Solving Nonconvex Optimization Problems using Outer Approximations of the Set-Copositive Cone

Markus Gabl* and Kurt Anstreicher†

November 15, 2023

Abstract

We consider the solution of nonconvex quadratic optimization problems using an outer approximation of the set-copositive cone that is iteratively strengthened with conic constraints and cutting planes. Our methodology utilizes an MILP-based oracle for a generalization of the copositive cone that considers additional linear equality constraints. In numerical testing we evaluate our algorithm on a variety of different nonconvex quadratic problems.

1 Introduction

In this paper we are interested in an optimization problem of the form

$$\begin{array}{llll} \text{CPOPT}: & \min & C \bullet Y & \\ & \text{s.t.} & B_i \bullet Y = b_i, & i = 1, \dots, m \\ & & Y \in \mathcal{CP}^{n+1}, & \end{array}$$

where $\mathcal{CP}^k$, the cone of $k \times k$ completely positive matrices, are those matrices that can be written in the form $\sum_{i=1}^r u_i u_i^T$ with each $u_i \in \mathbb{R}^k_+$. In CPOPT the matrix Y has the form

$$Y = \begin{pmatrix} Y_{00} & x^T \\ x & X \end{pmatrix}.$$

and the constraints of CPOPT include the constraint that $Y_{00} = 1$. The matrix X can then be viewed as a relaxation of the rank-one matrix xx^T, where $x \geq 0$. Since the constraints of CPOPT include $Y_{00} = 1$, and all of the equality constraints are written in terms of Y, we can assume without loss of generality that $b_i = 0$ for all other constraints. For convenience we will also consider the rows and columns of Y to be indexed $0, 1, \dots, n$, so that $Y_{ij} = X_{ij}$ for $i, j \geq 1$.

By the result of Burer [9], a variety of NP-Hard problems, including indefinite quadratic optimization with continuous and binary variables, can be exactly represented using

*Institute of Operations Research, Karlsruhe Institute of Technology, Karlsruhe, Germany

†Dept. of Business Analytics, University of Iowa, Iowa City, Iowa, USA

CPOPT. The difficulty with CPOPT is that for $k \geq 5$ the cone $\mathcal{CP}^k$ is very difficult to explicitly represent. A tractable relaxation for $\mathcal{CP}^k$ is $\mathcal{DNN}^k$, the cone of $k \times k$ doubly-nonnegative (DNN) matrices, and using $\mathcal{DNN}^{n+1}$ in place of $\mathcal{CP}^{n+1}$ results in a computable relaxation of the original problem. However the solution of this DNN relaxation will not in general solve CPOPT. If the DNN relaxation is not tight then its solution $Y^* \notin \mathcal{CP}^{n+1}$, and one approach to strengthening the DNN relaxation would then be to generate a cut that separates Y^* from $\mathcal{CP}^{n+1}$. This approach is applied in the case where Y^* has block structure in [14]. However in general the problem of determining if a matrix is CP is NP-Hard [11], and if not then finding a cut that separates the matrix from the CP cone is computationally difficult.

A different approach to strengthening the DNN relaxation of CPOPT is to use a hierarchy of cones $\mathcal{C}_r^k$, $r \geq 0$, where $\mathcal{C}_0^k = \mathcal{DNN}^k$ and $\mathcal{CP}^k \subset \mathcal{C}_{r+1}^k \subset \mathcal{C}_r^k$ for each r; see for example [23, 13]. In such a hierarchy the cones, $\mathcal{C}_r^k$ better approximate $\mathcal{CP}^k$ as r increases, so replacing $\mathcal{CP}^k$ with $\mathcal{C}_r^k$, $k > 0$ improves upon the DNN relaxation. However these hierachies involve extended-variable formulations whose size grows very rapidly in r, and therefore the use of $\mathcal{C}_r^k$ for $r > 1$ is typically impractical unless k is very small.

An alternative to working with the problem CPOPT is to use its conic dual. The dual problem has the form

$$\begin{aligned} \text{COPOPT}: \quad \max \quad & b^T v \\ \text{s.t.} \quad & C - \sum_{i=1}^m v_i B_i = S \\ & S \in \mathcal{COP}^{n+1}, \end{aligned}$$

where $\mathcal{COP}^k$ is the cone of $k \times k$ copositive (COP) matrices, $\mathcal{COP}^k = \{M \in \mathcal{S}^k : x^T M x \geq 0 \,\forall x \in \mathbb{R}_+^k\}$. Under a regularity condition, for example the existence of an interior solution in CPOPT, the primal and dual problems have equal solution values. Corresponding to the relaxations of $\mathcal{CP}^k$ described above there are dual cones that are interior to $\mathcal{COP}^k$. For example, the dual of $\mathcal{DNN}^k$ is the cone $\mathcal{SPN}^k$ of matrices that can be written as the sum of a $k \times k$ non-negative matrix and a $k \times k$ PSD matrix.

Our interest here is to work with the dual problem COPOPT, but replacing $\mathcal{COP}^{n+1}$ with an *outer* approximation which will then be strengthened via the addition of conic constraints and linear cuts. In order to use this approach we need an initial outer approximation for $\mathcal{COP}^{n+1}$ matrices, and a way to determine if a matrix is COP and if not generate a cut that separates it from $\mathcal{COP}^{n+1}$. From a theoretical standpoint the problem of determining whether or not a matrix is copositive is also very difficult [11]. However, recent papers [2, 17, 25] have shown that the problem of determining whether or not a matrix is copositive, and if not generating a cut that separates it from the COP cone, can be formulated as a mixed-integer linear programming (MILP) problem. The availability of very efficient software for MILP problems makes this problem computationally much more practical than the corresponding detection/separation problem for CP matrices.

There are different algorithmic frameworks in which one can employ cuts based on the failure of a matrix to be copositive. One possibility is to use an ellipsoid-type algorithm that at each iteration $k \geq 0$ has a candidate solution v^k for COPOPT and an ellipsoid $\mathcal{E}_k$ centered at v^k that must contain the optimal solution v^* of COPOPT. If $S^k = C - \sum_{i=1}^m v_i^k B_i \notin \mathcal{COP}^{n+1}$, then a cut vector $u \geq 0$ with $u^T S^k u < 0$ is generated, and the cut $\sum_{i=1}^m v_i(u^T B_i u) \le u^T Cu - u^T S_k u$ is added. If S^k is copositive then an objective cut $b^T v \ge b^T v^k$ is added. Following the addition of the cut, a new point v^{k+1} and ellipsoid $\mathcal{E}_{k+1}$ are generated. Algorithms of this type include the central-cut ellipsoid algorithm [19], the analytic center cutting-plane method (ACCPM) [16] , and the volumetric cutting plane algorithm with central cuts [1]. These algorithms have the advantage of all being provably convergent, with the ellipsoid algorithm and the volumetric cutting-plane algorithm requiring a polynomial number of steps, and the ACCPM being more efficient in practice. However in all cases the problem of determining the next iterate v^{k+1} becomes numerically ill-conditioned and the algorithm may fail before a sufficiently accurate solution is found.

The application of ellipsoid-type algorithms to a copositive programming problem of the form

$$\begin{array}{ll} \min & C \bullet X \\ \text{s.t.} & B_i \bullet X \le b_i, \quad i = 1, \dots, m, \\ & X \in \mathcal{COP}^n, \end{array} \tag{1}$$

was first suggested in [12] (see also [3]), and developed further in [5]. These papers all consider applying a problem of the type (1) to determining if a given matrix C is completely positive. A complexity result for this problem using the ellipsod algoriithm is given in [3, 12] while [5] gives computational results using the ellipsoid algorithm, the volumetric cutting-plane method and the ACCPM.

An alternative to an ellipsoid-type algorithm for solving COPOPT is to use a simpler outer approximation (OA) algorithm. On each iteration of such an algorithm we consider the problem COPOPT but with $\mathcal{COP}^{n+1}$ replaced by an outer approximation $\mathcal{C} \supset \mathcal{COP}^{n+1}$. This problem is solved, generating an iterate v^k and $S^k = C - \sum_{i=1}^m v_i^k B_i$. If $S^k \in \mathcal{COP}^{n+1}$ then v^k solves COPOPT. If $S^k \notin \mathcal{COP}^{n+1}$ then we generate a cut $u \ge 0$ with $u^T S^k u < 0$, and the cut $u^T S u \ge 0$ is added to $\mathcal{C}$. There are several advantages to this approach. First, although ellipsoid-type methods are provably convergent they are very susceptible to numerical ill-conditioning. Second, the placement of the cut $u^T S u \ge 0$ does not need to be "backed off" to $\sum_{i=1}^m v_i(u^T B_i u) \le u^T Cu - u^T S_k u$ as in an ellipsoid-type method using central cuts. Finally, we wish to consider the possibility of adding additional conic constraints that are more general than linear cuts. It is possible that the theory of the methods described above could be extended to deal with such constraints, but the algorithms as currently developed only incorporate linear cuts.

In the next section we describe how to initially approximate the cone $\mathcal{COP}^{n+1}$ using linear and conic constraints, and how a cut $u \ge 0$ with $u^T S^k u < 0$ can be generated when $S^k \notin \mathcal{COP}^{n+1}$. In Section 3 we modify the cut-generation procedure to explicitly incorporate additional linear constraints $Ax = d$ from CPOPT. In Section 4 we apply our outer-approximation algorithm to two different classes of non-convex quadratic optimization problems, and in Section 5 we consider the application of our algorithm to nonconvex quadratic problems arising from adjustable robust optimization. In Section 6 we make some concluding remarks.

Notation. For matrices A and B of the same size we use $A \bullet B$ to denote the matrix inner product $A \bullet B = \text{tr}(AB^T)$, and $A \circ B$ to denoted the Hadamard or componentwise product. For vectors x and y we use $(x; y)$ to denote the vector $(x^T, y^T)^T$. We use e to denote a vector of arbitrary dimension with each component equal to one.

2 Approximating the copositive cone

In order to implement an OA algorithm for COPOPT we need an initial outer approximation for the cone $\mathcal{COP}^{n+1}$ and a way to generate a cut for a matrix $S \notin \mathcal{COP}^{n+1}$. In this section we consider these two topics.

For an initial outer approximation, any $S \in \mathcal{COP}^{n+1}$ certainly satisfies the constraints

$$(e_i + e_j)^T S(e_i + e_j) \geq 0, \quad 1 \leq i \leq j \leq n+1, \tag{2}$$

where $e_i \in \mathbb{R}^{n+1}$ is the elementary vector with a one in the ith coordinate and zeros elsewhere. The use of (2) as an outer approximation for $\mathcal{COP}^{n+1}$ is dual to using the cone of nonnegative diagonally dominant matrices $\mathcal{DD}_+^{n+1}$ as an inner approximation of $\mathcal{CP}^{n+1}$ [18], and was used as an initial outer approximation for the copositive cone in [3, 12].

The inner approximation $\mathcal{DD}_+^{n+1} \in \mathcal{CP}^{n+1}$ can be expanded by considering symmetric diagonal scalings of matrices in $\mathcal{DD}_+^{n+1}$, leading to the cone $\mathcal{SDD}_+^{n+1}$ [18]. This cone corresponds to matrices all of whose 2×2 principal submatrices are completely positive, whose dual is the cone of matrices all of whose 2×2 principal submatrices are copositive. Since for $k \leq 4$ the cones $\mathcal{COP}^k$ and $\mathcal{SPN}^k$ are the same, we can then improve on the outer approximation (2) by using the constraints

$$\bar{S}_{ij}^2 \leq S_{ii} S_{jj}, \quad S_{ij} \geq \bar{S}_{ij}, \quad 1 \leq i < j \leq n+1. \tag{3}$$

Compared to (2), the constraints (3) require one additional variable $\bar{S}_{ij}$ and a rotated second-order cone (SOC) constraint for each $i < j$. A reasonable approach to avoid generating all of these constraints *a-priori* would be to first solve a problem using the approximation (2), and then add violated constraints from (3). Note that for a given i, j, checking if the constraint from (3) is satisfied amounts to checking if $S_{ij} \geq -\sqrt{S_{ii} S_{jj}}$, which is trivial. The use of the constraints (3) to strengthen an initial approximation of the copositive cone was also suggested in [20].

Let s_i^T denote the ith row of S, $\nu \in \mathbb{R}_{++}^{n+1}$ and assume that θ is a positive integer. The approach for checking if a matrix S is copositive from [2] is based on solving the MILP problem

$$\begin{array}{lll} \text{MILP}_{\mathcal{COP}}: & \max & \gamma \\ & \text{s.t.} & s_i^T u \leq -\gamma + \nu_i(1 - z_i), \ i = 1, \ldots, n+1 \\ & & \gamma \geq 0, \ 0 \leq u \leq z, \\ & & z \in \{0, 1\}^{n+1}, \ e^T z \geq \theta. \end{array}$$

For $\theta = 1$ the solution value in $\text{MILP}_{\mathcal{COP}}$ is zero if and only if $S \in \mathcal{COP}^{n+1}$, and assuming that $\text{diag}(S) \geq 0$ the same is true for $\theta = 2$. If $S \notin \mathcal{COP}^{n+1}$ then the solution of $\text{MILP}_{\mathcal{COP}}$ has $\gamma > 0$, and $u \geq 0$ with $u^T S u < 0$ demonstrates that the principal submatrix of S corresponding to $\{i : z_i = 1\}$ is not copositive. Although the equivalence between copositivity of S and $\gamma = 0$ in the solution of $\text{MILP}_{\mathcal{COP}}$ holds for any $\nu > 0$, in practice the components of ν should be related to the values in S to avoid the possibility of a very small but positive optimal value. A simple suggestion for such a ν is given in [2].

As described above, the constraints (3) correspond to all 2×2 submatrices of S being copositive. By solving $\text{MILP}_{\mathcal{COP}}$ with the constraint $1 \leq e^T z \leq 4$ we can also check to see if there are 3×3 or 4×4 principal submatrices of S that fail to be copositive. In the case where such a submatrix is found, we could use the fact that $\mathcal{COP}^k = \mathcal{SPN}^k$ for $k \leq 4$ to impose a constraint involving a PSD matrix that would enforce copositivity of the submatrix. Instead we will describe a method [21] for generating an SOC constraint that improves upon the linear cut found by MILP$_{\mathcal{COP}}$ while avoiding the computational expense of working with a PSD matrix.

Suppose that $S \in \mathcal{COP}^k$ for $k \in \{3, 4\}$. Then $S = \bar{S} + N$, where $\bar{S} \succeq 0$ and $N \geq 0$. Assume that $\bar{S}$ has the form:

$$\bar{S} = \begin{pmatrix} \sigma & \hat{s}^T \\ \hat{s} & \hat{S} \end{pmatrix} \tag{4}$$

where $\sigma \geq 0$ and $\hat{s} \in \mathbb{R}^{k-1}$. Then

$$\begin{aligned} \bar{S} \succeq 0 \quad &\iff \quad \sigma\hat{S} - \hat{s}\hat{s}^T \succeq 0, \\ &\iff \quad \hat{u}^T(\sigma\hat{S} - \hat{s}\hat{s}^T)\hat{u} \geq 0 \quad \forall \hat{u} \in \mathbb{R}^{k-1} \\ &\iff \quad (\hat{s}^T\hat{u})^2 \leq \sigma\hat{u}^T\hat{S}\hat{u} \quad \forall \hat{u} \in \mathbb{R}^{k-1}. \end{aligned} \tag{5}$$

For fixed $\hat{u}$ the constraint in (5) is a rotated second-order cone, which implicitly includes the constraints $\sigma \geq 0$ and $\hat{u}^T\hat{S}\hat{u} \geq 0$.

Now assume that $\bar{S}$ has the form (4) with $\sigma \geq 0$. We claim that for fixed $\bar{u} = (\mu; \hat{u})$ the constraint (5) implies that $\bar{u}^T\bar{S}\bar{u} \geq 0$. To see this, note that

$$\bar{u}^T\bar{S}\bar{u} = (\mu \ \hat{u}^T)\begin{pmatrix} \sigma & \hat{s}^T \\ \hat{s} & \hat{S} \end{pmatrix}\begin{pmatrix} \mu \\ \hat{u} \end{pmatrix} = \sigma\mu^2 + 2\mu\hat{s}^T\hat{u} + \hat{u}^T\hat{S}\hat{u}.$$

If $\sigma = 0$ then (5) implies that $\hat{s}^T\hat{u} = 0$, so $\bar{u}^T\bar{S}\bar{u} = \hat{u}^T\hat{S}\hat{u} \geq 0$. If $\sigma > 0$, then $\bar{u}^T\bar{S}\bar{u} \geq 0$ is equivalent to $\sigma\bar{u}^TS\bar{u} = \sigma^2\mu^2 + 2\sigma\mu\hat{s}^T\hat{u} + \sigma\hat{u}^T\hat{S}\hat{u} \geq 0$, and (5) implies that

$$\sigma^2\mu^2 + 2\sigma\mu\hat{s}^T\hat{u} + \sigma\hat{u}^T\hat{S}\hat{u} \geq \sigma^2\mu^2 + 2\sigma\mu\hat{s}^T\hat{u} + (\hat{s}^T\hat{u})^2 = (\sigma\mu + \hat{s}^T\hat{u})^2 \geq 0.$$

If MILP$_{\mathcal{COP}}$ finds a cut $u \geq 0$ with $3 \leq e^Tz \leq 4$, then we can apply the constraints

$$S_\beta \geq \bar{S}, \quad (\hat{s}^T\hat{u})^2 \leq \sigma\hat{u}^T\hat{S}\hat{u} \tag{6}$$

to the principal submatrix S_β corresponding to $\{i : z_i = 1\}$, where $\bar{S}$ has the form (4). These constraints are stronger than the simple cut $u^TSu \geq 0$, but avoid the computational expense of imposing the PSD constraint on $\bar{S}$.

3 Set-copositivity detection with linear constraints

An outer approximation algorithm based on MILP$_{\mathcal{COP}}$, as described in the previous section, uses MILP$_{\mathcal{COP}}$ to check if $S \in \mathcal{COP}^{n+1}$ for an iterate S, and if not to generate a cut $u \geq 0$ with $u^TSu < 0$. Exactly such an approach is implemented in [20] to solve the copositive dual of a unit-commitment problem in order to obtain a copositive pricing matrix that applies to an underlying problem with discrete variables. In our own preliminary implementation of such an algorithm, we observed cases where MILP$_{\mathcal{COP}}$ repeatedly generated cuts, but the addition of these cuts produced very little change in the objective.

On closer inspection, it was clear that in these cases the cut vector u had no relationship to the feasible region of the original problem, which contains equality constraints of the form $Ax = d$. Our goal here is to modify MILP$_{\mathcal{COP}}$ so as to explicitly incorporate information from these constraints.

Assume now that the original problem includes constraints $Ax = d$, $x \geq 0$ where A is an $r \times n$ matrix. Following the methodology of [9], each such constraint $a_i^T x = d_i$ produces two constraints in CPOPT,

$$\begin{pmatrix} 0 & a_i^T \\ a_i & 0 \end{pmatrix} \bullet Y = 2d_i, \quad \begin{pmatrix} 0 & 0 \\ 0 & a_i a_i^T \end{pmatrix} \bullet Y = d_i^2, \tag{7}$$

and if desired these constraints can be homogenized using the fact that $Y_{00} = 1$. To homogenize the original constraints, let $y = (y_0; x) = (x_0; x)$ and $\hat{A} = [-b, A]$. Then $Ax = b$ can be written as $\hat{A}y = 0$, $y_0 = 1$. Since $Y_{00} = 1$ in CPOPT, we can also consider y to be column zero of the matrix Y. To reduce notion, in the sequel we will replace $\hat{A}$ with A and write the homogenized constraints as $Ay = 0$. No confusion should result.

For an arbitrary convex cone $\mathcal{K} \in \mathbb{R}^k$, let $\mathcal{COP}(\mathcal{K})$ denote the matrices that are copositive over $\mathcal{K}$,

$$\mathcal{COP}(\mathcal{K}) = \{M \in \mathcal{S}^k : y^T M y \geq 0 \ \forall y \in \mathcal{K}\}.$$

The usual copositive cone is then $\mathcal{COP}^k = \mathcal{COP}(\mathbb{R}^k_+)$. We are now interested in the cone $\mathcal{COP}(\mathcal{K})$ with

$$\mathcal{K} = \{y \geq 0 : Ay = 0\}, \tag{8}$$

where A is an $r \times k$ matrix. Clearly $\mathcal{COP}^k \subset \mathcal{COP}(\mathcal{K})$ for any such matrix A. We call $\mathcal{COP}(\mathcal{K})$ the *set-copositive matrix cone* with respect to the *ground cone* $\mathcal{K} \subseteq \mathbb{R}^n$. The dual of this cone, i.e. $\mathcal{CP}(\mathcal{K}) = \operatorname{conv}\left\{xx^T : x \in \mathcal{K}\right\}$ we call the *set-completely positive matrix cone*. Some basic properties of these types of cones are summarized in a recent review paper [8, Section 2]; see especially Proposition 11. Such cones appear in earlier literature on copositive reformulations such as [10, 26, 7]. Within this literature the cone $\mathcal{COP}(\mathcal{K})$ is sometimes referred to as a *set-semidefinite cone* [15], but we prefer the set-copositive nomenclature.

Lemma 1. *Assume that* CPOPT *includes the constraints $Ay = 0$, and has an optimal solution Y that is rank-one; $Y = yy^T$, $y \geq 0$, $y_0 = 1$. Then if strong duality holds between* CPOPT *and* COPOPT*, it also holds if $S \in \mathcal{COP}^{n+1}$ is replaced by $S \in \mathcal{COP}(\mathcal{K})$ in* COPOPT*, where $\mathcal{K}$ is given by* (8)*.*

Proof. The objective gap between solutions Y and (v, S) that are feasible in CPOPT and COPOPT, respectively, is

$$\begin{aligned} C \bullet Y - b^T v &= \left(S + \sum_{i=1}^m v_i B_i\right) \bullet Y - \sum_{i=1}^m v_i b_i \\ &= \sum_{i=1}^m v_i (B_i \bullet Y - b_i) + S \bullet Y \\ &= S \bullet Y, \end{aligned}$$

and $Y \in \mathcal{CP}^{n+1}$, $S \in \mathcal{COP}^{n+1}$ implies that $S \bullet Y \geq 0$. Under the assumptions of the lemma, there is an optimal solution Y of the form $Y = yy^T$, $y \geq 0$, $y_0 = 1$, $Ay = 0$. For such a Y we have $S \bullet Y = y^T S y$, so $S \in \mathcal{COP}(\mathcal{K})$ implies that $S \bullet Y \geq 0$. Moreover if (v, S) is an optimal solution of COPOPT then $Y \bullet S = 0$, and (v, S) remains feasible when $\mathcal{COP}^{n+1}$ is replaced by $\mathcal{COP}(\mathcal{K})$. $\square$

Motivated by Lemma 1, we consider replacing the problem $\text{MILP}_{\mathcal{COP}}$ with a generalization that determines whether or not a matrix $S \in \mathcal{COP}(\mathcal{K})$ rather than $S \in \mathcal{COP}^{n+1}$. To do this, consider the problem

$$\begin{aligned} \text{QP}(S, A): \quad \min \quad & u^T S u \\ \text{s.t.} \quad & Au = 0 \\ & e^T u \leq 1 \\ & u \geq 0. \end{aligned}$$

The KKT conditions for $\text{QP}(S, A)$ problem are necesary, since it has only linear constraints, and therefore there are (λ, γ, t) so that the solution u also satisfies the system:

$$\begin{aligned} Su + A^T\lambda + \gamma e &= t \\ \gamma \geq 0,\ t \geq 0,\ \gamma(1 - e^T u) = 0,\ u^T t &= 0. \end{aligned}$$

Note that the solution u satisfies $u^T S u = u^T(t - A^T\lambda - \gamma e) = -\gamma$. In addition, note that it is possible that the solution of $\text{QP}(S, A)$ has $\gamma = 0$, but the solution has $\gamma > 0$ if the constraints $Au = 0$ are dropped.

We now define a generalization of $\text{MILP}_{\mathcal{COP}}$ that takes into account the presence of the equality constraints $Ay = 0$ in $\mathcal{K}$ from (8):

$$\begin{aligned} \text{MILP}_{\mathcal{COP}(\mathcal{K})}: \quad \max \quad & \gamma \\ \text{s.t.} \quad & Su + A^T\lambda \leq -\gamma e + \nu \circ (e - z), \\ & Au = 0,\ 0 \leq \gamma \leq \alpha e^T u, \\ & 0 \leq u \leq z,\ z \in \{0, 1\}^{n+1}. \end{aligned}$$

Lemma 2. *Assume that $\nu > 0$, $\alpha > 0$, and let $\mathcal{K}$ be as in (8). Then $S \in \mathcal{COP}(\mathcal{K})$ if and only if the solution of $\text{MILP}_{\mathcal{COP}(\mathcal{K})}$ has $\gamma = 0$.*

Proof. We will prove the equivalent statement that the solution of $\text{MILP}_{\mathcal{COP}(\mathcal{K})}$ has $\gamma > 0$ if and only if the solution value of $\text{QP}(S, A)$ is negative. Suppose that there is a solution of $\text{MILP}_{COP(\mathcal{K})}$ with $\gamma > 0$. Note that $u_i > 0 \implies z_i = 1$, so $u^T[\nu \circ (e - z)] = u^T(e - z) = 0$. Therefore $u^T S u \leq -\gamma e^T u \leq -\gamma^2/\alpha$, and scaling u by $1/(e^T u)$ obtains a feasible solution in $\text{QP}(S, A)$ with a negative objective value. Next assume that the solution value in $\text{QP}(S, A)$ is negative. Then there is a solution of the KKT system with $\gamma > 0$, $u \neq 0$. Let $z_i = 1$ if $u_i > 0$, and $z_i = 0$ otherwise. Scaling (u, λ, γ) then obtains a solution with $\gamma > 0$ that is feasible for all of the constraints of $\text{MILP}_{\mathcal{COP}(\mathcal{K})}$ except for possibly $\gamma \leq \alpha e^T u$. Reducing γ if necessary to satisfy this constraint, we obtain a solution of $\text{MILP}_{\mathcal{COP}(\mathcal{K})}$ with $\gamma > 0$. $\square$

Note that the condition $\gamma \leq \alpha e^T u$ in $\text{MILP}_{\mathcal{COP}(\mathcal{K})}$ prevents the possibility that $\gamma > 0$ but $u = 0$. In $\text{MILP}_{\mathcal{COP}}$ this was prevented by the condition $e^T z \geq \theta \geq 1$, but it is not obvious that this suffices in $\text{MILP}_{\mathcal{COP}(\mathcal{K})}$ due to the term $A^T\lambda$. It should also be noted that there is a somewhat counter-intuitive aspect to using $\mathcal{COP}(\mathcal{K})$ in place of $\mathcal{COP}^{n+1}$ in COPOPT. Since $\mathcal{COP}^{n+1} \subset \mathcal{COP}(\mathcal{K})$, it appears that using $\mathcal{COP}(\mathcal{K})^{n+1}$ in place of $\mathcal{COP}^{n+1}$ is weakening rather than strengthening the constraints of COPOPT, but Lemma 1 shows that this change has no effect on the solution. Our motivation in using $\mathcal{COP}(K)$ is not to strengthen the constraints of COPOPT but rather to restrict the cuts being generated to be more relevant to the underlying problem, which includes the constraints $Ay = 0$.

4 Computational results

In this section we consider applying an algorithm that uses cuts based on an outer approximation of the set-copositive cone, incorporating additional constraint information as described in the previous section. We consider two different classes of problems for our computations. All computations in this section, and the following section, were performed on an Intel(R) Core(TM) i5-9300H CPU running at 2.40GHz with 16GB RAM. Linear and mixed integer linear problems were solved via Gurobi 9.1 while nonlinear conic problems were solved via Mosek 9.2. Both were accessed from Matlab via the YALMIP interface [22].

4.1 MINLPLib instances

To begin, we consider problems from MINLPLib corresponding to nonconvex quadratic programming (QP) problems, with only continuous variables. Using the results from [9] such problems can be exactly represented in the form CPOPT under mild assumptions. In order to put problems into the required form for this representation, all variables must be nonnegative and slacks added to any inequalities to convert them to equalities. We considered instances where the number of variables after splitting free variables and/or adding slacks was no more than 70.

In Table 1 we give the results of applying our outer approximation algorithm to 62 nonconvex QP instances. In the table we give the (known) optimal value for each problem, the value for the DNN relaxation of CPOPT (equal to the value for the SPN restriction of COPOPT), and the value obtained by applying our outer approximation algorithm using cuts based on $\text{MILP}_{\mathcal{COP}(\mathcal{K})}$. We also record the gap between the optimal value and the DNN value, the gap between the COP outer approximation value and the optimal value and the number of cuts and time used by the OA algorithm. The OA algorithm was terminated once the time exceeded 3600 seconds, or the time attempting to generate a cut exceeded 600 seconds, or the number of cuts exceeded 100.

In Table 2 we summarize the number of MINLIBLib problems on which the DNN relaxation and COP-OA algorithm are numerically exact. Out of the 62 total problems, the DNN relaxation and the COP outer approximation algorithm are exact on almost the same number of problems, 46 for the DNN relaxation and 47 for the COP-OA algorithm, but the two methods are simultaneously exact on only 38 problems. Although the COP-OA algorithm is exact on one more instance than the DNN relaxation, it should be noted that there are several instances on which the COP-OA gap is much higher then the DNN gap, with most of these corresponding to larger problems where the algorithm was

Table 1: Results on MINLPLib problems

			Objective Value			**Gap**		**COP-OA**	
Instance	n	m	Opt	DNN	COP-OA	DNN	COP-OA	cuts	time
ex2_1_1	11	1	-17.000	-18.160	-17.000	1.160	0.000	2	3.82
ex2_1_2	13	2	-213.000	-213.000	-213.000	0.000	0.000	2	0.77
ex2_1_3	32	9	-15.000	-15.000	-15.000	0.000	0.000	4	49.87
ex2_1_4	15	5	-11.000	-11.000	-11.000	0.000	0.000	2	0.92
ex2_1_5	31	11	-268.015	-268.015	-268.015	0.000	0.000	101	91.23
ex2_1_6	25	5	-39.000	-39.828	-39.000	0.828	0.000	16	9.08
ex2_1_7	30	10	-4150.410	-4334.185	-2259.365	183.775	1891.045	12	3844.68
ex2_1_8	48	10	15639.000	15639.000	15639.000	0.000	0.000	2	7.67
ex2_1_9	10	1	0.375	-0.375	-0.375	0.750	0.000	2	0.63
ex2_1_10	30	10	49318.018	49318.018	49318.018	0.000	0.000	26	3713.29
meanvar	16	2	5.243	5.243	5.243	0.000	0.000	2	0.53
nemhaus	5	5	31.000	31.000	31.000	0.000	0.000	1	0.25
qp2	51	2	0.001	0.001	0.001	0.000	0.000	3	437.05
st_bpaf1a	30	10	-45.380	-45.380	-45.380	0.000	0.000	10	9.99
st_bpaf1b	30	10	-42.626	-42.963	-42.963	0.337	0.000	6	6.42
st_bpk1	10	6	-13.000	-13.000	-13.000	0.000	0.000	3	0.81
st_bpv1	12	4	10.000	10.000	10.000	0.000	0.000	3	0.77
st_bpv2	13	5	-8.000	-8.000	-8.000	0.000	0.000	3	0.86
st_bsj2	8	5	1.000	1.000	1.000	0.000	0.000	1	0.25
st_bsj4	16	4	-70262.050	-71232.380	-70262.050	970.330	0.000	5	1.65
st_cqpf	11	6	-2.750	-2.750	-2.748	0.000	0.002	5	1.26
st_e22	9	5	-85.000	-85.000	-85.000	0.000	0.000	3	0.79
st_e23	6	2	-1.083	-1.083	-1.083	0.000	0.000	4	0.99
st_e24	8	4	3.000	3.000	3.000	0.000	0.000	2	0.51
st_e25	12	8	0.890	0.890	0.890	0.000	0.000	2	0.53
st_e26	8	4	-185.779	-185.779	-185.779	0.000	0.000	2	0.50
st_fp7a	30	10	-354.751	-354.823	-72.879	0.072	281.872	9	3840.53
st_fp7b	30	10	-634.751	-634.820	-359.667	0.069	275.083	11	3823.94
st_fp7c	30	10	-8695.012	-8696.586	-5531.468	1.574	3163.544	15	3746.18
st_fp7d	30	10	-114.751	-114.819	8.075	0.069	122.826	9	3812.60
st_fp7e	30	10	-3730.410	-3914.185	-1839.365	183.775	1891.045	13	3640.41
st_fp8	44	20	15639.000	15639.000	15639.000	0.000	0.000	3	7.00
st_ht	7	3	-1.600	-2.000	-1.600	0.400	0.000	3	0.75
st_jcbpa	33	13	-794.856	-794.856	-11.000	0.000	783.856	101	530.38
st_m1	31	11	-461356.939	-461356.939	-461356.939	0.000	0.000	101	1817.54
st_m2	51	21	-856648.819	-856648.816	-856648.926	0.000	0.000	6	3613.95
st_pan1	7	4	-5.284	-5.284	-5.284	0.000	0.000	4	1.05
st_ph1	11	5	-230.117	-230.117	-230.117	0.000	0.000	5	1.49
st_ph2	11	5	-1028.117	-1028.117	-1028.117	0.000	0.000	5	1.51
st_ph3	11	5	-420.235	-420.235	-420.235	0.000	0.000	5	1.87
st_ph11	7	4	-11.281	-11.478	-11.281	0.196	0.000	2	0.53
st_ph12	7	4	-22.625	-23.088	-22.625	0.463	0.000	2	0.55
st_ph13	13	10	-11.281	-11.461	-11.281	0.180	0.000	2	0.59
st_ph14	13	10	-229.722	-229.722	-229.722	0.000	0.000	2	0.60
st_ph15	8	4	-392.704	-392.704	-392.704	0.000	0.000	3	0.83
st_ph20	12	9	-158.000	-158.000	-158.000	0.000	0.000	3	0.86
st_phex	7	5	-85.000	-85.000	-85.000	0.000	0.000	3	1.50
st_qpc-m0	4	2	-5.000	-5.000	-5.000	0.000	0.000	2	0.82
st_qpc-m1	5	10	-473.778	-473.778	-473.778	0.000	0.000	3	1.46
st_qpc-m3a	20	10	-382.695	-382.695	-382.695	0.000	0.000	3	1.47
st_qpc-m3b	20	10	0.000	0.000	0.000	0.000	0.000	101	39.70
st_qpc-m3c	20	10	0.000	0.000	0.000	0.000	0.000	1	3.16
st_qpc-m4	20	10	0.000	0.000	0.000	0.000	0.000	17	5.76
st_qpk1	6	4	-3.000	-3.000	-3.000	0.000	0.000	2	0.63
st_qpk2	18	12	-12.250	-12.250	-12.250	0.000	0.000	2	0.75
st_qpk3	33	22	-36.000	-36.000	-36.000	0.000	0.000	2	2.08
st_rv1	15	5	-59.944	-59.944	-59.904	0.000	0.040	3	1.25
st_rv2	30	10	-64.481	-64.481	-61.123	0.000	3.358	38	3663.46
st_rv3	40	20	-35.761	-35.761	-20.900	0.000	14.861	6	3618.24
st_rv7	50	20	-138.188	-138.187	-48.793	0.000	89.394	6	3622.44
st_rv8	60	20	-132.662	-132.661	-117.036	0.000	15.626	6	3626.00
st_rv9	70	20	-120.153	-120.427	-44.798	0.274	75.355	7	3725.03

Table 2: Solution status for DNN relaxation and COP-OA algorithm

		COP-OA		
		Gap $= 0$	Gap > 0	Total
DNN	Gap $= 0$	38	8	46
	Gap > 0	9	7	16
	Total	47	15	62

terminated after exceeding the 3600 second time limit. Another way of looking at the summary in Table 2 is that of the 16 problems on which the DNN relaxation is *not* exact, the COP-OA algorithm is exact on 9. The DNN relaxation is computationally relatively inexpensive, but as noted in the Introduction there is no practical way to strengthen it when it does not provide the optimal solution.

The results in Tables 1 and 2 were obtained by initializing the relaxation of COPOPT using the constraints (2) and then generating cuts using $\text{MILP}_{\mathcal{COP}(\mathcal{K})}$. We also experimented with versions of the COP-OA algorithm that considered additional cuts, including:

- Cuts obtained from $\text{MILP}_{\mathcal{COP}}$ in addition to those obtained from $\text{MILP}_{\mathcal{COP}(\mathcal{K})}$;
- Linear and SOC constraints (3) obtained from 2×2 submatrices;
- Linear and SOC constraints (6) obtained from 3×3 or 4×4 submatrices.

Note that none of these additional cuts make use of information from any linear constraints. In our experiments none of these additional cuts resulted in consistent improvements, and in some cases the performance of the algorithm was substantially degraded.

4.2 Binary QPs

For this set of experiments we consider binary quadratic programming (BQP) problems of the form

$$\begin{aligned} \text{BQP}: \quad \min \quad & x^T Q x + q^T x \\ \text{s.t.} \quad & Ax \leq b \\ & x_i \in \{0,1\}, i = 1, \ldots, n, \end{aligned}$$

where A is an $m \times n$ matrix. Using the methodology of [9], any such problem has an exact representation as a problem of the form CPOPT, obtained by adding slacks to convert the inequalities $Ax \leq b$ to equalities $Ax + s = b$, $s \geq 0$ and also adding explicit constraints $x + t = e$, $t \geq 0$. In the end we thus have a problem with variables $\bar{x} = (x; s; t)$ and equality constraints $\bar{A}\bar{x} = \bar{d}$, where

$$\bar{A} = \begin{pmatrix} A & I & 0 \\ I & 0 & I \end{pmatrix}, \quad \bar{d} = \begin{pmatrix} b \\ e \end{pmatrix}.$$

For each of the above equality constraints we include the original constraint as well as the "squared" constraint from (7). Finally, the binary conditions are incorporated via the added equality constraints $x = \text{diag}(X)$; note that these constraints are not considered in any way by the MILP$_{\mathcal{COP}(\mathcal{K})}$ cut-generation procedure since they do not correspond to linear constraints in the original variables.

For our numerical experiments we considered two different methodologies for randomly generating the objective coefficients Q and q. For the first (Type I), $\tilde{Q} = (\hat{Q} + \hat{Q}^T)/2$, where $\hat{Q}$ is an upper triangular matrix with elements uniformly chosen from $\{-1, 0, 1\}$ for $1 \leq i \leq j \leq n$, and the elements of q are similarly uniformly chosen from $\{-1, 0, 1\}$. For the second (Type II), $Q = (\hat{Q} + \hat{Q}^T)$, where $\hat{Q}$ is an upper triangular matrix with elements uniformly chosen on the interval $[-5, 5]$ for $1 \leq i \leq j \leq n$, and the elements of q are similarly uniformly chosen on $[-5, 5]$. For the constraints we iteratively generated coefficients with the coefficient of the first variable always equal to one, and the rest of the coefficients equal to zero with probability 60%, 1 with probability 20% and -1 with probability 20%. For the right-hand side b_i we drew a random number between zero and $\max(a_i^T e, 0)$ where a_i^T is the ith row of A. After each constraint was generated the problem was checked for feasibility, and if it became infeasible the constraint was discarded and the process repeated until a feasible problem with the desired number of constraints was obtained. We also discarded problems where the feasible set was reduced to a singleton.

Table 3: Results on BQP problems

Instance			**Solved**			**Ave. Time (sec)**		
n	m	Type	Gurobi	COP-OA	DNN	Gurobi	COP-OA	DNN
5	1	I	5	5	5	0.0802	0.4589	0.0872
5	1	II	5	5	4	0.0784	0.5677	0.0960
5	2	I	5	5	5	0.0798	0.4604	0.0874
5	2	II	5	5	3	0.0790	0.5504	0.0924
10	2	I	5	5	5	0.0922	1.2223	0.1164
10	2	II	5	5	3	0.0840	1.4727	0.1272
10	4	I	5	5	3	0.0938	1.3318	0.1578
10	4	II	5	5	2	0.0898	1.3546	0.1610
15	2	I	5	5	3	0.0912	14.8099	0.2798
15	2	II	5	5	0	0.0906	27.0382	0.3318
15	5	I	5	5	4	0.0952	79.4416	0.2882
15	5	II	5	5	0	0.0942	95.5523	0.4078

In Table 3 we summarize results for problems of the two types and several choices of (n, m). For each problem type and value of (n, m) we generated 5 instances and attempted to solve each instance using GuRoBi and our COP-OA algorithm. For each instance we also computed the value for the DNN relaxation of CPOPT. As seen in Table 3, Gurobi and the COP-OA algorithm successfully solve all of the instances. The number of exact values for the DNN relaxation decreases with n, and is also always lower for the Type II problems; most notably the DNN relaxation is not exact for any of the Type II problems with $n = 15$. Although the COP outer approximation algorithm successfully solves all of the problems, it is clear from Table 3 that it is much slower than GuRoBi on these problems, and that the time for the COP-OA algorithm is growing rapidly in the problem size. It is worthwhile to note that for these instances the COP-OA algorithm never required more than 3 cuts to converge, and therefore we can conclude that the MILP$_{\mathcal{COP}(\mathcal{K})}$ cut-generation problems were substantially harder for GuRoBi to solve than the original BQP problems themselves. In this regard it is important to note that advanced MILP codes such as GuRoBi utilize considerable logic to identify constraint structure and then automatically add valid inequalities to strengthen the LP relaxation of problems with discrete variables. For example, in the BQP problems inequalities such as triangle inequalities will be automatically generated, and these inequalities will dramatically reduce the amount of branching that would otherwise be required. On the other hand, problems such as MILP$_{\mathcal{COP}(\mathcal{K})}$ may not have constraint structure that can be readily exploited in this way.

5 Application to robust newsvendor problems

In this section we consider a robust multi-item newsvendor problem:

$$\begin{aligned}
&\max_{x\in\mathbb{R}^n}\min_{u\in\mathcal{U}}\sum_{i=1}^{n}(r_i-c_i)x_i-\min\left\{r_i(x_i-u_i),s_i(u_i-x_i)\right\}\\
=&\max_{x\in\mathbb{R}^n}\min_{u\in\mathcal{U}}\sum_{i=1}^{n}\max\left\{(r_i-c_i)x_i-r_i(x_i-u_i),(r_i-c_i)x_i-s_i(u_i-x_i)\right\}
\end{aligned}$$

where $x, r, c, u, s,$ all in $\mathbb{R}^n$ are vectors of quantities of stocked goods, revenues, costs, uncertain demand and shortage costs, respectively. We assume that the uncertainty set $\mathcal{U}$ is constructed via the following factor model

$$\begin{aligned}
\mathcal{U}&=\left\{u\in\mathbb{R}^n\colon \begin{array}{l} u=\bar{u}+\operatorname{Diag}(\hat{u})\,Fz,\\ z\in\mathbb{R}^n,\|z\|_\infty\leq 1,\|z\|_1\leq\rho\end{array}\right\}=\left\{u\in\mathbb{R}^n\colon \begin{pmatrix}1\\u\end{pmatrix}\in\mathcal{K}\right\},\\
\mathcal{K}&=\left\{\begin{pmatrix}u_0\\u\end{pmatrix}\in\mathbb{R}^{n+1}\colon \begin{array}{l} u=u_0\bar{u}+\operatorname{Diag}(\hat{u})\,Fz,\\ z\in\mathbb{R}^n,\|z\|_\infty\leq u_0,\|z\|_1\leq u_0\rho\end{array}\right\}.
\end{aligned}\tag{9}$$

In (9) the vectors $\bar{u}$ and $\hat{u}$ are given, as are ρ and the matrix F; $\bar{u}$ can then be seen as an average value of the uncertainty parameter around which the uncertainty set is constructed. We also assume F to be a matrix of full rank.

This problem was studied in [4] and later treated by [27] in the context of copositive optimization. The problem has an adjustable robust optimization (ARO) reformulation since the inner-most max problem is a function of the uncertainty parameter:

$$\begin{aligned}
\max_{x,y(\cdot)}\min_{u\in\mathcal{U}}&\sum_{i=1}^{n}y_i(u)\\
&y_i(u)\leq(r_i-c_i)x_i-r_i(x_i-u_i),\quad u\in\mathcal{U},\ i=1,\dots,n,\\
&y_i(u)\leq(r_i-c_i)x_i-s_i(u_i-x_i),\quad u\in\mathcal{U},\ i=1,\dots,n.
\end{aligned}\tag{10}$$

There are several known ways to approach (10) via copositive optimization techniques, most notably those discussed in [26, 27]. The former article presents an exact copositive reformulation while the latter investigates copositive reformulations of the so-called quadratic policy approximation. The quadratic policy approximation is itself inexact, and is further conservatively approximated using an inner approximation of the copositive cone. Hence, there are two sources of approximation errors and so far their individual respective contributions to the total error has not been investigated either analytically or empirically. Since we can now solve both the exact reformulation and the quadratic policy approximation using our cutting plane algorithm, while the conservative approximation of the quadratic policy can be solved with conic optimization solvers, we have the tools to examine these errors empirically for the first time.

In the following we will give a short review of the main techniques introduced in [26, 27], applying them to the ARO reformulation of the robust newsvendor problem. After that, we present numerical experiments investigating the gaps between the different approaches.

5.1 Three approaches based on copositive optimization

In [27] the authors considered conservative approximations, where the second-stage functions $y(\cdot)$ are restricted to the space of quadratic functions, so that

$$y(u) = \begin{pmatrix} u^T Y_1 u + y_1^T u + \gamma_1 \\ \vdots \\ u^T Y_n u + y_n^T u + \gamma_n \end{pmatrix}.$$

This restriction is called the *quadratic policy* or *quadratic decision rule* since the second-stage response to the outcome of the uncertain process is modeled as a function that is quadratic in the uncertainty parameter. After applying this quadratic policy, the constraints in (10) are given by

$$\begin{aligned} u^T Y_i u + y_i^T u + \gamma_i &\leq (r_i - c_i)x_i - r_i(x_i - u_i), \quad u \in \mathcal{U},\ i = 1, \ldots, n, \\ u^T Y_i u + y_i^T u + \gamma_i &\leq (r_i - c_i)x_i - s_i(u_i - x_i), \quad u \in \mathcal{U},\ i = 1, \ldots, n, \end{aligned}$$

which can be reformulated into set-copositive constraints

$$\begin{aligned} &\lambda_i \geq 0,\ \begin{pmatrix} -c_i x_i - \gamma_i & \frac{1}{2}(r_i e_i - y_i)^T \\ \frac{1}{2}(r_i e_i - y_i) & -Y_i \end{pmatrix} - \lambda_i e_1 e_1^T \in \mathcal{COP}(\mathcal{K}), \quad i = 1, \ldots, n, \\ &\mu_i \geq 0,\ \begin{pmatrix} (r_i - c_i + s_i)x_i - \gamma_i & -\frac{1}{2}(s_i e_i + y_i)^T \\ -\frac{1}{2}(s_i e_i + y_i) & -Y_i \end{pmatrix} - \mu_i e_1 e_1^T \in \mathcal{COP}(\mathcal{K}), \quad i = 1, \ldots, n. \end{aligned}$$

The objective function in (10) can be reformulated similarly after it is cast as a constraint by means of introducing an epigraphical variable.

In [27] the authors proposed a conservative and therefore inner approximation of these constraints. The COP-OA cutting plane algorithm, however, lets us solve the resulting model exactly so that we are now able to asses the quality of the conservative approximation by comparing its performance with the results from the exact evaluation of the quadratic policy. Details of applying our approach, as well as formulating the conservative approximation, are summarized in Appendix A.

We now discuss a third and final approach to (10) from [26], where the authors derive an exact copositive reformulation of a form that can also be solved via our cutting plane algorithm. Thus, the algorithm allows us to do both: solve (10) exactly via the exact copositve reformulation from [26], but also solve the approximations of (10) based on quadratic decision rules exactly, without the need to employ a further approximation of $\mathcal{COP}(\mathcal{K})$. Thus, we can assess the gap introduced by the quadratic policy in isolation for the first time.

To establish the exact reformulation, we observe that (10) is of the form

$$\max_{x,y(\cdot)} \left\{ \min_{\bar{\mathcal{U}}} \left\{ e^T y(u) \right\} : Ax + By(u) \geq Cu, \ \forall u \in \bar{\mathcal{U}} \right\}, \tag{11}$$

with

$$A = \begin{pmatrix} -\operatorname{Diag}(c) \\ \operatorname{Diag}(r-c+s) \end{pmatrix}, \ B = \begin{pmatrix} -I \\ -I \end{pmatrix}, \ C = \begin{pmatrix} -\operatorname{Diag}(r) \\ \operatorname{Diag}(s) \end{pmatrix} (\bar{u} + \operatorname{Diag}(\hat{u})F),$$

and $\bar{\mathcal{U}} = \left\{ u \in \bar{\mathcal{K}} \colon e_1^T u = 1 \right\}$ where $\bar{\mathcal{K}}$ is defined as in Appendix A.

The following equivalences hold by a classical argument in robust optimization (see [6]) and by strong duality in linear optimization:

$$\begin{aligned} &\max_{x,y(u)} \left\{ \min_{u \in \bar{\mathcal{U}}} \left\{ e^T y(u) \right\} : Ax + By(u) \geq Cu, \ \forall u \in \bar{\mathcal{U}} \right\} \\ &= \max_x c^T x + \min_{u \in \bar{\mathcal{U}}} \max_y \left\{ e_1^T u e^T y \colon Axe_1^T u + By \geq Cu \right\} \\ &= \max_x c^T x + \min_{(u,w) \in \bar{\mathcal{U}} \times \mathbb{R}^m_+} \left\{ w^T \left(C + Axe_1^T \right) u \colon B^T w = ee_1^T u \right\}. \end{aligned}$$

By the same argument used in [26] we can add redundant constraints $||(u;w)|| \leq \pi$ for some large enough $\pi \in \mathbb{R}$. The feasible set of the inner minimization problem can be written as

$$\begin{aligned} \mathcal{F} &= \left\{ \begin{pmatrix} u \\ w \end{pmatrix} \in \hat{\mathcal{K}} \colon e_1^T u = 1, \ \left\| \begin{matrix} u \\ w \end{matrix} \right\| \leq \pi \right\}, \\ \hat{\mathcal{K}} &= \left\{ \begin{pmatrix} u \\ w \end{pmatrix} \in \bar{\mathcal{K}} \times \mathbb{R}^{2n}_+ \colon B^T w = ee_1^T u \right\} = \left\{ \begin{pmatrix} u \\ w \end{pmatrix} \in \mathbb{R}^{n+1} \times \mathbb{R}^{2n}_+ \colon B^T w = ee_1^T u, \ Pu \geq 0 \right\}, \end{aligned}$$

where P is defined as in Appendix A, i.e. $Pu \geq 0$ encodes the ∞-norm and the 1-norm constraints in $\bar{\mathcal{K}}$. Then, the problem $v(x) = \min_{(u,w) \in \mathcal{F} \times \mathbb{R}^m_+} w^T \left(C + Axe_1^T \right) u$ has an exact completely positive reformulation whose dual is given by

$$\begin{aligned} &\max_{\lambda,\Lambda,\rho} \lambda + \pi\mu \\ &\text{s.t.}: \begin{pmatrix} 0 & \frac{1}{2}\left(C + Axe_1^T \right)^T \\ \frac{1}{2}\left(C + Axe_1^T \right) & 0 \end{pmatrix} + \lambda e_1 e_1^T + \mu I \in \mathcal{COP}\left(\hat{\mathcal{K}} \right), \end{aligned}$$

and its optimal value is equal to $v(x)$ since μI can be scaled to give a Slater point. Hence, we can solve $\max_{x \in \mathbb{R}^n} c^T x + v(x)$ as a copositive optimization problem. As noted in [26], strong duality may fail to hold in case we do not create a Slater point in the dual by introducing the redundant constraint to bound the conic primal. However, in any given iteration of our algorithm, we only solve linear approximations of that problem

Instances			Average Gap (%)			Num. Opt.	
n	ρ	v	ex-quad	ex-cons	quad-cons	quad	cons
2	1.0	10	0.000	0.000	0.000	10	10
2	1.0	50	0.000	0.000	0.000	10	10
2	1.5	10	-0.011	0.014	0.024	10	9
2	1.5	50	0.000	0.000	0.000	10	10
3	1.0	10	0.001	0.000	0.000	9	9
3	1.0	50	0.000	0.000	0.000	10	10
3	2.5	10	-0.009	0.025	0.034	9	5
3	2.5	50	0.056	0.074	0.018	8	4
4	1.0	10	0.002	0.000	-0.002	10	10
4	1.0	50	0.001	0.003	0.002	9	9
4	2.5	10	0.131	0.266	0.135	4	0
4	2.5	50	0.638	1.356	0.708	2	0

Table 4: Results for newsvendor problems

for which strong duality does not hinge on the existence of a Slater point, and therefore the term μI can be omitted. Again, membership in $\mathcal{COP}(\hat{\mathcal{K}})$ can be certified via our set-copositivity test via a reformulation similar to the one used for certifying $\mathcal{COP}(\mathcal{K})$. In order to execute this strategy we employ similar techniques to the ones used when reformulating the quadratic policy, detailed in Appendix A, and we omit the details here.

5.2 Computational experiments

For our experiments we chose the following specifications for the problem data

$$r = 80e,\ s = 60e,\ c = 40e + 20\tilde{c},\ \bar{u} = 60e,\ \hat{u} = ve,$$

where $\tilde{c}_i \sim$ Uniform$[0,1]$ for each i, and the parameter $v \in \mathbb{R}$ was varied across the instances. For the matrix F we generated a random matrix $\bar{F}$ with $\bar{F}_{ij} \sim$ Uniform$[0,2]$ for each i,j, and populated the rows of F with the rows of $\bar{F}$ normalized by their 1-norm. The impact of F on the uncertainty set is governed by the scaling via $\hat{u}$, which we controlled via v. The parameter ρ, which controls the shape of the uncertainty set, was also varied. Note that $\rho = 1$ eliminates the ∞-norm component from the description of $\mathcal{U}$, since in this case the respective 1-norm ball is contained in the respective ∞-norm ball. For $n > \rho > 1$ both components are relevant.

In Table 4 we have summarized the results of our experiments. The instances considered are organized by the values of (n, ρ, v). For each of these configurations 10 instances were generated and the respective rows summarize the results over these instances. All instances of the exact and quadratic policy models were successfully solved using the COP-OA algorithm. In the columns headed by "Average Gap" we report the gap between the exact optimal solutions and the solutions obtained from solving the quadratic policy approximation exactly ("ex-quad") the gap from the solution of the conservative approximation of the quadratic policy model ("ex-cons") and finally the gap between

Instances			Average Time (sec)			Ave. Iter.	
n	ρ	v	exact	quad	cons	exact	quad
2	1.0	10	0.307	7.412	0.095	1.0	7.8
2	1.0	50	0.299	8.546	0.094	1.0	8.5
2	1.5	10	1.281	8.361	0.090	4.3	8.8
2	1.5	50	1.342	9.326	0.089	4.7	9.6
3	1.0	10	0.655	26.021	0.122	1.0	13.5
3	1.0	50	0.528	33.116	0.139	1.0	14.0
3	2.5	10	2.895	46.748	0.185	6.1	20.5
3	2.5	50	2.662	48.610	0.152	7.1	24.6
4	1.0	10	7.301	391.577	0.315	1.0	18.1
4	1.0	50	7.964	286.086	0.358	1.0	21.3
4	2.5	10	21.515	526.407	0.486	8.2	31.5
4	2.5	50	16.696	430.150	0.437	8.8	45.7

Table 5: Time and iterations to solve newsvendor problems

the exact solution under the quadratic policy and its conservative approximation ("quad-cons"). For all of these values we present the avarage value of the percentage gaps over the ten instances we created. The final three columns, headed by "Num. Opt.", count the number of times the quadratic model ("quad") and its conservative approximation ("cons"), respectively, gave the exact optimal value within numerical tolerances.

Several features of these results are worth discussing in greater detail. First, the quality of both the quadratic policy and its conservative approximation deteriorate with dimension. This is, secondly, especially the case for instances where the structure of the uncertainty set is more complicated, i.e. where both norm constraints are relevant. Thirdly, the quality of the conservative approximation of the quadratic model is still quite good at least in the instances we considered. This demonstrates that comparatively cheap approximations of set-copositive constraints can still yield satisfactory results. Finally, the small but theoretically impossible negative values in the first three columns are a testament to the numerical instabilities that are attached to using the COP-OA algorithm to compute the exact and quadratic policy values, and also the interior-point solver used to compute values for the conservative approximation. For example, numerical degeneracies may develop as cutting planes are added in the COP-OA algorithm, and the indicator for set-copositivity γ in MILP$_{\mathcal{COP}}(\mathcal{K})$ may be close to zero for a super-optimal solution. These issues are especially taxing for the quadratic policy instances, since many set-copositivity tests and subsequent cutting planes are implemented on every iteration of the algorithm. For the purpose of reporting the number of optimal values in the final two columns we consider small negative gaps to correspond to optimal solutions.

In Table 5 we give the average times and number of iterations required for the same problems considered in Table 4. As we can see, the exact reformulation was much easier to solve for our algorithm than the quadratic policy model. The reason is that the latter requires testing set-copositivity for $2n$ matrix blocks of order $n+1$ ($4n+k+2$ after the reformulation described in the Appendix) on every iteration, while for the exact model only a single test per iteration has to be performed where the order of the matrix block is $3n+1$ ($4n+k+2$ after reformulation). In addition, the larger number of copositive matrix blocks increased the number of iterations needed substantially for the quadratic policy. Hence solving the quadratic policy exactly does not yield much benefit other than certifying the quality of the conservative approximation, at least for our solution approach.

6 Conclusion

In this paper we have described an algorithm for nonconvex quadratic problems that can be formulated as completely positive (CPOPT) optimization problems, whose duals are copositive optimization (COOPT) problems. Our method iteratively strengthens an outer approximation of the set-copositive cone that incorporates linear constraints from the original CPOPT problem. We extend a previous MILP-based method for generating a cut that separates a matrix from the copositive cone to incorporate such linear constraints. Computational results show that the method is capable of globally solving small nonconvex problems to optimality, with the computational bottleneck being the MILP-based separation routine. This observation suggests that modifications to the separation problem that make it more computationally tractable would be very beneficial. In addition, CPOPT formulations often include linear constraints that combine the original variables x and lifted matrix variavbles X; examples include constraints obtained from the reformulation-linearization technique (RLT) and constraints of the form $x = \mathrm{diag}(X)$ for binary x. An extension of our methodology that could incorporate such constraints in the separation routine would be an attractive enhancement.

Acknowledgement

Research for this paper was begun when one author (Anstreicher) was visiting the Vienna Center for Operations Research (VCOR) at the University of Vienna, Austria, where the other author (Gabl) was completing his dissertation. Support from the VCOR and especially Prof. Immanuel Bomze is gratefully acknowledged.

References

[1] Kurt M. Anstreicher. Towards a practical volumetric cutting plane method for convex programming. *SIAM J. Optim.*, 9:190–206, 1999.

[2] Kurt M. Anstreicher. Testing copositivity via mixed-integer linear programming. *Linear Algebra Appl.*, 609:218–230, 2021.

[3] Kurt M. Anstreicher, Samuel Burer, and Peter J.C. Dickinson. An algorithm to compute the CP-factorization of a completely positive matrix. *Oberwolfach Rep.*, 52:3079–3081, 2017.

[4] Amir Ardestani-Jaafari and Erick Delage. Linearized robust counterparts of two-stage robust optimization problems with applications in operations management. *INFORMS J. Comput.*, 33(3):1138–1161, 2020.

[5] Riley Badenbroek and Etienne de Klerk. An analytic center based cutting plane method to determine complete positivity of a matrix. *INFORMS J. Comput.*, To appear, 2021.

[6] Aharon Ben-Tal, Alexander Goryashko, Elana Guslitzer, and Arkadi Nemirovski. Adjustable robust solutions of uncertain linear programs. *Math. Prog.*, 99(2):351–376, 2004.

[7] Immanuel Bomze and Markus Gabl. Interplay of non-convex quadratically constrained problems with adjustable robust optimization. *Math. Meth. Oper. Res.*, 93:115–151, 2021.

[8] Immanuel M Bomze and Markus Gabl. Optimization under Uncertainty and Risk: Quadratic and Copositive Approaches. *Euro. J. Oper. Res.*, 310(2):449–476, 2023.

[9] Samuel Burer. On the copositive representation of binary and continuous nonconvex quadratic programs. *Math. Prog.*, 120:479–495, 2009.

[10] Samuel Burer. Copositive programming. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 201–218. Springer, 2012.

[11] Peter J. C. Dickinson and Luuk Gijben. On the computational complexity of membership problems for the completely positive cone and its dual. *Comput. Optim. Appl.*, 57:403–415, 2014.

[12] Peter J.C. Dickinson. *The Copositive Cone, the Completely Positive Cone and their Generalizations.* Ph.D. thesis, University of Groningen, 2013.

[13] Hongbo Dong. Symmetric tensor approximation hierarchies for the completely positive cone. *SIAM J. Optim.*, 23:1850–1866, 2013.

[14] Hongbo Dong and Kurt Anstreicher. Separating doubly nonnegative and completely positive matrices. *Math. Prog.*, 137:131–153, 2013.

[15] Gabriele Eichfelder and Janez Povh. On the set-semidefinite representation of nonconvex quadratic programs over arbitrary feasible sets. *Optim. Letters*, 7(6):1373–1386, 2013.

[16] Jean-Louis Goffin, Zhi-Quan Luo, and Yinyu Ye. Complexity analysis of an interior cutting plane method for convex feasibility problems. *SIAM J. Optim.*, 6(3):638–652, 1996.

[17] Jacek Gondzio and E. Alper Yildirim. Global solutions of nonconvex standard quadratic programs via mixed integer linear programming reformulations. Technical report, University of Edinburgh, Edinburgh, UK, 2018.

[18] João Gouveia, Ting Kei Pong, and Mina Saee. Inner approximating the completely positive cone via the cone of scaled diagonally dominant matrices. *J. Global Optim.*, pages 383–405, 2020.

[19] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization.* Springer-Verlag (Berlin), 1988.

[20] Cheng Guo, Merve Bodur, and Joshua A. Taylor. Copositive duality for discrete energy markets. Technical report, University of Toronto, 2021.

[21] Sunyoung Kim and Masakazu Kojima. Exact solutions of some nonconvex quadratic optimization problems via SDP and SOCP relaxations. *Comput. Optim. Appl.*, 26:143–154, 2003.

[22] Johan Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[23] Javier Peña, Juan Vera, and Luis F. Zuluaga. Computing the stability number of a graph via linear and semidefinite programming. *SIAM J. Optim.*, 18:87–105, 2007.

[24] Ralph Tyrell Rockafellar. *Convex Analysis.* Princeton University Press, 2015.

[25] Wei Xia, Juan Vera, and Luis F. Zuluaga. Globally solving nonconvex quadratic programs via linear integer programming techniques. *INFORMS J. Computing*, 32:40–56, 2020.

[26] Guanglin Xu and Samuel Burer. A copositive approach for two-stage adjustable robust optimization with uncertain right-hand sides. *Comput. Optim. Appl.*, 70(1):33–59, 2018.

[27] Guanglin Xu and Grani A Hanasusanto. Improved decision rule approximations for multi-stage robust optimization via copositive programming. *Oper. Res.*, to appear, 2023.

A Reformulating the quadratic policy model

In this appendix we descibe details necessary to apply our COP-OA algorithm to the quadratic policy model, and to compute the conservative approximation of the quadratic policy. First, we discuss how to construct the conservative approximation. Note that

$$\mathcal{K} = \left\{ \begin{pmatrix} u_0 \\ u_0\bar{u} + \operatorname{Diag}(\hat{u})\, Fz, \end{pmatrix} : \ z \in \mathbb{R}^n, \|z\|_\infty \le u_0, \|z\|_1 \le u_0\rho \right\} = M\bar{\mathcal{K}}$$

with

$$\bar{\mathcal{K}} = \left\{ \begin{pmatrix} u_0 \\ z, \end{pmatrix} : \ z \in \mathbb{R}^n, \|z\|_\infty \le u_0, \|z\|_1 \le u_0\rho \right\}, \quad M = \begin{pmatrix} 1 & o^T \\ \bar{u} & \operatorname{Diag}(\hat{u})\, F \end{pmatrix}$$

Then, from the definition of set-copositive matrix cones we get that

$$\mathcal{COP}(\mathcal{K}) = \mathcal{COP}\left(M\bar{\mathcal{K}}\right) = M^{-\mathsf{T}}\mathcal{COP}\left(\bar{\mathcal{K}}\right) M^{-1}$$

so that we can focus on the simpler cone $\mathcal{COP}\left(\bar{\mathcal{K}}\right)$. Following the approach in [27], we can approximate this cone via the inner approximation

$$\left\{S + P^T W P \colon S \in \mathcal{S}_+^{n+1},\ W \geq 0\right\} \subseteq \mathcal{COP}\left(\bar{\mathcal{K}}\right),$$

where $P \in \mathbb{R}^{k\times(n+1)}$ is a matrix so that

$$P\begin{pmatrix} u_0 \\ z \end{pmatrix} \geq 0 \Leftrightarrow \|z\|_\infty \leq u_0, \|z\|_1 \leq u_0\rho,$$

which exists since both norms are polyhedral convex functions (see e.g. [24, Corollary 19.1.2.]).

In order to apply our cutting plane algorithm, to solve the quadratic policy model exactly, we need to be able to test whether $S \in \mathcal{COP}\left(\mathcal{K}\right)$. There are two adaptations we have to make since, firstly, $(u_0; u) \in \mathbb{R}^{n+1}$ is not restricted to be nonnegative and, secondly, $\mathcal{K}$ is described in terms of equations and inequalities rather than just equations. However, it is easy to see that $v^T S v \geq 0\ \forall v \in \mathcal{K}$ if and only if

$$\begin{pmatrix} u_+ \\ u_- \end{pmatrix}\begin{pmatrix} S & -S \\ -S^T & S \end{pmatrix}\begin{pmatrix} u_+ \\ u_- \end{pmatrix} \geq 0$$

for all (u_+, u_-) such that

$$\begin{aligned} u_+ - u_- \quad &= \quad (u_0^+ - u_0^-)\bar{u} + \mathrm{Diag}\left(\hat{u}\right)F(z_+ - z_-), \\ P\begin{pmatrix} u_0^+ - u_0^- \\ z_+ - z_- \end{pmatrix} \quad &= \quad p, \\ (u_0^+; u_+) \quad &\in \quad \mathbb{R}_+^{n+1}, (u_0^-; u_-) \in \mathbb{R}_+^{n+1},\ z_+ \in \mathbb{R}_+^n,\ z_- \in \mathbb{R}_+^n,\ p \in \mathbb{R}_+^k, \end{aligned}$$

where the latter conditions describe a cone of the required form for our set-copositivity algorithm testing to be applicable.