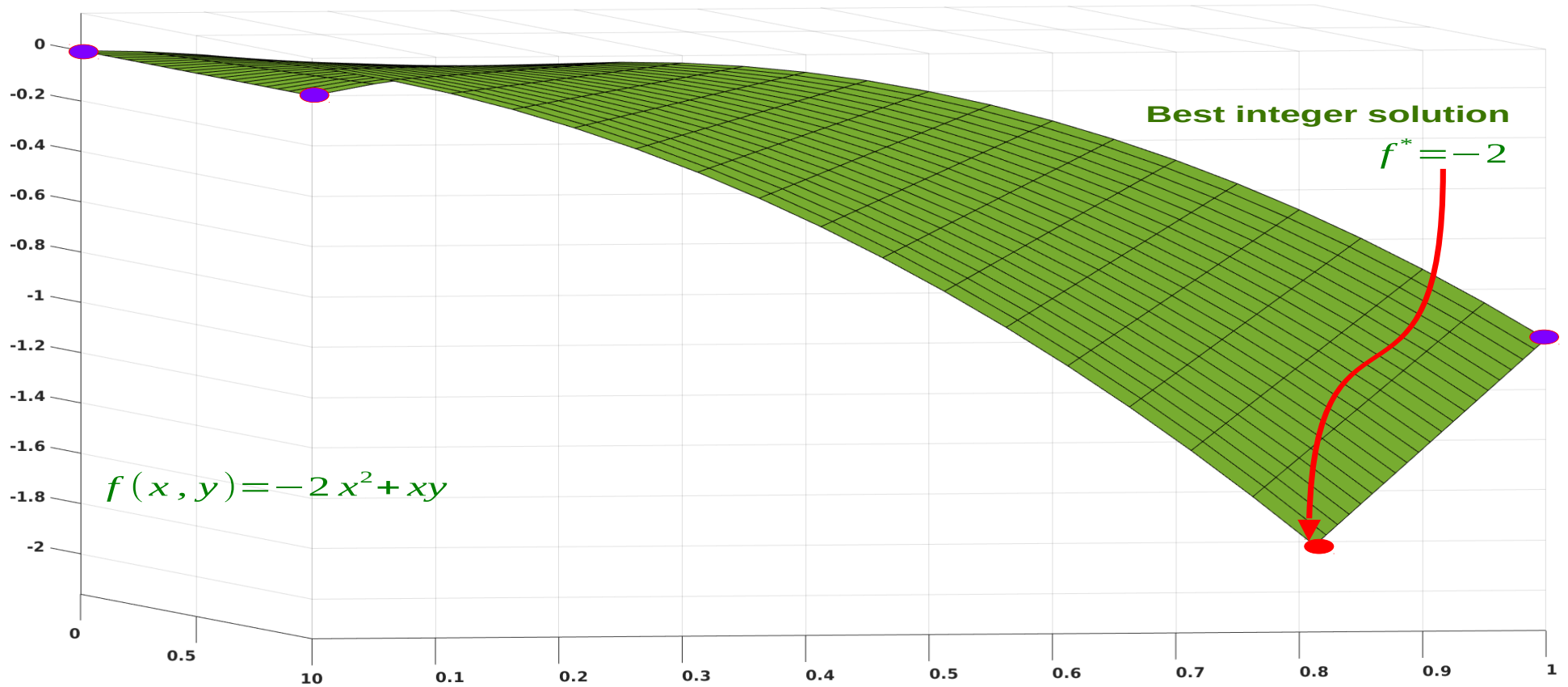
$$\begin{cases} \min\limits_{(x,y)\in\{0,1\}} & -2x^2 + xy \end{cases}$$

$$Q = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} \not\succeq 0$$

Convexification par la ppvp : Illustration graphique

$$\begin{cases} \min\limits_{(x,y)\in\{0,1\}} & -2x^2 + xy \end{cases}$$

$$Q = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} \not\succeq 0$$



Convexification par la ppvp : Illustration graphique

$$\begin{cases} \min\limits_{(x,y)\in\{0,1\}} & -2x^2 + xy \end{cases}$$

$$Q = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} \not\succeq 0$$

Soit $f_2(x,y) = \underbrace{-2x^2 + xy}_{f(x,y)} + \lambda \underbrace{(x^2 - x + y^2 - y)}_{=0 \text{ si } x \text{ et } y \text{ binaires}} = f(x,y)$

Convexification par la ppvp : Illustration graphique

$$\begin{cases} \min\limits_{(x,y)\in\{0,1\}} & -2x^2 + xy \end{cases}$$

$$Q = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} \not\succeq 0$$

Soit $f_2(x,y) = \underbrace{-2x^2 + xy}_{f(x,y)} + \lambda \underbrace{(x^2 - x + y^2 - y)}_{=0 \text{ si } x \text{ et } y \text{ binaires}} = f(x,y)$

Si $\lambda = -\lambda_{\min}(Q) = 2.1$,

$$f_2(x,y) = .1x^2 + 2.1y^2 + xy - 2.1(x+y)$$

avec $Q_\rho = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} + \begin{pmatrix} 2.1 & 0 \\ 0 & 2.1 \end{pmatrix} = \begin{pmatrix} .1 & 0.5 \\ 0.5 & 2.1 \end{pmatrix} \succeq 0$

Convexification par la ppvp : Illustration graphique

$$\left\{ \min_{(x,y)\in\{0,1\}} -2x^2 + xy \right. \quad \Leftrightarrow \left\{ \begin{array}{l} \min\limits_{(x,y)\in\{0,1\}} .1x^2 + 2.1y^2 \\ \qquad +xy - 2.1(x+y) \end{array} \right.$$

$$Q = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} \not\succeq 0 \qquad Q_\rho = \begin{pmatrix} .1 & 0.5 \\ 0.5 & 2.1 \end{pmatrix} \succeq 0$$

Soit $f_2(x,y) = \underbrace{-2x^2 + xy}_{f(x,y)} + \lambda \underbrace{(x^2 - x + y^2 - y)}_{=0 \text{ si } x \text{ et } y \text{ binaires}} = f(x,y)$

Si $\lambda = -\lambda_{\min}(Q) = 2.1$,

$$f_2(x,y) = .1x^2 + 2.1y^2 + xy - 2.1(x+y)$$

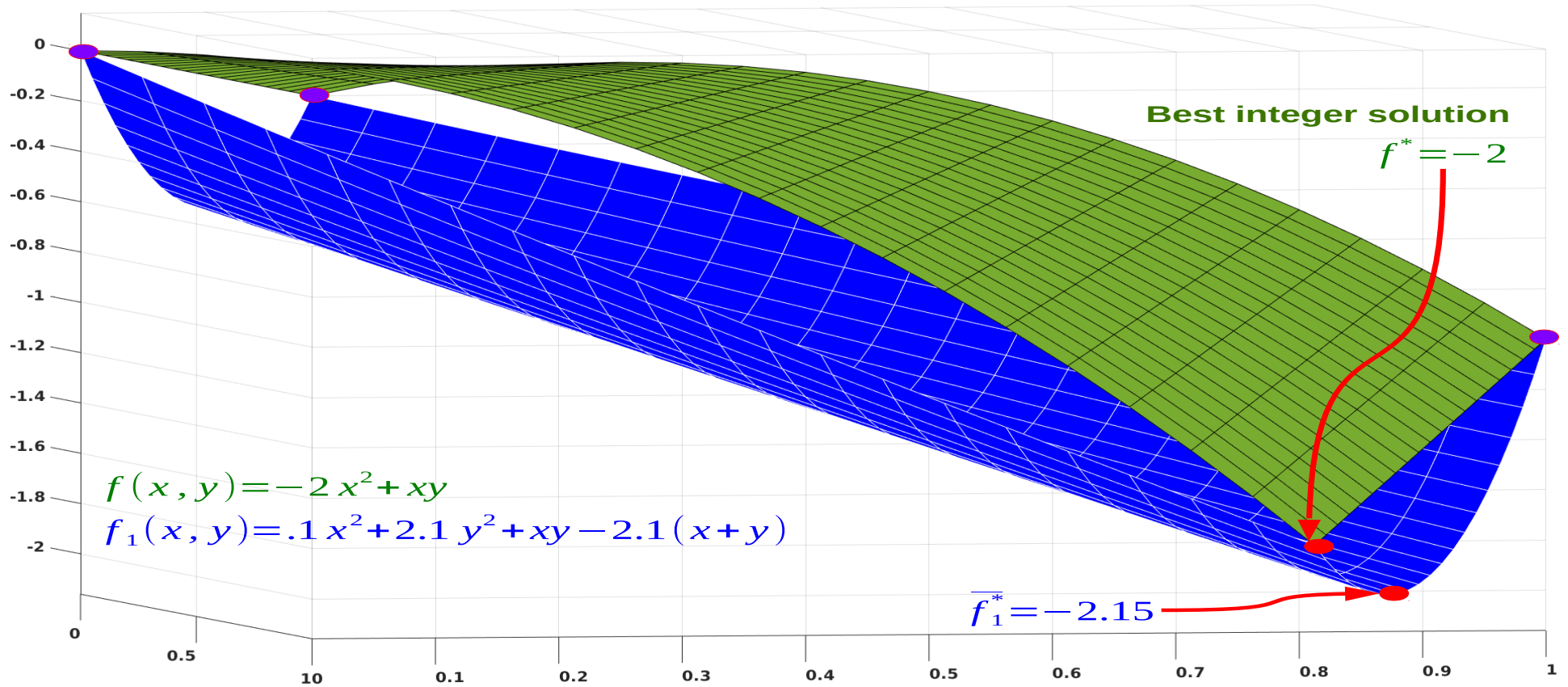
avec $Q_\rho = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} + \begin{pmatrix} 2.1 & 0 \\ 0 & 2.1 \end{pmatrix} = \begin{pmatrix} .1 & 0.5 \\ 0.5 & 2.1 \end{pmatrix} \succeq 0$

$\Rightarrow$ On a une équivalence sur tous les points binaires

Convexification par la ppvp : Illustration graphique

$$\left\{ \min_{(x,y)\in\{0,1\}} -2x^2 + xy \right. \quad \Leftrightarrow \quad \left\{ \min_{(x,y)\in\{0,1\}} .1x^2 + 2.1y^2 + xy - 2.1(x+y) \right.$$

$$Q = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} \not\succeq 0 \qquad Q_\rho = \begin{pmatrix} .1 & 0.5 \\ 0.5 & 2.1 \end{pmatrix} \succeq 0$$



Comparaison des bornes pour le QAP

Instance	opt	Lin	RLT	VP
nug06	86	0	86	-342.6
nug12	578	0	522.89	-4168
nug15	1150	0	1040.99	-10380.7

Convexification peut-on faire mieux ?

But : Améliorer (i.e. augmenter) la valeur de la relaxation continue.

Même idée : Modifier le Hessien de la fonction objectif.

Même méthode : Ajouter des fonctions qui s'annulent sur le domaine.

Mais considérer un vecteur de paramètre $\lambda \in \mathbb{R}^n$.

On considère la fonction

$$f_\lambda(x) = f(x) + \sum_{i=1}^{n} \lambda_i (x_i^2 - x_i)$$

On a bien $\forall \lambda \in \mathbb{R}^n$, $f_\lambda(x) = f(x)$ si $x \in \{0,1\}^n$.

Convexification peut-on faire mieux ?

But : Améliorer (i.e. augmenter) la valeur de la relaxation continue.

Même idée : Modifier le Hessien de la fonction objectif.

Même méthode : Ajouter des fonctions qui s'annulent sur le domaine.

Mais considérer un vecteur de paramètre $\lambda \in \mathbb{R}^n$.

On considère la fonction

$$f_\lambda(x) = f(x) + \sum_{i=1}^{n} \lambda_i (x_i^2 - x_i)$$

On a bien $\forall \lambda \in \mathbb{R}^n$, $f_\lambda(x) = f(x)$ si $x \in \{0,1\}^n$.

$$Q = \begin{pmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{nn} \end{pmatrix} \not\succeq 0 \quad \text{devient} \quad Q_\lambda = \begin{pmatrix} q_{11} + \lambda_1 & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{nn} + \lambda_n \end{pmatrix}$$

Il existe λ tel que $Q_\lambda \succeq 0$ ($\lambda_i = -\lambda_{min}(Q)$)

Quadratic Convex Reformulation (QCR) (Billionnet et Elloumi)

Ainsi pour le QAP, on obtient une famille de reformulation quadratiques convexes notée(QAP_λ), qui dépend d'un paramètre vectoriel $\lambda \in \mathbb{R}^{n^2}$:

$$(QAP_\lambda)\begin{cases} \min f_\lambda(x) = \displaystyle\sum_{(i,j,k,l)\in\mathcal{I}^4} q_{ijkl}x_{ij}x_{kl} + \sum_{(i,j)\in\mathcal{I}^2} \lambda_{ij}(x_{ij}^2 - x_{ij}) \\ \text{s.t.} \displaystyle\sum_{i=1}^{n} x_{ij} = 1 \quad j \in \mathcal{I} \\ \displaystyle\sum_{j=1}^{n} x_{ij} = 1 \quad i \in \mathcal{I} \\ x_{ij} \in \{0,1\} \quad (i,j) \in \mathcal{I}^2 \end{cases}$$

Quadratic Convex Reformulation (QCR) (Billionnet et Elloumi)

Il faut maintenant choisir $\lambda \in \mathbb{R}^{n^2}$ tel que :

1. La fonction $f_\lambda(x)$ soit convexe (son Hessien Q_λ soit SDP)
2. La valeur optimale de la relaxation continue de (QAP_λ) (notée $(\overline{QAP}_\lambda)$) soit la plus grande possible
   $\Longrightarrow$ on débute le b&b avec une bonne borne.

$$(QCR)\begin{cases} \max\limits_{\lambda \in \mathbb{R}^{n^2}} & v(\overline{QAP}_\lambda) \\ \text{s.t.} & Q_\lambda \succeq 0 \end{cases}$$

Relaxation semi-définie : exemple cas $\{0, 1\}$

$$\left\{ \min_{(x,y)\in\{0,1\}} -2x^2 + xy \right.$$

```
valeur opt : -2
sol opt : (1,0)
```

Relaxation semi-définie : exemple cas $\{0, 1\}$

$$\left\{ \min_{(x,y)\in\{0,1\}} -2x^2 + xy \right. \iff \begin{cases} \min & -2Z_{11} + 0.5(Z_{12} + Z_{21}) \\ \text{s.t.} & Z = \begin{pmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{pmatrix} = \begin{pmatrix} x^2 & xy \\ yx & y^2 \end{pmatrix} \\ & (x, y) \in \{0, 1\}^2, \quad Z \in \mathcal{S}^n \end{cases}$$

- On introduit Z une matrice variable qui modélise les produits xy^T.
  $\rightarrow$ On linéarise la fonction objectif.

Relaxation semi-définie : exemple cas $\{0, 1\}$

$$\left\{ \min_{(x,y)\in\{0,1\}} -2x^2 + xy \right. \xrightarrow{relax} \begin{cases} \min & -2Z_{11} + 0.5(Z_{12} + Z_{21}) \\ \text{s.t.} & \begin{pmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{pmatrix} - \begin{pmatrix} x^2 & xy \\ yx & y^2 \end{pmatrix} \succeq 0 \\ & (x, y) \in \{0, 1\}^2, \quad Z \in \mathcal{S}^n \end{cases}$$

- On introduit Z une matrice variable qui modélise les produits xy^T.
  $\rightarrow$ On linéarise la fonction objectif.
- On relâche la contrainte $Z - \begin{pmatrix} x \\ y \end{pmatrix} (x\ y) = 0$ en une contrainte SDP.

Relaxation semi-définie : exemple cas $\{0, 1\}$

$$\left\{ \min_{(x,y)\in\{0,1\}} -2x^2 + xy \right. \xrightarrow{relax} \begin{cases} \min & -2Z_{11} + 0.5(Z_{12} + Z_{21}) \\ \text{s.t.} & W = \begin{pmatrix} 1 & x & y \\ x & Z_{11} & Z_{12} \\ y & Z_{21} & Z_{22} \end{pmatrix} \succeq 0 \\ & (x, y) \in \{0, 1\}^2, \quad W \in \mathcal{S}^{n+1} \end{cases}$$

- On introduit Z une matrice variable qui modélise les produits xy^T.
  $\rightarrow$ On linéarise la fonction objectif.
- On relâche la contrainte $Z - \begin{pmatrix} x \\ y \end{pmatrix} (x\ y) = 0$ en une contrainte SDP.
- On ré-écrit la contrainte SDP grâce au complément de Shur.

Relaxation semi-définie : exemple cas $\{0, 1\}$

$$\left\{ \min_{(x,y)\in\{0,1\}} -2x^2 + xy \right. \xrightarrow{relax} \begin{cases} \min & -2Z_{11} + 0.5(Z_{12} + Z_{21}) \\ \text{s.t.} & Z_{11} = x \\ & Z_{22} = y \\ & W = \begin{pmatrix} 1 & x & y \\ x & Z_{11} & Z_{12} \\ y & Z_{21} & Z_{22} \end{pmatrix} \succeq 0 \\ & W \in \mathcal{S}^{n+1} \end{cases}$$

- On introduit Z une matrice variable qui modélise les produits xy^T.
  $\rightarrow$ On linéarise la fonction objectif.
- On relâche la contrainte $Z - \begin{pmatrix} x \\ y \end{pmatrix} (x\ y) = 0$ en une contrainte SDP.
- On ré-écrit la contrainte SDP grâce au complément de Shur.
- On ré-écrit les contraintes de binarité en $x = x^2$ et $y = y^2$.

Relaxation semi-définie : exemple cas $\{0, 1\}$

$$\left\{ \min_{(x,y)\in\{0,1\}} -2x^2 + xy \right. \xrightarrow{relax} \begin{cases} \min & -2x + t \\ \text{s.t.} & W = \begin{pmatrix} 1 & x & y \\ x & x & t \\ y & t & y \end{pmatrix} \succeq 0 \\ & W \in \mathcal{S}^{n+1} \end{cases}$$

- On introduit Z une matrice variable qui modélise les produits xy^T.
  $\rightarrow$ On linéarise la fonction objectif.
- On relâche la contrainte $Z - \begin{pmatrix} x \\ y \end{pmatrix} (x\ y) = 0$ en une contrainte SDP.
- On ré-écrit la contrainte SDP grâce au complément de Shur.
- On ré-écrit les contraintes de binarité en $x = x^2$ et $y = y^2$.
- $\text{opt}(\text{SDP}) = -2$, une solution optimale $W^* = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

Quadratic Convex Reformulation (QCR) (Billionnet et Elloumi)

Théorème : λ^* se déduit du vecteur de variables duales de (SDP_{QCR}).

$$(QCR)\begin{cases} \max\limits_{\lambda\in\mathbb{R}^{n^2}} & v(\overline{QAP}_\lambda) \\ \text{s.t.} & Q_\lambda \succeq 0 \end{cases} \quad \Leftrightarrow \quad (SDP_{QCR})\begin{cases} \min f(X,x) = \langle Q, X\rangle \\ \text{s.t.} \sum\limits_{i=1}^{n} x_{ij} = 1 \\ \sum\limits_{j=1}^{n} x_{ij} = 1 \\ X_{ijij} - x_{ij} = 0 \quad \leftarrow \lambda_{ij} \\ \begin{pmatrix} 1 & x \\ x^T & X \end{pmatrix} \succeq 0 \\ x \in \mathbb{R}^{n^2} \quad X \in \mathcal{S}_{n^2} \end{cases}$$

Théorème : on a $v(QCR) = v(SDP_{QCR})$

Reformulation quadratique convexe : exemple cas binaire

$$\begin{cases} \min\limits_{(x,y)\in\{0,1\}} & -2x^2 + xy \end{cases}$$

$$Q = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} \not\succeq 0$$

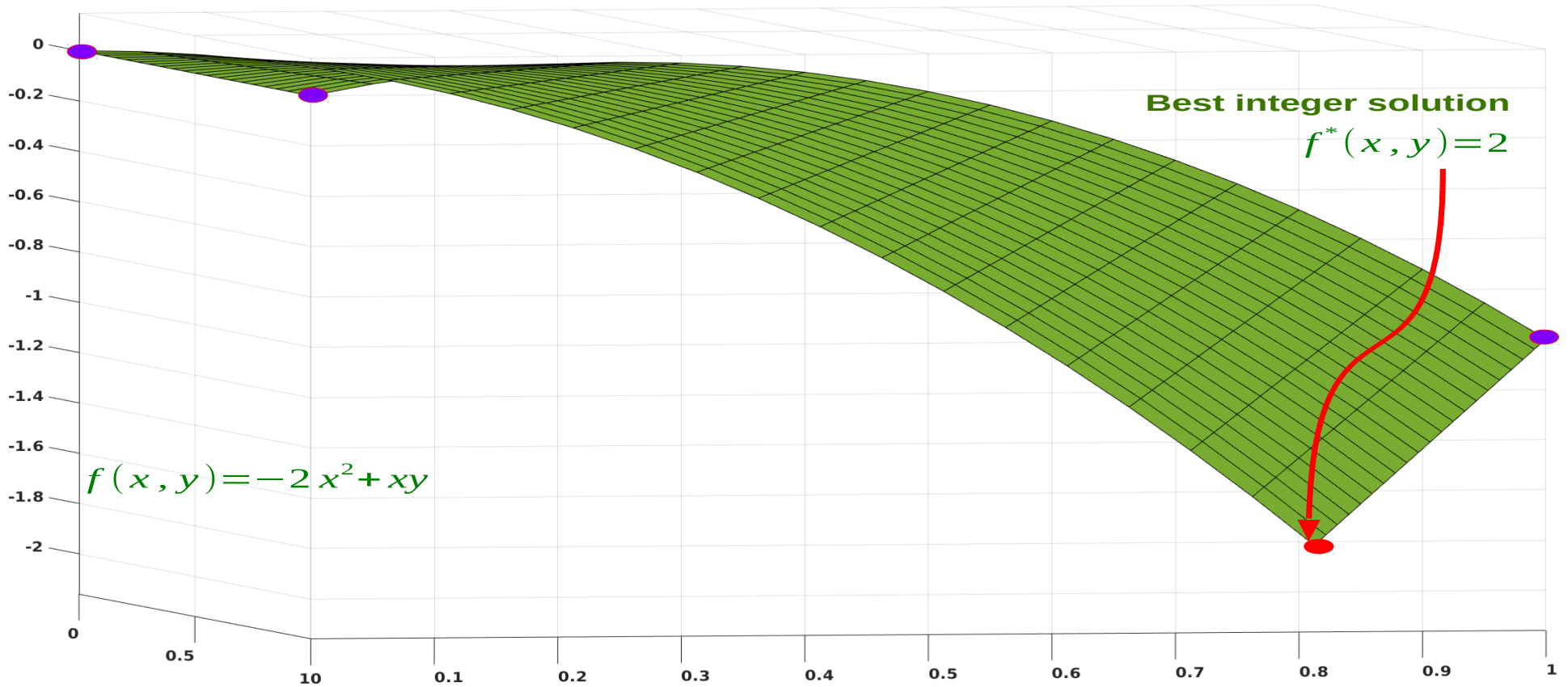
```
valeur opt : -2
sol opt : (1,0)
```

Reformulation quadratique convexe : exemple cas binaire

$$\begin{cases} \min\limits_{(x,y)\in\{0,1\}} & -2x^2 + xy \end{cases}$$

$$Q = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} \not\succeq 0$$

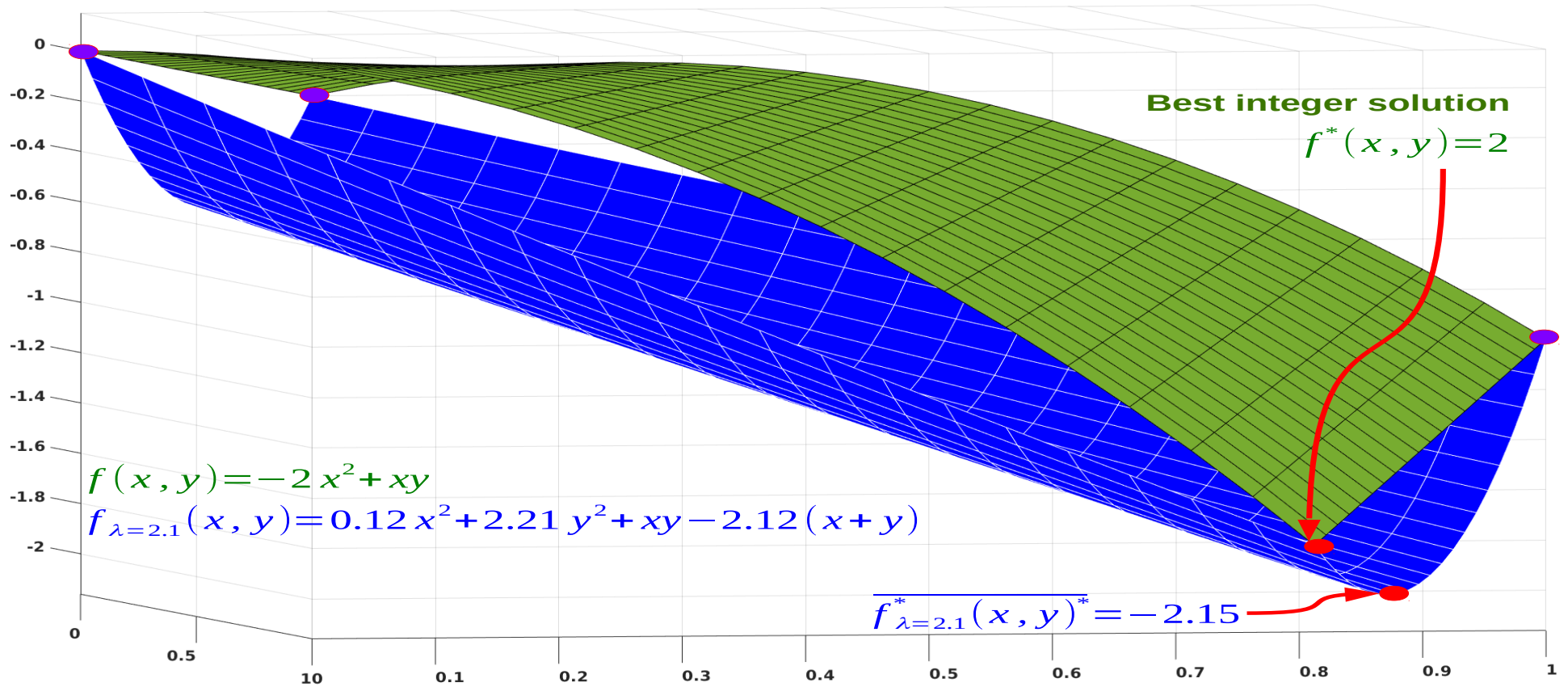
```
valeur opt : -2
sol opt : (1,0)
```



Reformulation quadratique convexe : exemple cas binaire

$$\left\{ \min_{(x,y)\in\{0,1\}} -2x^2 + xy \right. \Leftrightarrow \left\{ \min_{(x,y)\in\{0,1\}} .1x^2 + 2.1y^2 + xy - 2.1(x+y) \right.$$

$$Q = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} \not\succeq 0 \qquad S = \begin{pmatrix} .1 & 0.5 \\ 0.5 & 2.1 \end{pmatrix} \succeq 0$$



Reformulation quadratique convexe : exemple cas binaire

$$\left\{ \min_{(x,y)\in\{0,1\}} -2x^2 + xy \right. \Leftrightarrow \left\{ \begin{array}{l} \min\limits_{(x,y)\in\{0,1\}} .1x^2 + 2.1y^2 \\ \qquad +xy - 2.1(x+y) \end{array} \right.$$

$$Q = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} \not\succeq 0 \qquad S = \begin{pmatrix} .1 & 0.5 \\ 0.5 & 2.1 \end{pmatrix} \succeq 0$$

Soient λ_1, λ_2 et $f_{\lambda_1,\lambda_2}(x,y) = \underbrace{-2x^2 + xy}_{f(x,y)} + \lambda_1 \underbrace{(x^2 - x)}_{=0} + \lambda_2 \underbrace{(y^2 - y)}_{=0} = f(x,y)$

Reformulation quadratique convexe : exemple cas binaire

$$\left\{ \min_{(x,y)\in\{0,1\}} -2x^2 + xy \right. \Leftrightarrow \left\{ \min_{(x,y)\in\{0,1\}} .1x^2 + 2.1y^2 + xy - 2.1(x+y) \right.$$

$$Q = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} \not\succeq 0 \qquad S = \begin{pmatrix} .1 & 0.5 \\ 0.5 & 2.1 \end{pmatrix} \succeq 0$$

Soient λ_1, λ_2 et $f_{\lambda_1,\lambda_2}(x,y) = \underbrace{-2x^2 + xy}_{f(x,y)} + \lambda_1 \underbrace{(x^2 - x)}_{=0} + \lambda_2 \underbrace{(y^2 - y)}_{=0} = f(x,y)$

Calculer λ_1 et λ_2 qui maximisent la borne par relaxation continue

$\Rightarrow$ utiliser la programmation semi-définie positive

$$\left\{ \begin{array}{ll} \min & -2W_{22} + 0.5(W_{23} + W_{32}) \\ \text{s.t.} & W_{22} - 0.5(W_{12} + W_{21}) = 0 \quad \leftarrow \lambda_1 \\ & W_{33} - 0.5(W_{13} + W_{31}) = 0 \quad \leftarrow \lambda_2 \\ & W_{11} = 1 \quad \leftarrow \rho \\ & W \succeq 0,\ W \in \mathcal{S}^{n+1} \end{array} \right.$$

$$\left\{ \begin{array}{ll} \max & -\rho \\ \text{s.t.} & \\ & \begin{pmatrix} \rho & -0.5\lambda_1 & -0.5\lambda_2 \\ -0.5\lambda_1 & -2+\lambda_1 & 0.5 \\ -0.5\lambda_2 & 0.5 & 0+\lambda_2 \end{pmatrix} \succeq 0 \end{array} \right.$$

$$\lambda_1^* = 4, \lambda_2^* = 1, \rho^* = 2$$

Reformulation quadratique convexe : exemple cas binaire

$$\left\{ \min_{(x,y)\in\{0,1\}} -2x^2 + xy \right. \Leftrightarrow \left\{ \min_{(x,y)\in\{0,1\}} .1x^2 + 2.1y^2 + xy - 2.1(x+y) \right. \Leftrightarrow \left\{ \min_{(x,y)\in\{0,1\}} 2x^2 + y^2 + xy - 4x - y \right.$$

$$Q = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} \not\succeq 0 \qquad S = \begin{pmatrix} .1 & 0.5 \\ 0.5 & 2.1 \end{pmatrix} \succeq 0 \qquad S^* = \begin{pmatrix} 2 & 0.5 \\ 0.5 & 1 \end{pmatrix} \succeq 0$$

Soient λ_1, λ_2 et $f_{\lambda_1,\lambda_2}(x,y) = \underbrace{-2x^2 + xy}_{f(x,y)} + \lambda_1 \underbrace{(x^2 - x)}_{=0} + \lambda_2 \underbrace{(y^2 - y)}_{=0} = f(x,y)$

Calculer λ_1 et λ_2 qui maximisent la borne par relaxation continue

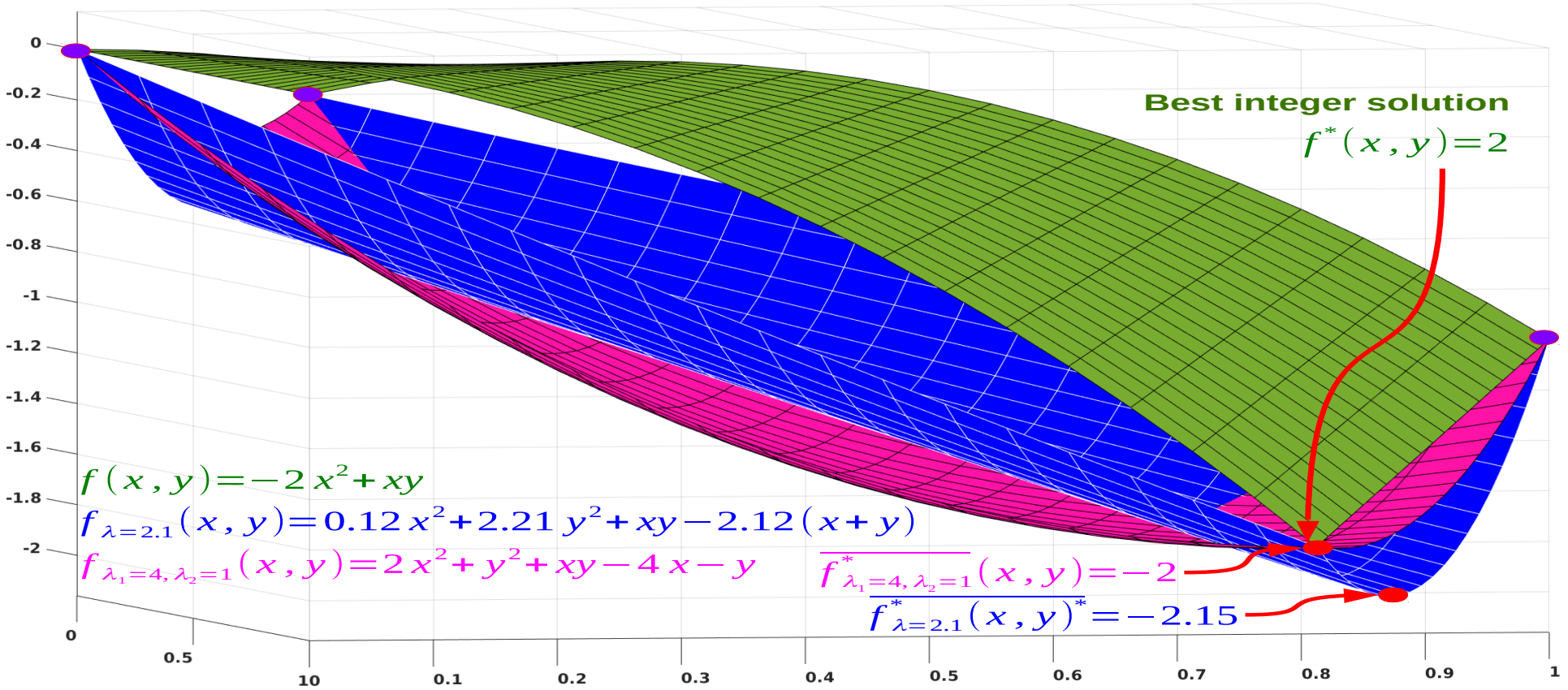
On a un nouveau Hessien $S^* = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} + \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0.5 \\ 0.5 & 1 \end{pmatrix} \succeq 0$

Quadratic Convex Reformulation [Billionnet, Elloumi, Plateau, 07-09]

Reformulation quadratique convexe : exemple cas binaire

$$\left\{ \min_{(x,y)\in\{0,1\}} -2x^2 + xy \right. \Leftrightarrow \left\{ \min_{(x,y)\in\{0,1\}} .1x^2 + 2.1y^2 + xy - 2.1(x+y) \right. \Leftrightarrow \left\{ \min_{(x,y)\in\{0,1\}} 2x^2 + y^2 + xy - 4x - y \right.$$

$$Q = \begin{pmatrix} -2 & 0.5 \\ 0.5 & 0 \end{pmatrix} \not\succeq 0 \qquad S = \begin{pmatrix} .1 & 0.5 \\ 0.5 & 2.1 \end{pmatrix} \succeq 0 \qquad S^* = \begin{pmatrix} 2 & 0.5 \\ 0.5 & 1 \end{pmatrix} \succeq 0$$



Amélioration de QCR (Billionnet, Elloumi et Plateau - 2009)

Remarque : Le paramètre λ ne modifie que la diagonale de Q.

Idée : ajouter d'autres fonctions quadratiques nulles qui modifie Q sur ses termes non-diagonaux.

Exemple : Si (QP) a m contraintes d'égalités linéaires :

$$\sum_{i=1}^{n} a_{ri} x_i = b_r \quad \forall r = 1, \dots m$$

Il est possible de les "quadratiser" - deux possibilités :

- Soit en les multipliant par les variables x_j (RLT) :

$$\Longrightarrow \sum_{i=1}^{n} a_{ri} x_i x_j = b_r x_j \quad \forall r = 1, \dots, m, \text{ et } j = 1, \dots, n$$

- soit en les élevant au carré, puis en les sommant :

$$\Longrightarrow \sum_{r=1}^{m} (\sum_{i=1}^{n} a_{ri} x_i - b_r)^2$$

Amélioration de QCR (Billionnet, Elloumi et Plateau)

On introduit un paramètre scalaire $\beta \in \mathbb{R}$ et la fonction :

$$f_{\lambda,\beta}(x) = \underbrace{f(x) + \sum_{i=1}^{n} \lambda_i (x_i^2 - x_i)}_{f_\lambda(x)} + \beta \sum_{r=1}^{m} \underbrace{(\sum_{i=1}^{n} a_{ri} x_i - b_r)^2}_{0 \text{ si } \sum_{i=1}^{n} a_{ri} x_i = b_r}$$

On a bien : $f_{\lambda,\beta}(x) = f(x)$ si $x_{ij} \in \{0,1\}$ et $\sum_{i=1}^{n} a_{ri} x_i = b_r \quad \forall r = 1, \dots m$

Amélioration de QCR (Billionnet, Elloumi et Plateau)

On introduit un paramètre scalaire $\beta \in \mathbb{R}$ et la fonction :

$$f_{\lambda,\beta}(x) = \underbrace{f(x) + \sum_{i=1}^{n} \lambda_i (x_i^2 - x_i)}_{f_\lambda(x)} + \beta \sum_{r=1}^{m} \underbrace{(\sum_{i=1}^{n} a_{ri} x_i - b_r)^2}_{0 \text{ si } \sum_{i=1}^{n} a_{ri} x_i = b_r}$$

On a bien : $f_{\lambda,\beta}(x) = f(x)$ si $x_{ij} \in \{0,1\}$ et $\sum_{i=1}^{n} a_{ri} x_i = b_r \quad \forall r = 1, \dots m$ avec

$$Q_{\lambda,\beta} = \begin{pmatrix} q_{11} + \lambda_1 + \beta \sum_{r=1}^{m} a_{r1}^2 & \dots & q_{1n} + \beta \sum_{r=1}^{m} a_{r1} a_{rn} \\ \vdots & \ddots & \vdots \\ q_{1n} + \beta \sum_{r=1}^{m} a_{r1} a_{rn} & \dots & q_{nn} + \lambda_n + \beta \sum_{r=1}^{m} a_{rn}^2 \end{pmatrix}$$

Il existe λ et β tel que $Q_{\lambda,\beta} \succeq 0$ ($\lambda_i = -\lambda_{min}(Q)$ et $\beta = 0$)

Amélioration de QCR : QAP

On obtient la famille de reformulations quadratiques convexes :

$$(QAP_{\lambda,\beta}) \begin{cases} \min f_{\lambda,\beta}(x) = f_\lambda(x) + \beta \sum_{r=1}^{m} (\sum_{i=1}^{n} a_{ri} x_i - b_r)^2 \\ \text{s.t.} \sum_{i=1}^{n} x_{ij} = 1 \quad j \in \mathcal{I} \\ \sum_{j=1}^{n} x_{ij} = 1 \quad i \in \mathcal{I} \\ x_{ij} \in \{0,1\} \quad (i,j) \in \mathcal{I}^2 \end{cases}$$

Amélioration de QCR (Billionnet, Elloumi et Plateau)

Il faut maintenant choisir $\lambda \in \mathbb{R}^{n^2}$ et $\beta \in \mathbb{R}$ tel que :

1. La fonction $f_{\lambda,\beta}(x)$ soit convexe (son Hessien $Q_{\lambda,\beta}$ soit SDP)
2. La valeur optimale de la relaxation continue de $(QAP_{\lambda,\beta})$ (notée $(\overline{QAP}_{\lambda,\beta})$) soit la plus grande possible
   $\Longrightarrow$ on débute le b&b avec une bonne borne.

$$(QCR')\begin{cases} \max\limits_{\lambda\in\mathbb{R}^{n^2},\beta\in\mathbb{R}} & v(\overline{QAP}_{\lambda,\beta}) \\ \text{s.t.} & Q_{\lambda,\beta} \succeq 0 \end{cases}$$

Amélioration de QCR (Billionnet, Elloumi et Plateau - 2009)

Théorème : (λ^*, β^*) se déduisent des variables duales de (SDP'_{QCR})

$$(QCR') \begin{cases} \max\limits_{\lambda \in \mathbb{R}^{n^2}, \beta \in \mathbb{R}} & v(\overline{QAP}_{\lambda,\beta}) \\ \text{s.t.} & Q_{\lambda,\beta} \succeq 0 \end{cases} \Leftrightarrow \quad (SDP'_{QCR}) \begin{cases} \min f(X,x) = \langle Q, X \rangle \\ \text{s.t.} \sum\limits_{i=1}^{n} x_{ij} = 1 \\ \sum\limits_{j=1}^{n} x_{ij} = 1 \\ X_{ijij} - x_{ij} = 0 \quad \leftarrow \lambda_{ij} \\ \sum\limits_{r=1}^{m} (\sum\limits_{i=1}^{n} (\sum\limits_{j=1}^{n} a_{ri} a_{rj} X_{ij} - 2 a_{ri} b_r x_i)) = -\sum\limits_{r=1}^{m} b_r^2 \leftarrow \beta \\ \begin{pmatrix} 1 & x \\ x^T & X \end{pmatrix} \succeq 0 \\ x \in \mathbb{R}^{n^2} \quad X \in \mathcal{S}_{n^2} \end{cases}$$

Théorème : on a $v(QCR') = v(SDP_{QCR'})$

Rque : Quadratisation des contraintes linéaires d'égalité

Les deux versions de quadratisation des contraintes linéaires d'égalités sont équivalentes du point de vue de la borne SDP.

Rappel des deux versions :

- les multiplier par les variables x_j (RLT) :

$$\Longrightarrow \sum_{i=1}^{n} a_{ri} x_i x_j = b_r x_j \quad \forall r = 1, \ldots, m, \text{ et } j = 1, \ldots, n$$

- les élever au carré, puis en les sommant : :

$$\Longrightarrow \sum_{r=1}^{m} (\sum_{i=1}^{n} a_{ri} x_i - b_r)^2$$

Comparaison des bornes pour le QAP

Instance	opt	Lin	RLT	VP	QCR'
nug06	86	0	86	-342.6	-1.1
nug12	578	0	522.89	-4168	-215.8
nug15	1150	0	1040.99	-10380.7	-823.1

Utilisation de la structure du (QAP)

Pour amélior la borne : déterminer des fonctions nulles sur le domaine

Pour le (QAP), nous avons les fonctions nulles suivantes :

- $x_{ij}x_{il} = 0$: un équipement ne peut-être affecté à plus d'un site.
- $x_{ij}x_{kj} = 0$: un site ne peut être affecté à plus d'un équipement.

En introduisant, en plus de λ_{ij}, les paramètres scalaires $\delta_{ijl}, \delta'_{ijk}$ et :

$$f_{\lambda,\delta,\delta'}(x) = \underbrace{f(x) + \sum_{(i,j)} \lambda_{ij}(x_{ij}^2 - x_{ij})}_{f_\lambda(x)} + \sum_{(i,j,l)} \delta_{ijl}x_{ij}x_{il} + \sum_{(i,j,k)} \delta'_{ijk}x_{ij}x_{kj}$$

On a $f_{\lambda,\delta,\delta'}(x) = f(x)$ sur le domaine du (QAP).

Utilisation de la structure du (QAP)

On obtient une famille de reformulation du (QAP) :

$$(QAP_{\lambda,\delta,\delta'}) \begin{cases} \min f_{\lambda,\delta,\delta'}(x) = f_\lambda(x) + \sum_{(i,j,l)} \delta_{ijl} x_{ij} x_{il} + \sum_{(i,j,k)} \delta'_{ijk} x_{ij} x_{kj} \\ \text{s.t.} \sum_{i=1}^{n} x_{ij} = 1 \quad j \in \{1,\dots,n\} \\ \sum_{j=1}^{n} x_{ij} = 1 \quad i \in \{1,\dots,n\} \\ x_{ij} \in \{0,1\} \quad (i,j) \in \mathcal{I}^2 \end{cases}$$

On note $Q_{\lambda,\delta,\delta'}$ le hessien de la nouvelle fonction objectif.

Utilisation de la structure du (QAP)

$$f_{\lambda,\delta,\delta'}(x) = f_\lambda(x) + \sum_{(i,j,l)} \delta_{ijl} x_{ij} x_{il} + \sum_{(i,j,k)} \delta'_{ijk} x_{ij} x_{kj}$$

Nous cherchons les paramètres $(\lambda, \delta, \delta')$ tels que :

i. $Q_{\lambda,\delta,\delta'}$ est semidefinie positive,

ii. La valeur de la relaxation continue du $(QAP_{\lambda,\delta,\delta'})$ soit la plus grande possible.

Cela revient à résoudre le problème suivant (CP) :

$$(CP)\left\{ \begin{array}{c} \max \\ Q_{\lambda,\delta,\delta'} \succeq 0 \end{array} \{v(\overline{QAP}_{\lambda,\delta,\delta'})\} \right.$$

où $(\overline{QAP}_{\lambda,\delta,\delta'})$ est la relaxation continue de $(QAP_{\lambda,\delta,\delta'})$

Utilisation de la structure du (QAP)

Théorème : $(\lambda^*, \delta^*, \delta'^*)$ se déduisent des variables duales de (SDP_{R2})

$$(CP)\left\{ \max_{Q_{\lambda,\delta,\delta'} \succeq 0} \{v(\overline{QAP}_{\lambda,\delta,\delta'})\} \right. \quad \Leftrightarrow \quad (SDP_{R2}) \begin{cases} \min f(X,x) = \langle Q, X \rangle \\ \text{s.t.} \sum_{i=1}^{n} x_{ij} = 1 \\ \sum_{j=1}^{n} x_{ij} = 1 \\ X_{ijij} - x_{ij} = 0 \quad \leftarrow \lambda_{ij} \\ X_{ijil} = 0 \quad \leftarrow \delta_{ijl} \\ X_{ijkj} = 0 \quad \leftarrow \delta'_{ijk} \\ \begin{pmatrix} 1 & x \\ x^T & X \end{pmatrix} \succeq 0 \\ x \in \mathbb{R}^{n^2} \quad X \in \mathcal{S}_{n^2} \end{cases}$$

Théorème : on a $v(CP) = v(SDP_{R2})$.

Comparaison des bornes pour le QAP

Instance	opt	Lin	RLT	VP	QCR'	R2
nug06	86	0	86	-342.6	-1.1	81.3
nug12	578	0	522.89	-4168	-215.8	524.1
nug15	1150	0	1040.99	-10380.7	-823.1	1053.7

Remarques

Avantages :

- On reste dans le même espace de variables que le problème initial.
- Calculer $(\lambda^*, \delta^*, \delta'^*)$ nécessite la résolution d'un programme SDP de taille raisonnable (i.e. $\mathcal{O}(n^4)$ variables, et $\mathcal{O}(n^3)$ contraintes).
- La taille (n^2) de $(QAP_{\lambda^*,\delta^*,\delta'^*})$, qui sera résolu à chaque nœud du b&b est raisonnable.

Mais en restant dans le même espace de variables, on limite la qualité de la borne à la racine du b&b.

Comment faire mieux ?

Des reformulations pour le (QAP) dans un espace étendu

On considère à nouveau n^2 nouvelles variables y telles que $x_{ij}x_{kl} = y_{ijkl}$

On introduit un paramètre ϕ_{ijkl}, et on considère la fonction quadratique :

$$f_{\lambda,\delta,\delta',\phi}(x,y) = f_{\lambda,\delta,\delta'}(x) + \sum_{(i,j,k,l)} \phi_{ijkl}(x_{ij}x_{kl} - y_{ijkl}) = f(x)$$

Des reformulations pour le (QAP) dans un espace étendu

On considère à nouveau n^2 nouvelles variables y telles que $x_{ij}x_{kl} = y_{ijkl}$

On introduit un paramètre ϕ_{ijkl}, et on considère la fonction quadratique :

$$f_{\lambda,\delta,\delta',\phi}(x,y) = f_{\lambda,\delta,\delta'}(x) + \sum_{(i,j,k,l)} \phi_{ijkl}(x_{ij}x_{kl} - y_{ijkl}) = f(x)$$

On a bien : $f_{\lambda,\delta,\delta',\phi}(x,y) = f(x)$

lorsque $x_{ij} \in \{0,1\}$ et $\sum_{i=1}^{n} a_{ri}x_i = b_r$ et $x_{ij}x_{kl} = y_{ijkl}$

avec

$$Q_{\lambda,\delta,\delta',\phi} = \begin{pmatrix} q_{\lambda,\delta,\delta',11} + \phi_{11} & \cdots & q_{\lambda,\delta,\delta',1n} + \phi_{1n} \\ \vdots & \ddots & \vdots \\ q_{\lambda,\delta,\delta',n1} + \phi_{n1} & \cdots & q_{\lambda,\delta,\delta',nn} + \phi_{nn} \end{pmatrix}$$

Des reformulations pour le (QAP) dans un espace étendu

On considère à nouveau n^2 nouvelles variables y telles que $x_{ij}x_{kl} = y_{ijkl}$

On introduit un paramètre ϕ_{ijkl}, et on considère la fonction quadratique :

$$f_{\lambda,\delta,\delta',\phi}(x,y) = f_{\lambda,\delta,\delta'}(x) + \sum_{(i,j,k,l)} \phi_{ijkl}(x_{ij}x_{kl} - y_{ijkl}) = f(x)$$

On a bien : $f_{\lambda,\delta,\delta',\phi}(x,y) = f(x)$

lorsque $x_{ij} \in \{0,1\}$ et $\sum_{i=1}^{n} a_{ri}x_i = b_r$ et $x_{ij}x_{kl} = y_{ijkl}$

On linéarise les égalités $x_{ij}x_{kl} = y_{ijkl}$ avec Fortet :

$$x_{ij}x_{kl} - y_{ijkl} = 0 \Leftrightarrow \begin{cases} y_{ijkl} \geq x_{ij} + x_{kl} - 1 \\ y_{ijkl} \geq 0 \end{cases}$$

Puisque tous les coefficients sont positifs

Des reformulations pour le (QAP) dans un espace étendu

On considère à nouveau n^2 nouvelles variables y telles que $x_{ij}x_{kl} = y_{ijkl}$

On introduit un paramètre ϕ_{ijkl}, et on considère la fonction quadratique :

$$f_{\lambda,\delta,\delta',\phi}(x,y) = f_{\lambda,\delta,\delta'}(x) + \sum_{(i,j,k,l)} \phi_{ijkl}(x_{ij}x_{kl} - y_{ijkl}) = f(x)$$

$$(QAP_{\lambda,\delta,\delta',\phi}) \begin{cases} \min \quad f_{\lambda,\delta,\delta',\phi}(x,y) \\ \text{s.t.} \sum_{i=1}^{n} x_{ij} = 1 \quad j \in \{1,\dots,n\} \\ \sum_{j=1}^{n} x_{ij} = 1 \quad i \in \{1,\dots,n\} \\ x_{ij} \in \{0,1\} \quad (i,j) \in \{1,\dots,n\}^2 \\ y_{ijkl} \geq x_{ij} + x_{kl} - 1 \quad (i,j,k,l) \in \{1,\dots,n\}^4 \\ y_{ijkl} \geq 0 \quad (i,j,k,l) \in \{1,\dots,n\}^4 \end{cases}$$

Des reformulations pour le (QAP) dans un espace étendu

$$f_{\lambda,\delta,\delta',\phi}(x,y) = f_{\lambda,\delta,\delta'}(x) + \sum_{(i,j,k,l)} \phi_{ijkl}(x_{ij}x_{kl} - y_{ijkl}) = f(x)$$

Nous cherchons les paramètres $(\lambda, \delta, \delta', \phi)$ tels que :

i. $Q_{\lambda,\delta,\delta',\phi}$ est semidefinie positive,

ii. La valeur de la relaxation continue du $(QAP_{\lambda,\delta,\delta',\phi})$ soit la plus grande possible.

Cela revient à résoudre le problème suivant (CP) :

$$(CP')\begin{cases} \max\limits_{Q_{\lambda,\delta,\delta',\phi} \succeq 0} \{v(\overline{QAP}_{\lambda,\delta,\delta',\phi})\} \end{cases}$$

où $(\overline{QAP}_{\lambda,\delta,\delta',\phi})$ est la relaxation continue de $(QAP_{\lambda,\delta,\delta',\phi})$

Des reformulations pour le (QAP) dans un espace étendu

Théorème : $(\lambda^*, \delta^*, \delta'^*, \phi^*)$ se déduisent des variables duales de (SDP_{R4}).

$$(CP') \left\{ \max_{Q_{\lambda,\delta,\delta',\phi} \succeq 0} \{v(\overline{QAP}_{\lambda,\delta,\delta',\phi})\} \right. \Leftrightarrow \quad (SDP_{R4}) \begin{cases} \min f(X,x) = \langle Q, X \rangle \\ \text{s.t.} \sum_{i=1}^{n} x_{ij} = 1 \\ \sum_{j=1}^{n} x_{ij} = 1 \\ X_{ijij} - x_{ij} = 0 \quad \leftarrow \lambda_{ij} \\ X_{ijil} = 0 \leftarrow \delta_{ijl} \\ X_{ijkj} = 0 \leftarrow \delta'_{ijk} \\ -X_{ijkl} \leq 0 \leftarrow \phi_{ijkl} \\ \begin{pmatrix} 1 & x \\ x^T & X \end{pmatrix} \succeq 0 \\ x \in \mathbb{R}^{n^2} \quad X \in \mathcal{S}_{n^2} \end{cases}$$

Théorème : on a $v(CP') = v(SDP_{R4})$.

Comparaison des bornes pour le QAP

Instance	opt	Lin	RLT	VP	QCR'	R2	R4
nug06	86	0	86	-342.6	-1.1	81.3	85.3
nug12	578	0	522.89	-4168	-215.8	524.1	557.7
nug15	1150	0	1040.99	-10380.7	-823.1	1053.7	1122.3

Pour résumer

La classe d'algorithmes présentée est la suivante :

Algorithme basé sur une reformulation quadratique convexe :

1. Résoudre un des programme semi-défini (SDP_{QCR}), (SDP'_{QCR}), (SDP_{R2}), ou (SDP_{R4})
2. Déduire $(\lambda^*, \delta^*, \delta'^*, \phi^*)$ (ou les mettre à 0 si non considérés dans la relaxation SDP) et construire $(QAP_{\lambda^*,\delta^*,\delta'^*,\phi^*})$.
3. Résoudre le programme $(QAP_{\lambda^*,\delta^*,\delta'^*,\phi^*})$ par un solveur standard de programmes quadratique convexe.
   (La valeur de sa relaxation continue est égale à la valeur optimale de la relaxation SDP considérée à l'étape 1)