Membership testing for Bernoulli and tail-dependence matrices

Daniel Krause

Chair of Mathematical Finance,
Technische Universität München,
Parkring 11, 85748 Garching-Hochbrück, Germany,
email: daniel.krause@tum.de, phone: +49(0)89-289-17400

Matthias Scherer

Chair of Mathematical Finance,
Technische Universität München,
Parkring 11, 85748 Garching-Hochbrück, Germany,
email: scherer@tum.de, phone: +49(0)89-289-17402

Jonas Schwinn

Institute of Mathematics – Business Mathematics,
Universität Augsburg,
Universitätsstraße 14, 86159 Augsburg, Germany,
email: jonas.schwinn@math.uni-augsburg.de, phone: +49(0)821-598-2160

Ralf Werner

Institute of Mathematics – Business Mathematics,
Universität Augsburg,
Universitätsstraße 14, 86159 Augsburg, Germany,
email: ralf.werner@math.uni-augsburg.de, phone: +49(0)821-598-5854

Version of: August 16, 2017

Keywords: Bernoulli-compatible matrix, tail-dependence matrix, column generation, binary quadratic programming

Abstract

Testing a given matrix for membership in the family of Bernoulli matrices is a longstanding problem, the many applications of Bernoulli vectors in computer science, finance, medicine, and operations research emphasize its practical relevance. A novel approach towards this problem was taken by [Fiebig et al., 2017] for low-dimensional settings $d \leq 6$. For the first time, they exploit the close relationship between the Bernoulli polytope (also known as correlation polytope) and the well-studied cut polytope, which plays a central role in membership testing of Bernoulli matrices. Inspired by this approach, we use results from [Deza and Laurent, 1997, Embrechts et al., 2016, Fiebig et al., 2017] in a pre-phase of our algorithm to check necessary and sufficient conditions, before actually testing a matrix on Bernoulli compatibility. In our main approach, we – however – build upon an early attempt by [Lee, 1993] based on the vertex representation of the correlation polytope and directly solve the corresponding linear program. To appropriately deal with the issue of exponentially many primal variables, we propose a specifically taylored column generation method. A straightforward, yet novel, analysis of the arising subproblem of determining the most violated dual constraint in the column generation process leads to an exact algorithm for membership testing. Although the membership problem is known to be NP-complete, we observe very promising performance up to dimension $d = 40$ on a broad variety of test problems.

1 Introduction and motivation

1.1 Motivation

Characterizing a correlation matrix in terms of its algebraic properties is classical content of an introductory course to multivariate statistics. The closely related question, however, namely testing if a matrix $B \in \mathbb{R}^{d \times d}$ is a *Bernoulli matrix* or a *matrix of pairwise tail-dependence coefficients*[1] is much harder. Literature related to this problem is spread over different communities ranging from probability and operations research to applications in various disciplines, see for example [Macke et al., 2009] and other references listed in this paper. This results in an inconsistent notation and nomenclature that makes it challenging to keep track of all relevant results. Our original interest in the problem stems from a probabilistic treatment by [Embrechts et al., 2016], research inspired by an actuarial application, which ends with the statement: "*Concerning future research, an interesting open question is how one can (theoretically or numerically) determine whether a given arbitrary nonnegative, square matrix is a tail-dependence or Bernoulli-compatible matrix. To the best of our knowledge there are no corresponding algorithms available.*" From a methodological point of view, however, much closer is the deep mathematical investigation of the geometry of the problem by [Fiebig et al., 2017] who succeeds in characterizing low-dimensional cases $d \leq 6$ in terms of an analysis of the geometry of the closely related cut polytope.

[1] It has been found in [Embrechts et al., 2016, Fiebig et al., 2017] that the membership problem for tail-dependence matrices can be reduced to the membership problem for Bernoulli matrices. Hence, we focus without loss of generality on this latter membership problem.

1.2 Review of existing literature

The abovementioned problem appears (explicitely or implicitely) in different communities. From a probabilistic point of view, the problem of working with multivariate Bernoulli vectors has, for instance, been treated in [Teugels, 1990, Qaqish, 2003, Preisser and Qaqish, 2014]. In [Qaqish, 2003] it is mentioned that "*However, specifying or computing p becomes impractical for n greater than about* 15.", which is linked to the exponentially increasing number of parameters in the problem, as also observed in [Teugels, 1990]. Correlation matrices of Bernoulli variables have also been studied by [Chaganty and Joe, 2006], one of their findings being that positive definiteness is a necessary but not sufficient criterion. Other authors have used specific families of copulas to create valid families of Bernoulli matrices, see for example [Emrich and Piedmonte, 1991] for a very prominent one, but this is of limited use for the membership problem we are dealing with. Given a certain matrix $T \in \mathbb{R}^{d\times d}$, the question of constructing a valid random vector $X = (X_1, \dots, X_d)^\top$ such that T_{ij} is precisely the tail-dependence coefficient of X_i and X_j was investigated in [Embrechts et al., 2016] and linked to the corresponding Bernoulli problem, a result derived independently in [Fiebig et al., 2017]. The paper by [Fiebig et al., 2017] contains, among other interesting results, a mathematical description of the underlying polytopes in dimensions $d \leq 6$. Further details on the corresponding polytopes and their characterization can, for example, be found in [Deza and Laurent, 1997]. Taking into account results by [Pitowsky, 1991] for the correlation polytope, we know that this problem is NP-complete and, hence, a simple solution to the problem cannot be expected.

1.3 Problem formulation

Research questions: For a given symmetric matrix $B \in \mathbb{R}^{d\times d}$, decide (numerically) if there exists a d-dimensional random vector X on some probability space $(\Omega, \mathcal{A}, \mathbb{P})$ such that each component X_i, $i = 1, \dots, d$ is Bernoulli-distributed and such that

$$B = \mathbb{E}_{\mathbb{P}}[XX^\top]. \tag{B}$$

If such a random vector exists, provide a method to draw random samples from this multivariate distribution. In this case, B is called *Bernoulli-compatible* (or *Bernoulli matrix* in short), otherwise B is called *Bernoulli-incompatible.*

As on each probability space $(\Omega, \mathcal{A}, \mathbb{P})$ it holds that

$$\mathbb{E}_{\mathbb{P}}[XX^\top] = \sum_{\mathbf{p}\in\{0,1\}^d} \mathbb{P}[X = \mathbf{p}]\, \mathbf{p}\mathbf{p}^\top$$

we obtain the following result.

Proposition 1.1 (Vertex representation of the Bernoulli polytope)
A matrix $B \in \mathbb{R}^{d \times d}$ is Bernoulli-compatible if and only if $B \in \mathcal{B}_d$, where $\mathcal{B}_d$ denotes the *Bernoulli polytope*

$$\mathcal{B}_d := \operatorname{conv}\left(\{\mathbf{p}\mathbf{p}^\top \mid \mathbf{p} \in \{0,1\}^d\}\right).$$

A formal proof of this Proposition can, for example, be found in [Embrechts et al., 2016], Theorem 2.2.

Remark 1.2 (Nomenclature in the OR community)
The Bernoulli polytope is a well-known polytope in the operations research community, where it is known under the name *correlation polytope*, introduced by [Pitowsky, 1991]. Already in [Pitowsky, 1991], it is shown that membership testing for this polytope is NP-complete. In Section 2.2 we provide more details on this important aspect of the problem. As already mentioned, due to this NP-completeness, it cannot be expected that there is any method that solves problem (B) for large d (i.e. $d > 40$) in reasonable time.

However, due to the well-researched structure of the correlation polytope and its connections to binary quadratic programming, cf. [Pitowsky, 1991, Deza and Laurent, 1997], we can hope for methods that solve at least medium sized instances (i.e. $20 \le d \le 40$) within a few minutes of computation time.

1.4 Membership testing by linear programming

Following the main idea of [Lee, 1993], from Proposition 1.1 it immediately follows that testing $B \in \mathcal{B}_d$ is equivalent to solving the following optimization problem

$$vp(B) := \min_{a \in \Lambda_{2^d}} |||\sum_{i=0}^{2^d-1} a_i \mathbf{B}_i - B|||_\infty,$$

where $|||A|||_\infty$ denotes the matrix max-norm of the matrix A, $\mathbb{1} = (1, \ldots, 1)^\top$, $\Lambda_m := \{\lambda \in \mathbb{R}^m_+ \mid \lambda^\top \mathbb{1} = 1\}$, and $\mathbf{B}_i = \mathbf{p}(i)\mathbf{p}(i)^\top$, $i = 0, \ldots, 2^d - 1$. Here, $\mathbf{p}(i)$ denotes the natural bijection[2] between all integers between 0 and $2^d - 1$ and all $\{0,1\}$-vectors of dimension d.

The interpretation of this problem is straightforward: find the convex combination of the vertices $\mathbf{B}_i$ of the Bernoulli polytope which is as close as possible to the given matrix

[2] More formally, the bijection $\mathbf{p} : \{0, \ldots, 2^d - 1\} \to \{0,1\}^d$ is given as the inverse of the bijection $\mathbf{i} : p \mapsto \sum_{j=1}^{d} p_j \cdot 2^{j-1}$ which maps $\{0,1\}$-vectors bijectively onto the integers $\{0, \ldots, 2^d - 1\}$.

B in matrix max-norm. This problem can easily be reformulated as a linear program:

$$\begin{aligned} vp(B) = \quad & \min_{\substack{a \in \Lambda_{2^d} \\ \alpha \in \mathbb{R}}} \quad \alpha \\ & \text{s.t.} \quad B \leq \sum_{i=0}^{2^d-1} a_i \mathbf{B}_i + \alpha E \\ & \qquad \sum_{i=0}^{2^d-1} a_i \mathbf{B}_i - \alpha E \leq B, \end{aligned} \tag{PB}$$

with $E = \mathbb{1}\mathbb{1}^\top$.

Remark 1.3 (Non-uniqueness of representation)
It has to be noted that if such a representation exists, it is not necessarily unique. An example illustrating that different stochastic models for the random vector $(X_1, X_2, X_3)^\top$ might imply the same Bernoulli matrix is the following: Independent and Bernoulli($\frac{1}{2}$) distributed $X_i, i = 1, 2, 3$ imply a Bernoulli matrix B with $\frac{1}{2}$ on the diagonal and $\frac{1}{4}$ on the off-diagonal. This can be constructed by: (i) the convex combination taking $\frac{1}{4}$ times the matrix E and $\frac{1}{4}$ times each of the matrices $e_i e_i^\top, i = 1, 2, 3$ or (ii) taking 1/8 times each of the eight extremal matrices in the tri-variate case. Interestingly, representation (ii) used in a mixture model corresponds to independence.

Remark 1.4 (Answer to the second part of the research question)
In case $B \in \mathcal{B}_d$, the well-known Theorem of Carathéodory yields that there is at least one representation of B which needs at most $d(d+1)/2 + 1$ vertices. However, it is not (yet) clear how such a sparse representation can be efficiently determined.

Practically more important – due to the fundamental theorem of linear programming (FTLP) – if the Simplex method is used to solve (PB), it will always yield a solution with at most[3] $d(d+1)$ strictly positive a_i. From such a similarily sparse representation, a random number generator can be built in a straightforward manner. This already yields an answer to the second part of the research question.

Remark 1.5 (Different formulations for membership testing)
As already noted in [Lee, 1993], the membership test (B) boils down to a test of the feasibility of a linear system of inequalities, or a convex hull problem, respectively. If the problem is formulated as an LP, the choice of the objective function is not determined. This allows to consider a variety of objective functions, which might be more suitable for certain purposes than the *distance-like* approach followed here. Interestingly, Lee's insight to formulate the test problem as an LP has not attracted any follow-up research on an efficient solution of the arising LP.

For our numerical approach, we also need to consider the corresponding dual problem. So far, the dual problem has not received much attention among probabilists,

[3] A close inspection of the structure of the constraints in (PB) actually yields the same number of non-zeros as the Theorem of Caratheodory.

with [Fiebig et al., 2017] constituting an exception. The dual problem to (PB) is given by:

$$
\begin{aligned}
vd(B) := \quad & \max_{\substack{Y,Z \in \mathbb{S}_d \\ Y,Z \geq 0 \\ \gamma \in \mathbb{R}}} \quad \langle B, Z - Y \rangle + \ \gamma \\
& \text{s.t.} \quad \langle \mathbf{B}_i, Z - Y \rangle + \gamma \leq 0 \quad i = 0, \dots, 2^d - 1 \\
& \qquad\ \ \langle E, Z + Y \rangle = 1
\end{aligned}
\tag{DB}
$$

where $\mathbb{S}_d$ denotes the Hilbert space of all real symmetric $d \times d$ matrices with scalar product $\langle G, H \rangle := \text{trace}(G^\top H)$ and its corresponding Frobenius norm.

The dual program (DB) can be interpreted as finding a separating hyperplane, represented by a matrix $G = Z - Y$, which separates the matrix B from the Bernoulli polytope, if possible. More exactly, the following theorem holds:

Theorem 1.6 (Membership testing by linear programming)
The following statements hold for any matrix $B \in \mathbb{S}_d$ with $B \in [0,1]^{d \times d}$:

1. Problems (PB) and (DB) are both feasible and it holds $0 \leq vd(B) \leq vp(B) \leq 0.5$.
2. Problems (PB) and (DB) possess a primal optimal solution (a^*, α^*) and a dual optimal solution (Y^*, Z^*, γ^*), respectively, and strong duality holds for (PB) and (DB), i.e. $vp(B) = ||| \sum_{i=0}^{2^d-1} a_i^* \mathbf{B}_i - B |||_\infty = \langle B, Z^* - Y^* \rangle + \ \gamma^* = vd(B)$.
3. $B \in \mathcal{B}_d \iff vp(B) = vd(B) = 0$.
4. $vd(B) > 0 \iff$ the dual optimal solution $G^* = Z^* - Y^*$ strictly separates B from the Bernoulli polytope.

The two conditions $B \in \mathbb{S}_d$ and $B \in [0,1]^{d \times d}$ are obviously necessary for any matrix B to be Bernoulli-compatible, cf. Proposition 2.1, therefore, we can focus on such matrices only in the above theorem.

Remark 1.7 (Early termination criterion for (DB))
It is important to note that according to (4) any dual feasible point with strictly positive objective value – i.e. a hyperplane separating B from the Bernoulli polytope – already gives a certificate that the matrix B is not Bernoulli-compatible. This especially means that in this case the dual program does not need to be solved to optimality for testing purposes.

Proof (of Theorem 1.6)

1. The statement follows from the observation that $(\hat{a}, \hat{\alpha})$ with $\hat{a}_i := 0$ for $i = 0, \dots, 2^d - 2$, $\hat{a}_i := 1/2$, for $i = 2^d - 1$ and $\hat{\alpha} := 1/2$, is primally feasible and that $(\hat{Y}, \hat{Z}, \hat{\gamma})$ with $\hat{Y} := 1/(2d^2)E$, $\hat{Z} := \hat{Y}$ and $\hat{\gamma} := 0$ is dually feasible. The general inequality is due to the weak duality for linear programs.

2. The second statement is a direct consequence of (1) together with strong LP duality theory.

The remaining statements follow from the first two statements in a straightforward manner. □

Remark 1.8 (Extension complexity of the correlation polytope)
Let us point out that problem (DB) has 2^d many linear inequality constraints. One might wonder, whether one is able to derive a formulation with significantly less constraints. Unfortunately, this is out of reach, as for instance demonstrated by [Kaibel and Weltge, 2015] in a nicely accessible way. There, it has been shown that there is no formulation as a linear program with less than $\mathbf{O}(1.5^d)$ constraints.

1.5 Main contribution and outline of the rest of the paper

In the remainder of this exposition, we first collect a variety of properties of Bernoulli matrices (classical necessary and sufficient conditions as well as less known connections to binary quadratic programs and novel connections to $\{0,1\}$-completely positive matrices), before we scrutinize in detail the numerical solution of (PB) and (DB). As a side effect, we thereby obtain a novel necessary condition for a matrix to be a $\{0,1\}$-completely positive matrix, which can be numerically verified. In essence, the easy-to-verify necessary and sufficient conditions can naturally be exploited in a pre-phase to avoid high numerical effort, before an actual membership test is performed.

Taking advantage of the increased computational power since the early paper by [Lee, 1993], we directly tackle the corresponding LPs by a standard LP solver. This is successful for dimensions up to $d \leq 17$. Due to the expected memory issues for larger d, to circumvent the curse of dimensionality, we then solve the primal-dual pair of LPs with a specifically designed column generation approach (or, taking the dual point of view, a cutting plane method). This already allows to solve higher dimensional problems ($d \leq 30$); however, the computation times still suffer severely from the expensive search for the most violated dual constraint in the column generation process.

To significantly enhance the performance of standard column generation methods, we therefore analyse the subproblem of finding the most violated dual constraint. We demonstrate that this subproblem can be reduced to a standard binary quadratic problem. The replacement of the exhaustive search by the BQP then results in a significant improvement in terms of calculation time.

Using a battery of test problems, we illustrate the efficiency of our algorithm by solving problems up to dimension $d = 30$ without numerical issues. By exploiting several heuristics (improved primal starting point, early termination criterion, and dual test on incompatibility) we can finally solve problems up to $d = 40$ within reasonable computation time.

2 Bernoulli-compatible matrices

Although Theorem 1.6 is our basis for testing a matrix B on Bernoulli compatibility, a lot of computation time can be saved, if efficient preliminary tests are run. For this purpose, in this section we recall[4] known necessary and sufficient conditions for a given matrix B to be Bernoulli-compatible. Special emphasis is put on criteria that can easily be verified numerically, although we also briefly touch upon further known properties for the reader's convenience. For some of these properties, we provide (sometimes novel) easily accessible proofs motivated from the stochastic representation. Further, we recall the less known equivalence of membership testing for the Bernoulli polytope to solving a binary quadratic program (BQP), which provides more insights into the complexity of testing Bernoulli compatibility. Finally, we provide a few novel connections to the class of $\{0, 1\}$-completely positive matrices. The section concludes by putting Bernoulli matrices into context concerning the hierarchy of popular convex cones used in combinatorial optimization.

2.1 Necessary and sufficient conditions

In the following proposition, several easy-to-verify properties of Bernoulli-compatible matrices are summarized.

Proposition 2.1 (Necessary conditions)
Let $B \in \mathbb{R}^{d\times d}$. Then each of the following conditions is necessary for B to be Bernoulli-compatible.

1. $B \in \mathbb{S}^d$.
2. $B \in [0, 1]^{d\times d}$.
3. B is positive semidefinite.
4. B satisfies the Fréchet–Hoeffding bounds, i.e.

$$\max(0, B_{ii} + B_{jj} - 1) \leq B_{ij} \leq \min(B_{ii}, B_{jj}).$$

Proof
See, for example, [Embrechts et al., 2016], Proposition 2.1 or [Fiebig et al., 2017]. □

Let us give some further necessary conditions for a matrix B to be Bernoulli-compatible based on *gap inequalities*[5] for the cut polytope.

[4] We mainly follow [Embrechts et al., 2016], [Fiebig et al., 2017], and [Deza et al., 1993, Deza and Laurent, 1997] in our presentation.

[5] For a concise introduction to the topic of *gap inequalities* for the cut polytope, and especially *hypermetric inequalities*, let us refer to [Laurent and Poljak, 1996] and [Deza et al., 1993, Deza and Laurent, 1997].

Proposition 2.2 (Necessary gap inequalities)
Let $B \in \mathbb{R}^{d \times d}$. Then each of the following conditions is necessary for B to be Bernoulli-compatible.

1. Negative type inequalities: $\forall z \in \mathbb{Z}^d : z^\top B z \geq 0$.
2. (Unrooted) Triangle inequalities:

$$\begin{aligned} \forall i, j, k : 0 &\leq B_{ij} - B_{jk} - B_{ik} + B_{kk} \leq 1 \\ 0 &\leq B_{ii} + B_{jj} + B_{kk} - B_{ij} - B_{ik} - B_{jk} \leq 1. \end{aligned}$$

3. Hypermetric inequalities: $\forall z \in \mathbb{Z}^d : z^\top B z \geq z^\top \mathrm{diag}(B)$.

Remark 2.3 (Application for membership testing)
A few comments are in order:

- The negative type inequalities for a matrix B are actually equivalent to the easy-to-verify positive semidefiniteness of the matrix B, and thus do not provide further information.
- The triangle inequalities constitute $\mathbf{O}(d^3)$ many easy-to-verify linear constraints on Bernoulli-compatible matrices. However, they can be derived directly from the hypermetric inequalities. Nevertheless, it is advisable that the triangle inequalities are tested for in advance. Sometimes, the Fréchet–Hoeffding bounds are also referred to as rooted triangle inequalities.
- Finally, each single inequality out of the infinitely many hypermetric inequalities[6] is again easy-to-verify. These inequalities are not implied by the previous necessary conditions. Actually, as for example shown in [Deza and Laurent, 1997], Proposition 14.2.4, the hypermetric inequalities already follow from a finite subset of these; at the time being the size of this subset grows faster than exponentially with d. For a fixed positive definite matrix B, it is straightforward to show that only exponentially many inequalities have to be verified.

  Unfortunately, results in [Avis and Grishukhin, 1993, Deza and Laurent, 1997] indicate that testing all hypermetric inequalities and finding a violated one, if there is any, is a hard problem itself. For more details on the complexity, and on the complexity of related questions, let us refer to [Deza and Laurent, 1997], Section 28.3.
- Besides the most important inequalities which we have stated here, there are further inequalities known in the literature, see for example [Deza and Laurent, 1997] for an overview.
- Of course, valid inequalities can also be easily derived by choosing appropriate elements from the dual cone WP_d^+ constructed in Section 2.4.

[6] To the best of our knowledge, [Fiebig et al., 2017] were the first (and so far only) to consider hypermetric inequalities in the context of Bernoulli-compatible matrices.

Proof (of Proposition 2.2)

1. Without loss of generality, z can be chosen from $\mathbb{Q}^d$ instead of $\mathbb{Z}^d$, the remaining statement for $\mathbb{R}^d$ follows by a continuity argument; thus, negative type inequalities are equivalent to the positive semidefiniteness of B.

2. All triangle inequalities follow immediately from an inspection of the eight possible outcomes for (X_i, X_j, X_k). As shown in [Deza and Laurent, 1997], Sections 27 and 28, they are a direct consequence of the hypermetric inequalities.

3. Let us rewrite the hypermetric inequality as:

$$\forall z \in \mathbb{Z}^d : \langle B, zz^\top - \operatorname{diag}(z) \rangle \geq 0$$

which is valid, as obviously $zz^\top - \operatorname{diag}(z) \in WP_d^+$ (see Definition 2.12 and Proposition 2.14).

An alternative, more elementary argumentation for Assertion 3 is as follows: note that for each Bernoulli vector X (or equivalently, each vertex of the Bernoulli polytope) it holds that $z^\top XX^\top z - z^\top X = (z^\top X)^2 - z^\top X$ which is obviously non-negative as $z^\top X$ only takes integer values. The assertion then follows by taking the expectation (or equivalently, convex combinations). □

For numerical purposes, as not all (of the at least exponentially many) hypermetric inequalities can be tested in advance, a few (small) $z \in \mathbb{Z}^d$ can be randomly selected and tested in advance. However, as we are interested in deterministic tests, we do not pursue this idea any further in this exposition and refrain from testing randomly chosen hypermetric inequalities.

Motivated by the corresponding result in [Berman and Xu, 2007], Theorem 2.1, we can also derive an upper bound on the i-th coefficient in the representation of a Bernoulli matrix.

Proposition 2.4 (Upper bounds on the probability of individual events)
Let $B \in \mathbb{R}^{d\times d}$ be a non-singular Bernoulli-compatible matrix with representation $B = \sum_{i=0}^{2^d-1} a_i \mathbf{B}_i$. Then it holds

$$a_i \leq \left(\mathbf{p}(i)^\top B^{-1} \mathbf{p}(i)\right)^{-1} \quad \text{and} \quad a_i \leq \min_{(k,l):(\mathbf{B}_i)_{kl}=1} B_{kl}$$

for all $i = 1, \ldots, 2^d - 1$.

Proof
Let $a_i > 0$, otherwise the two inequalities are trivially satisfied. Then, from the representation of B we obtain that $B - a_i \mathbf{p}(i)\mathbf{p}(i)^\top$ is positive semidefinite. The first statement now follows from standard reformulations in matrix algebra based on the Schur complement. The second statement is obvious. □

Thus, whenever a new variable is added within the column generation approach, we can immediately bound this variable by the bounds given above. Please note that it is a priori not clear which of the two bounds is the better one.

Starting from the dual LP (DB), let us now provide a novel, yet easy-to-verify, necessary condition for a matrix B to be Bernoulli-compatible. This condition relies on the idea that instead of finding a general cut separating the Bernoulli polytope from B, we aim at finding a cut with a very simple (low-dimensional) structure, where the separation test can be easily computed. As our numerical experiments show, this simple test is surprisingly effective in finding Bernoulli-incompatible matrices in practical settings.

Proposition 2.5 (Necessary condition via dual approximation)
Let $B \in \mathbb{R}^{d\times d}$ satisfy all necessary conditions from Proposition 2.1. If the optimal value of the LP

$$\begin{aligned}\min_{\alpha_Y,\beta_Y,\alpha_Z,\beta_Z,\gamma\in\mathbb{R}} \quad & (\alpha_Z-\alpha_Y)\operatorname{trace}(B)+(\beta_Z-\beta_Y)\langle B,E\rangle+\gamma\\ \text{s.t.}\quad & \beta_Y\ge 0,\quad \beta_Z\ge 0,\\ & \alpha_Y+\beta_Y\ge 0,\quad \alpha_Z+\beta_Z\ge 0,\\ & d(\alpha_Y+\beta_Y+\alpha_Z+\beta_Z)+(d^2-d)(\beta_Y+\beta_Z)=1,\\ & k(\alpha_Z-\alpha_Y)+k^2(\beta_Z-\beta_Y)+\gamma\le 0,\quad k=0,\dots,d.\end{aligned}$$

is strictly positive, then B is not Bernoulli-compatible.

Proof
Set $Y=\alpha_Y I+\beta_Y E$ and $Z=\alpha_Z I+\beta_Z E$. Then the triple (Y,Z,γ) is feasible in (DB) if and only if

$$\begin{aligned}& \beta_Y\ge 0,\quad \beta_Z\ge 0,\\ & \alpha_Y+\beta_Y\ge 0,\quad \alpha_Z+\beta_Z\ge 0,\\ & d(\alpha_Y+\beta_Y+\alpha_Z+\beta_Z)+(d^2-d)(\beta_Y+\beta_Z)=1,\quad \text{and}\\ & \langle \mathbf{B}_i, Z-Y\rangle+\gamma\le 0,\quad i=0,\dots,2^d-1.\end{aligned}$$

Due to the special structure of Y and Z, the last 2^d inequalities boil down to only $d+1$ inequalities:

$$k(\alpha_Z-\alpha_Y)+k^2(\beta_Z-\beta_Y)+\gamma\le 0,\quad k=0,\dots,d.$$

Thus, each feasible point of this reduced approximation is feasible in (DB) as well, which shows the claim. □

In the following, let us provide two obvious necessary and sufficient conditions. The second condition is quite useful for theoretical purposes (but not for numerical purposes), as it shows that a downscaled version of a Bernoulli-compatible matrix remains Bernoulli-compatible. In contrast, the first condition comes in quite handy for the determination of a suitable starting point for the solution of the primal LP. Accordingly, this has been used in our numerical implementation: Let us consider low dimensional principal diagonal blocks of the matrix B (e.g. $d = 10$). For these kind of blocks, the primal LP can be solved very efficiently. If one of these blocks is not Bernoulli-compatible, the same holds for B. In the other case, we can easily use these "local" solutions to construct a feasible starting point for the original LP. This starting point already fits the selected blocks and thus has reasonable small objective value.

Of course, also random (or alternatively selected) samples of a few low dimensional principal sub-matrices can be efficiently tested via solving low dimensional versions of (BP). For the same reasons as above, we do not consider these random tests any further, but mention it as an idea for further research.

Proposition 2.6 (Necessary and sufficient conditions)
Let $B \in \mathbb{R}^{d \times d}$. Then:

1. B is Bernoulli-compatible $\iff$ each principal sub-matrix of B is Bernoulli-compatible.
2. B is Bernoulli-compatible $\iff \forall \lambda \in [0, 1] : \lambda B$ is Bernoulli-compatible.

Proof
As the "$\Leftarrow$" parts of both statements are obvious, we only consider the "$\Rightarrow$" directions. The first statement follows from the fact that any sub-vector of X is again multivariate Bernoulli distributed, if X is multivariate Bernoulli distributed. The second statement follows directly from the convex hull representation of a Bernoulli-compatible matrix. □

The first condition of the subsequent proposition is again easy-to-verify. To the best of our knowledge, the second property cannot be exploited for numerical purposes.

Proposition 2.7 (Sufficient conditions)
Let $B, G, H \in \mathbb{R}^{d \times d}$ and let B satisfy all necessary conditions from Proposition 2.1. Then it holds:

1. If B is diagonally dominant, then $\frac{1}{\text{trace}(B)} B$ is Bernoulli-compatible.
2. If G and H are Bernoulli-compatible, then $G \circ H$ (i.e. the Hadamard product of G and H) is Bernoulli-compatible.

Proof
The first statement follows from Lemma 2.15 and [Embrechts et al., 2016], Proposition 2.5. For a proof of the second statement, let us refer to [Embrechts et al., 2016], Proposition 2.1 or [Fiebig et al., 2017], Corollary 14. □

2.2 Connection to BQPs and complexity of (PB) and (DB)

From [Pitowsky, 1991], Section 3.1 and 3.3, we know that "$B \in \mathcal{B}_d$?" is both in **NP** and it is NP-hard, thus NP-complete. However, it is not clear if the complementary question "$B \notin \mathcal{B}_d$?" is also in **NP**, i.e. it is not known if "$B \in \mathcal{B}_d$?" is in **co-NP**. For more details on the complexity discussion, let us refer to [Pitowsky, 1991] or [Deza and Laurent, 1997], who link this problem to the question "**NP** = **co-NP**?". Nevertheless, to foster a better understanding of the inherent complexity of the membership testing problem, let us consider the following observation, which can be traced back to [Deza and Laurent, 1997], but does not seem to be widely known:

$$\begin{aligned}\min_{\mathbf{p}\in\{0,1\}^d} \mathbf{p}^\top Q\mathbf{p} &= \min_{\lambda\in\Lambda_{2^d}} \sum_i \lambda_i \mathbf{p}(i)^\top Q\mathbf{p}(i) = \\ &\min_{\lambda\in\Lambda_{2^d}} \sum_i \lambda_i \langle Q, \mathbf{p}(i)\mathbf{p}(i)^\top\rangle = \\ &\min_{\lambda\in\Lambda_{2^d}} \sum_i \lambda_i \langle Q, \mathbf{B}_i\rangle = \\ &\min_{\lambda\in\Lambda_{2^d}} \langle Q, \sum_i \lambda_i \mathbf{B}_i\rangle = \min_{B\in\mathcal{B}_d} \langle Q, B\rangle.\end{aligned}$$

This shows that any[7] BQP can be formulated as a linear program over the Bernoulli polytope. Hence, if the membership problem (together with provision of a separating hyperplane) was in **P**, a polynomial time algorithm for the solution of BQPs could be found.

In our numerical approach we exploit the above connection between the Bernoulli polytope and BQPs in the reverse direction: we base our ansatz for membership testing for the Bernoulli polytope on the solution of a sequence of corresponding BQPs. To the best of our knowledge, this is the first attempt to use this reverse connection to significantly speed up the testing of Bernoulli compatibility.

2.3 Connection to $\{0,1\}$-completely positive matrices

Berman and Xu introduced in [Berman and Xu, 2005] and [Berman and Xu, 2007] the concept of $\{0,1\}$-completely positive matrices, together with a detailed investigation of the structure of special types of these matrices. In our notation, a matrix A is called a $\{0,1\}$-completely positive matrix if it can be written as $A = \sum_{i=1}^{2^d-1} \nu_i \mathbf{B}_i$ with[8] $\nu_i \in \mathbb{N}_0$. In this case, $\sum_{i=1}^{2^d-1} \nu_i$ is called the $\{0,1\}$-completely positive rank of A, in short $\text{rank}_{\{0,1\}}(A)$.

The following result establishes an immediate connection between these $\{0,1\}$-completely positive matrices and Bernoulli matrices. Based on this connection, a novel necessary condition for $\{0,1\}$-completely positive matrices is derived in the subsequent corollary, which is based on testing a scaled version of the matrix on Bernoulli compatibility.

[7] For completeness, note that

$$\min_{\mathbf{p}\in\{0,1\}^d} \mathbf{p}^\top Q\mathbf{p} + q^\top \mathbf{p} = \min_{\mathbf{p}\in\{0,1\}^d} \mathbf{p}^\top (Q + \text{diag}(q))\, \mathbf{p},$$

showing that *all* binary QPs are covered by the above.

[8] Note that the sum index starts at 1 instead of 0 on purpose.

Theorem 2.8 (Scaling of Bernoulli-compatible matrices)
Let $B \in \mathbb{S}_d^+$ satisfy the necessary conditions from Proposition 2.1 and let $\text{trace}(B) > 0$. Then it holds:

1. B Bernoulli-compatible $\Rightarrow \frac{1}{\text{trace}(B)}B$ Bernoulli-compatible
2. $\frac{1}{\text{trace}(B)}B$ Bernoulli-compatible and $\text{trace}(B) \leq 1 \Rightarrow B$ Bernoulli-compatible

From this theorem, the subsequent corollary, which yields a necessary condition for $\{0,1\}$-completely positive matrices, follows immediately.

Corollary 2.9 (Necessary condition for $\{0,1\}$-completely positive matrices)
Let $0 \neq A \in \mathbb{S}_d$ be a $\{0,1\}$-completely positive matrix. Then $B = \frac{1}{\text{trace}(A)}A$ is Bernoulli-compatible.

For the proof of the theorem, we need the following lemma which characterizes the trace of Bernoulli-compatible matrices.

Lemma 2.10 (Bounding a_0)
Let $B \in \mathbb{S}_d$ be Bernoulli-compatible. Then, for each $a \in \Lambda_{2^d}$ with

$$B = \sum_{i=0}^{2^d-1} a_i \mathbf{B}_i$$

it holds that

$$\sum_{i=1}^{2^d-1} a_i \leq \text{trace}(B) \leq d \sum_{i=1}^{2^d-1} a_i,$$

or, equivalently

$$1 - a_0 \leq \text{trace}(B) \leq d(1 - a_0).$$

Proof (of Lemma 2.10)
As $B_0 = 0$, we can leave out the first term of the representation of B and obtain

$$B = \sum_{i=1}^{2^d-1} a_i \mathbf{B}_i.$$

Now, as

$$1 \leq \text{trace}(\mathbf{B}_i) \leq d, \quad \text{for } i = 1, \dots, 2^d - 1$$

we have

$$\sum_{i=1}^{2^d-1} a_i \leq \text{trace}(B) \leq d \sum_{i=1}^{2^d-1} a_i$$

which proves the statement. □

Proof (of Theorem 2.8)

1. Note that

$$\frac{1}{\text{trace}(B)}B = \sum_{i=0}^{2^d-1} \frac{a_i}{\text{trace}(B)}\mathbf{B}_i = \sum_{i=1}^{2^d-1} \frac{a_i}{\text{trace}(B)}\mathbf{B}_i.$$

   From the above lemma, we have

$$\sum_{i=1}^{2^d-1} \frac{a_i}{\text{trace}(B)} \leq 1.$$

   Hence, the above representation yields a proper representation as a Bernoulli-compatible matrix, establishing the claim.

2. As $B = \text{trace}(B)\big(\frac{1}{\text{trace}(B)}B\big)$ and the matrix in brackets is Bernoulli-compatible according to the assumption, the claim directly follows from the second part of Proposition 2.6. □

Proof (of Corollary 2.9)
As A is a $\{0,1\}$-completely positive matrix, it has a representation

$$A = \sum_{i=1}^{2^d-1} \nu_i \mathbf{B}_i \quad \text{with} \quad \nu^* := \sum_{i=1}^{2^d-1} \nu_i > 0.$$

Thus, $\frac{1}{\nu^*}B$ is obviously a Bernoulli-compatible matrix. The rest of the statement follows from the first statement in Theorem 2.8. □

Unfortunately, there does not seem to be any 1-to-1 correspondence between Bernoulli-compatible matrices and $\{0,1\}$-completely positive matrices in the gist of the previous corollary. Nevertheless, the following (slightly weaker) result can be stated.

Proposition 2.11
Let B be Bernoulli-compatible and let B have only rational entries. Then there exists a $\kappa \in \mathbb{N}$ such that κB is a $\{0,1\}$-completely positive matrix.

Proof
As it is well-known that the Simplex algorithm yields a rational solution if started with rational inputs, there is at least one representation $a \in \Lambda_{2^d}$ with rational entries only. Then, setting κ to the main denominator of all a_i yields the required κ. □

Taking into account the fact that the optimal solution in an LP is determined by means of an optimal basis, the main denominator κ can be bounded by the Hadamard bound for the determinant of $\{0,1\}$-matrices (and the main denominator of B), see for instance [Ziegler, 1999], Lemma 24.

2.4 Relation of the Bernoulli cone to standard cones

Motivated by the previous considerations, let us introduce the following convex cones.

Definition 2.12 (Dual cone to the Bernoulli cone)
The cone $BC_d^+ := \mathrm{cone}(\mathcal{B}_d)$ shall be denoted as *Bernoulli cone*. Let us call its dual cone $WP_d^+ := (BC_d^+)'$ the *weakly positive cone*.

For example, all matrices $qq^\top + \mathrm{diag}(q)$ with $q \geq 0$ are weakly positive matrices. Similarly, the matrices $qq^\top - \mathrm{diag}(q)$ are weakly positive, if $q \leq -\mathbb{1}$ or if $0 \leq q \leq \frac{1}{d}\mathbb{1}$. Finally, the matrices $qq^\top \pm \mathrm{diag}(q)$ are weakly positive for all $q \in \mathbb{Z}^d$; they constitute exactly the matrices defining the hypermetric inequalities.

Before we give a complete characterization of the weakly positive cone and discuss its relation to other cones, let us state the following obvious observation. For this purpose, let

$$\begin{aligned}
Cop_d^+ &= \left\{ A \in \mathbb{R}^{d\times d} |\ x^\top A x \geq 0,\ \forall x \in \mathbb{R}_+^d \right\}, \\
CP_d^+ &= \left\{ A \in \mathbb{R}^{d\times d} |\ \exists B \geq 0 : A = BB^\top \right\}, \\
DNN_d^+ &= \left\{ A \in \mathbb{R}^{d\times d} |\ A \geq 0, A \succeq 0 \right\}
\end{aligned}$$

denote the usual copositive cone, the completely positive cone, and the doubly non-negative cone. The subsequent observation is not only true in our specific setup, but carries over to all kind of polytopes generated by expectations like $\mathbb{E}_{\mathbb{P}}[YY^\top]$ for non-negative random vectors Y with given marginals.

Proposition 2.13
It holds that $Cop_d^+ \subset WP_d^+$ and furthermore $BC_d^+ \subset CP_d^+$.

Proof
Note that for any copositive matrix C and $X \geq 0$ it holds

$$\langle C, \mathbb{E}_{\mathbb{P}}[XX^\top]\rangle = \mathbb{E}_{\mathbb{P}}[\langle C, XX^\top\rangle] = \mathbb{E}_{\mathbb{P}}[X^\top C X] \geq 0.$$

From this identity, both statements are immediate. □

In our specific setup, it is possible to give a complete characterization of the dual cone.

Proposition 2.14 (Characterization of weakly positive matrices)
It holds

$$WP_d^+ = \{A \in \mathbb{S}_d \,|\, 0 \leq \min_{x\in\{0,1\}^d} x^\top A x\},$$

i.e. the weakly positive cone consists of all matrices which yield non-negative optimal value in the standard BQP.

Proof
Using the above observations on the relation between linear optimization over the Bernoulli polytope and binary quadratic programming, we get

$$\begin{aligned} WP_d^+ &= (BC_d^+)' \\ &= \{A \in \mathbb{S}_d \,|\, 0 \le \langle A, B\rangle \;\forall B \in BC_d^+\,\} \\ &= \{A \in \mathbb{S}_d \,|\, 0 \le \min_{B \in BC_d^+} \langle A, B\rangle\,\} \\ &= \{A \in \mathbb{S}_d \,|\, 0 \le \min_{\lambda \ge 0, B \in \mathcal{B}_d} \lambda\langle A, B\rangle\} \\ &= \{A \in \mathbb{S}_d \,|\, 0 \le \min_{\lambda \ge 0, x \in \{0,1\}^d} \lambda x^\top A x\} \\ &= \{A \in \mathbb{S}_d \,|\, 0 \le \min_{x \in \{0,1\}^d} x^\top A x\}. \end{aligned}$$

□

The following lemma yields that membership testing for the Bernoulli cone is equivalent to membership testing for the Bernoulli polytope. Please note that this lemma was already used in the proof of Proposition 2.7 to derive a sufficient condition (i.e. diagonal dominance) for Bernoulli matrices.

Lemma 2.15 (Complexity of membership testing for the Bernoulli cone)
Let $A \in \mathbb{S}_d$ with $\mathrm{trace}(A) > 0$. Then

$$A \in BC_d^+ \quad \Longleftrightarrow \quad \frac{1}{\mathrm{trace}(A)} A \in \mathcal{B}_d.$$

Proof
The direction "$\Leftarrow$" is immediate from the definition of the cone BC_d^+. For the reverse direction, let $A \in BC_d^+$. Then there exists $\lambda > 0$ and $B \in \mathcal{B}_d$ such that $A = \lambda B$. As $B \in \mathcal{B}_d$ we also have $\frac{1}{\mathrm{trace}(B)} B \in \mathcal{B}_d$ due to Theorem 2.8.(1), from which the statement follows. □

Remark 2.16
Due to Lemma 2.15, testing membership in the Bernoulli cone, i.e. "$B \in BC_d^+$?", remains an NP-complete problem, as well as testing "$B \notin WP_d^+$?". Using the recent result from [Friedland and Lim, 2016], Theorem 15, we obtain that also membership testing for the weakly positive cone has to be NP-hard. In our specific situation, this directly follows from the representation derived in Proposition 2.14 due to the NP-hardness of (BQP). Please note, that the NP-completeness of the corresponding decision problem of (BQP) does not[9] imply that the membership problem of the weakly positive cone also is in **NP** (otherwise, it would be NP-complete).

[9] The decision problem for minimization problems is formulated in terms of "$\le$", which is the wrong direction for our purposes.

Summarizing the above observations, we obtain the following diagram of standard cones used in optimization theory.

$$\begin{array}{ccccccccccc} BC_d^+ & \subset & CP_d^+ & \subset & DNN_d^+ & \subset & \mathbb{S}_d^+ & \subset & Cop_d^+ & \subset & WP_d^+ \\ \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow \\ WP_d^+ & \supset & Cop_d^+ & \supset & \mathbb{S}_d^+ \oplus \mathbb{R}_+^{d\times d} & \supset & \mathbb{S}_d^+ & \supset & CP_d^+ & \supset & BC_d^+ \end{array}$$

where a cone is marked in red, when its membership problem is known to be NP-hard. If the membership problem is known to be in **P**, then it is marked in green. The arrow between cones denotes the dual relationship to each other.

3 A column generation approach

3.1 Motivation

Since LPs are often considered to be the most easy-to-solve optimization problems, it is tempting to solve problems (PB) and (DB) by standard LP-solvers. Accordingly, we solved the different formulations (PB) and (DB) with IBM's ILOG CPLEX. The results are presented in Figure 1, where computation time and typical[10] memory usage are illustrated. For small dimensions, i.e. $d \leq 17$, we were able to solve all instances within a second. However, due to memory limitations, we were not able to solve any problem instance for $d > 20$. As already mentioned in the introductory section, to overcome these difficulties, we subsequently propose a column generation approach.

3.2 A generic column generation method

In the following, let us recall the generic column generation method for linear programs; see, for example, [Lübbecke, 2010] for a detailed presentation. For this purpose, let us consider a linear optimization problem (P_J) in the following form, called the *master problem*:

$$\begin{aligned} v(J) := \min_{x\in\mathbb{R}^n} \quad & \sum_{j\in J} c_j x_j \\ \text{s.t.} \quad & \sum_{j\in J} \mathbf{d}_j x_j \leq \mathbf{b} \\ & x_j \geq 0 \qquad \forall\, j \in J \end{aligned} \tag{P_J}$$

[10] It is very difficult to exactly measure the average memory usage of an algorithm at any given time, thus we present approximated (slightly overestimated for small dimensions) values at this point.

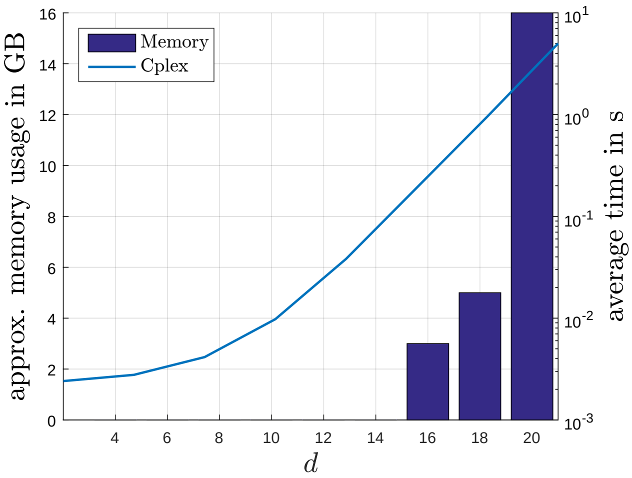


Figure 1 Typical memory usage and computation times of CPLEX, averaged over all problem classes. For $d = 20$ the maximum available memory of 16 GB is reached.

with $J = \{1, \ldots, n\}$, $\mathbf{d}_j \in \mathbb{R}^m$ for $j \in J$ and $\mathbf{b} \in \mathbb{R}^m$, where n is much larger than m. The corresponding dual[11] problem (D_J) is given by

$$\begin{array}{rlrcll} v(J) = & \max\limits_{y \in \mathbb{R}^m} & \mathbf{b}^\top \mathbf{y} & & & \\ & \text{s.t.} & \mathbf{d}_j^\top \mathbf{y} & \leq & c_j & \forall\, j \in J \\ & & \mathbf{y} & \leq & 0. & \end{array} \tag{D_J}$$

As Figure 1 shows, due to the large number of primal variables, directly solving the master problem becomes intractable beyond $n \approx 10^6$ due to memory issues. Therefore, one resorts to iteratively solving *restricted master problems* (P_{I_k}) for $k = 1, \ldots, K$:

$$\begin{array}{rlrcll} v(I_k) := & \min\limits_{x \in \mathbb{R}^n} & \sum\limits_{j \in I_k} c_j x_j & & & \\ & \text{s.t.} & \sum\limits_{j \in I_k} \mathbf{d}_j x_j & \leq & \mathbf{b} & \\ & & x_j & \geq & 0 & \forall\, j \in I_k. \end{array} \tag{P_{I_k}}$$

The sets $I_k \subset J$, usually called *inner sets*, represent subsets of indices (i.e. primal variables) which are used for the optimization – the remaining variables are simply set to 0 and thus excluded from the optimization. Starting with some initial inner set I_0, subsequently variables (columns) which are assumed to improve the current optimal solution are added (generated), when advancing from I_k to I_{k+1}. Thus, in the course of

[11] In this section, we assume that the primal master problem is feasible and bounded. Hence, by strong duality, the same holds for the dual problem, and both optimal values coincide.

the algorithm, a finite sequence of (small) subsets $I_0 \subsetneq I_1 \subsetneq \ldots \subsetneq I_K \subseteq J$ is considered. Due to the fundamental theorem of linear programming, there always exists an optimal basic solution for P_J and, by construction, an optimal solution for the master problem is obtained in the restricted problem as soon as an optimal base for the master problem is included in a set I_k. The number of elements in such a base is m, which is assumed to be much smaller than n. Hence, in practice, I_K will hopefully be substantially smaller than J for the majority of problem instances. As Figure 2 shows, this is not in vain, as

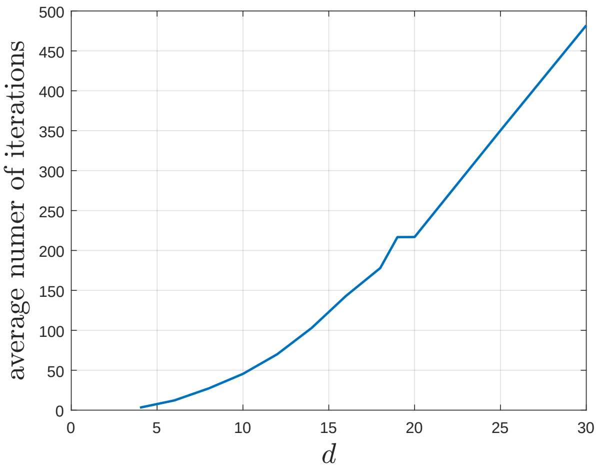


Figure 2 Average number of iterations of the pure column generation method, averaged over 10 instances of each problem classes, cf. Section 4. It can be observed that the number of column generation steps only increases mildly with the dimension d.

the average number of column generation steps grows only mildly with the dimension d and does not exceed 500 in all our examples for $d \leq 30$.

Denote by $\mathbf{x}^*_{I_k}$ an optimal solution of (P_{I_k}) and by $\mathbf{y}^*_{I_k}$ the corresponding optimal dual solution. Since $\mathbf{x}^*_{I_k}$ is feasible for (P_J), $\mathbf{x}^*_{I_k}$ is an optimal solution for the master problem if and only if $\mathbf{y}^*_{I_k}$ is feasible for (D_J).

The dual feasibilty of $\mathbf{y}^*_{I_k}$ can be determined by means of the *subproblem* (SP_k):

$$h^*_{I_k} := \max_{j \in J} \; h_j(\mathbf{y}^*_{I_k}), \qquad (SP_k)$$

where for some (not necessarily feasible) point $\mathbf{y}$, the violation of the j-th dual constraint is given by

$$h_j(\mathbf{y}) := \mathbf{y}^\top \mathbf{d}_j - c_j.$$

By construction, $h^*_{I_k} \leq 0$ implies $\mathbf{y}^*_{I_k}$ to be dual feasible and thus provides optimality of the current solution $\mathbf{x}^*_{I_k}$. In the case of $h^*_{I_k} > 0$ one sets $I_{k+1} := I_k \cup \{j^*_{I_k}\}$, that is, one adds the corresponding maximizing column $j^*_{I_k}$ in (SP_k) and sets $k := k+1$. Repeating this process, an optimal solution for the master problem is found after a finite number of steps. Altogether, we obtain

Algorithm 1 (Column Generation)

1. Choose an initial subset $I_0 \subset J$ such that the restricted master problem is feasible and bounded and set $k := 0$.
2. Solve the restricted master problem (P_{I_k}) to obtain $\mathbf{x}^*_{I_k}$ together with dual multipliers $\mathbf{y}^*_{I_k}$.
3. If $v(I_k) = 0$: $\mathbf{x}^*_{I_k}$ solves (P_J), stop.
4. Solve the subproblem (SP_k) to obtain $h^*_{I_k}$ and the corresponding maximizer $j^*_{I_k}$.
5. If $h^*_{I_k} \leq 0$: $\mathbf{x}^*_{I_k}$ solves (P_J), stop.
   Else, set $I_{k+1} := I_k \cup \{j^*_{I_k}\}$, $k := k+1$ and go to 2.

Remark 3.1
Let us emphasize a few important aspects of Algorithm 1:

(i) For finite J, if (P_{I_0}) is feasible and bounded, the algorithm inherits the finiteness and correctness properties of linear programming, cf. [Lübbecke, 2010]. This means, for some $K \leq n$, the iterate $x^*_{I_K}$ has to be an optimal solution for (P_J). Note that in the extreme case this may lead to $I_K = J$. In practice, as can be depicted from Figure 2, the average number of iterations only increases rather mildly with the dimension d.

(ii) By construction, any two resticted optimal solutions $\mathbf{x}^*_{I_k}$ and $\mathbf{x}^*_{I_l}$ with $k < l$ satisfiy $\mathbf{c}^T\mathbf{x}^*_{I_k} \geq \mathbf{c}^T\mathbf{x}^*_{I_l}$, i.e. the optimal values of the restricted problems converge from above to the optimal value of the master problem.

(iii) The stopping criterion in Step 3 of Algorithm 1 is not part of a classical column generation method. However, as we know that the optimal value of the primal problem is always non-negative, we can stop the column generation as soon as an objective value of 0 is obtained. In this case the given matrix B is Bernoulli-compatible. In case Algorithm 1 stops in Step 5, the matrix B is not Bernoulli-compatible.

(iv) It is critical to the overall efficiency of the column generation approach that both the restricted LP as well as the subproblem of determining the most violating constraint can be solved efficiently. In most successful applications of column generation, the problem structure of the subproblem can be exploited to avoid solving by full enumeration. For more details on the efficient solution of the subproblem in the present context let us refer to Section 3.3.

3.3 Efficient solution of (SP_k)

As mentioned above, an efficient implementation of the column generation is obtained if subproblem (SP_k) is solved efficiently. To avoid the full enumeration of all constraints, let us now exploit the specific structure of the subproblem. Given the dual variable $\mathbf{y} = (Y, Z, \gamma)$, the maximum dual violation can be computed as follows:

$$\begin{aligned}\max_{j\in J}\ h_j(\mathbf{y})\ &= \max_{j\in J}\ \gamma - \langle \mathbf{B}_j, Y - Z\rangle = \\ &\gamma + \max_{j\in J}\ \langle \mathbf{B}_j, Z - Y\rangle = \\ &\gamma + \max_{j\in J}\ \langle \mathbf{p}(j)\mathbf{p}(j)^\top, Z - Y\rangle = \gamma + \max_{\mathbf{p}\in\{0,1\}^d}\ \mathbf{p}^\top(Z-Y)\mathbf{p}\end{aligned}$$

From the optimal $\mathbf{p}^*$, the corresponding index can be determined immediately. Therefore, finding the most violating constraint and computing the maximum violation boils down to solving the binary quadratic program

$$\max_{\mathbf{p}\in\{0,1\}^d}\ \mathbf{p}^\top G\mathbf{p}, \qquad \text{(SP-BQP)}$$

with $G = Z - Y$. For this problem, it is well-known that it is NP-hard, as long as no special structure in G can be assumed, see, e.g., [Padberg, 1989], which is the case in the present situation. Until today, exact solution methods seem to be limited to a few hundred variables at most, see, for instance, [Kochenberger et al., 2014]. In our implementation, we have solved (SP-BQP) by CPLEX, which has shown to be much more efficient than full enumeration, cf. Figure 3. Furthermore, one observes that the time spent for solving the restricted master problem roughly equals the time needed for finding the most violating constraint. This indicates that Algorithm 1 can only be improved in terms of computation times, if both the LPs and the binary quadratic subproblems can be solved much faster.

3.4 Dual bound

As mentioned in Remark 1.7, Algorithm 1 can be terminated early in case a separating hyperplane is found. By linear duality we know that any feasible solution to (D_J) provides a lower bound to the optimal value of (P_J). Unfortunately, the dual solution $\mathbf{y}^*_{I_k}$ provided at step k is — in general, and excluding the optimal case — infeasible for (D_J). Therefore, we consider an additional dual bound based on a Slater point of the dual problem in the gist of [Daum and Werner, 2011].

Proposition 3.2
Let $\mathbf{y}$ be infeasible for (D_J). Further, let $\mathbf{y}_s$ be a Slater point for (D_J), i.e. $h^*(\mathbf{y}_s) < 0$. Then there exists some $\bar{\mu} \in]0,1[$ such that $\bar{\mathbf{y}} = \mu\mathbf{y} + (1-\mu)\mathbf{y}_s$ is a Slater point, i.e. $h^*(\bar{\mathbf{y}}) < 0$, for all $0 \le \mu < \bar{\mu}$. A suitable choice for $\bar{\mu}$ is given by

$$\bar{\mu} = \frac{-h^*(\mathbf{y}_s)}{h^*(\mathbf{y}) - h^*(\mathbf{y}_s)}.$$

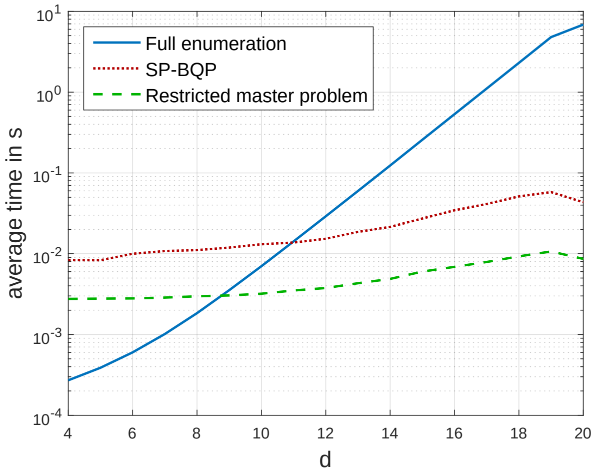


Figure 3 Comparison of average computation times for one full enumeration, the solution of one binary quadratic subproblem, and the solution of one restricted master problem. The average is taken over all problem instances. It can be observed that full enumeration is much slower than solving the BQP subproblem. Further, the restricted master problem and the BQP subproblem roughly have the same computational workload.

This implies that we can shift any infeasible iterate $\mathbf{y}^*_{I_k}$ along the line towards the Slater point $\mathbf{y}_s$ to a feasible iterate $\bar{\mathbf{y}}_{I_k}$. Whenever $\bar{\mathbf{y}}_{I_k}$ has a dual function value strictly greater than zero, it constitutes a separating hyperplane in the sense of Remark 1.7. As we will

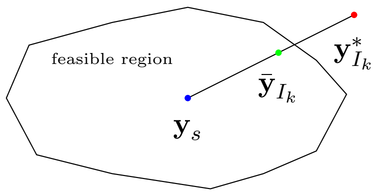


Figure 4 By means of a Slater point $\mathbf{y}_s$, shrink iterate $\mathbf{y}^*_{I_k}$ to a dual feasible iterate $\bar{\mathbf{y}}_{I_k}$ to obtain a lower bound for the optimal objective function value.

see in the following, this additional dual bound allows for a much earlier termination of the column generation and thus decreases computation time for Bernoulli-incompatible matrices significantly.

4 Numerical analysis

In this section, we report in detail the setup of our case study based on selected test instances, before we discuss the main numerical findings. In summary, our numerical analysis shows that the pure column generation method is quite efficient up to $d = 30$. Further, making use of several heuristics, we can efficiently test for Bernoulli compatibility up to dimension $d = 40$.

4.1 Test problems

Unfortunately, for testing Bernoulli-compatibility, there is no common test library available. Therefore, for the numerical tests, we have come up with five different families of test problems. The first two represent specifically selected parametrized problem classes, whereas the last three are based on random combinations of vertices of the Bernoulli polytope $\mathcal{B}_d$. All test cases satisfy the necessary conditions from Proposition 2.1, besides a few exceptions for $d < 6$, as well as a significant number of instances in class 1 which violate the Fréchet–Hoeffding bounds, cf. Figures 5 and 9.

Problem class 1: The matrices B of the first class are given by

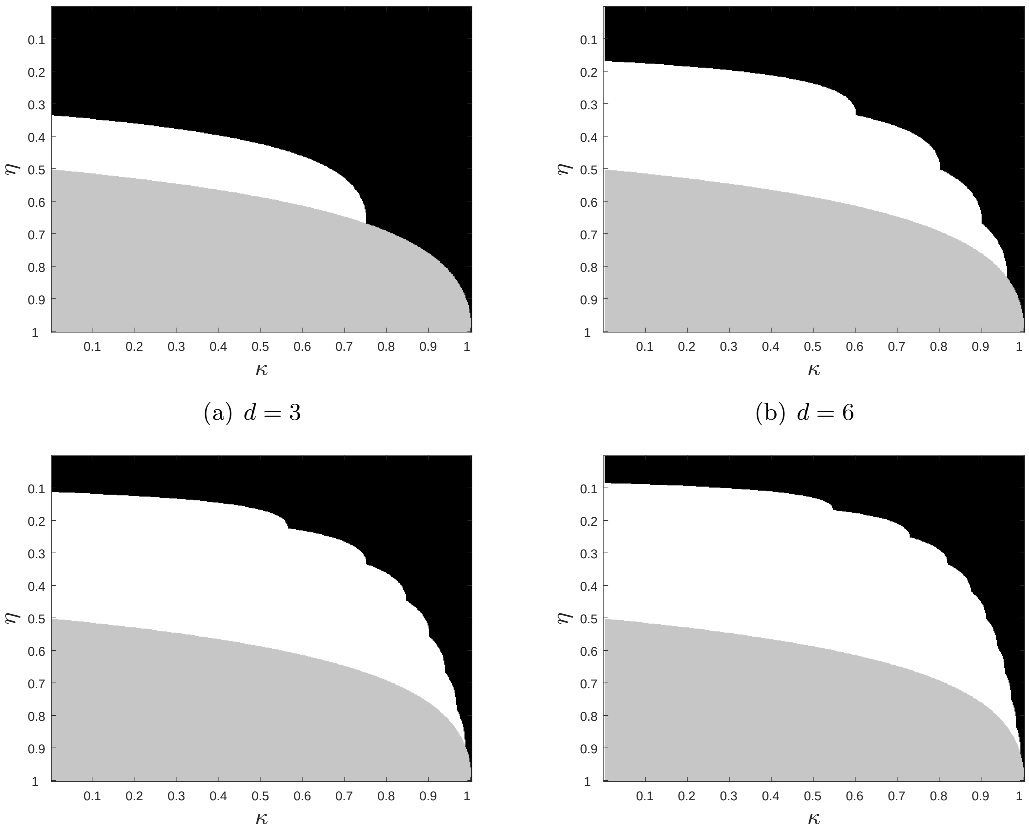
$$B := (\eta - \eta^2\kappa)I + \kappa\eta^2 E,$$

for some $0 \leq \eta \leq 1$ and $0 \leq \kappa \leq 1$, where I denotes the identity matrix. Instances of this problem class can be either Bernoulli-compatible or not, depending on the parameters, cf. Figure 5.

Problem class 2: Matrices B of the second class are given by

$$B_{ii} = \frac{p}{p+q}, \qquad i = 1, \ldots, d,$$
$$B_{ij} = \frac{p}{p+q}\frac{p+1}{p+q+1}, \qquad 1 \leq i \neq j \leq d,$$

for some $0 < p < 1$ and $0 < q < 1$. Instances of this problem class are always Bernoulli-compatible: Draw $(U_1, \ldots, U_d)$ from a copula that is defined as the convex combination of $1/(p+q+1)$ times the comonotonicity copula and $(p+q)/(p+q+1)$ times the independence copula. Further, let $X_i := \mathbf{1}_{U_i \leq p/(p+q)}$ for $i = 1, \ldots, d$. It is then easily verified (by conditioning) that $\mathbb{E}[X_i X_j] = \frac{p(1+p)}{(1+p+q)(p+q)}$ for $i \neq j$. Moreover, $\mathbb{E}[X_i^2] = \frac{p}{p+q}$ by the uniform margins property of a copula.



(a) $d = 3$ (b) $d = 6$

(c) $d = 9$ (d) $d = 12$

Figure 5 Bernoulli-compatibility of matrices in problem class 1, depending on η and κ for $d = 3, 6, 9, 12$. Black areas indicate Bernoulli-compatible matrices; gray areas indicate Bernoulli-incompatible instances that violate the Fréchet–Hoeffding bounds.

Problem class 3: The third class constitutes randomly generated matrices

$$B := \sum_{k=1}^{n} \lambda_{i_k} \cdot \mathbf{B}_{i_k}.$$

The number of terms n is uniformly distributed in the interval $[d^2, d^4]$ and the vertices B_{i_k} are uniformly distributed over all vertices of $\mathcal{B}_d$. Finally, the non-zero coefficients $\lambda_{i_1}, \ldots, \lambda_{i_n}$ of the convex combination are uniformly distributed on the standard (n)-simplex, i.e. sampled from a Dirichlet distribution. As a convex combination of extremal points of $\mathcal{B}_d$, B is always Bernoulli-compatible.

Problem class 4: Based on class 3, the matrices B of problem class 4 are given by

$$B := A + \frac{1}{10}\mathbf{B}_j,$$

where the matrix A is generated as in problem class 3. One specific index j with $\lambda_j > 0$ is randomly chosen and increased by 0.1. In practice, this usually leads to Bernoulli-incompatible matrices for $d > 14$.

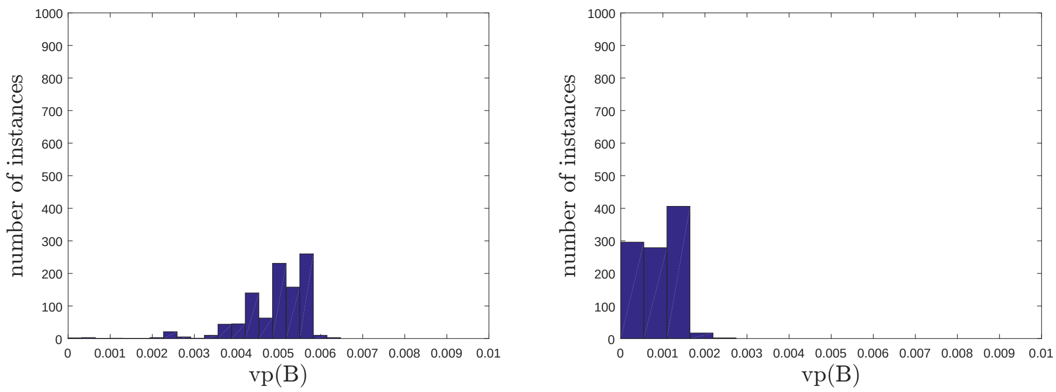
Problem class 5: Finally, we also consider a problem class which is supposed to produce "hard" problem instances, by setting

$$B := A + \frac{1}{d}\mathbf{B}_j.$$

Now, the matrix B is derived in a similar fashion as in class 4, however, the shift decreases with increasing dimension. This is supposed to produce both Bernoulli-compatible and Bernoulli-incompatible matrices which are close to the Bernoulli polytope's boundary[12].

In Figure 6, we have illustrated the distance of randomly generated instances from problem classes 4 and 5.

[12]This assumption is supported by our numerical findings in this section.



(a) Problem class 4 (b) Problem class 5

Figure 6 Distribution of the distance $vp(B)$ to the Bernoulli polytope for matrices B from problem classes 4 and 5 in the case of $d = 14$. Instances of class 4 are clustered around $5 \cdot 10^{-3}$ which is significantly larger than our threshold for Bernoulli-compatible matrices of 10^{-6}, see also Appendix A. For class 5, $vp(B)$ is significantly lower for most instances, whereof approximately ten percent satisfy $vp(B) < 10^{-6}$.

It can be observed that class 4 usually only contains Bernoulli-incompatible matrices, whereas the situation for class 5 is mixed, with on average much smaller distance than for class 4.

4.2 Results

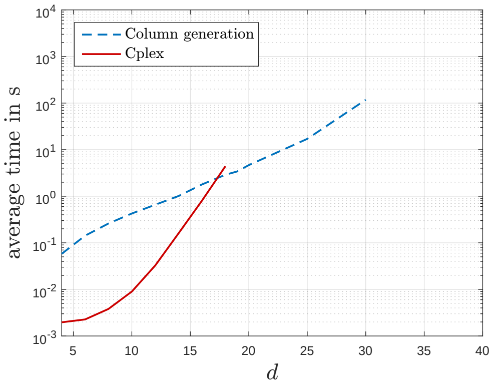
In this section, we first compare the performance of the pure column generation method to standard LP solvers, before we also analyze the added value of additional primal and dual heuristics. As mentioned before, any problem formulation of (PB) and (DB) can be solved directly by an arbitrary LP solver. Therefore, we tested different combinations of problem formulations and LP solvers of Cplex. As expected, we found, that — if one is just interested in any solution — the most efficient approach was to solve the feasibility problem of (PB) with the primal simplex of Cplex. None of the direct solvers could solve the problem for $d > 20$ due to memory issues. Therefore, direct methods are omitted for $d > 20$ in the following. For further implementation details, let us refer to Appendix A.

Let us start with the results of the column generation approach in its pure form. For this purpose, we refrain from checking any necessary or sufficient conditions presented in Section 2. We also exclude any sophisticated primal or dual heuristics. A simple initial inner set I_0 of size $\mathbf{O}(d^2)$ for Algorithm 1 is constructed by considering only those $\mathbf{p}$, where all bits equal to 1 appear in one run. The performance of the pure column generation method is illustrated in Figure 7. We can observe that the pure column generation method is able to solve instances up to $d = 30$ whereas all direct approaches fail to solve instances where $d > 20$.

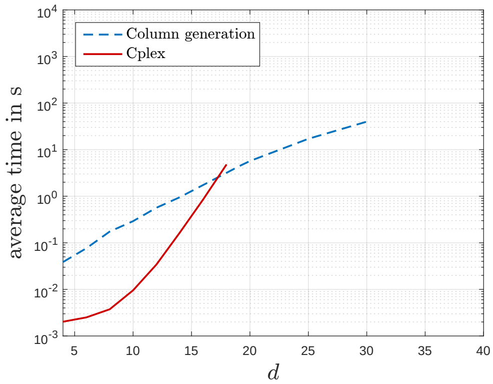
The computation times can be further significantly reduced when primal and dual heuristics are considered. To this end, we start with a more sophisticated initial inner set I_0 based on Proposition 2.6. In addition, we use Proposition 2.5 to obtain a Slater point for (DB), which in turn is beneficial for determining separating hyperplanes for Bernoulli-incompatible matrices, cf. Section 3.4. Finally, we also benefit from testing the necessary conditions of Proposition 2.1 before running the column generation algorithm. We incorporated all these heuristics into the final implementation of the column generation method. The performance of this enhanced method is illustrated in Figure 8. Again, very similar results as for the pure column generation method can be observed. For $d \geq 15$, the enhanced column generation method starts to outperform the direct solvers, as for a significant number of instances, the solution of large LPs can be avoided. This is also the main reason why the enhanced method is clearly faster than the pure column generation method as shown in Figure 8. This is further illustrated in Figure 9 in terms of statistical information on how problem instances were solved. Figure 9 shows that most problem instances can be successfully solved by primal or dual heuristics as long as d is small to medium sized. In the case of large d and Bernoulli-compatible matrices all problem instances are ultimately solved by column generation, whereas most Bernoulli-incompatible matrices could be identified via dual approximation.

5 Conclusion and outlook

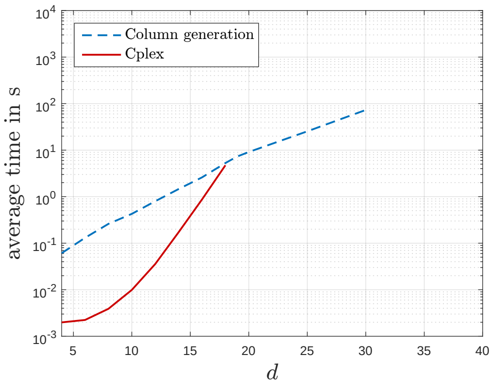
The main aim of this exposition was to come up with a computational approach to verify whether a matrix $B \in \mathbb{R}^{d \times d}$ is a *Bernoulli-compatible matrix* or not. Our main approach was based on linear programming, first considered by [Lee, 1993]. To deal with the problem of exponentially many variables in the primal LP, we naturally proposed to solve larger LPs with a column generation method. For an efficient implementation of the column generation method, it was crucial to replace the full enumeration for identifying the most violating constraint by something more efficient. Due to the specific structure of the problem, enumeration can be avoided, if instead a binary quadratic program is solved. Although the membership problem is known to be NP-complete, we observe very promising performance of such a pure column generation method up to dimension $d = 30$ on a variety of test problems. To improve the performance of the pure column generation method, we have enhanced the method by a novel dual bound for early termination, which has shown to be quite effective. In addition, primal and dual heuristics, which can be efficiently tested before applying column generation, further significantly increase the effectiveness of our approach. As these primal and dual heuristics rely on properties of Bernoulli-compatible matrices, we have provided a detailed overview of the main properties. Along doing so, we also derived a novel test for $\{0, 1\} - cp$ matrices.



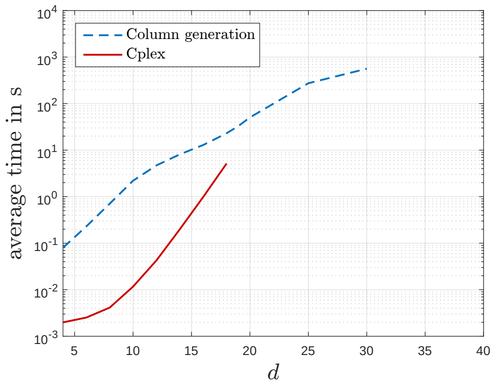
(a) Problem class 1



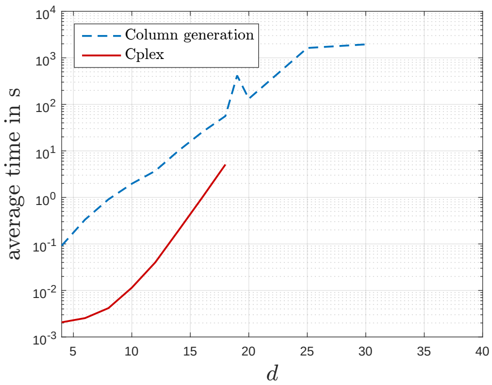
(b) Problem class 2



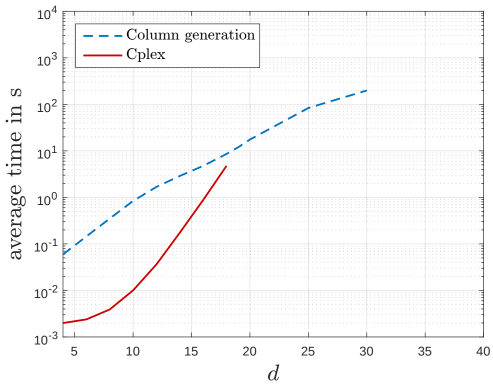
(c) Problem class 3



(d) Problem class 4

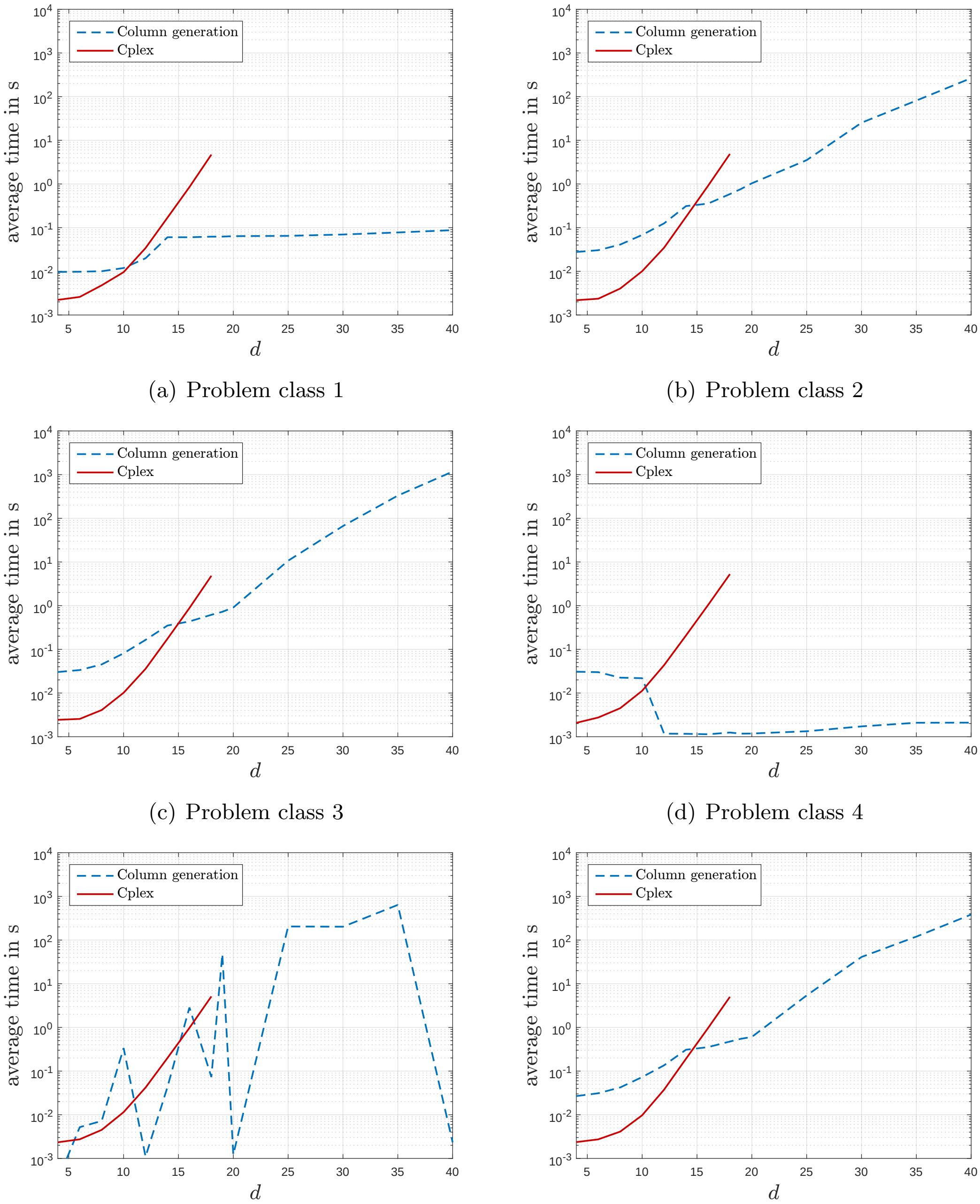


(e) Problem class 5



(f) Average for classes 1 to 4

Figure 7 In each subplot, the averaged computation times are illustrated for the pure column generation method. The average is taken over 9 instances for each d. The computation was aborted, whenever the time limit of 30 minutes was reached. This was the case for all instances of class 5 when $d \geq 25$. The comparison to CPLEX yields that the column generation method is able to solve much larger problems than CPLEX.

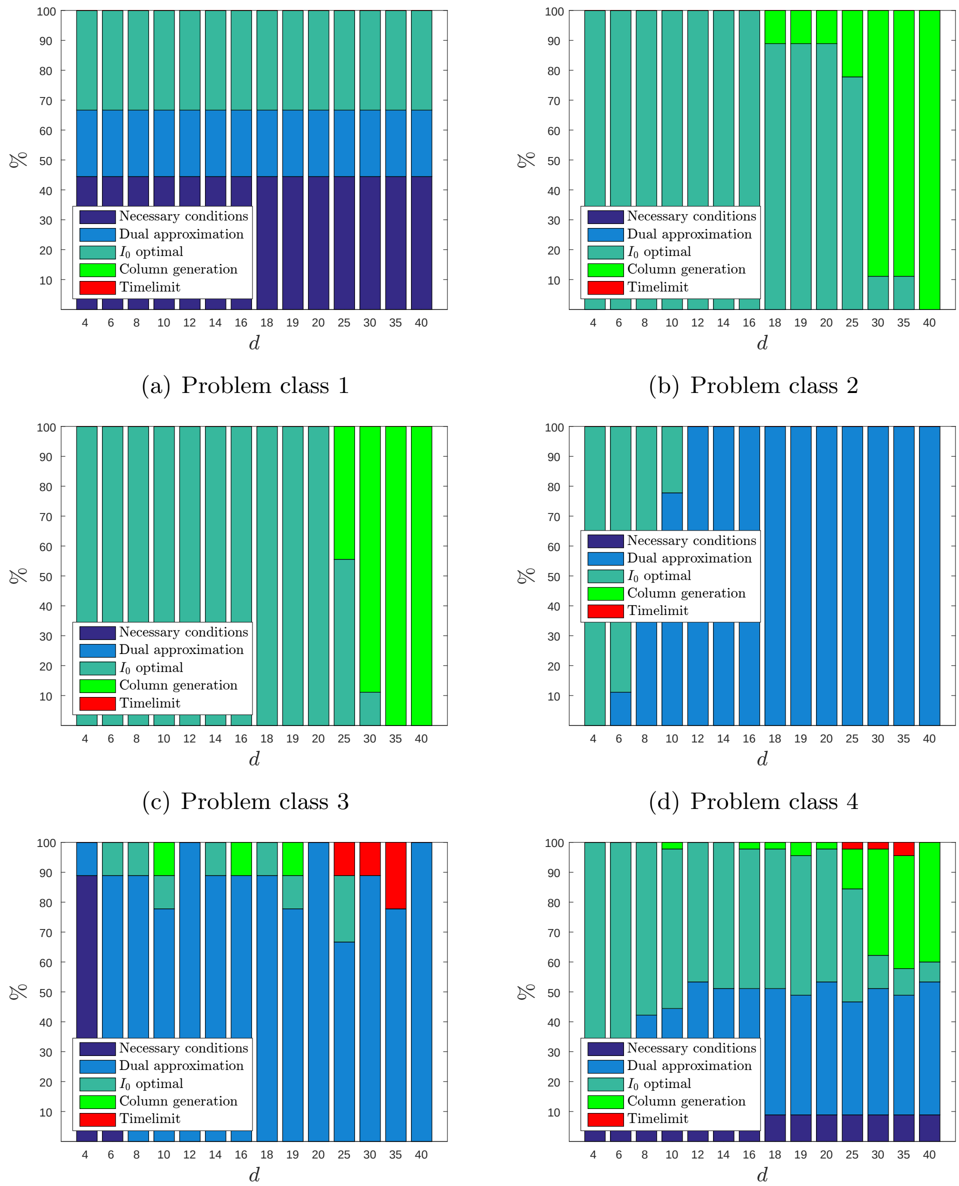


(a) Problem class 1 (b) Problem class 2

(c) Problem class 3 (d) Problem class 4

(e) Problem class 5 (f) Average for classes 1 to 4

Figure 8 In each subplot, the averaged computation times are illustrated for the enhanced column generation method. The average is taken over 16 instances for each d. The computation was aborted, whenever the time limit of 30 minutes was reached. The comparison to CPLEX yields that the enhanced column generation method is able to solve much larger problems than CPLEX. For problem class 4, we observe that for $d > 10$ only Bernoulli-incompatible instances are produced and all these instances can be solved via dual approximation (see Figure 9).



(a) Problem class 1

(b) Problem class 2

(c) Problem class 3

(d) Problem class 4

(e) Problem class 5

(f) Average

Figure 9 In each subplot, the average percentage shares of different solution types are illustrated for the enhanced column generation method. The average is taken over 16 instances for each d. "Necessary conditions" implies that one of the necessary conditions of Proposition 2.1 is not satisfied, "Dual approximation" implies that a separating hyperplane was found solving the dual relaxation presented in Proposition 2.5, "I_0 optimal" indicates that the inner set constructed by primal heuristics yields an optimal solution, "Column generation" implies that the solution was found after actually running column generation and finally "Time limit" implies that the time limit of 30 minutes for the column generation was reached.

So far, not all necessary conditions for Bernoulli-compatible matrices have been implemented. It would be interesting to see, if there are strategies how to incorporate for example hypermetric inequalities in the testing with reasonable effort. We also expect further improvement by fully incorporating triangle inequalities.

References

[Avis and Grishukhin, 1993] Avis, D. and Grishukhin, V. P. (1993). A bound on the k-gonality of facets of the hypermetric cone and related complexity problems. *Comput. Geom. Theory Appl.*, 2(5):241–254.

[Berman and Xu, 2005] Berman, A. and Xu, C. (2005). $\{0, 1\}$ completely positive matrices. *Linear Algebra and Its Applications*, 399(255):35–51.

[Berman and Xu, 2007] Berman, A. and Xu, C. (2007). Uniform and minimal $\{0, 1\}-cp$ matrices. *Linear and Multilinear Algebra*, 55(5):439–4561.

[Chaganty and Joe, 2006] Chaganty, N. R. and Joe, H. (2006). Range of correlation matrices for dependent Bernoulli random variables. *Biometrika*, 93(1):197–206.

[Daum and Werner, 2011] Daum, S. and Werner, R. (2011). A novel feasible discretization method for linear semi-infinite programming applied to basket option pricing. *Optimization*, 60(10-11):1379–1398.

[Deza et al., 1993] Deza, M., Grishukhin, V. P., and Laurent, M. (1993). The hypermetric cone is polyhedral. *Combinatorica*, 13(4):397–411.

[Deza and Laurent, 1997] Deza, M. M. and Laurent, M. (1997). *Geometry of Cuts and Metrics*. Springer Publishing Company, Incorporated, 1st edition.

[Embrechts et al., 2016] Embrechts, P., Hofert, M., and Wang, R. (2016). Bernoulli and tail-dependence compatibility. *Ann. Appl. Probab.*, 26(3):1636–1658.

[Emrich and Piedmonte, 1991] Emrich, L. J. and Piedmonte, M. R. (1991). A method for generating high-dimensional multivariate binary variates. *The American Statistician*, 45(4):302–304.

[Fiebig et al., 2017] Fiebig, U.-R., Strokorb, K., and Schlather, M. (2017). The realization problem for tail correlation functions. *Extremes*, 20(1):121–168.

[Friedland and Lim, 2016] Friedland, S. and Lim, L.-H. (2016). The computational complexity of duality. *SIAM Journal on Optimization*, 26(4):2378–2393.

[Kaibel and Weltge, 2015] Kaibel, V. and Weltge, S. (2015). A short proof that the extension complexity of the correlation polytope grows exponentially. *Discrete & Computational Geometry*, 53(2):397–401.

[Kochenberger et al., 2014] Kochenberger, G., Hao, J.-K., Glover, F., Lewis, M., Lü, Z., Wang, H., and Wang, Y. (2014). The unconstrained binary quadratic programming problem: a survey. *Journal of Combinatorial Optimization*, 28(1):58–81.

[Laurent and Poljak, 1996] Laurent, M. and Poljak, S. (1996). Gap inequalities for the cut polytope. *European Journal of Combinatorics*, 17(2):233 – 254.

[Lee, 1993] Lee, A. J. (1993). Generating random binary deviates having fixed marginal distributions and specified degrees of association. *The American Statistician*, 47(3):209–215.

[Lübbecke, 2010] Lübbecke, M. (2010). *Column Generation*, volume 8 of *Wiley Encyclopedia of Operations Research and Management Science (EORMS)*. Wiley.

[Macke et al., 2009] Macke, J. H., Berens, P., Ecker, A. S., Tolias, A. S., and Bethge, M. (2009). Generating spike trains with specified correlation coefficients. *Neural computation*, 21(2):397–423.

[Padberg, 1989] Padberg, M. (1989). The boolean quadric polytope: Some characteristics, facets and relatives. *Mathematical Programming*, 45(1):139–172.

[Pitowsky, 1991] Pitowsky, I. (1991). Correlation polytopes: Their geometry and complexity. *Mathematical Programming*, 50(1):395–414.

[Preisser and Qaqish, 2014] Preisser, J. S. and Qaqish, B. F. (2014). A comparison of methods for simulating correlated binary variables with specified marginal means and correlations. *Journal of Statistical Computation and Simulation*, 84(11):2441–2452.

[Qaqish, 2003] Qaqish, B. F. (2003). A family of multivariate binary distributions for simulating correlated binary variables with specified marginal means and correlations. *Biometrika*, 90(2):455–463.

[Teugels, 1990] Teugels, J. L. (1990). Some representations of the multivariate Bernoulli and binomial distributions. *Journal of Multivariate Analysis*, 32(2):256 – 268.

[Ziegler, 1999] Ziegler, G. M. (1999). Lectures on 0/1-polytopes. *ArXiv Mathematics e-prints*.

A Implementation details

A.1 Hardware and software choice

The implementation of the column generation method was carried out in Matlab 2015b. We have used IBM's Ilog Cplex 12.6.2 for Matlab to solve the arising LPs and BQPs. All tests were performed on a standard personal computer (processor: Intel Core i5-4090, 3.30 GHz, RAM: 16GB).

A.2 Cycling of column generation

One typical numerical issue in column generation is that due to numerical problems, some dual constraint corresponding to an index i of the inner set I_k might become slightly infeasible in some iteration. Especially in the proximity of the optimal solution, this can lead to the repeated introduction of this index i to the inner set and thus cycling. Using the feasibility-tolerance parameter of CPLEX, we can avoid this difficulty by forcing the LP solver to produce a "feasible enough" solution in each iteration.

A.3 Cardinality of I_0

In order to start the column generation one needs to find a suitable set I_0 such that P_{I_0} is feasible and ideally provides a reasonable small objective value. For this purpose, we use the primal heuristics presented in Proposition 2.6. By construction, we ensure $|I_0| \approx d^2$.

A.4 Computational accuracy

A matrix B is classified as Bernoulli-compatible whenever $vp(B) < 10^{-6}$.