Communications in Combinatorics and Optimization
Vol. xx, No. x (xxxx), pp. 1-21
https://doi.org/10.22049/cco.2026.30575.2540

CCO
Commun. Comb. Optim.

Research Article

Efficient semidefinite relaxation for Boolean quadratic programming problems with generalized upper bound constraints via row-by-row method

Hitarth Sharma[†,*], **Rupaj K. Nayak**[‡]

Department of Mathematics, IIIT Bhubaneswar, Odisha, India
[†]hitarth@iiit-bh.ac.in
[‡]rupaj@iiit-bh.ac.in

Received: 30 April 2025; Accepted: 5 February 2026
Published Online: 8 February 2026

Abstract: This study focuses on the low-complexity implementation of semidefinite relaxation (SDR) to generate bounds for the Boolean Quadratic Programming Problem with Generalized Upper Bound Constraints (BQP-GUB). Most current SDR approaches rely on interior-point methods (IPM), which, despite having worst-case polynomial complexity, can be computationally expensive in practice. We depart from the IPM framework and investigate the use of other low per-iteration-complexity techniques for the solution of BQP-GUB. Specifically, we apply the row-by-row (RBR) method, called NuclearRBR, to solve the semidefinite programs that emerge from reformulating the BQP-GUB as an unconstrained Boolean Quadratic Programming Problem (UBQP). In this formulation, a nonconvex rank-one constraint is relaxed by a convex nuclear norm constraint. The RBR method only requires matrix-vector multiplications in each iteration, making it highly efficient. Numerical results demonstrate that NuclearRBR outperforms the semidefinite dual (SDD) method and other similar existing methods like SDcutRBR method [R.K. Nayak and N.K. Mohanty, Improved row-by-row method for binary quadratic optimization problems, Ann. Oper. Res. 275 (2019), 2, 587–605].

Keywords: BQP-GUB, SDP, Nuclear norm, RBR method.

AMS Subject classification: 90C20, 90C22, 90C26, 90C27

1. Introduction

In this research, we examine the Boolean Quadratic Programming Problem with Generalized Upper Bound Constraints (BQP-GUB), a generalization of the quadratic

* *Corresponding Author*

semi-assignment problem (QSAP) for its solution by the Row-by-Row (RBR) method. Its applications include various domains, such as biology, engineering, and production planning [30]. To be more precise, BQP-GUB is used in co-clustering of image segments [26], correlation clustering [5], scheduling [4], asset allocation to tasks [7], and metric labeling [15], among other applications.

The BQP-GUB is closely related to the unconstrained Boolean quadratic programming problem (UBQP) as it is possible to formulate a BQP-GUB as a UBQP [18, 28]. UBQP is NP-Hard in general, even though there are a few special circumstances where it can be solved in polynomial time. Thus, one has to use relaxation methods that give suboptimal solutions to obtain the optimal solution. A BQP-GUB can be relaxed using both semidefinite and spectral relaxation techniques. Despite their widespread use in image segmentation [25, 32] and motion segmentation [17], spectral methods have certain limitations, such as loose bounds in many situations [11, 14, 16]. Therefore, semidefinite relaxation (SDR) methods are preferred over them as they produce tighter bounds than spectral methods in various problems such as image segmentation [12], co-segmentation [13], and sub-graph matching [23].

Recent advances include fast r-flip neighborhood evaluations [1] and multi-wave tabu search methods [24]. Fast r-flip evaluations accelerate local search for BQP-GUB by using closed-form expressions to compute move gains, reducing the computational complexity from $\mathcal{O}(n^r)$ to constant time per candidate. An et al. [1] reported significant speedups for tabu search on large-scale instances while maintaining solution quality.

Multi-wave tabu search (MWTS) integrates forward and reverse intensification waves with an Active Move Record (AMR) to dynamically prioritize candidate moves. Shang et al. [24] demonstrated that MWTS improves upper bounds on six Taillard benchmark instances through hybrid perturbation strategies and adaptive tabu tenure mechanisms.

It is well established that interior-point methods (IPMs) are computationally expensive for large-scale semidefinite programs (SDPs). In contrast, the RBR method proposed by Wen *et al.* [31] offers low per-iteration complexity [22, 31], making it a practical alternative for large-scale SDPs. Moreover, RBR has proven effective for combinatorial optimization problems; for instance, Dhouib [6] successfully applied it to the traveling salesman problem. Exploiting its first-order structure and computational efficiency, we employ an enhanced RBR method to solve the SDPs in this study.

1.1. The problem

Suppose $A = \{1, 2, \ldots, n\}$ is a finite set of first n natural numbers and p_i is a profit assigned to each $i \in A$ and r_{ij} is a profit assigned to each $(i, j) \in A \times A$. The linear vector $p = (p_1, p_2, \cdots, p_n)$ is called the profit vector and the $n \times n$ square matrix $R = (r_{ij})$ is called the quadratic profit matrix. Without loss of generality, we can assume that the matrix R is symmetric. If not, R can be substituted with the symmetric matrix $\frac{1}{2}(R + R^\top)$, which does not affect the optimal solution set and

the objective value. Let $C_1, C_2, \ldots, C_m$ be partitions of the set A i.e. $C_1, C_2, \ldots, C_m$ are mutually disjoint subsets of A and $A = \cup_{i=1}^{m} C_i$. Then the standard form of a BQP-GUB problem [30] is

$$\begin{aligned} \max \ f(x) &= \sum_{i=1}^{n} p_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} r_{ij} x_i x_j \\ \text{s.t.} \ \sum_{j \in C_k} x_j &= 1 \quad \text{for} \quad k = 1, 2, \ldots, m \\ x_j &\in \{0, 1\} \quad \text{for } j = 1, 2, \ldots, n. \end{aligned} \tag{1.1}$$

where the solution vector $x \in \{0,1\}^n$ is to be determined.
BQP-GUB can be interpreted in graph-theoretic terms as follows: considering an undirected graph $G = (V, E)$. The set of vertices $V = \{1, 2, \ldots, n\}$ is partitioned into m subsets, $C_1, C_2, \ldots, C_m$. By choosing, $r_{ij} = s_{ij}$ if $(i, j) \in E$ and $r_{ij} = 0$ if $(i, j) \notin E$, the cluster restricted maximum induced subgraph problem (CMISP) is to find a subset S of V such that S contains exactly one element from each C_k, $k = 1, 2, ..., m$ and the sum of the weights of the edges in the subgraph of G induced by S is maximized (see Fig. 1). Thus, CMISP can be solved as a BQP-GUB.

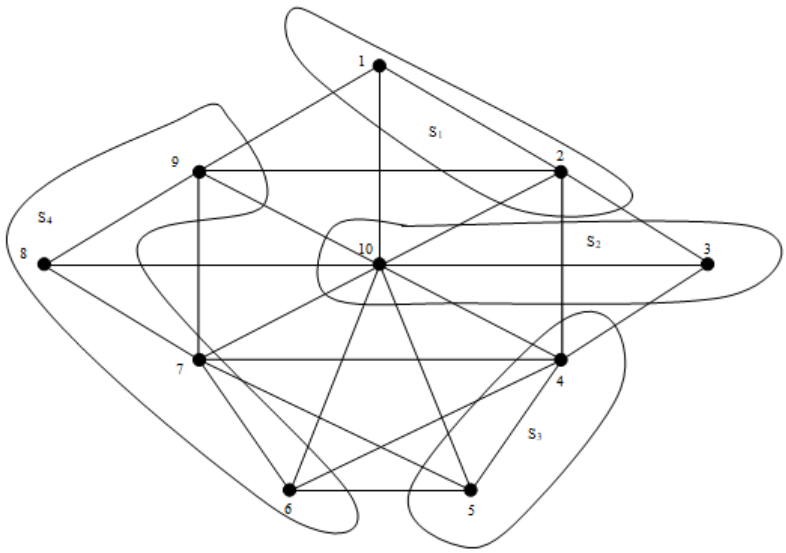

Figure 1. A CMISP with 10 vertices. The objective function is $\sum_{j=1}^{10} c_j x_j + 2x_1x_2 + 2x_1x_9 + 2x_1x_{10} + 2x_2x_3 + 2x_2x_4 + 2x_2x_9 + 2x_2x_{10} + 2x_3x_4 + 2x_3x_{10} + 2x_4x_5 + 2x_4x_6 + 2x_4x_7 + 2x_4x_{10} + 2x_5x_6 + 2x_5x_7 + 2x_5x_{10} + 2x_6x_7 + 2x_6x_{10} + 2x_7x_8 + 2x_7x_9 + 2x_7x_{10} + 2x_8x_9 + 2x_8x_{10} + 2x_9x_{10}$. The constraints are $x_1 + x_2 = 1$, $x_3 + x_{10} = 1$, $x_4 + x_5 = 1$, $x_6 + x_7 + x_8 + x_9 = 1$, $x_j \in \{0, 1\}$ for $j = 1, 2, \ldots, 10$. The SDP solution of it is 10.4538 and the exact solution of this problem is 10 with $x_2 = 1, x_4 = 1, x_6 = 1, x_{10} = 1$ or $x_2 = 1, x_4 = 1, x_7 = 1, x_{10} = 1$ or $x_2 = 1, x_4 = 1, x_9 = 1, x_{10} = 1$ and all other variables 0.

If $|C_k| = m$ for all $k = 1, 2, \ldots, m$ and $n = m^2$ it reduces to QSAP. Since QSAP is NP-hard, it follows that the BQP-GUB problem is also NP-hard. Wang and Punnen in [30] discussed the complexity and solvability of BQP-GUB and proved that BQP-GUB is strongly NP-hard even when $|C_k| \leq 2$ for all $k = 1, 2, \ldots, m$, R is of rank one and $p = (p_i)$ is the zero vector.

The contributions of this paper are mainly on establishing a first order method for the solution of rank-constrained Boolean quadratic programming problem with generalized upper bound constraints. The structure of the paper is as follows: After introducing the problem in Section 1, we present a novel UBQP formulation. Then, we obtain the Lagrangian relaxation and rank relaxation of the SDP generated from the UBQP in Section 2. In Section 3, we derive nuclear norm relaxation and introduce a parameter selection procedure. In Section 4, we discuss a modified RBR method for the problem. Finally, numerical experiments on BQP-GUB, CMISP, and UBQP datasets are performed with some concluding remarks given in Section 5.

2. UBQP formulation and some relaxations

Wang and Punnen [30] reformulated the BQP-GUB with n variables as a UBQP using the state-of-the-art path relinking algorithm (PR1) [28]. However, the converted UBQP is not an equivalent model of BQP-GUB [30]. Departing from the path relinking algorithm, our approach relies on an equivalent UBQP model whose rank is equal to the rank of R in BQP-GUB.

The objective function $f(x) = \sum_{i=1}^{n} p_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} r_{ij} x_i x_j$ of BQP-GUB, is written as $f(x) = p^\top x + x^\top R x$. Since $x_i x_j = x_j x_i$, the matrix R can be preprocessed to be in a lower triangular matrix [8], and henceforth we assume that R is a lower triangular matrix. Since x is binary, so $x_i^2 = x_i, \ \ \forall i = 1, \ldots n$. Therefore, the objective function $f(x)$ is represented as

$$\begin{aligned} f(x) =& \sum_{i=1}^{n} p_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{i} r_{ij} x_i x_j \\ =& \sum_{i=1}^{n} p_i x_i^2 + \sum_{i=1}^{n} \left(x_i r_{ii} x_i + \sum_{j=1}^{i-1} x_i r_{ij} x_j \right) \\ =& \sum_{i=1}^{n} \left((p_i + r_{ii}) x_i^2 + \sum_{j=1}^{i-1} x_i r_{ij} x_j \right) \\ =& x^\top \bar{Q} x \end{aligned}$$

where the lower triangular matrix $\bar{Q} = [\bar{q}_{ij}]$ $\forall i, j \in 1, 2, \ldots, n$ such that

$$\bar{q}_{ij} = \begin{cases} p_i + r_{ij} & \text{if } i = j, \\ r_{ij} & \text{otherwise,} \end{cases}$$

Thus, the problem (1.1) can be written as

$$\begin{aligned} \max\ & f(x) = x^\top \bar{Q} x \\ \text{s.t.}\ & \sum_{j \in C_k} x_j = 1 \quad \text{for} \quad k = 1, 2, \ldots, m \\ & x_j \in \{0, 1\} \quad \text{for } j = 1, 2, \ldots, n. \end{aligned} \tag{2.1}$$

Again, using the binarity of x_i, the Lagrangian function can be written as

$$\begin{aligned} L(x, \alpha) &= x^\top \bar{Q} x - \alpha \sum_{k=1}^{m} \left(\sum_{j \in C_k} x_j^2 - 1 \right) \\ &= \sum_{i=1}^{n} \left((p_i + r_{ii})\, x_i^2 + \sum_{j=1}^{i-1} x_i r_{ij} x_j \right) - \alpha \sum_{k=1}^{m} \left(\sum_{j \in C_k} x_j^2 - 1 \right) \\ &= x^\top \tilde{Q} x + \alpha m \end{aligned}$$

where the lower triangular matrix $\tilde{Q} = [\tilde{q}_{ij}]\ \forall i, j \in 1, 2, ..., n$ such that

$$\tilde{q}_{ij} = \begin{cases} p_i + r_{ij} - \alpha & \text{if } i = j \\ r_{ij} & \text{otherwise,} \end{cases}$$

and we choose

$$\alpha = 1 + \sum_{i=1}^{n} \sum_{j=1}^{n} \max(p_i + r_{ij}, 0).$$

Thus, the UBQP relaxation of BQP-GUB (1.1)is:

$$\begin{aligned} \max\quad & x^\top \tilde{Q} x \\ \text{s.t.}\ & x_j \in \{0, 1\} \text{ for } j = 1, 2, \ldots, n. \end{aligned} \tag{2.2}$$

It is clear that the main diagonal of the matrix $\tilde{Q}$ is nonzero. Since $\tilde{Q}$ and Q are lower triangular matrices with nonzero main diagonals, therefore, they have the same rank. Thus, we can also claim that $\tilde{Q}$ preserves the properties of $\bar{Q}$ and so are equivalent. Since $x \in \{0, 1\}^n$ can be replaced by $x_j^2 - x_j = 0,\ j = 1, 2..., n$ and replacing $\tilde{Q}$ by $R_s = -\frac{1}{2}(\tilde{Q} + \tilde{Q}^\top)$ in the case when $\tilde{Q}$ is not a symmetric matrix, we obtain the following problem

$$\begin{aligned} \min\ & x^\top R_s x \\ \text{s.t.}\ & x_j^2 - x_j = 0,\ j = 1, 2..., n. \end{aligned} \tag{2.3}$$

If $f(x_*)$ is the objective value of the problem (2.3), then $-f(x_*) + \alpha m$ is the objective value of the problem (2.2). Note that R_s is negative, and the problem (2.3) is a minimization problem while problem (2.2) is a maximization problem.

2.1. Lagrangian relaxation of UBQP

The Lagrangian of (2.3) is

$$\begin{aligned} L(x,\lambda) &= x^\top R_s x + \sum_{i=1}^{n} \lambda_i (x_i^2 - x_i) \\ &= x^\top (R_s + D_\lambda) x - \lambda^\top x \end{aligned}$$

where D_λ is a diagonal matrix with the main diagonal as the nonnegative vector λ. Using Schur complements, the dual of the problem (2.3) is

$$\begin{aligned} &\max \ \gamma \\ &\text{s.t.} \ \begin{bmatrix} R_s + D_\lambda & -\frac{\lambda}{2} \\ -\frac{\lambda^\top}{2} & -\gamma \end{bmatrix} \succeq 0 \\ &\qquad \lambda_i \geq 0, \ i = 1, 2, \ldots, n. \end{aligned} \tag{2.4}$$

Our objective is to solve the SDP by the RBR method, where the positive semidefinite matrices have only diagonal elements equal to 1. For this, we have to change the domain of x from $\{0,1\}$ to $\{-1,1\}$ and the required substitution for the domain change is $y = 2x - e$, where e is the vector of 1's.
Then problem (2.3) can be written as

$$\begin{aligned} &\min \ \left(\frac{y+e}{2}\right)^\top R_s \left(\frac{y+e}{2}\right) \\ &\text{s.t.} \ y \in \{-1,1\}^n, \end{aligned} \tag{2.5}$$

which can be simplified as

$$\begin{aligned} &\min \ y^\top \left(\frac{1}{4} R_s\right) y + \frac{1}{2} (R_s e)^\top y + e^\top \left(\frac{1}{4} R_s\right) e \\ &\text{s.t.} \ y \in \{-1,1\}^n, \end{aligned} \tag{2.6}$$

Since the problem (2.6) is a nonconvex programming problem, it needs a relaxation. Letting $z = \begin{bmatrix} y \\ y_{n+1} \end{bmatrix}$ and $C = \begin{bmatrix} \frac{1}{4} R_s & \frac{1}{4} R_s e \\ \frac{1}{4} (R_s e)^\top & e^\top \left(\frac{1}{4} R_s\right) e \end{bmatrix}$, the problem (2.6) can be written as:

$$\begin{aligned} &\min \ z^\top C z \\ &\text{s.t.} \ z \in \{-1,1\}^{n+1}. \end{aligned} \tag{2.7}$$

The UBQP problem has been addressed using various solution techniques, including approximation methods, heuristics, and metaheuristics. In the approximation solution techniques, some popular methods are semidefinite relaxation and its variants.

Similarly, for heuristic and metaheuristic solution techniques, variants of tabu search [3, 18, 19, 29, 34], simulated annealing [3], and genetic algorithm [3, 19] have been extensively used to solve the UBQP. In the current paper, we focus on the semidefinite relaxation technique (SDR) as it provides a robust solution even when the problem is nonconvex. However, due to its per iteration complexity, it is not advisable to choose SDR for large scale UBQP.

2.2. SDP relaxation of UBQP

Since $z \in \{-1, 1\}^{n+1}$ is equivalent to $z_i^2 = 1$ for $i = 1, 2, \ldots, n+1$, letting $Z = zz^\top$ the problem (2.7) can be written as

$$\begin{aligned} &\min \quad C \bullet Z \\ &\text{s.t. } \operatorname{diag}(Z) = e, \\ &\operatorname{rank}(Z) = 1, \\ &Z \succeq 0. \end{aligned} \tag{2.8}$$

Note that $Z = zz^\top$ implies $rank(Z) = 1$. The rank constraint yields the problem (2.8) as a nonconvex one. The challenge in solving the nonconvex optimization problem lies in relaxing the nonconvex rank constraint. Several authors have proposed rank relaxation methods, primarily in two forms: either by dropping the rank constraint or by convexifying it. In this work, We prefer the rank relaxation technique over the dropping one to get a tight SDP solution.

3. Nuclear norm relaxation

Definition 1. The nuclear norm of a matrix is the sum of its singular values.

In problem (2.8), we seek a matrix with rank constrained to one. Since for the problems where the rank of a matrix variable is desired to be kept small, and which are otherwise convex, the nuclear norm can often serve as a convex substitute for the rank. So we rely on it for the convex relaxation of the rank constraint.
Let $\Omega(n+1) = \{Z \in S_{n+1} \mid Z \succeq 0, Z = zz^\top \text{where } z \in \{-1, 1\}^{n+1}\}$. Note that $Z = zz^\top$ implies $\operatorname{trace}(Z) = n+1$ and $\Omega(n+1)$ is the intersection of a linear subspace (i.e. $\operatorname{trace}(Z) = n+1$) and the p.s.d. cone. For the set $\Omega(n+1)$, we have the following theorem.

Theorem 1. *For $Z \in \Omega(n+1)$, $\|Z\|_* = n+1$ holds if and only if $rank(Z) = 1$, where $\|\cdot\|_*$ is the nuclear norm.*

The theorem is trivial because from the definition of $\Omega(n+1)$, it is clear that every matrix from this set has rank 1 and is a positive semidefinite matrix. Also, the diagonal of these matrices has all ones. Therefore, the nuclear norm of such a matrix is the

sum of the eigenvalues, which is equal to the trace of the matrix. This theorem shows that the rank-one constraint is equivalent to $\|Z\|_* = n+1$ for positive semidefinite matrices with a fixed trace. Thus, the nonconvex spherical constraint $\|Z\|_* = n+1$ is relaxed to the convex inequality constraint $\|Z\|_* - (n+1) \leq 0$.
Thus, the convex reformulation of problem (2.8) can be expressed as:

$$\begin{aligned} &\min\ C \bullet Z \\ &\text{s.t. } \operatorname{diag}(Z) = e, \\ &\|Z\|_* - (n+1) \leq 0, \\ &Z \succeq 0, \end{aligned} \tag{3.1}$$

where $C \bullet Z$ means $\operatorname{trace}(CZ^\top)$. Introducing the nuclear norm penalty term, we can modify the problem (3.1) as

$$\begin{aligned} &\min\ C \bullet Z + \gamma(\|Z\|_* - (n+1)) \\ &\text{s.t. } \operatorname{diag}(Z) = e, \\ &Z \succeq 0, \end{aligned} \tag{3.2}$$

where γ is the regularization parameter.

3.1. Parameter Selection

The optimal solution of problem (3.2), as discussed in [20], is strongly dependent on the value of γ. The authors showed that formulating an optimization problem to minimize γ, subject to the dual constraints and using both γ and the dual variables as optimization variables, results in a convex problem. They also demonstrated that the objective function decreases as γ approaches 0 and increases as γ moves away from 0 in either direction. In this work, we adopt a similar approach for selecting the parameter γ. Since Z is square and positive semidefinite matrix, then $\|Z\|_* = \operatorname{trace}(Z)$. Thus, SDP (3.2) can be presented as

$$\begin{aligned} &\min\quad C \bullet Z + \gamma(I \bullet Z - (n+1)) \\ &\text{s.t. } (f_i f_i^\top) \bullet Z - 1 = 0 \quad \text{for} \quad i = 1, 2, \ldots, n+1, \\ &Z \succeq 0, \end{aligned} \tag{3.3}$$

where f_i is a square matrix of order $(n+1) \times (n+1)$ with diagonal element in $(i,i)^{\text{th}}$ position is 1 and other elements 0.
To obtain the value of γ, first, we derive the dual of the SDP (3.3), which is

$$\begin{aligned} &\max\quad \tilde{e}^\top \lambda \\ &\text{s.t. } C + \gamma I + \sum_{i=1}^{n} \lambda_i A_i \succeq 0 \quad \text{for} \quad i = 1, 2, \ldots, n \\ &\gamma \in \mathbb{R}, \lambda \in \mathbb{R}^n_+ \\ &\text{where}\ \ \tilde{e}\ \ \text{is a } n \times 1 \text{ vector of -1's,} \end{aligned} \tag{3.4}$$

We construct the parameter minimization problem by replacing the objective function of the problem (3.4) by $\min \gamma$. Therefore, the parameter minimization problem is

$$\begin{aligned} &\min \quad \gamma \\ &\text{s.t. } C + \gamma I + \sum_{i=1}^{n+1} \lambda_i (f_i f_i^\top) \succeq 0, \\ &\gamma \in \mathbb{R}, \lambda \in \mathbb{R}^n. \end{aligned} \tag{3.5}$$

Assuming the problem (3.5) is feasible and possesses a bounded solution, we denote the solution of the SDP by γ_{min}, which is finite. Now, one can obtain the best value of γ, denoted by γ_{best}, which we discussed in the numerical experiment section.

4. RBR method

Since problems (3.2) and (3.3) are equivalent, we focus on describing the RBR method for problem (3.2). The RBR method, introduced by Wen et al. [31], is a block coordinate descent algorithm that utilizes the special diagonal constraint structure in the SDP relaxation to reduce the complexity of the updated iterates. In the research Wai et al. [27], the authors employed the log-barrier approach for problem (2.8) with a dropped constraint, which is given by:

$$\begin{aligned} &\min \ C \bullet Z - \sigma \log\det(Z) \\ &\text{s.t. } \operatorname{diag}(Z) = e, \\ &Z \succ 0. \end{aligned} \tag{4.1}$$

The authors Nayak and Mohanty in [22] modified the original RBR method by using the Frobenius norm in convexifying the rank constraint in (2.8), which they named as SDcutRBR. The authors consider rank-relaxed-log-barrier approach in (2.8), which they present as

$$\begin{aligned} &\min \ C \bullet Z - \rho(\|Z\|_F^2 - (n+1)^2) - \sigma \log\det(Z) \\ &\text{s.t. } \operatorname{diag}(Z) = e, \\ &Z \succ 0. \end{aligned} \tag{4.2}$$

In our approach, we consider the logarithmic barrier problem for the problem (3.2)

$$\begin{aligned} &\min \ C \bullet Z - \gamma(\|Z\|_*^2 - (n+1)^2) - \sigma \log\det(Z) \\ &\text{s.t. } \operatorname{diag}(Z) = e, \\ &Z \succ 0. \end{aligned} \tag{4.3}$$

Theorem 2. *If $f_\rho(Z)$ is the objective function of the problem (4.2) and $f_\gamma(Z)$ is the objective function of the problem (4.3) with $\rho = \gamma > 0$, then $f_\gamma(Z) - f_\rho(Z) \leq 0 \quad \forall Z \succ 0$.*

The proof is straightforward using the relation $\|X\|_* \geq \|X\|_F$ for any matrix X and choosing $\rho = \gamma$. In the case when $\rho \neq \gamma$ no conclusive evidence is seen between f_ρ and f_γ. Thus, taking maxima over Z we got $\max f_\gamma(Z) \geq \max f_\rho(Z)$.
The above theorem proves that the NuclearRBR method gives a better optimal value than the SDcutRBR method for a maximum BQP-GUB problem.

4.1. Procedure of the NuclearRBR method

We use the cyclic optimization over the rows (or columns) of Z at each step, keeping other variables of Z fixed for the solution of (4.3). Using the symmetry we can partition the matrix Z as $\begin{bmatrix} Z_{11} & y_1^\top \\ y_1 & B \end{bmatrix}$ and the matrix C as $\begin{bmatrix} C_{11} & c_1^\top \\ c_1 & \bar{C}_{11} \end{bmatrix}$ where $Z_{11} = 1, y_1 \in \mathbb{R}^n, B \in S^n$. We assume that the rank of the matrix B is $r(> 0)$. We now construct a subproblem of (4.3) based on its partial optimization with respect to the first row. From the theorem 1.12 and 1.20 of Zhang [33] it is clear that

$$Z \succeq 0 \text{ iff, } B \succeq 0 \quad \text{and } (Z/B) = 1 - y_1^\top B^\dagger y_1 \geq 0, \; y_1 \in \; \mathcal{R}(B),$$

where Z/B is the generalized Schur complement of B in Z, $B^\dagger$ is the Moore-Penrose pseudoinverse of B and $\mathcal{R}(B)$ is the range of B.
Thus, factorizing the matrix Z as

$$Z = \begin{bmatrix} 1 & y_1^\top B^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} (Z/B) & 0 \\ y_1 & B \end{bmatrix}$$

the first term of the objective function of (4.3) $C \bullet Z$ can be simplified as

$$C \bullet Z = C_{11} Z_{11} + 2c_1^\top y_1 + \text{Trace}((\bar{C}_{11})^\top B).$$

When $(Z/B) = 1 - y_1^\top B^\dagger y_1$ then $\det(Z) = \det(B)(1 - y_1^\top B^\dagger y_1)$ therefore

$$\log\det(Z) = \log\det(B) + \log(1 - y_1^\top B^\dagger y_1).$$

The objective function of the problem (4.3) can be simplified as

$$\begin{aligned} \phi(z) &= C \bullet Z + \gamma(\|Z\|_* - (n+1)) - \sigma \log\det(Z) \\ &= C_{11} Z_{11} + 2c_1^\top y_1 + \text{Trace}((\bar{C}_{11})^\top B) \\ &+ \gamma(1 + \text{Trace}(B) - (n+1)) \\ &- \sigma(\log\det(B) + \log(1 - y_1^\top B^\dagger y_1)) \\ &= 2c_1^\top y_1 - \sigma \log(1 - y_1^\top B^\dagger y_1) + g(B) \end{aligned}$$

where $g(B)$ is a function of B.
As we consider partial optimization w.r.t. first row of Z then RBR subproblem is formulated as

$$\min_{y_1} \; 2c_1^\top y_1 - \sigma \log(1 - y_1^\top B^\dagger y_1) \tag{4.4}$$

The following theorem describes the optimal solution of (4.4).

Theorem 3. *The optimal solution of (4.4) is given by*

$$y_1^* = \begin{cases} \frac{\sigma - \sqrt{\sigma^2 + 4\alpha}}{2\alpha} B c_1 & \text{when } \alpha > 0, \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha = c_1^\top Q_r \Lambda_r^\top Q_r^\top c_1$.

Proof. The matrix $B \in S_n$ is a real symmetric matrix with rank $r > 0$. Therefore B has spectral decomposition

$$B = Q_r \Lambda_r Q_r^\top$$

where Q_r is an orthogonal matrix and

$$\Lambda_r = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_r)$$

with $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r > 0$, hence the pseudoinverse of B is

$$B^\dagger = Q_r \Lambda_r^{-1} Q_r^\top.$$

Letting $w = Q^\top y_1$. Since $y_1 \in \mathcal{R}(B) = \mathcal{R}(Q_r)$ and Q_r is orthogonal, then y_1 can be expressed as $y_1 = Q_r w_r$. Thus the subproblem (4.4) is equivalent to

$$\min_{w_r} \; 2c_1^\top Q_r w_r - \sigma \log(1 - w_r^\top Q_r^\top Q_r \Lambda_r^{-1} Q_r^\top Q_r w_r)$$

which can be simplified as

$$\min_{w_r} \; 2c_1^\top Q_r w_r - \sigma \log(1 - w_r^\top \Lambda_r^{-1} w_r) \tag{4.5}$$

If w_r^* is an optimal solution of (4.5) then $\nabla w_r = 0$ at w_r^*. On simplification,

$$2(c_1^\top Q_r)^\top + \frac{\sigma}{1 - w_r^\top \Lambda_r^{-1} w_r} (\Lambda_r^{-1} + (\Lambda_r^{-1})^\top) w_r = 0$$

$$2Q_r^\top c_1 + \frac{2\sigma \Lambda_r^{-1} w_r}{1 - w_r^\top \Lambda_r^{-1} w_r} = 0$$

$$\text{which implies } Q_r^\top c_1 = -\frac{\sigma \Lambda_r^{-1} w_r}{1 - w_r^\top \Lambda_r^{-1} w_r}.$$

Letting $\theta = 1 - w_r^\top \Lambda_r^{-1} w_r$, we obtain

$$Q_r^\top c_1 = -\frac{\sigma \Lambda_r^{-1} w_r}{\theta}$$

which implies

$$\Lambda_r^{-1} w_r = -\frac{\theta}{\sigma} Q_r^\top c_1.$$

Thus, the optimal value w_r^* is given as

$$w_r^* = -\frac{\theta}{\sigma} \Lambda_r Q_r^\top c_1. \tag{4.6}$$

Substituting $w_r = -\frac{\theta}{\sigma} \Lambda_r Q_r^\top c_1$ in the equation $\theta = 1 - w_r^\top \Lambda_r^{-1} w_r$ we obtain

$$\begin{aligned}
\theta &= 1 - \left(-\frac{\theta}{\sigma} \Lambda_r Q_r^\top c_1\right)^\top \Lambda_r^{-1} \left(-\frac{\theta}{\sigma} \Lambda_r Q_r^\top c_1\right) \\
&= 1 - \frac{\theta^2}{\sigma^2} c_1^\top Q_r \Lambda_r^\top \Lambda_r^{-1} \Lambda_r Q_r^\top c_1 \\
&= 1 - \frac{\theta^2}{\sigma^2} c_1^\top Q_r \Lambda_r^\top Q_r^\top c_1.
\end{aligned}$$

Since $c_1^\top Q_r \Lambda_r^\top Q_r^\top c_1$ is a scalar, assuming $\alpha = c_1^\top Q_r \Lambda_r^\top Q_r^\top c_1$ we obtain

$$\theta = 1 - \frac{\theta^2}{\sigma^2} \alpha.$$

The solution of the quadratic equation $\theta = 1 - \frac{\theta^2}{\sigma^2} \alpha$ gives

$$\theta = \frac{\sigma \sqrt{\sigma^2 + 4\alpha} - \sigma^2}{2\alpha}.$$

Therefore, the optimal solution of (4.5) is

$$\begin{aligned}
y_1^* &= Q_r w_r^* \\
&= -\frac{\theta}{\sigma} Q_r \Lambda_r Q_r^\top c_1 \\
&= -\frac{\sqrt{\sigma^2 + 4\alpha} - \sigma}{2\alpha} Q_r \Lambda_r Q_r^\top c_1 \\
&= \frac{\sigma - \sqrt{\sigma^2 + 4\alpha}}{2\alpha} Q_r \Lambda_r Q_r^\top c_1 \\
&= \frac{\sigma - \sqrt{\sigma^2 + 4\alpha}}{2\alpha} B c_1.
\end{aligned}$$

□

We now present the improved RBR algorithm for the SDP with the nuclear norm, hereafter referred to as NuclearRBR. As demonstrated by Nayak and Biswal [21], the computational complexity of the NuclearRBR algorithm is $\mathbb{O}(n^3)$.

Algorithm 1 A NuclearRBR method for the problem (4.3)

Input: $Z^0 \succ 0, k := 1, \delta > 0, \sigma > 0, \theta > 0$ and $\lambda > 0$
Set $F^0 = C \bullet Z^0 + \gamma(\|Z^0\|_* - (n+1))$;
repeat
for $i = 1, 2, \ldots, n$ do
  Set $w = -\frac{\theta}{\sigma}\Lambda_r Q_r^\top c_i$ and $\alpha = \frac{\sigma}{\theta} c_i^\top Q_r w$ where Λ_r, Q_r and c_i are described in Theorem (3)
  Set $Z_{ii}^k = 1$
  if $\alpha > 0$ then compute $y_i^k = Q_r w$
  else set $y_i^k = 0$
  end if
end for
compute $F^k = C \bullet Z^k + \gamma(\|Z^k\|_* - (n+1))$, set $Z^{k+1} = Z^k$ and $k = k+1$ until $\left|\frac{F^{k+1} - F^k}{F^k}\right| \leq \delta$.

4.2. Convergence and complexity discussion

The row-by-row (RBR) method is globally convergent, as established in Theorem 3.4 of [31]; however, the result guarantees only asymptotic convergence and does not provide an explicit convergence rate. As a consequence, the number of iterations required to reach a prescribed accuracy may be large, reflecting the generally slow convergence behavior of the RBR method.

From a computational standpoint, each RBR iteration involves only basic matrix–vector multiplications and simple linear algebraic operations. Therefore, the per-iteration computational cost is on the order of $\mathcal{O}(n^3)$. Let K denote the total number of iterations required to achieve a desired solution accuracy. The overall computational complexity of the RBR method is then given by $\mathcal{O}(Kn^3)$.

Although the iteration count K may be relatively large, the low per-iteration cost makes the RBR method attractive for large-scale semidefinite relaxation problems, particularly when compared with interior-point methods whose per-iteration complexity can be significantly higher.

5. Numerical experiment

This section presents computational experiments to evaluate the performance of the proposed NuclearRBR algorithm, comparing it with the SDcutRBR and SDD (Semidefinite Dual) methods. To assess the efficiency of the algorithm, we consider a total of 137 problem instances. The structure of these instances is outlined below:

1. The 25 BQP-GUB instances of varying dimensions, ranging from 20 to 500, are taken from the BQP-GUB dataset. These instances are randomly generated following the procedure outlined by Wang and Punen [30]. The cardinalities of the sets C_k were chosen randomly such that $|C_k| \geq 2$ and $\sum_{k=1}^{m} |C_k| = n$. The

number of partitions m was also selected randomly, with the constraint that $m \leq n$.

2. The next 15 datasets consist of CMISPs with dimensions ranging from 20 to 50, taken from the CMISP dataset. These instances are randomly generated following the procedure outlined by Wang and Punen [30]. The coefficients p_i and r_{ij} are random binary values chosen from $\{0, 1\}$. The procedure used to select C_k in the BQP-GUB dataset is also applied here. As in the previous dataset, the number of partitions m is chosen randomly, subject to the constraint $m \leq n$.

3. The next UBQP dataset is taken from BQPLIB. This dataset contains the symmetric coefficient matrix A and vector b of the BQP form (2.6). It consists of 57 instances with dimensions of A being 30, 50, 100, 300, 500, and 1000.

4. The final BQP dataset is sourced from the Beasley dataset. This dataset contains a symmetric coefficient matrix A as described in (2.3). It includes 40 test problems with dimensions of A being 50, 100, 250, and 500.

Although the BQPLIB and Beasley datasets are designed for minimization problems, we apply our proposed method to these datasets to evaluate its efficiency. The numerical results presented here reflect the performance of the NuclearRBR method on BQP-GUB, CMISP, and UBQP problems. To compare the results with some benchmark solutions, we provide the result of the SDD described in (2.4). Further, we also provide the solution by SDcutRBR described in (4.2). All the algorithms in our experiments are coded in MATLAB and compiled on a laptop equipped with an Intel(R) Core(TM) i3 CPU 3.20 GHz and 4GB of memory.
The value of the parameter γ_{best} is calculated using the formula $\gamma_{\text{best}} = \gamma_{min} \cdot cp$ where cp is the controlling parameter to keep the value of γ_{best} close to zero. Note that cp is considered a factor of the dimension of the problem.
The maximum number of iterations is set as 1000, the stopping criteria is set as $\left|\frac{f^{k+1}-f^k}{f^k}\right| \leq \delta$ where $\delta = 10^{-4}$. To keep the same framework of SDcutRBR, we have taken the same parameter values that are chosen for our proposed NuclearRBR Algorithm.
The rows in Table 1 to 3 show the profile of the problem of all dimensions, their objective value, and total CPU time (in seconds) for all the considered methods. The dual solution of SDP is obtained by the solver Mosek [2] when used as a computational engines within the optimization modeling language CVX [9, 10].

Table 1. Comparison among different algorithms on the objective value and the execution time for the maximization problems from BQP-GUB with $n = 20, 50, 100, 200$, and 500

filename	obj Dual	obj SDRBR	obj NRBR	gamma best	time SDD	time SDRBR	time NRBR
bqpgub-20-1	342.06	515.48	538.62	0.907	0.7280	0.0057	0.6894
bqpgub-20-2	1081.47	1096.30	1167.47	1.793	0.7648	0.0039	0.6938
bqpgub-20-3	628.64	814.49	896.16	1.462	0.7703	0.0023	0.6892
bqpgub-20-4	1885.89	1930.02	2050.35	3.343	0.7892	0.0019	0.7094
bqpgub-20-5	652.19	787.69	907.15	1.552	0.7727	0.0086	0.7772
Average	**918.05**	**1028.79**	**1111.95**	**1.811**	**0.7650**	**0.0045**	**0.7118**
bqpgub-50-1	635.61	673.19	733.67	0.151	0.6414	0.0069	0.6139
bqpgub-50-2	2208.82	2619.95	2908.65	0.840	0.6200	0.0099	0.5592
bqpgub-50-3	2683.66	2730.26	3537.37	1.043	0.6965	0.0019	0.6015
bqpgub-50-4	2068.98	2569.34	2705.54	0.780	0.6805	0.0023	0.5861
bqpgub-50-5	5917.27	6448.87	6475.73	1.860	0.6136	0.0064	0.5945
Average	**2702.87**	**3008.32**	**3272.19**	**0.935**	**0.6504**	**0.0055**	**0.5910**
bqpgub-100-1	108643.03	109921.05	111905.15	8.334	0.6818	0.0285	0.6278
bqpgub-100-2	72438.39	73462.31	76631.92	5.739	0.7865	0.0215	0.7898
bqpgub-100-3	79519.26	84924.37	88249.82	6.715	0.8508	0.0144	0.7348
bqpgub-100-4	100862.06	106743.62	109067.75	8.238	0.8058	0.0167	0.7730
bqpgub-100-5	83682.74	85254.69	86454.55	6.425	0.6802	0.0143	0.6492
Average	**89029.09**	**92061.21**	**94461.83**	**7.090**	**0.7610**	**0.0191**	**0.7149**
bqpgub-200-1	54800.07	56773.09	63989.20	1.226	1.0681	0.0693	0.8855
bqpgub-200-2	64365.76	67687.17	76218.44	1.468	1.0533	0.0547	0.8515
bqpgub-200-3	87233.82	90264.40	96424.43	1.828	1.1257	0.0612	0.8995
bqpgub-200-4	23133.47	26989.31	27528.76	0.516	1.7971	0.1154	1.5400
bqpgub-200-5	24468.17	27311.37	29153.34	0.548	1.8150	0.0509	1.1322
Average	**50800.26**	**53805.07**	**58662.84**	**1.117**	**1.3719**	**0.0703**	**1.0617**
bqpgub-500-1	1218047.02	1222933.02	1234209.18	3.702	2.6944	2.2295	2.5269
bqpgub-500-2	525977.96	538856.47	549340.17	1.658	2.4854	2.2737	2.7579
bqpgub-500-3	1351357.28	1369216.84	1408215.59	4.262	2.9521	2.2188	2.4317
bqpgub-500-4	2086425.82	2093203.50	2172577.43	6.579	3.4934	2.8077	2.8503
bqpgub-500-5	1716472.44	1752811.44	1787746.65	5.413	3.0914	2.9244	2.4306
Average	**1379656.10**	**1395404.26**	**1430417.80**	**4.323**	**2.9434**	**2.4908**	**2.5995**

The BQP-GUB problems in our dataset are formulated as maximization problems, as defined in (1.1). As shown in Table 1, the proposed Nuclear RBR method consistently achieves higher objective values compared to the SDcutRBR and SDD methods, demonstrating its superior performance on the BQP-GUB instances.

Table 2. Comparison among different algorithms on the objective value and the execution time for the maximization problems from CMISP with $n = 20, 50$ and 100.

filename	obj Dual	obj SDRBR	obj NRBR	gamma best	time SDD	time SDRBR	time NRBR
cmisp-20-1	1504.00	1507.76	1532.45	2.482	0.6502	0.0116	0.6307
cmisp-20-2	469.00	479.73	516.39	0.691	0.6196	0.0066	0.6239
cmisp-20-3	1507.00	1509.26	1536.17	2.487	0.6151	0.0035	0.5874
cmisp-20-4	469.00	479.73	516.39	0.691	0.6164	0.0014	0.6005
cmisp-20-5	1501.00	1506.26	1528.74	2.477	0.7229	0.0106	0.7057
Average	**1090.00**	**1096.55**	**1126.03**	**1.766**	**0.6449**	**0.0067**	**0.6296**
cmisp-50-1	1572.00	1596.04	1623.71	0.418	0.3484	0.0121	0.3009
cmisp-50-2	956.00	1041.96	1089.13	0.273	0.3954	0.0081	0.3637
cmisp-50-3	1243.00	1335.62	1398.43	0.364	0.3593	0.0046	0.2846
cmisp-50-4	909.00	995.46	1041.75	0.272	0.4756	0.0048	0.2929
cmisp-50-5	1244.00	1336.12	1399.79	0.364	0.3222	0.0057	0.3041
Average	**1184.80**	**1261.04**	**1310.56**	**0.338**	**0.3802**	**0.0071**	**0.3093**
cmisp-100-1	44856.00	47061.24	51353.91	3.929	0.8312	0.0313	0.7880
cmisp-100-2	35518.00	36146.79	40759.94	3.112	0.8617	0.0164	0.7364
cmisp-100-3	27289.00	29414.74	36442.65	2.889	0.8893	0.0164	0.7366
cmisp-100-4	55279.00	56753.29	63159.78	4.838	1.3929	0.0153	0.8966
cmisp-100-5	44850.00	47058.24	51345.57	3.928	0.8055	0.0155	0.9042
Average	**41558.40**	**43286.86**	**48612.37**	**3.739**	**0.9561**	**0.0190**	**0.8124**

The CMISP problems in our dataset are formulated as maximization problems, as outlined in [30]. As reported in Table 2, the proposed Nuclear RBR method con-

sistently achieves higher objective values than both SDcutRBR and SDD, thereby demonstrating its effectiveness on the CMISP instances.

Table 3. Comparison among different algorithms on the objective value and the execution time for the minimization problems in UBQP with $n = 30, 50, 100, 300$ and 500.

filename	obj	obj	obj	gamma	time	time	time
	Dual	SDRBR	NRBR	best	SDD	SDRBR	NRBR
set-30-1	-1186.07	-1496.98	-2107.31	2.425	0.7229	0.0126	0.0058
set-30-2	-1324.39	-1594.40	-1796.37	1.986	0.7162	0.0040	0.0018
set-30-3	-1576.86	-1735.92	-2597.28	2.695	0.6742	0.0025	0.0013
set-30-4	-1362.70	-1634.69	-2207.74	2.385	0.7413	0.0028	0.0020
set-30-5	-1169.98	-1590.55	-1909.47	2.111	0.7303	0.0046	0.0045
set-30-6	-1233.96	-1574.99	-2005.05	2.231	0.6672	0.0012	0.0008
set-30-7	-1309.40	-1605.03	-2082.90	2.282	0.7206	0.0015	0.0010
set-30-8	-1442.68	-1684.30	-2117.44	2.234	0.7001	0.0014	0.0010
set-30-9	-1369.09	-1694.64	-2403.45	2.531	0.7062	0.0026	0.0007
set-30-10	-1321.99	-1693.35	-2035.33	2.136	0.6615	0.0015	0.0012
Average	**-1329.71**	**-1630.49**	**-2126.23**	**2.301**	**0.7040**	**0.0035**	**0.0020**
set-50-1	-1633.27	-1721.13	-1605.37	0.670	0.6990	0.0020	0.0016
set-50-2	-1656.13	-1728.51	-1672.26	0.693	0.7234	0.0028	0.0021
set-50-3	-1606.97	-1724.47	-1669.81	0.694	0.7046	0.0043	0.0020
set-50-4	-1574.15	-1728.01	-1676.73	0.695	0.7922	0.0020	0.0032
set-50-5	-1707.56	-1770.12	-1808.49	0.730	0.7344	0.0023	0.0021
set-50-6	-1611.92	-1720.54	-1688.91	0.703	0.7265	0.0021	0.0021
set-50-7	-1653.26	-1744.36	-1657.81	0.681	0.7824	0.0030	0.0026
set-50-8	-1583.61	-1684.30	-1565.88	0.669	0.7990	0.0032	0.0035
set-50-9	-1641.80	-1723.87	-1570.57	0.655	0.7116	0.0031	0.0016
set-50-10	-1653.74	-1698.23	-1649.51	0.696	0.7110	0.0027	0.0016
Average	**-1632.24**	**-1724.36**	**-1656.53**	**0.688**	**0.7384**	**0.0028**	**0.0022**
set-100-1	-4927.58	-5366.42	-5036.96	0.520	0.8068	0.0094	0.0046
set-100-2	-4687.29	-5398.16	-4821.48	0.495	0.8040	0.0119	0.0059
set-100-3	-4886.50	-5388.65	-5061.71	0.520	0.7797	0.0097	0.0039
set-100-4	-4890.31	-5350.25	-5064.77	0.524	0.8516	0.0099	0.0061
set-100-5	-4734.29	-5323.77	-4933.82	0.514	0.8176	0.0109	0.0037
set-100-6	-4936.22	-5407.43	-5153.77	0.527	0.8001	0.0109	0.0039
set-100-7	-4707.43	-5370.21	-4786.53	0.494	0.8017	0.0121	0.0067
set-100-8	-4810.70	-5391.00	-4959.11	0.509	0.8210	0.0096	0.0054
set-100-9	-4805.02	-5352.38	-5031.91	0.520	0.7993	0.0150	0.0065
set-100-10	-4740.16	-5407.95	-4887.80	0.501	0.8169	0.0104	0.0060
Average	**-4812.55**	**-5375.62**	**-4973.79**	**0.512**	**0.8099**	**0.0110**	**0.0053**
set-300-1	-26648.99	-27546.84	-27726.73	0.314	1.6186	0.1498	0.0581
set-300-2	-26784.41	-27694.58	-28164.46	0.317	1.6689	0.1480	0.0584
set-300-3	-26583.08	-27562.75	-28366.83	0.321	1.5849	0.1531	0.0565
set-300-4	-25978.75	-27522.38	-26545.89	0.301	1.7247	0.1458	0.0604
set-300-5	-26950.19	-27644.41	-28286.65	0.319	1.6279	0.1596	0.0561
set-300-6	-26328.74	-27496.23	-27619.18	0.313	1.7811	0.1530	0.0478
set-300-7	-27258.53	-27557.09	-28449.35	0.322	1.5814	0.1603	0.0554
set-300-8	-26762.08	-27592.03	-28034.09	0.317	1.6028	0.1560	0.0579
set-300-9	-26144.54	-27544.92	-27366.67	0.310	1.5655	0.1470	0.0525
set-300-10	-26427.97	-27574.07	-27609.74	0.313	1.6575	0.1489	0.0579
Average	**-26586.73**	**-27573.53**	**-27816.96**	**0.315**	**1.6413**	**0.1521**	**0.0561**
set-500-1	-58092.41	-58933.75	-60696.55	0.246	3.9778	2.6523	0.5989
set-500-2	-57888.24	-58881.13	-60912.73	0.247	3.8937	2.7815	0.6529
set-500-3	-58144.82	-58898.06	-60553.98	0.246	4.0764	2.8401	0.6181
set-500-4	-58536.37	-58949.24	-61837.03	0.251	4.0412	2.7539	0.6532
set-500-5	-58373.58	-58958.50	-60168.15	0.244	3.9782	2.8051	0.6122
set-500-6	-58479.47	-58921.05	-61173.10	0.248	3.9176	2.6918	0.6573
set-500-7	-57904.23	-58920.85	-59999.51	0.244	4.1334	2.6531	0.5990
set-500-8	-58626.07	-58900.66	-61717.97	0.250	3.7641	2.8160	0.6512
set-500-9	-58129.24	-58954.47	-60651.12	0.246	3.9914	2.9077	0.6122
set-500-10	-58645.33	-59135.28	-60954.82	0.246	3.9051	2.5907	0.6222
Average	**-58281.98**	**-58945.30**	**-60866.49**	**0.247**	**3.9679**	**2.7492**	**0.6277**
	Dual	SDRBR	NRBR	best	SDD	SDRBR	NRBR
set-1000-1	-340206.70	-866683.63	-355972.36	0.358	20.4300	23.7557	5.3681
set-1000-2	-853236.38	-858732.93	-893527.27	0.903	20.1970	23.4533	5.0803
set-1000-3	-849708.89	-859350.84	-885950.80	0.894	18.9416	23.5406	5.1446
set-1000-4	-849342.58	-858772.63	-891972.02	0.901	20.6648	23.5301	5.1317
set-1000-5	-850796.12	-858945.21	-888577.09	0.898	21.6445	23.4675	70.7289
set-1000-6	-849644.67	-858917.45	-875407.95	0.884	23.0902	23.4392	5.4389
set-1000-7	-850040.01	-858436.13	-884028.36	0.893	20.0064	23.3765	5.0313
Average	**-777567.91**	**-859976.97**	**-810776.55**	**0.819**	**20.7106**	**23.5090**	**14.5605**

The BQPLIB problems are formulated as minimization problems. As shown in Ta-

ble 3, the proposed Nuclear RBR method consistently achieves lower objective values than both SDcutRBR and SDD, demonstrating its superior performance on the BQPLIB benchmark instances.

The Beasley benchmark problems are also formulated as minimization problems. As presented in Table 4, the proposed Nuclear RBR method yields smaller objective values compared to SDcutRBR and SDD, further highlighting its advantage on the Beasley instances.

Table 4. Comparison among different algorithms on the objective value and the execution time for the minimization problems in Beasley dataset with $n = 50, 100, 250$, and 500

filename	obj Dual	obj SDRBR	obj NRBR	gamma best	time SDD	time SDRBR	time NRBR
bqp50-1	-2345.47	-3644.76	-3966.34	1.858	0.0340	1.3795	0.0064
bqp50-2	-3796.34	-4558.76	-5588.94	2.136	0.0185	1.1114	0.0027
bqp50-3	-4659.10	-4926.76	-6346.74	2.289	0.0209	1.0167	0.0037
bqp50-4	-3645.14	-4434.76	-5187.93	2.028	0.0201	1.5272	0.0069
bqp50-5	-4151.82	-4551.76	-5652.95	2.164	0.0212	1.7999	0.0133
bqp50-6	-3733.79	-4770.26	-5365.38	1.966	0.0183	1.6679	0.0067
bqp50-7	-4686.00	-4922.26	-6303.21	2.274	0.0165	1.6036	0.0031
bqp50-8	-4373.94	-4763.26	-5497.70	2.020	0.0310	1.6545	0.0039
bqp50-9	-4054.33	-4603.26	-5698.47	2.162	0.0240	1.8987	0.0058
bqp50-10	-3725.81	-4588.26	-5083.29	1.926	0.0253	2.2252	0.0075
Average	**-3917.17**	**-4576.41**	**-5469.09**	**2.082**	**0.0230**	**1.5885**	**0.0060**
bqp100-1	-8721.11	-12684.54	-12713.61	1.363	0.1034	2.6144	0.0446
bqp100-2	-11704.18	-14127.54	-14388.72	1.386	0.1241	2.3684	0.0123
bqp100-3	-13336.70	-15300.54	-17589.88	1.587	0.1123	1.9676	0.0173
bqp100-4	-10927.93	-13744.04	-14356.04	1.421	0.1013	2.2978	0.0155
bqp100-5	-9736.93	-13246.54	-13296.42	1.365	0.1238	1.4855	0.1195
bqp100-6	-11073.07	-13945.54	-14665.47	1.432	0.0844	1.0030	0.0052
bqp100-7	-10906.86	-13942.04	-14333.47	1.399	0.0309	0.9562	0.0052
bqp100-8	-12078.48	-13580.54	-14861.23	1.487	0.0382	0.8921	0.0080
bqp100-9	-11926.97	-13912.54	-14821.17	1.450	0.0357	0.8722	0.0065
bqp100-10	-13151.28	-14691.54	-15723.73	1.463	0.0252	0.9576	0.0053
Average	**-11356.35**	**-13917.54**	**-14674.97**	**1.435**	**0.0779**	**1.5415**	**0.0240**
bqp250-1	-48732.37	-50103.77	-57490.59	0.921	0.3380	1.7100	0.0390
bqp250-2	-48093.50	-51986.27	-60793.44	0.944	0.3440	1.7725	0.0646
bqp250-3	-51745.40	-54622.77	-62520.14	0.929	0.3880	1.8308	0.0443
bqp250-4	-44391.58	-48474.77	-55407.87	0.914	0.3454	1.8645	0.0491
bqp250-5	-50803.63	-51600.77	-59139.78	0.924	0.3500	1.8263	0.0503
bqp250-6	-44547.53	-49684.27	-58255.00	0.940	0.3419	1.9007	0.0509
bqp250-7	-49709.76	-52527.27	-60708.16	0.934	0.3467	1.8496	0.0527
bqp250-8	-40005.60	-47125.77	-55290.40	0.934	0.3499	1.8606	0.0413
bqp250-9	-52330.23	-53435.27	-62640.08	0.950	0.3840	1.7129	0.0454
bqp250-10	-44026.14	-50121.27	-57124.50	0.915	0.3762	1.8888	0.0329
Average	**-47438.57**	**-50968.22**	**-58937.00**	**0.930**	**0.3564**	**1.8217**	**0.0470**
bqp500-1	-128402.72	-138148.75	-162346.12	0.651	6.3214	5.7111	0.7204
bqp500-2	-138237.20	-141625.75	-163987.27	0.644	5.3068	4.2865	0.5627
bqp500-3	-140738.05	-144635.25	-170103.74	0.656	4.3811	4.5854	0.5852
bqp500-4	-141602.11	-148216.25	-170362.37	0.643	4.4439	4.2572	0.5934
bqp500-5	-136578.72	-140071.25	-167840.63	0.665	4.4257	4.3602	0.5675
bqp500-6	-132960.20	-139459.25	-162888.73	0.648	4.3120	4.3941	0.5703
bqp500-7	-134273.56	-138866.75	-165891.54	0.662	4.6998	4.5534	0.5901
bqp500-8	-135438.79	-146398.25	-172039.18	0.657	4.5905	4.8108	0.6553
bqp500-9	-132615.73	-141456.75	-164046.42	0.645	4.5932	4.8955	0.6743
bqp500-10	-141076.28	-144256.25	-168851.80	0.653	4.6826	4.7991	0.7644
Average	**-136192.34**	**-142313.45**	**-166835.78**	**0.652**	**4.7757**	**4.6653**	**0.6284**

Table 5 presents the average objective values obtained by the different methods, along with the corresponding CPU times. The numerical results clearly indicate that NuclearRBR outperforms the other two competitive methods, achieving near-optimal solutions while requiring less CPU time.

Table 5. **Comparison of averages of all datasets**

filename	obj Dual	obj SDRBR	obj NRBR	gamma best	time SDD	time SDRBR	time NRBR
bqpgub-20	918.05	1028.79	1111.95	1.811	0.7650	0.0045	0.7118
bqpgub-50	2702.87	3008.32	3272.19	0.935	0.6504	0.0055	0.5910
bqpgub-100	89029.09	92061.21	94461.83	7.090	0.7610	0.0191	0.7149
bqpgub-200	50800.26	53805.07	58662.84	1.117	1.3719	0.0703	1.0617
bqpgub-500	1379656.10	1395404.26	1430417.80	4.323	2.9434	2.4908	2.5995
cmisp-20	1090.00	1096.55	1126.03	1.766	0.6449	0.0067	0.6296
cmisp-50	1184.80	1261.04	1310.56	0.338	0.3802	0.0071	0.3093
cmisp-100	41558.40	43286.86	48612.37	3.739	0.9561	0.0190	0.8124
set30	-1329.71	-1630.49	-2126.23	2.301	0.7040	0.0035	0.0020
set50	-1632.24	-1724.36	-1656.53	0.688	0.7384	0.0028	0.0022
set100	-4812.55	-5375.62	-4973.79	0.512	0.8099	0.0110	0.0053
set300	-26586.73	-27573.53	-27816.96	0.315	1.6413	0.1521	0.0561
set500	-58281.98	-58945.30	-60866.49	0.247	3.9679	2.7492	0.6277
set1000	-777567.91	-859976.97	-810776.55	0.819	20.7106	23.5090	14.5605
bqp50	-3917.17	-4576.41	-5469.09	2.082	0.0230	1.5885	0.0060
bqp100	-11356.35	-13917.54	-14674.97	1.435	0.0779	1.5415	0.0240
bqp250	-47438.57	-50968.22	-58937.00	0.930	0.3564	1.8217	0.0470
bqp500	-136192.34	-142313.45	-166835.78	0.652	4.7757	4.6653	0.6284

6. Conclusion

This paper presents a novel UBQP relaxation for the BQP-GUB and develops a computationally efficient first-order algorithm for solving the associated semidefinite relaxation, with guaranteed global convergence. Numerical experiments conducted on a range of benchmark datasets indicate that the proposed *NuclearRBR* method achieves consistently better performance than the *SDcutRBR* and *SDD* methods across problems of varying sizes. The results also demonstrate the sensitivity of solution quality to the choice of the regularization parameter γ_{best}, suggesting opportunities for further investigation and refinement. Taken together, the theoretical analysis and computational results support the effectiveness of NuclearRBR for large-scale BQP-GUB problems arising in practical applications.

Acknowledgments This research work is fully supported by the National Board for Higher Mathematics (NBHM), Department of Atomic Energy, Govt. of India (Grant no. 02011/22/2021 NBHM (R.P.)/R&D II/900 Date 06/08/2021).

Conflict of Interest: The authors declare that they have no conflict of interest.

Availability of data The datasets analyzed during the current study are available on the web.
BQP-GUB:https://sites.google.com/a/iiit-bh.ac.in/r-k-nayak/bqp-bqp-problems/bqp-gub-problems.
CMISP:https://sites.google.com/a/iiit-bh.ac.in/r-k-nayak/bqp-bqp-problems/cmisp.
UBQP:https://sites.google.com/a/iiit-bh.ac.in/r-k-nayak/bqp-bqp-problems/ubqp.
Beasley dataset: http://people.brunel.ac.uk/ mastjjb/jeb/orlib/bqpinfo.html.

References

[1] E.A. Anacleto, C.N. Meneses, and R.N. Liang, *Fast r-flip move evaluations via closed-form formulae for boolean quadratic programming problems with generalized upper bound constraints*, Comput. Oper. Res. **132** (2021), 105297. https://doi.org/10.1016/j.cor.2021.105297.

[2] M. ApS, *Mosek optimization toolbox for matlab*, User's Guide and Reference Manual, Version **4** (2019), no. 1, p.116.

[3] J.E. Beasley, *Heuristic algorithms for the unconstrained binary quadratic programming problem*, Tech. report, Working Paper, The Management School, Imperial College, London, England, 1998.

[4] P. Chrétienne, *A polynomial algorithm to optimally schedule tasks on a virtual distributed system under tree-like precedence constraints*, European J. Ope. Res. **43** (1989), no. 2, 225–230. https://doi.org/10.1016/0377-2217(89)90216-6.

[5] E.D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica, *Correlation clustering in general weighted graphs*, Theoretical Computer Science **361** (2006), no. 2-3, 172–187. https://doi.org/10.1016/j.tcs.2006.05.008.

[6] S. Dhouib, *A new column-row method for traveling salesman problem: the dhouib-matrix-TSP1*, International Journal of Recent Engineering Science **8** (2021), no. 1, 6–10. https://doi.org/10.14445/23497157/IJRES-V8I1P102.

[7] E. Gelenbe, S. Timotheou, and D. Nicholson, *Fast distributed near-optimum assignment of assets to tasks*, Comput. J. **53** (2010), no. 9, 1360–1369. https://doi.org/10.1093/comjnl/bxq010.

[8] F. Glover, Z. Lü, and J.K. Hao, *Diversification-driven tabu search for unconstrained binary quadratic problems*, 4OR **8** (2010), no. 3, 239–253. https://doi.org/10.1007/s10288-009-0115-y.

[9] M. Grant and S. Boyd, *Cvx: Matlab software for disciplined convex programming, version 2.1*, 2014.

[10] M. Grant, S. Boyd, and Y. Ye, *Cvx: Matlab software for disciplined convex programming*, 2008.

[11] S. Guattery and G.L. Miller, *On the quality of spectral separators*, SIAM J. Matrix Anal. Appl. **19** (1998), no. 3, 701–719. https://doi.org/10.1137/S0895479896312262.

[12] M. Heiler, J. Keuchel, and C. Schnörr, *Semidefinite clustering for image segmentation with a-priori knowledge*, Joint Pattern Recognition Symposium, Springer, 2005, pp. 309–317.

[13] A. Joulin, F. Bach, and J. Ponce, *Discriminative clustering for image co-segmentation*, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 1943–1950. https://doi.ieeecomputersociety.org/10.1109/CVPR.2010.5539868.

[14] R. Kannan, S. Vempala, and A. Vetta, *On clusterings: Good, bad and spectral*, Journal of the ACM (JACM) **51** (2004), no. 3, 497–515. https://doi.org/10.1145/990308.990313.

[15] J. Kleinberg and E. Tardos, *Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields*, Journal of the ACM (JACM) **49** (2002), no. 5, 616–639. https://doi.org/10.1145/585265.585268.

[16] K. Lang, *Fixing two weaknesses of the spectral method*, Advances in Neural Information Processing Systems (Vancouver, British Columbia, Canada).

[17] F. Lauer and C. Schnörr, *Spectral clustering of linear subspaces for motion segmentation*, 2009 IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 678–685. https://doi.org/10.1109/ICCV.2009.5459173.

[18] Z. Lü, F. Glover, and J.K. Hao, *A hybrid metaheuristic approach to solving the UBQP problem*, European J. Oper. Res. **207** (2010), no. 3, 1254–1262. https://doi.org/10.1016/j.ejor.2010.06.039.

[19] Z. Lü, J.K. Hao, and F. Glover, *A study of memetic search with multi-parent combination for UBQP*, Evolutionary Computation in Combinatorial Optimization: 10th European Conference, EvoCOP 2010, Istanbul, Turkey, April 7-9, 2010. Proceedings 10 (Istanbul, Turkey).

[20] V.S. Mai, D. Maity, B. Ramasubramanian, and M.C. Rotkowitz, *Convex methods for rank-constrained optimization problems*, 2015 Proceedings of the Conference on Control and its Applications, SIAM, 2015, pp. 123–130. https://doi.org/10.1137/1.9781611974072.18.

[21] R.K. Nayak and M.P. Biswal, *A low complexity semidefinite relaxation for large-scale MIMO detection*, J. Comb. Optim. **35** (2018), no. 2, 473–492. https://doi.org/10.1007/s10878–017–0186–1.

[22] R.K. Nayak and N.K. Mohanty, *Improved row-by-row method for binary quadratic optimization problems*, Ann. Oper. Res. **275** (2019), no. 2, 587–605. https://doi.org/10.1007/s10479-018-2978-9.

[23] C. Schellewald and C. Schnörr, *Probabilistic subgraph matching based on convex relaxation*, International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (Mannheim, Germany).

[24] Z. Shang, J.K. Hao, S. Zhao, Y. Wang, and F. Ma, *Multi-wave tabu search for the boolean quadratic programming problem with generalized upper bound constraints*, Comput. Oper. Res. **150** (2023), 106077. https://doi.org/10.1016/j.cor.2022.106077.

[25] J. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Transactions on pattern analysis and machine intelligence **22** (2000), no. 8, 888–905. https://doi.org/10.1109/34.868688.

[26] S.N. Vitaladevuni and R. Basri, *Co-clustering of image segments using convex optimization applied to EM neuronal reconstruction*, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 2203–2210.

https://doi.org/10.1109/CVPR.2010.5539901.

[27] H.T. Wai, W.K. Ma, and A.M.C. So, *Cheap semidefinite relaxation MIMO detection using row-by-row block coordinate descent*, 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (Prague, Czech Republic).

[28] Y. Wang, Z. Lü, F. Glover, and J.K. Hao, *Path relinking for unconstrained binary quadratic programming*, European J. Oper. Res. **223** (2012), no. 3, 595–604. https://doi.org/10.1016/j.ejor.2012.07.012.

[29] Y. Wang, Z. Lü, F. Glover, and J.K. Hao, *Backbone guided tabu search for solving the UBQP problem*, J. Heuristics **19** (2013), no. 4, 679–695. https://doi.org/10.1007/s10732-011-9164-4.

[30] Y. Wang and A.P. Punnen, *The boolean quadratic programming problem with generalized upper bound constraints*, Comput. Oper. Res. **77** (2017), 1–10. https://doi.org/10.1016/j.cor.2016.07.005.

[31] Z. Wen, D. Goldfarb, S. Ma, and K. Scheinberg, *Row by row methods for semidefinite programming*, Industrial Engineering (2009), 1–21.

[32] S.X. Yu and J. Shi, *Segmentation given partial grouping constraints*, IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004), no. 2, 173–183. https://doi.org/10.1109/TPAMI.2004.1262179.

[33] F. Zhang, *The schur complement and its applications*, vol. 4, Springer Science and Business Media, 2006.

[34] Y. Zhou, J. Wang, Z. Wu, and K. Wu, *A multi-objective tabu search algorithm based on decomposition for multi-objective unconstrained binary quadratic programming problem*, Knowledge-Based Systems **141** (2018), 18–30 https://doi.org/10.1016/j.knosys.2017.11.009.