



ELSEVIER

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Continuous Optimization

Two-stage stochastic standard quadratic optimization

Immanuel M. Bomze^{a,*}, Markus Gabl^a, Francesca Maggioni^b, Georg Ch. Pflug^a^aVienna Center of Operations Research (VCOR), University of Vienna, Oskar-Morgenstern-Platz 1, Vienna 1090, Austria^bDepartment of Management, Information and Production Engineering, University of Bergamo, Via Marconi n. 5, Dalmine, Bg 24044, Italy

ARTICLE INFO

Article history:

Received 29 December 2020

Accepted 27 October 2021

Available online xxx

Keywords:

Stochastic optimization

Standard quadratic problem

Non-convex optimization

Copositive optimization

ABSTRACT

Two-stage stochastic decision problems are characterized by the property that in stage one part of the decision has to be made before relevant data are observed and the rest of the decision can be made in stage two after data are available. It is generally assumed that while the data missing in stage one are not known, their probability distribution is known. In this paper, we consider a special form of such problems where the objective function is quadratic in the first and second stage decisions and the goal is to minimize the expected quadratic cost function. We assume that the decisions must lie in a standard simplex, but do not require convexity of the uncertain objective. It is well known that such quadratic optimization programs are NP-complete and thus belong to the most complex optimization problems. A viable method to attack such problems is to find bounds for the original complicated problem by solving easier approximate problems. We describe such approximations and show that by exploiting their properties, good but not necessarily optimal solutions can be found. We also present possible applications, such as selecting invited speakers for conferences with the aim to generate a community with high coherence. Finally, systematic numerical results are provided.

© 2021 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

1. Background and motivating examples

Consider the following, possibly non-convex, *deterministic standard quadratic optimization problem*, which is referred to as DStQP:

$$z_{\det}^* := \min \{q(z) := z^T Q z : z \in \Delta^n\}, \quad (1)$$

where Q is an $(n \times n)$ symmetric matrix, and $\Delta^n := \{z \in \mathbb{R}_+^n : e^T z = 1\}$ with $e = [1, \dots, 1]^T \in \mathbb{R}^n$ and T denoting transposition. Note that a linear term $2q^T z$ in the objective $q(z)$ with $q \in \mathbb{R}^n$ can be incorporated easily in homogenous form by replacing Q with the symmetric matrix $Q + eq^T + qe^T$.

Thus, a DStQP consists of finding (global) minimizers of a quadratic form over the standard simplex, so it is a global optimization problem which is in fact NP-hard. Even detection of local solutions is NP-hard (Bomze, 1998). Nevertheless, there are several exact procedures which try to exploit favourable data constellations in a systematic way, and to avoid the worst-case behaviour whenever possible. See, among many others

(Bomze & Palagi, 2005; Gondzio & Yildirim, 2021; Liuzzi, Locatelli, & Piccialli, 2019; Scozzari & Tardella, 2008).

On the other hand, the simplicity of the feasible set (the simplest polytope) and objective function (the simplest non-convex function) justify to view DStQP as the simplest of the hard problems (Bomze, Schachinger, & Ullrich, 2018). Seen from the application point of view, this problem class is rich and versatile enough to allow for modeling a huge variety of problems, ranging from biology through economics to classical discrete problems like the maximum-clique problem (Bomze & de Klerk, 2002; Bomze et al., 2000).

Notation

Throughout the text, matrices are denoted by sans-serif capital letters (e.g. O will be the zero matrix and E will be the matrix of all ones), vectors by boldface lower case letters (for example, o will denote the zero vector, and e the vector of all ones) and scalars (real numbers) by lower case letters. Note that for the zero matrix O , the matrix of ones E , the vector of ones e , the i th column vector e_i of the identity matrix I , the dimension will always be clear from the context. For two $m \times n$ matrices $\{A, B\}$ we use $A \bullet B := \text{trace}(A^T B) = \sum_{i=1}^m \sum_{j=1}^n A_{ij} B_{ij}$ as a shorthand notation for their Frobenius product. For a vector x , the i th entry will be denoted by x_i . For a matrix X the j th entry in the i th row will be

* Corresponding author.

E-mail addresses: immanuel.bomze@univie.ac.at (I.M. Bomze), markus.gabl@univie.ac.at (M. Gabl), francesca.maggioni@unibg.it (F. Maggioni), georg.pflug@univie.ac.at (G.Ch. Pflug).

given as X_{ij} . We denote by S^n the set of symmetric matrices of order n . We also use the shorthand notation $[1:S] := \{1, \dots, S\} \subset \mathbb{N}$.

The data of a DSTQP are fully described by the matrix Q . However, we not always have full information on these parameters. Recent attempts to deal with uncertainty in Q from a robust perspective can be found in [Bomze, Kahr, & Leitner \(2021a\)](#). Here we want to follow a different approach, but before we want to motivate this study by several examples.

Example 1. The organizer of a conference has to decide about whom to classify as invited speaker and which contributed papers will be accepted for oral presentation. It is assumed that the frictions between speaker i and speaker j are given by a real number Q_{ij} . The (relative) importance given to speaker i is denoted by $z_i \geq 0$. The deterministic problem of minimizing the frictions is given by the quadratic problem (1). The set of contributed speakers as well as their frictions among themselves and between them and the invited speakers are not known at the beginning. This requires to replace the matrix Q by an uncertain matrix \tilde{Q} (see [Section 3.1](#) for details) which is dissected into the parts A (frictions among the invited speakers), \tilde{B} (frictions between contributed speakers and invited speakers) and \tilde{C} (frictions between the contributed speakers). The pertaining decision problem for the organisation of a maximally friction-less conference is as described before.

Example 2. A car dealer has to decide about his portfolio of car types in his sales program. He wants to include as different car types as possible in order to cover the market. Denote by Q_{ij} the similarity between type i and type j . As before, the relative size of type i is denoted by $z_i \geq 0$. The problem under full information is again given by problem (1). If however some types are not yet available and their similarities are modeled as a random ones, we arrive at the decision problem described in [Section 3.1](#).

Example 3. A social network contains n members. For each pair (i, j) of members, the value Q_{ij} indicates the degree of dissimilarity. In order to find some representatives, the network owner wants to select a number of persons, which are maximally different from each other. In a relaxed form of this problem, he wants to find coefficients $z_i \geq 0$ such that $z^T Q z$ is maximal. Notice that maximizing $z^T Q z$ is equivalent to minimizing $z^T (-Q) z$, which is contained in model (1), since no assumptions about Q are made. The stochasticity enters the model if we assume that the similarity of some members is quite known, while the information about the similarity of others is not known, but follows some probability distribution, which will be revealed later.

Example 4. The minimal variance portfolio has to be found by investing in two classes of assets. The first group of assets is well known and their covariance matrix is well estimated. A second group of assets is new and their interrelation is not well known. However, part of the investment has to be made immediately, while the rest can be invested later. Notice that due to the lack of information about the second stage assets, it is not guaranteed that the random covariance matrices are almost surely positive-semidefinite. See [Section 5.5](#) for a numerical example.

In this paper, we consider a special form of two-stage stochastic programs which we propose to call *two-stage stochastic standard quadratic optimization problems* (2(St)³QP) where the objective function is quadratic in the first and second stage decisions and the goal is to minimize the expected quadratic cost function. We assume that the decisions must lie in a standard simplex, but do not require convexity of the uncertain objective resulting in a quadratic optimization program which is NP-complete. In order to solve such a difficult decision problem we propose different types of approximations: by bounding the recourse function; by providing an upper bound based on a Pairwise Frank-Wolfe algorithm, an efficient first-order procedure similar to the Infection-Immunitization-Dynamics (IID) algorithm (see [Rota Buló & Bomze,](#)

[2011](#)), applied to the scenario-tree formulation; by providing scalable copositive lower and upper bounds applied to the scenario-tree formulation; by providing lower bounds based on dissecting the probability measure (see [Maggioni & Pflug, 2016](#)).

Finally a computational study shows in a systematic way how the different types of bounds perform in terms of computation time and quality of the solution.

The paper is organized as follows. In [Section 2](#), basic facts on two-stage stochastic optimization problems are recalled and a literature review on their bounds provided. In [Section 3](#) we introduce the class of two-stage standard quadratic optimization problems and bounds on their recourse function. Bounds on the scenario problem are provided in [Section 4](#) by means of upper bounds based on the Infection-Immunitization-Dynamics (IID) algorithm, scalable copositive lower and upper bounds, and lower bounds based on dissecting the probability measure. In [Section 5](#) numerical results are provided, corroborating our approach. Conclusions follow.

2. Two-stage stochastic optimization: basic facts

The following mathematical model represents a general formulation of a two-stage stochastic optimization problem in which a decision maker needs to determine x in order to minimize expected total costs ([Birge & Louveaux, 2011](#); [King & Wallace, 2012](#); [Pflug & Pichler, 2014](#); [Ruszczynski & Shapiro, 2003](#)):

$$\min_{x \in X} f(x) = \min_{x \in X} \left\{ f_1(x) + \mathbb{E}_\xi [h_2(x, \xi)] \right\}, \quad (2)$$

where x is a first-stage decision variable restricted to the set $X \subseteq \mathbb{R}_+^{n_1}$, and \mathbb{E}_ξ stands for the expectation with respect to a random vector ξ , defined on some probability space $(\Xi, \mathcal{A}, \mathbb{P})$ with support Ξ and given probability distribution \mathbb{P} on the σ -algebra \mathcal{A} . The function h_2 is the value function of the second stage optimization problem defined for a specific realization ξ of the random vector ξ as

$$h_2(x, \xi) := \min_{y \in Y(x, \xi)} q(y; x, \xi), \quad (3)$$

which models the minimization of the cost $q(y; x, \xi)$ associated with adapting to information revealed through the realization ξ which also determines the feasible set $Y(x, \xi)$. The term $r(x) := \mathbb{E}_\xi [h_2(x, \xi)]$ in (2) is referred to as the *recourse function*. Introducing the function $z(x, y, \xi) := f_1(x) + q(y; x, \xi)$, problem (2) can be rewritten as follows:

$$z_{\text{stoch}}^* := \min_{x \in X} \mathbb{E}_\xi \left[\min_{y \in Y(x, \xi)} z(x, y, \xi) \right], \quad (4)$$

and its solution x^* is called the *here and now solution*.

This class of problems brings computational complexity which increases exponentially with the size of the *scenario tree*, representing a discretization of the underlying random process $\xi \in \Xi$. Even if a large discrete tree model is constructed, the problem might be untractable due to the curse of dimensionality. Several are the bounds and approximations proposed in the literature to cope with this problem.

Among them, [Frauendorfer \(1988\)](#), [Hausch & Ziemba \(1983\)](#), [Huang, Vertinsky, & Ziemba \(1977a\)](#), [Huang, Ziemba & Ben-Tal \(1997b\)](#) generalize Jensen's inequality ([Jensen, 1906](#)) for lower bounding and the [Edmundson-Madansky \(Edmundson, 1956; Madansky, 1960\)](#) inequality for upper bounding. An alternative method is to aggregate constraints and variables in the extensive-form and solve the resulting problem ([Birge, 1985](#); [Rosa & Takriti, 1999](#)). Easy-to-compute bounds have been also proposed in literature by solving small size sub-problems instead of the big one associated with the large discrete scenario tree model (see for instance [Kuhn, 2008](#); [Maggioni, Allevi, & Bertocchi, 2014](#);

Maggioni, Allevi, & Bertocchi, 2016; Maggioni & Pflug, 2016; Sandıkçı, Kong, & Schaefer, 2013).

An alternative approach is to construct two approximating trees, a lower tree and an upper tree, the solution of which lead to upper and lower bounds for the optimal value of the original *continuous* problem. The advantage of this approach is that it generates intervals in which the optimal value lies under guarantee. Results in this direction were for the first time obtained by Frauendorfer & Schürle (2009), followed by Frauendorfer, Kuhn, & Schürle (2011), Kuhn (2005). In Kuhn (2005), barycentric discretizations are adopted in a more general setting investigating convex multistage stochastic programs with a generalized non-convex dependence on the random variables. In Maggioni & Pflug (2019), the authors generalize the bounding ideas of Frauendorfer & Schürle (2009), Frauendorfer et al. (2011), Kuhn (2005) to not necessarily Markovian scenario processes and derive valid lower and upper bounds for the convex case. They construct new discrete probability measures directly from the simulated data of the whole scenario process.

In the following we will present, for the new class (2(St)³QP), new lower and upper bounds to their objective function value that are scalable.

3. Two-stage stochastic standard quadratic optimization (2(St)³QP)

3.1. Problem formulation and recourse function

What happens if the problem data of a DStQP are uncertain? While the constraints in (DStQP) are out of dispute as probabilities or portfolio shares always are non-negative and sum up to one, the only variable problem data are to be found in the matrix \tilde{Q} . Now suppose a (possibly) small $n_1 \times n_1$ principal submatrix A of \tilde{Q} is known (more or less) exactly, whereas the rest of the problem data are subject to uncertainty with known probability distribution:

$$\tilde{Q} := \begin{bmatrix} A & \tilde{B}^\top \\ \tilde{B} & \tilde{C} \end{bmatrix}.$$

Here $\xi := [\tilde{B}, \tilde{C}]$ are the uncertain data. Decomposing $z^\top := [x^\top, y^\top]$ with $x \in \mathbb{R}_+^{n_1}$ and $y \in \mathbb{R}_+^{n_2}$, we arrive at the following problem reformulation under data uncertainty:

$$z_{\text{stoch}}^* := \min_{x \in T^{n_1}} \{x^\top Ax + r(x)\} \quad (5)$$

with

$$T^{n_1} = \text{conv} \{o, e^i : i \in [1:n_1]\} = \text{conv} (\Delta^{n_1} \cup \{o\}) \\ = \{x \in \mathbb{R}_+^{n_1} : e^\top x \leq 1\}$$

and the recourse function given by

$$r(x) = \mathbb{E}_\xi \left[\min_{y \in P_x} (2x^\top \tilde{B}^\top y + y^\top \tilde{C} y) \right] \quad \text{with} \quad P_x : \\ = \{y \in \mathbb{R}_+^{n_2} : e^\top y = 1 - e^\top x\}.$$

As announced, we refer to problem (5) as the *two-stage Stochastic StQP* (2(St)³QP).

If we denote by $y(x; \xi)$ a minimizer of the random (via ξ) quadratic optimization problem $\min_{y \in P_x} (2x^\top \tilde{B}^\top y + y^\top \tilde{C} y)$ parametrized by x , the recourse function can also be written as

$$r(x) = \mathbb{E}_\xi [2x^\top \tilde{B}^\top y(x; \xi) + y(x; \xi)^\top \tilde{C} y(x; \xi)].$$

Typically, the expression inside the expectation brackets is neither linear in ξ nor quadratic in x because of the dependence of $y(x; \xi)$ on both x and ξ . Before we proceed to attack this problem, let us note some bounds for the recourse function.

3.2. Bounding the recourse function

Simple closed-form upper bounds for $r(x)$ are obtained by looking at their counterparts for (DStQP), the most elementary being

$$z_{\text{det}}^* \leq \min_i Q_{ii}.$$

Defining $\bar{y} = \frac{1}{1-e^\top x} y$, we have that the recourse function $r(x)$ can be rewritten as

$$r(x) = \mathbb{E}_\xi \left[\min_{\bar{y} \in \Delta^{n_2}} (2(1 - e^\top x) x^\top \tilde{B}^\top \bar{y} + (1 - e^\top x)^2 \bar{y}^\top \tilde{C} \bar{y}) \right],$$

and defining $\tilde{D}_x := (1 - e^\top x)^2 \tilde{C} + (1 - e^\top x) [e x^\top \tilde{B} + \tilde{B} x e^\top]$ with entries $\tilde{D}_{ij}(x)$ in row i , column j of \tilde{D}_x , we have the following estimate via Jensen's inequality:

$$r(x) \leq \mathbb{E}_\xi \left[\min_i \tilde{D}_{ii}(x) \right] \\ = \mathbb{E}_\xi \left[\min_i (1 - e^\top x)^2 \tilde{C}_{ii} + 2(1 - e^\top x) (\tilde{B} x)_i \right] \\ \leq \min_i \left((1 - e^\top x)^2 \mathbb{E}_x [\tilde{C}_{ii}] + 2(1 - e^\top x) \mathbb{E}_\xi [(\tilde{B} x)_i] \right),$$

and thus

$$z_{\text{stoch}}^* \leq \min_i \min \{x^\top Ax + (1 - e^\top x)^2 \mathbb{E}_\xi [\tilde{C}_{ii}] + 2(1 - e^\top x) \mathbb{E}_\xi [(\tilde{B} x)_i] : x \in T^{n_1}\},$$

the minimum of n_2 (possibly non-convex) QPs over T^{n_1} which can be reduced to n_2 DStQPs or to an equivalent copositive optimization problem.

Likewise we can use the edge-minimization upper bound using the argument of tightness with high probability (Chen & Peng, 2015; Chen, Peng, & Zhang, 2013):

$$z_{\text{det}}^* \leq \min_{i,j} \left[\min \{q(z) : z \in \text{conv} \{e_i, e_j\}\} \right],$$

which by a similar method can be used for an upper bound on z_{stoch}^* with a slightly more involved argument to fractional QPs with linear denominators over Δ^{n_1} . The fractions appear since the minimal values along a convex edge¹ $\text{conv} \{e_i, e_j\}$ of a DStQP are given by the fractions

$$\tilde{\tau}_{ij}(x) := \frac{\tilde{D}_{ij}^2(x) - \tilde{D}_{ii}(x)\tilde{D}_{jj}(x)}{2\tilde{D}_{ij}(x) - \tilde{D}_{ii}(x) - \tilde{D}_{jj}(x)}.$$

We reserve dealing with this type of problem in a way similar to Amaral, Bomze, & Júdice (2014) for a future research project but note beforehand that the common factor $(1 - e^\top x)$ occurring in the estimates of $r(x)$ does not depend on i, j which may simplify analysis considerably.

Turning to lower bounds, we note three variants: the trivial one

$$z_{\text{det}}^* \geq q_{\min} := \min_{i,j} Q_{ij}, \quad (6)$$

Nesterov's bound

$$z_{\text{det}}^* \geq \min_{i,j} (Q_{ij} + \frac{1}{2}(Q_{ii} + Q_{jj})) - \max_k Q_{kk}, \quad (7)$$

and a stronger one than both above from Bomze, Locatelli, & Tardella (2008)

$$z_{\text{det}}^* \geq q_{\min} + \left[\sum_i (Q_{ii} - q_{\min})^{-1} \right]^{-1}. \quad (8)$$

Despite being of closed form, it is not obvious how to proceed using these for a tractable procedure solving (2(St)³QP).

¹ i.e., an edge along which q is convex.

4. Bounds on (2(St)³QP) via scenario tree discretization

In stochastic optimization is customary to approximate the stochastic optimization problem by discretizing the underlying random process $\xi \in \Xi$ and subsequently solve the so called scenario problem, i.e. the stochastic optimization problem where the true distribution is replaced by its discretization. Several methods are adopted in the literature for discretizing distributions and generating scenarios. Among the most common we list: Conditional sampling, sampling from specified marginals and correlations, moment matching, path-based methods and optimal discretization. For an overview of scenario-generation methods, see Pflug & Pichler (2011).

In this section we will discuss upper and lower bounds on the scenario problem based on stochastic and quadratic optimization theory. But first let us derive the actual form of the scenario problem in case of (2(St)³QP).

Suppose now that the distribution of ξ has finite support. This means that ξ has a finite number of possible realizations, called scenarios $\xi_s := (\tilde{B}_s, \tilde{C}_s)$ with respective positive, known probabilities p_s , $s \in [1:S]$, so $\Xi := \{\xi_1, \dots, \xi_S\}$.

For a given x , the recourse function $r(x)$ is equal to the optimal value of the following problem

$$\begin{aligned} \min_{y_1, \dots, y_S} \quad & \sum_{s=1}^S p_s (2x^\top \tilde{B}_s^\top y_s + y_s^\top \tilde{C}_s y_s) \\ \text{s.t.} \quad & e^\top x + e^\top y_s = 1, \quad s \in [1:S], \\ & y_s \in \mathbb{R}_+^{n_2}, \quad s \in [1:S]. \end{aligned}$$

The whole two-stage stochastic problem can be approximated by the following large-scale deterministic equivalent quadratic problem

$$\begin{aligned} z_{\text{stoch}}^* := \min_{x, y_1, \dots, y_S} \quad & f(z) := x^\top A x + \sum_{s=1}^S p_s (2x^\top \tilde{B}_s^\top y_s + y_s^\top \tilde{C}_s y_s) \\ \text{s.t.} \quad & e^\top x + e^\top y_s = 1, \quad s \in [1:S], \\ & y_s \in \mathbb{R}_+^{n_2}, \quad s \in [1:S], \\ & x \in \mathbb{R}_+^{n_1}, \end{aligned} \quad (9)$$

where $z^\top = [x^\top, y_1^\top, \dots, y_S^\top]$ collects all decision variables.

Since (9) describes a class of non-convex QPs which contains the StQP as a special case, it is NP-hard. We will now proceed in presenting some bounds which can be computed with reasonable effort.

4.1. Upper bounds by Frank–Wolfe variants

Frank–Wolfe variants are fast and efficient first-order methods for smooth, linearly constrained optimization problems of the form

$$\min \{f(z) : z \in P\},$$

with P a polytope in \mathbb{R}^d of which the set of vertices $V(P)$ are known or can be generated quickly on the fly if they are too many. A recent survey is offered in Bomze, Rinaldi, & Zeffiro (2021). These methods enjoy increasing popularity in particular for large-scale problems as occurring in Data Science, also because of their good properties like global convergence and support identification in finite time (Bomze, Rinaldi, & Rota Bulò, 2019; Bomze, Rinaldi, & Zeffiro, 2020; Lacoste-Julien, 2016), even for non-convex problems. They are iterative primal methods which employ, at an iterate z_k , the knowledge of the gradient $g_k := \nabla f(z_k)$ to define directions \bar{d}_k along which then a line search is performed. Frequently employed line search strategies include predetermined step sizes (e.g., constant or $\frac{\gamma}{k}$ with a parameter $\gamma > 0$) as well as approximately optimal ones (à la Armijo) or even exact line searches over the interval

$[0, \bar{\alpha}(z_k, \bar{d}_k)]$ with

$$\bar{\alpha}(z, \bar{d}) = \inf \{t > 0 : z + t \bar{d} \notin P\}$$

denoting the maximal feasible step size at $z \in P$ in direction \bar{d} .

Fortunately, for our problem (9) with $d = n_1 + n_2 S$, both the structure of the feasible polytope P as well as the objective f enable a quick implementation of the algorithm even for the exact line search. Indeed, if \bar{d} is a feasible direction at z , i.e. $\bar{d} = t(x - z)$ for some $t > 0$ and $x \in P$, we have

$$\bar{\alpha}(z, \bar{d}) = \min_j \left\{ -\frac{z_j}{d_j} : d_j < 0 < z_j \right\}.$$

Furthermore, since f is quadratic, a closed-form solution for the exact step size α_k is given by means of the quadratic form of the Hessian $H = D^2 f(z)$ (the only second-order aspect of our approach; as f is quadratic, H is constant across all iterations):

$$\alpha_k^e = \begin{cases} \min \left\{ \bar{\alpha}(z_k, \bar{d}_k), -\frac{g_k^\top \bar{d}_k}{\bar{d}_k^\top H \bar{d}_k} \right\}, & \text{if } \bar{d}_k^\top H \bar{d}_k > 0 \geq g_k^\top \bar{d}_k, \\ \bar{\alpha}(z_k, \bar{d}_k), & \text{else.} \end{cases} \quad (10)$$

Convergence and active set identification with this choice of step size was recently discussed in Bomze et al. (2019, 2020). For better numerical stability, we resorted to a damped variant

$$\alpha_k = \begin{cases} \min \left\{ \bar{\alpha}(z_k, \bar{d}_k), -\beta \frac{g_k^\top \bar{d}_k}{\bar{d}_k^\top H \bar{d}_k} \right\}, & \text{if } \bar{d}_k^\top H \bar{d}_k > 0 \geq g_k^\top \bar{d}_k, \\ \bar{\alpha}(z_k, \bar{d}_k), & \text{else,} \end{cases} \quad (11)$$

where $0 < \beta < 1$, with β a tuning parameter. By thorough numerical experimentation we found that $\beta = \frac{1}{2}$ is most efficient.

We still need to define the search direction \bar{d}_k . To this end, we employ the so-called Pairwise Frank–Wolfe (PFW) approach, which combines the classical Frank–Wolfe direction with the Away-step Frank–Wolfe direction. More precisely, an update on an iterate z_k is performed by selecting a vertex $v_k \in P$ towards which we wish to move, and a vertex $w_k \in S_k \subseteq P$ from which we wish to move away, hence $z_{k+1} = z_k + \alpha_k(v_k - w_k)$. The vertex v_k is chosen as the vertex that minimizes a linear model of f based on $\nabla f(z_k)$ over P , and w_k is chosen as the maximizer of said model over S_k . The latter set is critical in this calculation. Since $z_k \in P$, it is a convex combination of vertices of P , and S_k is the set of currently used vertices. The idea behind moving away from vertices in S_k is to eliminate so-called bad atoms in the description of the current iterate in every iteration. Of course, the set S_k depends on z_k and it is imperative for algorithmic performance to update S_k in an efficient manner. We will discuss our solution to this obstacle in more detail in the discussion below.

First, we will show that determining v_k and w_k in every iteration amounts to extracting the extremes of a list of at most n_1 numbers plus S such extractions from lists of at most n_2 numbers. The argument requires the following characterization of vertices of P (recall that an $x \in P$ is a vertex or extremal point of P if $0 < \lambda < 1$ and $x = \lambda x_1 + (1 - \lambda)x_2$ implies $x = x_1 = x_2$).

We need the notation e_i for the i th column of an identity matrix I_d of suitable order, and abbreviate by $\mathcal{J} := [1:n_2]^S$.

Theorem 1. *The vertices of the set*

$$P := \{(x, y_1, \dots, y_S) \in \mathbb{R}_+^{n_1 + 5n_2} : e^\top x + e^\top y_i = 1, i \in [1:S]\}$$

are given by

$$V(P) = \{e_i : i \in [1:n_1]\} \cup \left\{ \sum_{s=1}^S e_{j_s + n_1 + (s-1)n_2} : \mathbf{j} \in \mathcal{J} \right\}. \quad (12)$$

Proof. The proof proceeds in two steps. First we show that the alleged points are indeed vertices of P , second we show that any

other point in P is a convex combination of our alleged points, so that there cannot be any other extremal point of P outside $V(P)$. So consider $x_1, x_2 \in P$, $0 < \lambda < 1$ so that $e_i = \lambda x_1 + (1 - \lambda)x_2$ for any $i \in [1:n_1]$. Then for any $j \neq i$ we get from $\lambda(x_1)_j + (1 - \lambda)(x_2)_j = 0$ and $x_1, x_2 \geq 0$ that $(x_1)_j, (x_2)_j = 0$. From $\lambda(x_1)_i + (1 - \lambda)(x_2)_i = 1$ we see that, perhaps after a relabeling, $(x_1)_i \geq 1 \geq (x_2)_i$ and since $x_1 \in P$, so that $(x_1)_i \leq 1$, we have $(x_1)_i = 1$, which necessitates $(x_2)_i = 1$. The analogous argument can be made for $\sum_{s=1}^S e_{j_s+n_1+(s-1)n_2}$ and any $\mathbf{j} \in \mathcal{J}$ by applying the same analysis to each of the S segments of length n_2 individually. Hence the points in $V(P)$ are vertices of P , and we will now show that these are the only extremal points of P . So let $\bar{x} = (x^T, y^T)^T \in P$ with $0 < e^T \bar{x} < 1$. Then

$$\bar{x} = \begin{pmatrix} x \\ y_1 \\ \vdots \\ y_S \end{pmatrix} = \begin{pmatrix} x \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ y_1 \\ \vdots \\ y_S \end{pmatrix} = e^T x \begin{pmatrix} \frac{x}{e^T x} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} + (1 - e^T x) \begin{pmatrix} \mathbf{0} \\ \hat{y}_1 \\ \vdots \\ \hat{y}_S \end{pmatrix},$$

with $\hat{y}_s = \frac{1}{1 - e^T x} y_s$, all $s \in S$. Evidently the two vectors on the right-hand side are members of P , so that \bar{x} is a convex combination of these two vectors. Further, since $e^T \frac{x}{e^T x} = 1$, we see that $\frac{x}{e^T x}$ is a convex combination of e_i , $i \in [1:n_1]$. Thus our claim will follow if we can show that the second vector is a convex combination of members of $\left\{ \sum_{s=1}^S e_{j_s+n_1+(s-1)n_2} : \mathbf{j} \in \mathcal{J} \right\}$. To this end, we shift our attention to the vector $\hat{y} := (\hat{y}_1^T, \dots, \hat{y}_S^T)^T \in \mathbb{R}^{Sn_2}$. For every $\mathbf{j} \in \mathcal{J}$ we define $e_{\mathbf{j}} := (e_{j_1}^T, \dots, e_{j_S}^T)^T \in \mathbb{R}^{Sn_2}$ and then need to show that the following system has a solution λ :

$$\hat{y} = \sum_{\mathbf{j} \in \mathcal{J}} \lambda_{\mathbf{j}} e_{\mathbf{j}}, \quad \sum_{\mathbf{j} \in \mathcal{J}} \lambda_{\mathbf{j}} = 1, \quad \lambda_{\mathbf{j}} \geq 0 \quad \forall \mathbf{j} \in \mathcal{J}.$$

By Farkas' lemma we can equivalently show that

$$e_{\mathbf{j}}^T \hat{z} + z_0 \geq 0, \quad \forall \mathbf{j} \in \mathcal{J}, \quad \hat{y}^T \hat{z} + z_0 < 0$$

has no solution (z_0, \hat{z}) . Using the subdivision $\hat{z} := (\hat{z}_1^T, \dots, \hat{z}_S^T)^T$ we can see that the system implies $\sum_{s=1}^S e_{j_s}^T \hat{z}_s > \sum_{s=1}^S \hat{y}_s^T \hat{z}_s$ for every $\mathbf{j} \in \mathcal{J}$. Note, that for every $s \in [1:S]$ we have that $e_{j_s}^T \hat{z}_s \leq \hat{y}_s^T \hat{z}_s$ for some $j_s \in [1:n_2]$, because the left-hand side equals a single component of \hat{z}_s while the right-hand side is a convex combination of all the components of \hat{z}_s due to $e^T \hat{y}_s = 1$ and $\hat{y}_s \geq 0$. Thus, for any \hat{z} we can always find $\mathbf{j} \in \mathcal{J}$ such that $\sum_{s=1}^S e_{j_s}^T \hat{z}_s \leq \sum_{s=1}^S \hat{y}_s^T \hat{z}_s$, which proves the inconsistency of the alternative system of equations. This concludes the proof, leaving the trivial cases $e^T x \in \{0, 1\}$ to the readers. \square

Let us note here a particular distinction contrasting our approach with usual Away-Step strategies in Frank-Wolfe variants. First note that the number of vertices of P is exponential in $|S|$, so that in general enumeration of all of them to identify bad atoms would be prohibitive. We circumvent this issue by shifting our attention from vertices active in the k th iterate z_k to the nonzero coordinates of that iterate. More specifically we will define for any iterate z_k ,

$$S_k := \{w \in P : w \text{ is a vertex of } P, w_i = 0 \text{ if and only if } z_k^i = 0\}.$$

Hence, once a coordinate of z_k is zero, all associated vertices are dropped from the description of S_k . This reduces the computational burden, as merely the index of the dropped coordinates have to be stored, and optimizing a linear function over S_k becomes a matter of identifying the respective extreme objective function coefficients of the nonzero coordinates. This is due to the particular structure of our optimization problem. We detail the procedure in the following theorem where, for simplicity of exposition, we shortly suppress the iteration counter in notation:

Theorem 2. At an iterate z with gradient $g = \nabla f(z)$, define

$$\text{supp } z := \{j \in [1:d] : z_j > 0\}$$

as well as

$$\begin{aligned} \min[g|x] &:= \min\{g_i : i \in [1:n_1]\} \quad \text{and} \\ \min[g|y] &:= \sum_{s=1}^S \min\{g_j : j \in [n_1+(s-1)n_2+1:n_1+sn_2]\} \end{aligned}$$

and similarly for the max operator but restricted to $\text{supp } z$:

$$\begin{aligned} \max[g|x] &:= \max\{g_i : i \in [1:n_1] \cap \text{supp } z\} \quad \text{and} \\ \max[g|y] &:= \sum_{s=1}^S \max\{g_j : j \in [n_1+(s-1)n_2+1:n_1+sn_2] \cap \text{supp } z\}. \end{aligned}$$

Let e_j denote the j th column of I_d . Then the PFW search direction \bar{d} is given by the vertex difference $\bar{d} = v - w$ where $\{v, w\} \subset V(P)$ satisfy

$$v = \left\{ \begin{array}{l} e_i, \quad \text{if } i \in [1:n_1] \text{ realizes } g_i = \min[g|x] \leq \min[g|y], \\ \sum_{s=1}^S e_{j_s}, \quad \text{if } j_s \in [n_1+(s-1)n_2+1:n_1+sn_2], \text{ all } s \in [1:S], \\ \text{realize } \sum_{s=1}^S g_{j_s} = \min[g|y] < \min[g|x], \end{array} \right\}, \quad (13)$$

and

$$w = \left\{ \begin{array}{l} e_i, \quad \text{if } i \in [1:n_1] \cap \text{supp } z \text{ realizes } g_i = \max[g|x] \geq \max[g|y], \\ \sum_{s=1}^S e_{j_s}, \quad \text{if } j_s \in [n_1+(s-1)n_2+1:n_1+sn_2] \cap \text{supp } z, \text{ all } s \in [1:S], \\ \text{realize } \sum_{s=1}^S g_{j_s} = \max[g|y] > \max[g|x]. \end{array} \right\}. \quad (14)$$

The conditions on $(j_s)_{s \in [1:S]}$ in (13) are met if and only if j_s minimize individually the numbers $\{g_j : j \in [(s-1)n_2+1:sn_2]\}$ (and likewise those in (14) for $\{g_j : j \in [(s-1)n_2+1:sn_2] \cap \text{supp } z\}$ and maximization), across all $s \in [1:S]$.

Proof. At an iterate z , PFW selects the search direction among all vertex differences $v - w$ such that $g^T v = \min\{g^T v' : v' \in V(P)\}$ and $g^T w = \max\{g^T w' : w' \in V_z(P)\}$, where $V_z(P) \neq \emptyset$ denotes those vertices with have positive weight in at least one barycentric coordinate representation of z with respect to $V(P)$. Taking convex combinations, it is always possible to have all vertices in $V_z(P)$ with positive weights in a barycentric representation (call it the least sparse one). The resultant \bar{d} is always a feasible first-order descent direction, $g^T \bar{d} \leq 0$. In our case, we get from Theorem 1 that the vertices of P are given by (12):

$$V(P) = \{e_i : i \in [1:n_1]\} \cup \left\{ \sum_{s=1}^S e_{j_s} : j_s \in [n_1+(s-1)n_2+1:n_1+sn_2], \text{ all } s \in [1:S] \right\},$$

and (considering a least sparse representation of z)

$$V_z(P) = \{e_i \in V(P) : i \in \text{supp } z\} \cup \left\{ \sum_{s=1}^S e_{j_s} \in V(P) : j_s \in \text{supp } z, \text{ all } s \right\}.$$

If $v = e_i$ with $i \in [1:n_1]$, we obviously get $g^T v = g_i$. Further, since any selection $j_s \in [n_1+(s-1)n_2+1:n_1+sn_2]$ across $s \in [1:S]$ gives a vertex v with

$$g^T v = \sum_{s=1}^S g_{j_s}, \quad (15)$$

and since extremizing the summands g_j across all $j \in [n_1+(s-1)n_2+1:n_1+sn_2]$ (with $j \in \text{supp } z$ in case of maximization) and summing over all $s \in [1:S]$, extremizes the sum in (15) across these vertices, the claim follows. \square

To summarize the PFW algorithm, we need also a stopping criterion for which we use the Frank/Wolfe gap Lacoste-Julien (2016) at an iterate $z \in P$ with gradient $g = \nabla f(z)$:

$$G(z) := -g^T (v - z) \quad \text{where } v \text{ is defined in (13)}. \quad (16)$$

Algorithm 1. Pairwise Frank/Wolfe method for (9).

Input: Problem data, bound $\varepsilon > 0$ for the Frank/Wolfe gap, initial guess $z_0 \in P$;

Output: upper bound $f(z^*)$

$k = 0$;

While $G(z_k) > \varepsilon$

 Generate search direction \bar{d}_k as in Theorem-2;

 calculate step size α_k as in (17);

$z_{k+1} := z_k + \alpha_k \bar{d}_k$;

 increment k ;

z* := z_k , upper bound is $f(z^*)$.

Multistarts may be necessary to improve upon the local quality of the bound.

4.2. Scalable copositive bounds

In this section we will derive lower and upper bounds to (9) based on a well known reformulation strategy described in the following theorem originally from Burer & Anstreicher (2013). However, the special structure of our problem (9) allows for an essential reduction in size which enables to solve much larger problems than it would have been possible with the standard reformulation. For better understanding, we first shortly refer the general case also including quadratic constraints – which are not present in (9).

Theorem 3. Let $\mathcal{K} \subseteq \mathbb{R}^n$ be a cone, and let

$$\mathcal{F} := \{x \in \mathcal{K} : x^T A_i x + 2a_i^T x \leq b_i, \quad i \in [1:m]\}$$

be a feasible set of a quadratically constrained quadratic optimization problem and denote

$$\mathcal{G}(\mathcal{F}) := \text{clconv}\{x, xx^T : x \in \mathcal{F}\},$$

where $\text{clconv}(\mathcal{C})$ stands for the closure of the convex hull of a set \mathcal{C} . Then for any $Q \in \mathbb{S}^n$, $q \in \mathbb{R}^n$ and $\omega \in \mathbb{R}$ we have

$$\inf_{x \in \mathcal{F}} (x^T Q x + 2q^T x + \omega) = \inf_{(x, X) \in \mathcal{G}(\mathcal{F})} (Q \bullet X + 2q^T x + \omega).$$

The theorem rests on a few basic ideas. First note $x^T Q x = Q \bullet xx^T$ by elementary properties of the trace function. The first step to obtain the right-hand side problem is lifting, i.e., to replace term $xx^T \subseteq \mathbb{S}^n$ by a matrix variable $X \subseteq \mathbb{S}^n$. In this way we obtain a linearization of the objective where X_{ij} replaces the quadratic term $x_i x_j$. The second, and much more challenging ingredient is $\mathcal{G}(\mathcal{F})$. By construction it is convex and all its extreme points are of the form (x, xx^T) where x is a member of the original feasible set \mathcal{F} . The linearized problem attains its optimal value at an extreme point of its feasible set, which by construction corresponds to a feasible solution of the original problem. So, on one hand the right-hand side problem gives a lower bound since any $x \in \mathcal{F}$ gives a solution $(x, xx^T) \in \mathcal{G}(\mathcal{F})$. On the other hand it has an optimal solution of the form (x, xx^T) with $x \in \mathcal{F}$, thus yielding an upper bound on the original problem. Hence the problems are equivalent.

However, the characterization of $\mathcal{G}(\mathcal{F})$ is challenging and can only be obtained for some specific choices of \mathcal{F} (see for example Bomze et al. (2000); Burer (2012)). The most famous example from Burer (2009) is the completely positive reformulation of a quite versatile class of QCQPs:

Theorem 4. Assume that $\mathcal{L} := \{x \in \mathbb{R}_+^n : Ax = b\}$ is bounded and that $x^T Q_i x + 2q_i^T x + \omega_i \geq 0$ for all $x \in \mathcal{L}$ and $i \in [1:l]$. Then

$$\min_{x \in \mathbb{R}_+^n} \{x^T Q x + 2q^T x : Ax = b, \quad x^T Q_i x + q_i^T x + \omega_i = 0, \quad i \in [1:l]\}$$

is equivalent to

$$\begin{aligned} \min_{x, X} \quad & \text{trace}(QX) + q^T x \\ \text{s.t.} \quad & Ax = b, \\ & \text{Diag}(AXA^T) = b \circ b, \\ & \text{trace}(Q_i X) + 2q_i^T x + \omega_i = 0, \quad i \in [1:l] \\ & \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \in \mathcal{CPP}_{n+1}, \end{aligned}$$

where $\mathcal{CPP}_k := \text{conv}\{xx^T : x \in \mathbb{R}_+^k\}$ for $k \in \mathbb{N}$.

The cone of completely positive matrices \mathcal{CPP}_n is intensively researched (see Berman & Shaked-Monderer, 2003). It is well known, that certifying membership in \mathcal{CPP}_n is NP-hard so that the above convex reformulation is intractable. However, there are powerful outer approximations of this cone the simplest of which is the doubly nonnegative cone $\mathcal{DN}_n := \mathbb{S}_+^n \cap \mathcal{N}_n$, i.e. the intersection of the cone of positive-semidefinite matrices and the nonnegative orthant. For applications in optimization see, e.g. Bomze (2012).

In the spirit of Theorem 3 we will provide a convex reformulation of (9) described in the theorem below.

Theorem 5. Consider the problem

$$\begin{aligned} (P) \quad & \min_{x, y_1, \dots, y_S} x^T A x + 2a^T x + \sum_{s=1}^S (2x^T B_s y_s + y_s^T C_s x + 2c_s^T y_s) \\ \text{s.t.} \quad & e^T x + e^T y_s = 1, \quad \text{all } s \in [1:S], \\ & x \in \mathbb{R}_+^n, \quad y_s \in \mathbb{R}_+^{n_s}, \quad \text{all } s \in [1:S]. \end{aligned} \quad (17)$$

The following conic optimization problem gives a lower bound. Further, if for all $s \in [1:S]$ we have $c_s = 0$, $B_s = b_s e^T$ and $b_s \in \mathbb{R}^{n_1}$, then the bound is actually tight:

$$\begin{aligned} (R) \quad & \min_{X, Y_s, Z_s, Y_s} A \bullet X + 2a^T x + \sum_{s=1}^S (2B_s \bullet Z_s + C_s \bullet Y_s + 2c_s^T y_s) \\ \text{s.t.} \quad & e^T x + e^T y_s = 1, \quad s \in [1:S], \\ & E \bullet X + 2 E \bullet Z_s + E \bullet Y_s = 1, \quad s \in [1:S], \\ & \begin{pmatrix} 1 & x^T & y_s^T \\ x & X & Z_s^T \\ y_s & Z_s & Y_s \end{pmatrix} \in \mathcal{CPP}_{n_1+n_2+1}, \quad s \in [1:S]. \end{aligned} \quad (18)$$

Proof. Take a feasible solution (x, y_1, \dots, y_S) to (17) and define $X = xx^T$, $Z_s = y_s x^T$, $Y_s = y_s y_s^T$, $s \in [1:S]$. Using again the identity $A \bullet xx^T = x^T A x$, we have identical objective function values. Clearly, the first S linear constraints are fulfilled and from squaring both sides we get $1^2 = (e^T x + e^T y_s)^2 = (e^T x)^2 + 2(e^T x e^T y_s) + (e^T y_s)^2 = E \bullet X + 2 E \bullet Z_s + E \bullet Y_s$, $s \in [1:S]$. Further, for all $s \in [1:S]$ we have,

$$\mathcal{CPP}_{n_1+n_2+1} \ni \begin{pmatrix} 1 \\ x \\ y_s \end{pmatrix} \begin{pmatrix} 1 \\ x \\ y_s \end{pmatrix}^T = \begin{pmatrix} 1 & x^T & y_s^T \\ x & X & Z_s^T \\ y_s & Z_s & Y_s \end{pmatrix}$$

and hence, denoting with $\text{val}(\cdot)$ the objective function value of a problem, we have $\text{val}(P) \geq \text{val}(R)$.

For the converse, under the additional assumptions we can always transform the objective function

$$\begin{aligned} & x^T A x + 2a^T x + \sum_{s=1}^S [2x^T b_s e^T y_s + y_s^T C_s y_s] \\ &= x^T A x + 2a^T x + \sum_{s=1}^S [2x^T b_s (1 - e^T x) + y_s^T C_s y_s] \\ &= x^T \left(A - \sum_{s=1}^S (b_s e^T + e b_s^T) \right) x + 2 \left(a + \sum_{s=1}^S b_s \right)^T x + \sum_{s=1}^S y_s^T C_s y_s, \end{aligned} \quad (19)$$

so we only need to consider the case where $B_s = O$ for all $s \in [1:S]$. Define

$$\phi_s(x, X) := \min_{y_s, Y_s} \left\{ C \bullet Y_s : \begin{array}{l} e^T x + e^T y_s = 1, \\ E \bullet X + 2E \bullet Z_s + E \bullet Y_s = 1, \end{array} \begin{pmatrix} 1 & x^T & y_s^T \\ x & X & Z_s^T \\ y_s & Z_s & Y_s \end{pmatrix} \in \mathcal{CPP}(\mathbb{R}_+^{n_1+n_2}) \right\},$$

and

$$\mathcal{X} := \left\{ (x, X) : \exists (y, Y) : \begin{array}{l} e^T x + e^T y = 1, \\ E \bullet X + 2E \bullet Z + E \bullet Y = 1, \end{array} \begin{pmatrix} x^T & y^T \\ x & X & Z^T \\ y & Z & Y \end{pmatrix} \in \mathcal{CPP}(\mathbb{R}_+^{n_1+n_2}) \right\}.$$

Note that by Burer's reformulation (see [Burer, 2009](#)) the set \mathcal{X} is the projection of $\mathcal{G}(\mathcal{F})$ onto the (x, X) coordinates, where

$$\mathcal{F} = \left\{ (x^T, y^T)^T \in \mathbb{R}^{n_1+n_2} : e^T x + e^T y = 1 \right\}.$$

Since the extreme points of a projection of a convex set correspond to extreme points of the projected set, it follows that the extreme points of \mathcal{X} are of the form (x, xx^T) . The feasible set in the description of $\phi(x, X)$ (we suppress the index for the moment to avoid double indices in the following paragraphs) is given by

$$\mathcal{Y}(x, X) := \left\{ Y : \exists (Z, y) : \begin{array}{l} e^T x + e^T y = 1, \\ E \bullet X + 2E \bullet Z + E \bullet Y = 1, \end{array} \begin{pmatrix} x^T & y^T \\ x & X & Z^T \\ y & Z & Y \end{pmatrix} \in \mathcal{CPP}(\mathbb{R}_+^{n_1+n_2}) \right\},$$

which again is a projection of $\mathcal{G}(\mathcal{F})$, but this time on the (Y) coordinates. We will first show that

$$\mathcal{Y}(x, X) \subseteq \left\{ Y : E \bullet Y = 1 - 2e^T x + E \bullet X, Y \in \mathcal{CPP}(\mathbb{R}_+^{n_2}) \right\} =: \tilde{\mathcal{Y}}(x, X).$$

Assume $Y \in \mathcal{Y}(x, X)$, then $Y \in \mathcal{CPP}(\mathbb{R}_+^{n_2})$. Further, by Burer's reformulation we have the following representation

$$\begin{pmatrix} 1 & x^T & y^T \\ x & X & Z^T \\ y & Z & Y \end{pmatrix} = \sum_{i=1}^k \lambda_i \begin{pmatrix} 1 \\ x_i \\ y_i \end{pmatrix} \begin{pmatrix} 1 \\ x_i \\ y_i \end{pmatrix}^T \quad (20)$$

where $(x_i, y_i) \in \mathbb{R}_+^{n_1} \times \mathbb{R}_+^{n_2}$, $e^T x_i + e^T y_i = 1$, and $\lambda_i \geq 0$, for all $i \in [1:k]$ as well as $\sum_{i=1}^k \lambda_i = 1$. We also have

$$\begin{aligned} E \bullet X + 2E \bullet Z + E \bullet Y = 1 &\iff \\ E \bullet \sum_{i=1}^k \lambda_i x_i x_i^T + 2E \bullet \sum_{i=1}^k \lambda_i y_i y_i^T + E \bullet \sum_{i=1}^k \lambda_i y_i y_i^T = 1 &\iff \\ \sum_{i=1}^k \lambda_i (e^T x_i)^2 + 2 \sum_{i=1}^k \lambda_i (e^T x_i)(e^T y_i) + \sum_{i=1}^k \lambda_i (e^T y_i)^2 = 1 &\iff \\ \sum_{i=1}^k \lambda_i (e^T x_i)^2 + 2 \sum_{i=1}^k \lambda_i (e^T x_i)(1 - e^T x_i) + \sum_{i=1}^k \lambda_i (e^T y_i)^2 = 1 &\iff \\ \sum_{i=1}^k \lambda_i (e^T x_i)^2 + 2 \left(\sum_{i=1}^k \lambda_i (e^T x_i) - \sum_{i=1}^k \lambda_i (e^T x_i)^2 \right) + \sum_{i=1}^k \lambda_i (e^T y_i)^2 = 1 &\iff \\ 2e^T x - E \bullet X + E \bullet Y = 1, & \end{aligned}$$

and thus we have $Y \in \tilde{\mathcal{Y}}(x, X)$. Also, we get $1 - 2e^T x + E \bullet X = \sum_{i=1}^k \lambda_i (1 - e^T x_i)^2 \geq 0$.

It is well known (see [Bomze et al. \(2000\)](#)) that $\tilde{\mathcal{Y}}(x, X)$ is a compact, conic intersection that has extreme points that are rank one matrices. We thus have

$$\min_{Y \in \tilde{\mathcal{Y}}(x, X)} C \bullet Y = C \bullet \tilde{y} \tilde{y}^T (1 - 2e^T x + E \bullet X) \text{ where } \tilde{y} \tilde{y}^T \in \arg \min_{Y \in \tilde{\mathcal{Y}}(x, X)} C \bullet Y,$$

so that $e^T \tilde{y} = 1$. Now consider the following problem

$$(\bar{R}) : \min_{(x, X) \in \mathcal{X}} A \bullet X + 2a^T x + \sum_{s=1}^S \min_{Y_s \in \tilde{\mathcal{Y}}(x, X)} C_s \bullet Y_s. \quad (21)$$

We have

$$\begin{aligned} \text{val}(R) &= \min_{(x, X) \in \mathcal{X}} A \bullet X + 2a^T x + \sum_{s=1}^S \phi_s(x, X) \\ &\geq \text{val}(\bar{R}) = \min_{(x, X) \in \mathcal{X}} A \bullet X + 2a^T x + \sum_{s=1}^S C_s \bullet \tilde{y}_s \tilde{y}_s^T (1 - 2e^T x + E \bullet X) \end{aligned}$$

$$= A \bullet \tilde{x} \tilde{x}^T + 2a^T \tilde{x} + \sum_{s=1}^S C_s \bullet \tilde{y}_s \tilde{y}_s^T (1 - e^T \tilde{x})^2$$

since the objective is linear and the extreme points of \mathcal{X} are rank one matrices. We have $\tilde{x} \in \mathbb{R}_+^{n_1}$, $(1 - e^T \tilde{x}) \tilde{y}_s \in \mathbb{R}_+^{n_2}$, $s \in [1:S]$ and $e^T \tilde{x} + (1 - e^T \tilde{x}) e^T \tilde{y}_s = e^T \tilde{x} + 1 - e^T \tilde{x} = 1$ so that we have a feasible solution for (P) that gives the same objective function value. In conclusion we have

$$\text{val}(P) \geq \text{val}(R) \geq \text{val}(\bar{R}) \geq \text{val}(P)$$

implying that $\text{val}(P) = \text{val}(R)$.

Note that in case we implemented the changes in the objective function mentioned in (19), the objective function of the reformulation becomes

$$\left(A - \sum_{s=1}^S (b_s e^T + e b_s^T) \right) \bullet X + 2 \left(a + \sum_{s=1}^S b_s \right) x + \sum_{s=1}^S C \bullet Y_s.$$

However, we can undo these changes so that we truly arrive at the representation that is stated in the theorem. We can rearrange the transformed objective function as follows:

$$\begin{aligned} &\left(A - \sum_{s=1}^S (b_s e^T + e b_s^T) \right) \bullet X + 2 \left(a + \sum_{s=1}^S b_s \right) x + \sum_{s=1}^S C \bullet Y_s \\ &= A \bullet X + 2a^T x - \sum_{s=1}^S [(b_s e^T + e b_s^T) \bullet X + 2b_s^T x + C_s \bullet Y_s], \end{aligned}$$

and consider the S terms in the sum individually. We have $(y_s, Y_s) \in \mathcal{Y}_s(x, X)$ for all $s \in [1:S]$, so that we can again consider the decomposition in (20). As before we suppress the index s of y_s, Z_s, Y_s, b_s and C_s in order to avoid double indices. The components of the decomposition again fulfil $e^T x_i + e^T y_i = 1$ for all $i \in [1:k]$, which allows us to rearrange the terms as follows:

$$\begin{aligned} &-(b e^T + e b^T) \bullet X + 2 b^T x_i + C \bullet Y \\ &= \sum_{i=1}^k \lambda_i (-(b e^T + e b^T) \bullet x_i x_i^T + 2 b^T x_i + C \bullet y_i y_i^T) \\ &= \sum_{i=1}^k \lambda_i (-2 x_i^T b (1 - e^T y_i) + 2 b^T x_i + C \bullet y_i y_i^T) \\ &= \sum_{i=1}^k \lambda_i (2 b e^T \bullet y_i x_i^T + C \bullet y_i y_i^T) \\ &= 2 B \bullet Z + C \bullet Y, \end{aligned}$$

so that we recover the form stated in problem (R). \square

Regardless whether the exactness condition is met or not, the optimal value of (R) is always a lower bound of (P). This remains true if we replace the intractable conic problems by tractable ones. Moreover, taking a part of its solution $z^T = [x^T, y_1^T, \dots, y_S^T]$, we see that z is (P)-feasible, so we also get an upper bound by the (P)-feasible value $f(z)$ which is generally good (the only problematic term is indeed the bilinear one). Note that (17) can be reformulated using [Theorem 4](#) where the size of the completely positive matrix block would be $n = n_1 + S n_2 + 1$, so that the number of variables would increase quadratically with S . On the other hand, in [Theorem 5](#) we have S completely positive matrix blocks of order $n_1 + n_2 + 1$ so that the number of variables is linear in S . It is obvious that this yields substantial advantages in computation time and memory consumption over the generic CP-representation formulated in [Theorem 4](#); and numerical experience confirms this expectation.

4.3. Lower bounds based on dissecting the probability measure

In order to have a good description of the uncertain parameters, the number of scenarios S to be considered in z_{stoch}^* is typically very large, making the problem computationally intractable. Nevertheless, chains of lower bounds can be constructed by solving sets of group sub-problems less complex than the original one, and recalculating the probabilities of each scenario in the group accordingly (see Maggioni & Pflug, 2016). We briefly recall now the construction.

Suppose again that the distribution of ξ has a finite support, i.e. ξ has a finite number of possible scenarios $\xi_s = (\tilde{B}_s, \tilde{C}_s)$ with respective positive probabilities p_s , $[1:S]$, i.e. $\Xi = \{\xi_1, \dots, \xi_S\}$.

Let

$$z_s^* := \min \{q_s(z) := z^\top \tilde{Q}_s z : z \in \Delta^n\},$$

be the deterministic optimization problem under scenario $s \in [1:S]$. The value

$$z^{1*} := \sum_{s=1}^S p_s z_s^*$$

takes the name of *wait and see* (WS) solution and it gives a lower bound to z_{stoch}^* , i.e. $z^{1*} \leq z_{\text{stoch}}^*$. Notice that in the computation of z^{1*} the value z_s^* , $s = 1, \dots, S$ of the each deterministic subproblem can be approximated with one of the different lower bounds for DStQP proposed in the previous section (see Eqs. (6)–(8)), and therefore the same is true for z^{1*} .

Better lower bounds can be obtained by dissecting the probability measure in convex combinations of probabilities defined in two or more scenarios. We consider the following refinement chain:

$$\begin{aligned} \Xi &= \{\xi_1, \dots, \xi_S\} \\ &\vdots \\ (\Xi_1^{(j)}, \Xi_2^{(j)}, \dots, \Xi_{m_j}^{(j)}) & \\ &\vdots \\ (\Xi_1^{(2)}, \Xi_2^{(2)}, \dots, \Xi_{m_2}^{(2)}) & \\ (\{\xi_1\}, \{\xi_2\}, \dots, \{\xi_S\}), & \end{aligned}$$

where each row is a collection of subsets of the probability space Ξ with the property that their union covers the whole space $\Xi = \cup_i \Xi_i^{(j)}$ for all j and that each set $\Xi_i^{(j)}$ is the union of sets from the next more refined collection

$$\Xi_i^{(j)} = \cup_{\Xi_k^{(j-1)} \subseteq \Xi_i^{(j)}} \Xi_k^{(j-1)}.$$

Denoting with

$$z^{j*} = \sum_{i=1}^{m_j} \pi_i^{(j)} z^*(\Xi_i^{(j)}),$$

where $\pi_i^{(j)} = \sum_{\xi_s \in \Xi_i^{(j)}} p_s$, we get to a chain of lower bounds expressed as follows

$$z^{1*} \leq z^{2*} \leq \dots \leq z^{j*} \leq \dots \leq z_{\text{stoch}}^*. \quad (22)$$

Notice that the higher the index j , the fewer problems have to be solved but with an increasing number of scenarios. We refer to Maggioni & Pflug (2016) for examples of constructions of refinement chains. Notice that for all these problems z^{j*} we can use the lower bounds defined in the previous section.

5. Numerical experiments

We conduct a series of experiments that investigate the performance of the upper and lower bounds proposed in the previous

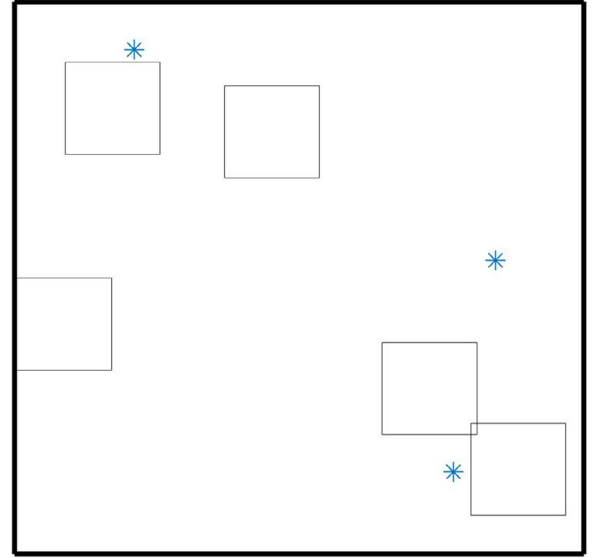


Fig. 1. Example of $n_1 = 3$ preset points (*) and the uncertainty region (squares with side length 0.2) of the other $n_2 = 5$ points.

sections. All of the experiments were implemented in Matlab, using the YALMIP environment for communication with solvers and the Mosek solver for solving semidefinite programs (SDPs). As a benchmark for our upper bounds we employed the `fmincon` function available in Matlab using an interior point solver. The experiments are aimed at answering the following questions:

- (a) How do the scalable bounds proposed in Section 4.2 perform in terms of computation time and quality of the solution when compared to the bounds based on Theorem 4?
- (b) How does the PFW algorithm for computing the upper bound proposed in Section 4.1 perform in terms of computation time and quality of the solution when compared to `fmincon`?
- (c) How good are these methods when initialized at a feasible solution obtained from the copositive relaxation?
- (d) What is the potential of using our techniques in global optimization procedures such as the ones employed by Gurobi?
- (e) When using an approximation based on dissections of the probability measure (see Section 4.3), what is the trade off between solution quality and size of the subsets of the probability space?
- (f) How consistent are the results for the approximations described in Section 4.3 for a fixed size of subsets, when the grouping of the scenarios is performed randomly?
- (g) How do our procedures perform when confronted with a real world problem such as the portfolio optimization problem?

In our experiments we generated data by two different processes for obtaining the data matrix \tilde{Q} :

For the first one, the elements of the matrix \tilde{Q} represent the mutual distances of $n = n_1 + n_2$ points sampled under a uniform distribution from the unit square $[0, 1] \times [0, 1]$; the n_1 points are fixed and generate the principal $n_1 \times n_1$ submatrix A , while for the other n_2 points it is only known that they lie in a square with side length 2ϵ . The actual location of the n_2 unknown points follows a uniform distribution in these squares and generates the matrices \tilde{B}_s and \tilde{C}_s , $s \in [1:S]$ of dimension $n_1 \times n_2$ and $n_2 \times n_2$ respectively. An example is given in Fig. 1 where the asterisks represent the $n_1 = 3$ preset points, and the squares the uncertainty regions of the other $n_2 = 5$ points.

For the second one, we choose $A_{ij} \sim \mathcal{U}_{\{0,1\}}$, $B_{ij} \sim \mathcal{U}_{\{0,1\}}$, $C_{ij} \sim \mathcal{U}_{\{0,0.1\}}$, independently of each other, where $\mathcal{U}_{\mathcal{M}}$ is the uniform distribution with support \mathcal{M} .

Table 1
Comparison of copositive bounds under the distance matrix strategy.

(n_1, n_2, S)	Time		Gap		max Gap		Vs.	
	DNN1	DNN2	DNN1	DNN2	DNN1	DNN2	avgG	maxG
(5,5,10)	1.438	0.154	0%	0%	0%	0%	0%	0%
(5,10,10)	34.246	0.230	0%	0%	0%	0%	0%	0%
(5,20,10)	1249.161	0.684	0%	0%	0%	0%	0%	0%
(5,40,10)	-	6.228	-	0%	-	0%	-	-
(10,5,10)	2.757	0.865	0%	0%	0%	0%	0%	0%
(20,5,10)	5.010	1.119	0%	0%	0%	0%	0%	0%
(40,5,10)	15.180	3.872	0%	0%	0%	0%	0%	0%
(5,5,20)	35.160	1.186	0%	0%	0%	0%	0%	0%
(5,5,40)	1539.243	1.788	0%	0%	0%	0%	0%	0%
(5,5,60)	-	2.945	-	0%	-	0%	-	-

Table 2
Comparison of copositive bounds under the uniformly random strategy.

(n_1, n_2, S)	Time		Gap		max Gap		Vs.	
	DNN1	DNN2	DNN1	DNN2	DNN1	DNN2	avgG	maxG
(5,5,10)	1.647	0.191	1.075%	1.373%	13.26%	14.167%	0.019%	0.016%
(5,10,10)	28.05	0.237	0.003%	0.039%	0.056%	0.379%	0.001%	0.001%
(5,20,10)	1132.45	0.754	0.428%	1.281%	5.136%	18.633%	0.001%	0.001%
(10,5,10)	2.942	0.438	1.365%	1.665%	17.201%	18.306%	0.014%	0.011%
(20,5,10)	5.515	0.977	9.432%	9.305%	35.772%	35.824%	0.002%	0.001%
(5,5,20)	37.588	0.546	0.807%	1.394%	13.756%	14.317%	0.029%	0.023%
(5,5,40)	1504.456	0.998	1.09%	1.579%	21.536%	21.625%	0.006%	0.004%

5.1. Comparing the copositive bounds

In the following we assessed the value and shortcomings of the compact copositive bound proposed in Section 4.2 over the traditional copositive bound from Burer (2009). To this end we picked different configurations for (n_1, n_2, S) and for all of them, we generated 20 instances from P_1 and P_2 , respectively. Then we solved the instances using the \mathcal{DNN} relaxations (where the cone of completely positive matrices \mathcal{CPP}_n is replaced by the outer approximation given by the doubly nonnegative cone $\mathcal{DNN}_n = \mathcal{S}_+^n \cap \mathcal{N}_n$) of the two copositive reformulations. The experiments of this section were conducted on an AMD Ryzen 7 1700X Eight-Core 3.4 gigahertz CPU, and 16 gigabyte RAM.

Note that from each of these relaxations we do not only get a lower bound, i.e. the optimal value of the relaxation, but also an upper bound obtained by considering the feasible values for the considered decision variables. If both bounds coincide we actually can certify optimally.

Table 1 summarizes the results obtained under the first data generation strategy, while Table 2 summarizes the results under the second one. In each of them the column **Time** gives the computation time in seconds. In the column **Gap** we give the average percentage gaps between the upper bound and the lower bound produced by the relaxations, and under **max Gap** we give the biggest such gaps that was encountered across all 20 instances. Column DNN1 refers to the results for the traditional copositive bound proposed in Burer (2009) and column DNN2 the results for our newly proposed bound given in Section 4.2. Finally, column **Vs.** reports the average percentage gap between the two copositive relaxations (avgG) and the biggest such gap across the 20 instances (maxG) respectively. Notice that two objective function values have been considered to be numerically identical if their percentage gap is smaller than 10^{-3} percent, which seems reasonable given the accuracy one can expect from semidefinite optimization software.

Table 1 shows that both the bounds are tight. It is a known phenomenon that the bounds obtained from the \mathcal{DNN} relaxations of copositive reformulations is often exact on random instances, and gaps start to appear only after introducing special structures. It is noteworthy that there is no gap between the two copositive

bounds: only the copositive bound from Burer (2009) is provably tight, while the bound we proposed is provably tight only in special cases. In conclusion our experiments show that, if there is a gap between the two copositive bounds, then it must be vanishingly small for instances generated by the distance matrix strategy. This is especially encouraging if we take into account the performance with regards to computation time (column **Time**). We see that our new bound scales much nicer with the number of scenarios S , showing a desired property for stochastic optimization problems, since the quality of the solution depends on the number of scenarios that are considered. Even instance types where the traditional \mathcal{DNN} lower bound was not solvable due to memory capacity limits (see instances (5,40,10) and (5,5,60)), our scalable lower bound was obtained in few seconds.

In Table 2 we see that under the uniformly random strategy, we were able to create instances where the relaxations actually exhibited a gap between the upper and lower bounds they produced (see columns **Gap** and **max Gap**). We found that the scalable copositive bound produced slightly bigger gaps on average. Also the percentage gap between the lower bounds seems to widen but remaining very small. It is questionable whether or not these are real gaps or whether our threshold for numerical identity (i.e. 10^{-3}) is too large. We have so far not been able to produce instances where the gaps between lower bounds are comfortably outside of what could have happened by numerical inaccuracies.

5.2. Optimality gaps without warm-starting

In Section 5.1 we have seen that we can create instances where the upper bounds generated by the copositive relaxation do not close the optimality gap. However, we can produce upper bounds using the PFW algorithm that might improve upon the upper bound generated using the copositive approach. This can be done by either starting the algorithm at an arbitrary point, perhaps restarting it multiple times, or by warm-starting the algorithm at the point previously calculated via the copositive procedure. In this section we focus on the result we obtain from cold-starting the procedure from the point $x = (n_1 + n_2)^{-1}e$ and likewise $y_s = (n_1 + n_2)^{-1}e$, all $s \in [1:S]$, or else from multi-starting

Table 3
Results of cold-starting the upper-bound procedures .

(n_1, n_2, S)	Time					Gap				
	DNN	Gurobi	fmincon	PFW	PFWM	DNN	Gurobi	fmincon	PFW	PFWM
(10,5,10)	0.274	291.987	0.701	0.031	5419	16.33%	0.32%	3.01%	1.92%	0.25%
(20,5,10)	0.492	300.406	0.994	0.061	7170	50.41%	2.13%	10.57%	2.10%	0.23%
(5,10,10)	0.362	300.483	2.622	0.039	8463	15.24%	0.09%	2.31%	1.32%	0.08%
(5,20,10)	0.614	300.743	2.872	0.082	16.843	13.76%	0.15%	2.86%	1.88%	0.04%
(5,5,10)	0.362	296.317	0.764	0.024	4.207	19.06%	0.32%	0.94%	1.26%	0.32%
(5,5,20)	0.108	300.527	1.698	0.050	13.319	19.58%	0.15%	1.29%	2.54%	0.15%

Table 4
Results of cold-starting the upper-bound procedures: best bounds and solved instances.

(n_1, n_2, S)	Best					Solved				
	DNN	Gurobi	fmincon	PFW	PFWM	DNN	Gurobi	fmincon	PFW	PFWM
(10,5,10)	0	4	1	0	15	3	5	1	3	13
(20,5,10)	1	0	1	0	18	0	0	0	0	0
(5,10,10)	0	6	2	0	13	0	8	1	3	12
(5,20,10)	0	0	0	0	20	0	0	0	0	5
(5,5,10)	0	8	8	0	5	3	10	5	4	11
(5,5,20)	0	10	7	0	4	0	9	5	1	9

it from points chosen at random. As a benchmark we also run Gurobi and fmincon, without providing an initial solution. We chose to use fmincon as benchmark since it employs an easily available interior point solver that is able to provide local solutions to non-convex problems. Hence, it is conceptually different from our Frank-Wolfe type approach but aims at achieving a similar goal, which makes the comparison worthwhile. Gurobi is a global solver, so that a direct comparison is not really meaningful. However, it is interesting to see whether our approach may aid a global procedure in that it is able to narrow the optimality gap faster, so that the results of such a pre-processing can then be further utilized by the global solver. The latter is of course only attractive if good upper bounds are obtained sufficiently fast, which is what we tried to demonstrate in these experiments. For the experiments of this and the succeeding section we used a Intel(R) Core(TM) i5-9300H CPU 2.40 gigahertz with 16 gigabyte of RAM.

In order to increase the gap in the copositive relaxation, we slightly modified the uniformly random strategy; now $A_{ij} \sim \mathcal{U}_{(0,1)}$, $B_{ij} \sim \mathcal{U}_{(0,10)}$ and $C_{ij} \sim \mathcal{U}_{(0,0.1)}$. Our reasoning was to find instances where the copositive bounds alone are not enough to satisfactorily narrow the optimality gaps, so that more elaborate machinery is necessary. The results of our experiments are summarized in Table 3. For each configuration of (n_1, n_2, S) , indicated in the first column, we generated 20 instances in this manner.

We take it that the notation in the table is self explanatory except for the mnemonic PFWM, which reports the result of multi-starting the Pairwise Frank-Wolfe algorithm at 100 random points, and then taking the best found solution as output. The optimality gaps are calculated with respect to the dual bound produced by the copositive relaxation. We see that Gurobi always exhausted its time limit, which we set to 300 seconds, but produced quite good gaps on average. However, the multistarted PFW procedure produced tighter bounds much more cheaply, albeit not as cheap as fmincon, which on the other hand performed similar to the PFW-algorithm in terms of optimality gap but much worse in terms of computation time.

It is also interesting to see how often a given procedure performed best among all the primal bounds and how often each of these were able to solve a given instance (see Table 4). We considered an instance solved if the reported gap (with respect to the dual bound from the copositive relaxation) was smaller than 0.01%.

Table 5
Optimality gaps of Gurobi relative to the dual bounds obtained from Gurobi and the copositive relaxation.

(n_1, n_2, S)	Gap		Solved	
	Gurobi S.	Gurobi+cop.	Gurobi S.	Gurobi+cop.
(10,5,10)	12.73%	0.32%	0	5
(20,5,10)	58.45%	2.13%	0	0
(5,10,10)	1.79%	0.09%	0	8
(5,20,10)	19.35%	0.15%	0	0
(5,5,10)	0.18%	0.32%	0	10
(5,5,20)	0.18%	0.15%	0	9

We see again that PFWM fares well with respect to both features. Interestingly, the instances where either n_1 or n_2 were large, turned out to be troublesome for all procedures. Note that in the numbers in the first block do not all add up to 20 since if the best and second best solutions were within a margin of 10^{-5} , both were awarded the 'best'-label.

These experiments also offered an opportunity to evaluate the potential use of the dual bounds from the copositive relaxation in a global optimizer like Gurobi. One key challenge is to provide bounds tight enough so that optimality can be proven quickly. In Table 5 we see that in our experiments, the dual bounds provided by the conic relaxation were on average better than the bounds produced by Gurobi itself, within the set time limit. Also, we never certified a solution to be optimal based on our criterion when using Gurobi's dual bound, while, when using the copositive bound, we were able to do so at least in some cases.

5.3. Narrowing the optimality gaps by warm-starting the upper bound procedures

We now investigate the usefulness of the copositive primal solution as a point for warm-starting Gurobi and the other primal procedures. To this end we used the same instances generated for the previous section and the respective copositive solution and warm-started Gurobi, fmincon and the PFW-algorithm at that solution, in order to further narrow the optimality gap. We also performed a two step procedure, where we used the improved solution generated by the PFW-algorithm as a starting point for Gurobi. We summarize the results in Table 6.

Table 6
Warmstarting the upper bound procedures: gaps and runtimes.

(n_1, n_2, S)	Gap				Time			
	DNN+FW+Gurobi	DNN+fmincon	DNN+Gurobi	DNN+FW	DNN+FW+Gurobi	DNN+fmincon	DNN+Gurobi	DNN+FW
(10,5,10)	0.26%	0.63%	0.31%	0.27%	292.431	1.139	291.379	0.534
(20,5,10)	0.54%	1.43%	1.87%	0.82%	301.150	1.797	301.163	0.796
(5,10,10)	0.08%	0.72%	0.09%	0.10%	301.038	1.819	301.048	0.591
(5,20,10)	0.04%	0.47%	0.17%	0.04%	301.598	6.654	301.595	0.908
(5,5,10)	0.32%	0.33%	0.32%	0.34%	301.769	1.125	301.419	0.548
(5,5,20)	0.15%	0.25%	0.15%	0.21%	301.089	1.982	301.006	0.505

Table 7
Warmstarting the upper bound procedures: best bounds and solved instances.

(n_1, n_2, S)	Best				Solved			
	DNN+FW+Gurobi	DNN+fmincon	DNN+Gurobi	DNN+PW	DNN+FW+Gurobi	DNN+fmincon	DNN+Gurobi	DNN+PW
(10,5,10)	9	10	5	4	13	13	7	12
(20,5,10)	9	10	1	6	1	1	0	1
(5,10,10)	6	12	4	0	12	12	8	12
(5,20,10)	5	15	0	5	3	14	0	0
(5,5,10)	6	11	9	0	11	11	10	8
(5,5,20)	6	9	9	0	9	9	9	6

Table 8
Results of the portfolio optimization experiment.

S	Obj	Gap	Time
100	163.59	0.00%	0.47
500	163.60	0.00%	1.07
1000	163.60	0.00%	2.27
5000	163.57	0.00%	10.84
10000	163.58	0.00%	22.49

We see that on average DNN+FW+Gurobi produced the best bounds. Of course, in theory, Gurobi would eventually solve the instance given enough time, which might be prohibitive. We again operated Gurobi under a 300 second time limit, which was always exhausted. Similar results were obtained via DNN+FW in a fraction of the time, which in our opinion is a desirable feature of our techniques. Using fmincon, the bounds were weaker on average and also took longer to compute. The picture changes slightly when we look at instances solved and best bounds provided as depicted in Table 7.

We see that fmincon, more often than the other methods, provides the best bound and is able to find an optimal solution (based on our criteria). This seems to indicate that fmincon has better potential to find good bounds, but is less reliable than the other methods and also slower than the PFW.

5.4. Experiments on dissecting the probability measure

This section presents some computational tests for problems being intractable involving a large number of scenarios. For this section we again used AMD Ryzen 7 1700X Eight-Core 3.4 gigahertz CPU, and 16 gigabyte RAM. We generate data by the distance matrix strategy with $n_1 = n_2 = 15$, $\epsilon = 0.1$ and considering $S = 200$ equiprobable scenarios. We constructed lower bounds as described in Section 4.3 by partitioning the scenario tree in disjoint subgroups.

We now describe the way we constructed the groups of each layer j of the dissection chain. We fix a certain group size, say n_g , with $1 < n_g < S$. The first layer always corresponds to the wait and see (WS) approach, i.e. we consider each scenario independently so that the number of "groups" is $S_1 := S$. In the second layer, we randomly group the scenarios into groups of size n_g . If the total

number of groups S_1 is not evenly divisible by n_g , a remainder group is constructed with cardinality $S_1 \bmod n_g$. This results in S_2 groups, with $S_2 = S_1 \div n_g$ if the division has no remainder, or $S_2 = (S_1 \div n_g) + 1$ if there is a remainder and an additional remainder group is added. These S_2 groups are again randomly grouped into super-groups of n_g groups of the previous layer. Thus, for the j th layer the number of super-groups is given by

$$S_j := \begin{cases} S_{j-1} \div n_g & \text{if } S_{j-1}/n_g \in \mathbb{N} \\ (S_{j-1} \div n_g) + 1 & \text{else.} \end{cases} \quad (23)$$

Fig. 2 provides a boxplot of the percentage gaps obtained from a chain of lower bounds composed by 5 layers over 20 random dissections as described below. The first bound in the chain (z^{1*}) refers to the wait and see (WS) solution and it has been obtained solving the $S_1 = 200$ scenarios independently. The next one (z^{2*}) by partitioning the scenario tree into $S_2 = 40$ subproblems of $n_g = 5$ scenarios each, solving them independently and taking their mean. These sets were again grouped making $S_3 = 8$ subproblems of 25 scenario each to obtain z^{3*} . Finally z^{4*} has been obtained by grouping 5 groups of 25 scenarios in one subproblem of 125 scenario and the remaining 3 groups of 25 scenarios in one of 75 with a total of $S_4 = 2$. The last level in the chain z^{5*} refers to the original problem z^{*}_{stoch} . The decision which scenarios, or sets of scenarios were grouped was done at random and we created 20 random dissections in this manner. For each of the dissections we calculated z^{j*} for each layer $j \in [1:5]$ using the scalable copositive bounds (which empirically we have found to be tight for the distance matrix strategy) and subsequently calculated the percentage gaps between z^{j*} and $z^{*}_{stoch} \equiv z^{5*}$.

Results show that even in the second layer, where groupings of five scenarios are considered, the values for z^{2*} substantially outperform z^{1*} across all 20 dissections passing from a percentage gap of -0.5658 to -0.1236. For deeper layers z^{3*} and z^{4*} , the quality of the bounds as well as the consistency between dissections improves, but the marginal improvement diminishes.

We repeated the experiment, this time considering a new tree with again 200 scenarios but this time with $n_g = 2$. Fig. 3 summarizes the results of this process confirming the fact that increasing the layer j , the quality of the bound improves, while the marginal improvement diminishes. In particular we notice that the seventh layer seems to outperform the eighth and the "true value" z^{*}_{stoch} . The reason for this phenomenon is the fact that deeper into the layers we have to solve exponentially larger semidefinite optimiza-

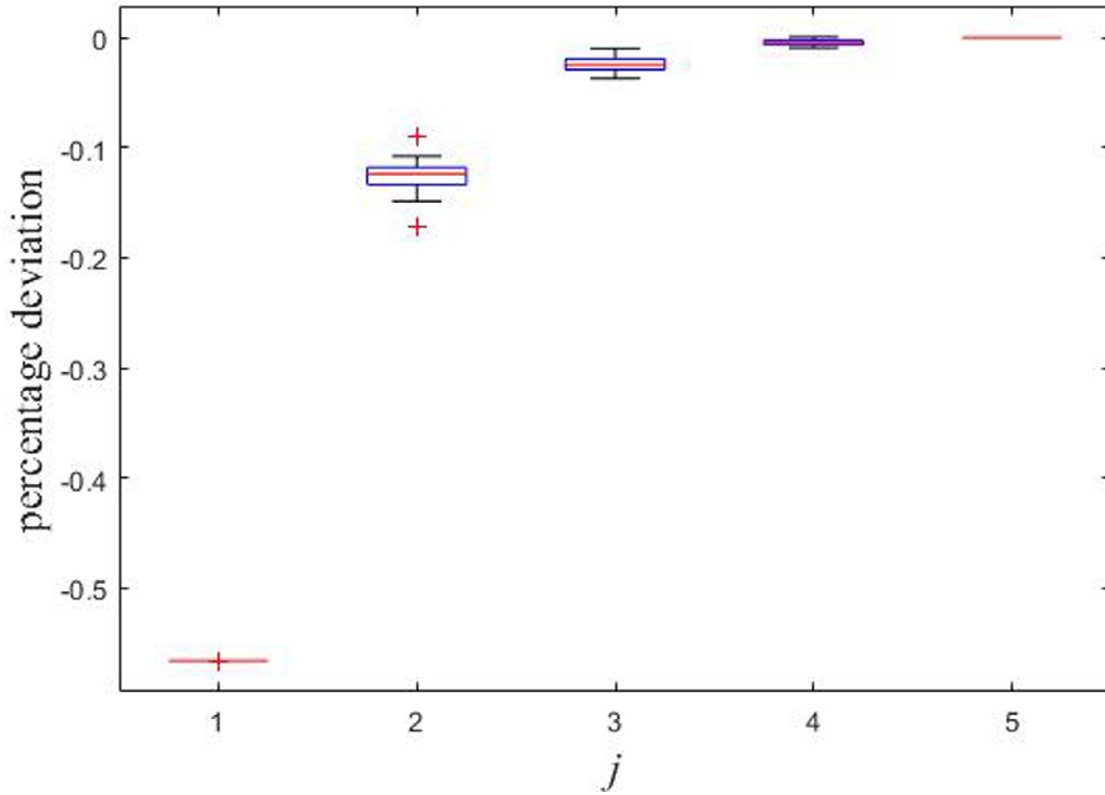


Fig. 2. Percentage deviation from the optimal objective value z_{stoch}^* for level j of a refinement chain composed by 5 layers.

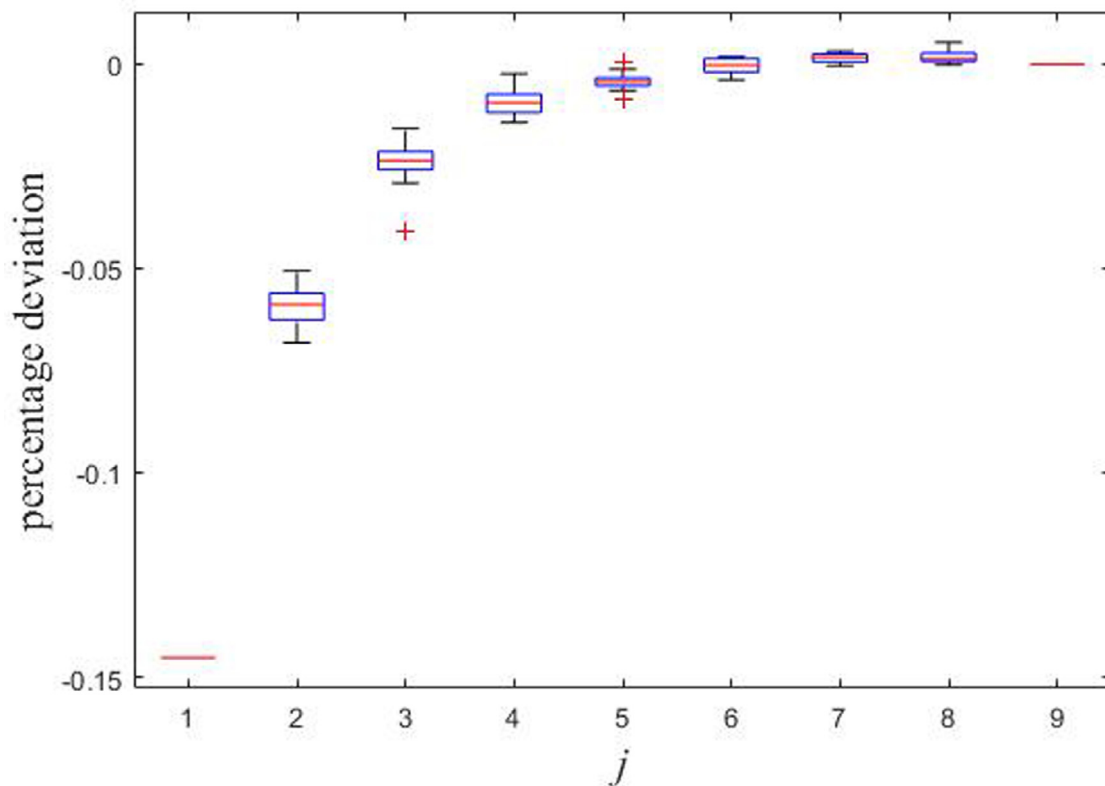


Fig. 3. Percentage deviation from the optimal objective value z_{stoch}^* for level j of a refinement chain composed by 9 layers.

tion problems which are known to have problems with scaling. Specifically, the accuracy of the solution suffers from the scale of the problem so that in fact the final few layers are numerically indistinguishable when taking into account the noise added by the solution method. This shortcoming, at least in our experiment, is partly redeemed by the fact that already for the shallow layers the quality of the bounds is satisfactory since the gap is close to zero.

5.5. Application to mean-variance-portfolio optimization

In this section we apply our framework to the classical mean-variance-portfolio optimization problem. To this end we consider the historical mean return and variance of returns of ten assets based on ten time series. For assets $i \in [1:5]$ a longer time series of 48 days is available while for assets $i \in [6:10]$ only a short time series of 12 days is available. Hence, we consider the portion of the covariance matrix that involve the latter assets to be uncertain and we consider the model:

$$\min_{x \in T^5} \left\{ \mu_x^\top x + x^\top \Sigma_{xx} x + \mathbb{E} \left[\min_{y \in P_k} \mu_y^\top y + 2x^\top \Sigma_{xy} y + y^\top \Sigma_{yy} y \right] \right\}.$$

In this model, $x, y \in \mathbb{R}^5$ are the respective shares of assets $i \in [1:5]$ and $i \in [6:10]$ in the portfolio, μ_x, μ_y are the historical mean returns and

$$\Sigma = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy}^\top \\ \Sigma_{xy} & \Sigma_{yy} \end{pmatrix}$$

is the historical covariance matrix of returns where we assume Σ_{xy} and Σ_{yy} to be uncertain. Despite the presence of a linear term, the above optimization problem can be rewritten as

$$\min_{x \in T^5} \left\{ x^\top \bar{\Sigma}_{xx} x + \mathbb{E} \left[\min_{y \in P_k} 2x^\top \bar{\Sigma}_{xy} y + y^\top \bar{\Sigma}_{yy} y \right] \right\},$$

where

$$\bar{\Sigma}_{xx} := \Sigma_{xx} + \frac{1}{2} (\mu_x e^\top + e \mu_x^\top),$$

$$\bar{\Sigma}_{xy} := \Sigma_{xy} + \frac{1}{2} (\mu_x e^\top + e \mu_y^\top),$$

$$\bar{\Sigma}_{yy} := \Sigma_{yy} + \frac{1}{2} (\mu_y e^\top + e \mu_y^\top).$$

We approximated the problem via the scenario approach, where the uncertain portions of the covariance matrix were generated from a elementwise normal distribution, i.e. $(\Sigma)_{ij} \sim N((\hat{\Sigma})_{ij}, \sigma)$ with $\hat{\Sigma}$ the empirical covariance matrix and $\sigma = 1$. Scenarios with $S \in \{100, 500, 1000, 5000, 10000\}$ were generated. The following table reports the results, where we solved instances with different and large numbers of scenarios.

In our experiments, all instances were solved using merely the reduced conic lower bound, which shows that despite the presence of hard instances we tackled in previous sections, there are real-world applications where gaps can be closed confidently using the basic but novel strategies we propose. Note that the large numbers of scenarios we worked with would have been highly prohibitive for the traditional copositive approach, while our reduced conic approach allowed Mosek to solve the respective doubly-nonnegative relaxation within seconds.

6. Conclusions

In this paper, we have defined a new class of two-stage stochastic optimization problems, namely two-stage standard stochastic quadratic optimization problems, where the objective function is quadratic in the first and second stage decisions, the feasible region is a standard simplex and the goal is to minimize the expected quadratic cost function which is not required to be convex. Possible applications of this class of problems have been introduced to motivate the investigation. In order to address

this difficult decision problem, different type of upper and lower bounds have been considered and compared with existing methods available in the literature. We proposed a scalable alternative to the copositive bound from Burer (2009) and numerical experiments show that they provide a good benefit in terms of computational resources without sacrificing solution quality. Further, we introduced a procedure for generating and improving upper bounds based on the Immunization-Infection-Dynamics algorithm. However, when compared to the `fmincon` function of Matlab, we find that results are mixed. While improving solution quality slightly, these procedures incur a substantial cost in terms of computation time. However, future research might reveal improvements which remedy these shortcomings. For problems including a large number of scenarios, bounds based on partitioning the scenario tree in sub-groups have also been considered, where the individual sub-problems were solved using our scalable copositive bound. A systematic numerical study over different instances, along with an application example taken from real-world data showed the efficiency of the proposed bounds.

Acknowledgments

M.G. was supported by the Austrian Science Fund (FWF, project W 1260-N35) as a Ph.D. student member of the Vienna Graduate School on Computational Optimization (VGSCO) involving as well I.B. and G.P. as faculty members, the latter as the speaker of the school (2016–2019). F.M. acknowledges Gruppo Nazionale per il Calcolo Scientifico (GNCS-INdAM) for the financial support.

References

- Amaral, P. A., Bomze, I. M., & Júdice, J. J. (2014). Copositivity and constrained fractional quadratic problems. *Mathematical Programming*, 146(1–2), 325–350.
- Berman, A., & Shaked-Monderer, N. (2003). *Completely positive matrices*. World Scientific.
- Birge, J. R. (1985). Aggregation bounds in stochastic linear programming. *Mathematical Programming*, 31(1), 25–41.
- Birge, J. R., & Louveaux, F. (2011). *Introduction to stochastic programming, 2nd edition*. Princeton, NJ, USA: Springer.
- Bomze, I. M. (1998). On standard quadratic optimization problems. *Journal of Global Optimization*, 13(4), 369–387.
- Bomze, I. M. (2012). Copositive optimization – recent developments and applications. *European Journal of Operational Research*, 216, 509–520.
- Bomze, I. M., & de Klerk, E. (2002). Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. *Journal of Global Optimization*, 24(2), 163–185.
- Bomze, I. M., Dür, M., De Klerk, E., Roos, C., Quist, A., & Terlaky, T. (2000). On copositive programming and standard quadratic optimization problems. *Journal of Global Optimization*, 18(4), 301–320.
- Bomze, I. M., Kahr, M., & Leitner, M. (2021a). Trust your data or not – StQP remains StQP: Community detection via robust standard quadratic optimization. *Mathematics of Operations Research*, 46(1), 301–316.
- Bomze, I. M., Locatelli, M., & Tardella, F. (2008). New and old bounds for standard quadratic optimization: dominance, equivalence and incomparability. *Mathematical Programming*, 115(1, Ser. A), 31–64.
- Bomze, I. M., & Palagi, L. (2005). Quartic formulation of standard quadratic optimization problems. *Journal of Global Optimization*, 32(2), 181–205.
- Bomze, I. M., Rinaldi, F., & Rota Bulò, S. (2019). First-order methods for the impact: Support identification in finite time with convergent Frank-Wolfe variants. *SIAM Journal on Optimization*, 29(3), 2211–2226.
- Bomze, I. M., Rinaldi, F., & Zeffiro, D. (2020). Active set complexity of the away-step Frank-Wolfe algorithm. *SIAM Journal on Optimization*, 30(3), 2470–2500.
- Bomze, I. M., Rinaldi, F., & Zeffiro, D. (2021). Frank-Wolfe and friends: a journey into projection-free first-order optimization methods. *4OR – A Quarterly Journal of Operations Research*, 19(3), 313–345.
- Bomze, I. M., Schachinger, W., & Ullrich, R. (2018). The complexity of simple models – A study of worst and typical hard cases for the standard quadratic optimization problem. *Mathematics of Operations Research*, 43(2), 651–674.
- Burer, S. (2009). On the copositive representation of binary and continuous nonconvex quadratic programs. *Mathematical Programming*, 120(2), 479–495.
- Burer, S. (2012). Copositive programming. In *Handbook on semidefinite, conic and polynomial optimization* (pp. 201–218). Springer.
- Burer, S., & Anstreicher, K. M. (2013). Second-order-cone constraints for extended trust-region subproblems. *SIAM Journal on Optimization*, 23(1), 432–451.
- Chen, X., & Peng, J. (2015). New analysis on sparse solutions to random standard quadratic optimization problems and extensions. *Mathematics of Operations Research*, 40(3), 725–738.

- Chen, X., Peng, J., & Zhang, S. (2013). Sparse solutions to random standard quadratic optimization problems. *Mathematical Programming*, 141(1), 273–293.
- Edmundson, H. (1956). Bounds on the expectation of a convex function of a random variable. *Technical Report*. RAND Corporation, Santa Monica.
- Frauendorfer, K. (1988). Solving SLP recourse problems with binary multivariate distributions: The dependent case. *Mathematics of Operations Research*, 13(3), 377–394.
- Frauendorfer, K., Kuhn, D., & Schürle, M. (2011). Barycentric bounds in stochastic programming: Theory and application. In G. Infanger (Ed.), *Stochastic programming: The state of the art in honor of George B. Dantzig* (pp. 67–96). New York, NY: Springer New York.
- Frauendorfer, K., & Schürle, M. (2009). Multistage stochastic programming: Barycentric approximation multistage stochastic programming: Barycentric approximation. In C. A. Floudas, & P. M. Pardalos (Eds.), *Encyclopedia of optimization* (pp. 2527–2531). Boston, MA: Springer US.
- Gondzio, J., & Yıldırım, E. A. (2021). Global solutions of nonconvex standard quadratic programs via mixed integer linear programming reformulations. *Journal of Global Optimization*, 81(2), 293–321.
- Hausch, D., & Ziemba, W. (1983). Bounds on the value of information in uncertain decision problems II. *Stochastics*, 10(3-4), 181–217.
- Huang, C., Vertinsky, I., & Ziemba, W. (1977a). Sharp bounds on the value of perfect information. *Operations Research*, 25(1), 128–139.
- Huang, C., Ziemba, W., & Aharon, B.-T. (1977b). Bounds on the expectation of a convex function of a random variable: With applications to stochastic programming. *Operations Research*, 25(2), 315–325.
- Jensen, J. (1906). Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1), 175–193.
- King, A. J., & Wallace, S. W. (2012). *Modeling with stochastic programming*. Springer Science & Business Media.
- Kuhn, D. (2005). Generalized bounds for convex multistage stochastic programs. *Lecture Notes in Economics and Mathematical Systems*: vol. 548. Springer Verlag Berlin Heidelberg.
- Kuhn, D. (2008). Aggregation and discretization in multistage stochastic programming. *Mathematical Programming*, 113(1), 61–94.
- Lacoste-Julien, S. (2016). Convergence rate of Frank-Wolfe for non-convex objectives. arXiv:1607.00345
- Liuzzi, G., Locatelli, M., & Piccialli, V. (2019). A new branch-and-bound algorithm for standard quadratic programming problems. *Optimization Methods and Software*, 34(1), 79–97.
- Madansky, A. (1960). Inequalities for stochastic linear programming problems. *Management Science*, 6(2), 197–204.
- Maggioni, F., Allevi, E., & Bertocchi, M. (2014). Bounds in multistage linear stochastic programming. *Journal of Optimization Theory and Applications*, 163(1), 200–229.
- Maggioni, F., Allevi, E., & Bertocchi, M. (2016). Monotonic bounds in multistage mixed-integer stochastic programming. *Computational Management Science*, 13(3), 423–457.
- Maggioni, F., & Pflug, G. C. (2016). Bounds and approximations for multistage stochastic programs. *SIAM Journal on Optimization*, 26(1), 831–855.
- Maggioni, F., & Pflug, G. C. (2019). Guaranteed bounds for general nondiscrete multistage risk-averse stochastic optimization programs. *SIAM Journal on Optimization*, 29(1), 454–483.
- Pflug, G., & Pichler, A. (2014). *Multistage stochastic optimization*. Springer.
- Pflug, G. C., & Pichler, A. (2011). Approximations for probability distributions and stochastic optimization problems. In M. Bertocchi, G. Consigli, & M. A. H. Dempster (Eds.), *Stochastic optimization methods in finance and energy: New financial products and energy market strategies* (pp. 343–387). New York, NY: Springer New York.
- Rosa, C. H., & Takriti, S. (1999). Improving aggregation bounds for two-stage stochastic programs. *Operations Research Letters*, 24(3), 127–137.
- Rota Buló, S., & Bomze, I. M. (2011). Infection and immunization: A new class of evolutionary game dynamics. *Games and Economic Behavior*, 71(1), 193–211. Special Issue In Honor of John Nash
- Ruszczynski, A. P., & Shapiro, A. (2003). *Stochastic programming*: vol. 10. Elsevier Amsterdam.
- Sandıkçı, B., Kong, N., & Schaefer, A. J. (2013). A hierarchy of bounds for stochastic mixed-integer programs. *Mathematical Programming*, 138(1), 253–272.
- Scozzari, A., & Tardella, F. (2008). A clique algorithm for standard quadratic programming. *Discrete Applied Mathematics*, 156(13), 2439–2448. Fifth International Conference on Graphs and Optimization