Solving unconstrained 0-1 polynomial programs through quadratic convex reformulation

Sourour Elloumi, Amélie Lambert, Arnaud Lazare

Solving unconstrained 0-1 polynomial programs through quadratic convex reformulation

Sourour Elloumi[1], Amélie Lambert[2] and Arnaud Lazare[1]

1 CEDRIC-ENSTA, 828 Boulevard des Maréchaux, 91120 Palaiseau, France
{sourour.elloumi,arnaud.lazare}@ensta-paristech.fr
2 CEDRIC-CNAM, 292 rue saint Martin, F-75141 Paris Cedex 03, France
amelie.lambert@cnam.fr

Abstract We propose a solution approach for the problem (P) of minimizing an unconstrained binary polynomial optimization problem. We call this method `PQCR` (`Polynomial Quadratic Convex Reformulation`). The resolution is based on a 3-phase method. The first phase consists in reformulating (P) into a quadratic program (QP). For this, we recursively reduce the degree of (P) to two, by use of the standard substitution of the product of two variables by a new one. We then obtain a linearly constrained binary program. In the second phase, we rewrite the quadratic objective function into an equivalent and parametrized quadratic function using the equality $x_i^2 = x_i$ and new valid quadratic equalities. Then, we focus on finding the best parameters to get a quadratic convex program which continuous relaxation's optimal value is maximized. For this, we build a semidefinite relaxation (SDP) of (QP). Then, we prove that the standard linearization inequalities, used for the quadratization step, are redundant in (SDP) in presence of the new quadratic equalities. Next, we deduce our optimal parameters from the dual optimal solution of (SDP). The third phase consists in solving (QP^*), the optimal reformulated problem, with a standard solver. In particular, at each node of the branch-and-bound, the solver computes the optimal value of a continuous quadratic convex program. We present computational results on instances of the *image restoration problem* and of the *low autocorrelation binary sequence problem*. We compare `PQCR` with other convexification methods, and with the general solver `Baron` 17.4.1 [39]. We observe that most of the considered instances can be solved with our approach combined with the use of `Cplex` [24].

Key words: Unconstrained binary polynomial programming, Global optimization, Semidefinite programming, Quadratic convex reformulation, Experiments

1 Introduction

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

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

where $I = \{1, .., n\}$, $f(x)$ is an $n-$variable polynomial 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 (P) is a quadratic function has been widely studied. In this case, (P) has many applications, including those from financial analysis [31], cluster analysis [37], computer aided design [27] or machine scheduling [38]. 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 problem [6,14,36]. In the cubic case, the important class of satisfiability problems known as 3-SAT, can be formulated as (P) [26]. In the case where $d \geq 3$, there also exists many applications including, for example: the construction of binary sequences with low aperiodic correlation [8] that is one of the most challenging problems in signal design theory, or the image restoration problem in computer vision [17].

Because of the non-convexity of $f(x)$ and of the integrality of its variables (P) is $\mathcal{NP}$-hard [20]. During the last decade, several algorithms that can handle (P) were introduced. In particular, the methods that 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 (P). More precisely, in a first step a convex relaxation is designed and then a branch-and-bound is performed based on this relaxation. The most classical relaxation consists in the complete linearisation of (P), but quadratic convex relaxation can also be used. For instance, the well known $\alpha-$branch-and-bound [3] computes convex underestimators 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` [39], `Antigone` [35], `SCIP` [2] or `Couenne` [7].

In the case where the objective is a polynomial, but the variables are continuous, Lasserre proposes in [29] an algorithm based on a hierarchy of semidefinite relaxations of (P). The idea is, at each rank of the hierarchy, to successively tighten semidefinite relaxations of (P) in order to reach its optimal solution value. It is also proven in [29] that this hierarchy converges in a finite number of iterations to the optimal solution of the considered problem. Further, this work has been extended to hierarchies of second order conic programs [4,21,28], and of sparse doubly nonnegative relaxation [25]. Although these algorithms were not originally tailored for binary programming, they can handle (P) by considering

the quadratic constraint $x_i^2 = x_i$. Methods devoted to the binary polynomial case were also proposed. In [15,30], the authors use separable or convex underestimators to approximate a given polynomial. Other methods based on linear reformulations can be found in [17,18,40], in which linear equivalent formulations to (P) are proposed. Another approach proposed in [5] consists in building a quadratic equivalent formulation to (P) where the objective function is still non-convex. The obtained reformulation is then solve by the solver `Cplex` [24].

In this paper, we focus on finding equivalent quadratic convex formulations of (P). `QCR (Quadratic Convex Reformulation)` methods [9,10] were introduced for the specific case where $d = 2$. The idea of these approaches is to build tight equivalent reformulations to (P) that have a convex objective function. This equivalent problem can be computed using the dual solution of a semidefinite relaxation of (P), and further solved by a branch-and-bound algorithm based on quadratic convex relaxation. Here, we consider the more general case where $d \geq 3$, and we propose to compute an equivalent convex formulation to (P). Hence, we present an exact solution method for problem (P) that can be split in three phases. The first phase consists in building an equivalent formulation to (P) where both objective function and constraints are at most quadratic. For this, we need to add some auxiliary variables. We then obtain problem (QP) that has a quadratic objective function and linear inequalities. Following [5], we call it *quadratization*. Then in the second phase, we focus on the convexification of the obtained problem. As illustrated in the experimentations, the original `QCR` and `MIQCR` method are not able to handle (QP). Indeed, `QCR` leads to a reformulation with a weak bound, and in method `MIQCR` the semidefinite program that we have to solve is too large. This is why, in this paper we propose a tailored convexification phase. For this, we need null quadratic functions on the domain of (QP) to perturb the Hessian matrix of the new objective function. One of these functions is the classic binary identity, $x_i^2 = x_i$. One contribution of this paper is the introduction of new null quadratic functions on the domain of (QP). We thus get a family of convex equivalent formulations to (QP) that depend on four parameters α, β, δ and λ. We then want to choose α^*, β^*, δ^* and λ^* such that the continuous relaxation bound of the convexified problem is maximized. We show that these best parameters can be computed thanks to a semidefinite program. Finally, the last phase consists in solving the convexified problem using general-purpose optimization software.

The outline of the paper is the following. In Section 2, we define and present our quadratizations of (P). 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 and we discuss about different possible quadratizations of (P). Section 5 draws a conclusion.

2 Phase 1: Quadratization of (P)

In this section, we present how we build equivalent quadratic formulations to (P). The basic idea is to reduce the degree of f to degree 2. For this, in each

monomial of degree 3 or greater, we simply recursively replace each product of two variables by an additional variable.

More formally, we define the set of indices of the additional variables $J = \{n+1, .., N\}$, where N is the total number of initial and additional variables. We also define the subsets $\mathcal{E}_i$ for the initial or additional variable i as follows:

Definition 1. *For all $i \in I \cup J$, we define $\mathcal{E}_i$ as the set of indices of the variables whose product is replaced by x_i:*

- *If $i \in I$, i.e. x_i is an initial variable, we set $\mathcal{E}_i = \{i\}$*
- *If $i \in J$, i.e. x_i is an additional variable, there exist $(i_1, i_2) \in (I \cup J)^2$ such that x_i replaces $x_{i_1} x_{i_2}$ and we set $\mathcal{E}_i = \mathcal{E}_{i_1} \cup \mathcal{E}_{i_2}$*

Using these sets, we define a *valid* quadratization as a reformulation with N variables where any monomial of degree at least 3 is replaced by the product of two variables.

Definition 2. *The sets $J = \{n+1, .., N\}$ and $\{\mathcal{E}_i,\ i \in I \cup J\}$ define a valid quadratization with N variables if for any monomial p of degree greater than or equal to 3, (i.e. $|\mathcal{M}_p| \geq 3$), there exist $(j, k) \in (I \cup J)^2$ 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.*

With this definition of a quadratization, we reformulate (P) as a non-convex quadratically constrained quadratic program $(QCQP)$ with N variables.

$$(QCQP) \begin{cases} \min g(x) = \displaystyle\sum_{\substack{|\mathcal{M}_p| \geq 3 \\ \mathcal{M}_p = \mathcal{E}_j \cup \mathcal{E}_k}} c_p x_j x_k + \sum_{|\mathcal{M}_p| \leq 2} c_p \prod_{i \in \mathcal{M}_p} x_i \\ \text{s.t.} \\ \quad x_i = x_{i_1} x_{i_2} \quad \forall (i, i_1, i_2) \in J \times (I \cup J)^2 :\ \mathcal{E}_i = \mathcal{E}_{i_1} \cup \mathcal{E}_{i_2} \quad (1) \\ \quad x \in \{0,1\}^N \end{cases}$$

As the variables are binary, Constraints (1) are equivalent to the classical set of Fortet inequalities [18]:

$$(C^i_{i_1,i_2}) \begin{cases} x_i - x_{i_1} \leq 0, \\ x_i - x_{i_2} \leq 0, \\ -x_i + x_{i_1} + x_{i_2} \leq 1, \\ -x_i \leq 0, \end{cases}$$

We now define set $\mathcal{F}_{\mathcal{E}}$:

$$\mathcal{F}_{\mathcal{E}} = \{x \in \{0,1\}^N : C^i_{i_1,i_2} \text{ is satisfied } \forall (i, i_1, i_2) \in J \times (I \cup J)^2 :\ \mathcal{E}_i = \mathcal{E}_{i_1} \cup \mathcal{E}_{i_2}\}.$$

We denote by $M = 4(N-n)$ the number of constraints of $\mathcal{F}_{\mathcal{E}}$. We thus obtain the following linearly constrained quadratic formulation that is equivalent to (P) and has N variables and M constraints:

$$(QP)\begin{cases}\min g(x) \equiv x^T Q x + c^t x \\ \text{s.t.} \\ \quad x \in \mathcal{F}_{\mathcal{E}}\end{cases}$$

where $Q \in S_N$ (the set of $N \times N$ real symmetric matrices), and $c \in \mathbb{R}^N$.

An important remark is that several valid quadratizations can be applied to (P), each of them leading to different sets $\mathcal{E}_i$. Different valid quadratizations were introduced and compared in [5]. In our case the comparison criterion is different, and we present a short experimental comparison from the continuous relaxation bound point of view in Section 4.

Example 1 *[Different valid quadratizations]*
Let us consider the following problem:

$$(Ex)\Big\{\min_{x\in\{0,1\}^4} 2x_1 + 3x_2x_3 - 2x_2x_3x_4 - 3x_1x_2x_3x_4$$

For instance, we can build three different equivalent functions:

1. $g_1(x) = 2x_1 + 3x_2x_3 - 2\underbrace{x_2x_4}_{x_5}x_3 - 3\underbrace{x_1x_4}_{x_6}\underbrace{x_2x_3}_{x_7}$
2. $g_2(x) = 2x_1 + 3x_2x_3 - 2\underbrace{x_3x_4}_{x_5}x_2 - 3\underbrace{x_1x_2}_{x_6}\underbrace{x_3x_4}_{x_5}$
3. $g_3(x) = 2x_1 + 3x_2x_3 - 2\underbrace{x_2x_4}_{x_5}x_3 - 3\underbrace{x_1x_2}_{x_6}\underbrace{x_3x_4}_{x_7}$

$$(QEx_1)\begin{cases}\min g_1(x) \\ s.t. \\ \quad (x_2,x_4,x_5) \in C^5_{2,4} \\ \quad (x_1,x_4,x_6) \in C^6_{1,4} \\ \quad (x_2,x_3,x_7) \in C^7_{2,3} \\ \quad x \in \{0,1\}^7\end{cases} (QEx_2)\begin{cases}\min g_2(x) \\ s.t. \\ \quad (x_3,x_4,x_5) \in C^5_{3,4} \\ \quad (x_1,x_2,x_6) \in C^6_{1,2} \\ \quad x \in \{0,1\}^6\end{cases} (QEx_3)\begin{cases}\min g_3(x) \\ s.t. \\ \quad (x_2,x_4,x_5) \in C^5_{2,4} \\ \quad (x_1,x_2,x_6) \in C^6_{1,2} \\ \quad (x_3,x_4,x_7) \in C^7_{3,4} \\ \quad x \in \{0,1\}^7\end{cases}$$

Here we obtain 3 different equivalent quadratic formulations to (Ex) with different sets $\mathcal{E}$. They are thus of different sizes: problems (QEx_1) and (QEx_3) have 7 variables and 12 constraints, while problem (QEx_2) has 6 variables and 8 constraints.

□

We have reduced the degree of the polynomial program (P) by building an equivalent quadratic program to (P). However, the solution of (QP) still has two difficulties, the non-convexity of the objective function $g(x)$ and the integrality of the variables. We now consider the solution of problem (QP).

In the state-of-the-art, some solvers can solve (QP) to global optimality (e.g. `Cplex 12.7` [24]). Unfortunately, these solvers are not able to solve dense

instances of (P). Here, we propose to compute an equivalent quadratic convex formulation to (QP). There exists several convexification methods devoted to quadratic programming (see, for example [10,12,16,22,34]). These approaches can be directly applied to (QP). For instance, one can use the `QCR` method, described in [12], that consists in computing an equivalent convex formulation to (QP) using semidefinite programming. The convexification is obtained thanks to a non uniform perturbation of the diagonal of the Hessian matrix. The semidefinite relaxation used can be easily solved due to its reasonable size. However, the bound obtained by continuous relaxation of the reformulation is very weak. As a consequence, the branch-and-bound used to solve the reformulation failed as soon as $n \geq 20$. Another alternative is to apply the `MIQCR` method [10]. In this method, the perturbation is generalized to the whole Hessian matrix and hence is more refined than the previous one. This leads to a reformulation with a significantly sharper bound. Unfortunately, the semidefinite relaxation used in this approach is too large and its computation failed even with instances containing 10 variables. In the next section, we present a new convexification that leads to sharper bounds than `QCR` but with a better tractability than `MIQCR`.

3 Phase 2: A quadratic convex reformulation of (QP)

In this section, we consider the problem of reformulating (QP) by an equivalent quadratic 0-1 program with a convex objective function. To do this, we define a new convex function which value is equal to the value of $g(x)$, but which Hessian matrix is positive semidefinite. More precisely, we first add to $g(x)$ a combination of four sets of functions where each of these functions vanishes on the feasible set $\mathcal{F}_{\mathcal{E}}$. For each function we introduce a scalar parameter. Then we focus on computing the best parameters that lead to a convex function and that maximize the optimal value of the continuous relaxation of the obtained problem.

3.1 Valid quadratic equalities for (QP)

For a quadratization characterized by $\mathcal{E}$, we introduce null quadratic functions over the set $\mathcal{F}_{\mathcal{E}}$.

Lemma 1 *The following quadratic equalities characterize null functions over* $\mathcal{F}_{\mathcal{E}}$*:*

$$\begin{cases} x_i^2 - x_i = 0 & i \in I \cup J \quad (2) \\ x_i - x_i x_j = 0 & (i,j) \in J \times (I \cup J) : \ \mathcal{E}_j \subset \mathcal{E}_i \quad (3) \\ x_i - x_j x_k = 0 & (i,j,k) \in J \times (I \cup J)^2 : \ \mathcal{E}_i = \mathcal{E}_j \cup \mathcal{E}_k \quad (4) \\ x_i x_j - x_k x_l = 0 & (i,j,k,l) \in (I \cup J)^4 : \ \mathcal{E}_i \cup \mathcal{E}_j = \mathcal{E}_k \cup \mathcal{E}_l \quad (5) \end{cases}$$

Proof. Constraints (2) trivially hold since $x_i \in \{0,1\}$. Constraints (4) come from Definition 1. We then prove the validity of the Constraints (3) and (5).

– *Constraints (3):* we have $x_i = \prod_{i' \in \mathcal{E}_i} x_{i'}$ and $x_j = \prod_{j' \in \mathcal{E}_j} x_{j'}$, then:

$$\begin{aligned} x_i x_j &= \prod_{i' \in \mathcal{E}_i} x_{i'} \prod_{j' \in \mathcal{E}_j} x_{j'} \\ &= \prod_{j' \in \mathcal{E}_j} x_{j'}^2 \prod_{i' \in \mathcal{E}_i \setminus \mathcal{E}_j} x_{i'} \text{ since } \mathcal{E}_j \subset \mathcal{E}_i \\ &= \prod_{i' \in \mathcal{E}_i} x_{i'} \text{ since } x_{j'}^2 = x_{j'} \text{ and } \mathcal{E}_j \cup (\mathcal{E}_i \setminus \mathcal{E}_j) = \mathcal{E}_i \\ &= x_i \end{aligned}$$

– *Constraints (5):* by definition we have:

$$\begin{aligned} x_i x_j &= \prod_{i' \in \mathcal{E}_i} x'_i \prod_{j' \in \mathcal{E}_j} x_{j'} \\ &= \prod_{i' \in \mathcal{E}_i \cup \mathcal{E}_j} x_{i'} \prod_{j' \in \mathcal{E}_i \cap \mathcal{E}_j} x_{j'} \\ &= \prod_{i' \in (\mathcal{E}_i \cup \mathcal{E}_j) \setminus (\mathcal{E}_i \cap \mathcal{E}_j)} x_{i'} \prod_{j' \in \mathcal{E}_i \cap \mathcal{E}_j} x_{j'}^2 \\ &= \prod_{i' \in \mathcal{E}_i \cup \mathcal{E}_j} x_{i'} \text{ since } x_{j'}^2 = x_{j'} \text{ and } (\mathcal{E}_i \cup \mathcal{E}_j) \setminus (\mathcal{E}_i \cap \mathcal{E}_j) \cup (\mathcal{E}_i \cap \mathcal{E}_j) = (\mathcal{E}_i \cup \mathcal{E}_j) \\ &= \prod_{k' \in \mathcal{E}_k \cup \mathcal{E}_l} x_{k'} \text{ since } \mathcal{E}_i \cup \mathcal{E}_j = \mathcal{E}_k \cup \mathcal{E}_l \\ &= x_k x_l \end{aligned}$$

□

We can now define set $\mathcal{S}_\mathcal{E}$:

$$\mathcal{S}_\mathcal{E} = \{x \in \{0,1\}^N : \text{Constraints (2)–(5) are satisfied }\}.$$

3.2 An equivalent quadratic convex reformulation to (QP)

We now compute a quadratic convex reformulation of (QP) and thus of (P). For this, we add to the objective function $g(x)$ the equalities (2)–(5). For each, we associate a real scalar parameter: α_i for Constraints (2), β_{ij} for Constraints (3), δ_{ijk} for Constraints (4), and λ_{ijkl} for Constraints (5). We get the following parametrized function:

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

Obviously $g_{\alpha,\beta,\delta,\lambda}(x)$ has the same optimal value as $g(x)$. Moreover, there exist vector parameters α, β, δ and λ such that $g_{\alpha,\beta,\delta,\lambda}(x)$ is a convex function. Take for instance, α equals to the opposite of the smallest eigenvalue of Q, and $\beta = \delta = \lambda = 0$.

By replacing $g(x)$ by the new function, we obtain the quadratic convex equivalent formulation to (QP):

$$(QP_{\alpha,\beta,\delta,\lambda}) \begin{cases} \min g_{\alpha,\beta,\delta,\lambda}(x) \equiv x^T Q_{\alpha,\beta,\delta,\lambda} x + c^T_{\alpha,\beta,\delta,\lambda} x \\ \text{s.t.} \\ \quad x \in \mathcal{F}_{\mathcal{E}} \end{cases}$$

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

In order to use $(QP_{\alpha,\beta,\delta,\lambda})$ within a branch-and-bound procedure, we are interested by parameters (α, β, δ, λ) such that $g_{\alpha,\beta,\delta,\lambda}(x)$ is a convex function. Moreover, in order to have a good behavior of the branch-and-bound algorithm, we want to find parameters that give the tightest continuous relaxation bound. More formally, we want to solve the following optimization problem:

$$(CP): \max_{\substack{\alpha \in \mathbb{R}^N, \beta \in \mathbb{R}^{T^1}, \delta \in \mathbb{R}^{T^2}, \lambda \in \mathbb{R}^{T^3} \\ Q_{\alpha,\beta,\delta,\lambda} \succeq 0}} \left\{ \min_{x \in \overline{\mathcal{F}}_{\mathcal{E}}} g_{\alpha,\beta,\delta,\lambda}(x) \right\}$$

where T^1, T^2 and T^3 are the number of Constraints (3), (4), and (5), respectively, and $\overline{\mathcal{F}}_{\mathcal{E}}$ is the set $\mathcal{F}_{\mathcal{E}}$ where the integrality constraints are relaxed, i.e. $x \in [0,1]^N$.

In the rest of the paper we will focus on solving (CP). For this, we first build a compact semidefinite relaxation that uses our new valid equalities and prove that its optimal dual variables provide an optimal solution to (CP).

3.3 Computing an optimal solution to (CP)

The following theorem shows that problem (CP) is equivalent to the dual of a semidefinite relaxation of (QP).

Theorem 1. *The optimal value of* (CP) *is equal to the optimal value of the following semidefinite program* (SDP)*:*

$$(SDP)\begin{cases} \min < Q, X > + c^T x & \\ s.t. & \\ \quad X_{ii} - x_i = 0 & i \in I \cup J \quad (6)\\ \quad -X_{ij} + x_i = 0 & (i,j) \in J \times (I \cup J) : \ \mathcal{E}_j \subset \mathcal{E}_i \quad (7)\\ \quad -X_{jk} + x_i = 0 & (i,j,k) \in J \times (I \cup J)^2 : \ \mathcal{E}_i = \mathcal{E}_j \cup \mathcal{E}_k \quad (8)\\ \quad X_{ij} - X_{kl} = 0 & (i,j,k,l) \in (I \cup J)^4 : \ \mathcal{E}_i \cup \mathcal{E}_j = \mathcal{E}_k \cup \mathcal{E}_l \quad (9)\\ \quad \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0 & \quad (10)\\ \quad x \in \mathbb{R}^N, \ X \in S^N & \quad (11) \end{cases}$$

The optimal values $(\alpha^*, \beta^*, \delta^*, \lambda^*)$ *of problem* (CP) *are given by the optimal values of the dual variables associated with constraints (6–9) respectively.*

Proof. For simplicity, we rewrite $\mathcal{F}_\mathcal{E}$ and $\mathcal{S}_\mathcal{E}$ as follows: $\mathcal{F}_\mathcal{E} = \{x \in \{0,1\}^N : Ax \leq b\}$ where A is $M \times N$-matrix, $b \in \mathbb{R}^M$, and $\mathcal{S}_\mathcal{E} = \{x \in \{0,1\}^N : x^T Q_r x + c_r^T x = 0,\ r = 1, \ldots, T\}$, where $T = N + T^1 + T^2 + T^3$ is the number of Constraints (2)–(5), and $\forall r = 1, \ldots, T$, $Q_r \in S_N$ and $c_r \in \mathbb{R}^N$.

We start by observing that $x \in [0,1]^N$ is equivalent to $x^2 \leq x$, thus, (CP) is equivalent to $(Q1)$:

$$(Q1) : \max_{\substack{\alpha \in \mathbb{R}^N, \beta \in \mathbb{R}^{T^1}, \delta \in \mathbb{R}^{T^2}, \lambda \in \mathbb{R}^{T^3} \\ Q_{\alpha,\beta,\delta,\lambda} \succeq 0}} \left\{ \min_{x \in \mathbb{R}^N, \ x^2 \leq x, \ Ax \leq b} g_{\alpha,\beta,\delta,\lambda}(x) \right\}$$

$(Q1)$ is a convex optimization problem over a convex set. If we consider the solution $x_i = 0.5$ for all $i \in I$ and $x_i = x_j x_k$ for all $(i,j,k) \in J \times (I \cup J)^2$, $\mathcal{E}_i = \mathcal{E}_j \cup \mathcal{E}_k$, it is an interior point and the Slater's conditions are satisfied. Then, by Lagrangian duality, we have $(Q1)$ equivalent to $(Q2)$:

$$(Q2) : \max_{\substack{\alpha \in \mathbb{R}^N, \beta \in \mathbb{R}^{T^1}, \delta \in \mathbb{R}^{T^2}, \lambda \in \mathbb{R}^{T^3}, \omega \in \mathbb{R}^N_+, \gamma \in \mathbb{R}^M_+ \\ Q_{\alpha,\beta,\delta,\lambda} \succeq 0}} \left\{ \min_{x \in \mathbb{R}^N} g_{\alpha,\beta,\delta,\lambda}(x) + \omega^T (x^2 - x) + \gamma^T (Ax - b) \right\}$$

Due to Constraints (2), it holds that $(Q2)$ is equivalent to $(Q3)$:

$$(Q3) : \max_{\substack{\alpha \in \mathbb{R}^N, \beta \in \mathbb{R}^{T^1}, \delta \in \mathbb{R}^{T^2}, \lambda \in \mathbb{R}^{T^3}, \gamma \in \mathbb{R}^M_+ \\ Q_{\alpha,\beta,\delta,\lambda} \succeq 0}} \left\{ \min_{x \in \mathbb{R}^N} g_{\alpha,\beta,\delta,\lambda}(x) + \gamma^T (Ax - b) \right\}$$

It is well known that a necessary condition for the quadratic function $g_{\alpha,\beta,\delta,\lambda,\gamma}(x) + \gamma^T(Ax - b)$ to have a minimum not equal to $-\infty$ is that matrix $Q_{\alpha,\beta,\delta,\lambda}$ is positive semidefinite. Therefore $(Q3)$ is equivalent to $(Q4)$:

$$(Q4): \max_{\alpha\in\mathbb{R}^N,\beta\in\mathbb{R}^{T^1},\delta\in\mathbb{R}^{T^2},\lambda\in\mathbb{R}^{T^3},\gamma\in\mathbb{R}^M_+} \left\{ \min_{x\in\mathbb{R}^N} g_{\alpha,\beta,\delta,\lambda,\gamma}(x) + \gamma^T(Ax-b) \right\}$$

We know from [32] that $(Q4)$ is equivalent to problem (D):

$$(D)\begin{cases} \max t \\ \text{s.t.} \\ \begin{pmatrix} -\gamma^T b - t & \frac{1}{2}(c^T_{\alpha,\beta,\delta,\lambda} + \gamma^T A) \\ \frac{1}{2}(c_{\alpha,\beta,\delta,\lambda} + A^T\gamma) & Q_{\alpha,\beta,\delta,\lambda} \end{pmatrix} \succeq 0 \\ t\in\mathbb{R},\ \alpha\in\mathbb{R}^N,\ \beta\in\mathbb{R}^{T^1},\ \delta\in\mathbb{R}^{T^2},\ \lambda\in\mathbb{R}^{T^3},\ \gamma\in\mathbb{R}^M_+ \end{cases}$$

By semidefinite duality of program (D), we get (SDP'):

$$(SDP')\begin{cases} \min < Q, X > + c^T x \\ \text{s.t.} \\ \quad (6)-(11) \\ \quad Ax \le b \end{cases}$$

We now prove that (SDP') and (SDP) are equivalent, i.e. that Constraints $Ax \le b$ are redundant in (SDP').

Lemma 2 *Due to Constraints (6)–(8), inequalities $Ax \le b$ are redundant in (SDP').*

Proof. Recall that $Ax \le b$ are the inequalities of $(C^i_{j,k})$, $\forall (i,j,k) \in J \times (I \cup J)^2 :\ \mathcal{E}_i = \mathcal{E}_j \cup \mathcal{E}_k$, i.e $x_i \ge 0$, $x_i \le x_j$, $x_i \le x_k$, and $x_i \ge x_j + x_k - 1$.

The basic idea here is that $\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0$ implies that all the symmetric minors are nonnegative.

- $x_i \ge 0$: We consider the determinant $\begin{vmatrix} 1 & x_i \\ x_i & X_{ii} \end{vmatrix}$, which implies $X_{ii} - x_i^2 \ge 0$. By (6) we obtain $x_i - x_i^2 \ge 0$ and thus $x_i \ge 0$.
- $x_i \le x_j$: Considering the determinant of the symmetric minor $\begin{vmatrix} X_{jj} & X_{ji} \\ X_{ij} & X_{ii} \end{vmatrix}$ implies $X_{ii}X_{jj} - X_{ij}^2 \ge 0$. By (6) we have $x_j x_i - X_{ij}^2 \ge 0$ and by (7) we obtain $x_i x_j - x_i^2 \ge 0$. We have either $x_i = 0$ or $x_i \le x_j$ and as $x_j \ge 0$, we have $x_i \le x_j$.
- $x_i \le x_k$: Same as the previous case by considering $\begin{vmatrix} X_{kk} & X_{ki} \\ X_{ik} & X_{ii} \end{vmatrix}$

– $x_i \geq x_j + x_k - 1$: $X - xx^T \succeq 0$ implies $\forall z \in \mathbb{R}^{N+1},\ z^T \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} z \geq 0$. By taking $\bar{z} = (1, 0, .., 0, \underbrace{-1}_{j}, 0, .., 0, \underbrace{-1}_{k}, 0, .., 0, \underbrace{1}_{i}, 0, .., 0)$, we have:

$$\bar{z}^T \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \bar{z} = (x_i + 1 - x_j - x_k) - (x_j - X_{jj} - X_{jk} + X_{ij}) \\ -(x_k - X_{kk} - X_{jk} + X_{ik}) + (x_i - X_{ij} - X_{ik} + X_{ii})$$

and using (6), (7) and (8), we have $\bar{z}^T \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \bar{z} = (x_i + 1 - x_j - x_k) \geq 0$, and we get the result.

□

We finally prove that there is no duality gap between (D) and (SDP'), which holds since:

(1) The feasible domain of (SDP') is nonempty (as $(QP_{\alpha,\beta,\delta,\lambda})$ contains 0 as a feasible solution) and (D) is bounded
(2) (D) satisfies Slater's condition. It is sufficient to take β, δ and λ equal to 0, α large enough so that $Q_{\alpha,\beta,\delta,\lambda} \succeq 0$ holds, and t a large negative number that ensures the diagonal dominance of the first row and the first column of matrix $\begin{pmatrix} -\gamma^T b - t & \frac{1}{2}(c^T_{\alpha,\beta,\delta,\lambda} + \gamma^T A) \\ \frac{1}{2}(c_{\alpha,\beta,\delta,\lambda} + A^T\gamma) & Q_{\alpha,\beta,\delta,\lambda} \end{pmatrix}$.

□

To sum up, we obtain (QP^*), the best equivalent convex formulation to (QP):

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

From Theorem 1, we deduce the Algorithm 1 to solve (P).

Algorithm 1 `PQCR` a global solution algorithm for (P)

Step 1: Apply a quadratization $\mathcal{E}$ to (P) and then generate sets $\mathcal{F}_{\mathcal{E}}$ and $\mathcal{S}_{\mathcal{E}}$.
Step 2: Solve (SDP), deduce optimal values α^*, β^*, δ^*, λ^*, and build (QP^*).
Step 3: Solve (QP^*) by a standard quadratic convex programming solver.

4 Numerical results

In this section, we evaluate PQCR on two applications: the *image restoration* problem, and the *low autocorrelation binary sequence* problem. For our experiments, we have arbitrarily chosen a quadratization for Step 1 of our method that is described in Algorithm 2. This choice impacts the number of constraints within $\mathcal{S}_{\mathcal{E}}$, and the associated continuous relaxation bound value can vary. We further illustrate this variation on a toy instance.

Experimental environment

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. For all algorithms, we use the multi-threading version of Cplex 12.7 with up to 48 threads.

For method PQCR, the quadratization is implemented in C, we used the solver Csdp [13] together with the Conic Bundle algorithm [23] for solving semi-definite programs (SDP), as described in [11]. We used the Ampl [19] interface of the solver Cplex 12.7 [24] for solving the quadratic convex problem (QP^*).

Parameters and algorithms within PQCR

– Phase 1: we choose the quadratization described in Algorithm 2.

Algorithm 2 Quadratization(f)

Require: A polynomial f of degree $d > 2$
Ensure: A quadratic function f' verifying $\forall x \in \{0,1\}^n,\ f'(x) = f(x)$

```
Sort all monomials by lexicographical order
for each monomial p from 1 to m do
  deg ← deg(p)
  while deg > 2 do
    deg ← ⌊deg/2⌋
    for i from 1 to deg do
      Replace the i^th disjointed product of two variables x_j x_k by a new variable x_i (if it does not already exist)
      E_i ← E_j ∪ E_k
    end for
  end while
end for
```

- Phase 2: Parameters `axtol, aytol` of `Csdp` [13] are set to 10^{-3}. The precision of the *Conic Bundle* [23] is set to 10^{-3}. Parameter p (see [11]) is set to $0.2 * T$.
- Phase 3: we let the default parameters, except the parameter `qptolin` that is set to `0`.

Methods used for comparison

- `Baron 17.4.1` [39] under `Gams`, that uses linear relaxations combined with domain reduction strategies within a branch-and-bound. We let the default parameters.
- `Cplex 12.7` [24] under `Ampl`, that uses a mix of linear and quadratic convex reformulations to solve (QP) by a branch-and-bound. We let the default parameters.
- `QCR` [9] we used the solver `Csdp` [13] for solving semi-definite programs and `Cplex 12.7` [24] for solving the quadratic convex problems.
- `MIQCR` [10] we used the solver `Csdp` [13] together with the Conic Bundle algorithm [23] for solving semidefinite programs and `Cplex 12.7` [24] for solving the quadratic convex problems.

4.1 The image restoration problem

The *vision* instances are inspired from the image restoration problem, which arises in computer vision. The goal is to reconstruct an original sharp base image from a blurred image. An image is a rectangle containing $l \times h$ pixels. This rectangle is modeled as a binary matrix of the same dimension. A complete description of these instances can be found in [17]. The size of the considered instances are $n = 100$, $n = 150$ and $n = 225$, with 15 instances of each size.

After Step 1 of our algorithm, several way are possible to solve the quadratic non-convex program (QP). For instance, the standard solver `Cplex` [24] can directly handle it, or one can apply the `QCR` [9] or `MIQCR` [10] methods. We compare `PQCR` with these three approaches, and with the direct submission of (P) to the solver `Baron 17.4.1`. Our observations for these instances are summed up in Table 1, where we compare, among all the instances of a given size, the maximum solution CPU time. The time-limit is set to 1 hour.

The time is given in seconds, `unsolved` means that none of the instance were solved within one hour, `(3/15)` means that only 3 instances out of 15 were solved within the time limit, and `out of memory` means that the method was not able to start the computation. Clearly, the quadratization followed by the direct submission to `Cplex` dominates the other considered methods. By comparing `PQCR` with `Baron`, we observe that the solver `Baron` is faster on the medium size instances, but it failed on most of the large instances, while `PQCR` is able to solve all the considered instances within 335 seconds. Concerning the two last methods, that consist in applying `QCR` and `MIQCR` after the quadratization phase, both methods failed. `QCR` because of the weakness of its bound, and `MIQCR` because of the size of the semidefinite problem considered for computing the best reformulation.

	vision instances		
Method	$n = 100$	$n = 150$	$n = 225$
`Baron 17.1.4`	$<$ 35s	$<$ 80s	$<$ 1238s (3/15)
`Quadratization + Cplex`	$<$ 18s	$<$ 51s	$<$ 125s
`Quadratization + QCR`	unsolved	unsolved	unsolved
`Quadratization + MIQCR`	out of memory	out of memory	out of memory
`PQCR`	$<$ 65s	$<$ 127 s	$<$ 335s

Table 1: *Comparison of the maximum time on 5 solution methods for the vision instances - time limit 3600 seconds*

4.2 The *Low Autocorrelation Binary Sequence* problem

We consider the problem of binary sequences with low off-peak autocorrelations. More formally, let S be a sequence $S = (s_1, \ldots, s_n)$ with $s \in \{-1, 1\}^n$, and for a given $k = 0, \ldots, n-1$, we define the autocorrelations $C_k(S)$ 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 $E(S)$:

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

This problem has numerous practical applications in communication engineering, or theoretical physics [8]. For our experiments, we consider truncated instances, i.e. sequences of length n where we compute low off-peak autocorrelation up to a certain distance $d \leq n$. These instances were introduced by [33] and can be found on the `polip` website [1].

For these instances, we compare `PQCR` with the solver `Baron 17.4.1` only, because none of the methods previously considered were able to solve any instances within the time limit of one hour.

Legends of Table 2

- *Name*: `Bernasconi.n.d` is the instance corresponding to computing optimal sequence of length `n` and with distance `d`.
- *Opt*: is the optimal solution value of the instance or the best known solution.
- *Gap*: $\left|\frac{Opt - Cont}{Opt}\right| * 100$, where *Cont* is the optimal value of the continuous relaxation of (QP^*), or the root relaxation value of `Baron`.
- *Sdp*: CPU time in seconds for solving (SDP). The time limit is set to 2400 seconds.
- *b&b* : CPU time in seconds for solving (QP^*) by a branch-and-bound.

- *Tot* (Total Time) : *Sdp* + *b&b* for `PQCR`, and total time branch-and-bound time for `Baron`. The time limit is set to 1 hour, and `-` means that the optimum is not found within the time limit.
- *Nodes*: Number of nodes visited by the branch-and-bound algorithm.

We present the results for the *Low Autocorrelation Binary Sequence* problem in Table 2. We observe that the solver `Baron` solves 9 instances out of 45 in 381 seconds on average, while `PQCR` solves 14 instances out of 45 in 1270 seconds on average. We set the time limit of Step 2 to 2400 seconds, because a feasible solution to (SDP) is sufficient to get parameters that convexify $g_{\alpha,\beta,\delta,\lambda}(x)$. We put in bold the best known solutions that `PQCR` improves in comparison to the solutions available on [1]. Indeed, `PQCR` is able to improve the best know solutions of 13 instances out of 45.

Instance		PQCR					Baron		
Name	*Opt*	*Gap*	*Sdp*	*b&b*	*Tot*	*Nodes*	*Gap*	*Tot*	*Nodes*
Bernasconi.20.10	-2936	7.5	837	9	846	31087	2918.0	59	7
Bernasconi.20.15	-5960	6.0	1228	14	1242	48252	3202.0	739	9
Bernasconi.20.3	-72	0.0	1	1	2	0	100.0	1	1
Bernasconi.20.5	-416	29.1	22	1	23	7621	1838.5	1	1
Bernasconi.25.13	-8148	4.6	1552	51	1603	99271	3109.4	-	68
Bernasconi.25.19	-14644	5.3	2400	233	2633	912383	3355.7	-	7
Bernasconi.25.25	-10664	6.7	2400	195	2595	432323	3404.7	-	5
Bernasconi.25.3	-92	0.0	1	1	2	6	100.0	5	1
Bernasconi.25.6	-960	21.4	461	8	469	132817	2306.7	17	27
Bernasconi.30.15	-15744	5.6	2400	325	2725	1834016	3220.5	-	6
Bernasconi.30.23	-30460	9.3	2400	1200	-	1278362	3449.7	-	1
Bernasconi.30.30	-22888	11.6	2400	1200	-	1469104	3469.7	-	1
Bernasconi.30.4	-324	63.6	58	20	78	511871	1346.9	2	7
Bernasconi.30.8	-2952	10.4	1940	100	2040	1562457	2696.2	2565	237
Bernasconi.35.18	-31168	12.2	2400	1200	-	2116286	3355.0	-	2
Bernasconi.35.26	-55232	45.6	2400	1200	-	92179	3511.2	-	1
Bernasconi.35.35	-40708	63.8	2400	1200	-	23907	3530.4	-	1
Bernasconi.35.4	-384	59.1	135	32	167	925225	1350.0	36	13
Bernasconi.35.9	-5108	11.1	2245	1108	3353	16425073	2826.2	-	38
Bernasconi.40.10	-8248	10.9	2400	1200	-	8578180	2953.0	-	16
Bernasconi.40.20	-50576	32.3	2400	1200	-	174108	3405.5	-	1
Bernasconi.40.30	-94776	156.7	2400	1200	-	18764	3564.7	-	1
Bernasconi.40.40	-67372	263.9	2400	1200	-	12330	3568.4	-	1
Bernasconi.40.5	-936	37.0	430	3170	-	24602867	1855.6	-	980
Bernasconi.45.11	-12748	12.3	2400	1200	-	4964503	3018.4	-	4
Bernasconi.45.23	**-85016**	147.7	2400	1200	-	25953	3487.3	-	1
Bernasconi.45.34	**-151144**	303.1	2400	1200	-	10438	3633.7	-	1
Bernasconi.45.45	-107984	487.1	2400	1200	-	7785	3720.8	-	1
Bernasconi.45.5	-1068	31.6	1384	2216	-	16951373	1853.6	-	600
Bernasconi.50.13	-23792	18.5	2400	1200	-	2559292	3130.9	-	2
Bernasconi.50.25	**-123692**	250.3	2400	1200	-	15649	3541.5	-	1
Bernasconi.50.38	**-231128**	650.9	2400	1200	-	20453	3667.9	-	1
Bernasconi.50.50	**-162400**	1518.7	2400	1200	-	9300	3671.2	-	1
Bernasconi.50.6	-2160	25.2	1130	2470	-	11641486	2321.5	-	76
Bernasconi.55.14	-33272	34.2	2400	1200	-	739121	3186.1	-	3
Bernasconi.55.28	**-189004**	383.1	2400	1200	-	16444	3581.5	-	1
Bernasconi.55.41	**-332940**	997.7	2400	1200	-	6122	3703.9	-	1
Bernasconi.55.55	**-232828**	2003.2	2400	1200	-	3857	3718.8	-	1
Bernasconi.55.6	-2400	24.6	2400	1200	-	10174973	2322.7	-	19
Bernasconi.60.15	**-44476**	82.7	2400	1200	-	45286	3292.7	-	1
Bernasconi.60.30	**-258152**	552.3	2400	1200	-	78118	3618.8	-	1
Bernasconi.60.45	**-473116**	1282.4	2400	1200	-	5196	713.5	-	1
Bernasconi.60.60	**-334240**	2096.0	2400	1200	-	3795	no bound	-	1
Bernasconi.60.8	**-6788**	18.2	2400	1200	-	6478048	2714.1	-	1

Table 2: *Results of PQCR and Baron for the 45 polip instances. Time limit 1 hour.*

4.3 A short discussion on the impact of the chosen quadratization

In this section, we shortly explore the impact of the chosen quadratization on the tightness of the associated continuous relaxation bound. In Table 3, we sum up the continuous relaxation bound values obtained by convexification `PQCR`, `QCR`, and `MIQCR` for the three quadratizations of (Ex) presented in Example 1.

Method		(QEx_1)	(QEx_2)	(QEx_3)
	Opt	*Cont*	*Cont*	*Cont*
`Quadratization + QCR`	0	-3	-3	-3
`Quadratization + MIQCR`	0	0	0	0
`PQCR`	0	-0.125	-0.625	-0.375

Table 3: *Comparison of continuous relaxation bound for (Ex) with convexifications `PQCR`, `QCR`, and `MIQCR`*

We observe that for this example, `MIQCR` and `QCR` are robust to the quadratization step. This is due to the fact that `MIQCR` considers all the possible perturbations of the Hessian matrix. This is the same for `QCR`, but only for the diagonal terms. On the contrary, method `PQCR` gets different bounds for each quadratization. An interesting question for a future research would thus be to determine which quadratizations lead to sharper bounds for method `PQCR`.

5 Conclusion

We consider the general problem (P) of minimizing a polynomial function where the variables are binary. In this paper, we present `PQCR` a solution approach for (P). `PQCR` can be split in 3 phases. We called the first phase *quadratization*, where we rewrite (P) as an equivalent quadratic program (QP). For this we have to add new variables and linear constraints. We get a linearly constrained quadratic program that still have a non-convex objective function and binary variables. Moreover, even for small instances of (P), the existing convexification methods failed for solving the associate (QP). This is why, we present a family of tailored quadratic convex reformulations of (QP) that exploits its specific structure. For this, we introduce new valid quadratic equalities that vanish on the feasible domain of (QP). Then, we focus on finding, within this family, the equivalent convex formulation that maximizes the continuous relaxation bound value. Then, we show that we can compute this "best" convex reformulation using a semidefinite relaxation of (QP). Finally, we solve our optimal reformulation with a standard solver.

We present computational results on two applications and compare our algorithm with other convexification methods and the general solver `Baron`. In particular, we show that for the *low autocorrelation binary sequence problem*,

PQCR is able to improve the best known solution of 13 instances out of 45. A future research would be to characterize which quadratization best fit with our convexification phase from the continuous relaxation value point of view.

References

1. Polip, library for polynomially constrained mixed-integer programming, 2014.
2. T. Achterberg. Scip : solving constraint integer programs. *Mathematical Programming Computation*, (1):1–41, 2009.
3. C.S. Adjiman, S. Dallwig, C.A. Floudas, and A. Neumaier. A global optimization method, αbb, for general twice-differentiable constrained nlps—i. theoretical advances. *Computers and Chemical Engineering*, 22(9):1137–1158, 1998.
4. 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.
5. M. Anthony, E. Boros, Y. Crama, and A. Gruber. Quadratic reformulations of nonlinear binary optimization problems. *Mathematical Programming*, 162:115–144, 2017.
6. B. Balasundaram and A.0. Prokopyev. On characterization of maximal independent sets via quadratic optimization. 19, 06 2011.
7. P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Br anching and bounds tightening techniques for non-convex minlp. *Optimization Methods and Software*, 4–5(24):597–634, 2009.
8. J. Bernasconi. Low autocorrelation binary sequences: statistical mechanics and configuration space analysis. *J. Physique*, 141(48):559–567, 1987.
9. 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.
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. Reformulation Techniques and Mathematical Programming.
13. B. Borchers. CSDP, A C Library for Semidefinite Programming. *Optimization Methods and Software*, 11(1):613–623, 1999.
14. E. Boros, P.L. Hammer, and X. Sun. Network flows and minimization of quadratic pseudo-boolean functions. Technical Report TR: 1991-17, RUTCOR, 1991.
15. C. Buchheim and C. D'Ambrosio. Monomial-wise optimal separable underestimators for mixed-integer polynomial optimization. *Journal of Global Optimization*, pages 1–28, 2016.
16. M.W. Carter. The indefinite zero-one quadratic problem. *Discrete Applied Mathematics*, pages 23–44, 1984.
17. Y. Crama and E. Rodriguez-Heck. A class of valid inequalities for multilinear 0-1 optimization problems. *Discrete Optimization*, pages 28–47, 2017.

18. R. Fortet. L'algèbre de Boole et ses Applications en Recherche Opérationnelle. *Cahiers du Centre d'Etudes de Recherche Opérationnelle*, 4:5–36, 1959.
19. R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming.* The Scientific Press (now an imprint of Boyd & Fraser Publishing Co.), Danvers, MA, USA, 1993.
20. M.R. Garey and D.S. Johnson. Computers and Intractability: A guide to the theory of NP-Completness. *W.H. Freeman, San Francisco, CA*, 1979.
21. Bissan Ghaddar, Juan C. Vera, and Miguel F. Anjos. A dynamic inequality generation scheme for polynomial programming. *Mathematical Programming*, 156(1):21–57, Mar 2016.
22. P.L. Hammer and A.A. Rubin. Some remarks on quadratic programming with 0-1 variables. *Revue Française d'Informatique et de Recherche Opérationnelle*, 4:67–79, 1970.
23. C. Helmberg. *Conic Bundle v0.3.10*, 2011.
24. IBM-ILOG. IBM ILOG CPLEX 12.7 Reference Manual. "`http://www-01.ibm.com/support/knowledgecenter/SSSA5P_12.7.0/ilog.odms.studio.help/Optimization_Studio/topics/COS_home.html`", 2017.
25. 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.
26. R.M. Karp. Reducibility among combinatorial problems. pages 85–103, 1972.
27. J. Krarup and P.M. Pruzan. Computer-aided layout design. pages 75–94, 1978.
28. X. Kuang, , B. Ghaddar, J . Naoum-Sawaya, and L.F. Zuluaga. Alternative SDP and SOCP Approximations for Polynomial Optimization. *ArXiv e-prints*, October 2015.
29. J.B. Lasserre. *An Introduction to Polynomial and Semi-Algebraic Optimization.* Cambridge University Press, Cambridge, 2015.
30. J.B. Lasserre and T.P. Thanh. Convex underestimators of polynomials. *Journal of Global Optimization*, pages 1–25, 2013.
31. J.D. Laughhunn. Quadratic binary programming with applications to capital budgeting problems. 18:454–461, 06 1970.
32. C. Lemarechal and F. Oustry. Semidefinite relaxations and lagrangian duality with application to combinatorial optimization. Technical report, RR-3710, INRIA Rhones-Alpes, 1999.
33. 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.
34. R.D. McBride and J.S. Yormark. An implicit enumeration algorithm for quadratic integer programming. *Management Science*, 1980.
35. 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.
36. P.M Pardalos and J. Xue. The maximum clique problem. *Journal of Global Optimization*, 4(3):301–328, Apr 1994.
37. M.R. Rao. Cluster analysis and mathematical programming. *Journal of the American Statistical Association*, 66(335):622–626, 1971.
38. J.M.W. Rhys. A selection problem of shared fixed costs and network flows. *Management Science*, 17(3):200–207, 1970.
39. N.V. Sahinidis and M. Tawarmalani. Baron 9.0.4: Global optimization of mixed-integer nonlinear programs. *User's Manual*, 2010.

40. 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.