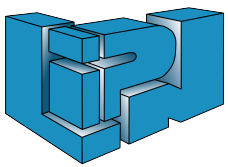
# Semidefinite Optimization for MINLP

Frédéric Roupin

LIPN CNRS UMR 7030, Paris Sorbonne cité
Université Paris Nord




MINO/COST Spring School on MINLP

# Outline

# Mixed Integer Non-Linear Programming

## Mathematical Model

- Mixed Integer Non-Linear Problems

$$
\begin{array}{lll}
\min & g_0(x) & \\
\text{subject to} & g_k(x) \leqslant 0 & k = 1, \ldots, m \\
& l \leqslant x \leqslant u & \\
& x_j \in \mathbb{Z} & i = 1, \ldots, p
\end{array}
$$

- Some $g_k(x)$ are nonlinear functions.
- Very hard to solve. These problems combine all the difficulties of both of their subclasses : integrality and non-convexity.
- In most cases, MINLPs have integer variables:
  - to make a multiple choice selection : decision variables
  - to turn on/off continuous variables and/or constraints.

# Mixed Integer Non-Linear Programming

## Mathematical Model

- Mixed Integer Non-Linear Problems

$$
\begin{array}{lll}
\min & g_0(x) & \\
\text{subject to} & g_k(x) \leqslant 0 & k = 1, \ldots, m \\
& l \leqslant x \leqslant u & \\
& x_j \in \mathbb{Z} & i = 1, \ldots, p
\end{array}
$$

- Few solvers can handle the general model. Frameworks: (spatial)Branch-and-Bound algorithms (convex relaxations), Cutting Plane methods, Outer Approximation or (general) Bender's Decomposition,...
- The underlying relaxations are highly impacted by the choice of the model/reformulation. Solving with black boxes solvers can be difficult (preprocessing, numerical instability, infeasibility).
- Convexifications : LP, CQP, SDP. For instance, nonconvex nonquadratic functions/constraints are first underestimated by a quadratic function.

# Quadratic specializations

## Mixed Integer Quadratically Constrained Quadratic Programming

$$(MIQCQP) \quad \begin{array}{lll} \min & x^T Q_0 x + c_0^T x & \\ \text{s.t.} & x^T Q_i x + c_i^T x \leqslant a_i & i \in \{1, \ldots, m\} \\ & l \leqslant x \leqslant u & \\ & x_j \in \mathbb{Z} & j = 1, \ldots, p \end{array}$$

- All the $g_k(x)$ are quadratic (or linear) functions.
- Can model a large class of problems.
- Can appear as a subproblem when solving general MINLP.

## Mixed Integer Quadratic Programming

$$(MIQP) \quad \begin{array}{lll} \min & x^T Q x + c^T x & \\ \text{s.t.} & Ax = b & \\ & l \leqslant x \leqslant u & \\ & x_j \in \mathbb{Z} & i = 1, \ldots, p \end{array}$$

# Notations

## Inner Product

$\mathcal{S}^n$ : $n \times n$ real symmetric matrices

Standard inner product on $\mathcal{S}^n$ :

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

## Matrix Form

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \Re^n \qquad xx^T = \begin{bmatrix} x_1^2 & \cdots & x_1 x_i & \cdots & x_1 x_n \\ \vdots & \ddots & & & \vdots \\ x_i x_1 & & x_i^2 & & x_i x_n \\ \vdots & & & \ddots & \vdots \\ x_n x_1 & \cdots & x_n x_i & \cdots & x_n^2 \end{bmatrix}$$

Example. $x_1^2 + 2x_2^2 - x_1 x_2 = \left\langle \begin{bmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 2 \end{bmatrix}, xx^T \right\rangle$

# Example: Matrix Form for (MIQCQP)

## Mathematical Model

$$(MIQCQP) \quad \begin{array}{lll} \min & x^T Q_0 x + c_0^T x & \\ \text{s.t.} & x^T Q_i x + c_i^T x \leqslant a_i & i \in \{1, \ldots, m\} \\ & x_j \in \mathbb{Z} & j = 1, \ldots, p \end{array}$$

## Matrix form

$$\begin{array}{lll} \min & \langle A_0, X \rangle & \\ \text{s.t.} & \langle A_i, X \rangle \leqslant 0, & i \in \{1, \ldots, m\} \\ & X = \begin{bmatrix} 1 & x^T \\ x & xx^T \end{bmatrix} & \\ & x_j \in \mathbb{Z} & j = 1, \ldots, p \end{array}$$

$$\langle A_i, X \rangle = \langle \begin{bmatrix} -a_i & \frac{1}{2} c_i^T \\ \frac{1}{2} c_i & -Q_i \end{bmatrix}, X \rangle = x^T Q_i x + c_i^T x - a_i$$

# Positive Semi-Definite matrices

## Definition ($\mathcal{S}_+^n$)

$A$ is Positive Semi-Definite (PSD), $A \succcurlyeq 0$, iff
$x^T Ax = \sum_{i=1}^n \sum_{j=1}^n A_{ij} x_i x_j = \langle A, xx^T \rangle \geqslant 0$ for all $x \in \Re^n$

## Proposition

$$A \in \mathcal{S}_+^n \text{ iff } A = M^T \begin{bmatrix} \lambda_1 & \dots & 0 \\ & \ddots & \\ 0 & \cdot & 0 \\ 0 & \dots & \lambda_n \end{bmatrix} M \text{ where } \lambda \geqslant 0 \text{ and } M^T M = I_n$$

## Definition

Principal minor Determinant of a $k \times k$ submatrix of $A$ that corresponds to a subset of rows and columns of $A$ with same indices

## Proposition

*$A \in \mathcal{S}_+^n$ iff if and only if all principal minors of $A$ are nonnegative*

# Linear matrix inequality

## Definition

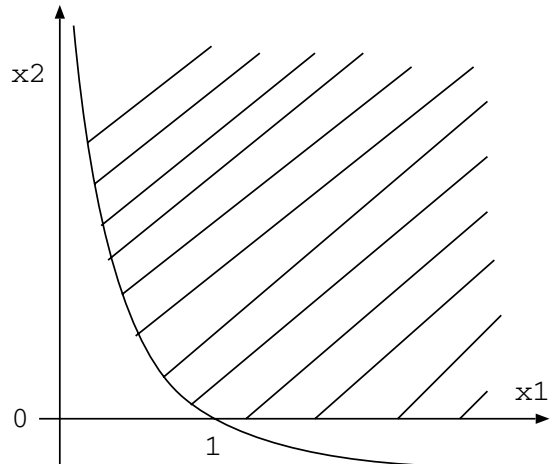
*A linear matrix inequality is* $\sum_{i=1}^{m} x_i A_i - A_0 \succcurlyeq 0$, *where* $A_i \in S_n$ $(i = 0, \ldots, m)$ *and* $x \in \Re^m$

## Example

Let $x \in \Re^2$ be such that

$$x_1 \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A_1} + x_2 \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{A_2} - \underbrace{\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A_0} = \begin{bmatrix} x_1 & 1 & 0 \\ 1 & x_2+1 & 0 \\ 0 & 0 & x_2 \end{bmatrix} \succcurlyeq 0$$

here, $x$ is such that: $x_1 \geqslant 0$, $x_2 \geqslant 0$, $x_1(x_2 + 1) \geqslant 0$



# Linear matrix inequality (2)

## Proposition

$S = \left\{ x \in \Re^m \, : \, F(x) = \sum_{i=1}^m x_i A_i - A_0 \succcurlyeq 0 \right\}$ *is a closed convex set but not necessarily a convex polytope if* $m \geqslant 2$.

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

## Convex polytope

If all $A_i$ are diagonal matrices then $\sum_{i=1}^m x_i A_i - A_0 \succcurlyeq 0$ reduces to $Ax \geqslant a_0$

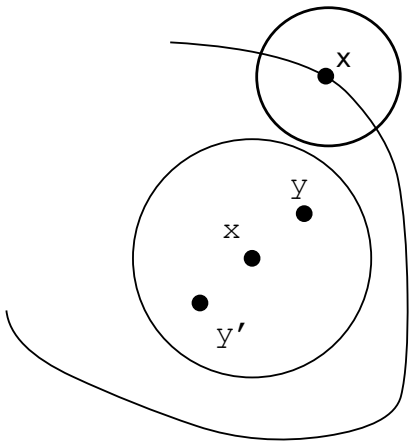
## Example

$$x_1 \begin{bmatrix} 2 & 0 \\ 0 & 5 \end{bmatrix} + x_2 \begin{bmatrix} -1 & 0 \\ 0 & 7 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & -6 \end{bmatrix} = \begin{bmatrix} 2x_1 - x_2 - 1 & 0 \\ 0 & 5x_1 + 7x_2 + 6 \end{bmatrix} \succcurlyeq 0$$

# Feasible set of a Semi-Definite Program

## Proposition

$$\overset{\circ}{S} = \{x \mid F(x) \succ 0\} \quad \textit{and} \quad \partial S = \{x \mid F(x) \succcurlyeq 0 \ \det F(x) = 0\}$$



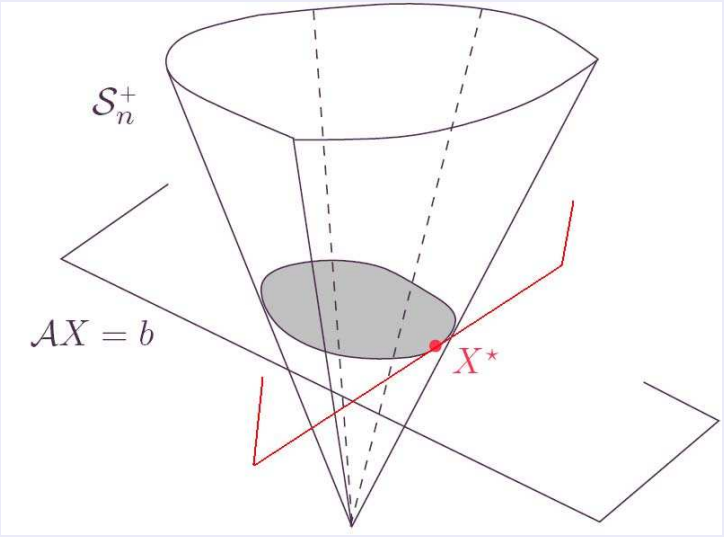

Continuity of determinants: if $x$ belongs to $\partial S$ then $F(x)$ is singular.

Density of invertible matrices: if $x$ belongs to $\overset{\circ}{S}$ then $F(x)$ is Positive Definite.

Algebraic surfaces: each $x$ on the boundary is a root of some polynomials (determinant is zero).

# Semi-Definite Optimisation

## Semidefinite Programming




$$(SDP) \quad \begin{array}{ll} \min & \langle c, x \rangle \\ \text{s.t.} & \sum_{i=1}^m x_i A_i - A_0 \succcurlyeq 0 \end{array}$$

For $x \in \Re^m$, $\sum_{i=1}^m x_i A_i - A_0$ is a matrix belonging to the affine space defined by the affine basis : $-A_0$ (origin) and $A_1, \ldots, A_m$ (basis)

## This affine space can also be defined by a set of linear constraints

$$S = \{X \mid X \succcurlyeq 0, \langle B_i, X \rangle = b_i \in \{1, \ldots, p\}\}, \; m = \frac{n(n+1)}{2} - p$$

$$(SDP) \leftrightarrow \begin{array}{lll} \min & \langle C, X \rangle + \langle C, A_0 \rangle & = \langle C, \sum_{i=1}^m x_i A_i \rangle \\ \text{s.t.} & \langle B_i, X \rangle = b_i \quad \forall i \in \{1, \ldots, p\} & \\ & X \succcurlyeq 0 & \end{array}$$

# Semidefinite Optimization

## A Semidefinite Programming (SDP) Problem

$$\text{(SDP)} \qquad \begin{array}{ll} \text{minimize} & \langle C, X \rangle \\ \text{subject to} & \langle A_i, X \rangle = b_i, \ \ i = 1, \ldots, m \\ & X \succcurlyeq 0 \end{array}$$

## Applications

- graph theory (graph embedding, graph rigidity, Max-Cut, maximum stable set, chromatic number)
- distance geometry (sensor network localization, protein structure determination, machine learning)
- combinatorial optimization (binary quadratic programming, quadratic assignment problem)
- polynomial optimization (Poly-SDP, SOS: sum-of-squares)
- low-rank matrix completion (nuclear norm minimization, phase-lift)

# Duality

## Dual Program

- $\mathcal{S}_+^n$ is a self-dual cone: $\mathcal{S}_+^n = \{X : \langle X, Y\rangle \geqslant 0 \, \forall Y \in \mathcal{S}_+^n\}$.
- Dual program of (SDP)

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

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

  - The Lagrange multiplier $Z$ corresponding to the linear matrix inequality $F(x) \succcurlyeq 0$ belongs to the dual cone of $\mathcal{S}_n^+$, thus $Z \succcurlyeq 0$.

- Dual function:
  - If $\langle A_i, Z\rangle = c_i \; \forall i \in \{1, \ldots, m\}$ then $\inf_{x \in \Re^m} L(x, Z) = \langle A_0, Z\rangle$
  - otherwise: some terms involving $x_i$ are not zero and thus $\inf_{x \in \Re^m} L(x, Z) = -\infty$

## Proposition

*The dual program of (SDP) is a Semi-Definite Program*

$$(DSDP) \begin{cases} \sup & \langle A_0, Z\rangle \\ \text{s.t.} & \langle A_i, Z\rangle = c_i \quad i \in \{1, \ldots, m\} \\ & Z \succcurlyeq 0 \end{cases}$$

# Semidefinite Optimisation

## Semidefinite Programming (SDP) versus Linear Programming (LP)

(PL)

$$\begin{array}{ll} \text{minimize} & \langle c, x\rangle \\ \text{subject to} & \langle a_i, x\rangle = b_i,\ i = 1, \ldots, m \\ & x \geqslant 0 : \mathbb{R}^{n+} \end{array}$$

(SDP)

$$\begin{array}{ll} \text{minimize} & \langle C, X\rangle \\ \text{subject to} & \langle A_i, X\rangle = b_i,\ i = 1, \ldots, m \\ & X \succcurlyeq 0 : \mathcal{S}^n_+ \end{array}$$

## Self-dual cones

LP : $\langle c, x\rangle = \sum_i c_i x_i$ cone $\mathbb{R}^{n+} = \{x : \langle x, y\rangle \geqslant 0\, \forall y \in \mathbb{R}^{n+}\}$

SDP : $\langle C, X\rangle = \sum_{ij} C_{ij} X_{ij}$ cone $\mathcal{S}^n_+ = \{X : \langle X, Y\rangle \geqslant 0\, \forall Y \in \mathcal{S}^n_+\}$

$\longrightarrow$ Here, each cone is its own dual cone: the dual program of a LP is a LP, and the dual program of a SDP is a SDP.

# Duality Theorems

## Weak duality

The duality gap between $(DSP)$ and $(SDP)$ is:
$\langle c, x\rangle - \langle A_0, Z\rangle = \sum_{i=1}^{m} \langle A_i, Z\rangle x_i - \langle A_0, Z\rangle = \langle F(x), Z\rangle \geqslant 0$

## Strong duality

("Slater" qualification)

- (SDP) has some strictly feasible point, i.e. $\exists x \,/\, F(x) \succ 0$.
- (DSDP) has some strictly feasible point, i.e.
  $\exists Z \,/\, Z \succ 0,\ \langle A_i, Z\rangle = c_i,\ i = 1, \ldots, m$.
- If at least one hypothesis is satisfied then there is no duality gap. If both are satisfied then there are optimal solutions for (SDP) and (DSDP)

# Complementary slackness

## Proposition

*Assume that the duality gap is zero ($\langle F(x), Z\rangle = 0$). $x$ and $Z$ are respectively optimal solutions for (SDP) and (DSDP) iff:*

$$F(x) \succcurlyeq 0$$
$$Z \succcurlyeq 0,\ \langle A_i, Z\rangle = c_i,\ \ i = 1, \ldots, m$$
$$ZF(x) = 0 \Leftrightarrow \langle F(x), Z\rangle = 0$$

## Generalizes the Linear Programming result

- In Linear programming: $F(x)$ and $Z$ are diagonal matrices
- $ZF(x)$ can be written as $Z_{ii}\left(A_i^T x - b_i\right) = 0\ \forall i$

# Schur complement

## Theorem

*Let $A$ a $p \times p$ Positive Definite matrix, $C \in \mathcal{S}^n$, and $B$ a $p \times n$ matrix, then* $\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succcurlyeq 0$ *is equivalent to* $C - B^T A^{-1} B \succcurlyeq 0$.

## Proof.

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

□

## Particular cases

- $A = I_p$. One has $\begin{bmatrix} I_p & B \\ B^T & C \end{bmatrix} \succcurlyeq 0 \Leftrightarrow C - B^T B \succcurlyeq 0$
- $p = 1$, $C = X$, $B = x^T$. Let $(X, x) \in S_n \times \Re^n$, one has

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

# Exploiting Schur Complement

## a Non-Linear Program

$$\text{(NLP)} \qquad \boxed{\begin{array}{ll} \min & \dfrac{\left(c^T x\right)^2}{d^T x} \\ \text{s.t.} & Ax \geqslant b \end{array}}$$

Assume that $d^T x > 0$ if $x$ is feasible.

## (NLP) can be stated as a Semi-Definite Program

$$\text{(NLP)} \leftrightarrow \boxed{\begin{array}{ll} \min & t \\ \text{s.t.} & Ax + b \geqslant 0 \\ & \dfrac{\left(c^T x\right)^2}{d^T x} \leqslant t \end{array}} \leftrightarrow \boxed{\begin{array}{ll} \min & t \\ \text{s.t.} & \begin{bmatrix} \operatorname{diag}(Ax+b) & 0 & 0 \\ 0 & t & c^T x \\ 0 & c^T x & d^T x \end{bmatrix} \succcurlyeq 0 \end{array}}$$

where $\operatorname{diag}(Ax - b) = \begin{bmatrix} A_1^T x - b_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & A_m^T x - b_m \end{bmatrix}$

# Convex Quadratic Optimization

## Formulation

$$(\text{CQP}) \quad \boxed{\begin{array}{ll} \min & f_0(x) \\ \text{s.t.} & f_i(x) \leqslant 0 \quad i = 1, \ldots, m \end{array}}$$

where all $f_i(x) = x^T A_i^T A_i x + c_i^T x + d_i$ are convex quadratic functions (Cholesky decomposition)

## Convex Quadratic Programs can be stated as Semi-Definite Programs

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

$$(CQP) \Leftrightarrow \boxed{\begin{array}{ll} \min & t \\ \text{s.t.} & \begin{bmatrix} I_n & A_0 x \\ x^T A_0^T & -c_0^T x - d_0 + t \end{bmatrix} \succcurlyeq 0 \qquad f_0(x) - t \leqslant 0 \\ & \begin{bmatrix} I_n & A_i x \\ x^T A_i^T & -c_i^T x - d_i \end{bmatrix} \succcurlyeq 0 \;\; i = 1, \ldots, m \end{array}}$$

# Convex Quadratic Optimization (2)

## Using a unique Linear Matrix Inequality

$$(CQP) \quad \boxed{\begin{array}{lll} \min & x^T A_0^T A_0 x + c_0^T x + d_0 & (\leqslant t) \\ \text{s.t.} & x^T A_i^T A_i x + c_i^T x + d_i \leqslant 0 & i = 1, \ldots, m \end{array}}$$

$$\Leftrightarrow (SDP) \quad \boxed{\begin{array}{ll} \min & t \\ \text{s.t.} & \left\langle \begin{bmatrix} d_0 - t & \frac{1}{2} c_0^T \\ \frac{1}{2} c_0 & A_0^T A_0 \end{bmatrix}, \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \right\rangle \leqslant 0 \\ & \left\langle \begin{bmatrix} d_i & \frac{1}{2} c_i^T \\ \frac{1}{2} c_i & A_i^T A_i \end{bmatrix}, \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \right\rangle \leqslant 0 \\ & \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0 \Leftrightarrow X - xx^T \succcurlyeq 0! \end{array}}$$

## Proof.

- If $x$ is feasible for $(Q)$ then $(x, xx^T)$ is feasible for $(SDP)$
- If $(x, X)$ is feasible for $(SDP)$ then
$$\underbrace{\langle A_i^T A_i, X - xx^T \rangle}_{\geqslant 0} + x^T A_i^T A_i x + c_i^T x + d_i \leqslant 0$$

# Mixed Integer Quadratically Constrained Quadratic Problem

## (MIQCQP) (matrix form)

$$
\begin{array}{lll}
\min & \langle A_0, Z \rangle & \\
\text{s.t.} & \langle A_i, Z \rangle \leqslant 0, & i \in \{1, \ldots, m\} \\
& Z = \begin{bmatrix} 1 & x^T \\ x & xx^T \end{bmatrix} & \\
& x_j \in \mathbb{Z} & j \in \{1, \ldots, p\}
\end{array}
$$

## Basic Semi-Definite relaxation

1. Drop the integrality constraints $x_j \in \mathbb{Z}$.
2. Replace $X = xx^T$ by $X$ such that $X - xx^T \succcurlyeq 0$.

Schur complement : $X - xx^T \succcurlyeq 0 \Leftrightarrow \underbrace{\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0}_{\text{Linear Matrix Inequality}}$

This is equivalent to drop the rank constraint $rank(\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix}) = 1$

# Decision variables

## Semidefinite Approach

$x \in \mathbb{R}^n$ contains $p$ decision variables. $Z$ is a rank 1 Positive Semi-Definite matrix

$$\begin{array}{lll} \min & \langle A_0, Z \rangle & \\ \text{s.t.} & \langle A_i, Z \rangle \leqslant 0, & i \in \{1, \ldots, m\} \\ & Z = \begin{bmatrix} 1 & x^T \\ x & xx^T \end{bmatrix} & \\ & x_j \in \{0, 1\} & j \in \{1, \ldots, p\} \end{array}$$

$x_j \in \{0, 1\}$ thus $X_{jj} = x_j$ is a valid equality.

## Semi-Definite relaxation

$$(SDP) \quad \begin{array}{lll} \min & \langle A_0, Z \rangle & \\ \text{s.t.} & \langle A_i, Z \rangle \leqslant 0, & i \in \{1, \ldots, m\} \\ & Z = \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0 & \\ & X_{jj} = x_j & j \in \{1, \ldots, p\} \end{array}$$

This implies $\begin{bmatrix} 1 & x_j \\ x_j & x_j \end{bmatrix} \succcurlyeq 0$ i.e. $0 \leqslant x_i^2 \leqslant x_i \leqslant 1$, but also many other valid polynomial inequalities (take any $k \times k$ principal minor of $Z$).

# -1,1 model for decision variables

## -1,1 reformulation

Convert from $x \in \{0,1\}^p$ to $y \in \{-1,1\}^p$ $(y_j^2 = 1)$: $\begin{pmatrix} 1 \\ x \end{pmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{2}e & \frac{1}{2}I_n \end{bmatrix} \begin{pmatrix} 1 \\ y \end{pmatrix}$

Following the same recipe, we get

$$\begin{array}{lll} \min & \langle B_0, Y \rangle & \\ \text{s.t.} & \langle B_i, Y \rangle \leqslant 0, & i \in \{1,\dots,m\} \\ & Y \succcurlyeq 0,\ rank(Y) = 1 & \\ & Y_{jj} = 1 & j \in \{1,\dots,p+1\} \end{array}$$

## Basic Semi-Definite relaxation: drop the rank constraint

(SDP)

$$\begin{array}{lll} \max & \langle Q, X \rangle & \\ \text{s.t.} & \langle A_i, X \rangle \leqslant a_i, & i \in \{1,\dots,m_I\} \\ & \langle B_i, X \rangle = b_i, & i \in \{1,\dots,m_E\} \\ & X \succcurlyeq 0 & \end{array}$$

# Equivalence of basic Semi-Definite relaxations

## Proposition

*Let $Q$ be a $n \times n$ invertible matrix and $M \in \mathcal{S}^n$. Consider $\phi(M) = QMQ^T$. $S_n^+$ is stable under $\phi$.*

## Changing variables in a Semi-Definite Program

$$(SDP) \quad \begin{array}{ll} \min & \langle C, X \rangle \\ \text{s.t.} & \langle A_i, X \rangle = (\text{ or } \leqslant) b_i \; ; \; i = 1, \ldots, m \\ & X \succcurlyeq 0 \end{array}$$

$W$ and $X$ are similar matrices: $W = QXQ^T$

$$(SDP_1)_Q \quad \begin{array}{ll} \min & \langle \left(Q^{-1}\right)^T C Q^{-1}, W \rangle \\ \text{s.t.} & \langle \left(Q^{-1}\right)^T A_i Q^{-1}, W \rangle = (ou \leqslant) \, b_i \; ; \; i = 1, \ldots, m \\ & W \succcurlyeq 0 \end{array}$$

## Lemma

*$X$ is a feasible point of $(SDP)$ iff $W = QXQ^T$ is a feasible point of $(SDP)_Q$. Moreover, one has $\langle C, X \rangle = \langle \left(Q^{-1}\right)^T C Q^{-1}, W \rangle$*

# Equivalent Semi-Definite relaxations

## Working with $\{0,1\}$ or $\{-1,1\}$ decision variables

Let $x \in \{0,1\}^n$ and $y \in \{-1,1\}^n$ be such that $x = \frac{1}{2}(y + e_n)$ Thus $\begin{pmatrix} 1 \\ x \end{pmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{2}e & \frac{1}{2}I_n \end{bmatrix} \begin{pmatrix} 1 \\ y \end{pmatrix}$. One has a $Q^{-1} = \begin{bmatrix} 1 & 0 \\ -e & 2I_n \end{bmatrix}$

$$
\begin{aligned}
\min \quad & \left\langle \begin{bmatrix} 0 & \frac{1}{2}b^T \\ \frac{1}{2}b & C \end{bmatrix}, X \right\rangle \\
\text{s.t.} \quad & X_{kk} = x_k \ \forall k \in \{1,\dots,p\} \\
& 0 \leqslant X_{ij} \ \forall i \neq j \\
& X_{ij} \leqslant X_{ii} \ \forall i \neq j \\
& X_{ij} \leqslant X_{jj} \ \forall i \neq j \\
& X_{ii} + X_{jj} \leqslant 1 + X_{ij} \ \forall i \neq j \\
& \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0
\end{aligned}
$$

$$
\begin{aligned}
\min \quad & \left\langle \left(Q^{-1}\right)^T \begin{bmatrix} 0 & \frac{1}{2}b^T \\ \frac{1}{2}b & C \end{bmatrix} Q^{-1}, Y \right\rangle \\
\text{s.t} \quad & Y_{kk} = 1 \ \forall k \in \{1,\dots,p+1\} \\
& Y_{ij} - Y_{i1} - Y_{j1} \geqslant -1 \ \forall i \neq j \\
& Y_{i1} - Y_{j1} - Y_{ij} \geqslant -1 \ \forall i \neq j \\
& -Y_{i1} + Y_{j1} - Y_{ij} \geqslant -1 \ \forall i \neq j \\
& Y_{1i} + Y_{1j} + Y_{ij} \geqslant -1 \ \forall i \neq j \\
& Y \succcurlyeq 0
\end{aligned}
$$

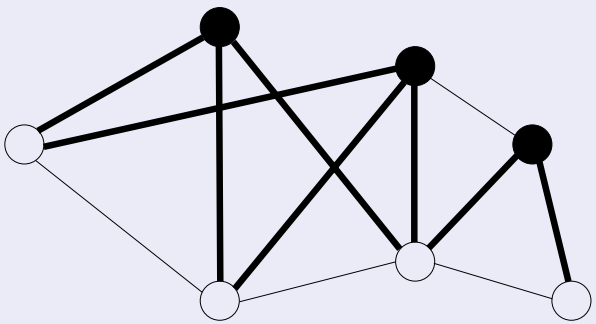
## Why should one consider two equivalent SDP ?

$\longrightarrow$ 0,1 models are easily obtained from an existing Linear Programming relaxation

$\longrightarrow$ -1,1 models have a geometrical interpretation: useful to design approximated solutions

# Example: the Max-Cut problem

## Max-Cut



$$\begin{array}{ll} \max & \displaystyle\sum_{ij\in E} w_{ij}\left(\frac{1-x_i x_j}{2}\right) \\ \text{s.t.} & x \in \{-1,1\}^n \end{array}$$

$G = (V, E)$ $\qquad$ $(n = |V|,\ w_{ij} = 0 \text{ if } ij \notin E)$

## Matrix form

(MC)
$$\begin{array}{ll} \max & x^T Q x \\ \text{s.t.} & x \in \{-1,1\}^n \end{array}$$

- $Q = \frac{1}{4}L \in \mathcal{S}^n$ $\quad$ ($L$ is the Laplacian matrix of $G$)

# Example: Max-Cut problem

## Max-Cut

$$\text{(MC)} \quad \boxed{\begin{array}{ll} \max & x^T Q x \\ \text{s.t.} & x \in \{-1,1\}^n \end{array}}$$

## $X = xx^T,\ x \in \{-1,1\}^n \iff \text{diag}(X) = e,\ X \succcurlyeq 0,\ \text{rank}(X) = 1$

$$\text{(MC)} \quad \boxed{\begin{array}{ll} \max & \langle Q, X \rangle \\ \text{s.t.} & \text{diag}(X) = e,\ X \succcurlyeq 0 \\ & \mathit{rank}(X) = 1 \end{array}}$$

## Basic SDP relaxation: drop the rank constraint

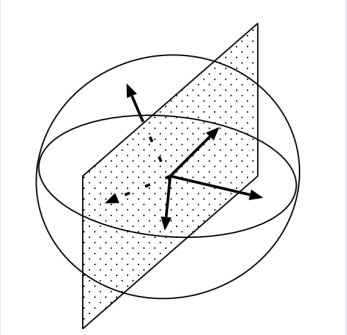
$$\text{(SDP)} \quad \boxed{\begin{array}{ll} \max & \langle Q, X \rangle \\ \text{s.t.} & \text{diag}(X) = e,\ X \succcurlyeq 0 \end{array}}$$

- provides an upper bound Max-Cut: $\text{(MC)} \leqslant \text{(SDP)} < 1.14\text{(MC)}$
- can be solved efficiently by interior-point solvers (strictly feasible points). But the bound is too weak to solve exactly Max-Cut

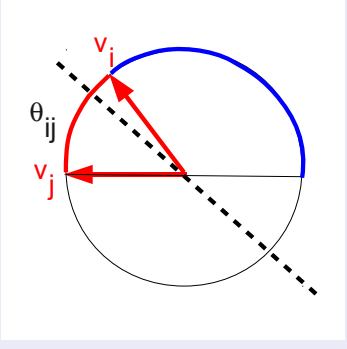
# Max-Cut : using the SDP optimal solution

## How to build a good feasible solution from the SDP relaxation solution

1. $X^* = V^T V$ optimal solution of the SDP.
2. Take vector $e_0$ randomly distributed on the unit sphere of $\mathbb{R}^n$.
   If $\langle v_i, e_0 \rangle \geqslant 0$ then $x_i = 1$ otherwise $x_i = -1$



3. $\langle v_i, v_j \rangle = cos(\theta_{ij})||v_i||||v_j|| = cos(\theta_{ij})$

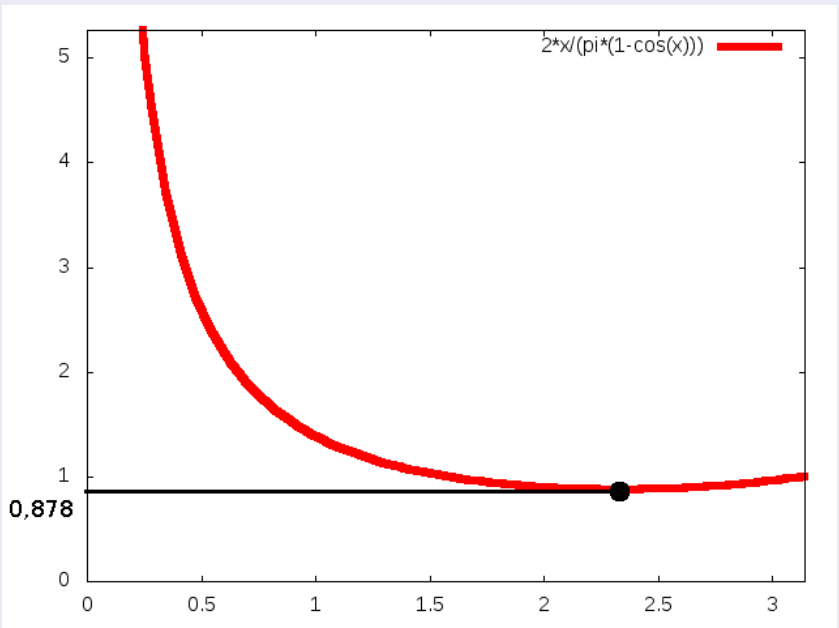



4. $Pr_{ij}$ : the probability to have $x_i \neq x_j$ is proportional to the angle $\theta_{ij} = \arccos(\langle v_i, v_j \rangle)$ : $Pr_{ij} = \frac{1}{\pi} \arccos(\langle v_i, v_j \rangle)$

# Max-Cut : basic Semi-Definite relaxation

## Building an approximated solution

- The expectation of the solution $y$ is
  $E[W] = \sum_{i<j} w_{ij} Pr_{ij} = \frac{1}{\pi} \sum_{i<j} w_{ij} \arccos(\langle v_i, v_j \rangle)$
- The SDP bound is $\frac{1}{2} \sum_{i<j} w_{ij}(1 - \langle v_i, v_j \rangle)$
- Guarantee: minimize $\frac{2}{\pi} \frac{\theta}{1-cos\theta}$ for $0 \leqslant \theta \leqslant \pi$.
  One has $\alpha = \min_{0 \leqslant \theta \leqslant \pi} \frac{2}{\pi} \frac{\theta}{1-cos\theta} > 0,878$




Thus $E[W] \geqslant \alpha \sum_{i<j} w_{ij} (1 - \langle v_i, v_j \rangle) \geqslant 0,878 val_{opt}(maxcut)$

# Strengthening the SDP bound

## Intersecting Semi-Definite and polyhedral relaxations

There are $4\binom{n}{3}$ triangle inequalities (for $1 \leqslant i < j < k \leqslant n$):

$$X_{ij} + X_{ik} + X_{jk} \geqslant -1, \quad -X_{ij} + X_{ik} - X_{jk} \geqslant -1$$
$$X_{ij} - X_{ik} - X_{jk} \geqslant -1, \quad -X_{ij} - X_{ik} + X_{jk} \geqslant -1$$

Adding a subset $\mathbb{I}$ of these inequalities, one gets

$$(\mathrm{SDP}_{\mathbb{I}}) \quad \boxed{\begin{array}{ll} \max & \langle Q, X \rangle \\ \text{s.t.} & \mathrm{diag}(X) = e,\ X \succcurlyeq 0 \\ & \mathcal{A}_{\mathbb{I}}(X) \geqslant -e \end{array}}$$

- can greatly enhance the bound::

$$(\mathrm{MC}) \leqslant (\mathrm{SDP}_{\mathbb{I}}) \leqslant (\mathrm{SDP})$$

- But solving this SDP is not an easy task: generally, specific solvers (depending on the problem structure) have to be used

# Linear constraints and Semi-Definite relaxations

## General Scheme

$\longrightarrow$ Use the Basic SDP relaxation models

$\longrightarrow$ These relaxations can be reinforced by cuts/valid inequalities (e.g. proposed in LP relaxations). In particular, non-negativity constraints $X_{ij} \geqslant 0$ and more generally triangle inequalities are efficient.

$\longrightarrow$ Warning: a common mistake is to leave alone linear constraints (such as $c^T x = d$) in the SDP relaxation.

## Some possible treatments

$\longrightarrow$ Replace $c^T x = d$ by (equivalent for SDP !)

1. $\langle cc^T, X \rangle - 2dc^T x + d^2 = 0$
2. or $\sum_{j=1}^{n} c_j X_{ij} = d x_i \; \forall i \in \{1, \ldots, n\}$ and $c^T x = d$

$\longrightarrow$ Replace $b \leqslant a^T x \leqslant b'$ by

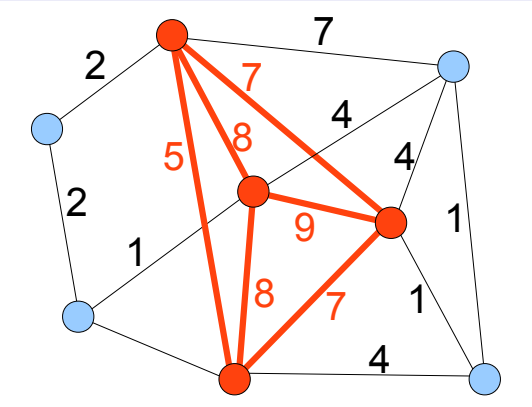
1. $\langle aa^T, X \rangle - (b + b') a^T x + bb' \leqslant 0$
2. or $b(1 - x_i) \leqslant \sum_{j=1}^{n} a_j (x_j - X_{ij}) \;\; \forall i$, $\sum_{j=1}^{n} a_j (x_j - X_{ij}) \leqslant b'(1 - x_i) \;\; \forall i$

## Choice depends on the solver

- Interior-point methods (e.g. CSDP) : use models with fewer constraints
- Dual approaches (e.g. Conic Bundle, quasi-Newton) : Product constraints are generally better

# Example: Maximum $k$-cluster

## Maximum $k$-cluster




$$\begin{array}{ll} \max & \frac{1}{2} z^T W z \\ \text{s.t.} & \sum_{i=1}^n z_i = \langle e_n, z \rangle = k \\ & z \in \{0,1\}^n \end{array}$$

$G = (V, E)$ $\qquad (n = |V|,\ w_{ij} = 0 \text{ if } ij \notin E)$

## Semi-Definite Relaxations

$$\begin{array}{ll} \max & \frac{1}{2} \langle W, Z \rangle \\ \text{s.t.} & \langle e_n e_n, Z \rangle - 2k e_n^T z + k^2 = 0 \quad \text{or} \ \sum_{i=1}^n Z_{ij} = k z_j,\ j \in \{1, \ldots, n\} \\ & \qquad\qquad\qquad\qquad\qquad\qquad \langle e_n, z \rangle = k \end{array}$$

$$Z_{ii} = z_i,\ i \in \{1, \ldots, n\}$$

$$\begin{bmatrix} 1 & z^T \\ z & Z \end{bmatrix} \succcurlyeq 0$$

# Squared constraint and Product constraints

## Proposition

*Let* $(X, x) \in S_n \times \mathbb{R}$ *be such that* $\langle cc^T, X\rangle - 2dc^T x + d^2 = 0$ *and* $\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0$, *then* $\sum_{j=1}^{n} c_j X_{ij} = dx_i \; \forall i \in \{1, \ldots, n\}$

## Proof.

$\langle cc^T, X\rangle - 2dc^T x + d^2 = 0 \rightarrow \langle cc^T, X - xx^T\rangle + \left(c^T x - d\right)^2 = 0$
Thus $c^T x = d$ and $\langle cc^T, X - xx^T\rangle = 0$ and consequently $cc^T \left(X - xx^T\right) = 0$
i.e.

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

- $c = 0 : \; 0 = 0$ is true !
- There exists $k_0$ such that $c_{k_0} \neq 0$ :

  $\forall j \in \{1, ..., n\} \; \sum_{i=1}^{n} c_i X_{ij} = x_j \sum_{i=1}^{n} c_i x_i = dx_j.$

□

# Example: Max-Independent Set Problem

## 0-1 LP model

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

- $(x_i + x_j - 1)(x_i + x_j) \leqslant 0 \ \forall (i,j) \in E$
- For all $(i,j)$, $x_i x_j \geqslant 0$. Thus $0 \leqslant x_i x_j \leqslant 0$ ! $\leftrightarrow X_{ij} = 0$ for $(i,j) \in E$

## Semi-Definite Relaxation

$$\begin{array}{lll} \max & \sum_{i=1}^{n} x_i & \\ \text{s.t.:} & X_{ij} = 0 & (i,j) \in E \\ & X_{ij} \geqslant 0 & (i,j) \notin E \\ & X_{ii} = x_i & i \in \{1,\ldots,n\} \\ & \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0 & \end{array}$$

# The Lagrangian framework to design Semi-Definite relaxations

## Lagrangian duality for QP

Consider any Quadratic Problem

$$(QP) \quad \boxed{\begin{array}{ll} \min_{x\in\Re^n} & x^T A_0 x + b^T x \\ \text{s.t.} & x^T A_i x + c_i^T x - d_i \leqslant 0 \quad \forall i \in \{1,\ldots,m\} \end{array}}$$

The dual program is

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

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

## Lemma

*$\Theta(\lambda)$ is finite iff $A(\lambda) \succcurlyeq 0$ and $\exists x_\lambda$ such that $2A(\lambda)x_\lambda + b(\lambda) = 0$*

# The dual program of CQP can be formulated as a SDP

## Proposition

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

*can be formulated as* $(SD)$

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

## Proof.

- $(r, \lambda)$ is feasible for (SD) $\Leftrightarrow \forall (\alpha, y) \in \Re \times \Re^n$ $q(\alpha, y) = -\alpha^2 r + \alpha b(\lambda)^T y + y^T A(\lambda) y$ is Positive Semi-Definite
- This implies $b(\lambda) \in Im(A(\lambda))$
- $\alpha = 0 : \ A(\lambda) \succcurlyeq 0$
- $\alpha \neq 0 : \ q(1, x) = -r + b(\lambda)^T x + x^T A(\lambda) x \geqslant 0$ for all $x$ in $\Re^n$. Thus $r - \lambda^T d \leqslant x^T A(\lambda) x + b(\lambda)^T x - \lambda^T d$

□

# Convexifying (QP) into a Semi-Definite Program

## Semi-Definite dual of (SD)

The Semi-Definite dual of

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

is

$$(SDP_{QP}) \quad \boxed{\begin{array}{lll} \min & \left\langle \begin{bmatrix} 0 & \frac{b^T}{2} \\ \frac{b}{2} & A_0 \end{bmatrix}, \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \right\rangle & \\ \text{s.t.} & \langle A_i, X \rangle + c_i^T x - d_i \leqslant 0 & \forall i \in \{1, \ldots, m\} \\ & \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0 & \end{array}}$$

We get the Basic Semi-Definite relaxation of QP !

# A generic Lagrangian scheme to obtain a Semi-Definite relaxation

## Decision variables can be formulated in a (QP)

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

1. Model the binary constraints as quadratic constraints: $x_j \in \{0,1\} \Leftrightarrow x_j^2 = x_j$
2. Add redundant quadratic constraints to close (as much as possible !) the gap before applying Lagrangian Duality.
3. Formulate the Lagrangian dual as a Semi-Definite program

This explains why linear constraints must be replaced by quadratic constraints in SDP relaxations: in our context *Lagrangian dual essentially ignores linear constraints*.

# The Quadratic Assignment Problem

## Model

- $x^T Cx = \sum_{i,j,k,l} C_{ijkl} x_{ij} x_{kl} = \langle C, xx^T \rangle$
- $Ax = e$ stands for $\sum_{i=1}^{n} x_{ij} = 1$, $\sum_{j=1}^{n} x_{ij} = 1$, $\forall i, j \in \{1, \ldots, n\}$

$$(QAP) \begin{cases} \min & \langle C, xx^T \rangle + c^T x \\ \text{s.t.} & Ax = e \\ & x \in \{0,1\}^{n^2} \end{cases}$$

## 20 years of Semi-Definite relaxations

- Solvers have improved a lot: interior-point methods, Bundle algorithms, augmented Lagrangian, low-rank factorization...
- Different stopping criteria to compute the bounds
- Various relaxations that turned out to be equivalent !

# Which redundant quadratic constraints should be added before dualizing ?

## Partial Lagrangian dual

Let $\mathcal{K}$ be any set of redundant quadratic constraints made from $Ax = e$

$(DP_K)$ :

$$\sup_{\omega,\mu} \inf_{x:Ax=e} x^T \left( C + \text{diag}(\mu) + \sum_{j\in K} \omega_j H_j \right) x + \left( \sum_{j\in K} \omega_k h_j + c - \mu \right)^T x + \omega^T \alpha$$

Lagrangian vectors: $\mu$ is associated to $x_{ij}^2 - x_{ij} = 0$ and $\omega$ is associated to the quadratic functions $q_j(x) = x^T H_j x + h_j^T x + \alpha_j$ of $\mathcal{K}$

## Proposition

*For any set $\mathcal{K}$, the partial Lagrangian dual $(DP_K)$ where $Ax = e$ is not dualized provides a better bound than $(SDP_K)$ the corresponding Semi-Definite relaxation.*

# Squared and Product constraints achieve the bound of the Partial Lagrangian Dual

## Theorem (valid for 0-1 quadratic programs)

*By setting* $\mathcal{K} = \left\{ x^T A^T A x - 2e^T A x + e^T e \right\}$,
$(DP_K)$ *is equivalent to* $(SDP_K)$

## For (QAP), it reduces to a single constraint !

One has $\sum_{i,j,k \neq j} (x_{ij}x_{ik} + x_{ji}x_{ki}) - \sum_{i,j} x_{ij} + n = 0$
Moreover $A^T A \succcurlyeq 0$ and $X - xx^T \succcurlyeq 0$ imply $\langle A^T A, X - xx^T \rangle \geqslant 0$
and thus $\langle A^T A, X - xx^T \rangle + (Ax - e)^2 = 0$ implies $Ax = e$

## Theorem (valid for any quadratic program)

*By setting* $\mathcal{K} = \{\langle A_r, x \rangle x_{kl} = x_{kl};\ r = 1, \ldots, 2n;\ k, l = 1, \ldots, n\}$,
$(DP_K)$ *is equivalent to* $(SDP_K)$

## For (QAP), we get $O(n^3)$ constraints !

We obtain $\sum_{j=1}^{n} x_{ij}x_{kl} - x_{kl} = 0 \quad \forall i, k, l$, and $\sum_{i=1}^{n} x_{ij}x_{kl} - x_{kl} = 0 \quad \forall j, k, l$, and we have to keep $Ax = e$ !

# Two Equivalent Semi-Definite Relaxations for the QAP

$(QAP_{\mathcal{P}})$

$$
\begin{array}{lll}
\min & \langle C, X\rangle + c^T x & \\
\text{s.t.} & \sum_{i=1}^n x_{ij} = 1 & j = 1,\dots,n \\
& \sum_{j=1}^n x_{ij} = 1 & i = 1,\dots,n \\
& \sum_{i=1}^n X_{ijkl} = x_{kl} & j,k,l = 1,\dots,n \\
& \sum_{j=1}^n X_{ijkl} = x_{kl} & i,k,l = 1,\dots,n \\
& X_{ijij} = x_{ij} & i,j = 1,\dots,n \\
& X_{ijkl} \geqslant 0 & i,j,k,l = 1,\dots,n \\
& \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0 &
\end{array}
$$

$(QAP_{\mathcal{C}})$

$$
\begin{array}{lll}
\min & \langle C, X\rangle + c^T x & \\
\text{s.t.} & \sum_{i,j,k\neq j} (X_{ijik} + X_{jiki}) & \\
& - \sum_{i,j} x_{ij} + n = 0 & \\
& X_{ijij} = x_{ij} & i,j = 1,\dots,n \\
& X_{ijkl} \geqslant 0 & i,j,k,l = 1,\dots,n \\
& \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0 &
\end{array}
$$

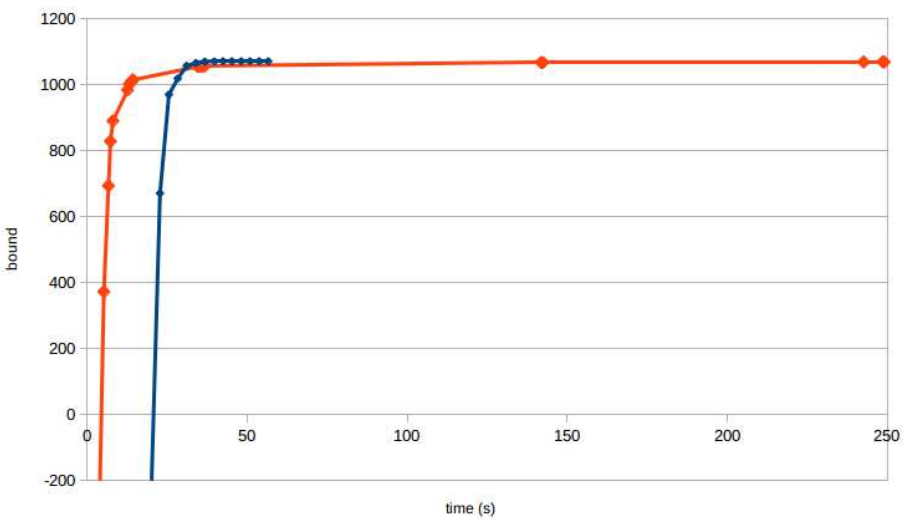
# Numerical Behaviour

## Choosing the relaxation and the solver

- CSDP : interior-point algorithm
- BiqCrunch : Quasi-Newton, sequel of regularized semidefinite programs

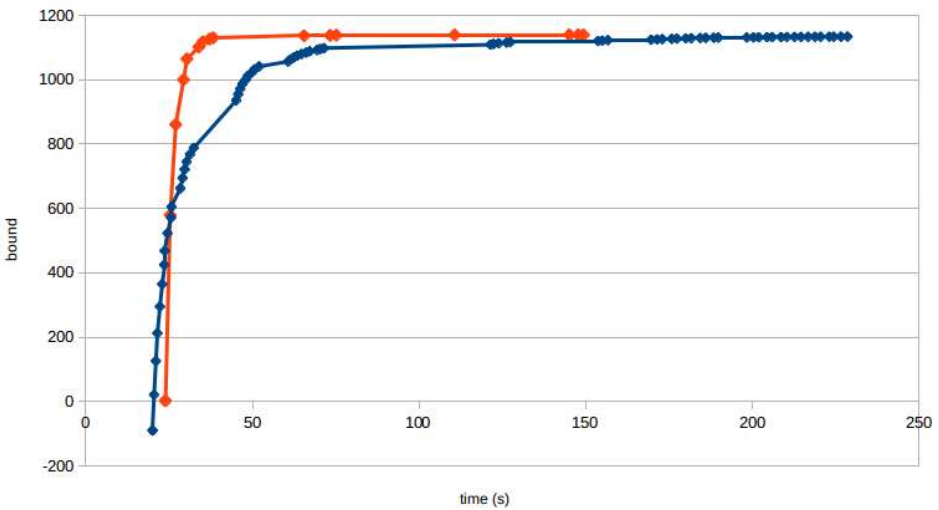
## Convergence curves for nug15 (225 decision variables)

CSDP and BiqCrunch




($QAP_C$) (squared constraint)
with *some* non-negativity constraints

BiqCrunch




($QAP_P$) (product constraints)
($QAP_C$) (squared constraints)
with *all* non-negativity constraints

# Semidefinite Relaxations for solving to optimality ?

## Semi-Definite relaxations in a Branch-and-Bound or a Cutting-plane algorithm

- Designing SDP from existing LP relaxations is easy: intersect semi-Definite and polyhedral relaxations
- Many commercial and free SDP solvers are available
- Since SDP provides stronger bounds than LP, should we use them for solving MINLP to optimality ?

## Some specific questions

$\longrightarrow$ Can we solve SDPs with a large number of constraints (e.g. cuts) ?

$\longrightarrow$ When adding cuts or fixing variables, how can one deal with an empty interior feasible set ? Moreover, how can one warm-start a children node after evaluation in the search tree ?

$\longrightarrow$ Can we build an integer solution from the SDP relaxation into a reasonable time ?

$\longrightarrow$ Do we always need to use a SDP-quality bound to prune a node ? How can we know in advance if a weaker bound is enough (and save time !) ?

# Semi-Definite Relaxations

## Basic Semi-Definite relaxation

(SDP)

$$\begin{array}{lll} \text{maximize} & \langle Q, X\rangle & \\ \text{subject to} & \langle A_i, X\rangle \leqslant a_i, & i \in \{1,\dots,m_I\} \\ & \langle B_i, X\rangle = b_i, & i \in \{1,\dots,m_E\} \\ & X \succcurlyeq 0 & \end{array}$$

## Enhancing the Semi-Definite bound

There are $4\binom{n}{3}$ triangle inequalities (for $1 \leqslant i < j < k \leqslant n$):

$$X_{ij} + X_{ik} + X_{jk} \geqslant -1, \quad -X_{ij} + X_{ik} - X_{jk} \geqslant -1$$
$$X_{ij} - X_{ik} - X_{jk} \geqslant -1, \quad -X_{ij} - X_{ik} + X_{jk} \geqslant -1$$

Choosing a subset $\mathbb{I}$ of the inequalities, we have

($\text{SDP}_{\mathbb{I}}$)

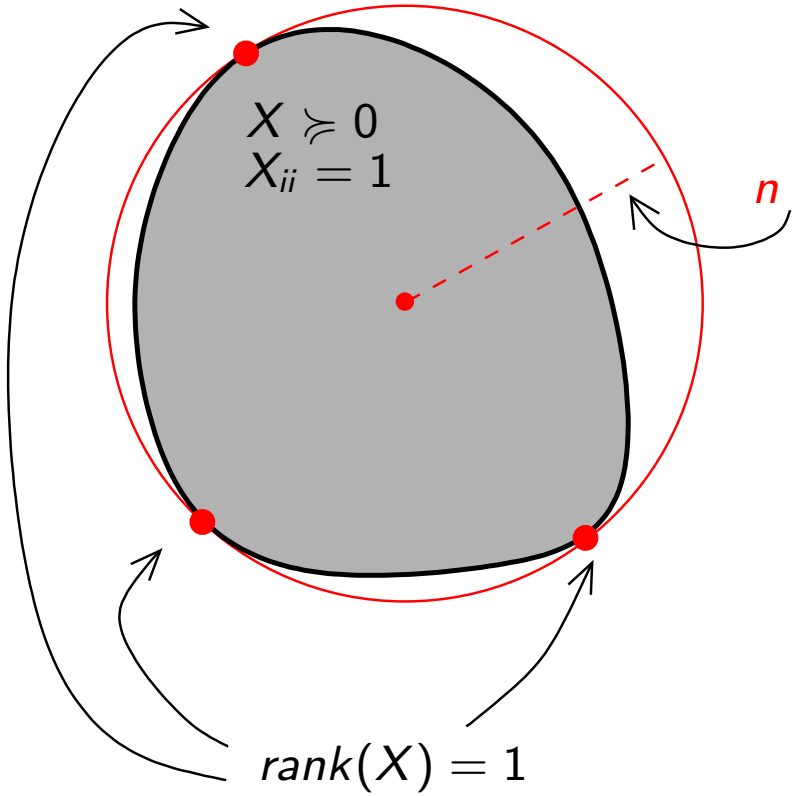
$$\begin{array}{lll} \text{maximize} & \langle Q, X\rangle & \\ \text{subject to} & \langle A_i, X\rangle \leqslant a_i, & i \in \{1,\dots,m_I\} \\ & \langle B_i, X\rangle = b_i, & i \in \{1,\dots,m_E\} \\ & \mathcal{A}_{\mathbb{I}}(X) \geqslant -e & \\ & X \succcurlyeq 0 & \end{array}$$

$\longrightarrow$ How can we solve this large SDP ?

# Non-Linear formulation of the rank constraint

**Lemma**

*Let* $X \succcurlyeq 0$ *be such that* $X_{ii} = 1$, *then*
$\|X\| \leqslant n$ *and*
$\|X\| = n \iff \text{rank}(X) = 1$




"Spherical" constraint : $\sum_{i,j} X_{ij}^2 = n^2$

$\longrightarrow$ non-convex quadratic constraint

# Back to Non-Linear Programming

## Continuous Non-Linear Reformulation of MINLP

(NLP)

$$
\begin{array}{lll}
\max & \langle Q, X \rangle & \\
\text{s.t.} & \langle A_i, X \rangle \leqslant a_i, & i \in \{1, \ldots, m_I\} \\
& \langle B_i, X \rangle = b_i, & i \in \{1, \ldots, m_E\} \\
& \mathcal{A}_{\mathbb{I}}(X) \geqslant -e & \\
& \sum_{i,j} X_{ij}^2 = n^2 & \\
& X \succcurlyeq 0 &
\end{array}
$$

## Apply Lagrangian duality to the unique "hard" constraint ?

- The Lagrangian is $\langle Q, X \rangle + \mu \left( n^2 - \sum_{i,j} X_{ij}^2 \right)$
- For $\mu < 0$ : hard to maximize !
- For $\mu = 0$ : standard SDP...
- For $\mu > 0$ : how tight is the bound ?

# A family of bounds with adjustable tightness

## Lemma

$$\text{diag}(X) = e,\ X \succcurlyeq 0,\ \text{rank}(X) = 1 \quad \Longrightarrow \quad \|X\|_F^2 = \sum_{i,j} X_{ij}^2 = n^2$$

## A set of Quadratic Semi-Definite Programs

$$(\text{SDP}_{\mathbb{I}}^{\alpha}) \qquad \begin{array}{lll} \max & \langle Q, X\rangle + \frac{\alpha}{2}\left(n^2 - \|X\|_F^2\right) & \\ \text{s.t.} & \langle A_i, X\rangle \leqslant a_i, & i \in \{1,\ldots,m_I\} \\ & \langle B_i, X\rangle = b_i, & i \in \{1,\ldots,m_E\} \\ & \mathcal{A}_{\mathbb{I}}(X) \geqslant -e & \\ & X \succcurlyeq 0 & \end{array}$$

- now $\alpha$ is a parameter, (no longer a Lagrange multiplier)
- for $\alpha \geqslant 0$: $(\text{SDP}_{\mathbb{I}}) \leqslant (\text{SDP}_{\mathbb{I}}^{\alpha})$
- a smaller $\alpha$ gives tighter upper bounds: $\alpha < \alpha' \implies (\text{SDP}_{\mathbb{I}}^{\alpha}) < (\text{SDP}_{\mathbb{I}}^{\alpha'})$
- $\lim_{\alpha\to 0}(\text{SDP}_{\mathbb{I}}^{\alpha}) = (\text{SDP}_{\mathbb{I}})$
- Warm-start : For a given $\mathbb{I}$ , all these Semi-Definite programs have the same feasible set

# A family of bounds with adjustable tightness

## A Quadratic Semi-Definite Program

$$(\mathrm{SDP}_{\mathbb{I}}^{\alpha}) \qquad \begin{array}{lll} \text{maximize} & \langle Q, X\rangle + \frac{\alpha}{2}\left(n^2 - \|X\|_F^2\right) & \\ \text{subject to} & \langle A_i, X\rangle \leqslant a_i, & i \in \{1,\ldots,m_I\} \\ & \langle B_i, X\rangle = b_i, & i \in \{1,\ldots,m_E\} \\ & \mathcal{A}_{\mathbb{I}}(X) \geqslant -e & \\ & X \succcurlyeq 0 & \end{array}$$

## Theorem (Expliciting the dual function)

*Define $[M]_+$ projection of $M \in \mathcal{S}^n$ onto the cone $\mathcal{S}_+^n$. The dual of $(SDP_{\mathbb{I}}^{\alpha})$ is*

$$(DSDP_{\mathbb{I}}^{\alpha}) \qquad \begin{array}{ll} \textit{minimize} & F_{\mathbb{I}}^{\alpha}(\lambda,\mu,\nu) \\ \textit{subject to} & \lambda \geqslant 0,\ \nu \geqslant 0 \end{array}$$

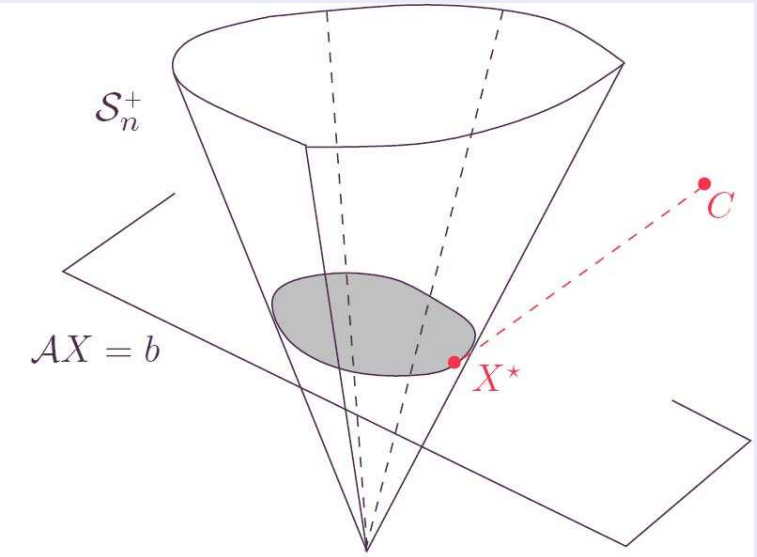
*with* $F_{\mathbb{I}}^{\alpha}(\lambda,\mu,\nu) = \frac{1}{2\alpha}\| \left[Q - \mathcal{A}^*(\lambda) - \mathcal{B}^*(\mu) + \mathcal{A}_{\mathbb{I}}^*(\nu)\right]_+ \|^2 + a^T\lambda + b^T\mu + e^T\nu + \frac{\alpha}{2}n^2$

# Expliciting the dual function

## Dual of $(SDP_{\mathbb{I}}^{\alpha})$

For $C \in \mathcal{S}^n$, define $[C]_+$
projection of $C$ onto the cone $\mathcal{S}_+^n$

$$\max_{X \succcurlyeq 0} \left\{ \langle C, X \rangle - \frac{\alpha}{2} \|X\|_F^2 \right\}$$
$$= \frac{1}{2\alpha} \|[C]_+\|_F^2$$




- To compute the projection : partial eigenvalue decomposition

$$M^T \begin{bmatrix} \lambda_1 & \dots & 0 \\ & \ddots & \\ 0 & & 0 \\ 0 & \dots & \lambda_n \end{bmatrix} M \to M^T \begin{bmatrix} \lambda_1 & \dots & 0 \\ & \ddots & \\ 0 & & 0 \\ 0 & \dots & 0 \end{bmatrix} M$$

(standard libraries : LAPACK, MKL,...)

# A family of bounds with adjustable tightness

## Dual function $F_{\mathbb{I}}^{\alpha}(\lambda, \mu, \nu)$

- The Lagrangian is ($\lambda \in \mathbb{R}_{+}^{m_I}, \mu \in \mathbb{R}^{m_E}, \nu \in \mathbb{R}_{+}^{|\mathbb{I}|}$):

$$\mathcal{L}(X; \lambda, \mu, \nu) := \langle Q - \mathcal{A}^*(\lambda) - \mathcal{B}^*(\mu) + \mathcal{A}_{\mathbb{I}}^*(\nu), X \rangle - \frac{\alpha}{2}\|X\|_F^2 + a^T\lambda + b^T\mu + e^T\nu + \frac{\alpha}{2}n^2$$

- The dual function is: $\max_{X \succcurlyeq 0} \left\{ \langle M, X \rangle - \frac{\alpha}{2}\|X\|_F^2 \right\} = \frac{1}{2\alpha}\|[M]_+\|_F^2$

$$\begin{aligned} F_{\mathbb{I}}^{\alpha}(\lambda, \mu, \nu) &:= \max_{X \succcurlyeq 0} \mathcal{L}(X; \lambda, \mu, \nu) \\ &= \frac{1}{2\alpha} \left| \left[ Q - \mathcal{A}^*(\lambda) - \mathcal{B}^*(\mu) + \mathcal{A}_{\mathbb{I}}^*(\nu) \right]_+ \right|_F^2 \\ &\quad + a^T\lambda + b^T\mu + e^T\nu + \frac{\alpha}{2}n^2 \end{aligned}$$

# A family of bounds with adjustable tightness

## Dual program : computing the best bound $F_{\mathbb{I}}^{\alpha}(\lambda, \mu, \nu)$

$$(\mathrm{DSDP}_{\mathbb{I}}^{\alpha}) \qquad \begin{array}{ll} \text{minimize} & F_{\mathbb{I}}^{\alpha}(\lambda, \mu, \nu) \\ \text{subject to} & \lambda \geqslant 0,\ \nu \geqslant 0 \end{array}$$

## $(\mathrm{DSDP}_{\mathbb{I}}^{\alpha})$ is a smooth convex optimization problem

$$\nabla_{\lambda} F_{\mathbb{I}}^{\alpha}(\lambda, \mu, \nu) = a - \mathcal{A}\left(\frac{1}{\alpha}\left[Q - \mathcal{A}^{*}(\lambda) - \mathcal{B}^{*}(\mu) + \mathcal{A}_{\mathbb{I}}^{*}(\nu)\right]_{+}\right)$$

$$\nabla_{\mu} F_{\mathbb{I}}^{\alpha}(\lambda, \mu, \nu) = b - \mathcal{B}\left(\frac{1}{\alpha}\left[Q - \mathcal{A}^{*}(\lambda) - \mathcal{B}^{*}(\mu) + \mathcal{A}_{\mathbb{I}}^{*}(\nu)\right]_{+}\right)$$

$$\nabla_{\nu} F_{\mathbb{I}}^{\alpha}(\lambda, \mu, \nu) = e + \mathcal{A}_{\mathbb{I}}\left(\frac{1}{\alpha}\left[Q - \mathcal{A}^{*}(\lambda) - \mathcal{B}^{*}(\mu) + \mathcal{A}_{\mathbb{I}}^{*}(\nu)\right]_{+}\right)$$

- can be solved with a quasi-Newton method like L-BFGS-B
- harder to solve for very small $\alpha$ (SDP quality !)
- but fast to solve for larger values of $\alpha$

# Using the Bounds in a B&B : BiqCrunch

## B&B context

To prune nodes one does not always need to use SDP quality bounds (i.e. small values $\alpha$).
Adjust tightness: by decreasing gradually $\alpha$ we can obtain a faster algorithm.

To build feasible solutions at no additional cost , we can use the eigenvectors (projection) and apply the max-cut technique (random hyperplanes).

## Improved semidefinite bounding procedure

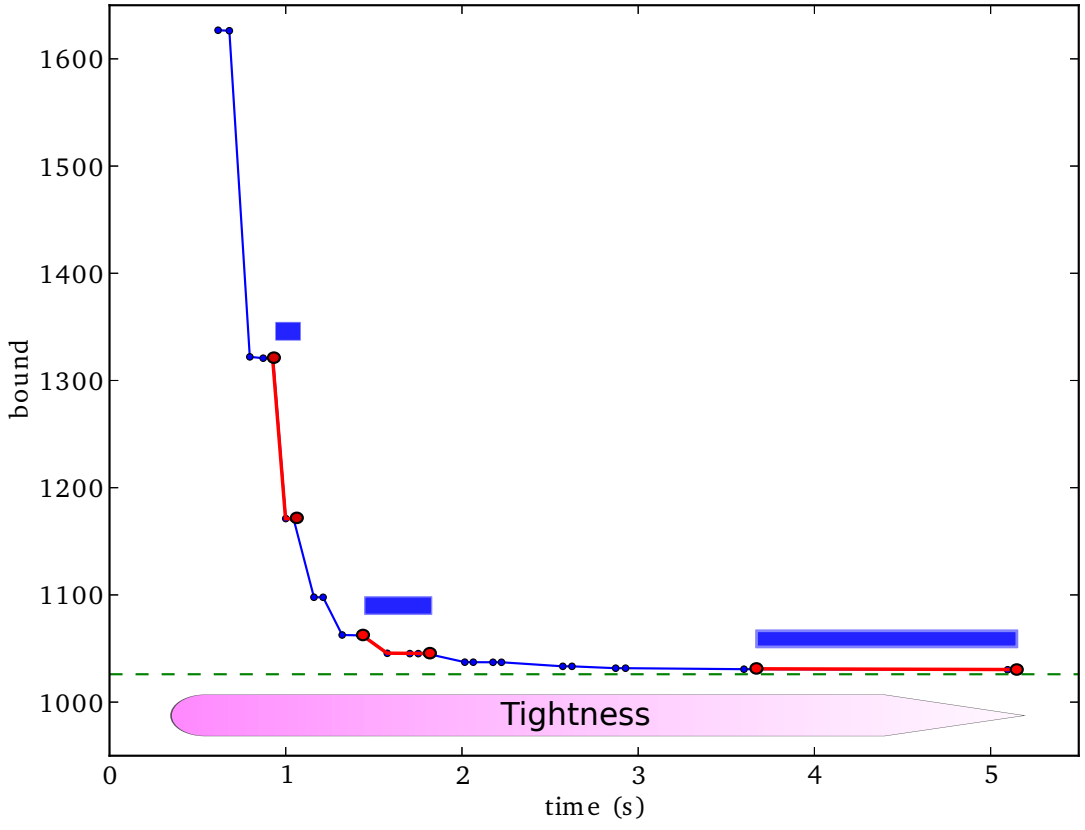
While *(the node can not be pruned)* and *($\alpha$ is not too small)* do:

1. Solve (DSDP$^{\alpha^k}_{\mathbb{I}^k}$) with L-BFGS-B. Warm start : use the optimal solution of (DSDP$^{\alpha^{k-1}}_{\mathbb{I}^{k-1}}$)
2. Update cuts, $\alpha$, and L-BFGS-B tolerance $\varepsilon$:
   - Add / remove triangle inequalities
   - if very few cuts are found then reduce $\alpha$ and $\varepsilon$

# Bounding Procedure: a sequel of cutting planes algorithms

## Duality : the process can be interrupted at any time (valid bound)

- The node can be pruned.
- We "give up": there is no hope to prune.
- $\alpha$ is too small: BFGS iterations are too numerous.



# Example (LP format)

```
$ cat example.lp

Maximize
  20 z1*z3 + 26 z1*z4 + 23 z2*z3 +
   8 z2*z5 + 32 z3*z4 + 13 z4*z5

Subject to
  z1 + z2 + z3 + z4 + z5 = 3

  12 z1*z3 + 24 z1*z4 + 14 z2*z3 +
  16 z2*z5 + 28 z3*z4 + 12 z4*z5 <= 30

Binary
  z1 z2 z3 z4 z5

End
```

# Example

```
$ problems/generic/biqcrunch
Usage : biqcrunch [-v (0|1)] file.bc file.param

$ problems/generic/biqcrunch -v 1 example.bc biq_crunch.param
Output file: example.bc.output
Input file:  example.bc
Parameter file: biq_crunch.param
Nodes = 7; Root node bound = 47.44
Maximum value = 43
Solution = { 1 2 3 }
CPU time = 0.0121 s
```

# Example

```
$ cat example.bc.output
...
*****************************************************************************
                                    Node 1
*****************************************************************************
Problem size 5
=============================================================================
Ite    gap     alpha     tol    nbit   Enorm    Inorm    minCuts NCuts  NSub NAdd
=============================================================================
  1    4.582  1.0e-01  1.0e-01   24  5.1e-02  0.0e+00  -2.5e-01     0  -0   +0
  2    4.574  1.0e-01  1.0e-01    2  4.5e-02  4.9e-02  -2.8e-01     0  -0   +15
  3    4.458  5.0e-02  9.5e-02    6  5.2e-02  3.8e-02  -8.7e-02    15  -0   +4
  4    4.452  2.5e-02  9.0e-02    5  5.1e-02  3.2e-02  -1.5e-01    19  -0   +20
  5    4.452  1.3e-02  8.6e-02    4  2.3e-02  6.8e-02  -6.8e-02    39  -0   +0
  6    4.444  6.3e-03  8.1e-02    8  2.9e-02  8.0e-02  -8.0e-02    39  -0   +0
...
 12    4.444  9.8e-05  6.0e-02    7  1.5e-02  2.7e-02  -2.7e-02    39  -0   +0
 13    4.444  5.0e-05  5.7e-02    7  1.5e-02  2.7e-02  -2.7e-02    39  -0   +0
=============================================================================
Bound = 47.44, BFGS = 13, alpha = 5.0e-05, tol = 5.7e-02, cuts =   39, time = 0.0
...
```

# Solving exactly the Quadratic Assignment Problem

## The strongest Semi-Definite relaxation can be used

$$(QAP_{\mathcal{P}}) \begin{cases} \min & \langle C, X \rangle + c^T x & \\ \text{s.t.} & \sum_{i=1}^n x_{ij} = 1 & j = 1, \ldots, n \\ & \sum_{j=1}^n x_{ij} = 1 & i = 1, \ldots, n \\ & \sum_{i=1}^n X_{ijkl} = x_{kl} & \forall j, k, l = 1, \ldots, n \\ & \sum_{j=1}^n X_{ijkl} = x_{kl} & \forall i, k, l = 1, \ldots, n \\ & X_{ijkl} \geqslant 0 & \forall i, j, k, l = 1, \ldots, n \\ & X_{ijij} = x_{ij} & i, j = 1, \ldots, n \\ & \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succcurlyeq 0 & \end{cases}$$

| | nodes | time(s) |
|---|---|---|
| nug12 | 11 | 63 |
| nug14 | 5 | 181 |
| nug15 | 15 | 303 |
| nug17 | 63 | 2487 |

| | nodes | time(s) |
|---|---|---|
| had16 | 1 | 42 |
| had18 | 3 | 426 |
| had20 | 3 | 741 |
| chr20 | 3 | 709 |
| chr22a | 3 | 1903 |

| | nodes | time(s) |
|---|---|---|
| rou12 | 25 | 52 |
| rou15 | 55 | 518 |
| scr15 | 57 | 244 |
| lipa20a | 1 | 146 |
| lipa20b | 1 | 36 |

BiqCrunch DELL T-1600 Intel Xeon E3-1270 3.40GHz, using single core

# Solving exactly the Max-Independent Set Problem

## Semi-Definite Relaxation

$$\begin{array}{lll} \max & \sum_{i=1}^{n} x_i & \\ \text{s.t.:} & X_{ij} = 0 & (i,j) \in E \\ & X_{ii} = x_i & i \in \{1,\ldots,n\} \\ & \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succcurlyeq 0 & \end{array}$$

Recall that triangle inequalities (e.g. $X_{ij} \geqslant 0$) are added/removed dynamically

| | nodes | time(s) |
|---|---|---|
| keller4(n=171) | 155 | 158 |
| brock200_1 | 2969 | 3006 |
| brock200_2 | 51 | 78 |
| brock200_3 | 105 | 167 |
| brock200_4 | 193 | 230 |

| | nodes | time(s) |
|---|---|---|
| san200_0.7_1 | 1 | 2.1 |
| san200_0.7_2 | 1 | 3.8 |
| san200_0.9_1 | 1 | 0.9 |
| san200_0.9_3 | 1 | 4.4 |

BiqCrunch DELL T-1600 Intel Xeon E3-1270 3.40GHz, using single core

# Co-Positive Optimization

Idea : stronger relaxations using a smaller cone than $S_n^+$

## $C_n$ and $Co - P$

- The cone of completely positive matrices is
  $C_n = \{X \in S_n \,|\, X = \sum_{i=1}^{k} y_i y_i^T \text{where } y_i \geqslant 0\}$
- Bad news: to determine if a matrix $X$ belongs to $C_n$ is co-NP-hard
- The dual cone of $C_n$ is the set of Co-Positive matrices
  $C_n^* = \{X \in S_n | y^T X y \geqslant 0 \ \forall y \geqslant 0\}$

## A convex formulation of NP-hard problems

$(P) : \min \langle Cx, x \rangle + \langle b, x \rangle \text{ s.t. } Ax = b,\ x \in \{0,1\}^n$ is equivalent to

$$
\begin{array}{lll}
\min & \langle C, X \rangle + \langle b, x \rangle & \\
\text{s.t.} & Ax = b & \\
 & X_{jj} = x_j & j = 1, \ldots, n \\
 & \langle A_i A_i^T, X \rangle = b_i^2 & i = 1, \ldots, m \\
 & \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \in C_{n+1} &
\end{array}
$$

# Strengthening the SDP bound : Lift-and-project

## Generalizing Linear Programming approaches

Feasible set of a discrete problem:

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

Feasible set of a semidefinite relaxation (hyp. : $0 \leqslant x_i \leqslant 1 \ \forall i$)

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

$\longrightarrow$ $\sum_{i=1}^{m} x_j x_i A_i - x_j A_0 \succcurlyeq 0$ and $\sum_{i=1}^{m} (1-x_j) x_i A_i - (1-x_j) A_0 \succcurlyeq 0$

$\longrightarrow$ Linearization

$$\begin{array}{|c|l|} \hline \sum_{i \neq j}^{m} y_j A_i + x_j (A_j - A_0) & \succcurlyeq 0 \\ \sum_{i=1}^{m} x_i A_i - x_j (A_j - A_0) - \sum_{i \neq j}^{m} y_j A_i - A_0 & \succcurlyeq 0 \\ \hline \end{array}$$

$\longrightarrow$ Projection $\Pi_j$ to get $x$ from $(x, y)$ that verifies the previous Linear matrix inequalities.

- The convex hull of $S \cap \{0,1\}^n$ can be sequentially generated by the projections $\Pi_j$.
- The set of valid inequalities is described by Linear Matrix Inequalities: to find the deepest cut $\leftrightarrow$ solve a SDP !

# Polynomial Semi-Definite Programs

## Polynomial Optimization

$$(PO) \quad \boxed{\min P_0(x) \text{ s.t. } P_k(x) \geqslant 0 \ k \in \{1,\ldots,m\}}$$

$r_k = \max_j \lceil \frac{\deg(P_j)}{2} \rceil - \frac{\deg(P_k)}{2}$ $\qquad q_r(x) = (1, x_1, x_2, \ldots, x_n, x_1^2, x_1x_2, \ldots, x_n^r)^T$

$$(P-SDP) \quad \boxed{\min P_0(x) \text{ s.t. } q_{r_k} q_{r_k}^T P_k(x) \succcurlyeq 0 \ k \in \{1,\ldots,m\}, \ q_r q_r^T \succcurlyeq 0}$$

## $(P-SDP)$ can be linearized into a linear SDP (relaxation)

- Consider $M_r(x) = (x_{I\cup J})_{I,J\in P_r(N)}$, where $P_r(N) = \{I \subseteq N, |I| \leqslant r\}$. The lines and columns of $M_r(x)$ are indexed by the monomial in $x$ of order at most $r$. $O(n^{2r})$ variables ! Linearization:
- $$\begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1x_2 \\ x_2^2 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1x_2 \\ x_2^2 \end{pmatrix}^T \succcurlyeq 0 \rightarrow \begin{bmatrix} 1 & x_{10} & x_{01} & x_{20} & x_{11} & x_{02} \\ x_{10} & x_{20} & x_{11} & x_{30} & x_{21} & x_{12} \\ x_{01} & x_{11} & x_{02} & x_{21} & x_{12} & x_{03} \\ x_{20} & x_{30} & x_{21} & x_{40} & x_{31} & x_{22} \\ x_{11} & x_{21} & x_{12} & x_{31} & x_{22} & x_{13} \\ x_{02} & x_{12} & x_{03} & x_{22} & x_{13} & x_{04} \end{bmatrix} \succcurlyeq 0$$

  To get a Linear SDP: expand the inequalities in $(P-SDP)$ and replace each monomial by the corresponding variable