Reformulation Quadratique Convexe pour l'optimisation Quadratique discrète : résultats de base et extensions récentes

Sourour Elloumi
CEDRIC-ENSIIE

Conférence de la ROADEF- Tutoriel

Bordeaux 2014

Plan

I- Approches classiques en programmation quadratique 0-1

II- Reformulation Quadratique Convexe pour la programmation quadratique 0-1

III- Reformulation Quadratique Convexe pour la programmation quadratique en nombres entiers, contraintes quadratiques

I- Programmation quadratique 0-1

Approches classiques

Programmes quadratiques binaires

QP01

$$\boxed{\begin{aligned} &\min \quad f(x) = \sum_i c_i x_i + \sum_i \sum_{j \neq i} q_{ij} x_i x_j \\ &s.t.: \\ &\qquad Ax = b \\ &\qquad x \in \{0,1\}^n \end{aligned}} = c^T x + x^T Q x$$

Deux approches
« classiques » de résolution

1- Linéarisation

2- Programmation Semi-Définie Positive

1- Linéarisation

-Principe de la linéarisation « classique » :

remplacer $x_i x_j$ par y_{ij}

$$\boxed{\begin{cases} y_{ij} \le x_i \\ y_{ij} \le x_j \\ y_{ij} \ge x_i + x_j - 1 \\ y_{ij} \ge 0 \end{cases}}$$

1- Linéarisation

MLP01

$$\min \quad f(x) = \sum_i c_i x_i + \sum_i \sum_{j \neq i} q_{ij} y_{ij}$$

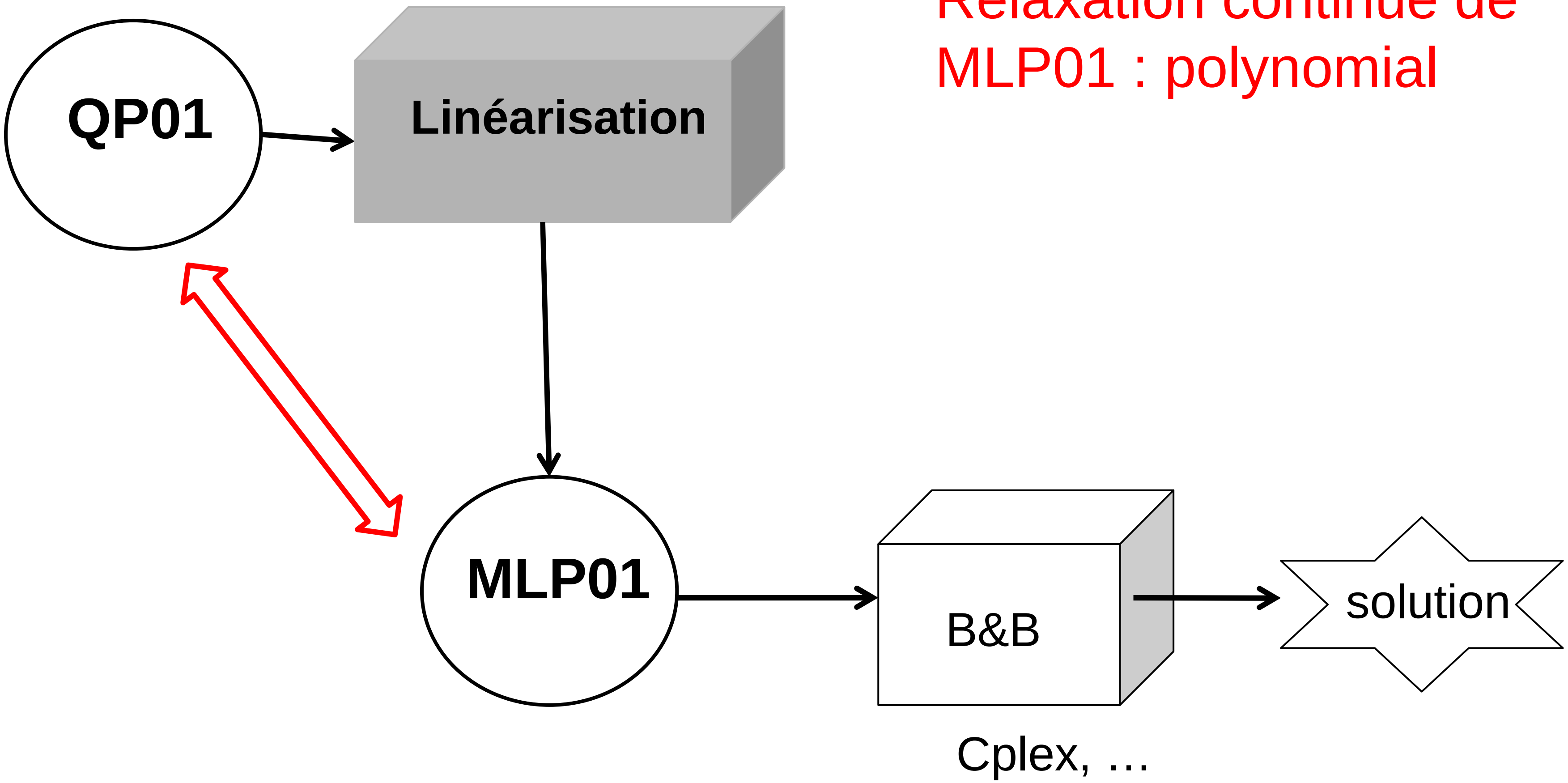
$$s.t.:$$

$$Ax = b$$

$$y_{ij} \leq x_i ; y_{ij} \leq x_j ; y_{ij} \geq x_i + x_j - 1; \; y_{ij} \geq 0$$

$$x \in \{0,1\}^n$$

1- Linéarisation (ou reformulation linéaire)



Relaxation continue de
MLP01 : polynomial

1- Linéarisation (ou reformulation linéaire)

Défaut : qualité de la borne par relaxation continue

(sauf si peu de termes quadratiques)

2- Programmation Semi-Définie Positive

-Principe :

remplacer xx^T ($x_i x_j$) par une matrice symétrique X (X_{ij})

$$\begin{aligned} \min \quad & c^T x + x^T Q x \\ s.t.: & \\ & Ax = b \\ & x \in \{0,1\}^n \end{aligned}$$

$$\begin{aligned} \min \quad & c^T x + < Q, xx^T > \\ s.t.: & \\ & Ax = b \\ & x \in \{0,1\}^n \end{aligned}$$

$$\begin{aligned} \min \quad & c^T x + < Q, X > \\ s.t.: & \\ & Ax = b \\ & X = xx^T \\ & x \in \{0,1\}^n \end{aligned}$$

2- Programmation Semi-Définie Positive

-Principe de la **relaxation** SDP :

relâcher $X = xx^T$ et $x \in \{0,1\}^n$ par :

$$X_{ii} = x_i$$
$$\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succ 0$$

(SDP1):

$$\begin{aligned}
&\min \quad c^T x + < Q, X > \\
&s.t.: \\
&\qquad Ax = b \\
&\qquad X_{ii} = x_i \\
&\qquad \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succ 0
\end{aligned}$$

Problème d'optimisation convexe : fournit une borne inférieure

2- Programmation Semi-Définie Positive

-La borne SDP peut être utilisée dans un algorithme B&B

-Très bonnes bornes

-Défaut : temps de calcul

2- Programmation Semi-Définie Positive

- 1ère amélioration

(SDP2):

$$\min \quad \sum_{i=1}^{n} c_i x_i + \sum_{i,j} q_{ij} X_{ij}$$

$$s.t.:$$

$$Ax = b$$

$$X_{ii} = x_i$$

$$\sum_{j=1}^{n} a_{kj} x_j - 2\sum_{j=1}^{n}\sum_{i=1}^{n} b_k a_{kj} a_{ki} X_{ij} = -b_k^2 \quad \forall k$$

$$\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succ 0$$

2- Programmation Semi-Définie Positive

- 2ème amélioration

(SDP3):

$$\min \quad \sum_{i=1}^{n} c_i x_i + \sum_{i,j} q_{ij} X_{ij}$$

$$s.t.:$$

$$Ax = b$$

$$X_{ii} = x_i$$

$$\sum_{j=1}^{n} a_{kj} x_j - 2\sum_{j=1}^{n}\sum_{i=1}^{n} b_k a_{kj} a_{ki} X_{ij} = -b_k^2 \quad \forall k$$

$$\boxed{X_{ij} \le x_i \,,\, X_{ij} \le x_j \,,\, X_{ij} \ge x_i + x_j - 1 \,,\, X_{ij} \ge 0}$$

$$\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succ 0$$

Exemple

$$\min f(x) = -9x_1 - 7x_2 + 2x_3 - 80x_4 + 12x_5 - 48x_1x_2 + 4x_1x_3 +$$
$$36x_1x_4 - 24x_1x_5 - 7x_2x_3 + 36x_2x_4 - 84x_2x_5 + 40x_3x_4 + 4x_3x_5 - 88x_4x_5$$

$s.t.$

$$x_1 + x_2 + 2x_4 + x_5 = 2 \quad x \text{ binaire}$$

$$c = \begin{pmatrix} -9 \\ -7 \\ 2 \\ 23 \\ 12 \end{pmatrix} \qquad Q = \begin{pmatrix} 0 & -24 & 2 & 18 & -12 \\ -24 & 0 & -3.5 & 18 & -42 \\ 2 & -3.5 & 0 & 20 & 2 \\ 18 & 18 & 20 & 0 & -44 \\ -12 & -42 & 2 & -44 & 0 \end{pmatrix}$$

Exemple : Linéarisation

$$\min f(x) = -9x_1 - 7x_2 + 2x_3 - 80x_4 + 12x_5 - 48y_{12} + 4y_{13} + 36y_{14} - 24y_{15} - 7y_{23} + 36y_{24} - 84y_{25} + 40y_{34} + 4y_{35} - 88y_{45}$$

$$s.t.$$

$$x_1 + x_2 + 2x_4 + x_5 = 2$$

$$\boxed{y_{ij} \geq 0 \quad y_{ij} \leq x_i \quad y_{ij} \leq x_j \quad y_{ij} \geq x_i + x_j - 1}$$

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

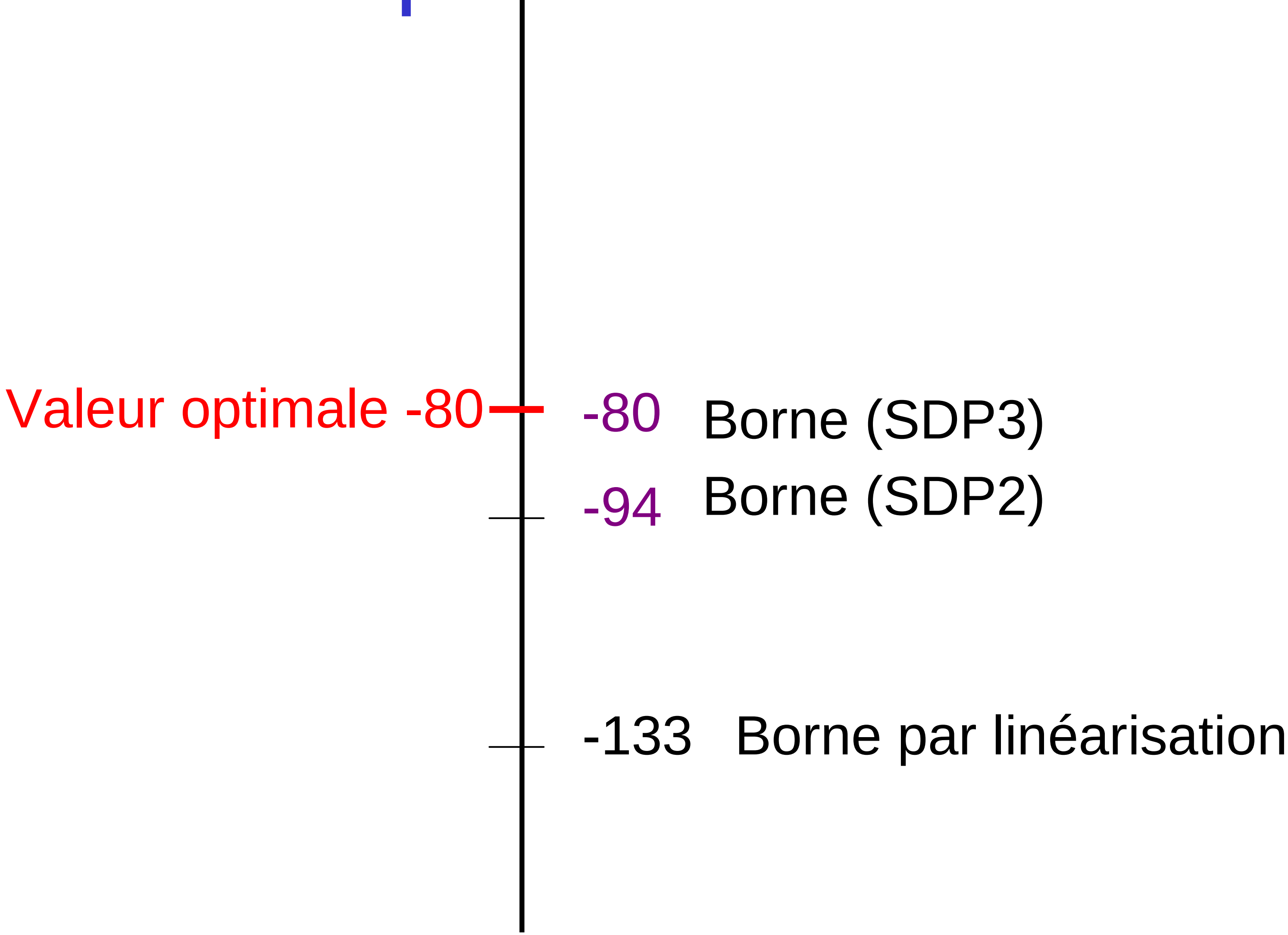
Exemple : Linéarisation

```
        Nodes                                               Cuts/
     Node  Left     Objective  IInf  Best Integer     Best Node    ItCnt     Gap

*      0+    0                          -65.0000                       0     ---
       0     0     -133.2000     5      -65.0000     -133.2000        13  104.92%
*      0+    0                          -80.0000     -133.2000        13   66.50%
       0     0        cutoff            -80.0000     -133.2000        15   66.50%

Times (seconds):
Input =  0.005
Solve =  0.004
Output = 0.001
CPLEX 11.2.0: optimal integer solution; objective -80
15 MIP simplex iterations
0 branch-and-bound nodes
```

Exemple : bornes SDP



Plan

I- Approches classiques en programmation quadratique 0-1

II- Reformulation Quadratique Convexe pour la programmation quadratique 0-1

III- Reformulation Quadratique Convexe pour la programmation quadratique en nombres entiers, contraintes quadratiques

Différence / approches classiques

S’appuie sur un autre classe de problèmes d’optimisation convexe :

la programmation quadratique continue convexe

Si objectif quadratique convexe : MI**L**P se « généralise » par MI**Q**P

Différence / approches classiques

S'appuie sur un autre classe de problèmes d'optimisation convexe :

la programmation quadratique continue convexe

QP01

$$\min \quad f(x) = \sum_i c_i x_i + \sum_i \sum_{j \neq i} q_{ij} x_i x_j \qquad = c^T x + x^T Q x$$

$$s.t.:$$

$$Ax = b$$

$$x \in \{0,1\}^n$$

si f convexe, (Q matrice SDP)

B&B (relaxation continue) – disponible dans les solveurs

Différence / approches classiques

S'appuie sur un autre classe de problèmes d'optimisation convexe :

la programmation quadratique convexe

QP01

$$\min \quad f(x) = \sum_i c_i x_i + \sum_i \sum_{j \neq i} q_{ij} x_i x_j \qquad = c^T x + x^T Q x$$

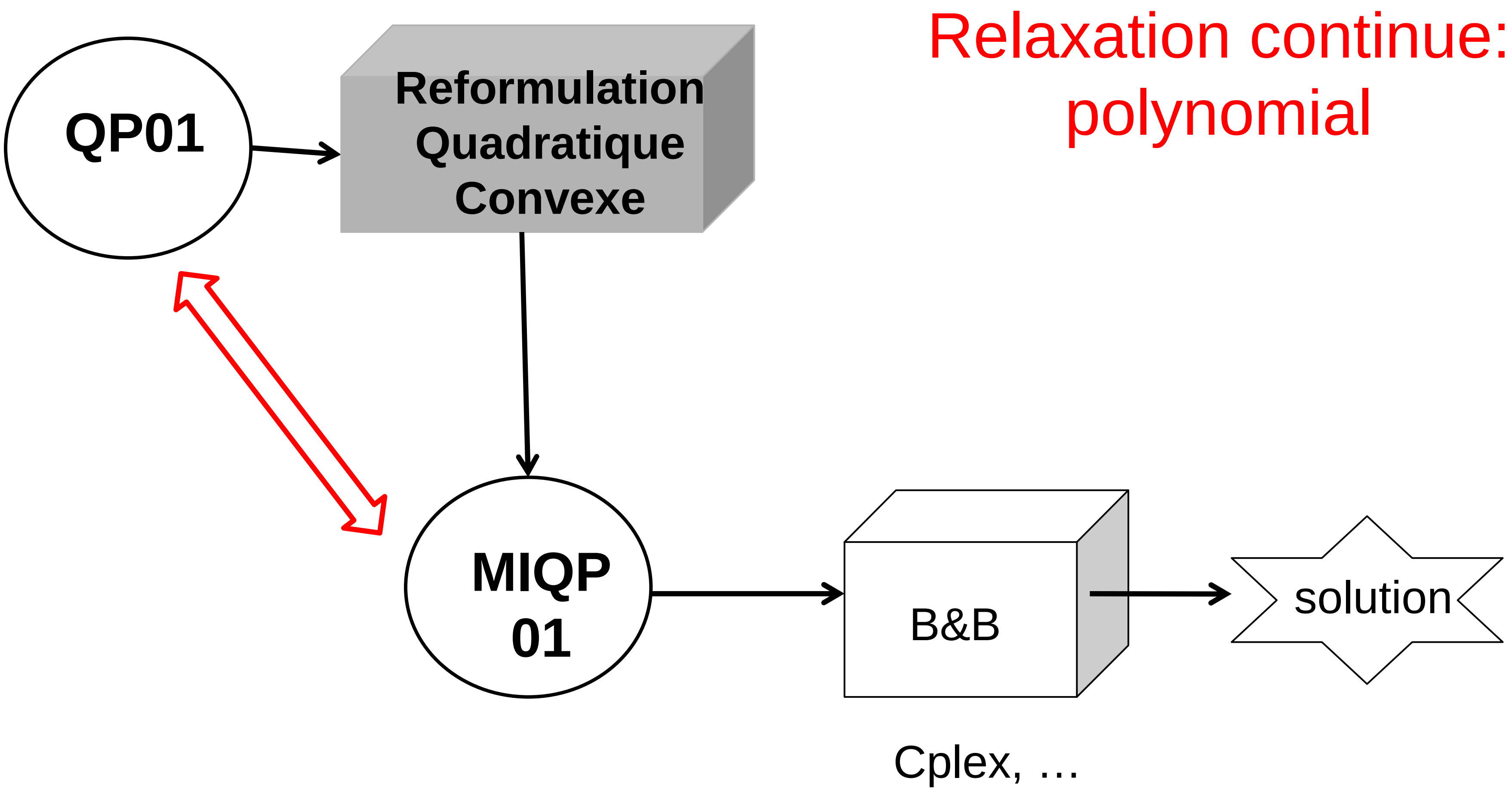
$$s.t.:$$

$$Ax = b$$

$$x \in \{0,1\}^n$$

Mais f n'est pas convexe en général

Reformulation Quadratique Convexe



Reformulation Quadratique Convexe

C'est toujours possible

Reformulation Quadratique Convexe « Naive »

Billionnet

$$q_{ij}x_i x_j = q_{ij}x_i x_j + \frac{1}{2}q_{ij}(x_i^2 - x_i) + \frac{1}{2}q_{ij}(x_j^2 - x_j)$$

$$= \frac{1}{2}q_{ij}\left[(x_i + x_j)^2 - x_i - x_j\right]$$

si $q_{ij} > 0$

Reformulation Quadratique Convexe « Naive »

$$\boxed{\begin{aligned} q_{ij}x_ix_j &= q_{ij}x_ix_j + \frac{1}{2}q_{ij}(x_i^2 - x_i) + \frac{1}{2}q_{ij}(x_j^2 - x_j) \\ &= \frac{1}{2}q_{ij}\left[(x_i + x_j)^2 - x_i - x_j\right] \end{aligned}} \qquad \text{si } q_{ij} > 0$$

$$\boxed{q_{ij}x_ix_j = \frac{1}{2}\left(-q_{ij}\right)\left[(x_i - x_j)^2 - x_i - x_j\right]} \qquad \text{si } q_{ij} < 0$$

Reformulation Quadratique Convexe « Naive »

$$\min f(x) = -9x_1 - 7x_2 + 2x_3 - 80x_4 + 12x_5 - 48x_1x_2 + 4x_1x_3 +$$
$$36x_1x_4 - 24x_1x_5 - 7x_2x_3 + 36x_2x_4 - 84x_2x_5 + 40x_3x_4 + 4x_3x_5 - 88x_4x_5$$

$s.t.$

$$x_1 + x_2 + 2x_4 + x_5 = 2 \qquad x \text{ binaire}$$

Problème reformulé :

$$\min f(x) = -9x_1 - 7x_2 + 2x_3 - 80x_4 + 12x_5 + (24(x_1 - x_2)^2 - x_1 - x_2) + 2...$$

$s.t.$

$$x_1 + x_2 + 2x_4 + x_5 = 2 \qquad x \text{ binaire}$$

Revient à :

$$c = \begin{pmatrix} -9 \\ -7 \\ 2 \\ -80 \\ 12 \end{pmatrix} \qquad Q = \begin{pmatrix} 0 & -24 & 2 & 18 & -12 \\ -24 & 0 & -3.5 & 18 & -42 \\ 2 & -3.5 & 0 & 20 & 2 \\ 18 & 18 & 20 & 0 & -44 \\ -12 & -42 & 2 & -44 & 0 \end{pmatrix}$$

$$c = \begin{pmatrix} -9 \\ -7 \\ 2 \\ -80 \\ 12 \end{pmatrix} \qquad Q = \begin{pmatrix} +24 & -24 & 2 & 18 & -12 \\ -24 & +24 & -3.5 & 18 & -42 \\ 2 & -3.5 & 0 & 20 & 2 \\ 18 & 18 & 20 & 0 & -44 \\ -12 & -42 & 2 & -44 & 0 \end{pmatrix}$$

Revient à :

$$c = \begin{pmatrix} -9 \\ -7 \\ 2 \\ -80 \\ 12 \end{pmatrix} \qquad Q = \begin{pmatrix} 0 & -24 & 2 & 18 & -12 \\ -24 & 0 & -3.5 & 18 & -42 \\ 2 & -3.5 & 0 & 20 & 2 \\ 18 & 18 & 20 & 0 & -44 \\ -12 & -42 & 2 & -44 & 0 \end{pmatrix}$$

$$c = \begin{pmatrix} -9-24 \\ -7-24 \\ 2 \\ -80 \\ 12 \end{pmatrix} \qquad Q = \begin{pmatrix} +24 & -24 & 2 & 18 & -12 \\ -24 & +24 & -3.5 & 18 & -42 \\ 2 & -3.5 & 0 & 20 & 2 \\ 18 & 18 & 20 & 0 & -44 \\ -12 & -42 & 2 & -44 & 0 \end{pmatrix}$$

Reformulation Quadratique Convexe « Naive »

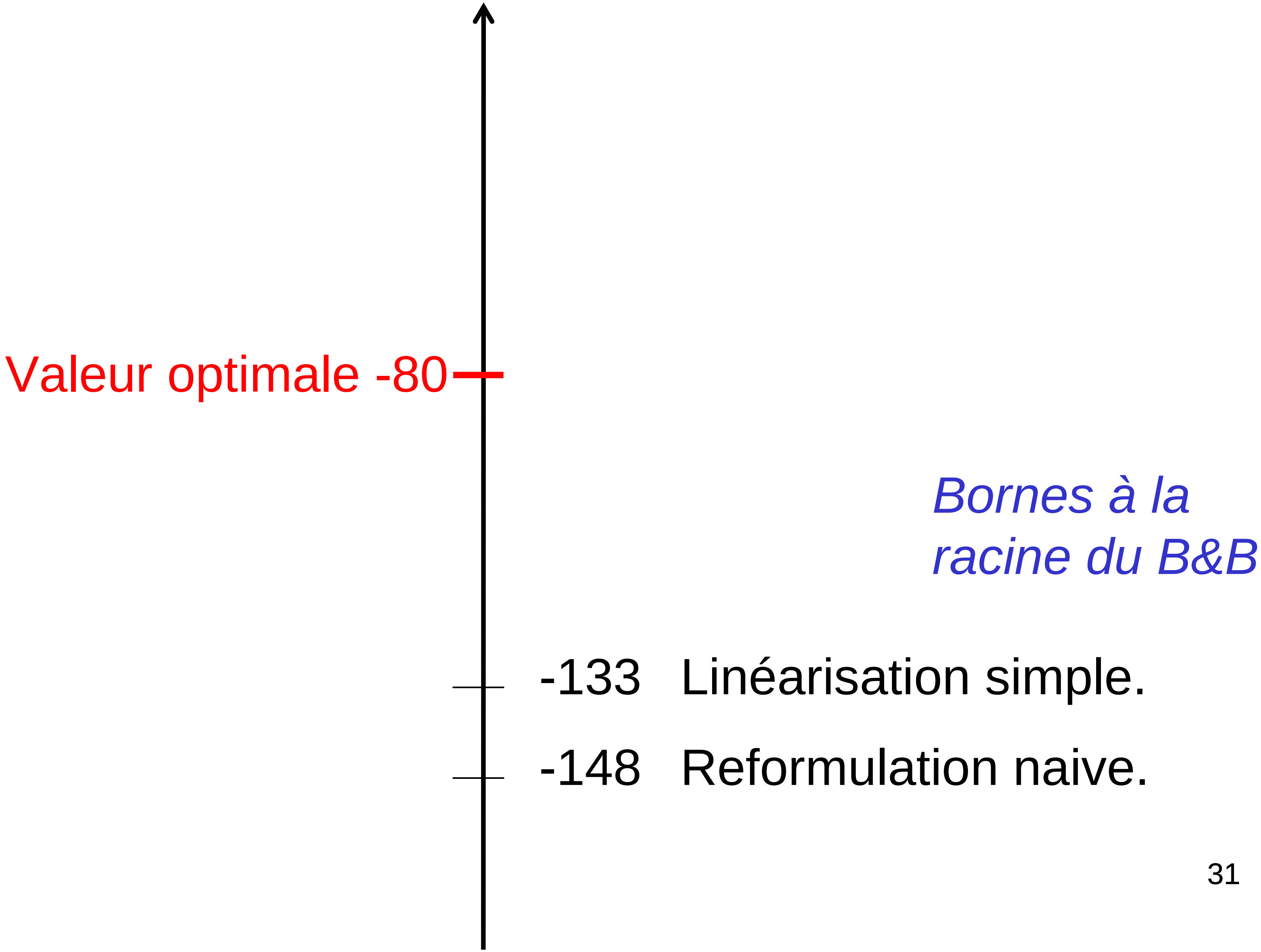
```
 Nodes                                              Cuts/
   Node  Left     Objective  IInf  Best Integer     Best Node    ItCnt     Gap

*     0+    0                          -79.0000                      0     ---
      0     0     -148.4014     5      -79.0000     -148.4014        7   87.85%
*     0+    0                          -80.0000     -148.4014        7   85.50%
      0     1     -133.7123     4      -80.0000     -107.8295        8   34.79%
      1     1     -107.8295     3      -80.0000      -86.3841        9    7.98%
      2     0      -86.3841     1      -80.0000      -80.0000       10    0.00%

Times (seconds):
Input =  0.006
Solve =  0.006
Output = 0
CPLEX 11.2.0: optimal integer solution; objective -80
10 MIP simplex iterations
3 branch-and-bound nodes
```

Chaque nœud : un programme quadratique convexe

Reformulation Quadratique Convexe « Naive »



Reformulation Quadratique Convexe : Plus Petite Valeur Propre

Hammer, Rubin

$$\boxed{\sum_i \sum_j q_{ij} x_i x_j = \sum_i \sum_j q_{ij} x_i x_j - \lambda_{\min} \sum_i \left(x_i^2 - x_i \right)}$$

$\boxed{\lambda_{\min}}$: plus petite valeur propre de Q (<0)

$$c_\lambda = \begin{pmatrix} -9 + \lambda_{\min} \\ -7 + \lambda_{\min} \\ 2 + \lambda_{\min} \\ -80 + \lambda_{\min} \\ 12 + \lambda_{\min} \end{pmatrix} \quad Q_\lambda = \begin{pmatrix} -\lambda_{\min} & -24 & 2 & 18 & -12 \\ -24 & -\lambda_{\min} & -3.5 & 18 & -42 \\ 2 & -3.5 & -\lambda_{\min} & 20 & 2 \\ 18 & 18 & 20 & -\lambda_{\min} & -44 \\ -12 & -42 & 2 & -44 & -\lambda_{\min} \end{pmatrix}$$

Reformulation Quadratique Convexe : Plus Petite Valeur Propre

$$\boxed{\lambda_{\min} = -56.88}$$

Optimal value -80

Bornes à la racine du B&B

-127 Reformulation PPVP

-133 Linéarisation simple.

-148 Reformulation naive.

- Comment faire mieux ?

$\Leftrightarrow$ Trouver une Reformulation quadratique convexe telle que **la borne à la racine du B&B soit maximale**

Amélioration 1

Billionnet, E. 2007

- Généraliser la perturbation de la diagonale

$$c_u = \begin{pmatrix} -9-u_1 \\ -7-u_2 \\ 2-u_3 \\ -80-u_4 \\ 12-u_5 \end{pmatrix} \qquad Q_u = \begin{pmatrix} u_1 & -24 & 2 & 18 & -12 \\ -24 & u_2 & -3.5 & 18 & -42 \\ 2 & -3.5 & u_3 & 20 & 2 \\ 18 & 18 & 20 & u_4 & -44 \\ -12 & -42 & 2 & -44 & u_5 \end{pmatrix}$$

$$\boxed{\begin{aligned} f_u(x) &= f(x) + \sum_{i=1}^{n} u_i\left(x_i^2 - x_i\right) \\ &= f(x) \quad \forall x \text{ binaire} \end{aligned}}$$

Amélioration 1

- Trouver un vecteur *u* qui maximise la borne à la racine du B&B

MIQP01$_u$

$$
\begin{aligned}
\min \quad & f_u(x) \\
s.t.: \quad & \\
& Ax = b \\
& x \in \{0,1\}^n
\end{aligned}
$$

P

$$
\begin{aligned}
\max_{u: f_u \text{ convexe}} \quad \min \quad & f_u(x) \\
& Ax = b \\
& x \in [0,1]^n
\end{aligned}
$$

Amélioration 1

- **Théorème** : opt(**P**)=opt(SDP1) et
  une solution optimale de **P** peut être obtenue à partir du dual de

(SDP1):
$$\begin{array}{ll} \min & c^T x + \langle Q, X \rangle \\ s.t.: & \\ & Ax = b \\ & X_{ii} = x_i \qquad \text{variables duales } u_i \\ & \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succ 0 \end{array}$$

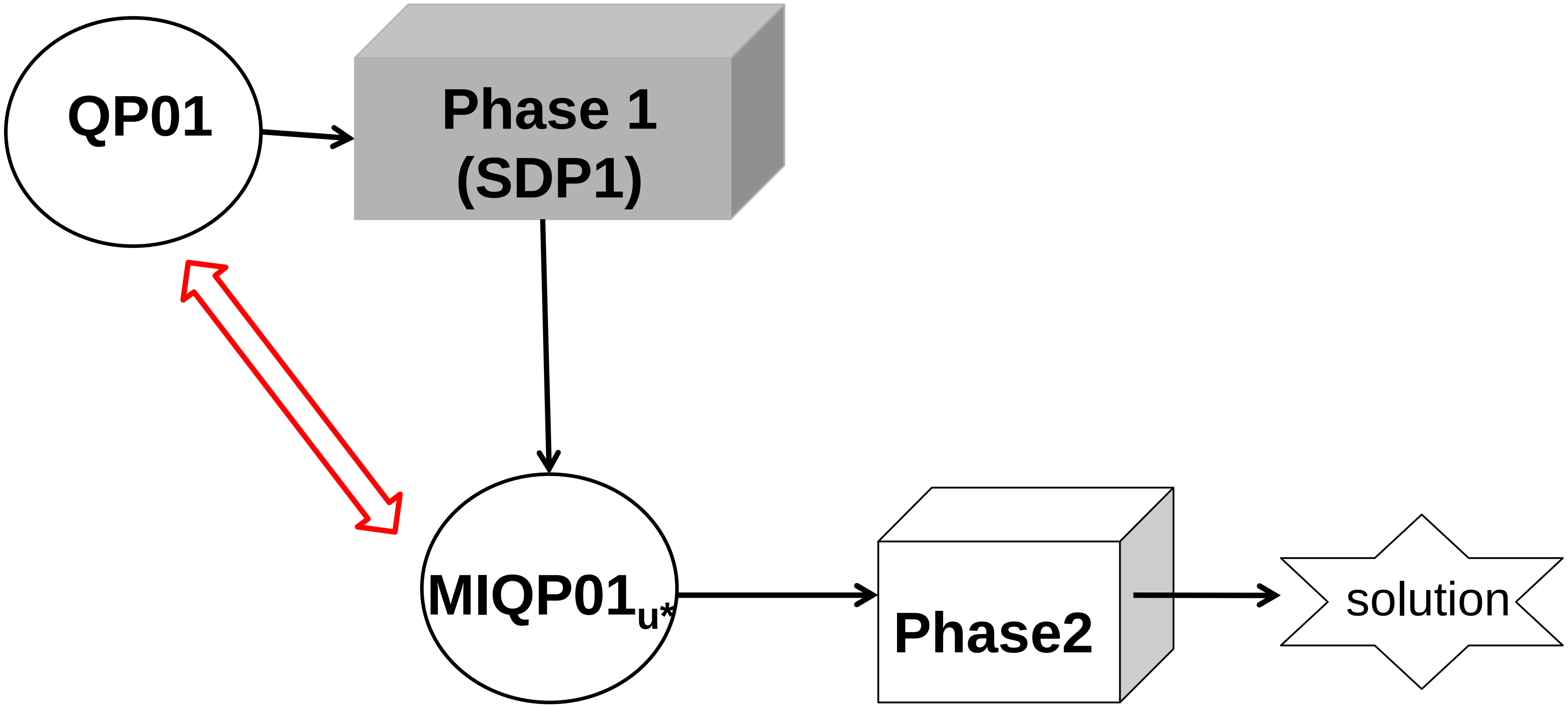
Amélioration 1

Résolution exacte par Reformulation Quadratique Convexe:

Phase 1 : Résolution de (SDP1) → vecteur u^*

Phase 2 : Résolution de **MIQP01**$_{u^*}$ par B&B

Borne à la racine du B&B : opt(SDP1)


QP01
Phase 1
(SDP1)
MIQP01u*
Phase2
solution

Premiers résultats

Programmation quadratique binaire sans contraintes:

100 variables, coefficients dans [-100, 100]

Linéarisation erreur à la racine : 266% non résolu (1h)

PPVP erreur à la racine : 15 % non résolu (1h)

QCR erreur à la racine : 7%   6 min en moyenne (incl. SDP)

Amélioration 2

Billionnet, E., Plateau, 2009

- Utiliser les contraintes d'égalité

QP01

$$\min \quad f(x) = \sum_i c_i x_i + \sum_i \sum_{j \neq i} q_{ij} x_i x_j$$

$$s.t.:$$

$$Ax = b$$

$$x \in \{0,1\}^n$$

Amélioration 2 (QCR)

Billionnet, E., Plateau, 2009

- Utiliser les contraintes d'égalité

$$\boxed{f_{u,\alpha}(x) = f(x) + \sum_{i=1}^{n} u_i\left(x_i^2 - x_i\right) + \sum_k \alpha_k \left(\sum_{j=1}^{n} a_{kj} x_j - b_k\right)^2}$$

$$= -\alpha b^T b + c_{u,\alpha}^T x + x^T Q_{u,\alpha} x$$

$$\boxed{f_{u,\alpha}(x) = f(x) \quad \forall x \text{ solution admissible}}$$

Même théorème et algorithme de résolution avec :

(SDP2):

$$
\begin{aligned}
&\min \quad \sum_{i=1}^{n} c_i x_i + \sum_{i,j} q_{ij} X_{ij} \\
&s.t.: \\
&\qquad X_{ii} = x_i \qquad \forall i \qquad (u_i) \\
&\qquad Ax = b \\
&\qquad \sum_{j=1}^{n} a_{kj} x_j - 2 \sum_{j=1}^{n} \sum_{i=1}^{n} b_k a_{kj} a_{ki} X_{ij} = -b_k^2 \quad \forall k \qquad (\alpha_k) \\
&\qquad \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succ 0
\end{aligned}
$$

var. duales

Quelques résultats numériques

k-cluster ou k-sous-graphe le plus dense

45 instances de 80 sommets

PPVP erreur à la racine : 83% aucune instance résolue en 1h

QCR erreur à la racine : 3.5% toutes les instances sont résolues, en moyenne 40 sec. (0.2 sec. pour SDP)

Amélioation 3 (MIQCR)

Billionnet, E., Lambert, 2012

- On veut changer potentiellement tous les coefficients de Q

$$f_{u,\alpha,\beta}(x,y) = f(x) + \sum_{i=1}^{n} u_i\left(x_i^2 - x_i\right) + \sum_k \alpha_k \left(\sum_{j=1}^{n} a_{kj}x_j - b_k\right)^2 + \sum_{i,j} \Phi_{ij}\left(x_i x_j - y_{ij}\right)$$

$$= -\alpha b^T b + c_{u,\alpha,\Phi}^T x + x^T Q_{u,\alpha,\Phi} x - \Phi y$$

$$\boxed{f_{u,\alpha,\Phi}(x,y) = f(x) \quad \forall x \text{ solution admissible et } y_{ij} = x_i x_j}$$

- On veut : $Q_{u,\alpha,\Phi}$ SDP

QP01
(Reform.)

$$\left\{\begin{array}{l} \min \quad f_{u,\alpha,\Phi}(x,y) \\ \text{s.t.}: \quad Ax = b \\ y_{ij} \le x_i \\ y_{ij} \le x_j \\ y_{ij} \ge x_i + x_j - 1 \\ y_{ij} \ge 0 \\ x_i \in \{0,1\} \end{array}\right.$$

$\Rightarrow$ $\boxed{y_{ij} = x_i x_j}$

$\boxed{\text{pour } \Phi_{ij} \neq 0}$

- Et on veut: maximiser la borne par relaxation continue

$$(P): \quad \max_{u,\alpha,\Phi:\, Q_{u,\alpha,\Phi} \succ 0} \begin{cases} \min & f_{u,\alpha,\Phi}(x,y) \\ \text{s.t.}: & Ax = b \\ & y_{ij} \le x_i \\ & y_{ij} \le x_j \\ & y_{ij} \ge x_i + x_j - 1 \\ & y_{ij} \ge 0 \\ & 0 \le x_i \le 1 \end{cases}$$

Même théorème et algorithme de résolution avec :

(SDP3):

$$\min \quad \sum_{i=1}^{n} c_i x_i + \sum_{i,j} q_{ij} X_{ij}$$

dual var

$$s.t.:$$

$$X_{ii} = x_i \qquad \forall i \qquad (u_i)$$

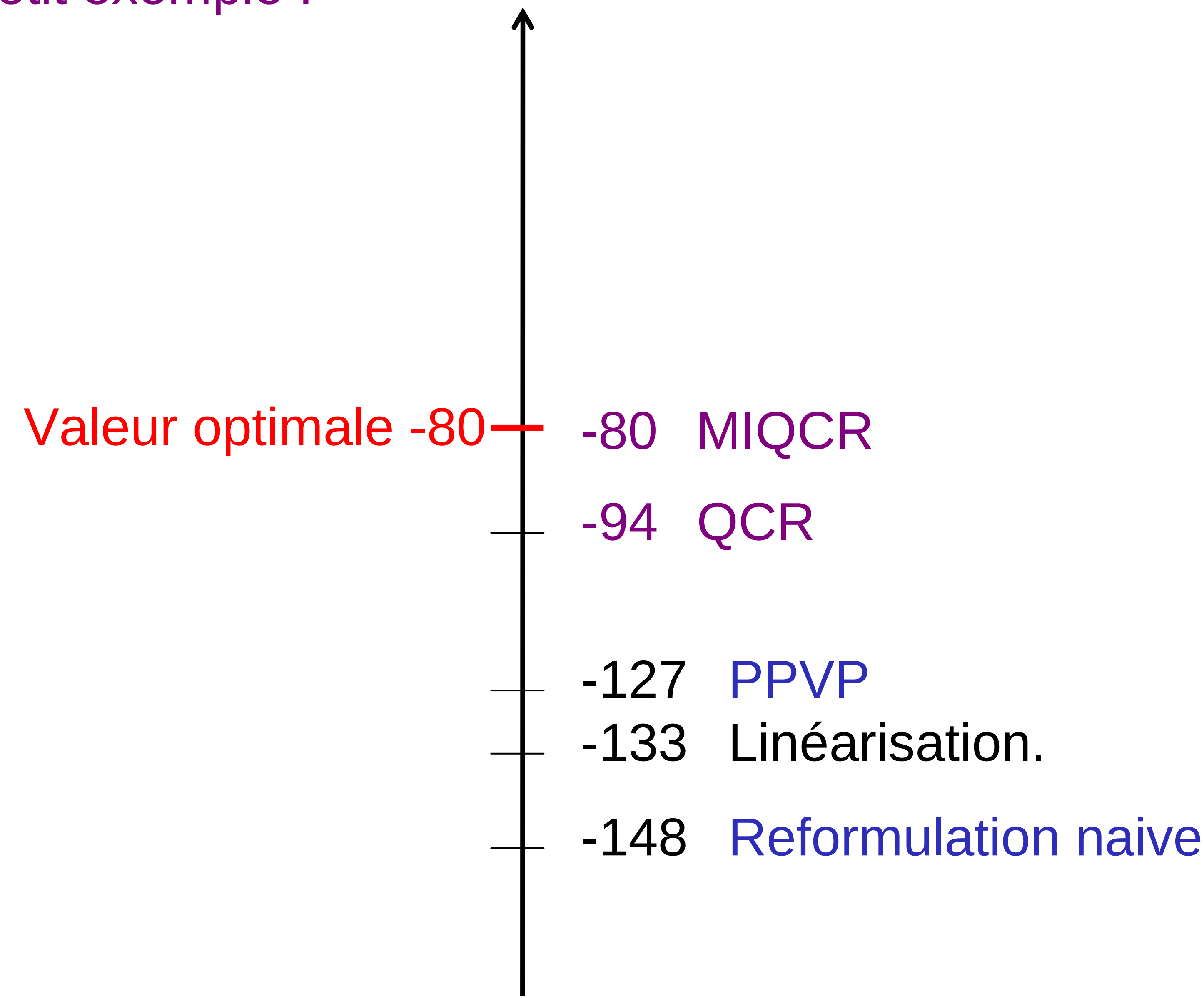
$$Ax = b$$

$$\sum_{j=1}^{n} a_{kj} x_j - 2\sum_{j=1}^{n}\sum_{i=1}^{n} b_k a_{kj} a_{ki} X_{ij} = -b_k^2 \quad \forall k \qquad (\alpha_k) \quad \text{(QCR)}$$

$$X_{ij} \le x_i \,,\, X_{ij} \le x_j \,,\, X_{ij} \ge x_i + x_j - 1 \,,\, X_{ij} \ge 0 \qquad (\Phi_{ij}) \quad \text{(MIQCR)}$$

$$\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succ 0$$

Petit exemple :



Résultats numériques

k-cluster ou k-sous-graphe le plus dense

45 instances de 80 sommets

QCR erreur à la racine 3.5%   toutes les instances sont résolues, en moyenne 40 sec. (0.2 pour SDP)
600 000 noeuds

MIQCR erreur à la racine : 1%   630 sec.
(450 pour SDP- 180 pour B&B)
6 000 noeuds

Résultats numériques

k-cluster ou k-sous-graphe le plus dense

Défaut de MIQCR : temps de calcul trop important pour (SDP3)

Méthode de faisceaux appliquée à la SDP : Conic Bundle Library (Helmberg) et CSDP

-> MIQCR-CB

Billionnet, E., Lambert, Wiegele (soumis)

Résultats numériques

k-cluster ou k-sous-graphe le plus dense

QCR erreur à la racine 3.5% toutes les instances sont résolues, en moyenne 40 sec. (0.2 pour SDP)

MIQCR erreur à la racine : 1% 630 sec.
(450 pour SDP- 180 pour B&B)

MIQCR-CB erreur à la racine : 1% 10 sec.
(6 pour SDP- 4 pour B&B)

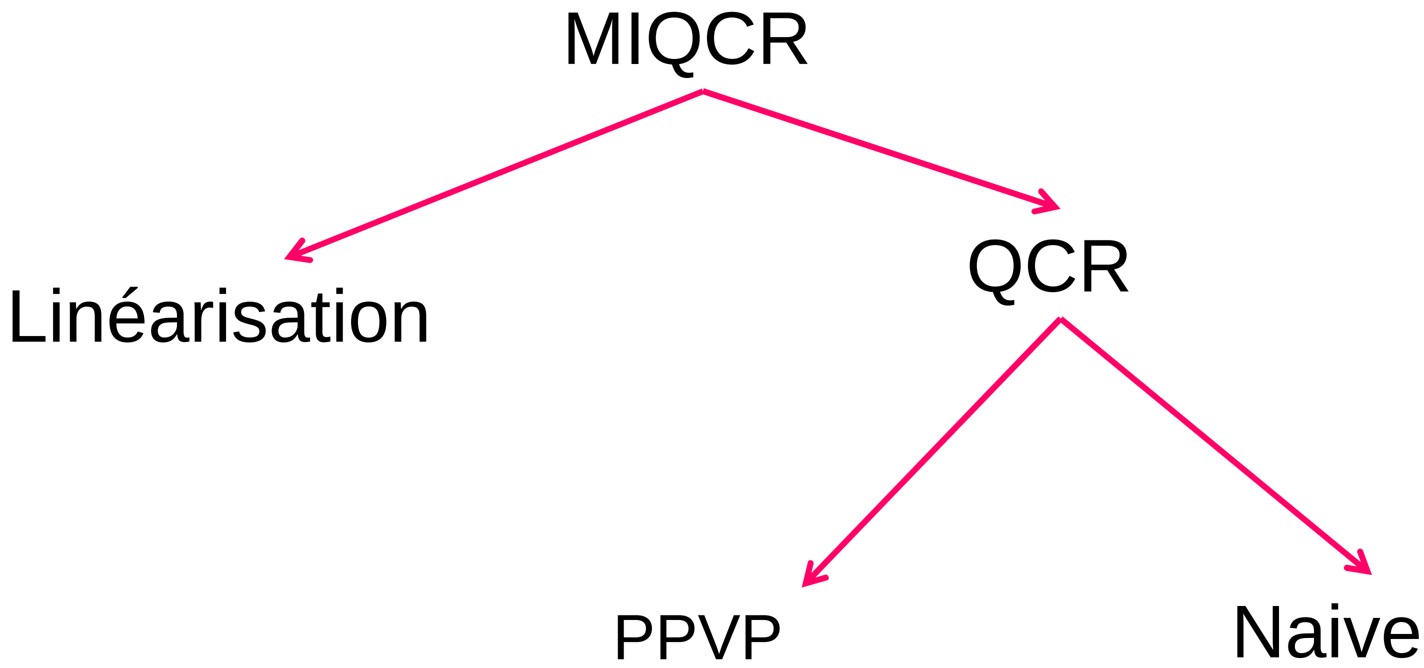
MIQCR-CB nous a permis de résoudre des instances de 160 sommets

Pour conclure cette partie :

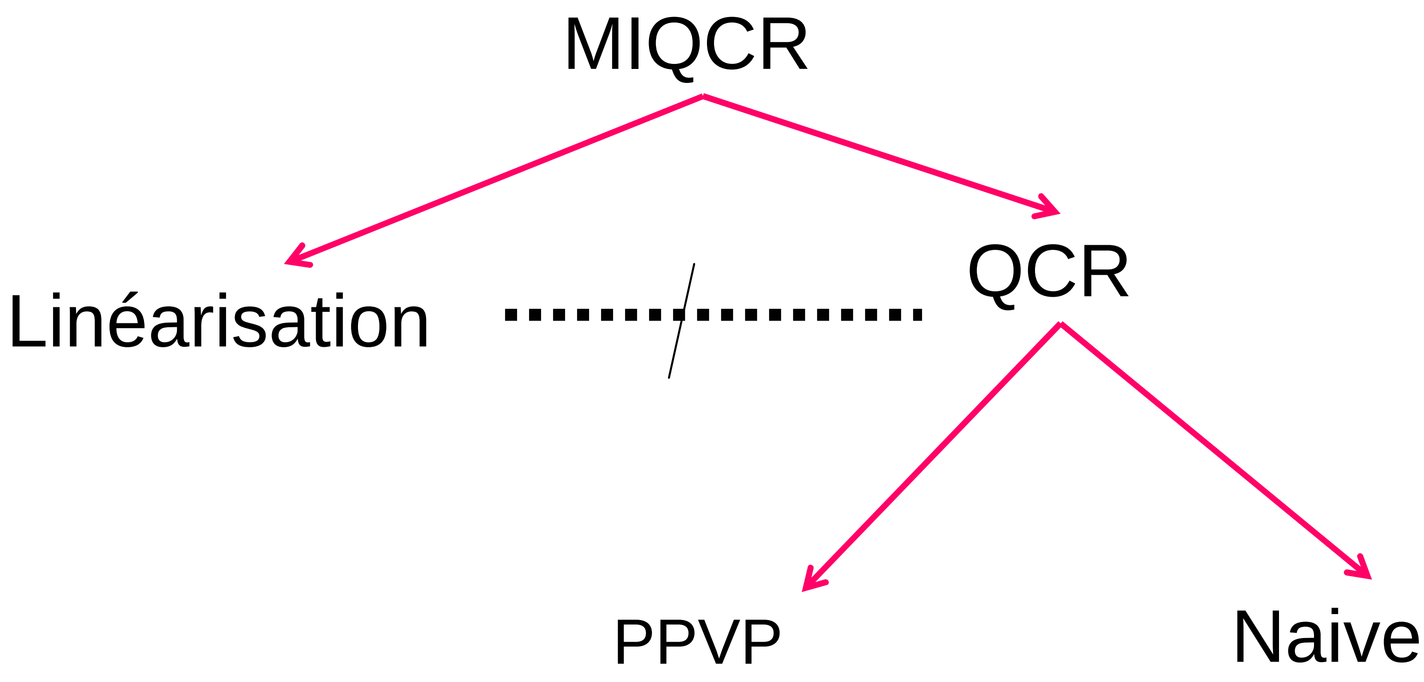
5 méthodes **générales** de résolution exacte des programmes quadratiques en variables 0-1 :

Linéarisation, Naive, PPVP, QCR et MIQCR

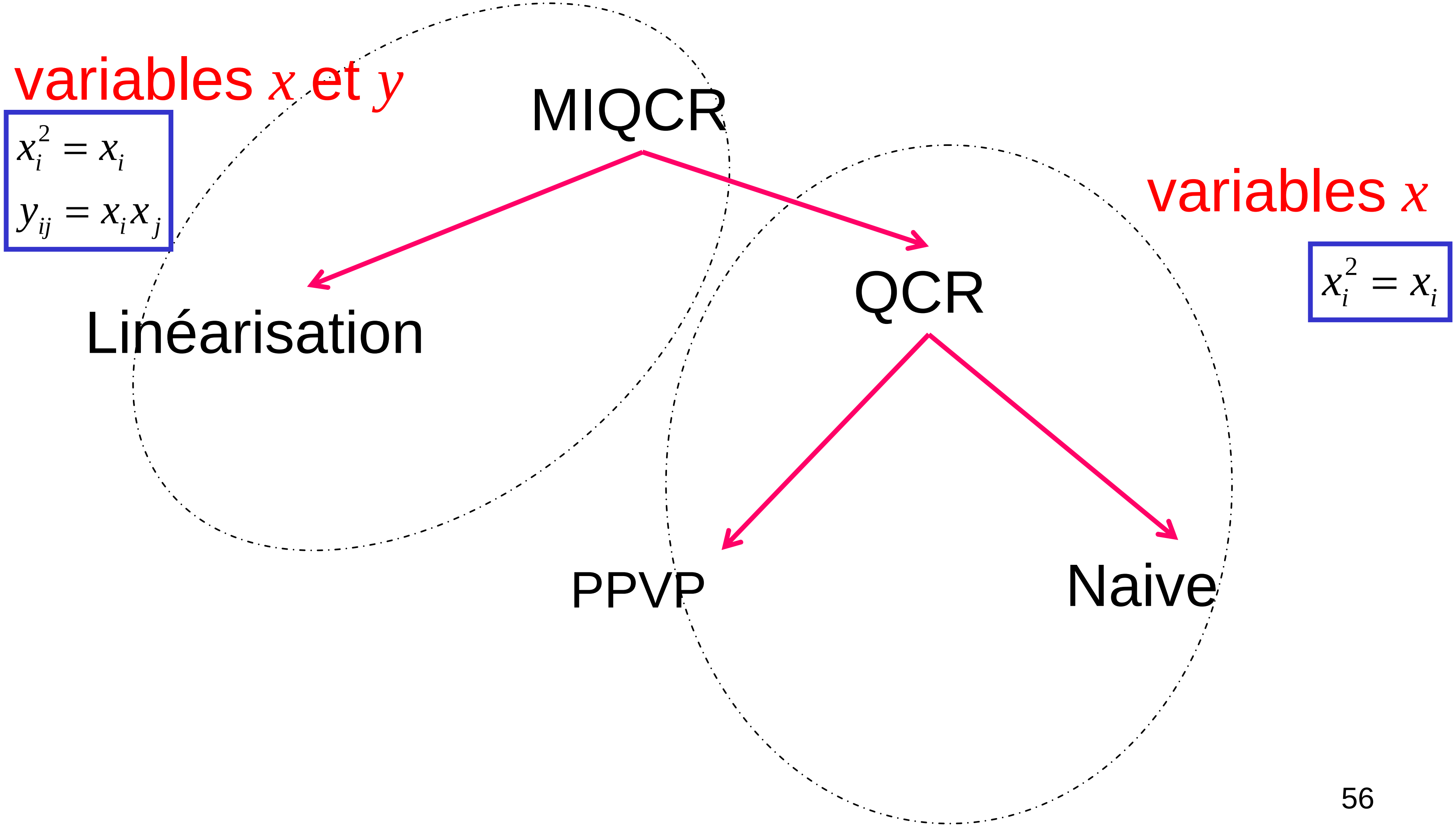
Pour conclure cette partie : hiérarchie des bornes à la racine du B&B



Pour conclure cette partie : hiérarchie des bornes à la racine du B&B



Pour conclure cette partie : hiérarchie des bornes à la racine du B&B



Plan

I- Approches classiques en programmation quadratique 0-1

II- Reformulation Quadratique Convexe pour la programmation quadratique 0-1

III- Reformulation Quadratique Convexe pour la programmation quadratique en nombres entiers, contraintes quadratiques

Billionnet, E., Lambert 2012

Billionnet, E., Lambert (soumis)

Extension aux cas des variables entières et des contraintes quadratiques (inégalités)

Variables entières

Variables entières bornées $0 \leq x_i \leq u_i$

Ici $x_i^2 \neq x_i$

Pas d'expression directe de $y_{ij} = x_i x_j$ par des contraintes linéaires

Variables entières : Linéarisation de $y_{ij} = x_i x_j$

$$\boxed{x_i = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k t_{ik}}$$

Variables binaires

$$\boxed{y_{ij} = x_i x_j = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k \boxed{t_{ik} x_j}}$$

Linéariser par de nouvelles variables et contraintes

$$\boxed{z_{ijk} = t_{ik} x_j}$$

$$\boxed{z_{ijk} \leq x_j \,, z_{ijk} \leq u_j t_{ik} \,, z_{ijk} \geq x_j + u_j t_{ik} - u_j \,, z_{ijk} \geq 0}$$

Variables entières : Linéarisation de $y_{ij} = x_i x_j$

On obtient:

$$x_i = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k t_{ik}$$

$$y_{ij} = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k z_{ijk}$$

$$z_{ijk} \leq x_j \,, z_{ijk} \leq u_j t_{ik} \,, z_{ijk} \geq x_j + u_j t_{ik} - u_j \,, z_{ijk} \geq 0$$

$$t_{ik} \in \{0,1\}$$

C'est une linéarisation

Variables entières : Linéarisation de $y_{ij} = x_i x_j$

On ajoute des inégalités valides :

$$\boxed{\begin{array}{l} y_{ij} = y_{ji} \\ y_{ij} \geq u_j x_i + u_i x_j - u_i u_j \\ y_{ij} \leq u_j x_i \\ y_{ii} \geq x_i \\ y_{ij} \geq 0 \end{array}}$$

On obtient:

ensemble P_{xyzt}

Contraintes linéaires et variables binaires t

Variables entières : Linéarisation de $y_{ij} = x_i x_j$

Propriété

Projection de $\overline{P_{xyzt}}$ sur (x,y) est

$$\overline{P_{xy}} \quad \boxed{\begin{array}{l} y_{ij} = y_{ji} \\ y_{ij} \geq u_j x_i + u_i x_j - u_i u_j \\ y_{ij} \leq u_j x_i \\ y_{ii} \geq x_i \\ y_{ij} \geq 0 \end{array}}$$

Variables entières : Linéarisation de $y_{ij} = x_i x_j$

Ce qu'il faut retenir :

- C'est possible
- Projection sur (x,y) facile

Relaxation quadratique convexe variables entières et contraintes quadratiques

QCQP

$$\boxed{\begin{array}{ll} \min & f(x) = x^T Q_0 x + c_0^T x \\ s.t.: & \\ & c_k^T x + x^T Q_k x \le b_k \qquad k = 1,\dots,K \\ & 0 \le x \le u \\ & x \ \text{entier} \end{array}}$$

Relaxation quadratique convexe - variables entières et contraintes quadratiques

Idée :

$$
\begin{aligned}
f(x) &= x^T Q_0 x + c_0^T x \\
&= x^T S_0 x + c_0^T x + x^T \left(Q_0 - S_0\right) x \\
&= x^T S_0 x + c_0^T x + \left\langle Q_0 - S_0, xx^T \right\rangle \\
&= x^T S_0 x + c_0^T x + \left\langle Q_0 - S_0, y \right\rangle
\end{aligned}
$$

Si S_0 matrice SDP alors fonction convexe

Même idée mais formalisme plus général que dans le cas des variables 0-1

Relaxation quadratique convexe - variables entières et contraintes quadratiques

-Ajout variables y t.q. $y_{ij}=x_i x_j$ (ou $y=xx^T$)
-Soit $S_0, S_1, \ldots, S_K$ SDP quelconques

$$\boxed{\begin{array}{ll} \min & f(x,y) = x^T S_0 x + c_0^T x + \langle Q_0 - S_0, y \rangle \\ s.t.: & \\ & x^T S_k x + c_k^T x + \langle Q_k - S_k, y \rangle \le b_k \qquad k = 1,\ldots,K \\ & 0 \le x \le u \\ & x \text{ entier} \\ & y = xx^T \end{array}}$$

« à linéariser »

Relaxation quadratique convexe - variables entières et contraintes quadratiques

Pour $S_0, S_1, \ldots, S_K$ SDP quelconques :

$$
\begin{array}{ll}
\min & f(x,y) = x^T S_0 x + c_0^T x + < Q_0 - S_0, y > \\
s.t.: & \\
& x^T S_k x + c_k^T x + < Q_k - S_k, y > \le b_k \qquad k = 1,\ldots,K \\
& 0 \le x \le u \\
& x \text{ entier} \\
& x, y, z, t \in P_{xyzt}
\end{array}
$$

Problème équivalent

Peut être résolu par les solveurs standard

Relaxation quadratique convexe - variables entières et contraintes quadratiques

Matrices SDP particulières :

- Linéarisation complète : tous S_k=0

-PPVP : $\boxed{S_k = Q_k - diag(\lambda_{\min}(Q_k))}$

- Déjà convexe : S_k=Q_k

Relaxation quadratique convexe - variables entières et contraintes quadratiques

Problème

Trouver les « meilleures » matrices SDP

Qui donnent la plus grande valeur par relaxation continue

Relaxation quadratique convexe - variables entières et contraintes quadratiques

Problème

$$\max_{S_0, S_1, \ldots, S_K \text{ SDP}}$$

$$\boxed{\begin{array}{ll} \min & f(x,y) = x^T S_0 x + c_0^T x + < Q_0 - S_0, y > \\ s.t.: & \\ & x^T S_k x + c_k^T x + < Q_k - S_k, y > \leq b_k \qquad k = 1,\ldots,K \\ & 0 \leq x \leq u \\ & x \ \text{entier} \\ & x, y, z, t \in \overline{P}_{xyzt} \end{array}}$$

On peut remplacer par $\boxed{x, y \in \overline{P}_{xy}}$

Théorème

Même valeur optimale que :

(SDP)

$$
\begin{aligned}
&\min \quad c_0^T x + Q_0 X \\
&s.t.: \\
&c_k^T x + Q_k X \le b_k \\
&X_{ij} \le u_j x_i \;, X_{ij} \le u_i x_j \;, \\
&X_{ij} \ge u_j x_i + u_i x_j - u_i u_j \;, X_{ii} \ge x_i, \\
&X_{ij} \ge 0 \\
&\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succ 0
\end{aligned}
$$

Var duales

(α_k)

(Φ_{ij})

Relaxation SDP de (QCQP)

Théorème-

et déduire :

$$\boxed{S_0^* = Q_0 + \sum_k \alpha_k Q_k + \Phi}$$

$$\boxed{S_k^* = 0 \qquad \text{pour } k \geq 1}$$

comme matrices « optimales »

Relaxation quadratique convexe - variables entières et contraintes quadratiques

Ainsi, dans notre reformulation optimale :

Fonction objectif : quadratique

Les contraintes : linéaires

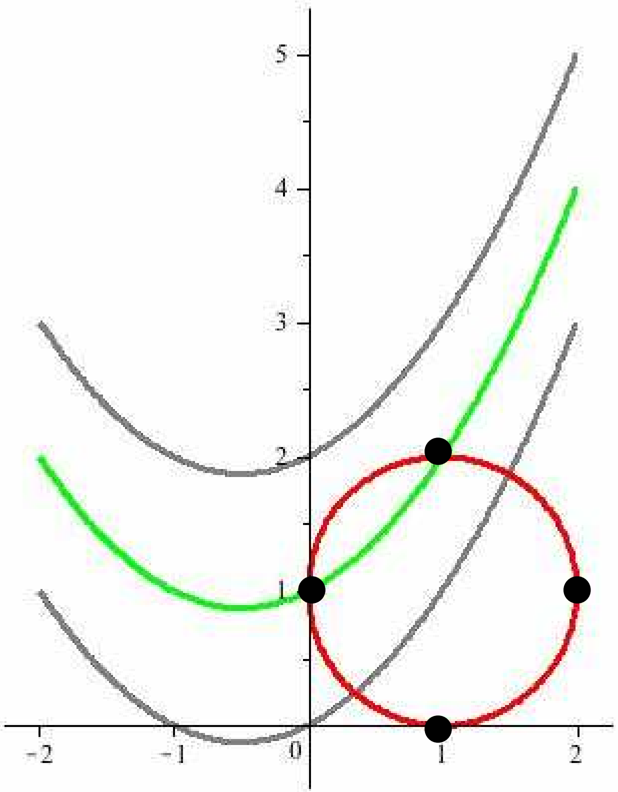
Exemple

$$\min \quad f(x) = x_1^2 + x_1 - 2x_2$$

$$s.t.:$$

$$x_1^2 + x_2^2 - 2x_1 - 2x_2 = -1$$

$$0 \leq x_i \leq 2 \quad \text{et entiers}$$



Valeur optimale : -2

Phase 1

Relaxation SDP

$$\min \quad X_{11} + x_1 - 2x_2$$

$$s.t.:$$

$$X_{11} + X_{22} - 2x_1 - 2x_2 = -1$$

$$X_{ij} \leq 2x_i$$

$$X_{ij} \geq 2x_i + 2x_j - 4 \,, X_{ii} \geq x_i,$$

$$X_{ij} \geq 0$$

$$\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succ 0$$

Valeur optimale : -2.5

$$S_0^* = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}$$

$$S_1^* = 0$$

Phase 2

Problème reformulé, résolu par Cplex

$$
\begin{aligned}
&\min \quad f(x) = 2x_2^2 + x_1 - 2x_2 + y_{11} - 2y_{22} \\
&s.t.: \\
&\qquad y_{11} + y_{22} - 2x_1 - 2x_2 = -1 \\
&\qquad 0 \le x_i \le 2 \quad \text{et entier} \\
&\qquad (x, y, z, t) \in P_{xyzt}
\end{aligned}
$$

Valeur optimale : -2

Valeur de la relax. continue : -2.5

Comparaison avec la linéarisation

$$\begin{aligned}
&\min \quad f(x) = x_1 - 2x_2 + y_{11} \\
&s.t.: \\
&\qquad y_{11} + y_{22} - 2x_1 - 2x_2 = -1 \\
&\qquad 0 \le x_i \le 2 \quad \text{et entier} \\
&\qquad (x, y, z, t) \in P_{xyzt}
\end{aligned}$$

Valeur optimale : -2

Valeur de la relax. continue : -3

Résultats numériques

- 1 objectif quadratique, 1 inégalité quadratique
- u=20

	(QP^*)				(LP)		
n	*Gap*	*Ref*	*Tot*	*Nodes*	*Gap*	*Tot*	*Nodes*
10	5.51	3.30	4.10	122.80	47.47	0.70	161.60
20	2.98	31.70	37.80	391.30	88.68	166.10	7410.10
30	3.21	144.90	179.50	877.90	124.09	(51.27%)	40269.30
40	5.01	165.3	543.25 (8)	18912.20	166.25	(122.47%)	12772.50

Autres résultats extensions

- Extension au cas des variables mixtes

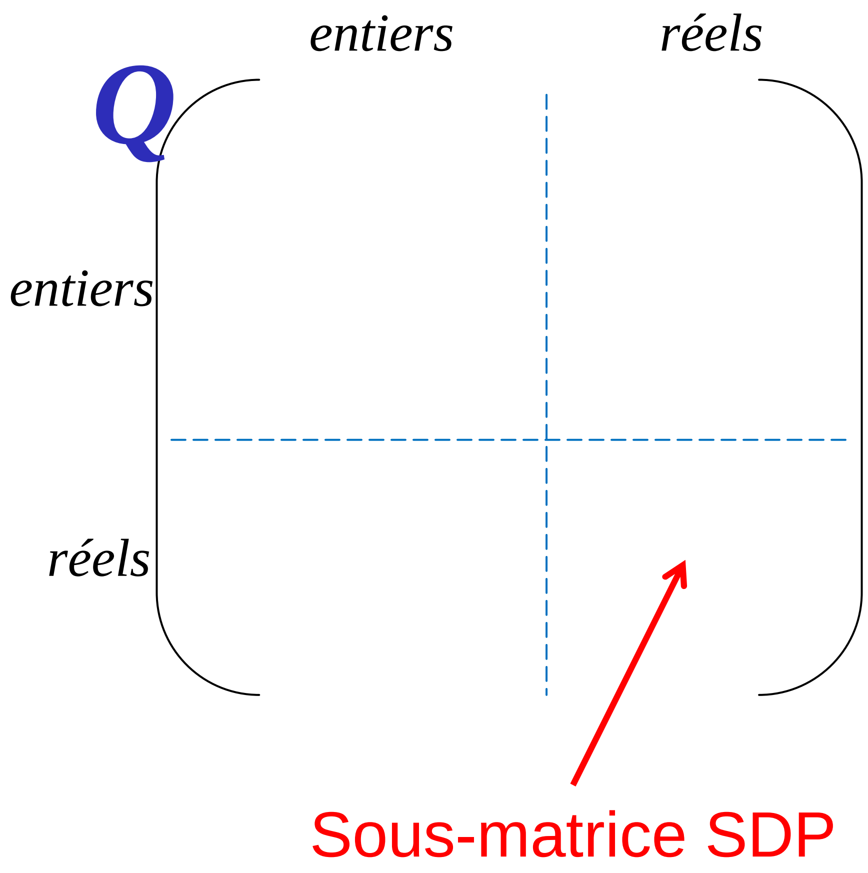
- Conception d'un B&B spécifique

- Mise en ligne d'un logiciel par Amélie Lambert : **SMIQP**
  http://cedric.cnam.fr/~lamberta/smiqp/smiqp.php

- D'autres extensions et expérimentations en cours

Extension au cas des variables mixtes

$$\boxed{\min \quad F(x) = c^t x + x^t Q x}$$



Conception d'un B&B spécifique

$$\min \quad f(x,y) = x^T S_0 x + c_0^T x + < Q_0 - S_0, y >$$

$$s.t.:$$

$$x^T S_k x + c_k^T x + < Q_k - S_k, y > \leq b_k \qquad k = 1,\dots,K$$

$$0 \leq x \leq u$$

$$x \text{ entier}$$

$$y = xx^T$$

Géré par B&B

$$y_{ij} = y_{ji}$$

$$y_{ij} \geq u_j x_i + u_i x_j - u_i u_j$$

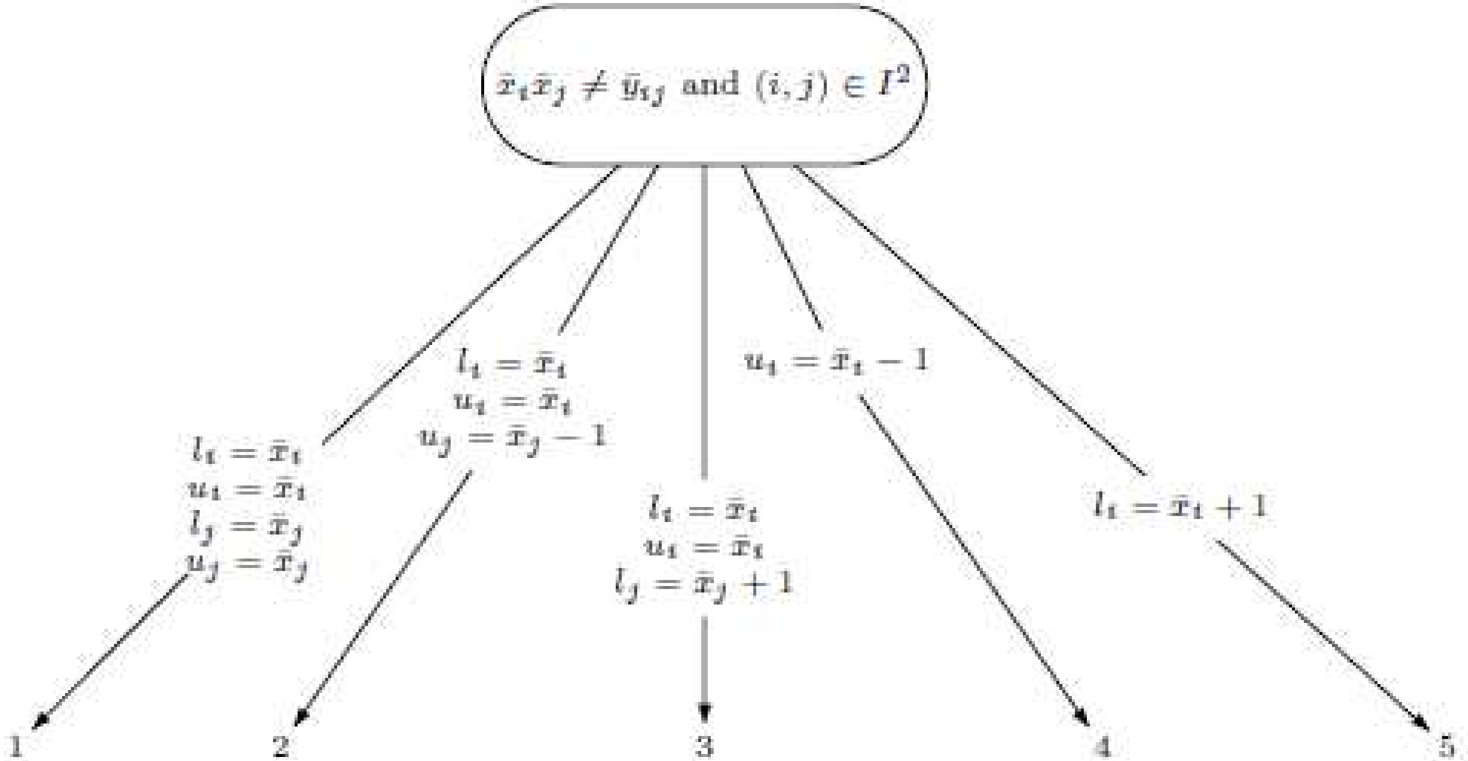
$$y_{ij} \leq u_j x_i$$

$$y_{ii} \geq x_i$$

$$y_{ij} \geq 0$$

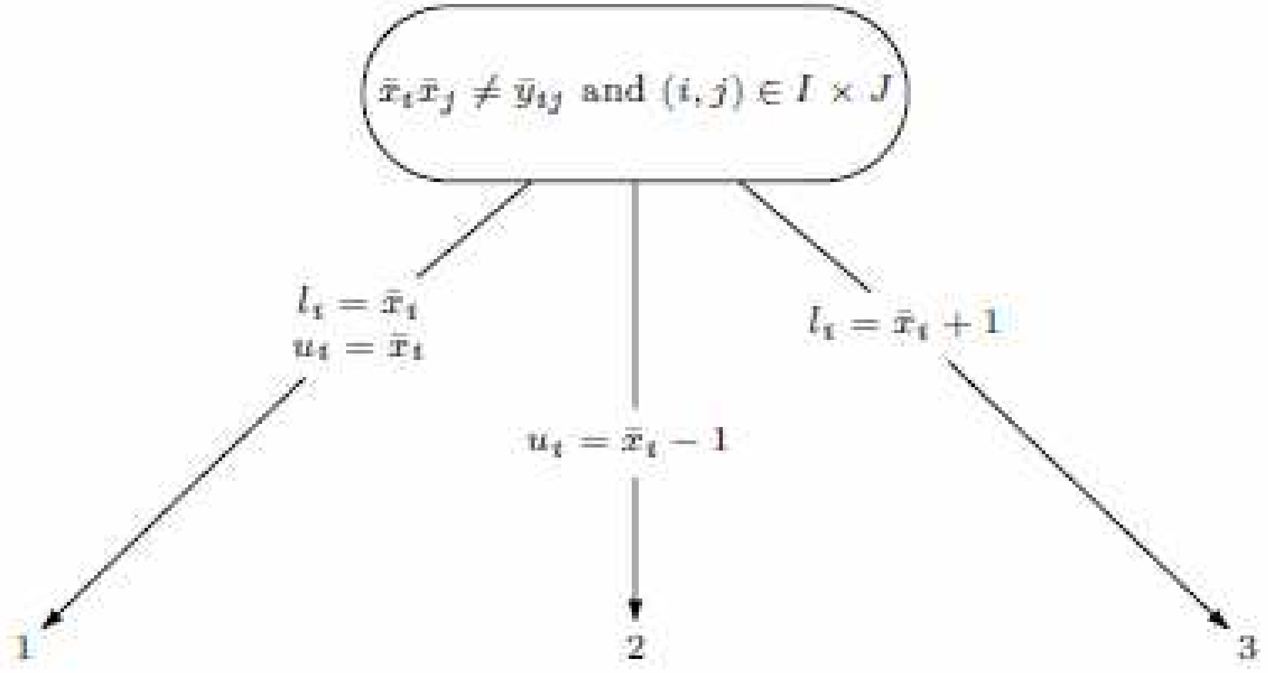
Branchement pour 2 variables entières

Branch 1	$\Longleftrightarrow x_i = \bar{x}_i$ and $x_j = \bar{x}_j$
Branches 2 and 3	$\Longleftrightarrow x_i = \bar{x}_i$ and $x_j \neq \bar{x}_j$
Branches 4 and 5	$\Longleftrightarrow x_i \neq \bar{x}_i$



Branchement pour 1 variable entière, 1 variable continue

$$\boxed{\begin{array}{ll} \text{Branch 1} & \iff x_i = \bar{x}_i \\ \text{Branches 2 and 3} & \iff x_i \neq \bar{x}_i \end{array}}$$



Branchement pour 2 variables continues :

Branch and bound « spatial »

Objet de notre projet PGMO

Conclusions

•Schéma général de reformulation en vue de la résolution exacte

•Incluant les linéarisations

•Familles de reformulations au sein desquelles on peut trouver une formulation « optimale » grâce à des relaxations SDP

•Bonne efficacité expérimentale

Références

- A. Billionnet, S. Elloumi, Using a Mixed Integer Quadratic Programming Solver for the Unconstrained Quadratic 0-1 Problem. *Mathematical Programming*, vol. 109(1), 2007, pp. 55-68.
- A. Billionnet, S. Elloumi, et M.C. Plateau. Quadratic 0-1 programming : tightening linear or quadratic convex reformulation by use of relaxations. *RAIRO Operations Research*, vol. 42(0), 2008, pp. 103-121.
- A. Billionnet, S. Elloumi et M.-C. Plateau. Improving standard solvers convex reformulation for constrained quadratic 0-1 programs: the QCR method.*Discrete Applied Math*, vol. 157(6), 2009, pp. 1185-1197.
- A. Billionnet, S. Elloumi et A. Lambert. Extending the QCR method to general mixed-integer programs. *Mathematical Programming*, vol. 131(1), 2012, pp.381-401
- A. Billionnet, S. Elloumi et A. Lambert. A Branch and Bound algorithm for general mixed-integer quadratic programs based on quadratic convex relaxation.*Journal of Combinatorial Optimization*, 2012
- A. Billionnet, S. Elloumi et A. Lambert. An efficient Compact Quadratic Convex Reformulation for general integer quadratic programs. *Computational Optimization and Applications*, vol. 54(1), pp. 141-162, 2013
- A. Billionnet, S. Elloumi et A. Lambert. Exact quadratic convex reformulations of Mixed-Integer Quadratically Constrained Problems. Article soumis.
- A. Billionnet, S. Elloumi, A. Lambert et A. Wiegele. Using a conic bundle method to accelerate both phases of a quadratic convex reformulation. Article soumis.