Unshackling Column Generation for Linearized Unconstrained Binary Quadratic Programs

Sven Mallach[1,2]

[1]Network & Data Science Management Group, University of Siegen, Germany
[2]University of Bonn, Germany

September 13, 2026

Abstract

When linearizing binary quadratic programs, the most usual way is to replace bilinear products with additional variables constrained to take on consistent values in any feasible solution. In this setting, column generation is a principally desirable solution technique, for instance because the number of such additional linearization variables may be large while many of them may turn out to be zero in an optimal solution. Even more, the consistency constraints of the most customary linearization technique add significantly to the degeneracy of the basic solutions traversed when solving the continuous relaxation with the simplex algorithm. At the same time, solving this relaxation by column generation poses difficulties. The foremost one is that essentially all linearization variables with a negative objective coefficient need to be incorporated in order to deduce a valid lower bound for minimization from such a procedure in general, with only rare exceptions where an actual decreasing effect on the objective might be ruled out by other means than reoptimization. This translates accordingly to positive coefficients for upper bounds when maximizing the objective. To resolve this, we present a column generation approach built upon a strategic combination of classical techniques such as boolean complementation and continuous Rhys forms in order to maintain dual feasibility and thus to provide a valid dual bound at any time.

1 Introduction

Quadratic Unconstrained Binary Optimization (QUBO) consists in finding a solution $x \in \{0,1\}^n$ that minimizes the quadratic function $f(x) \coloneqq x^\top Q x + c^\top x$ for a given symmetric matrix $Q \in \mathbb{R}^{n \times n}$ and a given vector $c \in \mathbb{R}^n$. It is $\mathcal{NP}$-hard in general [8].

A common approach to this problem that is also the basis of some state-of-the-art solvers like [4, 15] is via linearization: Exploiting that $x_i^2 = x_i$

for $x_i \in \{0,1\}$, define first $c'_i \coloneqq c_i + q_{ii}$ for all $i \in \{1,\ldots,n\}$. Then, define $E \coloneqq \{(i,j) : i,j \in \{1,\ldots,n\}, i < j, q_{ij} \neq 0\}$ and introduce a variable w_{ij}, supposed to represent the product $x_i \cdot x_j$, with associated objective coefficient $d_{ij} = q_{ij} + q_{ji}$ for all $(i,j) \in E$. Now, QUBO may be reformulated as the following binary linear program which is often referred to as the "standard" linearization:

$$
\begin{aligned}
\text{(StdLin)} \quad \min \; & \sum_{(i,j)\in E} d_{ij} w_{ij} + \sum_{i=1}^{n} c'_i x_i \\
\text{s.t. } & w_{ij} - x_i \leq 0 && \text{for all } (i,j) \in E && (1)\\
& w_{ij} - x_j \leq 0 && \text{for all } (i,j) \in E && (2)\\
& x_i + x_j - w_{ij} \leq 1 && \text{for all } (i,j) \in E && (3)\\
& w_{ij} \geq 0 && \text{for all } (i,j) \in E && (4)\\
& x_i \in \{0,1\} && \text{for all } i \in \{1,\ldots,n\}
\end{aligned}
$$

This linear reformulation goes back to Fortet [7] and Glover and Woolsey [9], and the inequalities (1)–(4) also appear as the convex envelopes described by McCormick [14] for the special cases of products of binary variables.

Despite the restriction of the set E to products with non-zero off-diagonal entries in Q, in general, the number of linearization variables and associated constraints may be costly to handle as an explicit part of the program. In addition, the linearization constraints introduce a considerable level of degeneracy into the linear programming (LP) relaxation of (StdLin). At the same time, QUBO instances frequently admit an optimum solution, likely to both (StdLin) and its relaxation, in which may w-variables take the value zero.

The aforementioned aspects strongly motivate a column generation (CG) approach to solve the LP relaxation of (StdLin) in a natural way: Work with a restricted collection of linearization variables and constraints associated with a subset $E' \subseteq E$, assuming $w_{ij} = 0$ for all $(i,j) \in E \setminus E'$, and alter E' dynamically until optimality is proven. The program comprising only the selected variables and constraints is then referred to as a *restricted master program* (RMP), and the task of determining a so far neglected variable that could still improve the solution value (if any) is called *pricing* [6].

However, while CG approaches for binary quadratic programs have been devised in other contexts as referenced in Section 2, particularly in terms of applying a Dantzig-Wolfe reformulation [6], we are not aware of any literature addressing a conventional CG approach for solving the LP relaxation of (StdLin) or some other QUBO linearization as an RMP.

One potential reason for this may be that one encounters several severe obstructions when trying to solve the LP relaxation of (StdLin) directly in this way. In particular, as long as there exists a pair $(i,j) \in E \setminus E'$ with $q_{ij} < 0$, the RMP with respect to E' does not guarantee a valid lower bound on the optimum value of f for $x \in \{0,1\}^n$. To the contrary, the canonical extension of the current RMP solution that keeps the newly introduced primal and dual variables at zero preserves primal feasibility and destroys dual feasibility with d_{ij} being the according reduced cost of w_{ij}, as is exposed in detail in the appendix, Section A. In other words, *every* variable w_{ij}, with $(i, j) \in E \backslash E'$ and $q_{ij} < 0$, is indicated as an entering candidate with a potential improvement of the objective. Moreover, only in very specific cases and when investing additional effort, it is possible to certify or rule out an *actual* strict decrease in the optimal objective when incorporating such w_{ij} *without reoptimization*. So in general, one may expect that deriving a valid lower bound will require to include a large subset of the pairs $(i, j) \in E \setminus E'$ with $q_{ij} < 0$. Consequently, there is little algorithmic control over the effectiveness of a customary CG approach for a "standard" linearized QUBO, rather it depends on the given sign pattern and coefficient distribution of the matrix Q and the vector c.

In this paper, we propose a column generation approach for an equivalent QUBO linearization that resolves the aforementioned issues, that provides a valid lower bound at any time, and that can be solved in a suitable way so as to iteratively reach the lower bound provided by the complete LP relaxation of (StdLin). Rather uncommonly for a minimization problem, our CG approach is thus designed so as to *increase* the *optimum* objective value of the RMPs solved successively, providing the additional possibility of early termination. The enabling key ingredients for our technique are the concepts of boolean complementation and continuous Rhys forms [16, 10], which we address in the upcoming section. Combining these strategically, we devise a well-defined and computationally attractive pricing mechanism that preserves dual feasibility as well as a monotone behavior of the objective function respectively lower bound. We discuss the key concepts and relevant technicalities to implement our method and provide a demonstrating computational example in Section 3. The paper closes with a brief conclusion in Section 4 and the subsequent appendix.

2 Preparations, Preliminaries, and Related Work

In this section, we briefly introduce the major prevalent techniques that our column generation approach is based on, and reference some different CG methods that have been designed for binary quadratic programs.

In all conscience, the most closely related literature on column generation for (unconstrained as well as constrained) binary quadratic optimization has so far mainly focused on Dantzig–Wolfe reformulations [3, 2, 13] and approaches based on alternative reformulations with an exponential number of variables [1]. In [13], Mauri and Lorena first partition the set of variables to construct (linearized) QUBO subproblems to which a Dantzig–Wolfe reformulation is subsequently applied, while the linking constraints are handled by means of Lagrangian relaxation. Moreover, Bettiol et al. [2] develop relaxations based on Dantzig–Wolfe reformulations in matrix space and extend them to block-decomposable binary quadratic programs. Ceselli, Létocart and Traversi [3], in turn, develop a systematic framework combining Dantzig–Wolfe reformulations with quadratic convex reformulations for constrained binary quadratic programs. Finally, Bayani et al. [1] develop a cost-splitting reformulation with exponentially many variables and exploit its structure through a decomposition-based column generation scheme.

In this sense, a fundamental difference of the present work is that it devises a column generation approach for the linearized *original* problem respectively its continuous relaxation, thereby addressing and resolving the issues that occur if such an approach was followed (too) straightforwardly for the "standard" linearization.

To prepare for this, we first partition the set E from Section 1 into the subsets

$$P \coloneqq \{(i,j) : i,j \subseteq \{1,\ldots,n\}, i<j, q_{ij}>0\}$$

respectively

$$N \coloneqq \{(i,j) : i,j \subseteq \{1,\ldots,n\}, i<j, q_{ij}<0\}$$

of index pairs with positive and negative coefficients in Q. Moreover, for any vector $x \in \mathbb{R}^n$, $0 \leq x \leq 1$, we define its *complement* vector $\bar{x} \coloneqq 1-x$ of the same dimension. The next ingredient is the observation, attributed to Rhys [16] by Hammer, Hansen and Simeone in [10], that, using complemented and uncomplemented variables, one may reformulate f into a quadratic function p in the variables x and $\bar{x}$ such that all the coefficients associated with *quadratic* terms are non-negative. In fact, we may simply obtain $p(x,\bar{x})$ from $f(x)$ by replacing any term $x_i x_j$ for $(i,j) \in N$ with $(1-\bar{x}_i)x_j = x_j - \bar{x}_i x_j$, i.e.:

$$p(x,\bar{x}) \coloneqq \sum_{(i,j)\in P} p_{ij} x_i x_j + \sum_{(i,j)\in N} p_{ij} \bar{x}_i x_j + \sum_{i=1}^{n} p_{ii} x_i$$

where $p_{ii} = \left(c_i + q_{ii} + \sum_{(j,i)\in N} 2q_{ji}\right)$, and $p_{ij} = \begin{cases} 2q_{ij}, & \text{if } (i,j) \in P \\ -2q_{ij}, & \text{if } (i,j) \in N \end{cases}$.

This reformulation now gives rise to an alternative linearization, called discrete Rhys form in [10], which removes the complements again. Its set of feasible solutions and also the lower bound provided by its LP relaxation coincide with (StdLin) [10], but the crucial difference is that the linearization variables, which are now denoted by y_{ij}, represent either products of the form $x_i x_j$ for $(i,j) \in P$ or $\bar{x}_i x_j$ for $(i,j) \in N$, while the respective consistency is enforced via linear constraints in the uncomplemented variables. In [10], the discrete Rhys form is stated for a maximization problem, while we state it here for minimization. This is reflected by replacing the upper bounding pendant constraints of (1) and (2) in [10] with the lower bounding pendant constraints (5) respectively (7) of (3) and (4) as shown subsequently.

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in P} p_{ij} y_{ij} + \sum_{(i,j)\in N} p_{ij} y_{ij} + \sum_{i=1}^{n} p_{ii} x_i \\
\text{s.t. } & x_i + x_j - y_{ij} \leq 1 && \text{for all } (i,j) \in P && (5)\\
& y_{ij} \geq 0 && \text{for all } (i,j) \in P && (6)\\
& x_j - x_i - y_{ij} \leq 0 && \text{for all } (i,j) \in N && (7)\\
& y_{ij} \geq 0 && \text{for all } (i,j) \in N && (8)\\
& x_i \in \{0,1\} && \text{for all } i \in \{1,\dots,n\}
\end{aligned}$$

Indeed, (5) coincides with (3) while (7) is essentially (3) as well but with x_i replaced with $1 - x_i$, and (6) and (8) just replicate (4). In the context of this relation, it is well known that – equivalently – in (StdLin) and its LP relaxation, the inequalities (1)–(2) can be restricted to the cases $(i,j) \in N$ while (3)–(4) can be restricted to the cases $(i,j) \in P$, in the sense that these inequalities are necessarily satisfied at an optimum solution [5].

3 A Column Generation Approach for linearized QUBO

We now present our column generation approach for linearized QUBO. We start with introducing our version of the primal and dual continuous Rhys forms related to the discrete one as defined in Section 2. Then, we derive the key results regarding the validity of the provided lower bounds as well as the pricing of so far neglected linearization variables. Finally, we demonstrate the addition of variables with an example.

3.1 Building Continuous Rhys Forms

From the discrete Rhys form presented in Section 2, we first build the following according LP relaxation, subsequently referred to as (primal) continuous Rhys form in analogy to [10]. Thereby, we now treat the lower bound constraints on the linearization variables as simple variable bounds.

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in P} p_{ij} y_{ij} + \sum_{(i,j)\in N} p_{ij} y_{ij} + \sum_{i=1}^{n} p_{ii} x_i && && (9)\\
\text{s.t. } & y_{ij} - x_i - x_j \geq -1 && \text{for all } (i,j) \in P \ [u_{ij}] && (10)\\
& y_{ij} + x_i - x_j \geq 0 && \text{for all } (i,j) \in N \ [u_{ij}] && (11)\\
& -x_i \geq -1 && \text{for all } i \in \{1,\dots,n\} \ [z_i] && (12)\\
& x_i \geq 0 && \text{for all } i \in \{1,\dots,n\} && (13)\\
& y_{ij} \geq 0 && \text{for all } (i,j) \in P \cup N && (14)
\end{aligned}$$

As in the discrete version, the variables y_{ij} here represent the product $x_i \cdot x_j$ if $(i,j) \in P$ and the product $\bar{x}_i \cdot x_j$ if $(i,j) \in N$. Observe that one may read off the valid lower bound

$$f_{\mathrm{NL}} \coloneqq \sum_{p_{ii}<0} p_{ii} = \min \left\{ \sum_{i=1}^{n} p_{ii} x_i : \text{ (12), (13)} \right\}$$

on the optimum value of f for $x \in \{0,1\}^n$ from this linear program. Indeed, we might start with an RMP consisting only of the variables x_i, and an initial set of constraints (12)–(13), such that f_{NL} is the value of an optimum solution to the so restricted continuous Rhys form.

The inequalities of the continuous Rhys form are attached with a naming of according dual variables, as we will also work with the dual continuous Rhys form:

$$\max \; - \sum_{(i,j)\in P} u_{ij} - \sum_{i=1}^{n} z_i \tag{15}$$

$$\text{s.t.} \sum_{(i,j)\in N} u_{ij} - \sum_{(i,j)\in P} u_{ij} - \sum_{(j,i)\in P\cup N} u_{ji} - z_i \le p_{ii} \quad \text{for all } i \in \{1,\dots,n\} \; [x_i] \tag{16}$$

$$u_{ij} \le p_{ij} \quad \text{for all } (i,j) \in P \cup N \; [y_{ij}] \tag{17}$$

$$u_{ij} \ge 0 \quad \text{for all } (i,j) \in P \cup N \tag{18}$$

$$z_i \ge 0 \quad \text{for all } i \in \{1,\dots,n\} \tag{19}$$

By linear programming duality, we have

$$f_{\mathrm{NL}} = \max \left\{ -\sum_{i} z_i : -z_i \le p_{ii} \text{ for all } i \in \{1,\dots,n\}, \text{(19)} \right\}$$

and

$$f_{\mathrm{LP}} \coloneqq \max \{(15) : (16)\text{–}(19)\} = \min \{(9) : (10)\text{–}(14)\}.$$

3.2 Maintaining Dual Feasibility, Pricing, Reoptimization

Starting from f_{NL}, our goal is to improve the bound to f_{LP} by successively adding a typically restricted subset of the initially missing variables and constraints. Thereby, any restricted subset leads to a minimum primal respectively maximum dual objective value that is a valid lower bound. We prove this with an intermediate lemma as follows.

Lemma 1. *Consider an optimum (basic feasible) solution to the restricted dual continuous Rhys form associated with the variable index sets $P' \subseteq P$ and $N' \subseteq N$ in addition to the z-variables. Then for every $(i,j) \in (P \cup N) \setminus (P' \cup N')$, there is a (basic) feasible solution with the same objective value for the dual program that results from adding the variable and constraint associated with (i,j).*

Proof. Setting $u_{ij} = 0$ (non-basic) gives a feasible extended solution, satisfying the new instances of (17) and (18), as well as the accordingly appended instance of (16). Choose the new slack variable of (17) to be basic with value p_{ij}. The objective value remains unchanged. □

Proposition 1. *Consider an optimum (basic feasible) solution to the restricted primal or dual continuous Rhys form associated with the variable index sets $P' \subseteq P$ and $N' \subseteq N$ in addition to the x- respectively z-variables. Then its objective value is a valid lower bound on the optimum value of f for $x \in \{0,1\}^n$.*

Proof. For the restricted dual continuous Rhys form, by Lemma 1, there is a feasible solution with the same objective value for *any* so far missing variable $(i,j) \in (P \cup N) \setminus (P' \cup N')$. Thus an optimum (maximum value) feasible solution of this program can only have an equal or larger objective value. By LP duality, the value of an optimum solution of the so appended restricted primal continuous Rhys form can only be equal or larger as well. Thus, the objective value prior to the extension is a valid lower bound. □

Besides providing a means of dual feasible extension, Lemma 1 also permits to append dual variables in a a computationally effective way, in particular supporting a continuation from the respective current LP solution ("warmstarting"). We now employ this to derive a reasonable pricing procedure for the primal continuous Rhys form and analyze which variables are entering candidates of interest.

Proposition 2. *Let $P' \subseteq P$ and $N' \subseteq N$ be the index sets describing the variables and constraints present in the restricted primal continuous Rhys form in addition to the x-variables. Then including the variable and constraint associated with a variable y_{ij} for $(i,j) \in (P \cup N) \setminus (P' \cup N')$ can increase the optimum objective value only if $(i,j) \in P$ and $x_i + x_j > 1$ or if $(i,j) \in N$ and $x_j - x_i > 0$.*

Proof. Consider to append the according dual variable u_{ij} and its associated constraints to the restricted dual continuous Rhys form, in the way described in the proof of Lemma 1, to first obtain an extended dual feasible solution of the same objective value. By definition (choosing signs so that feasible slacks are non-negative), the reduced cost of u_{ij} is then equal to the negated slack of its respective primal constraints, i.e., it is equal to $x_i + x_j - y_{ij} - 1$ if $(i,j) \in P$ and equal to $x_j - x_i - y_{ij}$ if $(i,j) \in N$. In other words, it is positive if and only if its corresponding primal constraint (10) respectively (11) is violated by the extended primal solution that emerges from "keeping" $y_{ij} = 0$ as before. Thus, u_{ij} is a candidate to enter the current basis if and only if $x_i + x_j > 1$ respectively $x_j - x_i > 0$. Otherwise, the constructed dual solution remains feasible and optimal with $u_{ij} = 0$, and so does then the primal one with $y_{ij} = 0$, leaving the previous objective value unchanged. □

Together with Lemma 1, the Propositions 1 and 2 express formally that our corresponding CG approach resolves the issues that one may encounter with a conventional one for (StdLin) as exposed in the appendix, Section A: Given the straightforward preservation of dual feasibility, the objective value of each restricted program is always a feasible lower bound that may only improve during the further course. Moreover, the proof of Proposition 2 provides an accurate "reverse" pricing mechanism that answers which variable to append in order to (potentially) *increase* the current optimum. Since this only involves to evaluate the reduced cost quantities $x_i + x_j - 1$ for $(i,j) \in P \setminus P'$ and $x_j - x_i$ for $(i,j) \in N \setminus N'$ with respect to the current primal solution (without actually truly extending the dual one beforehand), the pricing procedure is also comparably cheap from a computational point of view. Choosing for instance a variable that has the maximum (strictly positive) dual reduced cost is an analogue of Dantzig's pricing rule. Likewise, choosing the first one with strictly positive dual reduced costs is an analogue to Bland's pricing. After selecting a variable and according constraint to be appended to the primal program, the variable remains non-basic at value zero while the negative slack variable of the added constraint enters the basis. Starting from the so extended basis, the program can be reoptimized ("warmstarted") with the dual simplex algorithm. Alternatively, one may append the dual program in the feasible way described in the proof of Lemma 1 and resolve it then with the primal simplex algorithm. Moreover, incorporated linearization variables may of course also become redundant and be removed again from the program during the further course.

3.3 Demonstrating Example

We consider a QUBO instance with

$$Q = \begin{pmatrix} 0 & 3 & -1 & -5 \\ 3 & 0 & 1 & -4 \\ -1 & 1 & 0 & 0 \\ -5 & -4 & 0 & 0 \end{pmatrix} \text{ and } c^\mathsf{T} = \begin{pmatrix} 0 & 3 & 2 & 6 \end{pmatrix}.$$

Therefore, $f(x) = 3x_2 + 2x_3 + 6x_4 + 6x_1x_2 - 2x_1x_3 - 10x_1x_4 + 2x_2x_3 - 8x_2x_4$, with $P = \{(1,2),(2,3)\}$ and $N = \{(1,3),(1,4),(2,4)\}$. Building the according function in complemented and uncomplemented variables from Section 2 such that all the coefficients of quadratic terms are non-negative, we obtain

$$p(x,\bar{x}) = 3x_2 - 12x_4 + 6x_1x_2 + 2\bar{x}_1x_3 + 10\bar{x}_1x_4 + 2x_2x_3 + 8\bar{x}_2x_4.$$

The primal continuous Rhys form is then:

$$\begin{aligned}
&\min\ 3x_2 - 12x_4 + 6y_{12} + 2y_{13} + 10y_{14} + 2y_{23} + 8y_{24} \\
\text{s.t. } & y_{12} - x_1 - x_2 \geq -1 && [u_{12}] \\
& y_{13} + x_1 - x_3 \geq 0 && [u_{13}] \\
& y_{14} + x_1 - x_4 \geq 0 && [u_{14}] \\
& y_{23} - x_2 - x_3 \geq -1 && [u_{23}] \\
& y_{24} + x_2 - x_4 \geq 0 && [u_{24}] \\
& -x_1 \geq -1 && [z_1] \\
& -x_2 \geq -1 && [z_2] \\
& -x_3 \geq -1 && [z_3] \\
& -x_4 \geq -1 && [z_4] \\
& x_1, x_2, x_3, x_4 \geq 0 \\
& y_{12}, y_{13}, y_{14}, y_{23}, y_{24} \geq 0
\end{aligned}$$

An optimum solution is $x_1 = x_2 = x_4 = \frac{1}{2}$ and all other variables equal to zero. The objective function value f_{LP} is $-4\frac{1}{2}$ while the optimum binary value is -4 attained with $x_1 = x_4 = 1$ and all other variables equal to zero.

The dual continuous Rhys form is as follows:

$$
\begin{aligned}
& \max\ -u_{12} - u_{23} - z_1 - z_2 - z_3 - z_4 \\
\text{s.t. } & -u_{12} + u_{13} + u_{14} - z_1 \le 0 && [x_1] && (20) \\
& -u_{12} - u_{23} + u_{24} - z_2 \le 3 && [x_2] && (21) \\
& -u_{13} - u_{23} - z_3 \le 0 && [x_3] && (22) \\
& -u_{14} - u_{24} - z_4 \le -12 && [x_4] && (23) \\
& u_{12} \le 6 && [y_{12} = x_1 \cdot x_2] && (24) \\
& u_{13} \le 2 && [y_{13} = \bar{x}_1 \cdot x_3] && (25) \\
& u_{14} \le 10 && [y_{14} = \bar{x}_1 \cdot x_4] && (26) \\
& u_{23} \le 2 && [y_{23} = x_2 \cdot x_3] && (27) \\
& u_{24} \le 8 && [y_{24} = \bar{x}_2 \cdot x_4] && (28) \\
& u_1, u_2, u_3, u_4 \ge 0 \\
& u_{12}, u_{13}, u_{14}, u_{23}, u_{24} \ge 0
\end{aligned}
$$

Suppose that we start with the restricted primal program for $P' \cup N' = \emptyset$

$$\min \{3x_2 - 12x_4 : 0 \le x_1, x_2, x_3, x_4 \le 1\}$$

to compute f_{NL}. We then clearly obtain $f_{\mathrm{NL}} = -12$ by setting x_4 to one and all other x-variables to zero. This is reflected in the first row labeled "sol" (for primal solution) in Table 1. The next step is to check whether we can improve this lower bound by means of the "reverse" pricing described in Section 3.2. To this end, we first observe that, in accordance with (the proof of) Lemma 1, it is feasible to extend the current dual solution by setting $u_{ij} = 0$ for any $(i, j) \in P \cup N$, because the right-hand side of any constraint of the form (17), i.e., here the constraints (24)–(28), is positive by construction, and the constraints of the form (16), here (20)–(23), remain unaffected from this choice. Thus, the objective value is unaffected as well. Given a so extended dual solution, the dual reduced cost (the gain in the dual objective per unit of increase) of the new variable u_{ij} is, under the current assumption that $y_{ij} = 0$, equal to $x_i + x_j - 1$ if $(i, j) \in P \setminus P'$ and equal to $x_j - x_i$ if $(i, j) \in N \setminus N'$, see Proposition 2. Clearly, if the maximum of these values over all $(i, j) \in (P \cup N) \setminus (P' \cup N')$ is non-positive, we are certified that no increase of the (dual, and thus also the primal) objective is possible. Otherwise, we have at least one entering candidate u_{ij} which directly translates to the according entering candidate y_{ij} for the primal problem. For the currently considered solution, the dual reduced costs are displayed in the first row labeled "drc" in Table 1. Moreover, the feasible amount of increase of u_{ij} is clearly bounded by (at most) p_{ij} due to (17). The course of the follow-up iterations, until reaching optimality (primal and dual feasibility), when choosing y_{14} to enter first, is shown in the further rows of Table 1. As is also visible, the optimum is reached without having introduced all linearization variables.

	obj	x_1	x_2	x_3	x_4	y_{12}	y_{13}	y_{14}	y_{23}	y_{24}
sol	-12	0	0	0	1					
drc						-1	0	1	-1	1
sol	-12	1	0	0	1			0		
drc						0	-1		-1	1
sol	-9	1	1	0	1			0		0
drc						1	-1		0	
sol	$-\frac{9}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0		0		0
drc							$-\frac{1}{2}$		$-\frac{1}{2}$	

Table 1: Course of our CG procedure for the example instance.

4 Conclusion

In this paper, we have presented a resolution of the obstructions encountered when devising a conventional column generation approach for the most commonly used ("standard") linearization of binary quadratic programs. Our approach is based on a reformulation of the unconstrained binary quadratic optimization problem using boolean complementation and continuous Rhys forms. Strategically combining these classical concepts, we devise a principle to treat linearization variables as optional columns in a suitable way, along with a pricing mechanism that ensures the efficient maintenance of a valid dual bound throughout the entire optimization procedure. As opposed to that, a conventional approach generally requires to integrate a significant portion of the linearization variables with negative (positive) coefficients for minimization (maximization).

Several research directions emanate from this fundamental starting point. For instance, an analysis of the efficiency of different pricing rules that can be designed based on information beyond the reduced costs, and the determination of strategies to identify a well-defined and effective initial set of columns. Another possible and interesting subject of investigation is the presence and handling of further linear or linearized constraints. More generally, the here presented principal idea of handling linearization variables based on the sign of their objective coefficients might be transferable to other linearization techniques, see e.g. [11, 12] for further according references and an example method that exploits the presence of additional constraints and applies boolean complementation as well (if necessary).

Acknowledgements

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 537033028.

References

[1] M. Bayani, B. Rostami, Y. Adulyasak, and L.-M. Rousseau. A dual bounding framework through cost splitting for binary quadratic optimization. *INFORMS Journal on Computing*, 36(6):1501–1521, 2024. doi:10.1287/ijoc.2021.0186.

[2] E. Bettiol, I. Bomze, L. Létocart, F. Rinaldi, and E. Traversi. Mining for diamonds—matrix generation algorithms for binary quadratically constrained quadratic problems. *Computers & Operations Research*, 142(105735), 2022. doi:10.1016/j.cor.2022.105735.

[3] A. Ceselli, L. Létocart, and E. Traversi. Dantzig–Wolfe reformulations for binary quadratic problems. *Mathematical Programming Computation*, 14(1):85–120, Mar 2022. doi:10.1007/s12532-021-00206-w.

[4] J. Charfreitag, M. Jünger, S. Mallach, and P. Mutzel. McSparse: Exact solutions of sparse maximum cut and sparse unconstrained binary quadratic optimization problems. In C. A. Phillips and B. Speckmann, editors, *2022 Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX)*, pages 54–66, 2022. doi:10.1137/1.9781611977042.5.

[5] W. W. Chu. Optimal file allocation in a multiple computer system. *IEEE Trans. Comput.*, 18(10):885—-889, Oct 1969. doi:10.1109/T-C.1969.222542.

[6] G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors. *Column Generation*. Springer New York, NY, 1 edition, 2005. doi:10.1007/b135457.

[7] R. Fortet. L'algèbre de boole et ses applications en recherche opérationnelle. *Cahiers du Centre d'Etudes de Recherche Opérationnelle*, 1:5–36, 1959.

[8] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1 edition, 1979.

[9] F. Glover and E. Woolsey. Converting the 0–1 polynomial programming problem to a 0–1 linear program. *Oper. Res.*, 22(1):180–182, Feb. 1974. doi:10.1287/opre.22.1.180.

[10] P. L. Hammer, P. Hansen, and B. Simeone. Roof duality, complementation and persistency in quadratic 0–1 optimization. *Mathematical Programming*, 28(2):121–155, Feb 1984. doi:10.1007/BF02612354.

[11] S. Mallach. Inductive linearization for binary quadratic programs with linear constraints. *4OR*, 19(4):549–570, Dec 2021. doi:10.1007/s10288-020-00460-z.

[12] S. Mallach. Inductive linearization for binary quadratic programs with linear constraints: A computational study. *4OR*, Mar 2023. doi:10.1007/s10288-023-00537-5.

[13] G. R. Mauri and L. A. N. Lorena. A column generation approach for the unconstrained binary quadratic programming problem. *European Journal of Operational Research*, 217(1):69–74, 2012. doi:10.1016/j.ejor.2011.09.016.

[14] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I — convex underestimating problems. *Mathematical Programming*, 10(1):147–175, Dec 1976. doi:10.1007/BF01580665.

[15] D. Rehfeldt, T. Koch, and Y. Shinano. Faster exact solution of sparse maxcut and qubo problems. *Mathematical Programming Computation*, Apr 2023. doi:10.1007/s12532-023-00236-6.

[16] J. M. W. Rhys. A selection problem of shared fixed costs and network flows. *Management Science*, 17(3):200–207, 1970.

A Obstacles in Column Generation directly applied to the Standard Linearization

In this section, we elucidate some critical obstructions that one may and typically will encounter when devising a conventional CG approach for (StdLin) directly.

Since these obstructions remain when restricting the inequalities of the LP relaxation of (StdLin) to those that are relevant for the sets P and N (see also Section 2), and since this facilitates a comparison with our approach as presented in Section 3, we first rewrite the relaxation accordingly.

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in P\cup N} d_{ij}w_{ij} + \sum_{i=1}^{n} c'_i x_i \\
\text{s.t.} \quad & -w_{ij} + x_i \geq 0 && \text{for all } (i,j)\in N \quad [v_{ij}^{N1}] && (29)\\
& -w_{ij} + x_j \geq 0 && \text{for all } (i,j)\in N \quad [v_{ij}^{N2}] && (30)\\
& -x_i - x_j + w_{ij} \geq -1 && \text{for all } (i,j)\in P \quad [v_{ij}^{P1}] && (31)\\
& w_{ij} \geq 0 && \text{for all } (i,j)\in P \quad [v_{ij}^{P2}] && (32)\\
& -x_i \geq -1 && \text{for all } i\in\{1,\dots,n\} \quad [z_i] \\
& x_i \geq 0 && \text{for all } i\in\{1,\dots,n\}
\end{aligned}
$$

When treating the linearization variables as free variables, i.e., (32) is not considered as a variable bound but rather as a usual constraint (that only applies to P), the dual is then as follows:

$$\begin{aligned}
&\max\ -\sum_{(i,j)\in P} v_{ij}^{P1} - \sum_{i=1}^{n} z_i \\
&\text{s.t.} \sum_{(i,j)\in N} v_{ij}^{N1} + \sum_{(j,i)\in N} v_{ji}^{N2} \\
&\quad -\sum_{(j,i)\in P} v_{ji}^{P1} - \sum_{(i,j)\in P} v_{ij}^{P1} - z_i \le c'_i && \text{for all } i \in \{1,\dots,n\}\ [x_i] && (33)\\
&v_{ij}^{P1} + v_{ij}^{P2} = d_{ij} && \text{for all } (i,j) \in P\ [w_{ij}] && (34)\\
&-v_{ij}^{N1} - v_{ij}^{N2} = d_{ij} && \text{for all } (i,j) \in N\ [w_{ij}] && (35)\\
&v_{ij}^{P1}, v_{ij}^{P2} \ge 0 && \text{for all } (i,j) \in P\\
&v_{ij}^{N1}, v_{ij}^{N2} \ge 0 && \text{for all } (i,j) \in N\\
&z_i \ge 0 && \text{for all } i \in \{1,\dots,n\}
\end{aligned}$$

Suppose now that we have solved a restricted version of the primal LP relaxation that consists of a strict subset of the linearization variables and constraints for N. For simplicity, we can assume that we have solved the one comprising only the x-variables,

$$\min\left\{\sum_{i=1}^{n} c'_i x_i : 0 \le x_i \le 1 \text{ for all } i \in \{1,\dots,n\}\right\},$$

while $N \neq \emptyset$. Consider now incorporating $(i,j) \in N$ which means to add w_{ij} and the according constraints (29) and (30) to the primal program. In the dual program, we add v_{ij}^{N1}, v_{ij}^{N2}, and the according equation (35). However, while the canonical extension with $w_{ij} = 0$ preserves primal feasibility, the dual solution canonically extended with $v_{ij}^{N1} = v_{ij}^{N2} = 0$ is infeasible and assigns to w_{ij}, via its violated dual constraint (35), a reduced cost of $d_{ij} + v_{ij}^{N1} + v_{ij}^{N2} < 0$.

As a consequence, for *every* so far neglected $(i,j) \in N$, the variable w_{ij} is indicated as an entering candidate to potentially reduce the primal objective (move towards a valid lower bound), generally requiring subsequent pivoting with the primal simplex algorithm to deduce whether a strict decrease actually takes place. If $x_i > 0$ and $x_j > 0$ in the current primal solution, then we even have an immediate certificate that the current objective is not a valid lower bound because setting $w_{ij} = \min\{x_i, x_j\}$ gives a primal feasible extension with a strictly lower objective. In every other case, i.e., considering a so far neglected $(i,j) \in N$ with $x_i = 0$ or $x_j = 0$ (or both) in the current primal solution, additional effort (to avoid or do pivoting) is inevitable in order to rule out a potential strict decrease of the objective without reoptimization. More concretely, one can only avoid pivoting by testing whether dual feasibility can be retained by means of altering only the variables v_{ij}^{N1} and v_{ij}^{N2}, as this will keep the dual objective unchanged, which then implies that the reoptimized primal objective cannot be strictly smaller. Otherwise, reoptimization is necessary to validate the optimum objective value. To make this explicit, retaining dual feasibility means that we have to assign values to v_{ij}^{N1} and v_{ij}^{N2} such that (35) is satisfied *and* the instances of (33) associated with i and j remain satisfied as well. To be able to do this, we thus need that the current (non-negative) slacks of these two constraints, say s_i and s_j, satisfy $s_i + s_j \geq -d_{ij}$. In this case, the quantity $-d_{ij}$ can be accordingly distributed by setting $v_{ij}^{N1} = t$ and $v_{ij}^{N2} = -d_{ij} - t$ for any $t \in [\max\{0, -d_{ij} - s_j\}, \min\{s_i, -d_{ij}\}]$, which is feasible and does not affect the dual objective. However, if $s_i + s_j < -d_{ij}$, then we have to alter at least one of the variables z_i and z_j. This can only mean to increase z_i or z_j in order to allocate additional slack in the respective instances of (33) which can then be consumed by v_{ij}^{N1} respectively v_{ij}^{N2}. Since the latter variables do not contribute to the objective function, but the z-variables do with coefficient -1, this inevitably causes the dual objective to *decrease*. Specifically, a resolution with minimum decrease is as follows:

$$\begin{aligned}
v_{ij}^{N1} &= s_i \\
v_{ij}^{N2} &= -d_{ij} - s_i \\
z_i^{\text{new}} &= z_i \\
z_j^{\text{new}} &= z_j - d_{ij} - s_i - s_j
\end{aligned}$$

The objective value then decreases by exactly $-d_{ij} - s_i - s_j > 0$. Alternatively, one may exhaust the slack of the other instance of (33) and let z_i absorb the resulting deficit while the objective deterioration is the same. So in either case, reoptimization will then be necessary in order to deduce whether the old objective value could be restored. In fact, a corresponding situation with $s_i + s_j < -d_{ij}$ is natural to occur which is particularly easy to see when temporarily neglecting the dual P-variables. Then, for instance if $c'_i \leq 0$ and $c'_j \leq 0$, one must have $z_i = -c'_i$, $z_j = -c'_j$ and thus no slack that can be consumed by (further) dual N-variables. Similarly, if $c'_i > 0$ (or $c'_j > 0$) but also $-d_{ij} > c'_i$ (or $-d_{ij} > c'_j$), z_i respectively z_j has to be increased in order to obtain enough slack in the respective instance of (33). Naturally, such situations may already occur when starting from the restricted primal LP with only x-variables.

Overall, we conclude that we may be forced to introduce the variables and constraints for several $(i, j) \in N$, with little control on their actual number, before arriving at a valid lower bound. Moreover, dual feasibility is inevitably lost by the canonical extension of the primal and dual bases that results from keeping the introduced variables at zero. Only in very specific situations, we may deduce rather cheaply whether introducing some $(i, j) \in N$ will either definitely render the current objective value invalid as a lower bound or can be safely treated so as to *not* decrease the current objective. In any case, one pair (i, j) of the first category or one that does not belong to the second category suffices in order to cause additional reoptimization effort towards obtaining a valid lower bound.