Selecting cutting planes for quadratic semidefinite outer-approximation via trained neural networks

Radu Baltean-Lugojan · Pierre Bonami · Ruth Misener · Andrea Tramontani

November 26, 2018

Abstract Semidefinite programming relaxations complement polyhedral relaxations for quadratic optimization, but global optimization solvers built on polyhedral relaxations cannot fully exploit this advantage. This paper develops linear outer-approximations of semidefinite constraints that can be effectively integrated into global solvers. The difference from previous work is that our proposed cuts are (i) sparser with respect to the number of nonzeros in the row and (ii) explicitly selected to improve the objective. We select which cuts to generate using objective structure and explore engineering trade-offs for sparsity patterns, e.g. cut dimensionality and chordal extensions. A neural network estimator is key to our cut selection strategy: ranking each cut based on objective improvement involves solving a semidefinite optimization problem, but this is an expensive proposition at each Branch&Cut node. The neural network estimator, trained a priori of any instance to solve, takes the computation offline by predicting the objective improvement for any cut. Most of this paper discusses quadratic programming, but we show our ideas also extend to quadratic constraints.

Keywords mixed-integer nonlinear programming · quadratic programming · cutting planes · convex and semidefinite relaxations · neural networks · cut selection

Mathematics Subject Classification (2000) 90C26 · 90C57 · 90C20 · 90C22

1 Introduction

Nonconvex quadratic programming is fundamental to optimization and pervasive in diverse applications [72, 96], but solving such NP-hard [99] problems at large scale to global optimality is challenging. From an MINLP perspective [21], mixed-integer quadratic instances, e.g. QPLib

This work was funded by an Engineering & Physical Sciences Research Council (EPSRC) DTP (award ref. 1675949) and an IBM PhD Fellowship to RBL and EPSRC Research Fellowship to RM (grant number EP/P016871/1). This manuscript significantly benefited from discussions at Dagstuhl Seminar 18081.

{radu.baltean-lugojan09, r.misener}@imperial.ac.uk
Department of Computing, Imperial College London, 180 Queen's Gate, London, SW7 2AZ, UK
pierre.bonami@es.ibm.com
CPLEX Optimization, IBM Spain, Madrid, Spain
andrea.tramontani@it.ibm.com
CPLEX Optimization, IBM Italy, Bologna, Italy

[44], are typically solved using finite Branch&Cut frameworks mixing different relaxation classes. State-of-the-art solvers tackle continuous nonconvexity through computationally light relaxations (see Section 2 for details) that fuse well with the powerful and expansive cutting plane machinery for mixed-integer programming (MIP). A longstanding research effort integrates strong continuous quadratic relaxations into Branch&Cut, e.g. copositive [30] or semidefinite [11, 39]. These cutting surfaces include: (i) directly semidefinite [25, 27, 45], (ii) convex outer-approximations [38, 41, 90, 91], and (iii) linear outer-approximations [85, 95]. But the trade-off between bound strength and computational tractability in these convex nonlinear relaxations is not trivial. Copositive and semidefinite relaxations typically offer strong bounds for a heavier computational cost, although binary quadratic programs are a notable exception [16, 24, 60, 67, 86, 106].

Our paradigm, driven by our goal of effectively integrating semidefinite cutting surfaces into state-of-the-art global solvers, incorporates the following desirata:

1. **Easy integration into current technology**, i.e. the cutting surface should integrate well into state-of-the-art MINLP Branch&Cut technology and complement existing relaxations.
2. **Computationally light relaxations**, i.e. neither the cutting surface generation nor its incorporation into the Branch&Cut node should dominate solver execution time. An expensive approximation may be useful for a specific application, but general-purpose solvers cannot invest too much time in possibly unproductive cutting surfaces.
3. **Relatively few cuts** should approximate the relaxation, i.e. the *computationally light relaxations* should not become computationally heavy via an excessive number of cuts.
4. **Cutting surface generation should converge** to the desired relaxation, i.e. the relaxation should not miss critical parts of the semidefinite relaxation.

These desirata are familiar algorithmic considerations in [34], but we require a significantly augmented framework for quadratic optimization. In the spirit of efficient second-order cone programming (SOCP) polyhedral relaxations [100], we chose cheap, linear outer-approximations of semidefinite relaxations that mix well with the cutting planes in solver software [17, 18, 75]. MINLP solvers use extended formulations [13, 55, 73, 74, 87, 97, 101], i.e. replace nonlinear term $x_i x_j$ with auxiliary variable X_{ij}, which we also adopt to integrate into current technology. Moreover, such extended formulations also allow us to fulfill the desirata of using computationally light relaxations via low-dimensional or sparse cutting planes. Using sparse cuts drives MIP cut selection [34, 35, 104] and sparse cuts in extended formulations dominate cuts in the original polyhedron [35].

Developments in the semidefinite programming literature balance the conflicting desirata of (3) using relatively few cuts versus (4) converging to the desired relaxation. This paper decomposes the semidefinite relaxation [5] into low-dimensional positive semidefinite (PSD) cones. These low-dimensional cones are inspired by the d-exponential complexity of approximating d-dimensional convex bodies via polytopes [40, 50], empirically confirmed in PSD cone linearizations [85, 95]. Low-dimensional decomposition enables relatively few cuts, but we also want the relaxation to converge, e.g. that a possibly greedy method does not miss important cuts. The sparse semidefinite literature [42, 43, 79] motivates cut selection with variables in chordal extensions for sparse/near-chordal instances and offers convergence guarantees.

MIP cut selection [34], i.e. avoiding to add all generated cuts, is an active research area. Cut selection leads to speedups [4] and enables early branching [20, 63]. Prior work explores difficult-to-find [83] cut selection measures, e.g. [22, 34, 105]. These contributions motivate selecting *strong* rather than all *valid* cuts. We expect cuts maximizing objective improvement [32] may be stronger than those cutting random parts of feasible space. Most existing techniques focus on *feasibility cuts* selected based on feasibility violation [22]. These feasibility techniques attempt heuristically to improve the objective via parallelism to it or cut orthogonality [4] while

ignoring explicit structure. But our goal of developing relatively few cuts motivates directly our using the objective structure for *optimality(-based) selection* of localized *deep objective cuts* [22]. For quadratically-constrained quadratic optimization (QCQP), we use not only objective but also constraint structure, see Section 6. This paper takes cut selection complexity offline from the solver process via a trained (a priori and instance independent) neural network estimator, answering a hint to enlist machine learning in strong cut selection [34].

This paper focuses on outer-approximating the semidefinite relaxation by selecting strong, sparse, and linear cutting planes based on objective improvement (or optimality) via trained neural networks. The rest of the paper proceeds as follows: Section 2 introduces the background/notation for semidefinite (SDP) and other linear (base) relaxations the outer-approximation scheme relies on; Section 3 gives the theoretical framework for decomposing SDP relaxations and extracting optimality-based sparse cutting planes; Section 4 implements the neural network estimators to make optimality cut selection tractable, discussing their training and data sampling; Section 5 compares implemented cutting plane strategies on BoxQP instances, confirming in practice that: (i) neural networks are effective estimators, (ii) optimality cut selection identifies the strongest but not all valid cuts, and (iii) instance sparsity structure and cut sparsity influence cut selection strategies; Section 6 extends the selection of strong optimality cuts to QCQP and Section 7 concludes with further extensions. Source code reproducing our results is on Github [9].

2 Nonconvex quadratic formulations and relaxations

For the discussion in Sections 3 and 5, we first consider nonconvex quadratic problems that are box and linearly constrained, i.e.

$$z_{qp} = \min_{x} \left\{ x^T Q x + c^T x \mid Ax \leq b,\ x \in [0,1]^N \right\}, \tag{QP}$$

with an N-variable vector x, $A \in \mathbb{R}^{p \times N}$ and $Q \in \mathbb{R}^{N \times N}$ assumed to be an indefinite matrix. This class of problems encompasses many relevant instances, since the box constraints can be obtained by simple variable scaling from any closed and bounded domains [18, Appendix A.1]. We lift each quadratic term $x_i x_j$ by replacing it with a new variable X_{ij}. Let lifted variables $X_{ij}\ \forall i, j$ form matrix $X = xx^T$ and let $Q \bullet X = Tr(Q^T X) = \sum_{i,j} Q_{ij} X_{ij}$, representing the Frobenius inner product (applied to pairs of either matrices or vectors with the same dimensions). Then z_{qp} is lower-bounded by

$$z_{qp}(\mathcal{B}) := \min_{x,X} \left\{ Q \bullet X + c^T x \;\middle|\; Ax \leq b,\ x \in [0,1]^N \text{ and } (x, X) \in \mathcal{B} \right\},$$

parametric on any convex set $\mathcal{B}$ that adds valid constraints to the basic lifted formulation of QP.

Let $G(V, E)$ denote the sparsity pattern graph induced by matrix Q (linking lifted X variables) where vertex set V and edge set E are defined as

$$V = \{1, 2, \ldots, N\}, \qquad E = \{\{i, j\} \in V \times V \mid i > j,\ Q_{ij} \neq 0\},$$

and let $G(V, \overline{E})$ $(E \subseteq \overline{E})$ be a chordal extension of $G(V, E)$.

Relaxing nonconvex $X = xx^T$ to $X \succeq xx^T$, or equivalently (by Schur's complement) to $\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0$, results in the well-known $\mathcal{S}$ semidefinite relaxation to QP. Furthermore, as [5] argues,

the extra constraints $X_{ii} \leq x_i \ \forall i \in V$ (a subset of the reformulation linearization technique or RLT relaxations) are needed to bound the convex set $\mathcal{S}$,

$$\mathcal{S} := \left\{ (x, X) \middle| \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0, \ X_{ii} \leq x_i \ \forall i \in V \right\}.$$

The cutting plane approach in this paper relies on developing cuts in the variable space of extended formulations (both x and X), in order to be directly pluggable into extended linear relaxations employed at each node by Branch&Cut solvers such as Antigone [73, 74], Baron [87, 97], Couenne [13], CPLEX [55] or SCIP [1, 101]. This direction allows easy integration with other routinely used classes of cuts in the extended formulation space such as (i) the reformulation linearization technique (RLT) or the McCormick [69] hull [93], (ii) zero-half or triangle inequalities [18], and (iii) edge-concave cuts [10, 75]. To illustrate this potential, motivated by the findings in Anstreicher [5], in our implementation we augment any semidefinite relaxation such as $\mathcal{S}$ by the McCormick bounds $\mathcal{M}$, given in our case by

$$\mathcal{M} := \left\{ (x, X) \middle| \begin{array}{c} \forall i, j \in V \\ \{i, j\} \in E \end{array} : \begin{array}{r} X_{ij} \geq l_i x_j + l_j x_i - l_i l_j = 0, \\ X_{ij} \geq u_i x_j + u_j x_i - u_i u_j = x_i + x_j - 1, \\ X_{ij} \leq l_i x_j + u_j x_i - l_i u_j = x_i, \\ X_{ij} \leq u_i x_j + l_j x_i - u_i l_j = x_j. \end{array} \right\}.$$

A projection of the generated semidefinite cuts together with the McCormick bounds as in [38, 91] can lead to an equivalent projected formulation. However, we choose the intuitive results and easy integration of an extended formulation with any classes of cuts over any potential speedup a projection might bring in this case.

The cutting plane approach can be strengthened with any other class of valid extended inequalities to obtain tighter bounds, and based on [18], we also append in our experiments from Section 5 the triangle inequalities

$$\triangle := \left\{ (x, X) \middle| \ \forall i, j, k \in V : \begin{array}{r} X_{ij} + X_{jk} + X_{ki} - x_i - x_j - x_k \geq -1 \\ -X_{ij} + X_{jk} - X_{ki} + x_i \geq 0, \\ -X_{ij} - X_{jk} + X_{ki} + x_j \geq 0, \\ X_{ij} - X_{jk} - X_{ki} + x_k \geq 0 \end{array} \right\}.$$

Practically, due to their $O(N^3)$ number, $\triangle$ inequalities are not all added at the beginning as the $\mathcal{M}$ bounds, but instead are selected/separated by an ordering based on violation. Also, to avoid separating too many weak $\triangle$ inequalities in sparse QP instances, we heuristically first separate violated (i, j, k) triangle inequalities with all relevant objective coefficients $Q_{ij}, \ Q_{ik}, \ Q_{jk} \neq 0$, followed by violated (i, j, k) triangle inequalities with one zero among relevant objective coefficients. The described separation is intended to be similar to that in [18], in order to show in Section 5.4 that even a strong relaxation can be augmented by the semidefinite cuts the paper builds with negligible computational overhead.

For the discussion in Section 6 we introduce quadratically constrained programs as an extension of QP, i.e.

$$z_{qc} = \min_x \left\{ x^T Q^{(0)} x + c_0^T x \middle| \begin{array}{l} x^T Q^{(k)} x + c_k^T x \leq 0 \quad k = 1, \ldots, m \\ Ax \leq b, \ x \in [0, 1]^N \cap \mathcal{K}^N \end{array} \right\}, \tag{QCQP}$$

with an N-variable vector x, $A \in \mathbb{R}^{p \times N}$, $\mathcal{K}^N$ the intersection of second order cone constraints (SOCP) and $Q^{(k)} \in \mathbb{R}^{N \times N} \ \forall k = 0, \ldots, m$ assumed to be indefinite matrices. This class of problems encompasses any instances bounded by closed domains by the same reasoning as in the case

of QP. Furthermore, QCQP includes instances with a convex objective or quadratic constraints, which can be included in $\mathcal{K}^N$ by reformulation as second-order cone constraints. This is done to leverage the warm-starting possibilities of second-order cone programs, allowing effective iterative cutting plane algorithms. The extended space lifting to X variables and subsequent relaxations $\mathcal{S}, \mathcal{M}, \triangle$ from QP apply to QCQP as well, with E extended to

$$E = \{\{i,j\} \in V \times V \mid i > j,\ \exists k \in 0,\dots,m \text{ s.t. } Q_{ij}^{(k)} \neq 0\}.$$

Lastly, we introduce a class of second-order cone constraints (SOCP) that is used to augment the base $\mathcal{M}$ relaxation for the optimal power flow instance Section 6 studies, i.e.

$$\mathcal{M}^2 := \mathcal{M} \cap \left\{ (x,X) \,\middle|\, \forall i \in V: \quad X_{ii} \geq x_i^2 \ \Leftrightarrow\ \left\| \begin{matrix} 1 - X_{ii} \\ 2x_i \end{matrix} \right\|_2 \leq 1 + X_{ii} \right\}.$$

3 Low-dimensional cutting planes from the semidefinite relaxation

This section develops effective outer-approximations for the quadratic semidefinite relaxation, retaining (most of) its tightness in the form of sparse (low-dimensional) linear cuts well-suited to Branch&Cut. Section 3.1 introduces low-dimensional semidefinite decompositions, Section 3.2 develops the theoretical background for outer-approximation via matrix completion and optimality-based cutting planes, while Section 3.3 describes alternative cut selection strategies and introduces an estimator for optimality-based selection that is then developed in Section 4.

3.1 Low-dimensional semidefinite relaxations via (sparse) decomposition

Let $\mathcal{P}$ be the power set of the vertex/index set V and let $\rho \in \mathcal{P}$ ($\rho \subseteq V$) be any arbitrary index subset. Denote by $x_\rho \in \mathbb{R}^{|\rho|}$ the vector slice of x and by $X_\rho \in \mathbb{R}^{|\rho| \times |\rho|}$ the submatrix slice of X respectively, both introduced to select variable aggregations indexed by ρ. For any subset of $\mathcal{P}$, introduce the corresponding semidefinite relaxation,

$$(\forall \mathcal{F} \subseteq \mathcal{P}) \quad \mathcal{S}(\mathcal{F}) := \left\{ (x,X) \,\middle|\, \forall \rho \in \mathcal{F}: \ \begin{bmatrix} 1 & x_\rho^T \\ x_\rho & X_\rho \end{bmatrix} \succeq 0,\ X_{ii} \leq x_i \ \forall i \in \rho \right\},$$

consisting of semidefinite constraints of submatrices sliced on all $\mathcal{F}$ elements, where $\mathcal{S}(\mathcal{P}) = \mathcal{S}$. Further denote the restrictions of the above relaxations to the fixed vector of variables $x = \tilde{x}$ as:

$$(\forall \mathcal{F} \subseteq \mathcal{P}) \quad \mathcal{S}^{\tilde{x}}(\mathcal{F}) := \mathcal{S}(\mathcal{F}) \cap \{(x,X) \mid x = \tilde{x}\}.$$

Given that $|\mathcal{P}| = 2^N$ implies an exponential number of semidefinite constraints in $\mathcal{S}(\mathcal{P})$, consider restricting/imposing a chosen cardinality n ($1 \leq n \leq N$) on elements of $\mathcal{P}$ by denoting

$$\mathcal{P}_n := \{\rho \in \mathcal{P} \mid |\rho| = n\}\ , \text{ with } |\mathcal{P}_n| = \binom{N}{n}.$$

Remark 3.1.1 *For any $\rho \subseteq V$, $|\rho| = n$, we refer to $(n+1)$-dimensional constraint $\begin{bmatrix} 1 & x_\rho^T \\ x_\rho & X_\rho \end{bmatrix} \succeq 0$ as being n-dimensional to reflect the size of set ρ.* □

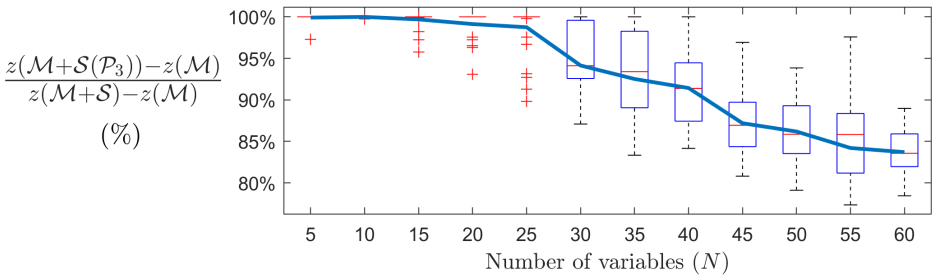


Fig. 1: The blue line shows the average percentage of gap (in blue) between $z(\mathcal{M})$ and $z(\mathcal{M}+\mathcal{S})$ closed by $z(\mathcal{M}+\mathcal{S}(\mathcal{P}_3))$ on 30 random QP instances for each number of variables N (iterates of 5). In terms of gap values for given N, the box captures the 25th-75th percentiles, the central red line indicates the median, the whiskers capture non-outliers outside the box and the '+' marks indicate outliers. For each QP random instance, c was assumed 0, $Ax \leq b$ was dropped, and random Q was sampled as explained in Table 1. Each instance sampled is dense, i.e. $\mathcal{S}(\mathcal{P}_3) = \mathcal{S}(\overline{\mathcal{P}_3}) = \mathcal{S}_3$ (see Lemma 3.1.5).

Therefore, for any given n, $\mathcal{S}(\mathcal{P}_n)$ decomposes the N-dimensional semidefinite constraint in $\mathcal{S}$ into a polynomial $\binom{N}{n}$ number of n-dimensional semidefinite constraints. Moreover, $\mathcal{S}(\mathcal{P}_n)$ follows the relaxation hierarchy in Lemma 3.1.2.

Lemma 3.1.2 *(Hierarchy of decomposed semidefinite relaxations)*
$\mathcal{S}(\mathcal{P}_1) \supseteq \mathcal{S}(\mathcal{P}_2) \supseteq \cdots \supseteq \mathcal{S}(\mathcal{P}_n) \supseteq \cdots \supseteq \mathcal{S}(\mathcal{P}_N) = \mathcal{S}$.

Proof Each inclusion for a fixed n $(1 \leq n \leq N-1)$ follows as

$$\mathcal{S}(\mathcal{P}_n) = \bigcap_{\rho \in \mathcal{P}_n} \mathcal{S}(\{\rho\}) \quad \supseteq \quad \bigcap_{\substack{\rho' \in \mathcal{P}_{n+1}, \\ \exists \rho \in \mathcal{P}_n \text{ s.t. } \rho \subset \rho'}} \mathcal{S}(\{\rho'\}) \quad = \mathcal{S}(\mathcal{P}_{n+1}),$$

since a positive semidefinite matrix implies any of its principal minors are positive semidefinite. □

Remark 3.1.3 *(Relaxation hierarchy - choices of n for tractability and strong bounds)*
• The choice of n in Lemma 3.1.2 directly impacts tractability. Due to the symmetry rule, the binomial coefficient $\binom{N}{n}$ is a concave function of $n \in 1, \ldots, N-1$ and lowest at extremes, i.e. for $n \in \{1, N-1\}$. Small n values are interesting because they imply more tractable $\mathcal{S}(\mathcal{P}_n)$ with respect to (i) number of constraints and (ii) constraint dimensionality.
• Even for small n values, $\mathcal{S}(\mathcal{P}_n)$ can lead to strong bounds. For example, consider $n = 3$ with fully dense QP instances of size $N \leq 60$ and no linear constraints $Ax \leq b$. Starting with a base McCormick $\mathcal{M}$ relaxation, Fig. 1 shows that $\mathcal{M} + \mathcal{S}(\mathcal{P}_3)$ closes a significant proportion of the gap between $\mathcal{M}$ and $\mathcal{M} + \mathcal{S}$. □

Furthermore, the $S(\mathcal{P}_n)$ $\forall n$ $(1 \leq n \leq N-1)$ decomposition can directly exploit any sparsity pattern in QP instances. Chordal sparsity, which is exploited by sparse semidefinite solvers such as SDPA-C [42, 43, 79], can be used to make the $\mathcal{S}(\mathcal{P}_n)$ decomposition more lightweight and effective via Lemma 3.1.4 or Lemma 3.1.5.

Lemma 3.1.4 *(Chordal extensions and their role in decomposition)*
For any $n \in 1, \ldots, N$, given a chordal extension $G(V, \overline{E})$ of sparsity pattern $G(V, E)$ of Q (Section 2) with maximal cliques $C_1, \ldots, C_k$, defining $\overline{\mathcal{P}_n} := \{\rho \in \mathcal{P}_n \mid \exists i \in 1, \ldots, k \text{ s.t. } \rho \subseteq C_i\} \subseteq \mathcal{P}_n$, then

$$\mathcal{S}(\mathcal{P}_n) = \mathcal{S}(\overline{\mathcal{P}_n}) \text{ and (clique number) } \max_{1 \leq i \leq k} |C_i| \leq n \Rightarrow \mathcal{S}(\overline{\mathcal{P}_n}) = \mathcal{S}.$$

Proof The result in [43, th. 2.5] based on [49, th. 7] implies that given the chordal extension $G(V, \overline{E})$,

$$X \succeq 0 \quad \Leftrightarrow \quad X_{C_i} \succeq 0 \quad \forall i \in 1, \dots, k.$$

Every of the variables x in the QP original space appears in $\mathcal{S}$ because $\forall i \in V$, $X_{ii} \leq x_i$ is required to bound $\mathcal{S}$. To incorporate all x variables into the sparsity pattern, create an extra vertex v and let $V' = V \cup \{v\}$ and $\overline{E}' = \overline{E} \cup \{(v, i)\ \forall i \in V\}$. Then $G(V', \overline{E}')$ is by definition chordal with maximal cliques $C_i \cup \{v\}\ \forall i \in 1, \dots, k$ and:

$$\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0 \quad \Leftrightarrow \quad \begin{bmatrix} 1 & x_{C_i}^T \\ x_{C_i} & X_{C_i} \end{bmatrix} \succeq 0 \quad \forall i \in 1, \dots, k.$$

Consequently, eliminating any $\rho \in \mathcal{P}_n,\ \rho \nsubseteq C_i$ from $\overline{\mathcal{P}_n}$ results in $\mathcal{S}(\mathcal{P}_n) = \mathcal{S}(\overline{\mathcal{P}_n})$. For a given n, the above equivalence also implies that, if all cliques are contained in $\mathcal{P}_n$, i.e. $n \geq \max_{1 \leq i \leq k} |C_i|$, then the decomposed relaxation is equivalent to the original, i.e. $\mathcal{S}(\overline{\mathcal{P}_n}) = \mathcal{S}$. □

Computing $\overline{\mathcal{P}_n}$ in Lemma 3.1.4 depends on constructing $G(V, \overline{E})$, practically possible via heuristic sparse matrix ordering packages (as in SDPA-C) such as Chompack [3]. Since $E \subseteq \overline{E}$, unless $G(V, E)$ is chordal, the relaxation $\mathcal{S}(\overline{\mathcal{P}_n})$ uses lifted variables X_{ij} with zero coefficients, i.e. $Q_{ij} = 0$. Intuitively, the farther $G(V, E)$ is from chordal, the more lifted variables with zero coefficients are used in $\mathcal{S}(\overline{\mathcal{P}_n})$, balancing the relaxation difficulty versus the convergence of cutting plane outer-approximation. Section 5 removes assumptions on instance chordal structure that involves using lifted variables with zero coefficients, except for Section 5.3.2 which discusses the impact of using chordal extensions. Lemma 3.1.5 sparsifies/relaxes $\mathcal{S}(\overline{\mathcal{P}_n})$ further to avoid lifted variables with zero coefficients - the results in Section 5.3.2 validate this choice.

Lemma 3.1.5 *(Dense restriction of decomposed semidefinite relaxations)*
Let $\mathcal{C}$ be the set of all cliques (including non-maximal ones) of sparsity pattern $G(V, E)$ of Q (Section 2). For any $n \in 1, \dots, N$, define the restricted relaxation $\mathcal{S}_n$ as

$$\mathcal{S}_n := \mathcal{S}(\mathcal{P}'_n), \text{ with } \mathcal{P}'_n := \{\rho \in \mathcal{C} \mid |\rho| \leq n,\ \nexists \rho_2 \in \mathcal{C} \text{ s.t. } |\rho_2| \leq n,\ \rho \subset \rho_2\}.$$

Then $\mathcal{S}_n \supseteq \mathcal{S}(\mathcal{P}_n)$, and $\mathcal{S}_n = \mathcal{S}(\mathcal{P}_n)$ when $G(V, E)$ is dense, i.e. $E = V \times V$.

Proof Since by construction $\mathcal{P}'_n \subseteq \bigcup_{i \in 1 \dots n} \mathcal{P}_i$, Lemma 3.1.2 implies $\mathcal{S}_n \supseteq \bigcap_{i \in 1 \dots n} \mathcal{S}(\mathcal{P}_i) = \mathcal{S}(\mathcal{P}_n)$. Furthermore, if $E = V \times V$, then $\mathcal{P}'_n = \mathcal{P}_n$ and thus $\mathcal{S}_n = \mathcal{S}(\mathcal{P}_n)$.

□

Corollary 3.1.6 *(Hierarchy of restricted decomposed semidefinite relaxations)*
$\mathcal{S}_1 \supseteq \mathcal{S}_2 \supseteq \cdots \supseteq \mathcal{S}_n \supseteq \cdots \supseteq \mathcal{S}_N = \mathcal{S}$. (follows directly from Lemmas 3.1.2 and 3.1.5). □

Remark 3.1.7 *($\mathcal{S}_n$ - an extension of SOCP Type 2 and lighter SDP outer-approx. in Branch&Cut)*
• Considering Corollary 3.1.6, for increasing $n \geq 2$, by design the relaxations $\{\mathcal{S}_n\}$ dominate and progressively extend an equivalent SOCP Type 2 relaxation [58, 59] which contains SOCP constraints for variable matrices of dimension 2.
• Other semidefinite outer-approximations consider high-dimensional convex cutting surfaces in all (x, X) variables [38, 90, 91]. Prior work separates linear outer-approximations using a high-dimension [85] or medium-dimension [94, 95] feasibility approach. Convex nonlinear cutting surfaces are less suitable to a Branch&Cut framework than the established (mixed-integer) linear relaxations which combine advanced cut classes, heuristics, preprocessing and easy warm-starting for reduced solution times at the tree nodes. But high-dimensional linear cuts lead to a computationally heavy LP [85]. Since $\binom{N}{n}$ is concave in n, separating medium-dimensional linear

cuts incurs the most separation sub-problems. This paper proposes a low-dimensional approach leading to lighter linear relaxations suitable for Branch&Cut. These strong cuts can be separated first via optimality-based selection (see Section 5). □

Remark 3.1.8 *($\mathcal{S}_n$ - a complementary relaxation to multiterm edge-concave relaxations)* The table below summarizes the main differences between outer-approximating the $\mathcal{S}_n$ relaxation and multiterm edge-concave relaxations [10, 18, 71, 75]:

	Low-dimensional cuts from $\mathcal{S}_n$	*Multiterm polyhedral relaxations*
Approach	Decomposition/relaxation of SDP constraints to select/ separate low-dim. cuts iteratively	Multi-term aggregation to explicitly identify dominant facets/ cuts
Variable groups & bound strength	All possible aggregations (including mixtures of convex and vertex-polyhedral terms), which can improve bounds overall	Only vertex-polyhedral/ multilinear aggregations where the convex envelope is polyhedral
Cuts generated	Outer-approximating cuts selected/separated (before generation via optimality selection) at current pt.	Explicit facets, either all added or separated at current point

Multiterm relaxations offer the convex envelope for specific variable aggregations. But vertex-polyhedral cut separation scales exponentially as dimensionality increases, so these cuts are most common for $n \in \{3, 4, 5\}$. Outer-approximating $\mathcal{S}_n$ separates cuts for all variable aggregations of a given size including convex terms and zero-coefficient terms, i.e. more general aggregations than multiterm edge-concave relaxations. Moreover, strong cuts from $\mathcal{S}_n$ can be separated first via optimality selection (see Section 5). A hybrid complementary approach can therefore use multiterm edge-concave relaxations for relevant aggregations and tighten the bound further via strong cuts selected from $\mathcal{S}_n$ outer-approximations. Another strong synergistic approach is to separate $\mathcal{S}_n$ sub-problems based on optimality and then add a mix of vertex-polyhedral and/or eigenvalue cuts for each sub-problem, having both strong cut separation/selection and strong generated cuts. □

Remark 3.1.9 *($\mathcal{S}_n$ - solving with interior point vs. outer-approx. linearization)* The number of semidefinite constraints scales polynomially with the number N of variables ($\binom{N}{n}$ constraints in the fully dense case), creating a heavy semidefinite formulation even for small n. Prior work finds that using an interior point solver combined with linear cuts, e.g. the widely used $\mathcal{M}$, leads to impractical computational times [18, 85, 90, 91]. As explained in Section 1, linear formulations are well-suited to the Branch&Cut framework and exploit a large body of existing work. Thus, we aim to construct strong linear outer-approximations rather than solving decompositions such as $\mathcal{S}_n$ $\forall n \in 1, \dots, N$ using interior point methods. □

The Remark 3.1.9 observations motivate the problem of selecting which low-dimensional semidefinite constraints to include in linearized form. The next subsections explore this idea for any decomposition $\mathcal{S}(\mathcal{F})$, $\mathcal{F} \subseteq \mathcal{P}$ (including $\mathcal{S}_n$, $\forall n \in 1, \dots, N$) by inferring strong linear cutting planes from semidefinite constraints which an optimality-based estimator can select based on.

3.2 From semidefinite relaxations to optimality-based linear cuts

First, Lemma 3.2.1 formalizes the intuition on using incumbent solutions through partial fixings to develop strong linearizations of a semidefinite constraint based on positive semidefinite (PSD)

matrix completion. Corollary 3.2.2 specializes the results to linearizing quadratic semidefinite relaxations with a specific fixing. Based on the previous, Lemma 3.2.3 creates a semidefinite outer-approximation retaining only cuts directly based on QP objective structure or optimality. Finally, Remark 3.2.4 discusses the implications of the fixing chosen for quadratic semidefinite relaxations.

Lemma 3.2.1 *(On the outer-approximation of a positive semidefinite (PSD) matrix completion)*

Denote by (and see Fig. 2 for an illustration):

- $A \in \mathbb{R}^{n\times n}$ a full symmetric matrix of variables.
- $A(a_m, a_f) \in \mathbb{R}^{n\times n}$ a partial fixing of A, obtained for a given m by partitioning A variables into:
  - vector $a_m \in \mathbb{R}^m$ of m unfixed variables;
  - vector $a_f \in \mathbb{R}^{n^2-m}$ of fixed variables.

  The partitioning assumes a list of matrix coordinates to position all elements of a_m and a_f within $A(a_m, a_f)$, called a *pattern* in the matrix completion literature [53]. Such a pattern is assumed to be given in all assertions made.
- $\mathbb{C} \subset \mathbb{R}^{n\times n}$ a bounded closed convex set, representing variable bounds and linear/convex cuts added during Branch&Cut.
- $\mathbb{C}(a_f)$ the projection (in the a_m space) of the set $\mathbb{C}$ for a given a_f.

Assuming for all a_f s.t. $\mathbb{C}(a_f) \neq \emptyset$ the PSD cone $\mathcal{S}(a_f) := \{a_m | A(a_m, a_f) \succeq 0\}$ is non-empty, bounded and closed ($A(a_m, a_f)$ admits PSD completion) then:

(i) *(For a given fixing, the PSD completion cone can be fully described by linear inequalities)* For a given fixing a_f, denoting $A(a_m, a_f)$ as $A(a_m)$ for simplicity,

$$A(a_m) \succeq 0 \;\Leftrightarrow\; \forall q \in \mathbb{R}^m :\; q \cdot a_m \geq q \cdot a_m^* = \min_{a_m} \{q \cdot a_m \mid A(a_m) \succeq 0\}\,.$$

(ii) *(Outer-approximation ($\mathcal{F}_m^q$) of the intersection ($\mathcal{F}_m$) of a convex and a PSD cone via one q)* For given $q \in \mathbb{R}^m$, defining the convex sets

$$\mathcal{F}_m^q = \left\{ A(a_m, a_f) \in \mathbb{C} \;\middle|\; \forall a_f \text{ s.t. } \mathbb{C}(a_f) \neq \emptyset : q \cdot a_m \geq q \cdot a_m^* = \min_{a_m} \{q \cdot a_m \mid A(a_m, a_f) \succeq 0\} \right\},$$
$$\mathcal{F}_m = \{\, A(a_m, a_f) \in \mathbb{C} \mid \forall a_f \text{ s.t. } \mathbb{C}(a_f) \neq \emptyset : A(a_m, a_f) \succeq 0\},$$

then $\mathcal{F}_m \subseteq \mathcal{F}_m^q$.

(iii) *(A smaller convex $\mathbb{C}$ size implies a tighter outer-approximation)* Marking the dependence on $\mathbb{C}$ explicitly as $\mathcal{F}_m^q(\mathbb{C})$, $\mathcal{F}_m(\mathbb{C})$ and introducing $\mathbb{C}' \subseteq \mathbb{C}$, then

$$\left(\mathcal{F}_m^q(\mathbb{C}') \setminus \mathcal{F}_m(\mathbb{C}')\right) \subseteq \left(\mathcal{F}_m^q(\mathbb{C}) \setminus \mathcal{F}_m(\mathbb{C})\right).$$

(iv) *(A larger fixing size implies a tighter outer-approximation)* If one additional variable is fixed in partition $A(a_{m-1}, a'_f)$ with $a_{m-1} \subseteq a_m$, $a'_f \supseteq a_f$ (assuming for all fixings a'_f s.t. $\mathbb{C}(a'_f) \neq \emptyset$ the PSD cone $\mathcal{S}(a'_f)$ is non-empty, bounded and closed), and a pattern augmenting the given one for $A(a_m, a_f)$ with the position of the additionally fixed variable, then

$$\left(\mathcal{F}_{m-1}^q \setminus \mathcal{F}_{m-1}\right) \subseteq \left(\mathcal{F}_m^q \setminus \mathcal{F}_m\right).$$

(v) *(An eigenvalue cut implicitly fixes a given q)*
For a given fixing a_f, denote $A(a_m, a_f)$ as $A(a_m)$ for simplicity, and let a complete fixing $A(\tilde{a}_m)$ of matrix A where a_m is fixed to $\tilde{a}_m$. Assume $A(\tilde{a}_m)$ has an eigen-pair $(\lambda_i < 0, v_i)$, $Q' = v_i v_i^T \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^m$ a vector of the Q' values at the positions (pattern) corresponding to those of the a_m variables in A. Then,

$$A(a_m) \succeq 0 \quad \Rightarrow \quad \begin{cases} q \cdot a_m \geq q \cdot a_m^* = \min\limits_{a_m} \{q \cdot a_m | A(a_m) \succeq 0\}, \\ \text{which is violated by } -\lambda_i = q \cdot (a_m^* - \tilde{a}_m) \text{ at } A(\tilde{a}_m). \end{cases}$$

Proof (i) Consider any point $\tilde{a}_m$ in the a_m space outside the PSD cone $\mathcal{S}(a_f)$. Via the separation theorem [14], $\exists q \in \mathbb{R}^m$ determining one RHS cut that separates $\tilde{a}_m$ from the bounded closed cone $\mathcal{S}(a_f)$. Such a cut determines/contains a tangent point on the boundary of $\mathcal{S}(a_f)$. Generating similar cuts for all points in the a_m space is possible via the infinite collection of hyperplanes on the RHS; collectively, such cuts describe the full boundary of $\mathcal{S}(a_f)$, determining it.

(ii) Assuming any given fixing a_f s.t. $\mathbb{C}(a_f) \neq \emptyset$, let the projections of $\mathcal{F}_m^q$ and $\mathcal{F}_m$ in the a_m space be $\mathcal{F}_m^q(a_f)$ and $\mathcal{F}_m(a_f)$, respectively. For the given q, (i) implies $\mathcal{F}_m(a_f) \subseteq \mathcal{F}_m^q(a_f)$, leading to

$$\bigcup_{\forall a_f \text{ s.t. } \mathbb{C}(a_f)\neq\emptyset} \mathcal{F}_m(a_f) = \mathcal{F}_m \quad \subseteq \quad \mathcal{F}_m^q = \bigcup_{\forall a_f \text{ s.t. } \mathbb{C}(a_f)\neq\emptyset} \mathcal{F}_m^q(a_f).$$

(iii) Since $\mathbb{C}' \subseteq \mathbb{C}$, we have

$$\begin{aligned} \mathcal{F}_m^q(\mathbb{C}') \setminus \mathcal{F}_m(\mathbb{C}') &= \left(\mathcal{F}_m^q(\mathbb{C}) \setminus \mathcal{F}_m^q(\mathbb{C} \setminus \mathbb{C}')\right) \setminus \left(\mathcal{F}_m(\mathbb{C}) \setminus \mathcal{F}_m(\mathbb{C} \setminus \mathbb{C}')\right) \\ &= \left(\mathcal{F}_m^q(\mathbb{C}) \setminus \mathcal{F}_m(\mathbb{C})\right) \setminus \left(\mathcal{F}_m^q(\mathbb{C} \setminus \mathbb{C}') \setminus \mathcal{F}_m(\mathbb{C} \setminus \mathbb{C}')\right) \subseteq \left(\mathcal{F}_m^q(\mathbb{C}) \setminus \mathcal{F}_m(\mathbb{C})\right). \end{aligned}$$

(iv) Fixing an additional variable on top of a fixing a_f is analogous to restricting $\mathbb{C}$ with an additional variable equality, and therefore the result from (iii) applies, concluding the proof.

(v) From $A(\tilde{a}_m)v_i = \lambda_i v_i$, we have $\lambda_i = v_i^T A(\tilde{a}_m) v_i = Q' \bullet A(\tilde{a}_m) < 0$. Since $A(a_m)$ admits PSD completion ($\exists a_m \in \mathbb{R}^m$ so that $v_i^T A(a_m) v_i = Q' \bullet A(a_m) > 0$) but $Q' \bullet A(\tilde{a}_m) < 0$, there $\exists \tilde{a}'_m \in \mathbb{R}^m$ so that $Q' \bullet A(\tilde{a}'_m) = 0$. This implies

$$\lambda_i = v_i^T A(\tilde{a}_m) v_i = Q' \bullet A(\tilde{a}_m) - Q' \bullet A(\tilde{a}'_m) = q \cdot \tilde{a}_m - q \cdot \tilde{a}'_m,$$

and therefore

$$A(a_m) \succeq 0 \Rightarrow v_i^T A(a_m) v_i > 0 \Rightarrow q \cdot a_m \geq q \cdot \tilde{a}'_m.$$

However, we have

$$\begin{aligned} a_m^* &= \arg\min_{a_m} \{q \cdot a_m \mid A(a_m) \succeq 0\} = \arg\min_{a_m} \left\{Q' \bullet A(a_m) = v_i^T A(a_m) v_i \;\middle|\; A(a_m) \succeq 0\right\} \\ &= \arg_{a_m} \left\{Q' \bullet A(a_m) = v_i^T A(a_m) v_i = 0\right\} \\ \Rightarrow Q' \bullet A(\tilde{a}'_m) &= Q' \bullet A(a_m^*) \Rightarrow q \cdot \tilde{a}'_m = q \cdot a_m^*. \end{aligned}$$

Therefore $A(a_m) \succeq 0 \Rightarrow q \cdot a_m \geq q \cdot a_m^*$ and $-\lambda_i = q \cdot (a_m^* - \tilde{a}_m)$, concluding the proof.
Note that the proof relies on $Q' = v_i v_i^T$ and can not be used for example to solve $P^{\tilde{x}}(\rho)$ when Q_ρ is indefinite and hence can not be similarly decomposed.

□

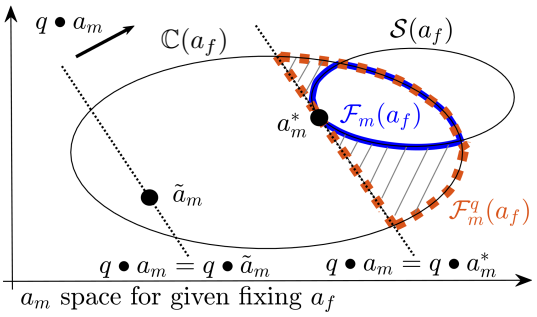


Fig. 2: The compact convex set $\mathbb{C}(a_f)$ represents the current feasible space at a Branch&Cut tree node projected onto the a_m-space for a given fixing a_f. The PSD cone $\mathcal{S}(a_f)$ intersects with $\mathbb{C}(a_f)$ to form $\mathcal{F}_m(a_f)$. Fixing one q refines $\mathbb{C}(a_f)$ to $\mathcal{F}_m^q(a_f)$ with a single cutting plane. The hashed region covers $\mathcal{F}_m^q(a_f) \setminus \mathcal{F}_m(a_f)$ and indicates the trade-offs of a single q: The q may be a strong search direction, but it may not eliminate all infeasible solutions.

Corollary 3.2.2 *(Lemma 3.2.1 specialisation to quadratic semidefinite relaxations via fixing x)* Given any $n \in \mathbb{N}, n \geq 1$, consider any fixing of variables $x \in \mathbb{R}^n$ to $\tilde{x} \in [0,1]^n$, lifted variables $X \in \mathbb{R}^{n \times n}$ $(x_i x_j = X_{ij})$. Then,

$$\begin{bmatrix} 1 & \tilde{x}^T \\ \tilde{x} & X \end{bmatrix} \succeq 0 \wedge \\ X_{ii} \leq \tilde{x}_i \ \forall i \quad \Leftrightarrow \quad \forall Q' \in \mathbb{R}^{n \times n} : \ Q' \bullet X \geq Q' \bullet X^* = \min_X \left\{ Q' \bullet X \middle| \begin{bmatrix} 1 & \tilde{x}^T \\ \tilde{x} & X \end{bmatrix} \succeq 0, \ X_{ii} \leq \tilde{x}_i \ \forall i \right\}.$$

Proof Any convex/ linear relaxation of a quadratic problem admits a PSD completion of $\begin{bmatrix} 1 & \tilde{x}^T \\ \tilde{x} & X \end{bmatrix}$, since there is a valid positive semidefinite relaxation at each fixed $\tilde{x}$ point. Thus, in Lemma 3.2.1(i) pick $A(a_m, a_f) = \begin{bmatrix} 1 & \tilde{x}^T \\ \tilde{x} & X \end{bmatrix}$ with $a_m = \text{vec}(X)$, $a_f = \tilde{x}$ and the associated fixing pattern. In this case, to ensure boundedness of the PSD cone $\mathcal{S}(a_f)$, $X_{ii} \leq \tilde{x}_i \ \forall i$ must be imposed. □

Lemma 3.2.3 *(From semidefinite decomposition to implied optimality-based cuts)* For a given SDP decomposition $\mathcal{F} \subseteq \mathcal{P}$, fix $x = \tilde{x}$ with $\tilde{x} \in [0,1]^N$ and $A\tilde{x} \leq b$, i.e. $\tilde{x}$ is an incumbent solution obtained after a cut round. Denote the QP objective as $f(X) := Q \bullet X$ and $(\forall \rho \in \mathcal{F})$ $f(X_\rho) := Q_\rho \bullet X_\rho$ with $Q_\rho \in \mathbb{R}^{|\rho| \times |\rho|}$ the submatrix slice of Q on ρ. Then,

$$z_{qp}(\mathcal{S}^{\tilde{x}}(\mathcal{F})) \geq z_{qp}^L(\mathcal{S}^{\tilde{x}}(\mathcal{F})) := \underbrace{\min_X \{ f(X) \mid \overbrace{\forall \rho \in \mathcal{F} : \ f(X_\rho) \geq f(X_\rho^* | \tilde{x}_\rho)}^{\text{optimality-based cuts}} \}}_{\text{main problem } P^{\tilde{x}}} + c^T \tilde{x}, \qquad (P^{\tilde{x}})$$

$$\text{where } (\forall \rho \in \mathcal{F}) \quad f(X_\rho^* | \tilde{x}_\rho) := \underbrace{\min_{X_\rho} \left\{ f(X_\rho) \middle| \begin{bmatrix} 1 & \tilde{x}_\rho^T \\ \tilde{x}_\rho & X_\rho \end{bmatrix} \succeq 0, \ X_{ii} \leq \tilde{x}_i \ \forall i \in \rho \right\}}_{\rho \text{ sub-problem } P^{\tilde{x}}(\rho)}. \qquad (P^{\tilde{x}}(\rho))$$

Proof Setting $x = \tilde{x}$ fixes the linear objective expression $c^T \tilde{x}$ and makes the constraints $A\tilde{x} \leq b, \ \tilde{x} \in [0,1]^N$ redundant. Then,

$$z_{qp}(\mathcal{S}^{\tilde{x}}(\mathcal{F})) - c^T \tilde{x} = \min_X \left\{ f(X) \middle| \forall \rho \in \mathcal{F} : \ \begin{bmatrix} 1 & \tilde{x}_\rho^T \\ \tilde{x}_\rho & X_\rho \end{bmatrix} \succeq 0, \ X_{ii} \leq \tilde{x}_i \ \forall i \in \rho \right\} \qquad (1)$$

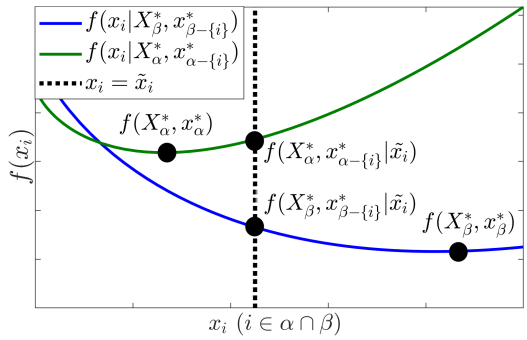


Fig. 3: Illustration of stronger outer-approximation via fixings $x_i = \tilde{x}_i$ for any semidefinite decomposition $\mathcal{F} \subseteq \mathcal{P}$.

Let distinct $\alpha, \beta \in \mathcal{F}$ and $i \in \alpha \cap \beta$. The figure shows the projections onto x_i of the semidefinite convex envelopes for the α, β sub-problems of type $P^{\tilde{x}}(\rho)$. Note that fixing a common $\tilde{x}_i$ and outer-approximating via a linear cut at successive iterations captures the full shape of the envelopes. This results in a tighter outer-approximation bound than the bound given by the overall envelope minimums $f(X^*_\alpha, x^*_\alpha)$ and $f(X^*_\beta, x^*_\beta)$, respectively.

$$= \min_X \left\{ f(X) \;\middle|\; \forall \rho \in \mathcal{F},\ \forall Q'_\rho \in \mathbb{R}^{n \times n} :\ Q'_\rho \bullet X_\rho \geq \min_{X_\rho} \left\{ Q'_\rho \bullet X_\rho \;\middle|\; \begin{bmatrix} 1 & \tilde{x}_\rho^T \\ \tilde{x}_\rho & X_\rho \end{bmatrix} \succeq 0, \; X_{ii} \leq \tilde{x}_i \ \forall i \in \rho \right\} \right\} \tag{2}$$

$$\geq \min_X \{ f(X) \mid \forall \rho \in \mathcal{F} :\ f(X_\rho) \geq f(X^*_\rho | \tilde{x}_\rho) \}, \tag{3}$$

where:

- Eq. (1) follows from $z_{qp}(\cdot)$ and $\mathcal{S}^{\tilde{x}}(\mathcal{F})$ definitions from Sections 2 and 3.1, respectively;
- Eq. (2) follows from Corollary 3.2.2;
- Eq. (3) follows by choosing only $(\forall \rho \in \mathcal{F})\ Q'_\rho = Q_\rho$ that match the QP objective (are optimality-based).

□

Remark 3.2.4 *(Implications of fixing $x = \tilde{x}$ and possible extensions)*
Fixing $x = \tilde{x}$:

- Ensures a valid matrix PSD completion at any value $\tilde{x}$ for any lifted QP relaxation, thus ensuring the validity of each linear inequality $(\forall \rho \in \mathcal{F})\ f(X_\rho) \geq f(X^*_\rho | \tilde{x}_\rho)$.
- Strengthens the linear outer-approximation. Assuming a decomposition $\mathcal{F} \subseteq \mathcal{P}$ without the fixing, the linear cuts $(\forall \rho \in \mathcal{F})\ f(X_\rho, x_\rho) \geq f(X^*_\rho, x^*_\rho)$ (where $f(X_\rho, x_\rho) := Q_\rho \bullet X_\rho + c_\rho^T x_\rho$ is dependent on unfixed x_ρ) capture the overall minimum (see Fig. 3) of each ρ sub-problem. Instead, as Lemma 3.2.1(iv) implies, fixing x captures stronger lower bounds by outer-approximating entire convex surfaces (see Fig. 3).
- Allows decomposition for ρ sub-problems $P^{\tilde{x}}(\rho)$ of both constraints and objective despite coupling constraints $Ax \leq b$ and linear objective parts, respectively.
- Allows augmented fixings and patterns in special cases. By Lemma 3.2.1(iv), a larger fixing size with additional variables, i.e. from X, that admits a PSD completion would strengthen $z^L_{qp}(\mathcal{S}^{\tilde{x}}(\mathcal{F}))$ with respect to fixing only $x = \tilde{x}$. Such fixings require searching for graph-based PSD completion patterns, e.g. [12, 31, 53, 56]. The Sections 3.3 and 4 approach in developing estimators for $P^{\tilde{x}}(\rho)$ solutions may be extended to any other fixing patterns and their associated semidefinite ρ sub-problems. But this paper focuses on the generic potential of outer-approximations based on partial fixings with optimality cut selection, and such extensions are outside its scope.

□

Intuitively, Lemma 3.2.3 linearizes semidefinite constraints (at $\tilde{x}$) based only on the objective structure/optimality and solutions of ρ sub-problems in the low dimensional (sparse) variable space X_ρ. Adding violated cutting planes for an increasing number of fixings $\tilde{x}$ where

$z_{qp}(\mathcal{S}^{\tilde{x}}(\mathcal{F})) \geq z^L_{qp}(\mathcal{S}^{\tilde{x}}(\mathcal{F}))$ converges to bounds $z_{qp}(\mathcal{S}(\mathcal{F})) \geq z^L_{qp}(\mathcal{S}(\mathcal{F}))$. Despite a weaker final bound z^L_{qp}, Section 5 shows that the $P^{\tilde{x}}$ relaxed formulation offers better convergence through the optimality selection of cuts corresponding to sub-problems $P^{\tilde{x}}(\rho)$. Therefore, we next discuss the optimality separation/selection of cutting planes from $\mathcal{S}_n$ using a parametric estimator, which is built in Section 4 for small $n \in \{2, 3, 4, 5\}$ using neural networks.

3.3 Cutting plane generation and selection (via estimator)

3.3.1 Cut generation via eigenvalues

Linear cutting planes need to be generated to outer-approximate any semidefinite constraint arising in a decomposition $\mathcal{F} \subseteq \mathcal{P}$ by cutting off the current solution point $(\tilde{X}_\rho, \tilde{x}_\rho)$.

Incorporating objective information, as in Lemma 3.2.3, can generate cuts tangent to the relaxation surface of a $\rho \in \mathcal{F}$ sub-problem $P^{\tilde{x}}(\rho)$, e.g. Fig. 4 for $|\rho| = n = 2$: (i) a tangent hyperplane at $f(X^*_\rho|\tilde{x}_\rho)$ (shown in transparent red); (ii) the closest tangent hyperplane at any point via projection. However, such approaches require numerically solving a semidefinite projection sub-problem, as any inexact estimation can lead to invalid cutting planes.

Following the literature, we also avoid the computational overhead of interior point solvers in generating cuts. Instead, for a selected ρ, we cut off solution point $(\tilde{X}_\rho, \tilde{x}_\rho)$ by a hyperplane based on the largest negative eigenvalue [85, 94], defined as

$$\lambda_{\min}(\rho) = \min(\Lambda), \text{ with } \Lambda \text{ the eigenvalues of } \begin{bmatrix} 1 & \tilde{x}_\rho^T \\ \tilde{x}_\rho & \tilde{X}_\rho \end{bmatrix}. \qquad (\lambda_{\min}(\rho))$$

If $\lambda_{\min}(\rho) < 0$ and $v(\rho)$ is the corresponding eigenvector, generate and add the violated cut,

$$(\lambda_{\min}(\rho) =) \qquad v^T(\rho) \begin{bmatrix} 1 & x_\rho^T \\ x_\rho & X_\rho \end{bmatrix} v(\rho) \geq 0. \qquad (\text{EigCut}(\rho))$$

These eigenvalue cuts are well-known, but they are typically high-dimensional [94]. Rather than sparsifying high-dimensional cuts (see, e.g. [85]), our new contribution allows choosing sparse but effective cuts by combining selection strategies based on cut violation (i.e. feasibility measure) and expected objective improvement (i.e. optimality measure).

3.3.2 Cut selection via different measures, e.g. optimality-based

For a decomposition $\mathcal{F} \subseteq \mathcal{P}$ and current solution point $(\tilde{X}_\rho, \tilde{x}_\rho)$, cut selection involves choosing/ranking (by a measure) which $\rho \in \mathcal{F}$ to generate eigenvalue cut EigCut(ρ) for. Given a $\rho \in \mathcal{F}$ and solution point $(\tilde{X}_\rho, \tilde{x}_\rho)$, Corollary 3.2.2 and Eqs. (1)–(3) in the Lemma 3.2.3 proof lead to,

$$\underbrace{\left(\begin{bmatrix} 1 & \tilde{x}_\rho^T \\ \tilde{x}_\rho & \tilde{X}_\rho \end{bmatrix} \succeq 0 \Leftrightarrow P^{\tilde{x}}(\rho) \text{ feasible at } \tilde{X}_\rho\right)}_{\text{feasibility}} \rightarrow \underbrace{\left(f(\tilde{X}_\rho) \geq f(X^*_\rho|\tilde{x}_\rho)\right)}_{\text{optimality}}. \qquad (4)$$

The negation/violation of either side of implication (4) at incumbent solution $(\tilde{X}_\rho, \tilde{x}_\rho)$ leads to a valid/violated EigCut(ρ) for ρ, which can be ranked and selected to generate based on the measures:

- **Feasibility measure.** The absolute value of the largest negative eigenvalue, $-\lambda_{\min}(\rho)$, found at incumbent point $(\tilde{X}_\rho, \tilde{x}_\rho)$, represents the violation of the ρ-sliced semidefinite constraint $\begin{bmatrix} 1 & x_\rho^T \\ x_\rho & X_\rho \end{bmatrix} \succeq 0$. The measure is an Euclidian distance in the variable space (X, x) and does not incorporate or correlate explicitly with the QP objective. So feasibility cut selection may not yield the best objective function improvement, identifying *valid cuts* that are not necessarily strong. Prior work uses these negative eigenvalues for individual cut selection [85, 94].
- **Optimality measure.** The expressions:

$$f(X_\rho^*|\tilde{x}_\rho) - f(\tilde{X}_\rho) \approx \qquad (\mathcal{I}_X(\rho))$$
$$\hat{f}_n^*(Q_\rho, \tilde{x}_\rho) - f(\tilde{X}_\rho) = \hat{f}_n^*(Q_\rho, \tilde{x}_\rho) - Q_\rho \bullet \tilde{X}_\rho, \qquad (\hat{\mathcal{I}}_X(\rho))$$

  with estimator $\hat{f}_n^*(Q_\rho, \tilde{x}_\rho)$ defined in Eq. (5), represent both:
  (i) The *estimated objective improvement* on X_ρ variables possible at $\tilde{x}_\rho$ by cutting off the infeasible variable solution $\tilde{X}_\rho$ and its associated objective $f(\tilde{X}_\rho)$ (see Figure 4). By transposing implication (4), a positive $\mathcal{I}_X(\rho)$ corresponds to $P^{\tilde{x}}(\rho)$ infeasible at $\tilde{X}_\rho$. This ensures a cut removing the solution point $(\tilde{X}_\rho, \tilde{x}_\rho)$ exists, with the $\mathcal{I}_X(\rho)$ magnitude indicating the objective improvement possible by adding the cut. For given n, selecting a limited number of ρ sub-problems with the most positive $\mathcal{I}_X(\rho)$ for cut generation would likely result in the greatest objective improvement over the entire problem $P^{\tilde{x}}$. Therefore, optimality-based measures are very likely to select *strong cuts*.
  (ii) The *violation of an optimality-based cut for the main problem* $P^{\tilde{x}}$ at $(\tilde{X}_\rho, \tilde{x}_\rho)$.
- **Combined measure** (optimality and feasibility). Introduced in Definition 5.2.1, $\mathcal{C}(\rho)$ combines $\hat{\mathcal{I}}_X(\rho)$ and $\lambda_{\min}(\rho)$.

Optimality measures provide an avenue to select cuts likely to improve the objective most. But for instance when $\mathcal{F} = \mathcal{P}_n$, finding X_ρ^* or $f(X_\rho^*|\tilde{x}_\rho)$ $\forall \rho \in \mathcal{P}_n$ requires solving $\binom{N}{n}$ semidefinite-constrained sub-problems $P^{\tilde{x}}(\rho)$, which is prohibitive via iterative methods such as interior point when N is large and/or $n \geq 3$, contravening the Section 1 desirata (2) of inexpensive cut selection.

Alternatively, we introduce a ***fast estimator*** $\hat{f}_n^*(Q_\rho, \tilde{x}_\rho)$ (computed in Section 4 for small n) parametric explicitly only on inputs Q_ρ and $\tilde{x}_\rho$, with X_ρ^* implicit, given by,

$$(\forall \rho \in \mathcal{F},\ |\rho| = n) \quad \hat{f}_n^*(Q_\rho, \tilde{x}_\rho) \approx f(X_\rho^*|\tilde{x}_\rho) = \min_{X_\rho} \left\{ Q_\rho \bullet X_\rho \;\middle|\; \begin{bmatrix} 1 & \tilde{x}_\rho^T \\ \tilde{x}_\rho & X_\rho \end{bmatrix} \succeq 0,\ X_{ii} \leq \tilde{x}_i\ \forall i \in \rho \right\}. \qquad (5)$$

The estimator $\hat{f}_n^*(Q_\rho, \tilde{x}_\rho)$ approximates $\mathcal{I}_X(\rho)$ via $\hat{\mathcal{I}}_X(\rho)$ for any given ρ within a given error to provide a quick deterministic measure for selecting ρ sub-problems $P^{\tilde{x}}(\rho)$ and their associated eigenvalue cuts. Therefore, the neural network estimators developed in Section 4 take the complexity of cut selection offline.

4 Estimating sub-problem solutions via trained neural networks

This section builds estimators for $\hat{f}_n^*$ $(n \leq 5)$ and, by extension, for optimality measure $\hat{\mathcal{I}}_X(\rho)$ to select the strong, optimality-based Section 3.3 cuts. We learn $\hat{f}_n^*$ in a supervised manner with the goal of taking offline the computational complexity of optimality-based cut selection.

The data for the supervised learning task relies on inputs $\tilde{x}_\rho, Q_\rho$ and output $f(X_\rho^*|\tilde{x}_\rho)$. The regression problem is nonlinear, consisting of all points $\tilde{x}_\rho$ on a collection of convex surfaces, each

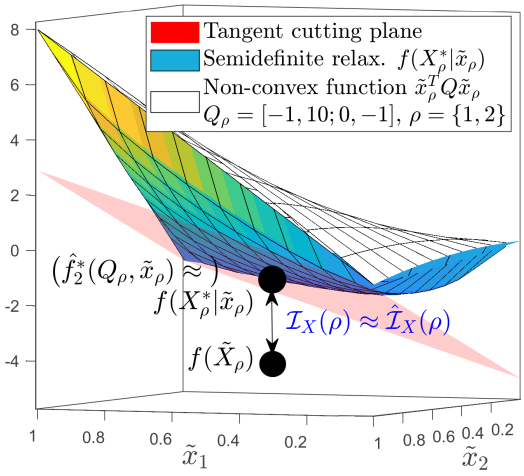


Fig. 4: Illustration, given $|\rho| = 2$, of the semidefinite relaxation (colored) that underestimates a quadratic nonconvex function (white) across $\tilde{x}_\rho \in [0, 1]^2$.

For a given incumbent solution point $(\tilde{X}_\rho, \tilde{x}_\rho)$ with objective $f(\tilde{X}_\rho)$ the $\mathcal{I}_X(\rho) \approx \hat{\mathcal{I}}_X(\rho)$ maximal objective improvement on X_ρ variables possible at $\tilde{x}_\rho$ is found vertically by optimization only over X_ρ variables till touching the semidefinite envelope at the point $f(X^*_\rho|\tilde{x}_\rho)$.

uniquely determined by Q_ρ. The solution X^*_ρ associated with output $f(X^*_\rho|\tilde{x}_\rho)$ acts as a latent variable. To mitigate potential model bias, we do not explicitly specify a model but rather adapt a flexible model based on data and the latent nonlinear features in that data. Abundant data can be sampled randomly as explained in Section 4.1 and scaled appropriately for training a model. Therefore, deep neural networks provide the appropriate learner: nonlinear features are implicitly learned in hidden layers and low variance and bias are expected given sufficient data and model complexity. Section 4.2 illustrates the engineering choices of neural network architectures and their training.

4.1 Data - sampling and scaling

A trained (supervised) model is only as good as its training data. First, to generalize the trained model, the input-output data sampled/generated for training needs to be uniformly distributed in the important input features. Second, the domain scaling of the sampled training data needs to enable model learning (in this case neural network learning, see Section 4.2). Third, any input evaluated by a trained estimator needs scaling to meet the training data assumptions.

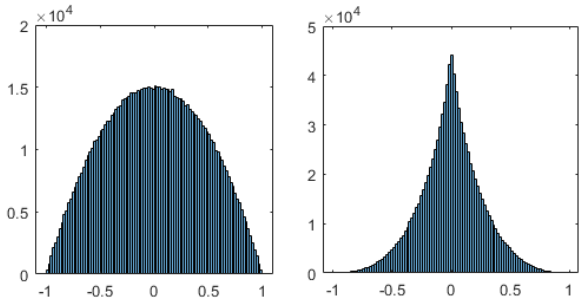


Fig. 5: Histograms, given $|\rho| = 3$, of any on-diagonal (left) and off-diagonal (right) elements for 1 million samples of Q_ρ generated through the eigen-decomposition shown in Table 1.

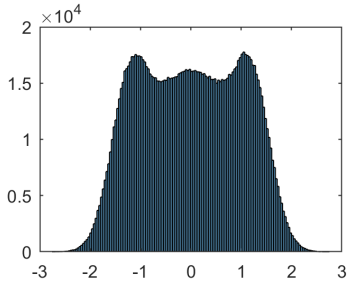


Fig. 6: Histogram, given $|\rho| = 3$, of eigenvalues for 1 million samples of Q_ρ generated via matrix entries uniformly distributed in $[0, 1]$.

Table 1: Sampling of inputs used as data to train neural networks for estimating $\hat{\mathcal{I}}_X(\rho)$ for a given ρ.

Inputs	Features to learn from inputs	(Input) Sampling uniformly on features	
		Sampled variables	Sample distributions & domains
$\tilde{x}_\rho$	Point positioning across a convex surface	Point $\tilde{x}_\rho$	$U([0,1]^{\lvert\rho\rvert})$
Q_ρ $(= V(\Lambda I_{\lvert\rho\rvert})V^T)$	Positive/negative semidefiniteness (shape/curvature of convex surface)	Eigenvalue vector Λ	$U([-1,1]^{\lvert\rho\rvert})$
		Orthonormal basis/matrix V	$U(O^{\lvert\rho\rvert\times\lvert\rho\rvert})$ (uniform on orthogonal group)

Analyzing these data sampling and scaling issues with regards to inputs $\tilde{x}_\rho$, Q_ρ and output $f(X^*_\rho|\tilde{x}_\rho)$ for every relevant ρ:

- ***Input sampling.*** Table 1 shows how independent inputs $\tilde{x}_\rho$ and Q_ρ are sampled in order to distribute the training data uniformly over features. Input $\tilde{x}_\rho$ localizes a point on a convex surface (the convex shell in Fig. 4) and can thus be directly sampled. However, we do not sample the symmetric coefficient matrix Q_ρ uniformly over its elements, even though it determines the properties (curvature) of a convex surface (again, the convex shell in Fig. 4). Fig. 6 shows sampling matrix Q_ρ entries directly results in an unwanted non-uniform eigenvalue distribution. The extensive literature on sampling positive definite [77] and correlation matrices [54] also avoids sampling Q_ρ entries directly. Instead, an easy extension to sampling negative definite matrices can be done if sampling is based on a random spectrum [54] by simply allowing negative eigenvalues. We adopt this sampling technique for any Q_ρ via random eigenvalues and orthonormal bases (sampled using the SciPy [81] function *scipy.stats.ortho_group* implementing [70]) as shown in Table 1. Fig. 5 shows that sampling Q_ρ via its eigen-decomposition does not lead to uniform distributions of Q_ρ elements, which conversely further validates not sampling uniform Q_ρ elements directly.
- ***Output sampling.*** For each $\tilde{x}_\rho$, Q_ρ sampled as described above, $f(X^*_\rho|\tilde{x}_\rho)$ is the exact solution of problem $P^{\tilde{x}}(\rho)$ computed using interior point solver Mosek [7] with default settings.
- ***Training data domain scaling.*** Table 1 limits sampled eigenvalues Λ to the range $[-1,1]$. This domain choice is driven by (i) the Λ features and (ii) the Λ-dependent domain of Q_ρ. The domain and sign of the Λ elements determines the shape/nonconvexity that Q_ρ induces in the QP objective, i.e. $x^T_\rho Q_\rho x_\rho$. Thus, Λ impacts the distance from $x^T_\rho Q_\rho x_\rho$ to its semidefinite relaxation in the sub-problem $P^{\tilde{x}}(\rho)$ with solution $f(X^*_\rho|\tilde{x}_\rho)$, and by extension impacts the magnitude of $\hat{f}^*$. Consequently, eigenvalue signs and their relative magnitudes directly impact $\hat{f}^*_n$ as relevant features and must be maintained. However, since symmetric (sigmoid-like) activation functions for neural networks have non-zero gradients only over limited domains, the domain of Q_ρ elements (as inputs to activation functions) must be bounded as well. A scaled down eigenvalue domain of $[-1,1]$ retains all relevant features and ensures the domain of Q_ρ elements is also $[-1,1]$-bounded (see Lemma 4.1.1). The $[-1,1]$-bounded Q_ρ elements and $[0,1]$-bounded $\tilde{x}_\rho$ (via the QP formulation) are therefore inputs a sigmoid-like activation function can learn over (see Section 4.2).
- ***Input rescaling (at evaluation).*** Any QP instance should be a possible input for the trained model. For the original QP to meet the training data assumptions, some inputs may need re-scaling, i.e. both Q_ρ elements and eigenvalues Λ are re-scaled to $[-1,1]$, $\forall\rho$. Dividing the input Q_ρ by a constant scaling factor appropriately bounds the Q_ρ elements and their implicit eigenvalues Λ. The model output obtained using the scaled Q_ρ can then be multiplied by the scaling factor to estimate the original Q_ρ input. This scaling is possible

because $\hat{f}_n^*(Q_\rho, \tilde{x}_\rho) \approx f(X_\rho^*|\tilde{x}_\rho)$ is linear in the Q_ρ input (see $P^{\tilde{x}}(\rho)$). Lemma 4.1.2 finds a Q_ρ scaling factor that bounds implicit Λ within $[-1, 1]$, but special cases and more elaborate proofs may enable tighter bounds with a smaller scaling factor, see e.g. [46]. The QP definition guarantees that $\tilde{x}_\rho$ does *not* need re-scaling, but more general QP with arbitrary bounds could also be re-scaled to $[0, 1]$.

Lemma 4.1.1 *(eigenvalue bounds → matrix bounds)*
If all eigenvalues of $M \in \mathbb{R}^{n\times n}$ are in $[-m, m]$ then any element in M is in $[-m, m]$.

Proof Let eigen-pairs of M be $(\lambda_i,\ v_i)$ for $\forall i \in 1, \ldots, n$, and let v_{ij} be the j-th element of eigenvector v_i. Then $|M_{ij}|$ satisfies,

$$|M_{ij}| = \left|\sum_{k=1}^{n} v_{ki}v_{jk}\lambda_k\right| \leq \sum_{k=1}^{n} |v_{ki}v_{jk}| \cdot |\lambda_k| \leq \sum_{k=1}^{n}((v_{ki}^2 + v_{jk}^2)/2) \cdot |\lambda_k| \leq \sum_{k=1}^{n}((v_{ki}^2 + v_{jk}^2)/2)m = m$$

□

Lemma 4.1.2 *(matrix bounds → eigenvalue bounds) [46], Theorem A1.*
Any eigenvalue λ of $M \in \mathbb{R}^{n\times n}$ satisfies $|\lambda| \leq n \cdot \max_{i,j} M_{ij}$. □

4.2 Neural network training

Motivated by Section 3.1 while accounting for n upper-bounds implied by Remark 3.1.3 and Fig. 1, we learn an estimate solution $\hat{f}_n^*$ of problem $P^{\tilde{x}}(\rho)$ for limited small $n \in \{2, 3, 4, 5\}$ by using neural network models (see Fig. 7 test results on trained networks). We generate sufficient random data with the Section 4.1 considerations and make the following neural architecture and training choices:

- Topology (by experimentation): $3 - 4$ hidden layers, each with $\approx 50 - 64$ neurons;
- Activation function in the input/hidden layers: *tanh*;
- Learning algorithm (back-propagation): scaled conjugate gradient;
- Performance function: mean squared error;
- 1 million sampled data points randomly partitioned for training, validation and testing in proportions 75%/ 15%/ 10%, respectively;
- Weights initialization: Nguyen-Widrow (Matlab Neural Network Toolbox default);
- Stopping criteria: gradient/performance thresholds, validation set performance;
- Training time: $\approx 5 - 20$ hours on an Intel i7-4770 CPU.

The activation function in the input/hidden layers is an important neural network modeling choice [48]. Two activation classes for neural network regression are sigmoid-based functions and variants of rectified linear units (*ReLU*) [47, 66]. Of the sigmoid-based functions, the hyperbolic tangent *tanh* is typically fastest and easiest to train [61]. Further, the *tanh* matches our data assumptions around learning $P^{\tilde{x}}(\rho)$ solutions:

- **The *tanh* activation is symmetric around 0.** Matches the symmetric, 0-centred domains of the data generated according to Section 4.1.
- **The *tanh* gradient is concave with maximum attained at 0.** For all ρ, $Q_\rho \in [-1, 1]^{n\times n}$ values close to 0 and their sign are crucial to solving $P^{\tilde{x}}(\rho)$ because a sign change near 0 may provoke a jump in the solution. Since *tanh* has its highest gradient at 0, inputs Q_ρ close to 0 will be well-learned.

- **The *tanh* activation is significantly positive on the domain $[-4, 4]$ with a bounded output range of $[-1, 1]$.** By being well inside the relevant *tanh* domain and co-domain, the bounded inputs $Q_\rho \in [-1, 1]^{n\times n}, \tilde{x}_\rho \in [0, 1]^n$ mitigate saturation, i.e. the activation function does not squeeze the inputs and thereby lose important features. Therefore, the common sigmoid activation problem of vanishing gradients [84] is likely to be avoided, as the Fig. 7 computational results show on test data.

Alternatively, we could use simple *ReLU* for learning $\hat{f}_n^*$ as a collection of smooth convex surfaces that convey curvature through deeper, sparser networks. This increased architecture involves potentially more testing and possibly requires efficient regularization, issues we seek to avoid. Another possibility to keep architectures dense is a leaky *ReLU* activation [66]. Given the good generalization obtained with *tanh* in Fig. 7, the only distinctive improvements correctly engineered *ReLU* architectures can bring are training speed (less significant for small $n \leq 5$) and faster (trained) neural net evaluation. So *ReLU* activations could potentially gain computational performance at evaluation time for any n and keep training time manageable for increased n ($\gg$ 5), both outside the scope of our analysis.

Alongside *tanh* activation, we employ the scaled conjugate gradient (SCG) learning algorithm [78]. SCG is typically used in moderate-sized neural network architectures [29, 88, 89], e.g. those in Fig. 7. Via approximate second-order information, SCG can further prevent *tanh* from under-fitting due to vanishing gradients and achieve faster convergence than either first-order backpropagation or the more accurate Levenberg-Marquardt [51, 76, 92].

Figure 7 illustrates, on independent test data, the good generalization of neural networks trained using our proposed methodology. Observe in Fig. 7: (i) tight fits, (ii) residuals that are normally distributed with no major skew and centered around 0, and (iii) scatter plots of residuals versus fits lacking any patterns. Low residuals coupled with stable results on a large test set imply the trained neural networks offers both low bias and variance. There is a slight deterioration in the fits and residuals variance (as evidenced by the 95% confidence lines) when increasing n from 2 to 5, which is expected when learning progressively more complex models. Thus, under the choices made in terms of data sampling and model training, Fig. 7 show neural networks can successfully learn the solution space of ρ sub-problems $P^{\tilde{x}}(\rho)$ via $\hat{f}_n^*$, implicitly estimating the optimality measure $\hat{\mathcal{I}}_X(\rho)$. Furthermore, the next section shows $\hat{\mathcal{I}}_X(\rho)$ adequately estimates $\mathcal{I}_X(\rho)$ in practice for the purpose of optimality selection.

5 Strong semidefinite cut selection for QP problems

This section implements and analyses the optimality-based selection of cutting plane discussed in Sections 3.2–3.3 for outer-approximating the low-dimensional semidefinite relaxation of QP problems. Algorithm 1 sets the framework to outer-approximate $\mathcal{B} + \mathcal{S}(\mathcal{F})$, given any $\mathcal{B}$ linear base relaxation and $\mathcal{F} \subseteq \mathcal{P}_n$ for small $n \leq 5$. We compare and analyze cut selection based on different orderings/selections of $\rho \in \mathcal{F} \subseteq \mathcal{P}_n$ sub-problems. Testing on all 99 BoxQP instances [26, 28, 98], grouped as in Table 3, we analyze optimality-based, i.e. $\hat{\mathcal{I}}_X(\rho)$-based, cut selection in different setups. The results, summarized in Table 2, illustrate the Lemma 3.2.1 implications and, more importantly, show the Section 1 desirata is met.

First, Section 5.1 shows that the highly-scalable, deterministic $\hat{\mathcal{I}}_X(\rho)$ evaluated via the Section 4 neural network estimators adequately substitutes $\mathcal{I}_X(\rho)$ in terms of optimality selection convergence. Section 5.2 then compares the convergence properties of selecting, within the set of cuts EigCut(ρ), the feasibility versus optimality versus random cuts. The optimality selection shows stronger early convergence but weaker bounds compared to feasibility selection. As a result,

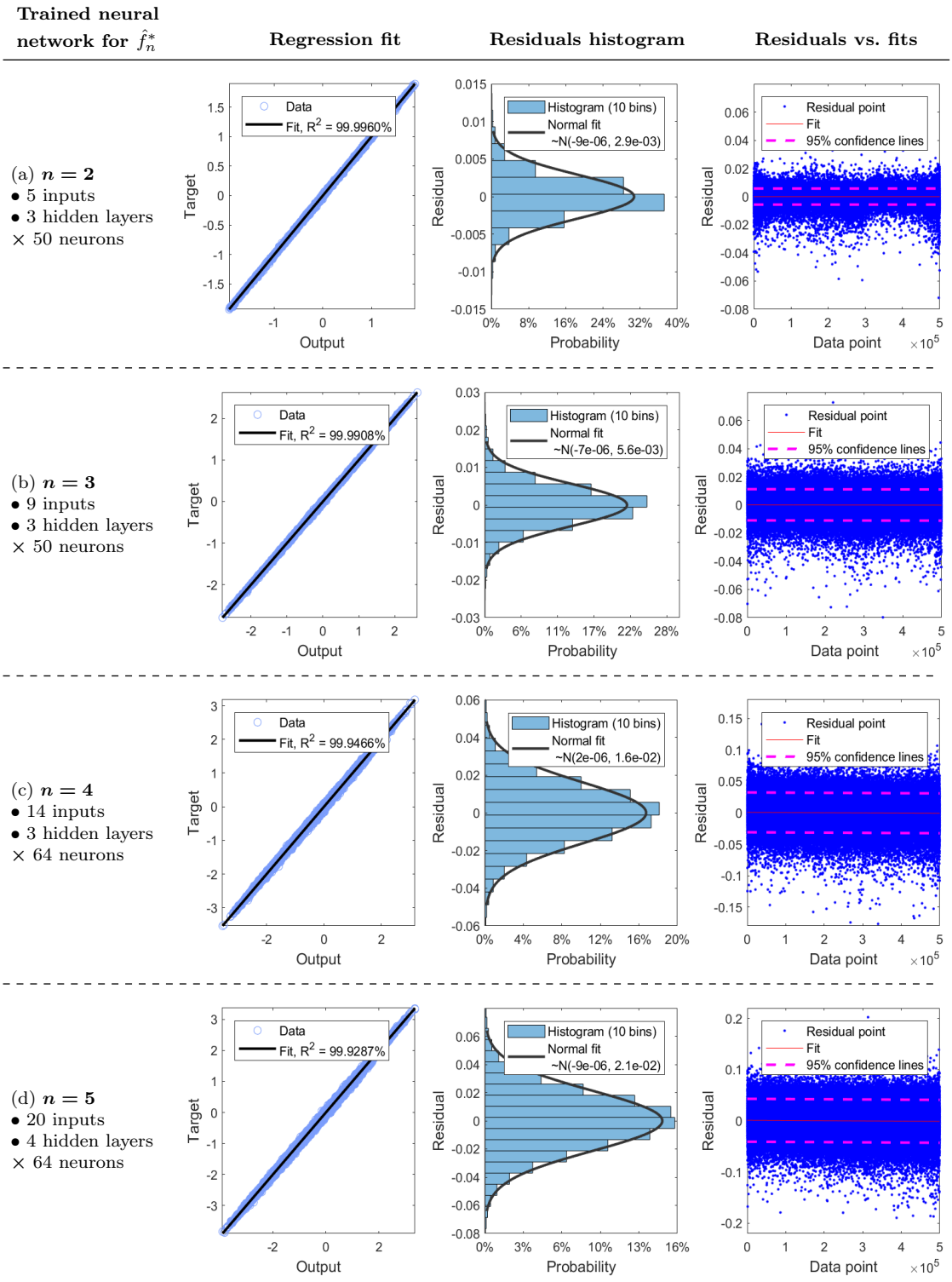


Fig. 7: Performance of trained neural networks for $\hat{f}_n^*, \forall n \in \{2, 3, 4, 5\}$. For each n (and the associated network topology), the performance of the trained estimator is shown on a test set of 0.5 million data points (not used in training/validation) in terms of fit and residuals.

Table 2: Summary of cut selection/ordering based on different measures, computational needs (iterative vs. deterministic), and convergence and bounds obtained employing them.

Ordering/ selection of cuts/ sub-problems $\rho \in \mathcal{F} \subseteq \mathcal{P}_n$		Measure evaluation (max #evals $\leq \binom{N}{n}$ for $\mathcal{F} \subseteq \mathcal{P}_n$)		Cuts selected	Convergence (+good, −bad)		Max bound
Type	Measure	Type	Iterative/ Deterministic		First iters	Last iters	
Feasibility	$-\lambda_{\min}(\rho)$	Eigenvalue decomp.	Iter.	valid	−	+	$= z_{qp}$
Optimality	$\mathcal{I}_X(\rho)$	SDP sol. (interior pt. [7])	Iter.	strong	+	−	$\geq z^L_{qp}$
	$\hat{\mathcal{I}}_X(\rho)$	Estimator eval. *(neural net, see Section 4)*	Det.				$\leq z_{qp}$
Combined (feas.+opt.)	$\mathcal{C}(\rho)$ *(see Section 5.2)*	Eigenvalue decomp. & Estimator eval.	Iter.+ Det.	strong & valid	+	+	$= z_{qp}$
Random	n/a	Random number	Det.	random	−	−	$= z_{qp}$

a new combined ordering measure selects cuts based on both optimality and feasibility, showing both strong convergence and strong final bounds. Section 5.3 first analyses the impact of cut sparsity/dimensionality on cut selection, proposing a heuristic based on instance structure, followed by a discussion on the tradeoffs of incorporating chordal extensions. Finally, Section 5.4 shows even tight/competitive QP relaxations can be cheaply improved via low-dimensional semidefinite cuts, meeting the Section 1 desirata. All results can be reproduced via the source code available on Github [9].

Throughout this section, we will refer to *valid* cuts, i.e. all cuts EigCut(ρ) violating the feasibility measure, versus *strong* cuts, i.e. the cuts favored by our optimality measure. Our hypothesis, which we will motivate in the following sections, is that an optimization solver should automatically select the subset of *strong* cuts within the set of *valid* cuts. Feasibility-based cut selection is known to converge slowly. But, as a consequence of Lemma 3.2.1(ii), naïve, greedy cut selection may not converge at all. We claim, and attempt to show in the sequel, that selecting the *strong* cuts within the set of *valid* cuts is key to good performance.

5.1 Optimality cut selection - estimated versus exact measures

First, we analyze optimality-based cut selection by comparing the convergence properties of ordering on either the estimated $\hat{\mathcal{I}}_X(\rho)$ (deterministic evaluation of a trained neural net) or the exact $\mathcal{I}_X(\rho)$ (iterative interior point [7] solution). The fast, vectorizable, parametric computation of $\hat{\mathcal{I}}_X(\rho)$ is shown empirically to be an adequate replacement for $\mathcal{I}_X(\rho)$, allowing the effective scaling needed for optimality selection of strong cuts to compete with feasibility selection (see Section 3.3).

Fig. 8 illustrates cut selection for outer-approximating $\mathcal{M} + \mathcal{S}_3$ for a small dense BoxQP instance where ρ sub-problems $P^{\tilde{x}}(\rho)$ are selected for cut generation across 4 rounds. In all subfigures for each round of cuts, cut selection is based on the $\hat{\mathcal{I}}_X(\rho)$ optimality measure, but $\hat{\mathcal{I}}_X(\rho)$ is plotted in the three subfigure columns of Fig. 8 against different orderings of the ρ sub-problems:

- The first column shows unordered sub-problems/cuts selected based on estimated $\hat{\mathcal{I}}_X(\rho)$ decrease positive (expected objective improvement) $\mathcal{I}_X(\rho)$ towards 0 across all sub-problems. In turn, this improves overall bounds towards $z(\mathcal{M} + \mathcal{S}_3)$ as shown in the Fig. 8 table.

Algorithm 1: Iterative SDP outer-approx. with cut selection/generation based on an ordering

Input: - current base LP relaxation $\mathcal{B}$ of QP, either fully added from the start, i.e. $\mathcal{M}$, or (independently) separated iteratively at each cut round ($\triangle$, $0-1/2$, edge-concave);
- decomposed SDP relax. $\mathcal{S}(\mathcal{F})$ to outer-approx., where $\mathcal{F} \subseteq \mathcal{P}_n$ with small n, i.e. $\overline{\mathcal{P}}_n$, $\mathcal{P}'_n$ etc.;
- incumbent LP solution $(\tilde{x}, \tilde{X})$;
- selection strategy/ordering metric $M(\rho)\ \forall \rho \in \mathcal{F}$ at $(\tilde{x}, \tilde{X})$ e.g. $\hat{\mathcal{I}}_X(\rho)$, -$\lambda_{\min}(\rho)$, $\mathcal{C}(\rho)$ etc.;
- selection size, i.e. a fixed % of $|\mathcal{F}|$ (upper capped at 5000);
- number of cut rounds R (set to 20);
- termination criteria - if active, terminate on an improvement between two consecutive cut rounds of $< 0.01\%$ of the gap closed overall so far from the $\mathcal{M}$ bound;

Output: Polyhedral outer-approximation that lower-bounds $z(\mathcal{B}+\mathcal{S}(\mathcal{F}))$ and SDP relax. $z(\mathcal{B}+\mathcal{S})$;

for *R cut rounds if termination criteria not met* **do**
  Sort $\mathcal{F}$ by descending $M(\rho)\ \forall \rho \in \mathcal{F}$ at current $(\tilde{x}, \tilde{X})$;
  for *top ρ sub-problems in sorted $\mathcal{F}$ within selection size* **do**
    if $\lambda_{min}(\rho) < 0$ *(violated PSD condition for* $\begin{bmatrix} 1 & \tilde{x}_\rho^T \\ \tilde{x}_\rho & \tilde{X}_\rho \end{bmatrix}$*)* **then**
      $\mathcal{B} = \mathcal{B} \cap \{$new EigCut$(\rho)$ based on $\lambda_{\min}(\rho)\}$;
  Resolve (warm-start) new LP relaxation $\mathcal{B}$ that includes added cuts;
  Update current incumbent solution $(\tilde{x}, \tilde{X})$;
Last obtained $z(\mathcal{B})$ lower bounds $z(\mathcal{B}+\mathcal{S}(\mathcal{F}))$ and $z(\mathcal{B}+\mathcal{S})$;

Table 3: BoxQP instances, grouped by size and density [18].

Size	Low	Medium	Large	Jumbo	Density	Low	Medium	High
# vars	20-40	50-70	80-90	100-125	%	[0-40]	(40-60]	(60,100]

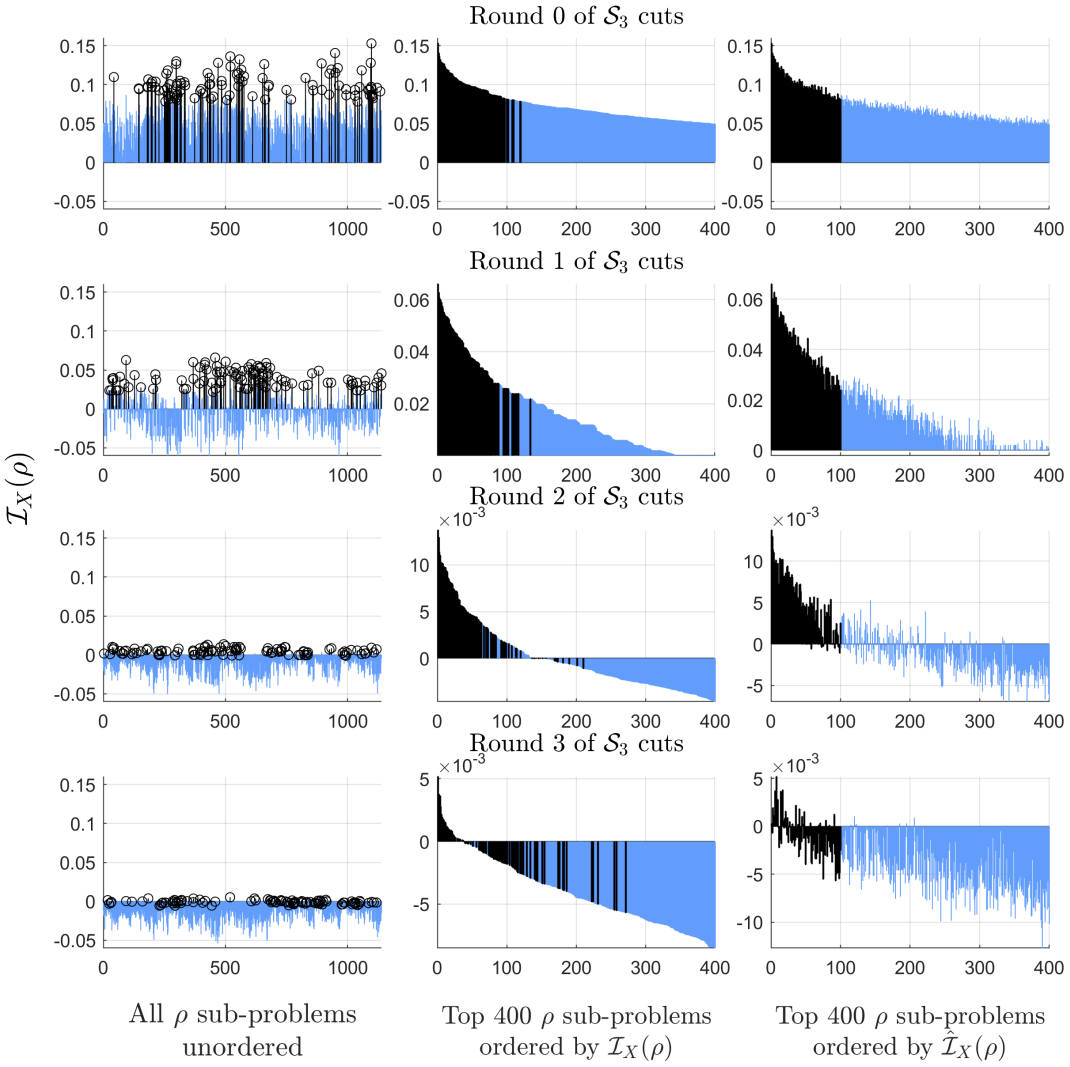
- The second column orders sub-problems decreasingly by $\mathcal{I}_X(\rho)$ - the more $\mathcal{I}_X(\rho)/\hat{\mathcal{I}}_X(\rho)$ discrepancies, the more gaps along the x-axis the black selected area has.
- The third column orders sub-problems decreasingly by estimated $\hat{\mathcal{I}}_X(\rho)$ - the more $\mathcal{I}_X(\rho)/\hat{\mathcal{I}}_X(\rho)$ discrepancies, the more non-decreasing $\mathcal{I}_X(\rho)$ is across the $\hat{\mathcal{I}}_X(\rho)$-ordered x-axis.

Across cut rounds (subfigure rows), the $\mathcal{I}_X(\rho)/\hat{\mathcal{I}}_X(\rho)$ discrepancies get more pronounced towards rounds 3-4 where the selection/ordering is made in a $\mathcal{I}_X(\rho)$ domain within the error range of the neural net estimators for $\hat{\mathcal{I}}_X(\rho)$ (see the residual plots in Section 4.2). The third column of the Fig. 8 table supports the same observation via the decreasing % of sub-problems selected by both $\mathcal{I}_X(\rho)$ and $\hat{\mathcal{I}}_X(\rho)$ orderings.

However, small differences in cut selection results between $\mathcal{I}_X(\rho)$ and $\hat{\mathcal{I}}_X(\rho)$ are to be expected, as any $\hat{\mathcal{I}}_X(\rho)$ estimator will reach its error range centred around 0 if given sufficient cut rounds. Nonetheless, as shown for BoxQP instances across varying sizes/densities in Fig. 9, in terms of bounds closed, the $\hat{\mathcal{I}}_X(\rho)$-based cut selection closely matches an equivalent $\mathcal{I}_X(\rho)$-based selection, showing the neural net estimators built in Section 4 are adequate models in practice.

5.2 Cut selection strategies - optimality versus feasibility versus random

This section compares, with respect to the % gap closed towards the $\mathcal{M}+\mathcal{S}$ relaxation bound across cut rounds, three selection strategies for outer-approximating $\mathcal{M}+\mathcal{S}_3$ via Algorithm 1. Fig. 9 analyses these selection strategies for BoxQP instances of different size (Figs. 9a–9b vs. Figs. 9c–9d) and density (Figs. 9a–9c vs Figs. 9b–9d). The results reveal:



Round #	% of $\mathcal{M}$ to ($\mathcal{M}$+$\mathcal{S}_3$) gap closed	% Same selection $\mathcal{I}_X(\rho)$ vs. $\hat{\mathcal{I}}_X(\rho)$
1	57.06	94
2	87.01	90
3	96.50	75
4	98.54	69

Fig. 8: Illustration (using Algorithm 1) on BoxQP dense instance *spar020-100-1* of optimality cut selection via neural net estimate $\hat{\mathcal{I}}_X(\rho)$ across 4 cut rounds, to converge from $z(\mathcal{M})$ towards $z(\mathcal{M}+\mathcal{S}_3)$.
For all subfigures, the y-axis represents $\mathcal{I}_X(\rho)$ (calculated using [7]) plotted as a function of sub-problems $\rho \in \mathcal{P}'_3$ on the x-axis. Each subfigure column orders the ρ sub-problems on the x-axis in a different way to compare $\mathcal{I}_X(\rho)$ and $\hat{\mathcal{I}}_X(\rho)$. Each subfigure row corresponds to a round of maximum 100 cuts EigCut(ρ) added for 100 selected ρ sub-problems $P^{\tilde{x}}(\rho)$. Selected sub-problems at each cut round are marked by black lines in all subfigures.

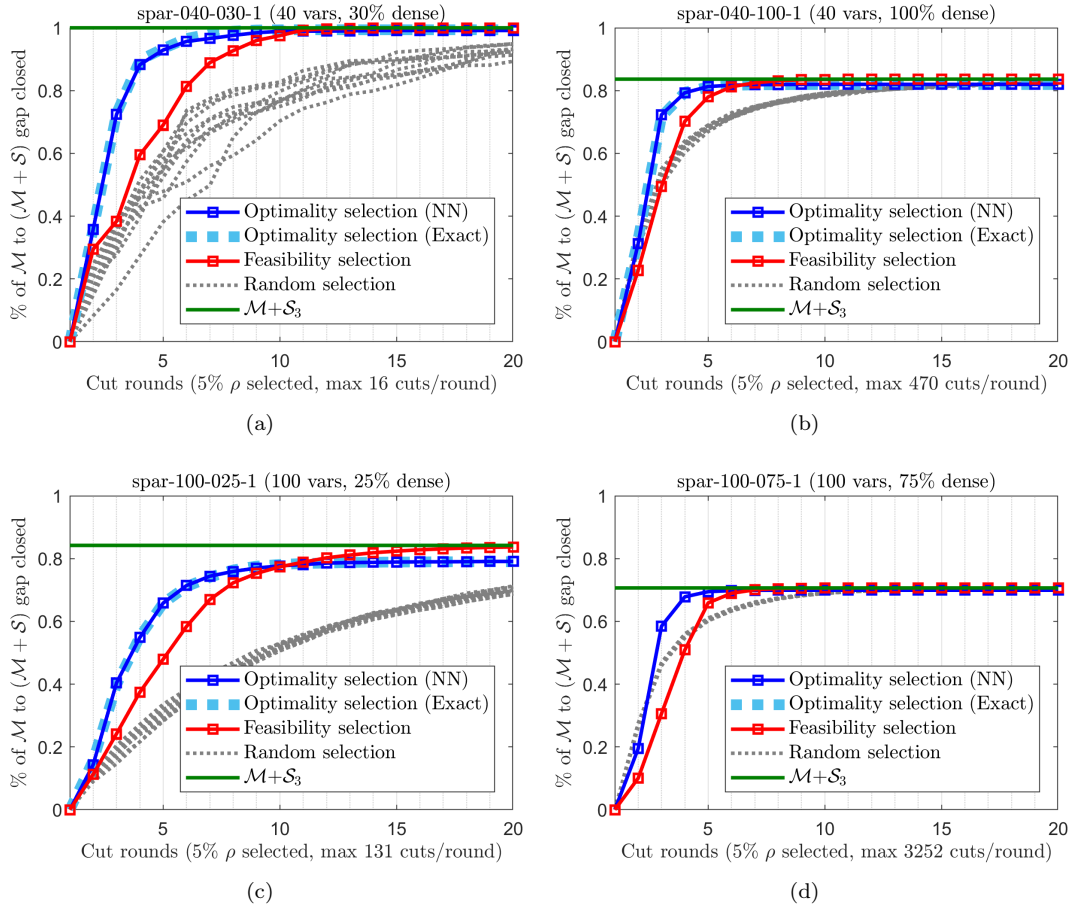


Fig. 9: Bound convergence (root node) for outer-approximating the $\mathcal{M}+\mathcal{S}_3$ relaxation starting from $\mathcal{M}$ for BoxQP instances with different sizes and/or densities. Algorithm 1 implements the Table 2 cutting plane selection/ordering strategies across 20 cut rounds, with selection size $5\% \cdot |\mathcal{P}'_3|$ (exact number shown for each instance). Each x-axis cut round shows the percentage of gap closed between the relaxations $\mathcal{M}$ and $\mathcal{M}+\mathcal{S}$, and the $\mathcal{M}+\mathcal{S}_3$ bound targeted by the outer-approximation is shown in green.

(o1) Low-dimensional SDP decomposition (Section 3.1) bounds can be closely outer-approximated using few linear cuts in a computationally light way as set out in the Section 1 desirata, e.g. the $\mathcal{M}+\mathcal{S}_3$ relaxations in Fig. 1.

(o2) Optimality-based selection converges the fastest, especially in early cut rounds. These results, showing strong cuts selection via objective information, are supported by the Section 1 assertions, the Sections 3.2–3.3 formalization, and the scalable implementation in Section 4.

(o3) Feasibility selection, which matches the $\mathcal{M}+\mathcal{S}_3$ bounds, exhibits stronger final bounds than optimality selection (Fig. 9c). These results are justified by Lemma 3.2.1(i), Lemma 3.2.1(v) and subsequently Eq. (4) in Section 3.3. Notably, the $\mathcal{M}+\mathcal{S}_3$ bounds can be reached by random selection, but poor convergence leads to large numbers of cuts and/or cut rounds.

A trade-off thus exists between faster convergence for optimality selection versus slightly better (given enough cut rounds) final bounds for feasibility selection at further cost. Fig. 10

recasts this trade-off into one between selecting fewer strong (valid) cuts and selecting all valid cuts, isolating these two competing goals for the Fig. 9c BoxQP instance:

- **Strong (valid) cuts are found by optimality selection** (Figs. 10a and 10c) .
  - Cut selection based on optimality measures (estimated or not) clearly leads to better bound (gap closed) improvements per cut than feasibility or random selection (with feasibility partially outperforming random). Moreover, Fig. 10c shows across gap closed that cuts selected by optimality are stronger until convergence to the final bound, i.e. until $\nexists\rho$ s.t. $\hat{\mathcal{I}}_X(\rho) > 0$.
- **All valid/violated cuts are found by feasibility selection** (Figs. 10b and 10d) .
  - Feasibility selection based on the violation $\lambda_{\min}(\rho)$ identifies valid cuts for the entire selection at every cut round assuming there are enough valid cuts remaining overall.
  - Random selection finds valid cuts only for a random selection subset, but is expected probabilistically, given sufficiently many cut rounds, to eventually find all valid cuts.
  - Optimality selection finds valid strong ($\hat{\mathcal{I}}_X(\rho) > 0$) cuts for the entire selection size in the first few cut rounds, but eventually, as expected by Eq. (4), finds no more valid strong cuts. However, Figure 10d shows the valid strong cut percentages drop only after most gap to $\mathcal{M} + \mathcal{S}_3$ is already closed.
  - Selecting by optimality measure $\mathcal{I}_X(\rho)$ rather than estimated $\hat{\mathcal{I}}_X(\rho)$ shows minor improvements, due to estimator error when $\mathcal{I}_X(\rho) \searrow 0$ (Section 5.1).

So optimality and feasibility selection are complementary, and a combined (optimality and feasibility) ordering can capture both strong and all valid cuts.

Definition 5.2.1 *(Combined measure for cut selection)*
For any sub-problem $\rho \in \mathcal{F} \subseteq \mathcal{P}$, define the combined measure

$$\mathcal{C}(\rho) = \begin{cases} \hat{\mathcal{I}}_X(\rho) + M, \text{ if } \hat{\mathcal{I}}_X(\rho) > 0 \text{ and } \lambda_{\min}(\rho) < 0, \\ -\lambda_{\min}(\rho) \text{ otherwise,} \end{cases} \qquad (\mathcal{C}(\rho))$$

where M is an arbitrary large positive number. □

By design, $\mathcal{C}(\rho)$ prioritizes strong valid cut selection via a positive expected objective improvement $\hat{\mathcal{I}}_X(\rho) > 0$ and uses a feasibility check to ensure cut violation (the weakness of optimality selection). Feasibility cut selection identifies further valid cuts in a second tier of sub-problems that offer no useful objective improvement information via $\hat{\mathcal{I}}_X(\rho)$. Combined cut selection via $\mathcal{C}(\rho)$ is thus guaranteed to outperform both optimality (using cut violation information) and feasibility (adding strong cuts first) selection, converging fast to a strong bound, as shown in Figs. 10–11. In terms of the combined selection running time: (i) minimal overhead (fast neural network evaluations) to feasibility selection alone is traded for faster convergence; (ii) eigendecomposition overhead (to get $\lambda_{\min}(\rho)$) and more valid cuts to optimality selection are traded for stronger final bounds.

Table 4 analyses the BoxQP final bounds obtained by pure optimality and feasibility selections, showing the (expected) gap in bounds between the two strategies is reduced by:

(o4) Increasing selection size, e.g. 5% to 10%, at each Algorithm 1 cut round;
(o5) Adding extra cuts, i.e. $\triangle$ on top of $\mathcal{M}$. The final $\mathcal{M}+\triangle$ relaxation bound is improved in almost equal measure by both strategies;
(o6) Higher instance density.

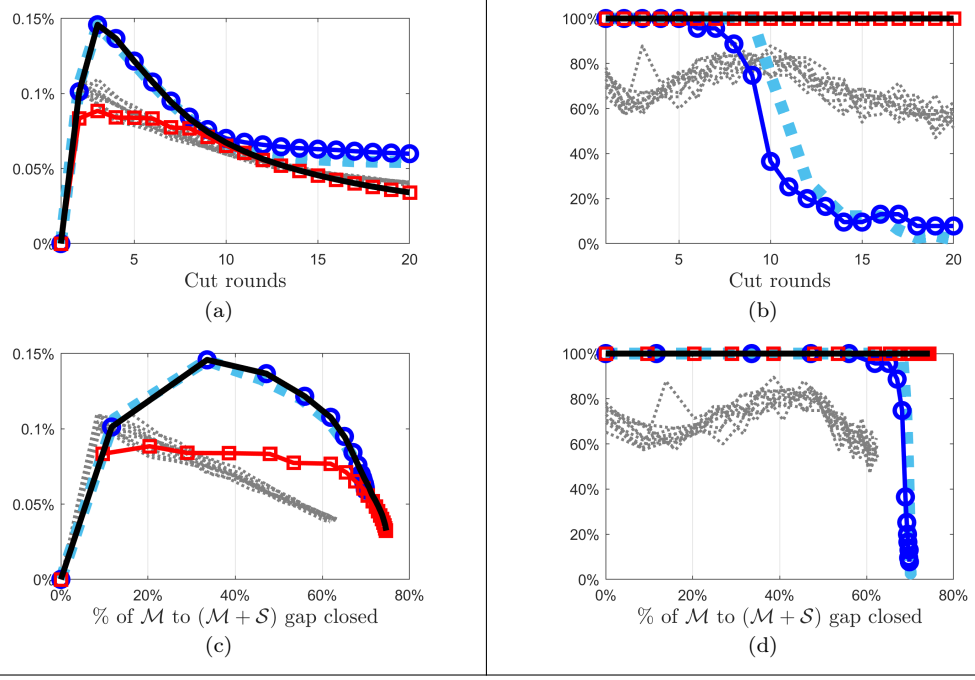
Observations (o4)-(o6) are all justified by Lemma 3.2.1(iii), where for any semidefinite ρ subproblem, more additional cut classes restrict the convex set $\mathcal{C}$ referenced in the lemma. In particular, Lemma 3.2.1(iii) justifies (o6) since a higher instance density implies a larger selection of

Strong (valid) cuts (by optimality sel.)

• comparison via y-axis % of $\mathcal{M}$ to $(\mathcal{M}+\mathcal{S})$ gap closed per number of valid cuts added (cumulative)

All valid cuts (by feasibility sel.)

• comparison via y-axis % of valid (violated) cuts identified in each selection of sub-problems



Legend:

- Combined (optimality + feasibility) selection based on $\mathcal{C}(\rho)$
- Optimality selection based on estimated $\hat{\mathcal{I}}_X(\rho)$, where only ρ with $\hat{\mathcal{I}}_X(\rho) > 0$ are considered for valid cuts
- Feasibility selection based on $\lambda_{\min}(\rho)$ finding most violated (valid) cuts
- Optimality selection based on exact $\mathcal{I}_X(\rho)$, where only ρ with $\mathcal{I}_X(\rho) > 0$ are considered for valid cuts
- Random selection

Fig. 10: Illustration on BoxQP instance *spar-100-025-1* (100 variables, 25% dense) of selecting strong (left) or valid (right) cuts when comparing selection/ordering strategies shown in the legend in Algorithm 1 to outer-approximate $\mathcal{M}+\mathcal{S}_3$ with 5% of ρ selected each round (max 115 cuts/round). Figures a-b and c-d plot the y-axis measures across rounds of cuts and percentages of gap closed, respectively.

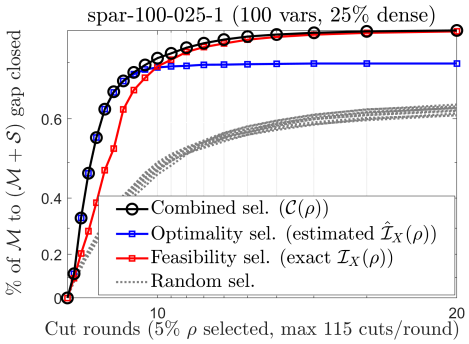


Fig. 11: Comparison of combined selection based on the $\mathcal{C}(\rho)$ measure against optimality and feasibility cut selection.

The plot shows bound convergence (root node) for outer-approximating the $\mathcal{M}+\mathcal{S}_3$ relaxation starting from $\mathcal{M}$ implemented via Table 2 with the same setup as in Fig. 9c.

Table 4: (BoxQP instances from Table 3) Comparison of % of $\mathcal{M}$ to optimality gap closed by Algorithm 1 at criteria termination, i.e. final bounds, for optimality versus feasibility selections with three setups: (i)-(ii) outer-approximate $\mathcal{M}$+$\mathcal{S}_3$ with selection sizes 5% and 10% of $|\mathcal{P}'_3|$, respectively; (iii) outer-approximate $\mathcal{M}$+$\triangle$+$\mathcal{S}_3$ (linear base $\mathcal{B} = \mathcal{M}$+$\triangle$) with selection size $10\% \cdot |\mathcal{P}'_3|$.

		5% ρ for $\mathcal{M}$+$\mathcal{S}_3$			10% ρ for $\mathcal{M}$+$\mathcal{S}_3$			10% ρ for $\mathcal{M}$+$\triangle$+$\mathcal{S}_3$			
Size	Density	Opt.	Feas.	Diff.	Opt.	Feas.	Diff.	$\mathcal{M}$+$\triangle$	Opt.	Feas.	Diff.
Small	Low	87.53	91.10	3.57	88.47	91.43	2.97	98.98	99.93	99.98	0.05
	Medium	85.44	87.79	2.35	86.28	87.86	1.58	98.16	99.86	99.88	0.01
	High	85.54	87.41	1.86	86.24	87.44	1.20	98.63	99.67	99.69	0.02
Medium	Low	79.22	84.01	4.79	80.71	84.74	4.03	99.75	99.98	100.00	0.01
	Medium	78.22	80.50	2.28	78.97	80.59	1.62	97.73	98.54	98.54	0.00
	High	72.18	73.27	1.09	72.72	73.28	0.56	92.72	94.50	94.50	0.01
Large	Low	73.27	76.81	3.54	74.43	77.45	3.03	98.81	99.50	99.53	0.03
	Medium	75.50	77.50	2.00	76.26	77.55	1.29	97.14	98.00	98.01	0.01
	High	70.12	70.97	0.85	70.54	70.99	0.45	90.06	91.36	91.35	-0.01
Jumbo	Low	71.62	74.71	3.10	72.42	74.99	2.57	97.58	97.88	97.90	0.01
	Medium	70.38	71.76	1.37	70.94	71.78	0.84	91.51	92.09	92.10	0.01
	High	66.64	67.36	0.73	66.77	67.37	0.60	85.93	86.26	86.26	0.00

external cuts/sub-problems outer-approximating a more restricted $\mathcal{S}_n$ relaxation (Lemma 3.1.5), in turn implying a more restricted convex set $\mathcal{C}$ for any ρ sub-problem.

These observations lead to the Table 2 conclusions. Cut selection via optimality converges faster than via feasibility at the cost of a final bounds gap (bridged by combined selection). But this gap can vanish after adding other cut classes, typically incorporated in Branch&Cut frameworks. In other words, when mixing different cut classes for lower bounding, only strong, fast-converging, low-dimensional semidefinite cuts are needed rather than all valid ones!

5.3 Sparsity of cutting planes

5.3.1 Dimensionality or sparsity of cuts

We explore the effect of increasing dimensionality (decreasing sparsity) n of sub-problems/cuts when outer-approximating relaxations ($\mathcal{M} + \mathcal{S}_n$). For cut dimensionality $n \in \{3, 4, 5\}$ and optimality/feasibility cut selection on BoxQP instances, the aggregated results reported in Table 5 allow to derive the following observations:

(o7) The final bound reached by feasibility-based cuts quickly tails off with increasing n. Specifically, the gap between $n = 5$ and $n = 4$ is much less marked than the one between $n = 4$ and $n = 3$, regardless of the instance size and the number of sub-problems $|\mathcal{P}'_n|$. This corresponds to Remark 3.1.3 that low-dimensional linear outer-approximations offer a good bound/complexity trade-off.

(o8) With increasing n, the final bound reached by optimality-based cuts decreases on low density instances, while it increases on medium and high density instances. This observed behavior seems independent of the size of the instances.

(o9) On low density instances, the gap between between feasibility- and optimality-based cuts increases with n. This is consistent with Lemma 3.2.1(iv), i.e., given any two sub-problems $\rho_1 \subset \rho_2$, the PSD completion in $P^{\tilde{x}}(\rho_1)$ has a larger fixing size than in $P^{\tilde{x}}(\rho_2)$ with extra fixed variables $X_{\rho_2 \setminus \rho_1}$, implying the lower-dim. ρ_1 sub-problem is better outer-approximated than ρ_2 via optimality selection.

However, the picture is less clear on medium and high density instances. Indeed, on medium density instances the gap between feasibility- and optimality-based cuts increases from $n = 3$ to $n = 4$, but decreases from $n = 4$ to $n = 5$. This is not unexpected because: (i) the bound reached by feasibility cuts tails off from $n = 4$ to $n = 5$ (see (o7)) and (ii) the gap between feasibility and optimality cuts decreases when the density of the instances increases (see (o6)).

(o10) The number of sub-problems $|\mathcal{P}'_n|$ drastically increases with n on high density instances, and becomes intractable on large and jumbo size models due to the excessive computational costs.

From observations (o7)-(o10) the following implications arise:

(i) The limitation of training an estimator (Section 4) only for low-dim. sub-problems is not an impediment for using optimality strong cut selection at its most effective;
(ii) The intuition of sparsifying cuts [85] is valid, with bound improvements that tail off with decreasing cut sparsity (increasing n) - this justifies our bottom-up approach to cut dimensionality n, contrary to [85, 95];
(iii) The MIP intuition [35–37] that sparse cuts are strong extends to a continuous setting.

Finally, observations (o1)-(o3), (o6)-(o10) and the combined selection strategy $\mathcal{C}(\rho)$ introduced in Section 5.2 inform Heuristic 2, tailored to instance size/sparsity for obtaining both strong convergence and final bounds while managing computational cost. A specific variant of such heuristic is used in the computational experiments reported in Section 5.4.

Heuristic 2: Cut selection from $\mathcal{S}_n$ for Algorithm 1 (cut sel. strategies and cut dim. n) based on instance size N and sparsity.

1 Set $n \in \{3, 4, 5\}$ and find $\mathcal{P}'_n$, keeping 'reasonable' cut selection and/or LP solve times (depending on n, N, $|\mathcal{P}'_n|$ and on the particular requirements of a solver);
2 Choose cut selection strategy:
(1) *Sparse instance with no large high-density clusters:*
(i) If cut selection cost not prohibitive (*up to jumbo instances*), use combined selection.
(ii) Otherwise (*very large instances*), use a combined approach with optimality selection on low-dim. sub-problems and feasibility selection on higher-dim. sub-problems.
(2) *Dense instance*: Use optimality selection.
(3) *Sparse instance with identified large high-density clusters:*
Apply (2) on the cluster sub-problems and (1) on the remaining sub-problems.

5.3.2 Incorporating chordal extensions

We explore the effect on final bounds for cut selection strategies using chordal extensions of an instance sparsity graph. For $n = 3$, we compare selecting cuts based on set $\overline{\mathcal{P}_3}$ (Lemma 3.1.4) and (chordal) relaxation $\mathcal{S}(\overline{\mathcal{P}_3})$ versus based on set $\mathcal{P}'_3$ (Lemma 3.1.5) and relaxation $\mathcal{S}_3$ for BoxQP instances. Table 6 shows that, by considering chordal extensions, feasibility final bounds (fourth column) marginally improve, although not in all cases, but optimality final bounds (in the third column) tend instead to decrease, especially on large and jumbo instances. The latter result is not surprising, taking into account that chordal extensions to uniformly (not chordal) sparse BoxQP instances add many additional sub-problems and cuts (see $|\overline{\mathcal{P}_3}|$) with variables not participating in the objective. Selecting from $\overline{\mathcal{P}_3}$ therefore involves many cuts offering little

Table 5: (BoxQP instances from Table 3) Comparison of % of $\mathcal{M}$ to optimality gap closed by Algorithm 1 at criteria termination, i.e. final bounds, for optimality and feasibility selections outer-approximating ($\mathcal{M}+\mathcal{S}_n$) with selection size $10\% \cdot |\mathcal{P}'_n|$ for all $n \in \{3,4,5\}$. For missing entries we run out of memory within 20 rounds of cuts.

		$\mathcal{M}$+$\mathcal{S}_3$		$\mathcal{M}$+$\mathcal{S}_4$		$\mathcal{M}$+$\mathcal{S}_5$		# sub-problems ρ		
Size	Density	Opt.	Feas.	Opt.	Feas.	Opt.	Feas.	$\lvert\mathcal{P}'_3\rvert$	$\lvert\mathcal{P}'_4\rvert$	$\lvert\mathcal{P}'_5\rvert$
Small	Low	88.47	91.43	84.38	94.53	83.99	95.00	389	248	206
	Medium	86.28	87.86	90.90	95.47	92.09	96.45	1,250	1,716	1,112
	High	86.24	87.44	95.41	98.18	97.66	99.46	3,220	14,460	42,244
Medium	Low	80.71	84.74	76.26	87.89	75.31	88.59	616	416	357
	Medium	78.97	80.59	87.08	91.75	89.56	93.60	3,799	6,131	4,033
	High	72.72	73.28	88.66	92.33	92.22	96.28	21,540	142,302	543,633
Large	Low	74.43	77.45	68.99	81.04	68.43	81.98	1,473	896	805
	Medium	76.26	77.55	87.95	91.40	91.17	94.00	11,055	25,595	23,203
	High	70.54	70.99	86.62	90.74	90.76	-	39,022	316,784	1,490,876
Jumbo	Low	72.42	74.99	68.85	80.64	67.94	81.46	3,530	2,195	1,821
	Medium	70.94	71.78	84.92	88.58	88.85	91.85	27,441	90,216	116,039
	High	66.77	67.37	84.10	87.76	-	-	85,339	1,013,518	4,803,170

Table 6: (BoxQP instances from Table 3) Comparison of % of $\mathcal{M}$ to optimality gap closed by Algorithm 1 at criteria termination, i.e. final bounds, for outer-approximating: (i) dense-only relaxation ($\mathcal{M}+\mathcal{S}_3$); (ii) relaxation ($\mathcal{M}+\mathcal{S}(\overline{\mathcal{P}_3})$) based on chordal extensions; (iii) relaxation ($\mathcal{M}+\mathcal{S}(\overline{\mathcal{P}^*_3})$) restricting zero-coefficient variables in chordal extensions. Algorithm 1 is used with selection size 10% of $|\mathcal{P}'_3|$, $|\overline{\mathcal{P}_3}|$ and $|\overline{\mathcal{P}^*_3}|$, respectively; the cardinality of the previous three sets is compared in the last three columns.

		$\mathcal{M}$+$\mathcal{S}_3$		$\mathcal{M}$+$\mathcal{S}(\overline{\mathcal{P}_3})$		$\mathcal{M}$+$\mathcal{S}(\overline{\mathcal{P}^*_3})$	# sub-problems ρ (thousands)		
Size	Density	Opt.	Feas.	Opt.	Feas.	Opt.	$\lvert\mathcal{P}'_3\rvert$	$\lvert\overline{\mathcal{P}_3}\rvert$	$\lvert\overline{\mathcal{P}^*_3}\rvert$
Small	Low	88.47	91.43	88.17	92.24	89.41	389	3,995	1,654
	Medium	86.28	87.86	86.50	87.91	86.84	1,250	5,024	3,413
	High	86.24	87.44	86.37	87.44	86.39	3,220	4,801	4,499
Medium	Low	80.71	84.74	81.87	87.26	83.77	616	11,531	3,402
	Medium	78.97	80.59	79.06	80.65	79.65	3,799	22,509	12,763
	High	72.72	73.28	72.96	73.28	72.97	21,540	49,462	41,965
Large	Low	74.43	77.45	73.87	79.47	77.09	1,473	46,935	10,252
	Medium	76.26	77.55	75.87	77.55	76.91	11,055	74,379	39,504
	High	70.54	70.99	70.40	70.97	70.54	39,022	90,061	76,374
Jumbo	Low	72.42	74.99	71.35	76.10	74.57	3,530	122,106	25,829
	Medium	70.94	71.78	69.85	71.73	71.01	27,441	181,912	97,975
	High	66.77	67.37	66.61	67.35	66.70	85,339	211,684	179,703

objective improvement. To alleviate this problem, we restrict most zero-coefficient variables from appearing in $\overline{\mathcal{P}_3}$ sub-problems by defining, for example,

$$\overline{\mathcal{P}^*_3} := \{\rho \subseteq \rho' \in \overline{\mathcal{P}_3} \mid \nexists Q_{ij} \neq 0 \ \forall i \in \rho,\ j \in \rho' \setminus \{i\}\}.$$

Table 6 shows bound improvements for optimality cut selection by outer-approximating ($\mathcal{M}+\mathcal{S}(\overline{\mathcal{P}^*_3})$), especially for sparser instances where the number of sub-problems is significantly reduced. However, combined selection strategy $\mathcal{C}(\rho)$ converges to higher final bounds for sparser instances then optimality selection alone. Furthermore, for BoxQP instances with low to medium density, increasing cut dimensionality improves bounds more with less of an increase in sub-problem numbers.

Therefore, chordal extensions, with(out) restrictions such as $\overline{\mathcal{P}^*_3}$, are not suited to uniform sparsity (such as in BoxQP), but may offer advantages [33] for other instances with (nearly [23]) chordal sparsity patterns.

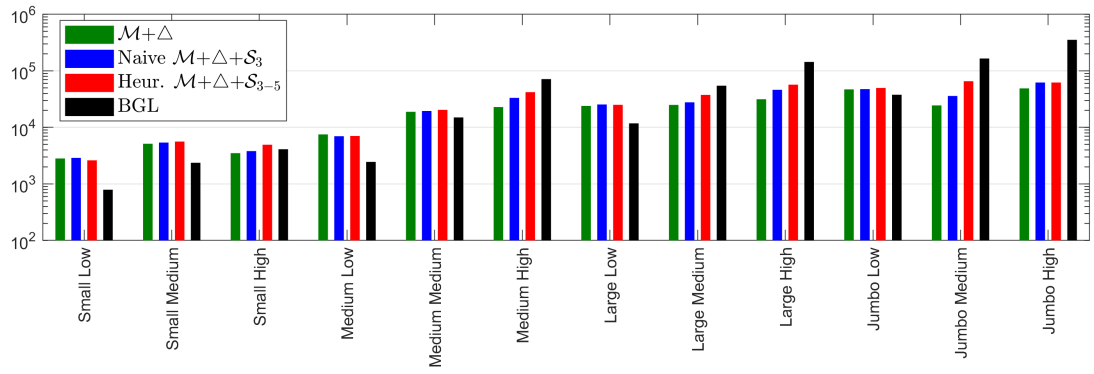


Fig. 12: Number of cuts generated for all BoxQP instances (Table 3) to obtain Table 7 bounds.

5.4 Mixing with other linear cut classes

This section shows that, even in conjunction with other cut classes for QP as base relaxations, our outer-approximations (see Remark 5.4.1) are: (i) complementary by further improving bounds (Table 7), (ii) computationally light by minimal running time overhead (Fig. 13) and (iii) light in numbers of extra generated cutting planes (Fig. 12).

Remark 5.4.1 *(Relaxations used in the results in Table 7, Figs. 12 and 13)*

- $\mathcal{M} + \triangle$ is used as a linear base relaxation in Algorithm 1 and a proxy for the relaxation $\mathcal{M} + 1/2$ [18]. The $\mathcal{M}$+△ bounds in Table 7 are stronger than those of $\mathcal{M} + 1/2$ [18] which omits $\mathcal{M}$ cutting planes $(\forall i \in V)\ X_{ii} \geq 2x_i - 1$.
- The naïve $\mathcal{M}$+△+$\mathcal{S}_3$ relaxation is implemented via Algorithm 1 with optimality cut selection.
- The heuristic $\mathcal{M} + \triangle + \mathcal{S}_{3-5}$ relaxation is implemented via Algorithm 1 with choices based on Heuristic 2, outer-approximating:
  - $\mathcal{M} + \triangle + \mathcal{S}_5$ with combined cut selection (Definition 5.2.1) for low and medium density BoxQP instances;
  - $\mathcal{M} + \triangle + \mathcal{S}_4$ with optimality cut selection for high density, <jumbo size BoxQP instances;
  - $\mathcal{M} + \triangle + \mathcal{S}_3$ with optimality cut selection for high density, jumbo size BoxQP instances.
- $\mathcal{M}^2 + 1/2$ [18] (denoted by BGL) is a very good proxy of the current CPLEX linear relaxation for BoxQP instances. Since BGL relaxation models explicitly quadratic terms x_i^2 $(\forall i \in V$ s.t. $Q_{ii} \geq 0)$, its bounds are comparable to $\mathcal{M} + 1/2 + \mathcal{S}_1$, and is therefore a useful benchmark for both naïve $\mathcal{M} + \triangle + \mathcal{S}_3$ and heuristic $\mathcal{M} + \triangle + \mathcal{S}_{3-5}$.

□

More specifically, in Table 7, improving upon both BGL and the base linear relaxation $\mathcal{M}$+△ bounds, both the naïve and heuristic outer-approximations offer bounds that for sparser, non-jumbo size instances beat $\mathcal{S}$ bounds and are competitive with expensive $\mathcal{M} + \mathcal{S}$ bounds [5]. The heuristic $\mathcal{M} + \triangle + \mathcal{S}_{3-5}$ relaxation is slightly heavier than the naïve $\mathcal{M} + \triangle + \mathcal{S}_3$ one, showing further improved bounds in exchange for extra running time and cuts (Figs. 12 and 13). However, both naïve or heuristic relaxations can be used in practice, as they both involve cut numbers similar to BGL (Fig. 12) and minimal extra running time to the base relaxation $\mathcal{M} + \triangle$ (or potentially $\mathcal{M} + 1/2$ or BGL).

Table 7: (BoxQP instances from Table 3) Comparison of % of $\mathcal{M}$ to optimality gap closed by Algorithm 1 at criteria termination, i.e. final bounds, using a strong base relaxation $\mathcal{M}+\triangle$ against the SDP, SDP+linear and state-of-the-art BGL [18] relaxations. The relaxations $\mathcal{M}+\triangle+\mathcal{S}_3$ and $\mathcal{M}+\triangle+\mathcal{S}_{3-5}$ are outer-approximated via Algorithm 1 as described in Remark 5.4.1 with selection size 10% of all sub-problems at each cut round.

		SDP	SDP+linear		CPLEX	Base	Naïve $\mathcal{M}+\triangle+\mathcal{S}_3$			Heur. $\mathcal{M}+\triangle+\mathcal{S}_{3-5}$		
Size	Density	$\mathcal{S}$	$\mathcal{S}^{\geq}$	$\mathcal{M}+\mathcal{S}$	BGL [18]	$\mathcal{M}+\triangle$		Diff. BGL	Diff. $\mathcal{M}+\triangle$		Diff. BGL	Diff. $\mathcal{M}+\triangle$
Small	Low	80.65	99.11	99.29	99.51	98.98	99.93	0.42	0.94	99.98	0.47	0.99
	Medium	89.79	99.40	99.46	99.29	98.16	99.86	0.57	1.70	99.96	0.67	1.80
	High	94.15	99.76	99.80	99.13	98.48	99.60	0.47	1.12	99.73	0.60	1.26
Medium	Low	85.85	99.33	99.55	99.90	99.75	99.98	0.08	0.23	100.00	0.10	0.24
	Medium	93.00	98.77	98.86	98.01	97.73	98.54	0.53	0.81	99.03	1.02	1.30
	High	95.68	99.24	99.31	93.52	92.72	94.50	0.98	1.77	95.94	2.42	3.22
Large	Low	88.61	98.20	98.65	98.28	98.81	99.50	1.22	0.69	99.58	1.30	0.77
	Medium	94.96	99.05	99.25	97.48	97.14	98.00	0.52	0.86	98.74	1.26	1.59
	High	96.34	99.14	99.29	90.60	90.06	91.36	0.76	1.29	93.01	2.41	2.95
Jumbo	Low	92.90	98.35	98.84	96.28	96.92	97.88	1.60	0.96	98.07	1.79	1.15
	Medium	95.25	98.60	98.82	91.42	90.99	92.09	0.67	1.09	93.79	2.37	2.80
	High	96.67	98.96	99.16	85.68	85.36	86.26	0.58	0.90	86.26	0.58	0.90

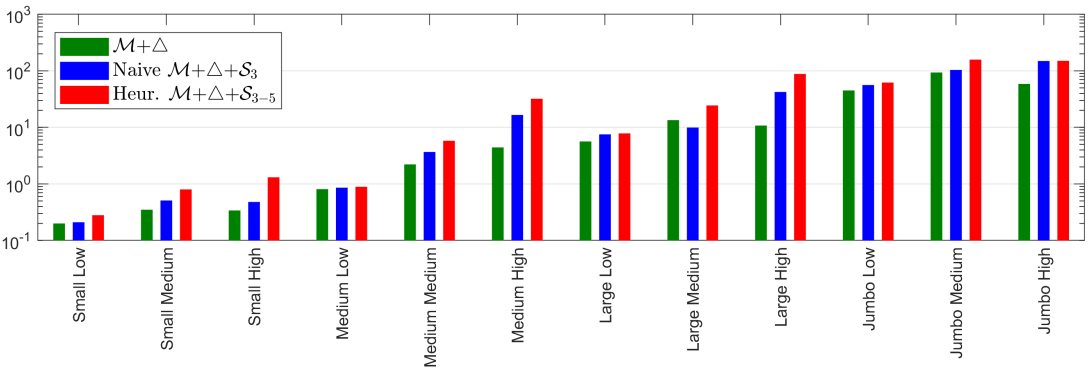


Fig. 13: Running time (in seconds) for all BoxQP instances (Table 3) to obtain Table 7 bounds.

6 Extending strong semidefinite cuts to QCQP problems

This section extends our optimality-based cut selection to QCQP by relating nonconvex quadratic constraints to their optimality contribution.

A natural way to relate constraints to optimality in the QCQP extended formulation is via the Lagrangian function. The QCQP Lagrangian is a proxy for the QP objective and fixing $x = \tilde{x}$ as in Corollary 3.2.2 reduces it to a quadratic function of variables X and multipliers $\beta^{(m)} \in \mathbb{R}^m_+$,

$$\mathcal{L}(X, \beta|\tilde{x}) := Q^{(0)} \bullet X + \sum_{k=1}^{m} \beta_k^{(m)} (Q^{(k)} \bullet X) = \sum_{k=0}^{m} \beta_k (Q^{(k)} \bullet X),$$

where, for notation convenience, the vector $\beta \in \mathbb{R}^{m+1}_+$ extends $\beta^{(m)}$ with the element 1 to accommodate the objective in the summation. Semidefinite decompositions $\mathcal{P}_n$, $\overline{\mathcal{P}_n}$, $\mathcal{P}'_n \subseteq \mathcal{P}$, equivalent to those in Section 3.1, are enabled by extending the sparsity pattern $G(V, E)$ to include edges E from the constraints. Applying Lemma 3.2.3 to obtain implied optimality-based cuts from $\mathcal{L}(X, \beta|\tilde{x})$ given a decomposition $\mathcal{F} \subseteq \mathcal{P}$ results in low-dimensional sub-problems $\rho \in \mathcal{F}$

similar to $P^{\tilde{x}}(\rho)$, but with a bilinear objective in variables X_ρ and β, i.e.

$$\mathcal{L}_0(\rho) := \mathcal{L}(X^*_\rho, \beta^* | \tilde{x}_\rho) = \min_{X_\rho,\ \beta_\rho} \left\{ \sum_{k \in \Theta_\rho} \beta_k Q^{(k)}_\rho \bullet X_\rho \;\middle|\; \begin{bmatrix} 1 & \tilde{x}^T_\rho \\ \tilde{x}_\rho & X_\rho \end{bmatrix} \succeq 0,\ X_{ii} \leq \tilde{x}_i\ \forall i \in \rho \right\}$$

where $\Theta_\rho := \{k \in 0, \ldots, m \mid \text{the incidence graph of } Q^{(k)}_\rho \text{ has a clique with} \geq 2 \text{ vertices in } \rho\}$.

Remark 6.1 *For a given $\rho \in \mathcal{F}$, Θ_ρ omits constraints k where $Q^{(k)}_\rho$ only offers a one vertex clique, since such cliques are already covered by the SOCP relaxation $\mathcal{K}$.* □

While possible to train a neural net to estimate $\mathcal{L}_0(\rho)$, it involves learning a collection of nonconvex surfaces corresponding to a SDP problem with bilinear objective. This implies the training, architecture and evaluation for such a neural net (or any other estimator) are more complex and costly than for the neural nets developed in Section 4 which learn a collection of convex surfaces.

However, by fixing $\beta = \tilde{\beta}$, we can construct linear ρ sub-problems such as,

$$\mathcal{L}_1(\rho) := \mathcal{L}(X^*_\rho | \tilde{\beta}, \tilde{x}_\rho) = \min_{X_\rho} \left\{ \sum_{k \in \Theta_\rho} \tilde{\beta}_k Q^{(k)}_\rho \bullet X_\rho \;\middle|\; \begin{bmatrix} 1 & \tilde{x}^T_\rho \\ \tilde{x}_\rho & X_\rho \end{bmatrix} \succeq 0,\ X_{ii} \leq \tilde{x}_i\ \forall i \in \rho \right\},$$

$$\mathcal{L}_2(\rho) := \sum_{k \in \Theta_\rho} \tilde{\beta}_k \min_{X_\rho} \left\{ Q^{(k)}_\rho \bullet X_\rho \;\middle|\; (\tilde{x}_\rho, X_\rho) \in \mathcal{S}^{\tilde{x}}(\rho) \right\}.$$

Furthermore, given an incumbent solution point $(\tilde{X}, \tilde{\beta}, \tilde{x})$, $\forall \rho \in \mathcal{F}$ an expected objective improvement measure for optimality selection such as $\hat{\mathcal{I}}_X(\rho)$ in Section 3.3 can be built for each $\mathcal{L}_i(\rho)$ $\forall i \in \{0, 1, 2\}$ in the form

$$\left(\mathcal{L}_i(\rho) - \mathcal{L}(\tilde{X}, \tilde{\beta}) \approx\right) \quad \hat{\mathcal{L}}_i(\rho) - \mathcal{L}(\tilde{X}_\rho, \tilde{\beta}, \tilde{x}_\rho) = \hat{\mathcal{L}}_i(\rho) - \sum_{k \in \Theta_\rho} \beta_k (Q^{(k)}_\rho \bullet \tilde{X}_\rho), \qquad (\hat{\mathcal{I}}^{(i)}_X(\rho))$$

where $\hat{\mathcal{L}}_i(\rho)$ $\forall i \in \{0, 1, 2\}$ are estimated values. Due to fixing $\tilde{\beta}$, we have $\hat{\mathcal{L}}_1(\rho) \geq \hat{\mathcal{L}}_0(\rho)$ and $\hat{\mathcal{I}}^{(1)}_X(\rho) \geq \hat{\mathcal{I}}^{(0)}_X(\rho)$. Contrasting the estimated optimality measures in terms of bound strength and computational complexity:

- $\hat{\mathcal{I}}^{(0)}_X(\rho)$ is the most expensive estimator leading to worse final bounds than cut selection based on $\hat{\mathcal{I}}^{(1)}_X(\rho)$ ($\hat{\mathcal{I}}^{(1)}_X(\rho) \geq \hat{\mathcal{I}}^{(0)}_X(\rho)$); $\hat{\mathcal{L}}_0(\rho)$ requires one evaluation of a neural net for semidefinite sub-problems with bilinear objective.
- $\hat{\mathcal{I}}^{(1)}_X(\rho)$ **(aggregating all constraints)** is the cheapest estimator since $\hat{\mathcal{L}}_1(\rho)$ requires one evaluation of a neural net for semidefinite sub-problems with linear objective, built in Section 4.
- $\hat{\mathcal{I}}^{(2)}_X(\rho)$ **(constraint-by-constraint)** is more expensive than $\hat{\mathcal{I}}^{(1)}_X(\rho)$, since $\hat{\mathcal{L}}_2(\rho)$ requires $|\Theta(\rho)|$ evaluations of a neural net from Section 4, where $|\Theta(\rho)|$ is high if the variable structures in ρ are heavily repeated though many constraints.

In addition to avoiding extra neural net complexity and reusing those from Section 4, the previous two measures also do not use the multipliers β which are not explicit variables that can be included in a generated cut. However, due to fixing $\tilde{\beta}$ at every iteration, an optimality-only selection based on Algorithm 1 with adapted optimality measures $\hat{\mathcal{I}}^{(1)}_X$, $\hat{\mathcal{I}}^{(2)}_X$ is likely to not converge on some instances. This is visible in Figs. 14c–14d tests on a QCQP power flow

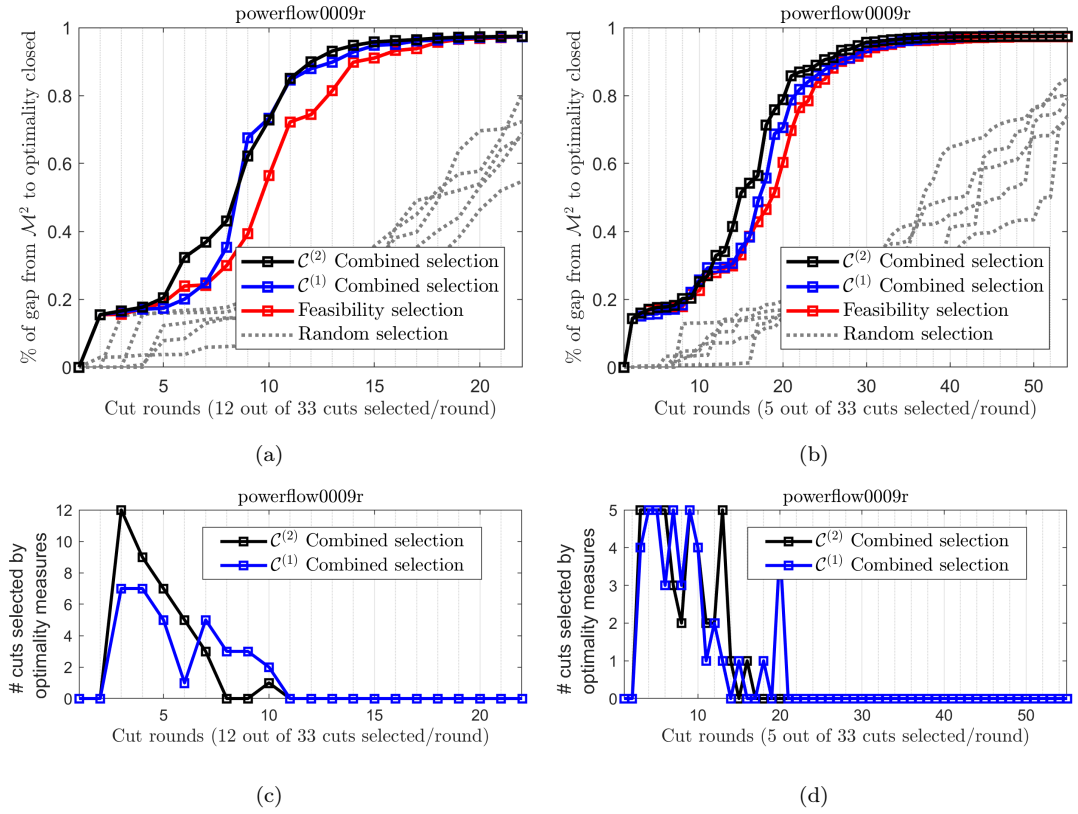


Fig. 14: (MINLPLib [102, 103] optimal power flow instance *powerflow009r*, rectangular formulation) Comparison of % of $\mathcal{M}^2$ to optimality gap closed by Algorithm 1 across cut rounds for combined selections $\mathcal{C}^{(i)}(\rho)$ versus feasibility selection outer-approximating $(\mathcal{M}^2 + \mathcal{S}_2)$.

optimization instance, where no optimality selected cuts were found on the first iteration. The underlying justification is that certain β fixings can nullify any constraints associated to positive optimality measures.

To ensure convergence in all cases and still exploit the strong cuts selected by optimality, we build a combined (optimality and feasibility) ordering similar to $\mathcal{C}(\rho)$ in Section 5.2,

$$i \in \{0,1,2\},\ \forall \rho \in \mathcal{F},\ \mathcal{C}^{(i)}(\rho) = \begin{cases} \hat{\mathcal{I}}_X^{(i)}(\rho) + M, \text{ if } \hat{\mathcal{I}}_X^{(i)}(\rho) > 0 \text{ and } \lambda_{\min}(\rho) < 0, \\ -\lambda_{\min}(\rho) \text{ otherwise,} \end{cases} \qquad (\mathcal{C}^{(i)}(\rho))$$

where M is an arbitrary large positive number.

Fig. 14 implements combined cut selection strategies $\mathcal{C}^{(2)}$ and $\mathcal{C}^{(1)}$ (via Algorithm 1 with adapted measures $\hat{\mathcal{I}}_X^{(2)}$, $\hat{\mathcal{I}}_X^{(1)}$ on decomposition $\mathcal{F} = \mathcal{P}_2'$) and compares them against feasibility cut selection for a small optimal power flow QCQP instance. Selecting strong cuts via optimality selection in both $\mathcal{C}^{(2)}$ and $\mathcal{C}^{(1)}$ strategies clearly improves the convergence compared to feasibility selection of valid cuts. This result is promising for many other QCQP problem classes, where the trade-off between $\mathcal{C}^{(2)}$ and $\mathcal{C}^{(1)}$ selection strategies depends on the specific repetition of variable structures in multiple constraints.

Therefore, this section clearly shows the potential of using strong cuts selected via optimality measures in the context of QCQP, with applications to a wide array of practical instances.

7 Conclusion

Cuts enforcing a semidefiniteness constraint, e.g. the eigenvalue cuts considered in this paper, are well-known in the literature. But our work, for both QP and QCQP, shows that, inside the set of all possible *valid* cuts, there exists a set of fewer, *strong* cuts that tightly outer-approximate the semidefinite constraint(s). Most of this paper discusses quadratic programming, but we show our ideas also extend to quadratic constraints. This paper began with four desirata and we motivate their fulfillment as follows:

1. **Easy integration into current technology**, e.g. we develop linear cutting planes with few nonzeros in each row. Lemma 3.2.1(ii)-(iii) motivate that complementary cutting surfaces, e.g. polyhedral cuts, work in synergy with the cuts we propose. Section 5.4 discusses our computational experience with QP instances.
2. **Computationally light relaxations**, e.g. Section 5.3.1 shows the effectiveness on QP of integrating computationally cheap (low-dimensional) cutting planes (Remark 3.1.3) with a neural network estimator (Section 4). The neural network takes the cut selection offline. Section 5.3 discusses the engineering trade-offs associated with cut dimensionality and chordal extensions.
3. **Relatively few cuts**, e.g. our search for cutting planes that will most improve the objective value leads to the performance in Fig. 9 and Fig. 13. We expect relatively few cuts because of our emphasis on finding *strong* cuts within the set of *valid* cuts.
4. **Cutting surface generation should converge**, e.g. our combined feasibility/optimality strategy in Table 2 incorporates all valid cutting planes while favoring the strong cuts. Figure 10 and Figure 11 show that we have maintained convergence without giving up the fast initial convergence.

With respect to previous work, our proposed cuts are (i) sparser with respect to the number of nonzeros in the row and (ii) explicitly selected to improve the objective. A neural network estimator is key to this cut selection strategy. Building the estimator was a challenge comprising (i) estimator output choice, e.g. the metric for cut selection, (ii) data generation, i.e. sampling uniformly, (iii) model choice, e.g. variance versus bias trade-off, (iv) feature selection, i.e. understanding the problem space, (v) problem decomposition, i.e. to escape the curse of dimensionality. Many have observed that estimators may accurately guess good decisions to make within a Branch&Cut framework [2, 15, 19, 52, 57, 62, 64, 68, 80, 82], so our contributions in building this neural network estimator may be more broadly applicable. For example, we conjecture that a similar estimator could be used for cuts based on copositive optimization, e.g. with 3D relaxations [6] or for the pooling problem which is known to have strong cuts [8, 65].

References

1. T. Achterberg. SCIP: Solving constraint integer programs. *Mathematical Programming Comput.*, 1(1):1–41, 2009.
2. A. M. Alvarez, Q. Louveaux, and L. Wehenkel. A machine learning-based approximation of strong branching. *INFORMS Journal on Computing*, 29(1):185–195, 2017.
3. M. S. Andersen and L. Vandenberghe. Chompack: a python package for chordal matrix computations, 2015.

4. G. Andreello, A. Caprara, and M. Fischetti. Embedding {0, 1/2}-cuts in a branch-and-cut framework: A computational study. *INFORMS Journal on Computing*, 19(2):229–238, 2007.
5. K. M. Anstreicher. Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 43(2-3):471 – 484, 2009.
6. K. M. Anstreicher and S. Burer. Computable representations for convex hulls of low-dimensional quadratic forms. *Mathematical Programming*, 124(1-2):33–43, 2010.
7. MOSEK ApS. Mosek fusion api for python. version 8.1., 2018. URL https://docs.mosek.com/8.1/pythonfusion.pdf.
8. R. Baltean-Lugojan and R. Misener. Piecewise parametric structure in the pooling problem: from sparse strongly-polynomial solutions to NP-hardness. *Journal of Global Optimization*, 71(4):655–690, 2018.
9. R. Baltean-Lugojan and R. Misener. Implementation to select cutting planes for quadratic semidefinite outer-approximation via trained neural networks, 2018. URL https://github.com/cog-imperial/SDPCutSel-via-NN/.
10. X. Bao, N. V. Sahinidis, and M. Tawarmalani. Multiterm polyhedral relaxations for nonconvex, quadratically-constrained quadratic programs. *Optimization Methods and Software*, 24(4-5):485 – 504, 2009.
11. X. Bao, N. V. Sahinidis, and M. Tawarmalani. Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. *Mathematical Programming*, 129(1):129–157, 2011.
12. W. W. Barrett, C. R. Johnson, and R. Loewy. Critical graphs for the positive definite completion problem. *SIAM Journal on Matrix Analysis and Applications*, 20(1):117–130, 1998.
13. P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4-5):597–634, 2009.
14. A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*, volume 2. Siam, 2001.
15. Y. Bengio, A. Lodi, and A. Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon. *arXiv preprint arXiv:1811.06128*, 2018.
16. A. Billionnet and S. Elloumi. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Mathematical Programming*, 109(1):55–68, 2007.
17. P. Bonami, A. Lodi, J. Schweiger, and A. Tramontani. Solving standard quadratic programming by cutting planes. Technical report, Polytechnique Montréal, 2016.
18. P. Bonami, O. Günlük, and J. Linderoth. Globally solving nonconvex quadratic programming problems with box constraints via integer programming methods. *Mathematical Programming Computation*, 10(3): 333–382, 2018.
19. P. Bonami, A. Lodi, and G. Zarpellon. Learning a classification of mixed-integer quadratic programming problems. In *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 595–604. Springer, 2018.
20. B. Borchers and J. E. Mitchell. An improved branch and bound algorithm for mixed integer nonlinear programs. *Computers & Operations Research*, 21(4):359–367, 1994.
21. F. Boukouvala, R. Misener, and C. A. Floudas. Global optimization advances in mixed-integer nonlinear programming, MINLP, and constrained derivative-free optimization, CDFO. *Eur. J. Oper. Res.*, 252(3):701 – 727, 2016.
22. S. Boyd and L. Vandenberghe. Localization and cutting-plane methods. *From Stanford EE 364b lecture notes*, 2007.
23. A. Brandstädt and C. T. Hoàng. On clique separators, nearly chordal graphs, and the maximum weight stable set problem. *Theoretical Computer Science*, 389(1-2):295–306, 2007.
24. C. Buchheim and E. Traversi. Quadratic combinatorial optimization using separable underestimators. *INFORMS Journal on Computing*, 30(3):424–437, 2018.
25. C. Buchheim and A. Wiegele. Semidefinite relaxations for non-convex quadratic mixed-integer programming. *Mathematical Programming*, 141(1):435–452, 2013.
26. S. Burer. Optimizing a polyhedral-semidefinite relaxation of completely positive programs. *Mathematical Programming Computation*, 2(1):1–19, 2010.
27. S. Burer and D. Vandenbussche. A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. *Mathematical Programming*, 113(2):259–282, 2008.
28. S. Burer and D. Vandenbussche. Globally solving box-constrained nonconvex quadratic programs with semidefinite-based finite branch-and-bound. *Computational Optimization and Applications*, 43(2):181–195, 2009.
29. H. Chel, A. Majumder, and D. Nandi. Scaled conjugate gradient algorithm in neural network based approach for handwritten text recognition. In D. Nagamalai, E. Renault, and M. Dhanuskodi, editors, *Trends in Computer Science, Engineering and Information Technology*, pages 196–210, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
30. J. Chen and S. Burer. Globally solving nonconvex quadratic programming problems via completely positive programming. *Mathematical Programming Comput.*, 4(1):33–52, 2012.

31. J. Y. Choi, L. M. DeAlba, L. Hogben, M. S. Maxwell, and A. Wangsness. The p0-matrix completion problem. *Electronic Journal of Linear Algebra*, 9(1-20):99–100, 2002.
32. S. Coniglio and M. Tieves. On the generation of cutting planes which maximize the bound improvement. In *International Symposium on Experimental Algorithms*, pages 97–109. Springer, 2015.
33. J. Dahl, L. Vandenberghe, and V. Roychowdhury. Covariance selection for nonchordal graphs via chordal embedding. *Optimization Methods & Software*, 23(4):501–520, 2008.
34. S. S. Dey and M. Molinaro. Theoretical challenges towards cutting-plane selection. *Mathematical Programming*, pages 1–30, 2018.
35. S. S. Dey, M. Molinaro, and Q. Wang. How good are sparse cutting-planes? In *International Conference on Integer Programming and Combinatorial Optimization*, pages 261–272. Springer, 2014.
36. S. S. Dey, A. Iroume, and M. Molinaro. Some lower bounds on sparse outer approximations of polytopes. *Operations Research Letters*, 43(3):323–328, 2015.
37. S. S. Dey, M. Molinaro, and Q. Wang. Analysis of sparse cutting planes for sparse MILPs with applications to stochastic MILPs. *Mathematics of Operations Research*, 43(1):304–332, 2017.
38. H. Dong. Relaxing nonconvex quadratic functions by multiple adaptive diagonal perturbations. *SIAM Journal on Optimization*, 26(3):1962–1985, 2016.
39. H. Dong and N. Krislock. Semidefinite approaches for MIQCP: Convex relaxations and practical methods. In B. Defourny and T. Terlaky, editors, *Modeling and Optimization: Theory and Applications*, pages 49–75, Cham, 2015. Springer International Publishing.
40. R. M. Dudley. Metric entropy of some classes of sets with differentiable boundaries. *Journal of Approximation Theory*, 10(3):227–236, 1974.
41. S. Elloumi and A. Lambert. Global solution of non-convex quadratically constrained quadratic programs. *Optimization Methods and Software*, 2017. doi: 10.1080/10556788.2017.1350675.
42. K. Fujisawa, M. Fukuda, M. Kojima, K. Nakata, and M. Yamashita. *SDPA-C (semidefinite Programming Algorithm-Completion Method). User's Manual-Version 6-10*. Inst. of Technology, 2004.
43. M. Fukuda, M. Kojima, K. Murota, and K. Nakata. Exploiting sparsity in semidefinite programming via matrix completion i: General framework. *SIAM Journal on Optimization*, 11(3):647–674, 2001.
44. F. Furini, E. Traversi, P. Belotti, A. Frangioni, A. Gleixner, N. Gould, L. Liberti, A. Lodi, R. Misener, H. Mittelmann, N. V. Sahinidis, S. Vigerske, and A. Wiegele. QPLIB: a library of quadratic programming instances. *Mathematical Programming Computation*, 2018. doi: 10.1007/s12532-018-0147-4.
45. T. Gally, M. E. Pfetsch, and S. Ulbrich. A framework for solving mixed-integer semidefinite programs. *Optimization Methods and Software*, 33(3):594–632, 2018.
46. K. R. Garren. Bounds for the eigenvalues of a matrix. *NASA Technical Note*, 1968.
47. X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proc. International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.
48. I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*, volume 1. MIT press Cambridge, 2016.
49. R. Grone, C. R. Johnson, E. M. Sá, and H. Wolkowicz. Positive definite completions of partial hermitian matrices. *Linear algebra and its applications*, 58:109–124, 1984.
50. P. M. Gruber. Asymptotic estimates for best and stepwise approximation of convex bodies i. *Forum Mathematicum*, 5(5):281–298, 1993.
51. M. T. Hagan and M. B. Menhaj. Training feedforward networks with the Marquardt algorithm. *IEEE transactions on Neural Networks*, 5(6):989–993, 1994.
52. H. He, H. Daume III, and J. M. Eisner. Learning to search in branch and bound algorithms. In *Advances in neural information processing systems*, pages 3293–3301, 2014.
53. L. Hogben. Graph theoretic methods for matrix completion problems. *Linear Algebra and Its Applications*, 328(1-3):161–202, 2001.
54. R. B. Holmes. On random correlation matrices. *SIAM Journal on Matrix Analysis and Applications*, 12(2): 239–272, 1991.
55. IBM CPLEX Optimizer. URL https://www.ibm.com/analytics/cplex-optimizer.
56. C. R. Johnson and B. K. Kroschel. The combinatorially symmetric P-matrix completion problem. *Electronic Journal of Linear Algebra*, 1(1):5, 1996.
57. E. B. Khalil, P. Le Bodic, L. Song, G. L. Nemhauser, and B. N. Dilkina. Learning to branch in mixed integer programming. In *AAAI*, pages 724–731, 2016.
58. S. Kim and M. Kojima. Exact solutions of some nonconvex quadratic optimization problems via SDP and SOCP relaxations. *Computational Optimization and Applications*, 26(2):143–154, 2003.
59. S. Kim, M. Kojima, and M. Yamashita. Second order cone programming relaxation of a positive semidefinite constraint. *Optimization Methods and Software*, 18(5):535–541, 2003.
60. N. Krislock, J. Malick, and F. Roupin. Improved semidefinite bounding procedure for solving max-cut problems to optimality. *Mathematical Programming*, 143(1-2):61–86, 2014.
61. Y. A. LeCun, L. Bottou, G. B. Orr, and K. R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

62. L. HS Lelis, L. Otten, and R. Dechter. Predicting the size of depth-first branch and bound search trees. In *IJCAI*, pages 594–600, 2013.
63. S. Leyffer. Integrating SQP and branch-and-bound for mixed integer nonlinear programming. *Computational optimization and applications*, 18(3):295–309, 2001.
64. A. Lodi and G. Zarpellon. On learning and branching: a survey. *TOP*, 25(2):207–236, 2017.
65. J. Luedtke, C. D'Ambrosio, J. Linderoth, and J. Schweiger. Strong convex nonlinear relaxations of the pooling problem. *arXiv preprint arXiv:1803.02955*, 2018.
66. A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. *Proc. International Conference on Machine Learning*, 30(1):3, 2013.
67. J. Malick and F. Roupin. On the bridge between combinatorial optimization and nonlinear optimization: a family of semidefinite bounds for 0–1 quadratic problems leading to quasi-Newton methods. *Mathematical Programming*, 140(1):99–124, 2013.
68. A. Marcos Alvarez, L. Wehenkel, and Q. Louveaux. Machine learning to balance the load in parallel branch-and-bound. *Technical Report, Université de Liège*, 2015.
69. G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part 1-convex underestimating problems. *Mathematical Programming*, 10(1):147 – 175, 1976.
70. F. Mezzadri. How to generate random matrices from the classical compact groups. *arXiv preprint math-ph/0609050*, 2006.
71. R. Misener and C. A. Floudas. Global optimization of mixed-integer quadratically-constrained quadratic programs (MIQCQP) through piecewise-linear and edge-concave relaxations. *Mathematical Programming B*, 136:155–182, 2012.
72. R. Misener and C. A. Floudas. Global optimization of mixed-integer models with quadratic and signomial functions: A review. *Applied and Computational Mathematics*, 11(3):317–336, 2012.
73. R. Misener and C. A. Floudas. GloMIQO: Global Mixed-Integer Quadratic Optimizer. *Journal of Global Optimization*, 57(1):3–50, 2013.
74. R. Misener and C. A. Floudas. ANTIGONE: Algorithms for coNTinuous Integer Global Optimization of Nonlinear Equations. *Journal of Global Optimization*, 59(2-3):503–526, 2014.
75. R. Misener, J. B. Smadbeck, and C. A. Floudas. Dynamically generated cutting planes for mixed-integer quadratically constrained quadratic programs and their incorporation into GloMIQO 2. *Optimization Methods and Software*, 30(1):215–249, 2015.
76. S. Mishra, R. Prusty, and P. K. Hota. Analysis of Levenberg-Marquardt and scaled conjugate gradient training algorithms for artificial neural network based LS and MMSE estimated channel equalizers. In *Man and Machine Interfacing (MAMI), 2015 International Conference on*, pages 1–7. IEEE, 2015.
77. M. Mittelbach, B. Matthiesen, and E. A. Jorswieck. Sampling uniformly from the set of positive definite matrices with trace constraint. *IEEE Transactions on Signal Processing*, 60(5):2167–2179, 2012.
78. M. F. Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, 6(4): 525–533, 1993.
79. K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota. Exploiting sparsity in semidefinite programming via matrix completion ii: Implementation and numerical results. *Mathematical Programming*, 95 (2):303–327, 2003.
80. G. Nannicini, P. Belotti, J. Lee, J. Linderoth, F. Margot, and A. Wächter. A probing algorithm for minlp with failure prediction by svm. In *International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems*, pages 154–169. Springer, 2011.
81. T. E. Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9(3), 2007.
82. L. Otten and R. Dechter. And/or branch-and-bound on a computational grid. *Journal of Artificial Intelligence Research*, 59:351–435, 2017.
83. M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1):60–100, 1991.
84. R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.
85. A. Qualizza, P. Belotti, and F. Margot. Linear programming relaxations of quadratically constrained quadratic programs. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 407–426. Springer New York, 2012.
86. F. Rendl, G. Rinaldi, and A. Wiegele. Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Mathematical Programming*, 121(2):307–335, 2010.
87. N. V. Sahinidis. BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8(2):201–205, 1996.
88. L. M. Saini and M. K. Soni. Artificial neural network-based peak load forecasting using conjugate gradient methods. *IEEE Transactions on Power Systems*, 17(3):907–912, 2002.
89. Y. Sari. Performance evaluation of the various training algorithms and network topologies in a neural-network-based inverse kinematics solution for robots. *International Journal of Advanced Robotic Systems*, 11(4):64, 2014.

90. A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: extended formulations. *Mathematical Programming*, 124(1-2):383–411, 2010.
91. A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations. *Mathematical Programming*, 130(2):359–413, 2011.
92. B. Sharma and K. Venugopalan. Comparison of neural network training functions for hematoma classification in brain CT images. *IOSR Journal of Computer Engineering*, 16:31–35, 2014.
93. H. D. Sherali and W. P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, Dordrecht, Netherlands, 1999.
94. H. D. Sherali and B. M. P. Fraticelli. Enhancing RLT relaxations via a new class of semidefinite cuts. *Journal of Global Optimization*, 22(1-4):233–261, 2002.
95. H. D. Sherali, E. Dalkiran, and J. Desai. Enhancing RLT-based relaxations for polynomial programming problems via a new class of v-semidefinite cuts. *Computational Optimization and Applications*, 52(2):483–506, 2012.
96. R. Sotirov. SDP relaxations for some combinatorial optimization problems. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 795–819. Springer, 2012.
97. M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.
98. D. Vandenbussche and G. L. Nemhauser. A branch-and-cut algorithm for nonconvex quadratic programs with box constraints. *Mathematical Programming*, 102(3):559–575, 2005.
99. S. A. Vavasis. Quadratic programming is in NP. Technical report, Cornell University, 1990.
100. J. P. Vielma, I. Dunning, J. Huchette, and M. Lubin. Extended formulations in mixed integer conic quadratic programming. *Mathematical Programming Computation*, 9(3):369–418, 2017.
101. S. Vigerske. *Decomposition in Multistage Stochastic Programming and a Constraint Integer Programming Approach to Mixed-Integer Nonlinear Programming*. PhD in Mathematics, Humboldt-University Berlin, 2012.
102. S. Vigerske. MINLPLib 2. In *Proceedings of the XII global optimization workshop MAGO*, volume 2014, pages 137–140, 2014.
103. S. Vigerske. MINLPLib: A library of mixed-integer and continuous nonlinear programming instances, 2018. URL http://www.minlplib.org/.
104. M. Walter. Sparsity of lift-and-project cutting planes. In *Operations Research Proceedings 2012*, pages 9–14. Springer, 2014.
105. F. Wesselmann and U. Stuhl. Implementing cutting plane management and selection techniques. Technical report, University of Paderborn, 2012.
106. X. Zheng, X. Sun, and D. Li. Improving the performance of MIQP solvers for quadratic programs with cardinality and minimum threshold constraints: A semidefinite program approach. *INFORMS Journal on Computing*, 26(4):690–703, 2014.