



HAL
open science

Solving unconstrained 0-1 polynomial programs through quadratic convex reformulation

Sourour Elloumi, Amélie Lambert, Arnaud Lazare

► **To cite this version:**

Sourour Elloumi, Amélie Lambert, Arnaud Lazare. Solving unconstrained 0-1 polynomial programs through quadratic convex reformulation. *Journal of Global Optimization*, 2021, 80 (2), pp.231-248. 10.1007/s10898-020-00972-2 . hal-01872996v4

HAL Id: hal-01872996

<https://hal.science/hal-01872996v4>

Submitted on 12 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Solving unconstrained 0-1 polynomial programs through quadratic convex reformulation

Sourour Elloumi^{1,2} · Amélie Lambert¹ ·
Arnaud Lazare^{1,2}

12 March 2020

Abstract We propose a method called Polynomial Quadratic Convex Reformulation (PQCR) to solve exactly unconstrained binary polynomial problems (UBP) through quadratic convex reformulation. First, we quadratize the problem by adding new binary variables and reformulating (UBP) into a non-convex quadratic program with linear constraints (MIQP). We then consider the solution of (MIQP) with a specially-tailored quadratic convex reformulation method. In particular, this method relies, in a pre-processing step, on the resolution of a semi-definite programming problem where the link between initial and additional variables is used. We present computational results where we compare PQCR with the solvers `Baron` and `Scip`. We evaluate PQCR on instances of the image restoration problem and the low auto-correlation binary sequence problem from `MINLPLib`. For this last problem, 33 instances were unsolved in `MINLPLib`. We solve to optimality 10 of them, and for the 23 others we significantly improve the dual bounds. We also improve the best known solutions of many instances.

Keywords Unconstrained binary polynomial programming, Global optimization, Semi-definite programming, Quadratic convex reformulation

1 Introduction

In this paper, we are interested in solving the unconstrained binary polynomial optimization problem that can be stated as follows:

1. UMA-ENSTA
828 Boulevard des Maréchaux, 91120 Palaiseau, France
E-mail: {sourour.elloumi,arnaud.lazare}@ensta-paris.fr

2. CEDRIC-Cnam
292 rue saint Martin, F-75141 Paris Cedex 03, France
E-mail: amelie.lambert@cnam.fr

$$(UBP) \begin{cases} \min F(x) = \sum_{p=1}^m a_p \prod_{i \in \mathcal{M}_p} x_i \\ \text{s.t.} \\ x_i \in \{0, 1\}, \quad i \in I \end{cases}$$

where $I = \{1, \dots, n\}$, $F(x)$ is an n -variable multi-linear function of degree d and m is the number of monomials. For a monomial p , \mathcal{M}_p is the subset of I containing the indexes of the variables involved in p . It follows that $d = \max_p |\mathcal{M}_p|$.

Unconstrained binary polynomial optimization is a general model that allows to formulate many important problems in optimization. The special case where the polynomial objective function of (UBP) is a quadratic function (i.e. $d = 2$) has been widely studied. In this case, (UBP) has many applications, including those from financial analysis [28], cluster analysis [35], computer aided design [24] or machine scheduling [36]. Moreover, many graph combinatorial optimization problems such as determining maximum cliques, maximum cuts, maximum vertex packing or maximum independent sets can be formulated as quadratic optimization problems [5, 13, 34]. In the cubic case (i.e. $d = 3$), the important class of satisfiability problems known as 3-SAT, can be formulated as (UBP) [23]. In the case where $d \geq 3$, there also exists many applications including, for example: the construction of binary sequences with low aperiodic correlation [7] that is one of the most challenging problems in signal design theory, or the image restoration problem in computer vision [16].

Problem (UBP) is NP-hard in general [18]. Practical difficulties come from the non-convexity of F and the integrality of its variables. Several algorithms can handle (UBP). In particular, methods were designed to solve the more general class of mixed-integer nonlinear programs. These methods are branch-and-bound algorithms based on a convex relaxation of (UBP). The most classical relaxation consists in the complete linearization of (UBP), but quadratic convex relaxations can also be used. For instance, the well known α -branch-and-bound algorithm [2] computes, at each node, convex under-estimators of nonlinear functions by perturbing the diagonal of the Hessian matrix of the objective function. Several implementations of these algorithms are available, see for instance **Baron** [37], **Antigone** [32], **SCIP** [1] or **Couenne** [6].

In the case where the objective function is a polynomial, but the variables are continuous, Lasserre proposes in [26] an algorithm based on a hierarchy of semi-definite relaxations. The idea is, at each rank of the hierarchy, to successively tighten semi-definite relaxations in order to reach its optimal solution value. It is also proven in [26] that this hierarchy converges in a finite number of iterations. Further, this work has been extended to hierarchies of second order conic programs [3, 19, 25], and of sparse doubly non-negative relaxation [22]. Although these algorithms were not originally tailored for binary programming, they can handle (UBP) by considering the quadratic constraint $x_i^2 = x_i$. Methods devoted to the binary polynomial case were also proposed. In [14, 27], the authors use separable or convex under-estimators to approximate a given polynomial. Other methods based on linear reformulations can be found in [16, 38], in which linear equivalent formulations to (UBP) are proposed and then improved. In [15], the authors focus on a polyhedral description of the linearization of a binary polynomial program. Finally, the work in [4] considers *quadratizations* with a minimal number of additional variables.

In the quadratic case ($d = 2$), (UBP) can be exactly solved by quadratic convex reformulation. This general approach allows to reformulate any non-convex mixed integer quadratic problem into a convex mixed-integer quadratic problem. An equivalent convex formulation, with the sharpest continuous relaxation bound, is obtained from the solution of a semi-definite programming relaxation, in a pre-processing phase of the whole method. An initial version, named QCR, was introduced in [8, 12]. An improved version, named MIQCR, was further proposed in [9,10,17]. MIQCR relies on the solution of a heavier SDP program than QCR in order to reach sharper reformulated problems.

Our contribution in this paper:

We extend quadratic convex reformulation to polynomial problems like (UBP). We start by adding some auxiliary variables in order to reformulate (UBP) into an equivalent mixed integer quadratic problem having a quadratic objective function, linear constraints, and binary variables. To the best of our knowledge, this *quadratization* is original. Then, we experimentally observe that QCR and MIQCR methods are not able to solve our equivalent quadratic formulation. We hence devise a specific quadratic convex reformulation that uses the links between the original and auxiliary variables. This new quadratic convex reformulation, that we call PQCR (Polynomial Quadratic Convex Reformulation), relies on an SDP relaxation which is stronger than QCR, but weaker than MIQCR. We show that PQCR computationally outperforms both, and is moreover competitive with state-of-the-art exact methods for (UBP).

The outline of the paper is the following. In Section 2, we define and present our quadratizations of (UBP). In Section 3, we introduce our family of convex reformulations and we prove how we compute the best parameters. Then, in Section 4, we present our computational results on polynomial instances of degree 4 coming from the literature. Section 5 draws a conclusion.

2 Phase 1: Quadratzation of (UBP)

In this section, we present how we build equivalent quadratic formulations to (UBP). For this, in each monomial of degree 3 or greater, we iteratively replace each product of two variables by an additional variable. Hence, any additional variable is the product of two other initial or additional variables.

More formally, we define the set of indexes of the additional variables $\mathcal{A} = \{n + 1, \dots, N\}$, where N is the total number of initial and additional variables. We also define the subsets \mathcal{E}_i for both initial and additional variables, and the mapping f and s defining any additional variable as the product of two other variables as follows:

Definition 1 Quadratzation $\mathcal{Z} = (\mathcal{A}, \mathcal{E}, f, s)$

We define $\mathcal{E}_i \forall i \in I \cup \mathcal{A}$ and mappings $f : \mathcal{A} \rightarrow I \cup \mathcal{A}$ and $s : \mathcal{A} \rightarrow I \cup \mathcal{A}$ as follows:

- If $i \in I$, i.e. x_i is an initial variable, then we set $\mathcal{E}_i = \{i\}$
- If $i \in \mathcal{A}$, i.e. x_i is an additional variable, then there exist two indexes $(f(i), s(i)) \in (I \cup \mathcal{A})^2$ such that

$$x_i = x_{f(i)}x_{s(i)} \tag{1}$$

and we set $\mathcal{E}_i = \mathcal{E}_{f(i)} \cup \mathcal{E}_{s(i)}$

□

It follows from Definition 1 that, $\forall i \in I \cup \mathcal{A}$, \mathcal{E}_i is the subset of indexes from I whose product is equal to x_i , precisely:

$$x_i = \prod_{i' \in \mathcal{E}_i} x_{i'} \quad (2)$$

We now define a *valid* quadratization as a reformulation with N variables where any monomial of degree at least three is replaced by the product of two variables.

Definition 2 Valid quadratization

A quadratization $\mathcal{Z} = (\mathcal{A}, \mathcal{E}, f, s)$ is a valid quadratization with $N = |I \cup \mathcal{A}|$ variables if, for any monomial p of degree greater than or equal to 3 (i.e. $|\mathcal{M}_p| \geq 3$), there exist $(j, k) \in (I \cup \mathcal{A})^2$ such that $\mathcal{M}_p = \mathcal{E}_j \cup \mathcal{E}_k$ and $\prod_{i \in \mathcal{M}_p} x_i = x_j x_k$. Then the monomial p is replaced by a quadratic term. □

In Section 3 (see Algorithm 2), we present an algorithm that allows to build a valid quadratization, and that we use in our experiments. Now, we are rather interested in the reformulation that one can deduce from a given quadratization.

Given a valid quadratization $\mathcal{Z} = (\mathcal{A}, \mathcal{E}, f, s)$ and with a slight misuse of notation, we will consider that x is a vector of N variables and that the objective function F of (UBP) can be applied to this N -dimensional vector. We derive from Definition 2 the following reformulation of $F(x)$ into a quadratic form $g^{\mathcal{Z}}(x)$:

$$g^{\mathcal{Z}}(x) = \sum_{\substack{p: |\mathcal{M}_p| \geq 3 \\ \mathcal{M}_p = \mathcal{E}_j \cup \mathcal{E}_k}} a_p x_j x_k + \sum_{p: |\mathcal{M}_p| \leq 2} a_p \prod_{i \in \mathcal{M}_p} x_i$$

and by splitting $g^{\mathcal{Z}}$ into its quadratic and linear parts and introducing the symmetric $N \times N$ matrix Q and the N -vector c with the right entries deduced from coefficients a_p , we will below adopt the following notation for function $g^{\mathcal{Z}}$:

$$g^{\mathcal{Z}}(x) \equiv x^T Q x + c^T x$$

From Definitions 1 and 2, it holds that $g^{\mathcal{Z}}(x) = F(x)$ for any x satisfying (1). Our reformulation of (UBP) into a non-convex mixed-integer quadratic program ($MIQP^{\mathcal{Z}}$) follows from all this.

$$(MIQP^{\mathcal{Z}}) \left\{ \begin{array}{ll} \min g^{\mathcal{Z}}(x) = x^T Q x + c^T x & \\ \text{s.t.} & \\ x_i \leq x_{f(i)} & i \in \mathcal{A} \quad (3) \\ x_i \leq x_{s(i)} & i \in \mathcal{A} \quad (4) \\ x_i \geq x_{f(i)} + x_{s(i)} - 1 & i \in \mathcal{A} \quad (5) \\ x_i \geq 0 & i \in \mathcal{A} \quad (6) \\ x_i \in \{0, 1\} & i \in I \cup \mathcal{A} \quad (7) \end{array} \right.$$

Where Constraints (3)-(6) are the standard linearization inequalities that, together with (7), enforce the identity (1). In the following we denote by $\mathcal{F}^{\mathcal{Z}}$ the set of solutions defined by Constraints (3)-(7).

We now focus on the solution of problem $(MIQP^{\mathcal{Z}})$ in order to solve our original problem (UBP). The quadratic reformulation $(MIQP^{\mathcal{Z}})$ still has two difficulties: the non-convexity of its objective function $g^{\mathcal{Z}}$ and the integrality of its variables. Some state-of-the-art solvers can solve $(MIQP^{\mathcal{Z}})$ to global optimality. These solvers may not be enough efficient for solving dense instances. In the next section, we present method PQCR, a new algorithm tailored for binary polynomial optimisation. The key idea is that the convexification phase can take advantage of the quadratization \mathcal{Z} to compute an equivalent convex formulation to $(MIQP^{\mathcal{Z}})$.

3 Phase 2: A quadratic convex reformulation of $(MIQP^{\mathcal{Z}})$

In this section, we consider the problem of reformulating $(MIQP^{\mathcal{Z}})$ by an equivalent quadratic 0-1 program with a convex objective function. To do this, we define a new convex function whose value is equal to the value of $g^{\mathcal{Z}}(x)$, but whose Hessian matrix is positive semi-definite. More precisely, we first add to $g^{\mathcal{Z}}(x)$ a combination of four sets of functions that vanish on the feasible set $\mathcal{F}^{\mathcal{Z}}$. Then we focus on computing the best combination that leads to a convex function and that maximize the optimal value of the continuous relaxation of the reformulated problem.

3.1 Valid quadratic equalities on the set $\mathcal{F}^{\mathcal{Z}}$

Given a quadratization $\mathcal{Z} = (\mathcal{A}, \mathcal{E}, f, s)$, we introduce null quadratic functions over the set $\mathcal{F}^{\mathcal{Z}}$.

Lemma 1 *The following quadratic forms vanish over the set $\mathcal{F}^{\mathcal{Z}}$:*

$$(S^{\mathcal{Z}}) \begin{cases} x_i^2 - x_i = 0 & i \in I \cup \mathcal{A} & (8) \\ x_i - x_{f(i)}x_{s(i)} = 0 & i \in \mathcal{A} & (9) \\ x_i - x_i x_j = 0 & (i, j) \in \mathcal{A} \times (I \cup \mathcal{A}) : \mathcal{E}_j \subset \mathcal{E}_i & (10) \\ x_i x_j - x_k x_l = 0 & (i, j, k, l) \in (I \cup \mathcal{A})^4 : \mathcal{E}_i \cup \mathcal{E}_j = \mathcal{E}_k \cup \mathcal{E}_l & (11) \end{cases}$$

Proof. Constraints (8) trivially hold since $x \in \{0, 1\}^N$. Constraints (9) come from Equality (1). For proving the validity of Constraints (10) and (11), we start by observing that by definition and since $x \in \{0, 1\}^N$, we have:

$$x_i x_j = \prod_{i' \in \mathcal{E}_i} x_{i'} \prod_{j' \in \mathcal{E}_j} x_{j'} = \prod_{i' \in \mathcal{E}_i \cup \mathcal{E}_j} x_{i'}$$

– *Constraints (10):* since $x \in \{0, 1\}^N$, and $\mathcal{E}_j \subset \mathcal{E}_i$, we have:

$$x_i x_j = \prod_{i' \in \mathcal{E}_i} x_{i'} = x_i$$

– *Constraints (11):* since $x \in \{0, 1\}^N$, and $\mathcal{E}_i \cup \mathcal{E}_j = \mathcal{E}_k \cup \mathcal{E}_l$, we have:

$$x_i x_j = \prod_{i' \in \mathcal{E}_i \cup \mathcal{E}_j} x_{i'} = \prod_{k' \in \mathcal{E}_k \cup \mathcal{E}_l} x_{k'} = x_k x_l$$

□

3.2 An equivalent quadratic convex formulation to $(MIQP^{\mathcal{Z}})$

We now compute a quadratic convex reformulation of $(MIQP^{\mathcal{Z}})$ and thus of (UBP). For this, we add to the objective function $g^{\mathcal{Z}}$ the null quadratic forms in (8)–(11). For each of them, we associate a real scalar parameter: α_i for Constraints (8), β_{ij} for Constraints (10), δ_{ijk} for Constraints (9), and λ_{ijkl} for Constraints (11). We get the following parameterized function:

$$\begin{aligned} g_{\alpha,\beta,\delta,\lambda}^{\mathcal{Z}}(x) = & g^{\mathcal{Z}}(x) + \sum_{i \in I \cup \mathcal{A}} \alpha_i (x_i^2 - x_i) + \sum_{i \in \mathcal{A}} \delta_i (x_i - x_{f(i)} x_{s(i)}) \\ & + \sum_{\substack{(i,j) \in \mathcal{A} \times (I \cup \mathcal{A}) \\ \mathcal{E}_j \subset \mathcal{E}_i}} \beta_{ij} (x_i - x_i x_j) + \sum_{\substack{(i,j,k,l) \in (I \cup \mathcal{A})^4 \\ \mathcal{E}_i \cup \mathcal{E}_j = \mathcal{E}_k \cup \mathcal{E}_l}} \lambda_{ijkl} (x_i x_j - x_k x_l) \end{aligned}$$

It follows from Lemma 1 that $g_{\alpha,\beta,\delta,\lambda}^{\mathcal{Z}}(x)$ has the same value as $g^{\mathcal{Z}}(x)$ for any $x \in \mathcal{F}^{\mathcal{Z}}$. Moreover, there exist vector parameters α , β , δ and λ such that $g_{\alpha,\beta,\delta,\lambda}^{\mathcal{Z}}$ is a convex function. Take for instance, α equals to the opposite of the smallest eigenvalue of Q (the Hessian matrix of $g^{\mathcal{Z}}$), and $\beta = \delta = \lambda = 0$.

By replacing $g^{\mathcal{Z}}$ by the new function, we obtain the following family of convex mixed-integer quadratic equivalent formulations to $(MIQP^{\mathcal{Z}})$:

$$(MIQP_{\alpha,\beta,\delta,\lambda}^{\mathcal{Z}}) \begin{cases} \min g_{\alpha,\beta,\delta,\lambda}(x) = x^T Q_{\alpha,\beta,\delta,\lambda} x + c_{\alpha,\beta,\delta,\lambda}^T x \\ \text{s.t.} \\ x \in \mathcal{F}^{\mathcal{Z}} \end{cases}$$

where $Q_{\alpha,\beta,\delta,\lambda} \in S_N$ is the Hessian matrix of $g_{\alpha,\beta,\delta,\lambda}^{\mathcal{Z}}(x)$, and $c_{\alpha,\beta,\delta,\lambda} \in \mathbb{R}^N$ is the vector of linear coefficients of $g_{\alpha,\beta,\delta,\lambda}^{\mathcal{Z}}(x)$.

We are interested in parameters $(\alpha, \beta, \delta, \lambda)$ such that:

- i) $g_{\alpha,\beta,\delta,\lambda}^{\mathcal{Z}}$ is a convex function,
- ii) the continuous relaxation of $(MIQP_{\alpha,\beta,\delta,\lambda}^{\mathcal{Z}})$ gives the the tightest bound.

More formally, we want to solve the following optimization problem:

$$(CP) : \max_{\substack{\alpha,\beta,\delta,\lambda \\ Q_{\alpha,\beta,\delta,\lambda} \succeq 0}} \left\{ \min_{x \in \overline{\mathcal{F}}^{\mathcal{Z}}} g_{\alpha,\beta,\delta,\lambda}^{\mathcal{Z}}(x) \right\}$$

where $\overline{\mathcal{F}}^{\mathcal{Z}}$ is the set $\mathcal{F}^{\mathcal{Z}}$ where Constraints (7) are relaxed, i.e. $x \in [0, 1]^N$.

In the rest of this section, we will focus on solving (CP).

3.3 Computing an optimal solution to (CP)

The following theorem shows that problem (CP) is equivalent to the dual of a semi-definite relaxation of $(MIQP^{\mathcal{Z}})$.

Theorem 1 *The optimal value of (CP) is equal to the optimal value of the following semi-definite program (SDP^Z):*

$$(SDP^Z) \left\{ \begin{array}{ll} \min \langle Q, X \rangle + c^T x & \\ \text{s.t.} & \\ X_{ii} - x_i = 0 & i \in I \cup \mathcal{A} \quad (12) \\ x_i - X_{f(i)s(i)} = 0 & i \in \mathcal{A} \quad (13) \\ x_i - X_{ij} = 0 & (i, j) \in \mathcal{A} \times (I \cup \mathcal{A}) : \mathcal{E}_j \subset \mathcal{E}_i \quad (14) \\ X_{ij} - X_{kl} = 0 & (i, j, k, l) \in (I \cup \mathcal{A})^4 : \mathcal{E}_i \cup \mathcal{E}_j = \mathcal{E}_k \cup \mathcal{E}_l \quad (15) \\ \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0 & (16) \\ x \in \mathbb{R}^N, X \in S_N & (17) \end{array} \right.$$

Proof. For simplicity, we rewrite \mathcal{F}^Z as follows: $\mathcal{F}^Z = \{x \in \{0, 1\}^N : Ax \leq b\}$ where A is a $M \times N$ -matrix, $b \in \mathbb{R}^M$, and we introduce $T = N + T_1 + T_2 + T_3$ the number of Constraints (8)–(11) respectively. We also introduce $(\alpha, \beta, \delta, \lambda)$ the dual variables associated with constraints (12)–(15) respectively.

We start by observing that $x \in [0, 1]^N$ is equivalent to $x^2 \leq x$, thus, (CP) is equivalent to:

$$\max_{\substack{\alpha \in \mathbb{R}^N \beta \in \mathbb{R}^{T_1} \delta \in \mathbb{R}^{T_2} \lambda \in \mathbb{R}^{T_3} \\ Q_{\alpha, \beta, \delta, \lambda} \succeq 0}} \left\{ \min_{x \in \mathbb{R}^N, x^2 \leq x, Ax \leq b} g_{\alpha, \beta, \delta, \lambda}(x) \right\} \quad (18)$$

(18) is a convex optimization problem over a convex set. If we consider the solution $\tilde{x}_i = 0.5 \forall i \in I$ and $\tilde{x}_i = \tilde{x}_{f(i)} \tilde{x}_{s(i)}$, $\forall i \in \mathcal{A}$, the obtained \tilde{x} is an interior point and the Slater's conditions are satisfied for the minimization sub-problem. Then, by Lagrangian duality, (18) is equivalent to :

$$\max_{\substack{\alpha \in \mathbb{R}^N \beta \in \mathbb{R}^{T_1} \delta \in \mathbb{R}^{T_2} \\ \lambda \in \mathbb{R}^{T_3} \omega \in \mathbb{R}_+^M \gamma \in \mathbb{R}_+^M \\ Q_{\alpha, \beta, \delta, \lambda} \succeq 0}} \left\{ \min_{x \in \mathbb{R}^N} g_{\alpha, \beta, \delta, \lambda}(x) + \omega^T(x^2 - x) + \gamma^T(Ax - b) \right\} \quad (19)$$

where γ are the dual variables associated to Constraints $Ax \leq b$. It holds that (19) is equivalent to:

$$\max_{\substack{\alpha \in \mathbb{R}^N \beta \in \mathbb{R}^{T_1} \delta \in \mathbb{R}^{T_2} \lambda \in \mathbb{R}^{T_3}, \gamma \in \mathbb{R}_+^M \\ Q_{\alpha, \beta, \delta, \lambda} \succeq 0}} \left\{ \min_{x \in \mathbb{R}^N} g_{\alpha, \beta, \delta, \lambda}(x) + \gamma^T(Ax - b) \right\} \quad (20)$$

Indeed, if $(\alpha^*, \beta^*, \delta^*, \lambda^*, \omega^*, \gamma^*)$ is an optimal solution of (19), the solution $(\alpha' = \alpha^* + \omega^*, \beta^*, \delta^*, \lambda^*, \gamma^*)$ is also feasible to (20) with the same value. Moreover, a necessary condition for the quadratic function $g_{\alpha, \beta, \delta, \lambda}^Z(x) + \gamma^T(Ax - b)$ to have a finite minimum is that matrix $Q_{\alpha, \beta, \delta, \lambda}$ is positive semi-definite. Therefore (20) is equivalent to:

$$\max_{\alpha \in \mathbb{R}^N \beta \in \mathbb{R}^{T_1} \delta \in \mathbb{R}^{T_2} \lambda \in \mathbb{R}^{T_3} \gamma \in \mathbb{R}_+^M} \left\{ \min_{x \in \mathbb{R}^N} g_{\alpha, \beta, \delta, \lambda}(x) + \gamma^T(Ax - b) \right\} \quad (21)$$

We know from [29] that (21) is equivalent to problem (D):

$$(D) \begin{cases} \max -\gamma^T b - t \\ \text{s.t.} \\ \begin{pmatrix} t & \frac{1}{2}(c_{\alpha,\beta,\delta,\lambda}^T + \gamma^T A) \\ \frac{1}{2}(c_{\alpha,\beta,\delta,\lambda} + A^T \gamma) & Q_{\alpha,\beta,\delta,\lambda} \end{pmatrix} \succeq 0 \\ t \in \mathbb{R}, \alpha \in \mathbb{R}^N, \beta \in \mathbb{R}^{T_1}, \delta \in \mathbb{R}^{T_2}, \lambda \in \mathbb{R}^{T_3}, \gamma \in \mathbb{R}_+^M \end{cases}$$

By semi-definite duality of program (D), and with $\alpha, \beta, \delta, \lambda$ being the dual variables associated with Constraints (12)–(15) respectively, we get the following program ($SDP^{\mathcal{Z}'}$):

$$(SDP^{\mathcal{Z}'}) \begin{cases} \min \langle Q, X \rangle + c^T x \\ \text{s.t.} \\ (12) - (17) \\ Ax \leq b \end{cases}$$

We check that there is no duality gap between ($SDP^{\mathcal{Z}'}$) and (D). For this, following [33], we verify that the two following conditions hold.

- (i) The primal ($SDP^{\mathcal{Z}'}$) has a non empty feasible region since the solution $x = 0$, $X = 0$ is feasible, and then (D) is bounded.
- (ii) (D) has a non empty-interior, take $\bar{\beta}$, $\bar{\delta}$, $\bar{\lambda}$, and $\bar{\gamma}$ equal to 0, we get $Q_{\bar{\alpha}, \bar{\beta}, \bar{\delta}, \bar{\lambda}} = Q + \text{diag}(\bar{\alpha})$, and $c_{\bar{\alpha}, \bar{\beta}, \bar{\delta}, \bar{\lambda}} = c - \bar{\alpha}$. By Schur complement, we have:

$$\begin{pmatrix} t & \frac{1}{2}(c - \bar{\alpha})^T \\ \frac{1}{2}(c - \bar{\alpha}) & Q + \text{diag}(\bar{\alpha}) \end{pmatrix} \succ 0 \iff Q + \text{diag}(\bar{\alpha}) \succ 0 \text{ and } t - (c - \bar{\alpha})^T (Q + \text{diag}(\bar{\alpha}))^{-1} (c - \bar{\alpha}) > 0. \text{ Take for all } i \text{ a positive } \bar{\alpha}_i > \sum_{j \neq i} |q_{ij}| - q_{ii} \text{ so that the diagonal dominance holds for matrix } Q + \text{diag}(\bar{\alpha}) \text{ and } t \text{ large enough.}$$

From these equivalences, we know that we can build an optimal solution of (CP) from the optimal dual variables of ($SDP^{\mathcal{Z}'}$). However, we show that constraints $Ax \leq b$ are redundant in ($SDP^{\mathcal{Z}'}$) and we thus prove that ($SDP^{\mathcal{Z}'}$) and ($SDP^{\mathcal{Z}}$) are equivalent.

Claim Due to Constraints (12)–(14) and (16), $Ax \leq b$ are redundant in ($SDP^{\mathcal{Z}'}$).

Proof.

Recall that $Ax \leq b$ correspond to Constraints (3)–(6). The basic idea used here is that, since matrix $\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}$ is positive semi-definite, all its symmetric minors are non-negative.

- Constraint (6): $x_i \geq 0$. We consider the determinant $\begin{vmatrix} 1 & x_i \\ x_i & X_{ii} \end{vmatrix}$, which implies $X_{ii} - x_i^2 \geq 0$. By (12) we obtain $x_i - x_i^2 \geq 0$ and thus $x_i \geq 0$.

- Constraint (3): $x_i \leq x_{f(i)}$. Considering the determinant of the symmetric minor $\begin{vmatrix} X_{f(i)f(i)} & X_{f(i)i} \\ X_{if(i)} & X_{ii} \end{vmatrix} \stackrel{(12) \text{ and } (14)}{=} \begin{vmatrix} x_{f(i)} & x_i \\ x_i & x_i \end{vmatrix} = x_i(x_{f(i)} - x_i) \geq 0$, and since $x_i \geq 0$, we get $x_i \leq x_{f(i)}$.
- Constraint (4): $x_i \leq x_{s(i)}$. By symmetry, i.e. considering the determinant $\begin{vmatrix} X_{s(i)s(i)} & X_{s(i)i} \\ X_{is(i)} & X_{ii} \end{vmatrix}$, the inequality holds.
- Constraint (5): $x_i \geq x_{f(i)} + x_{s(i)} - 1$. By definition (16) implies $z^T \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} z \geq 0$, $\forall z \in \mathbb{R}^{N+1}$. By taking $\bar{z} = (1, 0, \dots, 0, \underbrace{-1}_{f(i)}, 0, \dots, 0, \underbrace{-1}_{s(i)}, 0, \dots, 0, \underbrace{1}_i, 0, \dots, 0)$, we have:

$$\begin{aligned} \bar{z}^T \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \bar{z} &= (x_i + 1 - x_{f(i)} - x_{s(i)}) - (x_{f(i)} - X_{f(i)f(i)} - X_{s(i)f(i)} + X_{if(i)}) \\ &\quad - (x_{s(i)} - X_{s(i)s(i)} - X_{f(i)s(i)} + X_{is(i)}) + (x_i - X_{f(i)i} - X_{s(i)i} + X_{ii}) \\ &= (x_i + 1 - x_{f(i)} - x_{s(i)}) \geq 0 \text{ by (12), (14) and (13)}. \end{aligned}$$

◇

We now conclude the proof of Theorem 1 since:

- (i) $v(CP) = v(D)$,
- (ii) there is no duality gap between (D) and $(SDP^{\mathcal{Z}'})$, then we have $v(D) = v(SDP^{\mathcal{Z}'})$,
- (iii) finally, by Claim 1, we get $v(CP) = v(D) = v(SDP^{\mathcal{Z}'}) = v(SDP^{\mathcal{Z}})$.

□

From the proof of Theorem 1, we state Corollary 1 that shows that from an optimal dual solution to $(SDP^{\mathcal{Z}})$ we can build an optimal solution to (CP) .

Corollary 1 *By Theorem 1, we have $v(CP) = v(SDP^{\mathcal{Z}})$, where $v(\cdot)$ is the optimal value of problem (\cdot) . Moreover, an optimal solution $(\alpha^*, \beta^*, \delta^*, \lambda^*)$ of (CP) corresponds to the optimal values of the dual variables associated with constraints (12)–(15) of $(SDP^{\mathcal{Z}})$ respectively.*

To sum up, we obtain $(MIQP^*)$, the best equivalent convex formulation to $(MIQP^{\mathcal{Z}})$:

$$(MIQP^*) \begin{cases} \min g_{\alpha^*, \beta^*, \delta^*, \lambda^*}(x) \\ \text{s.t.} \\ x \in \mathcal{F}^{\mathcal{Z}} \end{cases}$$

Corollary 1 also says that the continuous relaxation bound of $(MIQP^*)$ is equal to the optimal value of $(SDP^{\mathcal{Z}})$, i.e. $v(SDP^{\mathcal{Z}}) = v(MIQP^*)$. From Theorem 1, we deduce the following Algorithm 1 to solve (UBP).

Algorithm 1 PQCR an exact solution method for (UBP)

Step 1: Apply a quadratization $\mathcal{Z} = (\mathcal{A}, \mathcal{E}, f, s)$ to (UBP). In our experiments, we use the Lex quadratization described in Algorithm 2.

Step 2: Solve $(SDP^{\mathcal{Z}})$, deduce optimal values $\alpha^*, \beta^*, \delta^*, \lambda^*$, and build $(MIQP^*)$.

Step 3: Solve $(MIQP^*)$ by a branch-and-bound based on continuous relaxation.

Algorithm 2 Lex quadratization (F)

Require: A polynomial F of degree $d > 2$

Ensure: $(\mathcal{A}, \mathcal{E}, f, s)$, a valid quadratization of F (cf. Definition 2)

Initialization : $\mathcal{A} = \emptyset$ and $\mathcal{E}_i = \{i\} \forall i \in I$

for each monomial p from 1 to m **do**

Sort p by lexicographical order

$d \leftarrow \deg(p)$

while $d > 2$ **do**

for ℓ from 1 to $\lfloor \frac{d}{2} \rfloor$ **do**

 Consider the ℓ^{th} consecutive pair of variables $x_j x_k$

if there is no $i \in \mathcal{A}$, with $f(i) = j$ and $s(i) = k$ **then**

 Create an additional variable x_i and add it to \mathcal{A}

 set $f(i) = j$ and $s(i) = k$

 set $\mathcal{E}_i \leftarrow \mathcal{E}_{f(i)} \cup \mathcal{E}_{s(i)}$

end if

 Replace $x_j x_k$ by x_i

end for

$d \leftarrow \lceil \frac{d}{2} \rceil$

end while

end for

4 Numerical results

In this section, we evaluate PQCR on two applications. The first one is the *image restoration* problem [16], which results are presented in Section 4.1. The instances of this application are quite sparse with an average ratio $\frac{m}{n}$ of about 7. Then, in Section 4.2, we present the results of the second application, the *Low Auto-correlation Binary Sequence (LABS)* problem [7] which instances are much denser (average ratio $\frac{m}{n}$ of about 212). These instances are available on the MINLPLib website [31], and are very hard to solve. For most of them, the optimal solution value is not known.

In our experiments, we compare several algorithms/solvers:

- i) PQCR: the Lex quadratization is implemented in C. For solving $(SDP^{\mathcal{Z}})$, we used a tailored semi-definite solver that combines `csdp` together with the `Conic Bundle` library [20], see [11] for more details. Then, we used the C interface of the solver `Cplex 12.9` to solve $(MIQP^*)$.
- ii) Lex + QCR and Lex + MIQCR: we used the Lex quadratization, followed by the original QCR [12] or MIQCR [9,10]

iii) **Baron** 17.4.1 [37] and **Scip** 6.0.2 [1] both with sub-solver **Cplex** 12.9, and the **gams** interface. We also tried to run the instances with **Gloptipoly** [21], but the it fails even for the smaller/sparser considered instances.

Our experiments were carried out on a server with 2 CPU Intel Xeon each of them having 12 cores and 2 threads of 2.5 GHz and 4*16 GB of RAM using a Linux operating system.

4.1 The image restoration problem

The goal of the *image restoration* problem is to reconstruct an original sharp base image from a blurred image. An image is a rectangle containing $n = l \times h$ pixels. This rectangle is modeled as a binary matrix of the same dimension. The problem can be written as the minimization of a degree 4 polynomial of binary variables where each variable represents a pixel. The coefficients of the monomials are indicative of how likely a configuration is to appear on the sharp base image. The size of the considered instances are $l \times h = 10 \times 10$, 10×15 , and 15×15 , or in the polynomial formulation $n = 100, 150$ and 225 , with a number of monomials of $m = 668, 1033$, and 1598 respectively. In our experiments, 15 instances of each size are considered obtaining a total of 45 instances. Observe that the 15 instances of the same size have identical monomials with different coefficients. The name of each *image restoration* instance describes its characteristics **im.l.h.r**: **l** is the number of lines, **h** is the number of columns and **r** is the index of the instance with the same characteristics.

We run these instances with the solvers **Baron** 17.4.1 and **Scip** 6.0.2 and show that, only **PQCR** is able to solve the whole set of 45 instances within the time limit of one hour. We report the results in Table 1. Column Gap_i is the initial gap of method **PQCR**, i.e. $Gap_i = \left| \frac{BKN - LB_i}{BKN} \right| * 100$, where BKN is the optimal solution value, and LB_i is the root bound value (from Theorem 1 it is also $v(SDP^Z)$). Column $Time$ reports the CPU time in seconds or the final gap (Gap_f) if the time limit of one hour is reached, where $Gap_f = \left| \frac{BKN - LB}{BKN} \right| * 100$, with LB the final lower bound. Column $Nodes$ is the number of nodes visited by the branch-and-bound algorithm. We observe that the solvers **Baron** and **Scip** are significantly faster than **PQCR** on the medium size instances ($n = 100$ or 150), but fail in solving most of the larger instances within the time limit. Indeed, for $n = 225$, **Baron** solves 7 instances, and **Scip** 9 instances out of 15. The final gaps (after one hour of computation) of **Baron** and **Scip** are about 193 and 1631, respectively, times larger than the initial gap of **PQCR** equals to 0.27% on average. We mention that the reformulation time that corresponds to the solution of (SDP^Z) represents most of the CPU time, since the CPU time for solving $(MIQP^*)$ is always smaller than 600 seconds.

We now focus on the comparison with convexifications **QCR** or **MIQCR** applied to $(MIQP^*)$. Our observations are summed up in Tables 2 and 3, where each line corresponds to the average of the instances with the same sizes. In Column $\#var$ QP we report the number of variables of the equivalent convex reformulations of each method, and Column $\#cont$ SDP the number of constraints considered in the

Instance		PQCR			Baron 17.4.1		Scip 6.0.2	
Name	n	Gap _i	Time	Nodes	Time	Nodes	Time	Nodes
im.10.10.1	100	0.84	122	272	1	1	98	2800
im.10.10.2	100	0.47	125	65	1	1	73	2825
im.10.10.3	100	0.13	84	0	1	1	35	961
im.10.10.4	100	0.33	89	6	1	1	28	759
im.10.10.5	100	0.40	91	8	1	1	38	1046
im.10.10.6	100	0.12	120	57	1	1	72	2007
im.10.10.7	100	0.05	115	32	1	1	76	1580
im.10.10.8	100	1.69	103	45	1	1	37	1067
im.10.10.9	100	3.34	111	108	1	1	35	1084
im.10.10.10	100	4.04	134	151	1	1	36	1074
im.10.10.11	100	0.79	127	106	3	1	83	3548
im.10.10.12	100	0.21	131	76	2	1	135	6325
im.10.10.13	100	0.00	79	0	1	1	28	954
im.10.10.14	100	0.04	79	0	1	1	25	722
im.10.10.15	100	0.06	83	0	1	1	28	807
Average		0.83	106.2	62	1.0	1	55.2	1837
im.10.15.1	150	0.55	536	492	3	1	318	6349
im.10.15.2	150	0.13	502	8	4	1	311	4565
im.10.15.3	150	0.02	406	0	1	1	226	5538
im.10.15.4	150	0.71	439	53	2	1	135	2566
im.10.15.5	150	0.06	408	10	1	1	163	3291
im.10.15.6	150	0.83	567	649	5	1	665	13927
im.10.15.7	150	0.07	511	33	4	1	396	5884
im.10.15.8	150	1.85	458	70	1	1	197	3894
im.10.15.9	150	2.24	421	50	2	1	167	4835
im.10.15.10	150	2.09	432	53	2	1	155	2643
im.10.15.11	150	0.12	552	73	5	1	357	8045
im.10.15.12	150	0.62	520	111	21	1	388	9784
im.10.15.13	150	0.06	367	0	1	1	103	1924
im.10.15.14	150	0.13	392	0	1	1	118	2397
im.10.15.15	150	0.01	370	0	1	1	91	1757
Average		0.63	458.7	107	3.6	1	252.6	5160
im.15.15.1	225	0.32	2622	1046	(42.50%)	39	(149.03%)	75251
im.15.15.2	225	0.61	2661	2586	(70.75%)	42	(235.98%)	107023
im.15.15.3	225	0.09	1928	16	(6.68%)	45	404	5001
im.15.15.4	225	0.01	1886	13	(34.08%)	39	738	6688
im.15.15.5	225	0.04	1874	13	736	7	294	2915
im.15.15.6	225	0.30	2263	597	(51.78%)	44	131.61%	76508
im.15.15.7	225	0.46	2235	1458	(73.19%)	49	(825.55%)	339448
im.15.15.8	225	0.52	2017	38	2437	35	566	5357
im.15.15.9	225	0.44	2008	33	1868	27	575	7444
im.15.15.10	225	0.48	1972	33	909	13	393	4159
im.15.15.11	225	0.36	2180	150	(64.96%)	49	(1133.55%)	522819
im.15.15.12	225	0.34	2256	560	(74.77%)	53	(162.00%)	131142
im.15.15.13	225	0.05	1700	0	213	3	364	4136
im.15.15.14	225	0.08	1834	0	750	11	505	5113
im.15.15.15	225	0.02	1750	0	16	1	249	3117
Average		0.27	2079.1	436	989.8 (7)	14 (7)	454.2 (9)	4881 (9)

Table 1: Comparison of PQCR, Scip 6.0.2 and Baron 17.4.1 for the image restoration instances. Time limit 1 hour.

semi-definite program solved to compute the best reformulation. Column Gap_i is the initial gap, and Column $Time$ is the total CPU times in seconds necessary to solve the instances. If the time limit of one hour is reached, we report the final gap (Gap_f). The most striking observation is that MIQCR is unable to handle these instances. We also remark that QCR has a very poor bound at the root node of the branch-and-bound. The weakness of this bound leads to combinatorial explosion and makes it impracticable to resolve these instances with QCR.

Instance			PQCR		Lex+QCR		Lex+ MIQCR	
Name	n	N	#var QP	#cont SDP	#var QP	#cont SDP	#var QP	#cont SDP
im.10.10	100	352	352	3550	352	352	62128	496672
im.10.15	150	542	542	5610	542	542	147153	1176682
im.15.15	225	827	827	8740	827	827	342378	2738197

Table 2: Comparison of the number of variables of the equivalent formulation, and of the number of constraints of the associated semi-definite relaxation for 3 methods PQCR, Lex+QCR, and Lex+MIQCR on the image restoration instances.

Instance			PQCR		Lex+QCR	
Name	n	N	Gap _i	Time	Gap _i	Time
im.10.10	100	352	0.83	106	812.13	(477%)
im.10.15	150	542	0.63	459	794.6	(602%)
im.15.15	225	827	0.27	2079	717.07	(616%)

Table 3: Comparison of the 2 methods PQCR and Lex+QCR for the image restoration instances: initial gaps and time criterion - time limit 1 hour. Lex+MIQCR cannot handle these instances.

4.2 The Low Auto-correlation Binary Sequence problem

The problem is to find binary sequences with low off-peak auto-correlations. More formally, let S be a sequence $S = (s_1, \dots, s_n)$ with $s_i \in \{-1, 1\}$, and for a given $k = 1, \dots, n-1$, we define the auto-correlation $C_k(S)$ of S :

$$C_k(S) = \sum_{i=1}^{n-k} s_i s_{i+k}$$

The problem is to find a sequence S of length n that minimizes $E(S)$, a degree 4 polynomial:

$$E(S) = \sum_{k=1}^{n-1} C_k^2(S)$$

This problem has numerous practical applications in communication engineering, or theoretical physics [7]. For our experiments, we consider truncated instances, i.e. sequences of length n where we compute low off-peak auto-correlation up to a certain distance $n_0 \leq n$, i.e. we consider the following function to minimize:

$$E_{n_0}(S) = \sum_{k=1}^{n_0-1} C_k^2(S)$$

In order to apply PQCR, we convert the variables from $\{-1, 1\}$ to $\{0, 1\}$ using the standard transformation $x = \frac{s+1}{2}$. This problem admits a lot of symmetries. In particular the correlations C_k are identical for a sequence S and its complement. We exploited this symmetry by fixing to 0 the variable that appears the most. Each instance is labeled **b.n.n₀**. These instances were introduced by [30] and can be found on the MINLPLib [31] website.

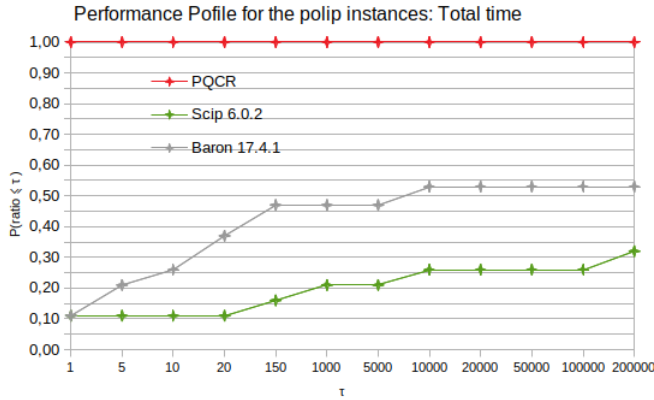


Fig. 1: Performance profile of the CPU times between PQCR, Baron 17.4.1 and Scip 6.0.2 for the LABS instances - Time limit 3 hours

We do not report the results for methods Lex+QCR and Lex+MIQCR since they have failed to solve all the considered instances.

In Figure 1, we present the performance profile of the CPU times for PQCR, Baron 17.4.1 and Scip 6.0.2 over the 19 LABS instances solved within the time limit of 3 hours by at least one method. The detailed results are reported in Table 5 (Appendix). In this profile we can see that PQCR outperforms the two solvers both in terms of the total CPU time and of the number of instances solved. We observe that Baron solves 10 instances, Scip solves 6 instances, while PQCR solves 19 instances out of 45 within the time limit of 3 hours. We mention that the number of valid equalities generated by PQCR in (SDP^Z) can be more than one million for the largest/densest instances (1484800 for b.60.60). Moreover, the average final gaps of Baron and Scip have the same order of magnitude (842% and 881%, respectively), while the one of PQCR is much smaller (3.62% on average).

Last, we increase the time limit for PQCR and compare our results to the best lower bounds and the best known solution values reported in MINLPLib. We present in Table 4 the values of the best solutions (BKN) and of the final lower bounds ($BestLB$) obtained by PQCR, and those available on the MINLPLib website. Each line corresponds to one instance stated as unsolved on MINLPLib. For MINLPLib, Columns BKN and $BestLB$ are the best upper and lower bounds, respectively, among the results of the solvers Antigone, Baron, Couenne, Lindo, and Scip. We also report the final gaps in Column $Gap_f = \left| \frac{BKN - BestLB}{BKN} \right| * 100$. In the MINLPLib website, the time limit is not reported. For method PQCR, we limit the time for steps 2 and 3 of Algorithm 1 to 5 hours each. We mention that the time for the generation of the valid equalities of (SDP^Z) can be significant for the larger/denser instances. We observe that PQCR solves to optimality 10 unsolved instances (labeled as **). It also improves the best known solution values of 13 instances (labeled as #), and improves the final dual bounds of all the unsolved instances (labeled as *). Column Imp is the improvement factor of PQCR.

Instance	MINLPLib[31]			PQCR			
	<i>BKN</i>	<i>BestLB</i>	<i>Gap_f</i>	<i>BKN</i>	<i>BestLB</i>	<i>Gap_f</i>	<i>Imp.</i>
b.25.19**	-14644	-16108	10.00	-14644	-14644.00		
b.25.25**	-10664	-12494	17.16	-10664	-10664.00		
b.30.15**	-15744	-19780	25.64	-15744	-15744.00		
b.30.23**	-30420	-72030	136.79	-30460	-30460.00		
b.30.30**	-22888	-54014	135.99	-22888	-22888.00		
b.35.09**	-5108	-6312	23.57	-5108	-5108.00		
b.35.18**	-31160	-74586	139.36	-31168	-31168.00		
b.35.26#*	-55184	-191466	246.96	-55288	-55484.64	0.36	694
b.35.35#*	-41068	-290424	607.18	-41068	-41730.96	1.61	376
b.40.10**	-8240	-14618	77.40	-8248	-8248.00		
b.40.20#*	-50516	-162365	221.41	-50576	-51248.66	1.33	166
b.40.30#*	-94768	-398617	320.62	-94952	-102375.97	7.82	41
b.40.40*	-67964	-302028	344.39	-67928	-78364.82	15.36	22
b.45.11**	-12740	-30771	141.53	-12748	-12748.00		
b.45.23#*	-85248	-320397	275.84	-85424	-88547.05	3.66	75
b.45.34*	-152368	-752427	393.82	-152248	-164316.84	7.93	50
b.45.45*	-112764	-685911	508.27	-112568	-142414.81	26.51	19
b.50.06#*	-2160	-2921	35.23	-2160	-2199.87	1.85	19
b.50.13**	-23772	-74768	214.52	-23792	-23792.00		
b.50.25#*	-124748	-562446	350.87	-124948	-138994.51	11.24	31
b.50.38#*	-232496	-1318325	467.03	-232664	-270363.98	16.20	29
b.50.50*	-168216	-1173058	597.35	-167824	-246094.07	46.64	13
b.55.06#*	-2400	-3439	43.29	-2400	-2460.02	2.50	17
b.55.14#*	-33168	-116748	251.99	-33272	-33717.52	1.34	188
b.55.28#*	-190472	-989145	419.31	-190696	-214840.20	12.66	33
b.55.41*	-337388	-2494477	639.35	-335840	-474910.72	41.41	15
b.55.55*	-241912	-1947633	705.10	-241780	-302301.26	25.03	28
b.60.08#*	-6792	-13915	104.87	-6792	-7008.60	3.19	33
b.60.15#*	-44896	-169767	278.13	-45232	-46160.31	2.05	136
b.60.30*	-261048	-1491016	471.17	-260304	-324244.90	24.56	19
b.60.45*	-478528	-3687344	670.56	-476664	-951320.31	99.58	7
b.60.60*	-350312	-3021077	762.40	-349560	-496399.25	42.01	18

Table 4: Comparison of the best known solution and best lower bound values of PQCR and of MINLPLib for the unsolved LABS instances. **: solved for the first time, #: best known solution improved, and *: best known lower bound improved

5 Conclusion

We consider the general problem (UBP) of minimizing a multi-linear function where the variables are binary. In this paper, we present PQCR a solution approach for (UBP). We start by a quadratization \mathcal{Z} , where we rewrite (UBP) as an equivalent quadratic program ($MIQP^{\mathcal{Z}}$). Then, by introducing new valid quadratic equalities, we build a family of tailored quadratic convex reformulations of ($MIQP^{\mathcal{Z}}$) that exploits its specific structure. We further focus on finding, within this family, the

best equivalent convex formulation ($MIQP^*$) from the continuous relaxation point of view. We show that it can be computed by solving a semi-definite relaxation of ($MIQP^Z$). Finally, we solve ($MIQP^*$) with a quadratic convex programming solver. A future research direction would be to characterize which quadratization best fit with our convexification phase from the continuous relaxation value point of view.

Acknowledgements The authors are thankful to Elisabeth Rodriguez-Heck and Yves Crama for useful discussions. This work was supported by a public grant as part of the Investissement d’avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH, in a joint call with Gaspard Monge Program for optimization, operations research and their interactions with data sciences.

Appendix

References

1. T. Achterberg. Scip : solving constraint integer programs. *Mathematical Programming Computation*, (1):1–41, 2009.
2. C.S. Adjiman, S. Dallwig, C.A. Floudas, and A. Neumaier. A global optimization method, α bb, for general twice-differentiable constrained nlp*s*—i. theoretical advances. *Computers and Chemical Engineering*, 22(9):1137–1158, 1998.
3. A. A. Ahmadi and A. Majumdar. Dsos and sdsos optimization: Lp and socp-based alternatives to sum of squares optimization. In *2014 48th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–5, March 2014.
4. M. Anthony, E. Boros, Y. Crama, and A. Gruber. Quadratic reformulations of nonlinear binary optimization problems. *Mathematical Programming*, 162:115–144, 2017.
5. B. Balasundaram and A.O. Prokopyev. On characterization of maximal independent sets via quadratic optimization. *Journal of Heuristics - HEURISTICS*, 19, 06 2011.
6. P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex minlp. *Optimization Methods and Software*, 4–5(24):597–634, 2009.
7. J. Bernasconi. Low autocorrelation binary sequences: statistical mechanics and configuration space analysis. *J. Physique*, 141(48):559–567, 1987.
8. A. Billionnet and S. Elloumi. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Mathematical Programming*, 109(1):55–68, 2007.
9. A. Billionnet, S. Elloumi, and A. Lambert. Extending the QCR method to the case of general mixed integer program. *Mathematical Programming*, 131(1):381–401, 2012.
10. A. Billionnet, S. Elloumi, and A. Lambert. Exact quadratic convex reformulations of mixed-integer quadratically constrained problems. *Mathematical Programming*, 158(1):235–266, 2016.
11. A. Billionnet, S. Elloumi, A. Lambert, and A. Wiegele. Using a Conic Bundle method to accelerate both phases of a Quadratic Convex Reformulation. *INFORMS Journal on Computing*, 29(2):318–331, 2017.
12. A. Billionnet, S. Elloumi, and M. C. Plateau. Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method. *Discrete Applied Mathematics*, 157(6):1185 – 1197, 2009.
13. E. Boros, P.L. Hammer, and X. Sun. Network flows and minimization of quadratic pseudo-boolean functions. Technical Report TR: 1991-17, RUTCOR, 1991.
14. C. Buchheim and C. D’Ambrosio. Monomial-wise optimal separable underestimators for mixed-integer polynomial optimization. *Journal of Global Optimization*, pages 1–28, 2016.
15. C. Buchheim and G. Rinaldi. Efficient reduction of polynomial zero-one optimization to the quadratic case. *SIAM Journal on Optimization*, 18(4):1398–1413, 2007.
16. Y. Crama and E. Rodriguez-Heck. A class of valid inequalities for multilinear 0-1 optimization problems. *Discrete Optimization*, pages 28–47, 2017.
17. S. Elloumi and A. Lambert. Global solution of non-convex quadratically constrained quadratic programs. *Optimization Methods and Software*, 34(1):98–114, 2019.

Instance		PQCR			Scip 6.0.2			Baron 17.4.1		
Name	BKN	Best	Nodes	Time	Best	Nodes	Time	Best	Nodes	Time
b.20.03	-72	-72	0	0	-72	1	0	-72	1	0
b.20.05	-416	-416	1201	0	-416	131731	191	-416	1	3
b.20.10	-2936	-2936	4388	34	-2936	267990	1087	-2936	7	193
b.20.15	-5960	-5960	60274	3435	-5904	9	(1556.9%)	-5952	5278	(227.9%)
b.25.03	-92	-92	0	0	-92	1	0	-92	1	0
b.25.06	-960	-960	43883	6	-960	3286874	5154	-960	37	107
b.25.13	-8148	-8148	20843	3665	-8148	1170781	(165.2%)	-8148	61	3706
b.25.19	-14644	-14644	74440	8788	-14624	689115	(168.6%)	-14644	31	(33.58%)
b.25.25	-10268	-10268	60475	4832	-10660	446419	(367.3%)	-10656	19	(54.0%)
b.30.04	-324	-324	2079	1	-324	12964181	9179	-324	19	13
b.30.08	-2952	-2952	235867	857	-2952	4492349	(280.8%)	-2952	217	2665
b.30.15	-15744	-15744	174797	7456	-15648	911094	(262.3%)	-15668	21	(62.9%)
b.30.23	-30460	-30460	527903	8118	-30460	365485	(518.8%)	-30240	5	(158.1%)
b.30.30	-22888	-22888	201710	9838	-22728	120753	(644.9%)	-22840	3	(174.8%)
b.35.04	-384	-384	7846	2	-384	15024343	(110.5%)	-384	25	(44.8%)
b.35.09	-5108	-5108	1989577	7833	-5056	1458986	(615.3%)	-5108	199	(30.2%)
b.35.18	-31168	-31168	2337227	(0.3%)	-31072	308769	(472.0%)	-30960	4	(158.7%)
b.35.26	-55288	-55240	60441	(2.8%)	-53984	5765	(963.5%)	-54872	1	(262.9%)
b.35.35	-41068	-41052	19765	(5.2%)	ME	ME	ME	-40640	8	(637.5%)
b.40.05	-936	-936	275307	34	-932	9337240	(325.6%)	-936	1019	3837
b.40.10	-8248	-8248	11039761	10550	-8120	613581	(664.2%)	-8200	29	(79.9%)
b.40.20	-50576	-50576	434637	(3.9%)	-49364	30692	(681.8%)	-50140	1	(212.3%)
b.40.30	-94952	-94600	-	-	-93560	9402	(1012.4%)	-93816	1	(426.2%)
B.40.40	-67964	-67912	7787	(10.6%)	-64076	1432	(1416.2%)	-60636	1	(610.3%)
b.45.05	-1068	-1068	851051	67	-1048	8467858	(378.6%)	-1064	2444	(6.7%)
b.45.11	-12748	-12748	5925351	(1.3%)	-12428	729065	(748.7%)	-12600	12	(149.6%)
b.45.23	-85424	-85192	11046	(4.9%)	-82252	4275	(1036.2%)	-84704	1	(275.1%)
b.45.34	-152368	-151240	-	-	-147144	11512	(1323.1%)	0	1	(566.3%)
b.45.45	-112764	-111276	-	-	-106940	5365	(1398.3%)	0	1	(3539.5%)
b.50.06	-2160	-2160	25015489	(2.8%)	-2128	4012214	(590.5%)	-2160	688	(33.6%)
b.50.13	-23792	-23788	2114159	(1.7%)	-23324	46707	(1073.2%)	-23152	4	(213.6%)
b.50.25	-124948	-124144	-	-	-118268	2366	(1200.8%)	-123628	-	(322.5%)
b.50.38	-232664	-231560	-	-	-221192	55	(1729.0%)	0	1	(692.3%)
b.50.50	-168216	-163912	-	-	-157512	1215	(1601.6%)	0	1	(3587.8%)
b.55.06	-2400	-2400	20567148	(3.1%)	-2400	3824456	(620.1%)	-2400	461	(45.6%)
b.55.14	-33272	-33240	98692	(3.7%)	-23736	38	(1515.5%)	-32368	3	(246.2%)
b.55.28	-190696	-189668	-	-	-184316	890	(1497.4%)	-188556	1	(413.5%)
b.55.41	-337388	-	-	-	ME	ME	ME	0	1	(3659.5%)
b.55.55	-241912	-239052	-	-	ME	ME	ME	0	2	(3628.2%)
b.60.08	-6792	-6792	6664937	(3.2%)	-6548	902923	(942.0%)	-6728	45	(103.8%)
b.60.15	-45232	-44248	-	-	-41132	38	(1552.6%)	-44608	1	(266.2%)
b.60.30	-261048	-258856	-	-	-236808	222	(1635.4%)	-255552	1	(424.7%)
b.60.45	-478528	-	-	-	ME	ME	ME	0	2	(3687.0%)
b.60.60	-350312	-349940	-	-	ME	ME	ME	0	2	(3629.6%)

Table 5: Results of PQCR, Scip 6.0.2 and Baron 17.4.1 for the 45 LABS instances. Column BKN is the best know solution (reported in the MINLPLib website [31] or found by PQCR). In Columns Best, we report the best solution found by each solver within the time limit, where ME means that the solution fails because of a memory error, and for method PQCR symbol - means that the time limit of 3 hours was too small to start the solution of (SDP^Z), or/and of (MIQP*).

18. M.R. Garey and D.S. Johnson. Computers and Intractability: A guide to the theory of NP-Completeness. *W.H. Freeman, San Francisco, CA*, 1979.
19. B. Ghaddar, J. C. Vera, and M. F. Anjos. A dynamic inequality generation scheme for polynomial programming. *Mathematical Programming*, 156(1):21–57, Mar 2016.
20. C. Helmberg. *Conic Bundle v0.3.10*, 2011.
21. D. Henrion and J. B. Lasserre. Gloptipoly: Global optimization over polynomials with matlab and sedumi. *ACM Transactions on Mathematical Software*, 29(2):165–194, 2003.

22. N. Ito, S. Kim, and M. Kojima nad A.Takeda K.C. Toh. BBCPOP: A Sparse Doubly Nonnegative Relaxation of Polynomial Optimization Problems with Binary, Box and Complementarity Constraints. *ArXiv e-prints*, April 2018.
23. R.M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations*, pages 85–103, 1972.
24. J. Krarup and P.M. Pruzan. Computer-aided layout design. *Mathematical Programming in Use*, pages 75–94, 1978.
25. X. Kuang, , B. Ghaddar, J . Naoum-Sawaya, and L.F. Zuluaga. Alternative SDP and SOCP Approximations for Polynomial Optimization. *ArXiv e-prints*, October 2015.
26. J.B. Lasserre. *An Introduction to Polynomial and Semi-Algebraic Optimization*. Cambridge University Press, Cambridge, 2015.
27. J.B. Lasserre and T.P. Thanh. Convex underestimators of polynomials. *Journal of Global Optimization*, pages 1–25, 2013.
28. J.D. Laughhunn. Quadratic binary programming with applications to capital budgeting problems. *Operations Research*, 18:454–461, 06 1970.
29. C. Lemarechal and F. Oustry. Semidefinite relaxations and lagrangian duality with application to combinatorial optimization. Technical report, RR-3710, INRIA Rhones-Alpes, 1999.
30. F. Liers, E. Marinari, U. Pagacz, F. Ricci-Tersenghi, and V. Schmitz. A non-disordered glassy model with a tunable interaction range. *Journal of Statistical Mechanics: Theory and Experiment*, page L05003, 2010.
31. MINLPLib. Library of mixed integer non linear programs. "<http://www.minlplib.org/>", 2012.
32. R. Misener and C.A. Floudas. Antigone: algorithms for continuous/integer global optimization of nonlinear equations. *Journal of Global Optimization*, 59(2-3):503–526, 2014.
33. Y. Nesterov and A. Nemirovsky. Conic formulation of a convex programming problem and duality. *Optimization Methods and Software*, 1(2):95–115, 1992.
34. P.M Pardalos and J. Xue. The maximum clique problem. *Journal of Global Optimization*, 4(3):301–328, Apr 1994.
35. M.R. Rao. Cluster analysis and mathematical programming. *Journal of the American Statistical Association*, 66(335):622–626, 1971.
36. J.M.W. Rhys. A selection problem of shared fixed costs and network flows. *Management Science*, 17(3):200–207, 1970.
37. N.V. Sahinidis and M. Tawarmalani. Baron 9.0.4: Global optimization of mixed-integer nonlinear programs. *User's Manual*, 2010.
38. H.D. Sherali and C.H. Tuncbilek. A global optimization algorithm for polynomial programming using a reformulation-linearization technique. *Journal of Global Optimization*, 2:101–112, 1992.