The Impact of Quadratization in Convexification-Based Resolution of Polynomial Binary Optimization

Sourour Elloumi[1,2], Amélie Lambert[2], Arnaud Lazare[1,2], Elisabeth Rodríguez-Heck[3]

[1]UMA Ensta Paris-Tech, [2]Cedric-CNAM, [3]RWTH Aachen University

December 4, PGMO Days 2019
EDF'Lab Palaiseau, France

The problem

We are interested in solving the following problem:

$$\begin{aligned} \min \ & f(x) \\ \text{s. t. } & x \in \{0,1\}^n \end{aligned} \tag{P}$$

where f is a polynomial on n binary variables, and there are no additional constraints.

The problem

We are interested in solving the following problem:

$$\begin{aligned} \min\ & f(x) \\ \text{s. t. } & x \in \{0,1\}^n \end{aligned} \tag{P}$$

where f is a polynomial on n binary variables, and there are no additional constraints.

(P) is NP-hard, and the difficulties come from

- non-convexity of f
- integer variables

Several resolution methods for (P) are based on the idea of working in two phases:

- Phase 1: Define an equivalent linear or quadratic problem using auxiliary variables.
- Phase 2: Solve the (lower degree) reformulated problem.

Reformulation / relaxation based resolution methods

Several resolution methods for (P) are based on the idea of working in two phases:

- Phase 1: Define an equivalent linear or quadratic problem using auxiliary variables.
- Phase 2: Solve the (lower degree) reformulated problem.

Motivation: To draw benefit from literature, algorithms and software available for the reformulated problems.

Reformulation / relaxation based resolution methods

Several resolution methods for (P) are based on the idea of working in two phases:

- Phase 1: Define an equivalent linear or quadratic problem using auxiliary variables.
- Phase 2: Solve the (lower degree) reformulated problem using convexification techniques.

Motivation: To draw benefit from literature, algorithms and software available for the reformulated problems.

Two complementary approaches

Quadratic reformulations of nonlinear binary optimization problems

Phase 1: Quadratization
Carefully chosen

↓

Phase 2: Convexification:
Simple, Linearization

(Anthony, Boros, Crama, & Gruber, 2017)
(Boros, Crama, & Rodríguez-Heck, 2018)
(Rodríguez-Heck, 2018)

`PQCR`: Polynomial binary optimization through Quadratic Convex Reformulation

Phase 1: Quadratization
Simple algorithm

↓

Phase 2: Convexification:
Carefully chosen, tailored

(Elloumi, Lambert, & Lazare, 2019)
(Lazare, 2019)

Two complementary approaches

Quadratic reformulations of nonlinear binary optimization problems

Phase 1: Quadratization
Carefully chosen

↓

Phase 2: Convexification:
Simple, Linearization

(Anthony, Boros, Crama, & Gruber, 2017)
(Boros, Crama, & Rodríguez-Heck, 2018)
(Rodríguez-Heck, 2018)

PQCR: Polynomial binary optimization through Quadratic Convex Reformulation

Phase 1: Quadratization
Simple algorithm

↓

Phase 2: Convexification:
Carefully chosen, tailored

(Elloumi, Lambert, & Lazare, 2019)
(Lazare, 2019)

Quadratizations without constraints

Quadratization: definition and desirable properties

Definition (Anthony, Boros, Crama, & Gruber, 2017)

Given a polynomial $f(x)$ on $x \in \{0,1\}^n$, a *quadratization* $g(x,y)$ is a function satisfying

- g is quadratic
- $g(x,y)$ depends on the original variables x and on m auxiliary variables y
- satisfies

$$f(x) = \min\{g(x,y) : y \in \{0,1\}^m\} \quad \forall x \in \{0,1\}^n.$$

Quadratization: definition and desirable properties

Definition (Anthony, Boros, Crama, & Gruber, 2017)

Given a polynomial $f(x)$ on $x \in \{0,1\}^n$, a *quadratization* $g(x,y)$ is a function satisfying

- g is quadratic
- $g(x,y)$ depends on the original variables x and on m auxiliary variables y
- satisfies

$$f(x) = \min\{g(x,y) : y \in \{0,1\}^m\} \quad \forall x \in \{0,1\}^n.$$

- Which quadratizations are "good"?
  - Small number of auxiliary variables
  - Lead to relaxations with tight bound?

Quadratization: definition and desirable properties

Definition (Anthony, Boros, Crama, & Gruber, 2017)

Given a polynomial $f(x)$ on $x \in \{0,1\}^n$, a *quadratization* $g(x,y)$ is a function satisfying

- g is quadratic
- $g(x,y)$ depends on the original variables x and on m auxiliary variables y
- satisfies

$$f(x) = \min\{g(x,y) : y \in \{0,1\}^m\} \quad \forall x \in \{0,1\}^n.$$

- Which quadratizations are "good"?
  - Small number of auxiliary variables
  - Lead to relaxations with tight bound?
- Two main classes of approaches: termwise and non-termwise.

Termwise quadratizations

Example 1: Main idea

Quadratize monomial by monomial using disjoint sets of auxiliary variables.

$$f(x) = 2x_1 + 3x_2x_3 - 2x_2x_3x_4 + 3x_1x_2x_3x_4$$

Termwise quadratizations

Example 1: Main idea

Quadratize monomial by monomial using disjoint sets of auxiliary variables.

$$f(x) = 2x_1 + 3x_2x_3 - 2x_2x_3x_4 + 3x_1x_2x_3x_4$$

Negative monomial

(Kolmogorov & Zabih, 2004; Freedman & Drineas, 2005)

$$-\prod_{i=1}^{n} x_i = \min_{y \in \{0,1\}} -y(\sum_{i=1}^{n} x_i - (n-1))$$

- One variable is sufficient!

Termwise quadratizations

Example 1: Main idea

Quadratize monomial by monomial using disjoint sets of auxiliary variables.

$$f(x) = 2x_1 + 3x_2x_3 - 2x_2x_3x_4 + 3x_1x_2x_3x_4$$

Negative monomial

(Kolmogorov & Zabih, 2004; Freedman & Drineas, 2005)

$$-\prod_{i=1}^{n} x_i = \min_{y\in\{0,1\}} -y\left(\sum_{i=1}^{n} x_i - (n-1)\right)$$

- One variable is sufficient!

Positive monomial

(Boros, Crama, & Rodríguez-Heck, 2018): For $\ell = \lceil \log(n) \rceil$

$$\prod_{i=1}^{n} x_i = \min_{y\in\{0,1\}^{\ell-1}} \frac{1}{2}\left(\sum_{i=1}^{n} x_i - \sum_{i=1}^{\ell-1} 2^i y_i\right)\cdot \left(\sum_{i=1}^{n} x_i - \sum_{i=1}^{\ell-1} 2^i y_i - 1\right)$$

- Number of auxiliaries: $\lceil \log(n) \rceil - 1$.
- Proved to be smallest possible.

Non-Termwise: Rosenberg's quadratization

First quadratization method (Rosenberg, 1975)

1. Take a product $x_i x_j$ from a highest-degree monomial of f and substitute it by a new variable y_{ij}.
2. Add penalty $P(x_i x_j - 2x_i y_{ij} - 2x_j y_{ij} + 3y_{ij})$ (P large enough) to objective function to force $y_{ij} = x_i x_j$ at all optimal solutions.
3. Iterate until obtaining a quadratic function.

Example 1

$$f(x) = 2x_1 + 3x_2x_3 - 2x_2x_3x_4 + 3x_1x_2x_3x_4$$

Apply Rosenberg with $y_1 = x_2x_3$ and $y_2 = x_1x_4$. We obtain

$$\begin{aligned} g(x,y) = & 2x_1 + 3x_2x_3 - 2y_1x_4 + 3y_1y_2 + P(x_2x_3 - 2x_2y_1 - 2x_3y_1 + 3y_1) \\ & P(x_1x_4 - 2x_1y_2 - 2x_4y_2 + 3y_2) \end{aligned}$$

- Different substitution choices = different quadratizations (!)
- A substitution choice corresponds to a pairwise cover

Non-termwise quadratizations

(Anthony, Boros, Crama, & Gruber, 2017)

Definition: Pairwise cover or 2×2 quadratization schemes

- Let $\mathcal{M}$ be the set of monomials of polynomial f.
- A pairwise cover of $\mathcal{M}$ is a set of monomials $\mathcal{H}$ such that for each monomial $M \in \mathcal{M}$ of degree > 2, there exist two monomials $A(M), B(M) \in \mathcal{H}$ such that $|A(M)| < |M|$, $|B(M)| < |M|$ and $A(M) \cup B(M) = M$.

Example 1

$$f(x) = 2x_1 + 3x_2x_3 - 2x_2x_3x_4 + 3x_1x_2x_3x_4$$

Two different pairwise covers:

- $\mathcal{H}_1 = \{\{2,4\},\{3\},\{1,2\},\{3,4\}\}$
- $\mathcal{H}_2 = \{\{2,3\},\{1,2,3\},\{4\}\}$

Non-Termwise: ABCG quadratization

Theorem (Anthony, Boros, Crama, & Gruber, 2017)

Given f with set of monomials $\mathcal{M}$, and a pairwise cover $\mathcal{H}$ of $\mathcal{M}$ such that $\mathcal{H} \subset \mathcal{M}$, one can define a quadratization for f as follows

$$f(x) = \min_{y \in \{0,1\}^{|\mathcal{H}|}} \sum_{M \in \mathcal{M}} a_M y_{A(M)} y_{B(M)} + \sum_{H \in \mathcal{H}} b_H \left(y_H \left(|H| - \frac{1}{2} - \sum_{j \in H} x_j \right) + \frac{1}{2} \prod_{j \in H} x_j \right)$$

where $b_H = 0$ for $H \in \mathcal{M} \backslash \mathcal{H}$ and

$$\frac{1}{2} b_H = \sum_{\substack{M \in \mathcal{M} \\ H \in \{A(M), B(M)\}}} \left(|a_M| + \frac{1}{2} b_M \right)$$

- Different pairwise covers lead to different ABCG quadratizations.
- Similar to Rosenberg but with a different penalty (smaller coefficients).

Heuristics for small Pairwise Covers

Small pairwise covers

- Finding pairwise cover of smallest size (i.e., introducing smallest number of auxiliary variables) is NP-hard.

Heuristics for small Pairwise Covers

Small pairwise covers

- Finding pairwise cover of smallest size (i.e., introducing smallest number of auxiliary variables) is NP-hard.
- Three heuristics developed (Rodríguez-Heck, 2018)
  - PC1: Separate first two variables from the rest.
  - PC2: Most “popular” intersections first.
  - PC3: Most “popular” pairs of variables first.

Heuristics for small Pairwise Covers

Small pairwise covers

- Finding pairwise cover of smallest size (i.e., introducing smallest number of auxiliary variables) is NP-hard.
- Three heuristics developed (Rodríguez-Heck, 2018)
  - PC1: Separate first two variables from the rest.
  - PC2: Most “popular” intersections first.
  - PC3: Most “popular” pairs of variables first.
- Fourth heuristic developed (Lazare, 2019)
  - PC0: Sort monomials in lexicographical order + “greedy” heuristic.

Heuristics for small Pairwise Covers

Small pairwise covers

- Finding pairwise cover of smallest size (i.e., introducing smallest number of auxiliary variables) is NP-hard.
- Three heuristics developed (Rodríguez-Heck, 2018)
  - PC1: Separate first two variables from the rest.
  - PC2: Most "popular" intersections first.
  - PC3: Most "popular" pairs of variables first.
- Fourth heuristic developed (Lazare, 2019)
  - PC0: Sort monomials in lexicographical order + "greedy" heuristic.
- Main idea: identifying *subterms* that appear as subsets of one or more monomials *more often* in the input monomial set $\mathcal{M}$.

Computational results: LABS

- Instances from `http://polip.zib.de/autocorrelated_sequences/`
- Quadratization solved with CPLEX 12.7, time limit: 1h

Instance			Quadratization + CPLEX					
			Non-Termwise				Termwise	
Name	n	m	N	PC1	PC2	PC3	N	logn-1
b.20.5	20	207	90	10.58	5.05	4.27	137	35.34
b.20.10	20	833	155	90.28	159.47	137.69	698	365.47
b.25.6	25	407	135	106.67	80.17	121.03	297	466.92
b.25.13	25	1782	247	2311.09	> 3600	> 3600	1560	> 3600
b.30.4	30	223	114	13.52	7.17	7.03	139	36.08
b.35.4	35	263	134	24.13	13.25	11.2	164	54.14

- Non-Termwise always better.
- These instances have a very particular structure (and are all of degree 4).

Quadratizations with constraints

Non-Termwise: Rosenberg with constraints

First quadratization method (Rosenberg, 1975)

1. Take a product $x_i x_j$ from a highest-degree monomial of f and substitute it by a new variable y_{ij}.
2. Add **penalty** $P(x_i x_j - 2x_i y_{ij} - 2x_j y_{ij} + 3y_{ij})$ **to objective function** to force $y_{ij} = x_i x_j$ at all optimal solutions.
3. Iterate until obtaining a quadratic function.

Example 1

$$f(x) = 2x_1 + 3x_2x_3 - 2x_2x_3x_4 + 3x_1x_2x_3x_4$$

Apply Rosenberg with $y_1 = x_2x_3$ and $y_2 = x_1x_4$. We obtain

$$g(x, y) = 2x_1 + 3x_2x_3 - 2y_1x_4 + 3y_1y_2 + P(x_2x_3 - 2x_2y_1 - 2x_3y_1 + 3y_1)$$
$$P(x_1x_4 - 2x_1y_2 - 2x_4y_2 + 3y_2)$$

- Different substitution choices = different quadratizations (!)
- A substitution choice corresponds to a pairwise cover

Non-Termwise: Rosenberg with constraints

First quadratization method (Rosenberg, 1975)

1. Take a product $x_i x_j$ from a highest-degree monomial of f and substitute it by a new variable y_{ij}.
2. Add constraints to force $y_{ij} = x_i x_j$ at all optimal solutions.
3. Iterate until obtaining a quadratic function.

Example 1

$$f(x) = 2x_1 + 3x_2x_3 - 2x_2x_3x_4 + 3x_1x_2x_3x_4$$

Apply Rosenberg with $y_1 = x_2x_3$ and $y_2 = x_1x_4$. We obtain

$$g(x,y) = 2x_1 + 3x_2x_3 - 2y_1x_4 + 3y_1y_2 + P(x_2x_3 - 2x_2y_1 - 2x_3y_1 + 3y_1)$$
$$P(x_1x_4 - 2x_1y_2 - 2x_4y_2 + 3y_2)$$

Different substitution choices = different quadratizations (!)

A substitution choice corresponds to a pairwise cover

Non-Termwise: Rosenberg with constraints

First quadratization method (Rosenberg, 1975)

1. Take a product $x_i x_j$ from a highest-degree monomial of f and substitute it by a new variable y_{ij}.
2. Add constraints to force $y_{ij} = x_i x_j$ at all optimal solutions.
3. Iterate until obtaining a quadratic function.

Example 1

$$f(x) = 2x_1 + 3x_2x_3 - 2x_2x_3x_4 + 3x_1x_2x_3x_4$$

Apply Rosenberg with $y_1 = x_2x_3$
and $y_2 = x_1x_4$

$$\begin{aligned} \min \;\; & g(x,y) = 2x_1 + 3x_2x_3 - 2y_1x_4 + 3y_1y_2 \\ \text{s. t.} \;\; & y_1 = x_2x_3 \\ & y_2 = x_1x_4 \\ & x_1, x_2, x_3, x_4, y_1, y_2 \in \{0,1\} \end{aligned}$$

Non-Termwise: Rosenberg with constraints

First quadratization method (Rosenberg, 1975)

1. Take a product $x_i x_j$ from a highest-degree monomial of f and substitute it by a new variable y_{ij}.
2. Add constraints to force $y_{ij} = x_i x_j$ at all optimal solutions.
3. Iterate until obtaining a quadratic function.

Example 1

$$f(x) = 2x_1 + 3x_2x_3 - 2x_2x_3x_4 + 3x_1x_2x_3x_4$$

Apply Rosenberg with $y_1 = x_2x_3$ and $y_2 = x_1x_4$

$$\begin{aligned} \min \quad & g(x,y) = 2x_1 + 3x_2x_3 - 2y_1x_4 + 3y_1y_2 \\ \text{s. t.} \quad & y_1 = x_2x_3 \\ & y_2 = x_1x_4 \\ & x_1, x_2, x_3, x_4, y_1, y_2 \in \{0,1\} \end{aligned}$$

Instead of $y_{ij} = x_i x_j$, use:

$$\begin{aligned} y_{ij} &\leq x_i \\ y_{ij} &\leq x_j \\ y_{ij} &\geq x_i + x_j - 1 \\ y_{ij} &\geq 0 \end{aligned}$$

Non-Termwise: ABCG quadratization with constraints

(With constraints) ABCG = Rosenberg

- Given an appropriate pairwise cover $\mathcal{H}$ of $\mathcal{M}$, the only difference between Rosenberg's and ABCG quadratization is the penalty term.

Non-Termwise: ABCG quadratization with constraints

(With constraints) ABCG = Rosenberg

- Given an appropriate pairwise cover $\mathcal{H}$ of $\mathcal{M}$, the only difference between Rosenberg's and ABCG quadratization is the penalty term.
- Hence, when using constraints instead of penalties, both methods lead to the same quadratization.

RWTH AACHEN UNIVERSITY

Termwise with constraints?

Not easy to derive a quadratization with constraints

- Quadratization for the positive monomial ($\ell = \lceil \log(n) \rceil$):

$$\prod_{i=1}^{n} x_i = \min_{y \in \{0,1\}^{\ell-1}} \frac{1}{2}(\sum_{i=1}^{n} x_i - \sum_{i=1}^{\ell-1} 2^i y_i)(\sum_{i=1}^{n} x_i - \sum_{i=1}^{\ell-1} 2^i y_i - 1)$$

Termwise with constraints?

Not easy to derive a quadratization with constraints

- Quadratization for the positive monomial ($\ell = \lceil \log(n) \rceil$):

$$\prod_{i=1}^{n} x_i = \min_{y \in \{0,1\}^{\ell-1}} \frac{1}{2} (\sum_{i=1}^{n} x_i - \sum_{i=1}^{\ell-1} 2^i y_i)(\sum_{i=1}^{n} x_i - \sum_{i=1}^{\ell-1} 2^i y_i - 1)$$

- To one monomial we associate auxiliary variables $y_1, y_2, \ldots, y_\ell$, but we lose the link of each single variable with the original variables.

Termwise with constraints?

Not easy to derive a quadratization with constraints

- Quadratization for the positive monomial ($\ell = \lceil \log(n) \rceil$):

$$\prod_{i=1}^{n} x_i = \min_{y \in \{0,1\}^{\ell-1}} \frac{1}{2}\left(\sum_{i=1}^{n} x_i - \sum_{i=1}^{\ell-1} 2^i y_i\right)\left(\sum_{i=1}^{n} x_i - \sum_{i=1}^{\ell-1} 2^i y_i - 1\right)$$

- To one monomial we associate auxiliary variables $y_1, y_2, \ldots, y_\ell$, but we lose the link of each single variable with the original variables.
- Which constraints should we add?

Summary of quadratization methods

Unconstrained		
Non-termwise		**Termwise**
Rosenberg	ABCG	$\lceil \log(n) \rceil - 1$

Constrained	
Non-termwise	~~**Termwise**~~
Rosenberg = ABCG	~~$\lceil \log(n) \rceil - 1$~~

Two complementary approaches

Quadratic reformulations of nonlinear binary optimization problems

Phase 1: Quadratization
Carefully chosen

↓

Phase 2: Convexification:
Simple, Linearization

(Anthony, Boros, Crama, & Gruber, 2017)
(Boros, Crama, & Rodríguez-Heck, 2018)
(Rodríguez-Heck, 2018)

`PQCR`: Polynomial binary optimization through Quadratic Convex Reformulation

Phase 1: Quadratization
Simple algorithm

↓

Phase 2: Convexification:
Carefully chosen, tailored

(Elloumi, Lambert, & Lazare, 2019)
(Lazare, 2019)

PQCR: Phase 1 - Quadratization

Input: a polynomial $f(x)$ with monomial set $\mathcal{M}$

1. A pairwise cover $\mathcal{H}$ of $\mathcal{M}$ is defined heuristically (PC0).
2. Relation between artificial and original variables is enforced using (linearized) constraints.

(Linearly Constrained) Quadratic Program

$$\begin{aligned} \min \ & g(x) = x^t Q x + c^t x \\ \text{s. t. } & x \in \mathcal{F}_{\mathcal{E}} \end{aligned} \tag{QP}$$

Where $\mathcal{F}_{\mathcal{E}}$ are Fortet's constraints for all appropriate indices of artificial and original binary variables coming from PC0:

$$\begin{aligned} x_i &\le x_{i_1} \\ x_i &\le x_{i_2} \\ x_i &\ge x_{i_1} + x_{i_2} - 1 \\ x_i &\ge 0 \end{aligned}$$

PQCR: Phase 2 - Convexification

Input: (Linearly Constrained) Quadratic Program

$$\begin{aligned} \min \;\; & g(x) = x^t Q x + c^t x \\ \text{s. t. } \; & x \in \mathcal{F}_{\mathcal{E}} \end{aligned} \tag{QP}$$

- Objective: define a function the value of which is equal to $g(x)$ with a positive semi-definite Hessian matrix Q.
- Can be achieved by adding to $g(x)$ null-functions over the domain $\mathcal{F}_{\mathcal{E}}$.

Convexification: Option 1

Smallest eigenvalue convexification: (Hammer & Rubin, 1970)

$$\min \; g_\lambda(x) = g(x) + \lambda \sum_{i=1}^{N} (x_i^2 - x_i) \qquad (QP_\lambda)$$
$$\text{s. t. } x \in \mathcal{F}_\mathcal{E}$$

- Modify diagonal entries of the hessian matrix of g by adding null functions to it.
- (QP_λ) is a quadratic program parametrized by λ such that:
  - $g_\lambda(x) = g(x), \forall x \in \mathcal{F}_\mathcal{E}$
  - Setting $\lambda = -\frac{\lambda_{\min}}{2}$ leads to convex $g_\lambda(x)$ and provides tightest continuous relaxation

Convexification: Option 1

Smallest eigenvalue convexification

$$g(x) = 2x_1 + 2x_2x_3 - 2x_6x_2 - 3x_5x_6$$

(where $x_6 = x_3x_4$ and $x_5 = x_1x_2$)
Hessian matrix:

$$Q = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.5 & 0 & 0 & -1 \\ 0 & 1.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.5 \\ 0 & -1 & 0 & 0 & -1.5 & 0 \end{pmatrix}$$

Convexification: Option 1

Smallest eigenvalue convexification

$$g(x) = 2x_1 + 2x_2x_3 - 2x_6x_2 - 3x_5x_6$$

(where $x_6 = x_3x_4$ and $x_5 = x_1x_2$)
Hessian matrix:

$$Q_\lambda = \begin{pmatrix} 2.08 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.08 & 1.5 & 0 & 0 & -1 \\ 0 & 1.5 & 2.08 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.08 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.08 & -1.5 \\ 0 & -1 & 0 & 0 & -1.5 & 2.08 \end{pmatrix}$$

where $\lambda_{min} = -2.08$,

Convexification: Option 1

Smallest eigenvalue convexification

$$g(x) = 2x_1 + 2x_2x_3 - 2x_6x_2 - 3x_5x_6$$

(where $x_6 = x_3x_4$ and $x_5 = x_1x_2$)
Hessian matrix:

$$Q_\lambda = \begin{pmatrix} 2.08 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.08 & 1.5 & 0 & 0 & -1 \\ 0 & 1.5 & 2.08 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.08 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.08 & -1.5 \\ 0 & -1 & 0 & 0 & -1.5 & 2.08 \end{pmatrix}$$

where $\lambda_{min} = -2.08$, hence

$$\min \ 2x_1 + 2x_2x_3 - 2x_6x_2 - 3x_5x_6 + 1.04 \sum_{i=1}^{N} (x_i^2 - x_i) \qquad (QP_\lambda)$$

$$\text{s. t. } x \in \mathcal{F}_\mathcal{E}$$

Convexification: Option 2

Non-uniform diagonal convexification: `QCR` (Billionnet & Elloumi, 2007)

$$\min \; g_\alpha(x) = g(x) + \sum_{i=1}^{N} \alpha_i (x_i^2 - x_i) \qquad (QP_\alpha)$$
$$\text{s. t. } x \in \mathcal{F}_\mathcal{E}$$

How to compute α such that

- g_α is convex, and
- **continuous relaxation** bound value of (QP_α) is **maximized**?

Convexification: Option 2

Non-uniform diagonal convexification: `QCR` (Billionnet & Elloumi, 2007)

$$\min \; g_\alpha(x) = g(x) + \sum_{i=1}^{N} \alpha_i (x_i^2 - x_i) \qquad (QP_\alpha)$$
$$\text{s. t. } x \in \mathcal{F}_\mathcal{E}$$

How to compute α such that

- g_α is convex, and
- **continuous relaxation** bound value of (QP_α) is **maximized**?

$\rightarrow$ can be done by solving an SDP relaxation of (QP_α).

RWTH AACHEN UNIVERSITY

Convexification: Option 3 (PQCR)

General convexification framework: PQCR (Elloumi, Lambert, & Lazare, 2019)

- Use the quadratization *constraints* to add further null functions

$$x_i^2 - x_i = 0, \quad \text{for original variables } i \tag{1}$$

$$x_i - x_i x_j = 0, \quad \text{for variables } j \text{ in pairwise cover of } i \tag{2}$$

$$x_i - x_j x_k = 0, \quad \text{for variables } j \text{ and } k \text{ in pairwise cover of } i \tag{3}$$

$$x_i x_j - x_k x_\ell = 0, \quad \text{for two different decompositions of a monomial} \tag{4}$$

Convexification: Option 3 (PQCR)

General convexification framework: PQCR (Elloumi, Lambert, & Lazare, 2019)

- Use the quadratization *constraints* to add further null functions

$$
\begin{aligned}
&x_i^2 - x_i = 0, && \text{for original variables } i && (1)\\
&x_i - x_i x_j = 0, && \text{for variables } j \text{ in pairwise cover of } i && (2)\\
&x_i - x_j x_k = 0, && \text{for variables } j \text{ and } k \text{ in pairwise cover of } i && (3)\\
&x_i x_j - x_k x_\ell = 0, && \text{for two different decompositions of a monomial} && (4)
\end{aligned}
$$

- Resulting in the following parametrized function:

$$
\begin{aligned}
g_{\alpha,\beta,\delta,\lambda}(x) =& g(x) + \sum_{i \in I \cup J} \alpha_i (x_i^2 - x_i) + \sum_{\substack{(i,j) \in J \times (I \cup J) \\ \mathcal{E}_j \subset \mathcal{E}_i}} \beta_{ij} (x_i - x_i x_j) \\
& \sum_{\substack{(i,j,k) \in J \times (I \cup J)^2 \\ \mathcal{E}_i = \mathcal{E}_j \cup \mathcal{E}_k}} \delta_{ijk} (x_i - x_j x_k) + \sum_{\substack{(i,j,k,\ell) \in (I \cup J)^4 \\ \mathcal{E}_i \cup \mathcal{E}_j = \mathcal{E}_k \cup \mathcal{E}_\ell}} \lambda_{i,j,k,\ell} (x_i x_j - x_k x_\ell)
\end{aligned}
$$

Convexification: Option 3 (PQCR)

General convexification framework: PQCR

$$
\begin{aligned}
\min\ & 2x_1 + 3x_2x_3 - 2x_2x_6 - 3x_5x_6 + 1(x_1^2 - x_1) + 1(x_2^2 - x_2) \qquad (QP_{\alpha,\beta,\delta,\gamma})\\
& + 0.7(x_3^2 - x_3) + 0.09(x_4^2 - x_4) + 2.2(x_5^2 - x_5) + 1.3(x_6^2 - x_6)\\
& - 3.96(x_1x_5 - x_5) - 1.96(x_1x_5 - x_5) - 3.18(x_3x_6 - x_6) - 0.36(x_4x_6 - x_6)\\
& - 0.04(x_1x_2 - x_5) + 0.18(x_3x_4 - x_6)\\
\text{s. t. } & x \in \mathcal{F}_{\mathcal{E}}
\end{aligned}
$$

- Inequalities from the quadratization $x_5 = x_1x_2$ and $x_6 = x_3x_4$
- Derived valid inequalities:
  - $x_1x_5 - x_5$ and $x_2x_5 - x_5$
  - $x_3x_6 - x_6$ and $x_4x_6 - x_6$

Convexification: Option 3 (PQCR)

General convexification framework: PQCR

$$\begin{aligned}\min\ & 2x_1 + 3x_2x_3 - 2x_2x_6 - 3x_5x_6 + 1(x_1^2 - x_1) + 1(x_2^2 - x_2) \qquad (QP_{\alpha,\beta,\delta,\gamma})\\ & + 0.7(x_3^2 - x_3) + 0.09(x_4^2 - x_4) + 2.2(x_5^2 - x_5) + 1.3(x_6^2 - x_6)\\ & - 3.96(x_1x_5 - x_5) - 1.96(x_1x_5 - x_5) - 3.18(x_3x_6 - x_6) - 0.36(x_4x_6 - x_6)\\ & - 0.04(x_1x_2 - x_5) + 0.18(x_3x_4 - x_6)\\ \text{s. t. }\ & x \in \mathcal{F}_{\mathcal{E}}\end{aligned}$$

- Inequalities from the quadratization $x_5 = x_1x_2$ and $x_6 = x_3x_4$
- Derived valid inequalities:
  - $x_1x_5 - x_5$ and $x_2x_5 - x_5$
  - $x_3x_6 - x_6$ and $x_4x_6 - x_6$

Method	Continuous relaxation bound
Smallest eigenvalue	-1.7
QCR	-1.6
PQCR	-0.6

Convexification: Option 3 (PQCR)

Theorem (Elloumi, Lambert, & Lazare, 2019)

The optimal values $(\alpha^*, \beta^*, \delta^*, \lambda^*)$ are given by the optimal values of the dual variables associated with the constraints (5)–(8) of the following (SDP)

$$
\begin{aligned}
\min\ & \langle Q, X\rangle + c^T x && && \text{(SDP)}\\
\text{s. t. } & X_{ii} - x_i = 0 && i \in I \cup J && (5)\\
& -X_{ij} + x_i = 0 && (i,j) \in J \times (I \cup J) : \mathcal{E}_i \subset \mathcal{E}_j && (6)\\
& -X_{jk} + x_i = 0 && (i,j,k) \in J \times (I \cup J)^2 : \mathcal{E}_i = \mathcal{E}_j \cup \mathcal{E}_k && (7)\\
& X_{ij} - X_{kl} = 0 && (i,j,k,l) \in (I \cup J)^4 : \mathcal{E}_i \cup \mathcal{E}_j = \mathcal{E}_k \cup \mathcal{E}_l && (8)\\
& \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0 && &&\\
& x \in \mathbb{R}^N, X \in S^N && &&
\end{aligned}
$$

Computational results: LABS

- Instances from `http://polip.zib.de/autocorrelated_sequences/`
- Time limit: 5h (3h SDP, 2h CPLEX)

Instance			PQCR					Baron 17.4.1		
Name	*n*	*m*	*N*	*Gap (%)*	*tSdp*	T_t	*Nodes*	*Gap (%)*	T_t	*Nodes*
b.20.03	20	38	20	**0**	1	2	0	100	**1**	1
b.20.05	20	207	65	**23**	22	23	5886	1838	**2**	1
b.20.10	20	833	124	**8**	837	846	24183	2918	**125**	7
b.20.15	20	1494	164	**5**	1228	1242	9130	3202	**728**	9
b.25.03	25	48	25	**0**	1	2	0	100	**0**	1
b.25.06	25	407	105	**17**	461	469	163903	2307	**65**	27
b.25.13	25	1782	206	**4**	1552	**1603**	76828	3109	3750	75
b.25.19	25	3040	265	**4**	–	**13433**	224550	3356	14399	129
b.25.25	25	3677	289	**5**	–	**13395**	167423	3405	(12 %)	100
b.30.04	30	223	82	**23**	58	78	134635	1347	**7**	7
b.30.08	30	926	174	**10**	1940	**2040**	752765	2696	2778	237
b.30.15	30	2944	296	**5**	–	**13525**	438278	3221	(21 %)	103
b.30.23	30	5376	390	**11**	5953	**6865**	9337391	3450	(135 %)	8
b.30.30	30	6412	422	**4**	8500	**15352**	452460	3470	(161 %)	5
b.35.04	35	263	97	**19**	135	167	156085	1350	**32**	13
b.35.09	35	1381	234	**10**	2245	**4630**	8163651	2826	(29 %)	354

Results:

- `Q+Cplex` and `Q+QCR` did not solve any instance.
- `PQCR` very tight gaps, solves several previously unsolved instances.

Two complementary approaches

Quadratic reformulations of nonlinear binary optimization problems

Phase 1: Quadratization
Carefully chosen

↓

Phase 2: Convexification:
Simple, Linearization

(Anthony, Boros, Crama, & Gruber, 2017)
(Boros, Crama, & Rodríguez-Heck, 2018)
(Rodríguez-Heck, 2018)

PQCR: Polynomial binary optimization through Quadratic Convex Reformulation

Phase 1: Quadratization
Simple algorithm

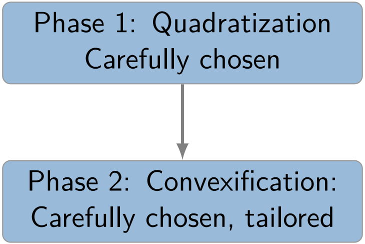
↓

Phase 2: Convexification:
Carefully chosen, tailored

(Elloumi, Lambert, & Lazare, 2019)
(Lazare, 2019)

This project

Combine both methods into a single one:



RWTH AACHEN UNIVERSITY

Quadratizations without constraints

- Instances from `http://polip.zib.de/autocorrelated_sequences/`
- Time limit: 1h

Instance		ROS+QCR				T+QCR			
Name	*Opt*	*N*	*Gap (%)*	*Nodes*	T_t	*N*	*Gap (%)*	*Nodes*	T_t
b.20.5	-416	65	804612	73437542	-	137	239	32889233	-
b.20.10	-2936	124	31206	38511235	-	698	361	24897180	-
b.25.6	-960	105	1843723	36766956	-	297	328	30810993	-
b.25.13	-8144	206	59614	20336389	-	1560	403	19983753	-
b.30.4	-324	82	829465	38587389	-	139	191	19931268	-
b.35.4	-384	97	1038548	39040424	-	164	196	20091757	-

- Both methods give **very** bad bounds.
- T+QCR has better bounds than ROS+QCR.
- ROS+QCR is not a viable method.

Quadratizations without constraints

- Instances from http://polip.zib.de/autocorrelated_sequences/
- Time limit: 1h

Instance		ROS+QCR				T+QCR			
Name	*Opt*	*N*	*Gap (%)*	*Nodes*	T_t	*N*	*Gap (%)*	*Nodes*	T_t
b.20.5	-416	65	804612	73437542	-	137	239	32889233	-
b.20.10	-2936	124	31206	38511235	-	698	361	24897180	-
b.25.6	-960	105	1843723	36766956	-	297	328	30810993	-
b.25.13	-8144	206	59614	20336389	-	1560	403	19983753	-
b.30.4	-324	82	829465	38587389	-	139	191	19931268	-
b.35.4	-384	97	1038548	39040424	-	164	196	20091757	-

- Both methods give **very** bad bounds.
- T+QCR has better bounds than ROS+QCR.
- ROS+QCR is not a viable method.

We lose the advantage of PQCR over QCR, because we lose link between original and artificial variables!

Quadratizations with constraints: PC heuristics

Instance		PQCR with PC1			PQCR with PC2		
Name	Opt	N	LB	T_t	N	LB	T_t
b.20.5	-416	64	-435	70	56	-439	63
b.20.10	-2936	123	-3052	112	135	-3115	132
b.40.10	-8248	303	-8590	4385	315	-8659	4562

Instance		PQCR with PC3			PQCR with PC0		
Name	Opt	N	LB	T_t	N	LB	T_t
b.20.5	-416	40	-436	59	65	-435	64
b.20.10	-2936	93	-3068	112	124	-3051	130
b.40.10	-8248	262	-8745	2162	304	-8589	3723

Best quadratization in terms of:

- LB: PQCR with PC0 / PQCR with PC1
- N: PQCR with PC3
- T_t: PQCR with PC3 (difference only significant for b.40.10)

Full quadratization

Definition: Full quadratization (Lazare, 2019)

The *full quadratization* of $f(x)$, $x \in \{0,1\}^n$ is defined on a pairwise cover that introduces an auxiliary variable for **every** product $\prod_j x_j$ of variables of degree at least two and at most $\lceil \frac{d}{2} \rceil$.

Example 2

$$f(x) = 2x_1x_5 + 3x_2x_3 - 2x_2x_3x_4 - 3x_1x_2x_3x_4$$

The full quadratization for f introduces a variable for every product in red:

$$xx^T = \begin{pmatrix} x_1^2 & \color{red}{x_1x_2} & \color{red}{x_1x_3} & \color{red}{x_1x_4} & \color{red}{x_1x_5} \\ x_2x_1 & x_2^2 & \color{red}{x_2x_3} & \color{red}{x_2x_4} & \color{red}{x_2x_5} \\ x_3x_1 & x_3x_2 & x_3^2 & \color{red}{x_3x_4} & \color{red}{x_3x_5} \\ x_4x_1 & x_4x_2 & x_4x_3 & x_4^2 & \color{red}{x_4x_5} \\ x_5x_1 & x_5x_2 & x_5x_3 & x_5x_4 & x_5^2 \end{pmatrix}$$

Partial quadratization

Definition: Partial quadratization (Lazare, 2019)

The *partial quadratization* of $f(x)$, $x \in \{0,1\}^n$ is defined on a pairwise cover that introduces an auxiliary variable for every product $\prod_j x_j$ of variables of degree at least two and at most $\lceil \frac{d}{2} \rceil$ **appearing in at least one monomial** of f.

Example 2

$$f(x) = 2x_1x_5 + 3x_2x_3 - 2x_2x_3x_4 - 3x_1x_2x_3x_4$$

The partial quadratization for f introduces a variable for every product in blue:

$$xx^T = \begin{pmatrix} x_1^2 & x_1x_2 & x_1x_3 & x_1x_4 & x_1x_5 \\ x_2x_1 & x_2^2 & x_2x_3 & x_2x_4 & x_2x_5 \\ x_3x_1 & x_3x_2 & x_3^2 & x_3x_4 & x_3x_5 \\ x_4x_1 & x_4x_2 & x_4x_3 & x_4^2 & x_4x_5 \\ x_5x_1 & x_5x_2 & x_5x_3 & x_5x_4 & x_5^2 \end{pmatrix}$$

Quadatizations with constraints: Full and Partial

Instance			PQCR with Full		PQCR with Partial	
Name	n	Opt	N	Const SDP	N	Const SDP
b.20.5	20	-416	210	22156	83	1308
b.20.10	20	-2936	210	22156	138	6275
b.20.15	20	-5960	210	22156	176	13757

Instance			PQCR with Full		PQCR with Partial	
Name	n	Opt	LB	T_t	LB	T_t
b.20.5	20	-416	-417	9700	-422	3
b.20.10	20	-2936	-3016	7439	-3040	180
b.20.15	20	-5960	-6025	10831	-6059	2060

Best quadratization in terms of:

- LB: PQCR with Full
- T_t: PQCR with Partial

Quadatizations with constraints: Full and Partial

Instance			PQCR with Full		PQCR with Partial	
Name	n	Opt	N	Const SDP	N	Const SDP
b.20.5	20	-416	210	22156	83	1308
b.20.10	20	-2936	210	22156	138	6275
b.20.15	20	-5960	210	22156	176	13757

Instance			PQCR with Full		PQCR with Partial	
Name	n	Opt	LB	T_t	LB	T_t
b.20.5	20	-416	-417	9700	-422	3
b.20.10	20	-2936	-3016	7439	-3040	180
b.20.15	20	-5960	-6025	10831	-6059	2060

Best quadratization in terms of:

- LB: PQCR with Full
- T_t: PQCR with Partial

PQCR with Partial gives a good compromise!

Perspectives

Next steps

- Compare and understand link between quadratizations and linearizations.
- Which quadratizations are best for which convexification method?
- Add valid inequalities to the resolution of the SDPs.
- Test on other instances than LABS, which have a very special structure, and are especially difficult to solve.

Perspectives

Next steps

- Compare and understand link between quadratizations and linearizations.
- Which quadratizations are best for which convexification method?
- Add valid inequalities to the resolution of the SDPs.
- Test on other instances than LABS, which have a very special structure, and are especially difficult to solve.

Practical challenges

- Interaction between codes.
- Many different variants of Phase 1 and of Phase 2, many experiments to be carried out to choose a good combination.

Bibliography I

Anthony, M., Boros, E., Crama, Y., & Gruber, A. (2017). Quadratic reformulations of nonlinear binary optimization problems. *Mathematical Programming*, *162*(1-2), 115–144.

Billionnet, A., & Elloumi, S. (2007). Using a mixed integer quadratic programming solver for the unconstrained quadratic 0–1 problem. *Mathematical Programming*, *109*(1), 55–68.

Boros, E., Crama, Y., & Rodríguez-Heck, E. (2018). *Compact quadratizations for pseudo-Boolean functions.* (Submitted)

Buchheim, C., & Rinaldi, G. (2007). Efficient reduction of polynomial zero-one optimization to the quadratic case. *SIAM Journal on Optimization*, *18*(4), 1398–1413.

Elloumi, S., Lambert, A., & Lazare, A. (2019). *Solving unconstrained 0–1 polynomial programs through quadratic convex reformulation.*

RWTH AACHEN UNIVERSITY

Bibliography II

Freedman, D., & Drineas, P. (2005, June). Energy minimization via graph cuts: settling what is possible. In *Ieee conference on computer vision and pattern recognition* (Vol. 2, pp. 939–946).

Hammer, P. L., & Rubin, A. A. (1970). Some remarks on quadratic programming with 0–1 variables. *Revue Française d'Informatique et de Recherche Opérationnelle*, *4*, 67–79.

Kolmogorov, V., & Zabih, R. (2004, Feb). What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(2), 147–159.

Lazare, A. (2019). *Global optimization of polynomial programs with mixed-integer variables* (Unpublished doctoral dissertation). (PhD thesis, Université Paris-Saclay)

Bibliography III

Rodríguez-Heck, E. (2018). *Linear and quadratic reformulations of nonlinear optimization problems in binary variables* (Unpublished doctoral dissertation). (PhD thesis, Université de Liège)

Rosenberg, I. G. (1975). Reduction of bivalent maximization to the quadratic case. *Cahiers du Centre d'Études de Recherche Opérationnelle*, *17*, 71–74.