OR Spectrum (2021) 43:777–808
https://doi.org/10.1007/s00291-021-00620-5

REGULAR ARTICLE

A column-oriented optimization approach for the generation of correlated random vectors

Jorge A. Sefair[1] · **Oscar Guaje**[2] · **Andrés L. Medaglia**[2]

Received: 20 August 2019 / Accepted: 28 January 2021 / Published online: 25 February 2021

Abstract

To induce a desired correlation structure among random variables, widely popular simulation software relies upon the method of Iman and Conover (IC). The underlying premise is that the induced Spearman rank correlation is a meaningful way to approximate other correlation measures among the random variables (e.g., Pearson's correlation). However, as expected, the desired a posteriori correlation structure often deviates from the Spearman correlation structure. Rooted in the same principle of IC, we propose an alternative distribution-free method based on mixed-integer programming to induce a Pearson correlation structure to bivariate or multivariate random vectors. We also extend our distribution-free method to other correlation measures such as Kendall's coefficient of concordance, Phi correlation coefficient, and relative risk. We illustrate our method in four different contexts: (1) the simulation of a healthcare facility, (2) the analysis of a manufacturing tandem queue, (3) the imputation of correlated missing data in statistical analysis, and (4) the estimation of the budget overrun risk in a construction project. We also explore the limits of our algorithms by conducting extensive experiments using randomly generated data from multiple distributions.

Keywords Correlated random vectors · Iman–Conover method · Spearman rank correlation · Pearson product-moment correlation · Kendall coefficient of concordance · Phi correlation coefficient · Relative risk · Simulation · Data imputation

✉ Jorge A. Sefair
jorge.sefair@asu.edu

[1] School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA

[2] Centro para la Optimización y Probabilidad Aplicada (COPA), Departamento de Ingeniería Industrial, Universidad de los Andes, Bogotá, Colombia

1 Introduction

Simulation studies often require the use of correlated input variables, as they may reflect realistic features of the problem at hand. For instance, Mitchell et al. (1977) describe a paper mill operation modeled as a tandem queue with two service stations: inspection and cutting. If a paper roll is of poor quality, it takes a long time to inspect because defective sections need to be removed and splices are made. When the roll reaches the cutting station, it needs to be processed slowly to avoid breaking it and to repair the splices. Thus, there is a high positive correlation between the processing times of the two stations that severely impact the system performance. In the project management context, Touran (1993) and Touran and Suphot (1997) investigated the impact of correlations on input variables when simulating the total cost of a construction project. They found that many of the cost components such as mechanical and electrical costs are positively correlated due to their common economic drivers, which could affect the risk of underestimating the budget. Cario and Nelson (1997) described that service times of a customer in a store may be correlated depending on the order size. Large orders may take longer processing times on the order desk, cashier, and loading dock stations than smaller orders. In a similar context, Patuwo et al. (1993) studied the impact in a queue performance due to correlated arrivals, and Cahen et al. (2018) derived models to determine the probability of large delays occurring in two correlated queues. Further, Hill and Reilly (2000) studied the effect that the correlation among input parameters has on the performance of optimization solvers like CPLEX and a heuristic by Toyoda (1975) when solving the two-dimensional knapsack problem. Similarly, Reilly (2009) analyzed the effect of parameter correlation in the 0-1 knapsack, capital budgeting (or multidimensional knapsack), set-covering, and generalized assignment problems.

Additional applications where correlated random variables also play a significant role include modeling the correlated pixel structure in image processing algorithms (Chakraborty 2006), the study of correlated survival rates in animal population modeling (Todd and Ng 2001; Dias et al. 2008), modeling correlated insurance claim amounts (Kolev and Paiva 2008), designing inventory policies with random correlated demands (Nasr and Maddah 2015), designing systems under correlated failures (Levitin and Xie 2006; Dai et al. 2004), generating correlated asset returns in portfolio optimization problems (Sefair et al. 2017), computing reliable shortest paths with correlated travel times (Zhang and Khani 2019; Corredor et al. 2020), modeling the propagation of flight leg delays (Yan and Kung 2016), constructing medical decision trees (Clark and El-Taha 1998), among others. Regardless of the field of application, it is well known that ignoring such input correlations may result in erroneous simulation results (see, for instance, Altiok and Melamed (2001) for an illustration in manufacturing systems).

In an era of data-intensive applications, the generation of correlated data becomes even more relevant in statistical analysis with missing data and imputation methods for machine learning and database management (Batista and Monard 2003). The motivation of such methods is that missing data can be critical for meaningful statistical inferences and may reduce the confidence in the

predictions (Little and Rubin 2019). Instead of reducing the database size by eliminating observations for which one or more variables have missing values, imputation techniques aim to complete the data using available information. Of particular interest to us are the imputation methods aiming not only to complete the missing data but also to preserve (or induce) a correlation structure between a pair of variables (Deb and Liew 2016).

There are well-known procedures to generate correlated multivariate random vectors for some probability distributions. For instance, Schmeiser and Lal (1982) and Rosenfeld (2008) proposed methods for generating correlated bivariate Gamma variates. Stanfield et al. (2004) generated multivariate Johnson random vectors that represent correlated product-reuse operation times in a production environment. Park and Dong (1998) focused on generating nonnegative correlated random variables for a class of infinitely divisible distributions, whereas Young and Beaulieu (2000) developed a Fourier-transform-based method to generate Rayleigh random samples. In the case of discrete random variables, Xiao (2017) focused on generating correlated discrete samples for any general distribution, whereas Shin and Pasupathy (2010) provided an algorithm for the generation of bivariate Poisson random vectors. Park et al. (1996), Qaqish (2003), and Shults (2017) proposed methods to generate correlated binary variables with specific marginal distributions and Biswas (2004) studied the generation of correlated categorical variables. Even though these methods take advantage of specific properties of the underlying distributions, they lack the flexibility to be easily extended to most distributions or a mixture of distributions.

To avoid the dependence on the properties of a specific probability distribution, more general procedures have focused on transforming a multivariate normal distribution into a multivariate distribution with target marginals and correlations. Li and Hammond (1975) proposed an analytical method based on this principle, but their procedure leads to the numerical solution of double-integral equations that might become computationally intensive and unstable given a certain degree of accuracy. To overcome this practical limitation, Van der Geest (1998) developed an algorithm to stabilize and increase the accuracy of Li and Hammond 's method, while Lurie and Goldberg (1998) presented a modified version in which a nonlinear optimization procedure minimizes the distance between the achieved and target correlation matrix. Cario and Nelson (1997) proposed the NORTA (normal-to-anything) approach in which a standard multivariate normal distribution is transformed into any multivariate distribution with a target correlation matrix. Hill and Reilly (1994) generated random vectors with the desired marginals and correlations through mixtures of distributions with extreme correlations. Further, Haas (1999) studied methods to generate bivariate correlated random numbers based on copulas. In this case, the copula and its parameters have to be specified (or estimated) as input to the random number generation procedure. Finding the appropriate copula for any given correlation metric may be challenging. Although all of these methods can deal with any desired marginal distributions, some rely on modifications of the input data and exploit the specific properties of the subjacent distributions. In some cases, they are able to induce correlations only between variables of the same type (i.e., continuous or discrete) but cannot handle distribution mixtures. Further, some approaches are

designed for a specific correlations metric and may fail to induce target correlation levels even if they are attainable (Ghosh and Henderson 2003).

Other methods have focused on reordering the samples of previously generated random variates *a posteriori*, to induce the desired correlation structure. Polge et al. (1973) proposed an algorithm to sort the samples from a single univariate distribution to induce a desired sample autocorrelation. Based on the premise that rank correlation is a meaningful way to define dependencies among variables, Iman and Conover (1982) proposed a reordering transformation scheme to induce a target Spearman rank correlation. Using a heuristic procedure, Charmpis and Panteli (2004) reorder multivariate samples to induce target Pearson product-moment correlations. In the context of design of experiments, Harris et al. (1995a) and Harris et al. (1995b) devise heuristic methods to produce minimum-correlation samples using Latin hypercube. Although these approaches do not explicitly use the joint probability density functions (or cumulative density functions), given their (approximate) numerical solution there is no guarantee that the correlation found is the closest to the target given the available samples. Indeed, some methods require a fine-tuning of the search parameters, which may have a big impact on the solution quality. Additionally, existing methods are not easily adaptable to other correlation coefficients beyond Pearson and Spearman.

To induce a given target correlation structure, widely popular simulation software like Crystal Ball and @Risk seem to rely upon the Iman–Conover (IC) method for generating correlated random variables, rather than sampling from the multivariate distribution (Van der Geest 1998; Haas 1999; Oracle 2019). The IC method works as follows (Mildenhall 2005). Given samples of n values from two known marginal distributions X and Y and a desired correlation ρ between them, reorder the samples to have the same rank-order correlation as a reference distribution (of size $n \times 2$) with desired linear correlation ρ. Since linear correlation and rank correlation are typically close, the reordered output will have approximately the desired correlation structure. Some of the key characteristics of the Iman–Conover (IC) method is that it: (1) is simple to use; (2) is distribution-free; (3) preserves the exact form of the marginal distributions; 4) and may be used with any type of sampling scheme. As stated in Iman and Conover (1982), the underlying premise is that the induced rank correlations are a meaningful way to define the dependencies among the random variables. To some degree, the Spearman rank correlation is used as a proxy to induce a target Pearson product-moment (linear) correlation.

This paper is rooted on the same principles of IC, yet it presents a more general framework based on a mixed-integer programming (MIP) formulation that is able to induce different correlation structures. We use the modeling approach of Medaglia and Sefair (2009) and extend its application to the multivariate case and other correlation measures. Further, we propose an efficient solution algorithm that handles problems with larger samples. The proposed method, aside from coping with the widely popular Spearman rank correlation and Pearson product-moment correlation, is able to induce other correlation measures such as Kendall's W, Phi correlation coefficient, and relative risk. We present an MIP model that induces such correlation structures to bivariate vectors, as well as a column generation approach that efficiently generates large samples of correlated vectors. We then extend the bivariate

methodology to the multivariate case, in which random vectors are correlated one at a time. This strategy produces good quality results in terms of the achieved correlations. Our method handles both discrete and continuous random variables (or a mixture of them). To illustrate its practical use, we provide two case studies that describe how our method can be integrated in Monte Carlo and discrete-event simulation models, including the simulation of a healthcare facility and a manufacturing tandem queue. We also discuss how our approach can be used for data imputation in machine learning and estimating the risk of running over the budget in a construction project.

The remainder of this article is organized as follows: Section 2 presents the proposed framework with the underlying MIP model for inducing bivariate correlations and a column generation procedure to accelerate its solution. Section 3 extends the results from the bivariate case to the generation of multivariate correlated random vectors. Section 4 illustrates how to modify our models to induce other correlations such as Spearman, relative risk, and Kendall and Phi coefficients of concordance. Section 5 illustrates the use of the proposed approach in different settings, including hospital logistics, simulation of tandem queues with correlated service times, data imputation, and estimating the risk of overrunning the budget in construction projects. Section 6 describes the computational performance of our procedure for the generation of large-scale samples. Section 7 contains conclusions and recommendations for future research.

2 Generating bivariate correlated random vectors

In this section, we describe a mixed-integer programming (MIP) model and a column generation approach to induce a given Pearson correlation to a bivariate random sample. Although we center our attention on the Pearson correlation, our methodology can be extended to induce other correlation measures. We describe such extensions in Sect. 4, where we outline the required changes and adjustments.

2.1 Mixed-integer programming model

Let X and Y be a pair of random variables with probability density functions $f_X(x)$ and $f_Y(y)$, and let $\{x_i : i = 1, 2, \ldots, n\}$ and $\{y_i : i = 1, 2, \ldots, n\}$ be a pair of random samples from X and Y, respectively. The Pearson linear or product-moment correlation, denoted by ρ^P_{XY}, is a statistic that measures the degree of the linear association between two variables. Formally, the Pearson correlation is defined as follows:

Definition 1 (*Pearson correlation*) Let $\rho^P_{XY} \equiv \frac{\sum_{i=1}^n (x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^n (x_i-\bar{x})^2 \sum_{i=1}^n (y_i-\bar{y})^2}}$ be the Pearson product-moment correlation, where $\bar{x}$ and $\bar{y}$ are the sample means for X and Y, respectively.

Using the samples for X and Y, our goal is to induce a desired Pearson correlation while obtaining vectors with marginal distributions identical to that of $f_X(x)$ and $f_Y(y)$. To do so, we use a *matching* approach, in which x-variates are paired one-to-one with y-variates. This procedure not only induces the desired correlation, but also preserves the marginal distributions because the input data are not modified. Formally, we use a (bipartite) graph denoted by $G = (\mathcal{N}_X \cup \mathcal{N}_Y, \mathcal{A})$, where $\mathcal{N}_X$ and $\mathcal{N}_Y$ are the node sets representing the observations from the random samples of X and Y such that $\mathcal{N}_X \cap \mathcal{N}_Y = \emptyset$; and $\mathcal{A} = \mathcal{N}_X \times \mathcal{N}_Y$ are the arcs matching observations between sets $\mathcal{N}_X$ and $\mathcal{N}_Y$. Figure 1 shows this bipartite graph.

To model the matching decisions, we use the binary decision variable z_{ij} that takes the value of 1 if variates x_i and y_j are paired as a random sample (x_i, y_j), and takes the value of 0, otherwise. Additionally, we denote the *correlation cost* of matching variates x_i and y_j by c_{ij}. Under Pearson correlation, this cost corresponds to $c_{ij} \equiv (x_i - \bar{x})(y_j - \bar{y})$ for a given arc $(i,j) \in \mathcal{A}$, which allows us to rewrite the Pearson correlation coefficient in terms of the z-variables as in Eq. 1. For a given matching, Eq. (1) returns the *achieved* Pearson coefficient

$$\widehat{\rho}^{P}_{XY} \equiv \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} z_{ij}}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}. \tag{1}$$

The following integer program induces a given Pearson product-moment, $\overline{\rho}^{P}_{XY}$, to the random sample.

$$\min \quad \left| \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} z_{ij}}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} - \overline{\rho}^{P}_{XY} \right| \tag{2a}$$

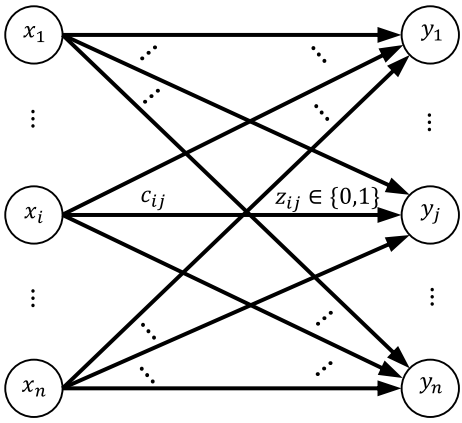

Fig. 1 Underlying bipartite graph for bivariate correlation

$$\text{s.t.} \quad \sum_{j=1}^{n} z_{ij} = 1, \quad \forall i = 1, \ldots, n \tag{2b}$$

$$\sum_{i=1}^{n} z_{ij} = 1, \quad \forall j = 1, \ldots, n \tag{2c}$$

$$z_{ij} \in \{0, 1\}, \quad \forall i, j = 1, \ldots, n \tag{2d}$$

The objective function in (2a) minimizes the absolute gap between the achieved correlation, $\widehat{\rho}^{P}_{XY}$, and the target correlation $\overline{\rho}^{P}_{XY}$ after matching the observations into bivariate vectors. Constraints (2b) guarantee that each observation from the sample of the random variable X (i.e., in $\mathcal{N}_X$) is assigned to one observation from the sample of the random variable Y (i.e., in $\mathcal{N}_Y$). Likewise, (2c) guarantees that each observation from $\mathcal{N}_Y$ is assigned one observation of $\mathcal{N}_X$. Constraints (2d) define the binary nature of the assignment variables. We denote the set of feasible solutions $\mathbf{z} \in \{0, 1\}^{n \times n}$ satisfying (2b)–(2d) as $\mathcal{Z}$.

To linearize the model described by (2a)–(2d), we introduce the nonnegative deviation variables δ^- and δ^+. These deviation variables capture the negative or positive gap between $\widehat{\rho}^{P}_{XY}$ and $\overline{\rho}^{P}_{XY}$, respectively. The resulting linearized MIP, which we call bivariate correlated vector generation MIP (BCVG-MIP), is given by

$$[\text{BCVG-MIP}] \quad \min \delta^- + \delta^+ \tag{3a}$$

$$\text{s.t.} \quad \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} z_{ij}}{\sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2 \sum_{i=1}^{n} (y_i - \overline{y})^2}} + \delta^- - \delta^+ = \overline{\rho}^{P}_{XY} \tag{3b}$$

$$\mathbf{z} \in \mathcal{Z} \tag{3c}$$

$$\delta^-, \delta^+ \geq 0, \tag{3d}$$

where (3a) minimizes the deviation from the target correlation $\overline{\rho}^{P}_{XY}$ defined in (3b). Note that BCVG-MIP is a well-structured *assignment problem* with side constraints (3b) and (3d).

2.2 Column generation procedure

Our formulation for BCVG-MIP has $n^2 + 2$ variables (i.e., n^2 z-variables and 2 δ -variables). However, in any optimal solution, only n out of the n^2 z-variables are nonzero. This means that most of the z-variables may not be necessary to solve the MIP, yet they create a computational burden to the branch-and-bound algorithm. Based on this observation, we develop a column generation procedure that seeks to solve the original problem by using only a subset of the z-variables (see Desaulniers et al. (2006) for further details on the column generation procedure).

To this end, we relax the binary nature of the z-variables in BCVG-MIP (i.e., Constraints (2d)) and denote its linear relaxation as BCVG-LP. We denote the dual variable associated with constraint (3b) as σ and the duals associated with (2b) and (2c) as u_i and v_j, respectively. In matrix notation, the column corresponding to variable z_{ij} in BCVG-LP has the form

$$\mathbf{a}_{ij} = \begin{bmatrix} c_{ij}/S_{XY} \\ \mathbf{e}_i \\ \mathbf{e}_j \end{bmatrix},$$

where $\mathbf{e}_i$ is an n-dimensional vector of zeros with a 1 in the i-th position and $S_{XY} = \sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}$. Using this notation, the restricted master problem corresponding to BCVG-LP is given by

$$[\text{BCVG-RMP}] \quad \min \delta^- + \delta^+ \tag{4a}$$

$$\text{s.t.} \quad \sum_{(i,j)\in\mathcal{C}} \mathbf{a}_{ij} z_{ij} + \tilde{\mathbf{e}}_1 \delta^- - \tilde{\mathbf{e}}_1 \delta^+ = \mathbf{b} \tag{4b}$$

$$z_{ij} \geq 0, \quad \forall (i,j) \in \mathcal{C} \tag{4c}$$

$$\delta^-, \delta^+ \geq 0, \tag{4d}$$

where $\mathcal{C} \subseteq \mathcal{N}_X \times \mathcal{N}_Y$ represent a subset of the columns in BCVG-LP; $\tilde{\mathbf{e}}_1$ is a $(2n+1)$ -dimensional vector of zeros with a 1 in the first position; and $\mathbf{b}$ is a $(2n+1)$-dimensional vector of ones, except for the first position which equals $\overline{\rho}^P_{XY}$. Because z-variables are not part of (4a), then the reduced cost of any non-basic variable z_{ij} is given by

$$\begin{aligned} r_{ij} &= 0 - [\sigma \ldots u_i \ldots v_j] \cdot \mathbf{a}_{ij} \\ &= -\frac{c_{ij}}{S_{XY}}\sigma - u_i - v_j. \end{aligned} \tag{5}$$

Equation (5) allows us to compute the reduced cost for any non-basic variable in closed-form without solving a subproblem.

Algorithm 1 describes our column generation procedure. In Step 1, the algorithm constructs an initial basis for BCVG-RMP consisting of $2n+1$ columns (i.e., the number of constraints in the model, excluding non-negativity constraints). To do so, we use the columns of the n z-variables corresponding to the IC method's output, which induces a rank-order correlation that *approximates* the Pearson's product-moment correlation. Using this initialization, our model can only improve (or replicate) the results of the IC method. We also add the columns corresponding to the deviation variables δ^+ and δ^-, and $n-1$ additional columns corresponding to assignment z-variables for the input data as produced by the random number generator. This initialization provides a feasible solution consisting of the assignment variables

from the IC method solution being equal to one, one of the deviation variables equal to $|\overline{\rho}^P_{XY} - \widehat{\rho}^P_{XY}|$ and the other equal to zero (depending on whether the IC method underestimates or overestimates $\overline{\rho}^P_{XY}$), and the remaining variables equal to zero.

In Line 2, Algorithm 1 solves BCVG-RMP to obtain the initial values for the dual variables, which are used to calculate the reduced costs in Line 3. The `while` loop in Lines 4–9 iterates until there is no attractive non-basic variables outside $\mathcal{C}$ in terms of reduced cost. Line 5 finds the non-basic variable with the most negative reduced cost (i.e., we use the Dantzig rule (Lübbecke and Desrosiers 2005)), and Line 6 augments BCVG-RMP with the information of the corresponding column. Using the updated information, Line 7 re-solves BCVG-RMP, whose dual variables are used to calculate the reduced costs in Line 8. Upon termination of the `while` loop, Line 10 solves the integer version of BCVG-RMP with the columns generated thus far. Line 11 returns the optimal solution to this problem and the corresponding deviations from the target correlation.

Algorithm 1: Column generation procedure to induce bivariate correlation

Input: n observations from each random variable X and Y, target correlation $\overline{\rho}^P_{XY}$

Output: Correlated vectors with Pearson coefficient $\widehat{\rho}^P_{XY}$ and deviation from target correlation $\overline{\rho}^P_{XY}$

1 Construct an initial feasible solution to BCVG-RMP

2 Solve BCVG-RMP to obtain σ, u_i, and v_j, $\forall i, j = 1, \ldots, n$

3 Calculate reduced costs r_{ij}, $\forall (i, j) \notin \mathcal{C}$

4 **while** $\exists r_{ij} < 0, (i, j) \notin \mathcal{C}$ **do**

5 $\quad (i^*, j^*) \leftarrow \arg\min_{(i,j) \notin \mathcal{C}} r_{ij}$

6 $\quad \mathcal{C} \leftarrow \mathcal{C} \cup (i^*, j^*)$, add $z_{i^* j^*}$ and $\mathbf{a}_{i^* j^*}$ to BCVG-RMP

7 $\quad$ Solve BCVG-RMP to obtain σ, u_i, and v_j, $\forall i, j = 1, \ldots, n$

8 $\quad$ Calculate reduced costs r_{ij}, $\forall (i, j) \notin \mathcal{C}$

9 **end**

10 Solve BCVG-RMP with (2d) to obtain $\mathbf{z}^*$, δ^{-*}, and δ^{+*}

11 Return $\mathbf{z}^*$, δ^{-*}, and δ^{+*}

Algorithm 1 terminates in at most $n^2 - 2n - 1$ iterations, in which case BCVG-RMP will contain all the columns in BCVG-MIP. Note that in Line 10 we solve the integer version of BCVG-RMP with the columns generated thus far, which provides an upper bound on the optimal solution to BCVG-MIP. Although this strategy may not provide the optimal solution to BCVG-MIP, our computational results illustrate that this procedure is enough to produce near-optimal solutions without the need of a branch-and-price algorithm.

3 Generating multivariate correlated random vectors

In this section, we extend our bivariate approach to the multivariate case. We induce a given matrix of target Pearson correlation coefficients on a set of multivariate random samples. We process the given random vectors merging them one at a time, iteratively inducing bivariate correlations with all vectors already processed. Once a vector is processed, its observations cannot be shuffled and our method proceeds with the next vector. This process is repeated until all vectors are merged. As in the bivariate case, our approach can be easily extended to include other multivariate correlations metrics, which we discuss in Sect. 4.

3.1 Mixed-integer programming model

Let $\mathcal{K}$ be the index set of random variables X_k, and let $\{x_{ki} : i = 1, 2, \ldots, n\}$ be a sample of n random observations for random variable X_k ($k \in \mathcal{K}$). Let $\overline{\rho}_{k_1,k_2}$ be the target correlation between variables X_{k_1} and X_{k_2}, all of which are stored in the $|\mathcal{K}| \times |\mathcal{K}|$ symmetric correlation matrix $\overline{\mathbf{R}} \equiv [\overline{\rho}_{k_1,k_2}]$, for $(k_1, k_2) \in \mathcal{K} \times \mathcal{K}$. Let $\bar{\mathcal{K}} \subset \mathcal{K}$ be the subset of indices corresponding to variables whose samples will not be shuffled. For a given $k \in \mathcal{K} \setminus \bar{\mathcal{K}}$, let $y_{k,i,j}$ be a binary decision variable that equals 1 if the i-th observation of variable X_k is placed in position j of the correlated random (output) vector, and equals 0, otherwise. Associated with variable $y_{k,i,j}$, there is a cost $c_{k,k',i,j} \equiv (x_{k,i} - \bar{x}_k)(x_{k',j} - \bar{x}_{k'})$ that accounts for the interaction of the i-th observation of variable X_k and the j-th observation of variable $X_{k'}$ in the induced correlation, where $\bar{x}_k$ and $\bar{x}_{k'}$ are the sample means. To induce Pearson product-moment correlations between X_k and $X_{k'}$, for $k \in \mathcal{K} \setminus \bar{\mathcal{K}}$ and all $k' \in \bar{\mathcal{K}}$, we define the MIP in (6a)–(6f), which we denote by MCVG-MIP(k, $\bar{\mathcal{K}}$). This problem is parameterized to emphasize that it seeks to induce target correlations $\bar{\rho}_{k,k'}$ for $k \in \mathcal{K} \setminus \bar{\mathcal{K}}$ and all $k' \in \bar{\mathcal{K}}$ by only rearranging X_k. Similar to the bivariate case, we use $\delta^-_{k,k'}$ and $\delta^+_{k,k'}$ to calculate the deviation from the target correlation for each $k' \in \bar{\mathcal{K}}$. For a given k, vector $\mathbf{y}_k$ contains the y-variables. We point out that MCVG-MIP(k, $\bar{\mathcal{K}}$) is very similar to BCVG-MIP with the distinction that it simultaneously induces bivariate correlations between X_k and all $X_{k'}$ variables for $k' \in \bar{\mathcal{K}}$. MCVG-MIP(k, $\bar{\mathcal{K}}$) and BCVG-MIP are identical when $|\bar{\mathcal{K}}| = 1$.

$$[\text{MCVG-MIP}(k, \bar{\mathcal{K}})] \quad \min \quad \sum_{k' \in \bar{\mathcal{K}}} (\delta^-_{k,k'} + \delta^+_{k,k'}) \tag{6a}$$

$$\text{s.t.} \quad \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} c_{k,k',i,j} y_{k,i,j}}{\sqrt{\sum_{i=1}^{n} (x_{k,i} - \bar{x}_k)^2 \sum_{i=1}^{n} (x_{k',i} - \bar{x}_{k'})^2}} + \delta_{k,k'} - \delta^+_{k,k'} = \overline{\rho}^P_{k,k'}, \quad k' \in \bar{\mathcal{K}} \tag{6b}$$

$$\sum_{j=1}^{n} y_{k,i,j} = 1, \quad \forall i = 1, \ldots, n \tag{6c}$$

$$\sum_{i=1}^{n} y_{k,i,j} = 1, \quad \forall j = 1, \ldots, n \tag{6d}$$

$$y_{k,i,j} \in \{0, 1\}, \quad \forall i, j = 1, \ldots, n \tag{6e}$$

$$\delta^{-}_{k,k'}, \delta^{+}_{k,k'} \geq 0, \quad k' \in \bar{\mathcal{K}} \tag{6f}$$

3.2 Column generation procedure

We modify the column generation approach from the bivariate case in order to solve (6a)–(6f). To this end, we define $S_{x_k, x_{k'}} = \sqrt{\sum_{i=1}^{n}(x_{k,i} - \bar{x}_k)^2 \sum_{i=1}^{n}(x_{k',i} - \bar{x}_{k'})^2}$ and the vectors

$$\mathbf{a}_{k,i,j} = \left[\cdots \; c_{k,k',i,j}/S_{x_k, x_{k'}} \; \cdots \mid \mathbf{e}_i \mid \mathbf{e}_j\right]^T,$$

for each $(i, j) \in \{1, \ldots, n\} \times \{1, \ldots, n\}$. Using this notation, the restricted master problem for the column generation approach to solve MCVG-MIP($k, \bar{\mathcal{K}}$) is given by

$$[\text{MCVG-RMP}(k, \bar{\mathcal{K}})] \quad \min \sum_{k' \in \bar{\mathcal{K}}} (\delta^{-}_{k,k'} + \delta^{+}_{k,k'}) \tag{7a}$$

$$\text{s.t.} \quad \sum_{(i,j) \in \mathcal{C}} \mathbf{a}_{k,i,j} y_{k,i,j} + \tilde{\mathbf{I}}\boldsymbol{\delta}^{-} - \tilde{\mathbf{I}}\boldsymbol{\delta}^{+} = \tilde{\mathbf{b}} \tag{7b}$$

$$y_{k,i,j} \geq 0, \quad \forall (i, j) \in \mathcal{C} \tag{7c}$$

$$\delta^{-}_{k,k'}, \delta^{+}_{k,k'} \geq 0, k' \in \bar{\mathcal{K}}, \tag{7d}$$

where $\mathcal{C}$ is a subset of columns of the linear relaxation of MCVG-MIP($k, \bar{\mathcal{K}}$), $\tilde{\mathbf{I}}$ is a $\left(|\bar{\mathcal{K}}| + 2n\right) \times |\bar{\mathcal{K}}|$ matrix consisting of an identity matrix in its top $|\bar{\mathcal{K}}| \times |\bar{\mathcal{K}}|$ portion, and zeros everywhere else. Additionally, $\tilde{\mathbf{b}}$ is a $(|\bar{\mathcal{K}}| + 2n)$-dimensional vector that contains the ρ-values in the top $|\bar{\mathcal{K}}|$ positions, and ones in the remaining $2n$ positions. Vectors $\boldsymbol{\delta}^{-}$ and $\boldsymbol{\delta}^{+}$ are of dimension $|\bar{\mathcal{K}}|$ and contain the values of $\delta^{-}_{k,k'}$ and $\delta^{+}_{k,k'}$, $\forall k' \in \bar{\mathcal{K}}$, respectively. We denote the dual variables associated with Constraints (6b) as $\sigma_{k'}$ and the duals associated with (3c) as u_i and v_j—similar to those in (2b) and (2c)—respectively. The reduced cost of any non-basic variable $y_{k,i,j}$ in MCVG-RMP($k, \bar{\mathcal{K}}$), for $(i, j) \notin \mathcal{C}$, is given by

$$r_{i,j} = -\sum_{k' \in \bar{\mathcal{K}}} \frac{c_{k,k',i,j}}{S_{x_k,x_{k'}}} \sigma_{k'} - u_i - v_j. \tag{8}$$

Algorithm 2 describes our column generation procedure for the multivariate case. The underlying principle is to induce the target correlations by shuffling one random vector at a time. After a vector is processed, the arranged data becomes fixed and cannot be shuffled in future iterations. By doing so, we are able to iteratively use the method for the bivariate case. In Step 1, the algorithm constructs the permutation in which random vectors will be processed and whose target correlations will be sequentially induced. The `for` loop in Lines 2–14 induces the target correlation between a random vector and all those already in $\bar{\mathcal{K}}$, according to the order in the permutation. Line 3 finds an initial basis for MCVG-RMP($(k, \bar{\mathcal{K}})$) consisting of $2n + 2$ columns (i.e., the number of constraints in the model, excluding non-negativity constraints). To do so, we use the same strategy as in the bivariate case. In Line 4, Algorithm 2 solves MCVG-RMP($(k, \bar{\mathcal{K}})$) to obtain the initial values for the dual variables, which are used to calculate the reduced costs in Line 5. The `while` loop in Lines 6–11 is identical to that in Algorithm 1 but solving MCVG-RMP($(k, \bar{\mathcal{K}})$) at each iteration. Upon termination of the `while` loop, Line 12 solves the integer version of MCVG-RMP($(k, \bar{\mathcal{K}})$) with the columns generated thus far. Line 13 adds variable X_k to the list of variables that cannot be shuffled in future iterations. In this step, the samples of X_k are sorted as prescribed by $\mathbf{y}_k^*$. Line 15 returns the optimal vectors obtained at each iteration for each variable , which are used to calculate the deviations from the target correlations.

In Sect. 6, we demonstrate that Algorithm 2 produces better results than the approach of Iman and Conover (1982) in terms of inducing correlations that are closer to the target correlations. This is despite its dependency on the initial permutation of variables, which determines the order in which the vectors are processed, and the fact that solving the integer version of MCVG-RMP with the columns generated thus far (i.e., Line 12) may not produce an optimal solution to (6a)–(6f).

Algorithm 2: Column generation procedure to induce multivariate correlations

Input: n observations for each variable $X_k, k \in \mathcal{K}$; target correlation matrix $\overline{\mathbf{R}}$

Output: Correlated vectors with Pearson coefficient $\widehat{\mathbf{R}}$

Construct a permutation $k_{[1]}, k_{[2]}, \ldots, k_{[|\mathcal{K}|]}$ of the indices in $\mathcal{K}$; initialize $\bar{\mathcal{K}} = \{k_{[1]}\}$
for $k = k_{[2]}$ *to* $k_{[|\mathcal{K}|]}$ **do**
  Construct an initial feasible solution to MCVG-RMP$(k, \bar{\mathcal{K}})$
  Solve MCVG-RMP$(k, \bar{\mathcal{K}})$ to obtain $\sigma_{k'}$, for each $k' \in \bar{\mathcal{K}}$, u_i, and v_j, $\forall i, j = 1, \ldots, n$
  Calculate reduced costs r_{ij}, $\forall (i, j) \notin \mathcal{C}$
  while $\exists r_{ij} < 0, (i, j) \notin \mathcal{C}|$ **do**
    $(i^*, j^*) \leftarrow \arg\min_{(i,j) \notin \mathcal{C}} r_{i,j}$
    $\mathcal{C} \leftarrow \mathcal{C} \cup (i^*, j^*)$, add y_{k,i^*,j^*} and $\mathbf{a}_{k,i^*,j^*}$ to MCVG-RMP$(k, \bar{\mathcal{K}})$
    Solve MCVG-RMP$(k, \bar{\mathcal{K}})$ to obtain $\sigma_{k'}$, for each $k' \in \bar{\mathcal{K}}$, u_i, and v_j, $\forall i, j = 1, \ldots, n$
    Calculate reduced costs r_{ij}, $\forall (i, j) \notin \mathcal{C}$
  end
  Solve MCVG-RMP$(k, \bar{\mathcal{K}})$ with $\mathbf{y}_k \in \{0, 1\}^{n \times n}$ to obtain $\mathbf{y}_k^*$
  $\bar{\mathcal{K}} \leftarrow \{k\}$
end
Return $\mathbf{y}_k^*$ for each $k \in \mathcal{K}$

4 Extensions to other correlation measures

In Sect. 2, we focused on inducing Pearson product-moment correlation. However, there are other correlation measures that are better suited for certain applications. Our methodology is capable of targeting these correlation measures with only minor changes to the constraints while maintaining the same underlying structure in the MIP. In this section, we present alternative correlation measures and describe how to pursue them for the bivariate case. The extension to the multivariate can be achieved in the same manner as in the Pearson product-moment correlation.

4.1 Targeting Spearman rank correlation

The Spearman rank correlation, denoted by ρ_{XY}^S, is a nonparametric (distribution-free) rank statistic that measures the degree of association between two variables. It is a measure of monotone association that is used when the distribution of the data

makes the Pearson's correlation coefficient misleading. The Spearman rank correlation is defined as follows:

Definition 2 (*Spearman rank correlation*) Let $\rho^S_{XY} \equiv 1 - \frac{6\sum_{i=1}^n d_i^2}{n(n^2-1)}$ be the Spearman rank correlation, where $d_i \equiv r_X(x_i) - r_Y(y_i)$ is the difference between the ranks of the variates x_i and y_i within the corresponding samples for X and Y, respectively.

Under the Spearman rank correlation, let $c_{ij} \equiv (r_X(x_i) - r_Y(y_j))^2$ be the correlation cost for a given arc $(i,j) \in \mathcal{A}$. Then, the mathematical programming model that induces a given Spearman rank correlation $\overline{\rho}^S_{XY}$ to the random sample is given by (3a)–(3d), replacing (3b) by

$$\frac{n(n^2-1) - 6\sum_{i=1}^n \sum_{j=1}^n c_{ij} z_{ij}}{n(n^2-1)} + \delta^- - \delta^+ = \overline{\rho}^S_{XY}. \tag{9}$$

In this case, each column in the corresponding BCVG-RMP is the same as in (4a)–(4d) but using $S_{XY} = -n(n^2-1)/6$ and with the top entry of $\mathbf{b}$ equal to $\overline{\rho}^S_{XY}$.

4.2 Targeting Kendall's W

Kendall's W (also called Kendall's coefficient of concordance) is a measure of the agreement between m sets of ranks for n objects (Kendall and Babington-Smith 1939; Wallis 1939; Sheskin 2000). Given its properties, Kendall's W has been used in areas such as ecology (Legendre 2005), medical decision making (Leschied et al. 2016), psychology (Sigler and Tallent-Runnels 2006), among others. For $m = 2$, Kendall's W is defined as follows:

Definition 3 (*Kendall's W*) Let $\rho^K_{XY} \equiv \frac{3U - 3n(n+1)^2}{n(n^2-1)}$ be the Kendall's W coefficient, where $U \equiv \sum_{i=1}^n (x_i + y_i)^2$ and x_i and y_i are the ranks assigned to the i-th object in the sets of ranks X and Y, respectively.

Under Kendall's W, let $c_{ij} \equiv (x_i + y_j)^2$ be the correlation cost for a given arc $(i,j) \in \mathcal{A}$. Then, the corresponding MIP model that induces a given Kendall's W ρ^K_{XY} to the random sample (i.e., permutation of n numbers) is given by (3a)–(3d), replacing (3b) by:

$$\frac{3\sum_{i=1}^n \sum_{j=1}^n c_{ij} z_{ij} - 3n(n+1)^2}{n(n^2-1)} + \delta^- - \delta^+ = \overline{\rho}^K_{XY} \tag{10}$$

In this case, each column in the corresponding BCVG-RMP is the same as in (4a)–(4d) but using $S_{XY} = n(n^2-1)/3$ and with the top entry of $\mathbf{b}$ equal to $\overline{\rho}^K_{XY}$.

4.3 Targeting Phi correlation coefficient

The proposed MIP model is also able to induce correlations to categorical variables. The Phi correlation coefficient (also known as the mean square contingency coefficient) is a special case of the Pearson correlation coefficient that measures the association between two binary variables (Sheskin 2000). The corresponding column generation formulation is identical to (4a)–(4d).

4.4 Targeting relative risk

The relative risk (or risk ratio) is also a measure of association between categorical variables. The relative risk ratio, defined as the ratio of the probability of occurrence of an event in a group to its probability in another group, has been widely used in epidemiological studies. For instance, to compare the relative probabilities of contracting a disease (Cornfield 1951; Morris and Gardner 1988; Sheskin 2000), the relative risk is typically calculated as the ratio of the probability of contracting the disease in the higher-risk group, divided by the probability of the same event, but in the lower risk group. Mathematically, the relative risk is defined as follows:

Definition 4 (*Relative risk*) Let $\rho^R_{XY} \equiv \frac{(1-\bar{y})}{\bar{y}}\left(\frac{n\bar{x}}{\sum_{i=1}^{n} x_i(1-y_i)} - 1\right)$ be the relative risk measure, where $\bar{x}$ and $\bar{y}$ are the means of the binary observations within the corresponding samples for X and Y, respectively. Then, the relative risk can be expressed as $\rho^R_{XY} = \frac{P\{X=1|Y=1\}}{P\{X=1|Y=0\}}$.

Under the relative risk measure let $c_{ij} \equiv x_i(1-y_j)$ be the correlation cost for a given arc $(i,j) \in \mathcal{A}$. Then, the corresponding MIP model that induces a given relative risk measure $\bar{\rho}^R_{XY}$ to the random sample is given by (3a)–(3d), replacing (3b) by:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}z_{ij} + \delta^- - \delta^+ = \frac{n\bar{x}(1-\bar{y})}{\left(1+\bar{y}(\bar{\rho}^R_{XY}-1)\right)} \tag{11}$$

Although in this case variables δ^- and δ^+ are not deviations from the target correlation, they still drive the MIP to produce a correlation as close as possible to the target value $\bar{\rho}^R_{XY}$. In this case, each column in the corresponding BCVG-RMP is the same as in (4a)–(4d) but using $S_{XY} = 1$ and with the top entry of $\mathbf{b}$ equal to the right-hand side of (11).

5 Illustrative applications

The purpose of this section is to illustrate, by means of different examples, that our proposed approach can be used in different practical settings. In Sect. 5.1, we illustrate the use of our models in a healthcare application, whereas in Sect. 5.2, we illustrate how our models can be blended with Monte Carlo and discrete-event

Table 1 Capacity overestimation of the hospital laboratory facility by simulating X and Y as independent variables

	$X = 1$	$X = 0$	Total
$Y = 1$	25	70	95
$Y = 0$	19	36	55
Total	44	106	150

Table 2 Determining the right capacity for the hospital laboratory facility by inducing $\widehat{\rho}^{\phi}_{XY} = 0.40$

	$X = 1$	$X = 0$	Total
$Y = 1$	41	54	95
$Y = 0$	3	52	55
Total	44	106	150

simulation environments. In Sect. 5.3, we illustrate how to use our model for data imputation in machine learning and database applications. Section 5.4 presents a multivariate example related to the budget overrun risk in construction projects.

5.1 Simulating correlated laboratory exams

As a first example, we consider the case of a hospital laboratory that analyzes two blood tests, A and B. Let X and Y be two Bernoulli distributed random variables representing whether a patient takes test A or B, with probabilities $p_X = 0.3$ and $p_Y = 0.6$, respectively. The hospital is interested in estimating the number of patients attending the laboratory facility, considering that only one blood sample is collected if a patient needs both examinations. The fact that some patients need both examinations is modeled by a Phi correlation coefficient $\overline{\rho}^{\phi}_{XY} = 0.4$.

We simulate the laboratory operation considering $n = 150$ patients. Table 1 shows the simulation results when variables X and Y are simulated ignoring their correlation. In this scenario, a total of 114 patients ($= 25 + 19 + 70$) require the laboratory services for at least one examination. However, we obtain $\widehat{\rho}^{\phi}_{XY} = -0.09$, which is far from the target of 0.4.

We solve BCVG-MIP using the definitions in Sect. 4.3 to induce the desired Phi correlation coefficient. Results in Table 2 show that in this case, 98 patients ($= 41 + 54 + 3$) go to the laboratory facility to take a blood examination. Moreover, we obtain $\widehat{\rho}^{\phi}_{XY} = 0.40$ (perfect fit), avoiding the overestimation in the number of patients attending the laboratory facility.

5.2 Performance evaluation of a tandem queue with correlated service times

To illustrate how BCVG-MIP can be embedded into a discrete-event simulation environment, we consider a tandem queue system with two stations and correlated service times. As pointed out by Law and Kelton (2000), such a system could represent a shop where incoming parts are inspected for defects in the first station and

marked for repair in the second station. In this setting, it is reasonable to expect that a badly damaged part takes a long time to inspect and repair, as opposed to a small defect that could be rapidly detected and fixed. Indeed, Mitchell et al. (1977) show that ignoring such correlations may lead to inaccuracies in the system's performance measures.

We embed the BCVG-MIP into a tandem queue simulation model using the Stochastic Simulation in Java (SSJ) library (L'Ecuyer et al. 2002), with two M/M/1 queues of infinite buffer space between the stations. Also, we assume a Poisson arrival rate $\lambda = 1$ and exponential service times with mean $1/\mu = 0.75$ at each station. For validation purposes, we use the expected time to clear the system, W, and the expected delay in the second queue, $W_{q,2}$, under correlated service times given by a Pearson correlation ρ. To integrate the proposed model with the simulation, we generate correlated service times in batches of N observations to be consumed by the simulation. Once these vectors are consumed, a new batch of service times is generated by the simulation model and sent to BCVG-MIP to induce the target correlation structure. To do so, we slightly modify BCVG-MIP to consider that at the k-th iteration of this procedure, $M = (k-1)N$ observations (history) have already been used by the simulation model and are now fixed. Thus, the only degrees of freedom to induce the target correlation are given by the new batch of N pairs of observations.

In this particular example, we use BCVG-MIP to correlate batches of size $N = 200$ and stop the simulation after a run of 100, 000 time units. Figure 2a shows that BCVG-MIP quickly achieves the target correlation structure for $\rho = -0.50$, -0.25, 0, $+0.50$, and $+0.95$. However, for $\rho = -0.95$ our model achieves a correlation of $\widehat{\rho} = -0.640$. This large gap is explained by the fact that the theoretical minimal (negative) correlation that can be induced in a pair of exponential random variables is given by $\rho = 1 - \pi^2/6 = -0.645$ which is close to the achieved correlation (Hill and Reilly 1994). Notably, without explicitly knowing the statistical properties of the induced correlations, BCVG-MIP achieves the minimal (negative) possible correlation.

Figure 2b, c shows the convergence to the system performance measures $W(M)$ and $W_{q,2}(M)$ once M customers have left the system. Our results validate those

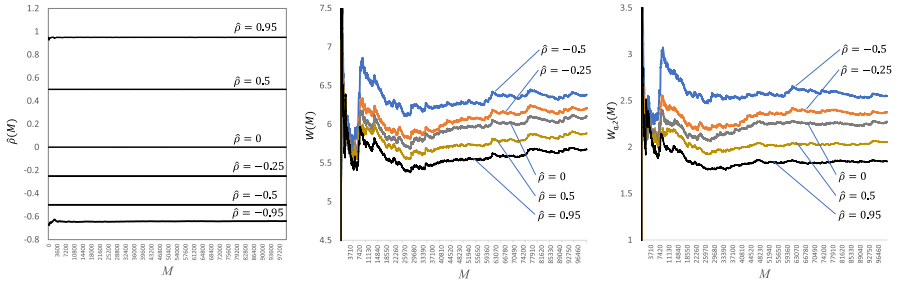

Fig. 2 Convergence of ρ, W, and $W_{q,2}$

obtained by Mitchell et al. (1977), where positive correlations in service times improve system performance. We also compare against the analytical performance metrics for a tandem of queues with infinite buffer space without correlated service times. In such case, the analytical results of $W = 6$ and $W_{q,2} = 2.25$ (Gross and Harris 1985) were closely approximated by our simulation model (inducing $\rho = 0$) which achieved $W(M) = 6.101$ and $W_{q,2}(M) = 2.272$ when $M = 100166$.

5.3 Correlated missing-data imputation

This section describes how to use our methodology to induce bivariate correlations in missing-data imputation analysis. Consider two vectors in $\mathbb{R}^n$, $\mathbf{x}$ and $\mathbf{y}$, containing samples of random variables X and Y, respectively. Although these vectors may have been collected through direct observation or experiments, we assume that they have missing data. We define I_X and I_Y as the sets of indices corresponding to samples for which one variate is available, but the other one is missing. That is, $i \in I_X$ means that the x-value in sample (x_i, y_i) is available, but the y-value is missing. Similarly, $i \in I_Y$ means that the y-value in sample (x_i, y_i) is available, but the x-value is missing. Additionally, we use set I_{XY} to denote samples missing both x- and y-values. Instead of deleting them, we preserve samples indexed in I_{XY} as they may have complete information for other variables different from X and Y. Further, we denote by I the set of indices corresponding to the complete samples (x_i, y_i) for which no data are missing.

We assume that there is a set of data points to replace the missing values for each variable. These values can be found using the empirical probability distribution of each variable, learning or regression methods based on other variables, or any other method (see Little and Rubin (2019) for a review of existing techniques). To represent the new data, we use index sets $\tilde{I}_X$ and $\tilde{I}_Y$, where $|\tilde{I}_X| = |I_Y \cup I_{XY}|$ and $|\tilde{I}_Y| = |I_X \cup I_{XY}|$. This means that there is enough new data to fill in the missing values in samples whose indices are in I_X with values indexed in $\tilde{I}_Y$ as well as samples whose indices are in I_Y with observations indexed in $\tilde{I}_X$. The remaining values indexed in $\tilde{I}_X$ and $\tilde{I}_Y$ are used to fill in samples indexed in I_{XY}. Note that index sets $\tilde{I}_X$ and $\tilde{I}_Y$ contain the new data to fill in all missing values, including those in I_{XY}. Figure 3a shows this construction, where the solid black nodes indicate missing values. Note that it is not necessarily true that $|\tilde{I}_X| = |\tilde{I}_Y|$, as there may be more missing values in one variable than the other.

Although our approach can handle any target Pearson correlation, the goal in this case is to fill in the missing x- and y-values using the new observations while preserving the *observed* Pearson correlation from the available data. Using the construction in Fig. 3a, we create the assignment graph shown in Fig. 3b, where the set of arcs is given by $\mathcal{A}' = \{(i,j) : i \in I_X, j \in \tilde{I}_Y \quad \vee \quad i \in \tilde{I}_X, j \in I_Y \quad \vee \quad i \in \tilde{I}_X, j \in \tilde{I}_Y\}$. Intuitively, this set of arcs consist of all the choices to complete the missing values, including those samples missing both x- and y-values. Note that there are no arcs related to samples indexed in I because they are complete and input data cannot be modified.

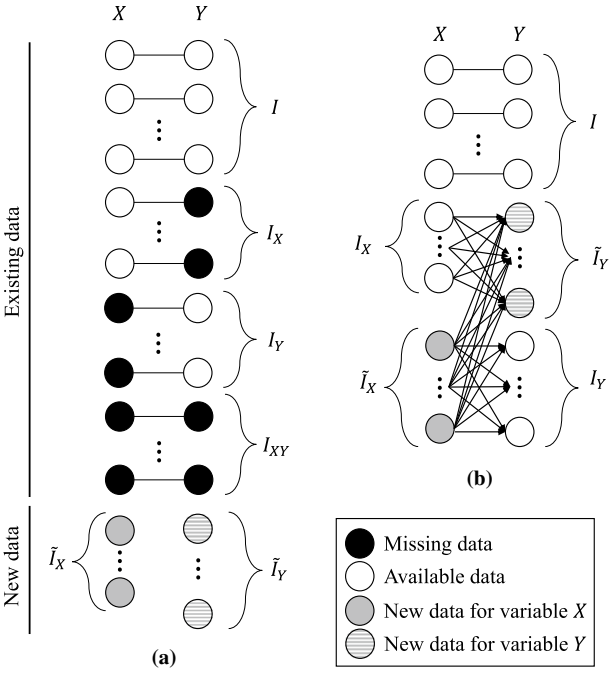

Fig. 3 **a** Existing and new data indices. **b** Assignment graph for correlated missing-data imputation

The set of z-variables in BCVG-MIP consists of z_{ij} for $(i,j) \in \mathcal{A}'$, which allows us to rewrite (4b) in BCVG-RMP as

$$\sum_{(i,j)\in\mathcal{A}'} \mathbf{a}_{ij} z_{ij} + \tilde{\mathbf{e}}_1 \delta^- - \tilde{\mathbf{e}}_1 \delta^+ = \mathbf{b}',$$

where $\mathbf{b}'$ is a $\{2(n - |I|) + 1\}$-dimensional vector of ones but with the first position equal to $\bar{\rho}^P_{XY} - \sum_{i\in I} c_{ii}/S_{XY}$ to account for the *partially achieved* correlation produced by the samples indexed in I—i.e., achieved by the data that cannot be modified by our algorithm. In this case, the procedure to compute the reduced cost for non-basic z-variables in (5) only includes pairs $(i,j) \in \mathcal{A}'$.

We illustrate this approach on a case study that presents data on standardized tests to evaluate the effects of parental psychological disorders on children's development (p.228, example 11.1, Little and Rubin 2019). Data collected include verbal and reading comprehension scores in a standardized test for the first and second child of the sampled families and among three risk groups: control, moderate-risk, and high-risk. All sampled families have two children, and their risk level is chosen based on the severity of the psychological disorder of the parents, if any. Table 3 presents the data obtained for the second child of 27 families in the moderate-risk group. Observations marked by "—" denote missing values. In this case, $I = \{1, \ldots, 17\}$, $I_X = \{18, \ldots, 23\}$, $I_Y = \{24\}$, and $I_{XY} = \{25, 26, 27\}$.

Table 3 Correlated missing-data imputation example

Family	Verbal	Reading
1	140	93
2	113	96
3	108	98
4	120	101
5	128	105
6	133	105
7	150	109
8	155	110
9	125	114
10	140	115
11	148	116
12	123	118
13	158	126
14	118	104
15	85	87
16	140	130
17	185	139
18	100	–
19	110	–
20	130	–
21	135	–
22	150	–
23	63	–
24	–	126
25	–	–
26	–	–
27	–	–

The Pearson correlation between the verbal and reading comprehension scores using available information in I is given by $\bar{\rho}^P_{XY} = 0.77$.

Using the estimates available in Little and Rubin (2019), we generate new data to fill in the missing scores using normal distributions with parameters $\mu_X = 128.57$ and $\sigma_X = 25.90$ for verbal and $\mu_Y = 110.67$ and $\sigma_Y = 13.75$ for reading comprehension. These μ- and σ-values are estimated using the mean and standard deviations of the available information. After sampling from these distributions and rounding to the nearest integer, we obtain new verbal scores given by 156, 90, 129, and 145, and new reading comprehension scores given by 93, 117, 117, 121, 127, 127, 110, 108, and 106. With these values, we construct the coefficients for each decision variable z_{ij} in (3b), which are given by $c_{ij} \equiv (x_i - \bar{x})(y_j - \bar{y})$ for $(i,j) \in \mathcal{A}'$ and where the sample averages $\bar{x}$ and $\bar{y}$ and S_{XY} are calculated using both existing and new observations for each variable.

Table 4 shows the results of our approach for the correlated missing-data example. In this case, the achieved correlation (including all samples) is equal to 0.68. This correlation is the closest possible to the observed sample correlation before imputing new data. Note that if sample averages are used to replace the missing information, the achieved correlation would be 0.54.This illustrates the well-documented importance of inducing a relevant correlation in addition to finding imputation values (Moorthy et al. 2014).

We emphasize that our method can be paired with any other imputation method to generate candidate values as the *raw* data are never changed but only reorganized. Moreover, other realistic extensions such as limiting the possible samples where a new value can be imputed, fixing the sample where a value must go, or inducing an autocorrelation within the same variable (Abdella and Marwala 2005) can be easily incorporated in our method by modifying the set of arcs in the assignment graph (as in Fig. 3b).

5.4 Estimating the budget overrun risk in a construction project

To illustrate the use of our approach for multivariate correlations, we use the case study presented in Touran (1993), where the budget overrun risk of a construction project is determined using a Monte Carlo simulation. The goal is to avoid underestimating the overrun risk by incorporating the correlations among cost components, as raw material prices depend on the same economic conditions and typically fluctuate together. We define X_1, X_2, and X_3 as the random variables representing the total cost (per square foot) of electrical systems, mechanical systems, and moisture protection, respectively. Following Touran (1993), we assume that X_1, X_2, and X_3 are log-normally distributed with parameters $\mu_{X_1} = 5.14$ and $\sigma_{X_1} = 2.76$ for X_1, $\mu_{X_2} = 9.47$ and $\sigma_{X_2} = 6.58$ for X_2, and $\mu_{X_3} = 1.81$ and $\sigma_{X_3} = 2.12$ for X_3. We want to induce the target Pearson correlation coefficients $\bar{\rho}_{X_1,X_2} = 0.8$ and $\bar{\rho}_{X_1,X_3} = \bar{\rho}_{X_2,X_3} = 0.45$, which capture moderate (=0.45) and strong (=0.8) correlations between cost components.

We conduct 50 random experiments to compare the performance of the column generation (CG) approach in Algorithm 2 versus the IC method. Each experiment consists of generating 100 samples for each log-normal variable, which we then

Table 4 Correlated missing-data imputation results

Family	Verbal	Reading
18	100	**108**
19	110	**110**
20	130	**121**
21	135	**127**
22	150	**127**
23	63	**106**
24	**156**	126
25	**90**	**93**
26	**129**	**117**
27	**145**	**117**

Bold values represent the new generated data

correlate using each method. The box plots in Fig. 4 present the results, including lines at the lower quartile, median, and upper quartile values for the mean and maximum absolute deviations from the target correlation (using only the upper triangular portion of the correlation matrix) across the 50 experiments. The plots for CG are significantly narrower and have smaller absolute deviations than those for the IC method. On average, the deviation from the target correlation for the IC method is close to 0.4, whereas it is less than 0.15 for the CG approach. Moreover, the average maximum deviations produced by IC and CG are 0.6 and 0.2, respectively. These results show that the CG method provides more accurate correlations compared to IC, which is critical when making decisions in systems featuring random variability and interdependencies.

6 Computational experiments

In this section, we test the computational limits of our approach by generating random instances of up to 3000 observations. In each experiment, we generate the random samples independently. We use Java (version 1.8.0_151) to code our algorithms and Gurobi (version 8.1.0) to solve the optimization problems on a machine with an Intel i7-4610M processor @3.00GHz and 16GB of RAM. All the reported solution times for the column generation approaches include the time taken by IC, which is less than one second for any of the instances considered. This is expected as IC runs in polynomial time. We use the package MC2D in `R` to run the Iman and Conover method (Pouillot and Delignette-Muller 2010). Further, we set a time limit of 1000

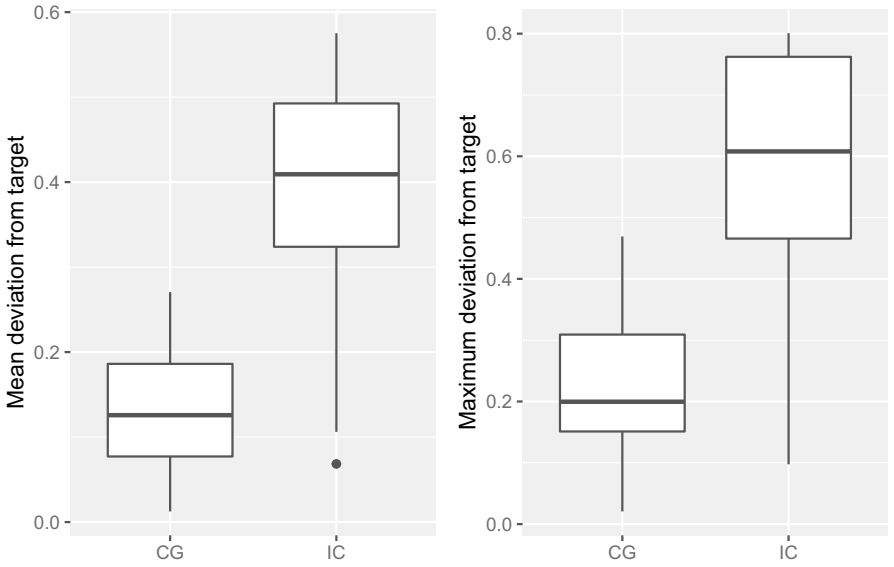

Fig. 4 Box plots of the mean and maximum absolute difference between the achieved and target correlations for three log-normal random vectors and 100 samples

s for all the bivariate experiments and 3600 s to solve each iteration of the loop in Lines 2–14 in Algorithm 2, where the most computationally intensive operation is to solve the MIP problem in Line 12.

6.1 Bivariate correlations

We generate random values from exponential ($\lambda = 1/2$), normal ($\mu = 10, \sigma = 3$), and uniform [0, 1] distributions, with sample sizes of 500, 1000, 2000, and 3000 observations. For each pair of distributions and sample size, we induce values of $\overline{\rho}^P_{XY}$ equal to −0.8, −0.5, 0, 0.5, and 0.8.

Table 5 compares the average absolute deviations from the target correlation between Iman–Conover (IC), BCVG-MIP (MIP), and our column generation algorithm (CG) approach (i.e., Algorithm 1) over 10 replications. In all the experiments, the average absolute deviation is no more than 0.052 for our method, and it is very close to 0 for BCVG-MIP in every instance. This result indicates that in most instances the target correlation is achievable with the given samples. The maximum average absolute deviation is 0.171 for IC when inducing a correlation of −0.5 between the two exponential distributions with 500 samples. The performance of IC drastically deteriorates as the target correlations become more negative for combinations of variables involving the exponential distribution. This is because the use of Spearman correlation relies on the ranking of each observation rather than on its value, which seems to be more sensitive when inducing extreme correlations on asymmetric distributions. On the contrary, the CG method exhibits a more consistent performance across distributions and target correlations. Note that we omit results for $\overline{\rho}^P_{XY} = -0.8$ when correlating two exponential variables, as this is theoretically not possible to achieve (Conway 1979). Because of the large sample size, BCVG-MIP times out for almost all instances of 3000 samples except for problems inducing a correlation coefficient of 0, where almost all instances are solved to optimality. In such cases, the MIP produces an average optimal solution with an absolute deviation of 0.

Table 6 shows that in no case was the MIP faster than the CG approach. Indeed, the MIP almost always timed out for instances with 3000 samples, whereas the CG approach solved every problem within the time limit. Problems inducing a correlation of 0 are easier to solve given that the initial set of columns in the CG approach contains the IC solution as well as $n - 1$ columns reflecting the data pairings produced by the random number generator. Because the samples are independently generated, the CG approach has an initial solution of very good quality. This feature may also explain the performance of the MIP when solving instances of 0 target correlation, as a feasible solution is readily available. For the same correlation level, the CG method improves the IC solution in almost every case after generating only a few columns (see Table A.1 in the Supplementary Material for more details on the CG performance). In both approaches, CG and MIP, instances are more difficult to solve as the target correlation becomes more extreme regardless of its sign.

Table 5 Average absolute deviation from target correlation in the bivariate case

Sample size	$\bar{\rho}$	exp-exp			norm-norm			unif-unif			exp-norm			exp-unif			norm-unif		
		IC	MIP	CG	IC	MIP	CG	IC	MIP	CG	IC	MIP	CG	IC	MIP	CG	IC	MIP	CG
500	− 0.8				0.035	0.000	0.000	0.007	0.000	0.000	0.133	0.000	0.000	0.133	0.000	0.000	0.018	0.000	0.000
	− 0.5	0.171	0.000	0.000	0.041	0.000	0.000	0.069	0.000	0.000	0.122	0.001	0.012	0.164	0.000	0.002	0.026	0.000	0.000
	0.0	0.106	0.001	0.043	0.051	0.001	0.032	0.055	0.002	0.043	0.087	0.001	0.035	0.094	0.002	0.045	0.046	0.001	0.041
	0.5	0.052	0.002	0.052	0.039	0.001	0.039	0.014	0.001	0.014	0.021	0.001	0.021	0.022	0.000	0.000	0.046	0.002	0.046
	0.8	0.038	0.000	0.000	0.008	0.000	0.000	0.013	0.002	0.013	0.064	0.000	0.000	0.084	0.000	0.000	0.003	0.001	0.002
1000	− 0.8				0.023	0.000	0.000	0.002	0.001	0.001	0.107	0.000	0.000	0.108	0.000	0.000	0.008	0.000	0.000
	− 0.5	0.127	0.000	0.000	0.003	0.000	0.002	0.021	0.000	0.000	0.057	0.000	0.000	0.091	0.000	0.000	0.011	0.001	0.011
	0.0	0.002	0.000	0.002	0.006	0.000	0.005	0.002	0.000	0.002	0.007	0.000	0.006	0.010	0.000	0.008	0.005	0.001	0.005
	0.5	0.041	0.000	0.000	0.006	0.001	0.005	0.025	0.000	0.000	0.043	0.000	0.000	0.080	0.000	0.000	0.010	0.001	0.010
	0.8	0.077	0.000	0.000	0.017	0.000	0.000	0.003	0.001	0.003	0.092	0.000	0.000	0.102	0.000	0.000	0.006	0.000	0.000
2000	− 0.8				0.024	0.000	0.000	0.000	0.000	0.000	0.102	0.000	0.000	0.105	0.000	0.000	0.010	0.000	0.000
	− 0.5	0.126	0.000	0.000	0.002	0.000	0.001	0.022	0.000	0.000	0.047	0.000	0.000	0.078	0.000	0.000	0.010	0.001	0.010
	0.0	0.002	0.000	0.002	0.017	0.000	0.016	0.007	0.000	0.007	0.004	0.000	0.004	0.003	0.000	0.003	0.015	0.001	0.011
	0.5	0.059	0.000	0.000	0.008	0.000	0.008	0.020	0.000	0.000	0.056	0.000	0.000	0.088	0.000	0.000	0.014	0.001	0.014
	0.8	0.103	0.000	0.000	0.018	0.000	0.000	0.004	0.000	0.004	0.105	0.000	0.000	0.107	0.000	0.000	0.006	0.000	0.000
3000	− 0.8				0.022	–	0.000	0.000	–	0.000	0.107	–	0.000	0.109	–	0.000	0.011	–	0.000
	− 0.5	0.133	–	0.000	0.007	–	0.000	0.038	–	0.000	0.062	–	0.000	0.096	–	0.001	0.001	–	0.001
	0.0*	0.007	0.000 (8)	0.005	0.009	0.000 (10)	0.009	0.017	0.000 (10)	0.016	0.007	0.000 (10)	0.006	0.013	0.000 (10)	0.012	0.012	0.000 (9)	0.011
	0.5	0.041	–	0.000	0.014	–	0.014	0.010	–	0.000	0.043	–	0.000	0.074	–	0.000	0.024	–	0.024
	0.8	0.095	–	0.000	0.018	–	0.000	0.005	–	0.005	0.102	–	0.000	0.105	–	0.000	0.005	–	0.000

*For sample size 3000 and $\bar{\rho} = 0$, numbers in parenthesis represent the number of instances solved to optimality by the MIP

–, No instance solved to optimality

Table 6 Average computational time in seconds for the bivariate case

Sample size	$\bar{\rho}$	exp-exp		norm-norm		unif-unif		exp-norm		exp-unif		norm-unif	
		MIP	CG	MIP	CG	MIP	CG	MIP	CG	MIP	CG	MIP	CG
500	− 0.8			13.57	2.05	11.78	0.83	14.60	1.95	15.93	2.61	13.38	1.62
	− 0.5	7.07	1.62	9.86	0.44	10.67	0.76	9.53	0.90	10.38	0.86	10.01	0.28
	0.0	4.49	0.05	6.67	0.03	11.00	0.04	7.86	0.04	6.20	0.05	8.94	0.04
	0.5	5.26	0.04	9.17	0.03	9.38	0.03	8.90	0.04	9.71	0.31	9.75	0.04
	0.8	6.82	1.87	12.53	0.75	12.45	0.04	16.69	1.92	17.12	2.37	12.51	0.09
1000	− 0.8			136.59	11.85	154.33	0.09	170.15	14.11	178.55	17.29	153.93	4.90
	− 0.5	70.87	8.43	88.06	0.16	89.83	1.56	98.03	3.68	90.88	5.90	109.32	0.11
	0.0	24.95	0.11	47.90	0.13	49.85	0.13	48.93	0.13	46.63	0.14	54.58	0.13
	0.5	40.40	2.54	89.93	0.14	91.03	1.79	104.32	3.23	103.06	5.50	110.17	0.12
	0.8	75.63	10.32	145.81	9.59	124.89	0.09	179.28	14.02	190.83	18.07	154.32	4.11
2000	− 0.8			562.33	83.22	713.07	2.21	675.02	83.89	802.34	107.46	668.62	39.62
	− 0.5	537.87	52.51	354.43	0.92	411.69	9.52	349.42	20.90	380.40	32.63	404.52	0.33
	0.0	142.83	0.37	135.59	0.33	173.38	0.42	134.22	0.34	121.30	0.41	157.40	0.37
	0.5	297.60	19.15	348.94	0.31	447.80	10.87	346.68	26.94	374.64	36.70	413.48	0.29
	0.8	516.91	61.21	556.81	71.08	674.69	0.34	690.57	87.15	813.46	107.88	675.99	27.65
3000	− 0.8			TL	231.36	TL	3.16	TL	281.12	TL	333.10	TL	135.02
	− 0.5	TL	174.62	TL	10.65	TL	44.19	TL	84.39	TL	113.19	TL	1.30
	0.0*	727.77 (8)	0.78	487.79 (10)	0.62	418.91 (10)	0.72	531.64 (10)	0.63	621.75 (10)	0.64	786.44 (9)	0.66
	0.5	TL	42.51	TL	0.65	TL	14.45	TL	59.71	TL	91.87	TL	0.63
	0.8	TL	198.90	TL	198.79	TL	0.62	TL	273.49	TL	325.90	TL	63.50

*For sample size 3000 and $\bar{\rho} = 0$, numbers in parenthesis represent the number of instances solved to optimality by the MIP

TL: All instances reached the time limit

Table 7 shows that overall, the column generation procedure resulted in a computational speedup of at least 3× and up to 2000× faster than the MIP. These mixed results show that the instance difficulty depends on the combination of distributions, target correlation, and sample size. In general, we observe that the MIP's solution time increases as the target correlation becomes extreme (i.e., moves closer to +1 or −1). This is also the case for the column generation (CG) approach. The speedup tends to be larger for those instances inducing a target correlation of 0 given that the initialization of the CG method is an advantage over the MIP. For some distributions, however, the speedup is larger when inducing extreme correlations (e.g., uniform-uniform). Tables 5, 6, and 7 demonstrate the advantages of the CG approach in terms of the quality of the solution, which can be achieved at very reasonable computational time for some correlation levels, distributions, and sample sizes.

Table 7 Average speedup of CG versus MIP

Sample size	$\bar{\rho}$	Distributions					
		exp-exp	norm-norm	unif-unif	exp-norm	exp-unif	norm-unif
500	− 0.8		6.70	32.14	7.60	6.46	8.81
	− 0.5	4.60	26.60	14.61	11.19	12.63	39.65
	0.0	101.25	207.27	289.81	209.41	167.79	231.61
	0.5	154.75	304.55	285.50	268.00	47.41	263.91
	0.8	3.64	25.39	377.43	8.63	7.50	290.64
1000	− 0.8		11.76	1727.82	12.23	10.36	35.16
	− 0.5	8.58	712.51	60.11	29.85	16.51	1012.82
	0.0	240.65	419.00	399.68	389.88	378.11	447.11
	0.5	17.97	825.82	57.70	38.87	19.22	980.36
	0.8	7.39	15.78	1361.84	13.05	10.73	52.56
2000	− 0.8		6.80	1220.16	8.17	7.58	17.61
	− 0.5	10.57	695.65	44.07	17.50	11.85	1243.75
	0.0	402.83	431.52	421.11	415.26	312.00	446.82
	0.5	15.77	1138.34	44.57	13.05	10.34	1484.24
	0.8	8.46	7.89	2016.09	7.96	7.56	27.39
3000	− 0.8		–	–	–	–	–
	− 0.5	–	–	–	–	–	–
	0.0*	1007.26 (8)	803.45 (10)	589.09 (10)	848.46 (10)	1025.26 (10)	1241.90 (9)
	0.5	–	–	–	–	–	–
	0.8	–	–	–	–	–	–

*For sample size 3000 and $\bar{\rho} = 0$, numbers in parenthesis represent the number of instances solved to optimality by the MIP

–, No instance solved to optimality

6.2 Multivariate correlations

We use Algorithm 2 to induce multivariate correlations among 5 and 10 random variables from various distributions and sample sizes. We generate random values using 10 different distributions, including normal (X_1 with $\mu = 5$ and $\sigma = 3$; and X_6 with $\mu = 8$ and $\sigma = 16$), exponential (X_2 and X_7 with $\lambda = 4$ and $\lambda = 10$, respectively), uniform (X_3 and X_8 both in the interval [0, 1]), log-normal (X_4 with $\mu = 5.14$ and $\sigma = 2.76$; and X_9 with $\mu = 9.74$ and $\sigma = 6.58$) and gamma (X_5 with shape= 8 and scale = 5; and X_{10} with shape= 10 and scale= 7). For the experiments with 5 variables, we use $X_1, \ldots, X_5$, and for those with 10 random variables, we use $X_1, \ldots, X_{10}$. Similar to the bivariate case, we use sample sizes of 500, 1000, 2000, and 3000 observations. We generate five instances of random samples for each combination of number of variables and sample size. Our goal is to induce target Pearson correlation matrices with entries of different intensities and signs, given by

$$\bar{\rho}^P = \begin{pmatrix} 1 & 0.8 & 0.45 & 0.2 & -0.3 \\ & 1 & 0.45 & 0.5 & -0.1 \\ & & 1 & 0.1 & -0.3 \\ & & & & -0.5 \\ & & & & 1 \end{pmatrix}$$

and

$$\bar{\rho}^P = \begin{pmatrix} 1 & -0.68 & 0.93 & 0.82 & -0.58 & 0.73 & 0.65 & 0.67 & 0.64 & -0.05 \\ & 1 & -0.76 & -0.72 & 0.34 & -0.55 & -0.43 & -0.24 & -0.65 & 0.5 \\ & & 1 & 0.9 & -0.49 & 0.77 & 0.66 & 0.62 & 0.75 & -0.13 \\ & & & 1 & -0.24 & 0.69 & 0.64 & 0.45 & 0.7 & -0.29 \\ & & & & 1 & -0.45 & -0.26 & -0.63 & -0.2 & -0.28 \\ & & & & & 1 & 0.66 & 0.68 & 0.66 & 0.23 \\ & & & & & & 1 & 0.29 & 0.64 & 0.03 \\ & & & & & & & 1 & 0.29 & 0.41 \\ & & & & & & & & 1 & -0.07 \\ & & & & & & & & & 1 \end{pmatrix},$$

where the column order is given by $X_1, \ldots, X_5$ and $X_1, \ldots, X_{10}$, respectively.

Table 8 shows the performance of the CG approach (Algorithm 2) versus the IC method when inducing a multivariate correlation structure in 5- and 10-variable instances. We report two complementary performance metrics for each method: the average and the maximum absolute deviations from the target. For each instance, we calculate the average absolute deviation across target correlation coefficients in the upper triangular portion of the correlation matrix, excluding the diagonal. This is because the matrix is symmetric and the correlations in the diagonal are trivially achieved. Table 8 reports the average of these values across instances in the column "Avg. abs. deviation". Because calculating the average absolute deviation may not provide information on pairwise target correlations, we also report the maximum absolute deviation observed across instances

Table 8 Performance of CG versus IC in the multivariate case

Variables	Sample size	Avg. abs. deviation			Max. abs. deviation			Time (s)	
		CG	IC	IC/CG	CG	IC	IC/CG	CG	IC
5	500	0.002	0.056	30.60	0.012	0.141	11.64	7.95	< 1
	1000	0.003	0.044	17.12	0.006	0.153	25.06	52.52	< 1
	2000	0.006	0.038	6.59	0.015	0.116	7.93	434.85	< 1
	3000	0.009	0.039	4.25	0.024	0.130	5.43	559.12	< 1
10	500	0.006	0.056	9.33	0.129	0.251	1.94	1362.63	< 1
	1000	0.007	0.046	6.57	0.116	0.219	1.88	1496.13	< 1
	2000	0.011	0.047	4.27	0.108	0.225	2.09	2821.34	< 1
	3000	0.014	0.047	3.36	0.114	0.222	1.94	6262.47	< 1

in the column "Max. abs. deviation". We report the solution time for each method in seconds. In this case, we omit the comparison with MCVG-MIP($k, \bar{\mathcal{K}}$) as its performance is not competitive as $\bar{\mathcal{K}}$ increases in size, even for five variables and 500 samples.

Table 8 shows that the CG approach achieves correlations that are closer to the targets compared to the IC method. The average absolute deviations for IC are up to 30.6 times greater than those in CG for five variables, and up to 9.3 times for 10 variables. Notably, the relative improvement of CG with respect to IC decreases as more samples are used, with the largest improvement seen for 500 samples. The maximum deviation from the target correlation matrix shows a similar pattern. The IC method induces a correlation structure whose maximum deviation is almost twice as large as that induced by the CG approach in any instance. For five variables and 1000 samples, the maximum deviation in the IC method is more than 25 times that observed in the CG method. The improvement is moderate for 10 variables, with the IC method producing maximum deviations that are at least 1.88 times larger than those in the CG approach. Along the same lines, we observe that the largest deviations in the IC method consistently occur when dealing with at least one asymmetrical distribution. This is the case in 85% of the experiments with 5 and 10 variables (34 out of 40 experiments), where the maximum deviations occur when attempting to correlate exponential and normal (16 experiments) and exponential and log-normal (18 experiments) distributions. A similar pattern is observed in the CG approach, where 80% of the maximum deviations are related to correlations involving the exponential distribution. However, the CG emerges as a valid and effective alternative to reduce the maximum deviation compared to IC as shown in Table 8.

In general, the CG approach provides a more accurate correlation structure but at the expense of significantly longer solution times with respect to the IC method. Given its integer programming component, the solution times of the CG approach quickly deteriorate as more samples and variables are used. In critical applications, where precision is of utmost importance for the decision maker, the CG method may become a valuable alternative.

7 Concluding remarks and future research

This paper proposes a new approach based on mixed-integer programming (MIP) to induce different types of correlation structures to bivariate and multivariate random vectors. Even though it shares the same principle of the Iman–Conover (IC) method, our method is able to induce target correlations more accurately than IC. With simple adjustments, our MIP method is able to target a wide range of correlation structures, including the Spearman rank, Pearson correlation, relative risk, Phi correlation coefficient, and Kendall's coefficient of concordance. We propose a column generation procedure to improve the scalability and computational performance of the solution approach without drastically compromising the quality of the solution. Although the IC method is fast as it runs in polynomial time, throughout this paper we show that it can induce correlations that are far from the targets. This is more evident for some combinations of target correlations, sample size, and distributions. We empirically found that IC leads to large errors when inducing correlations on asymmetrical distributions (e.g., negative correlations between exponential distributions in Table 5, maximum deviations in Table 8), when using a small sample size for some combinations of distributions (e.g., 100 log-normal samples in Sect. 5.4 and Table 8), as the number of variables increases (e.g., 10-variable case in Table 8). In general, the proposed models based on mathematical programming provide more accurate correlations, but at the expense of longer solution times.

Following the results of Sect. 5.2, a future task is to implement the proposed CG approaches as a callable library so that it could be possible to generate correlated random vectors on-the-go within existing simulation environments. This strategy is rooted in the advantages of the CG approach, as it is fast and accurate when correlating small batches of data.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00291-021-00620-5.

Acknowledgements The authors would like to thank Professor Jim Wilson from NC State University for sharing his encouraging and valuable input at an earlier stage of this work. The authors sincerely thank Professor Douglas Montgomery at Arizona State University for his valuable comments to improve the manuscript. Also, authors thank Gurobi and FICO for providing access to their commercial optimization solvers under their academic licensing programs. The authors would like to thank the two anonymous reviewers, whose comments greatly improved the article. This material is based upon work supported by Dr. Sefair's National Science Foundation Grant No. 1740042.

References

Abdella M, Marwala T (2005) The use of genetic algorithms and neural networks to approximate missing data in database. In: IEEE 3rd international conference on computational cybernetics, 2005 (ICCC 2005). IEEE, pp 207–212

Altiok T, Melamed B (2001) The case for modeling correlation in manufacturing systems. IIE Trans 33(9):779–791

Batista G, Monard MC (2003) An analysis of four missing data treatment methods for supervised learning. Appl Artif Intell 17(5–6):519–533

Biswas A (2004) Generating correlated ordinal categorical random samples. Stat Probab Lett 70(1):25–35
Cahen EJ, Mandjes M, Zwart B (2018) Estimating large delay probabilities in two correlated queues. ACM Trans Model Comput Simul 28(1):2
Cario MC, Nelson BL (1997) Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix. Technical report, Department of Industrial Engineering and Management Science, Northwestern University
Chakraborty A (2006) Generating multivariate correlated samples. Comput Stat 21(1):103–119
Charmpis DC, Panteli PL (2004) A heuristic approach for the generation of multivariate random samples with specified marginal distributions and correlation matrix. Comput Stat 19(2):283
Clark DE, El-Taha M (1998) Generation of correlated logistic-normal random variates for medical decision trees. Methods Inf Med 37(03):235–238
Conway DA (1979) Multivariate distributions with specified marginals. Technical report no. 145, Stanford University. https://statistics.stanford.edu/sites/default/files/OLK%20NSF%20145.pdf
Cornfield J (1951) A method of estimating comparative rates from clinical data: applications to cancer of the lung, breast, and cervix. J Natl Cancer Inst 11(6):1269–1275
Corredor D, Cabrera N, Medaglia AL, Akhavan-Tabatabaei R (2020) Data-driven approach for the shortest α-reliable path problem. COPA working paper
Dai YS, Xie M, Poh KL, Ng SH (2004) A model for correlated failures in n-version programming. IIE Trans 36(12):1183–1192
Deb R, Liew AW-C (2016) Missing value imputation for the analysis of incomplete traffic accident data. Inf Sci 339:274–289
Desaulniers G, Desrosiers J, Solomon MM (2006) Column generation, vol 5. Springer, New York
Dias CTDS, Samaranayaka A, Manly B (2008) On the use of correlated beta random variables with animal population modeling. Ecol Model 215:293–300
Ghosh S, Henderson SG (2003) Behavior of the Norta method for correlated random vector generation as the dimension increases. ACM Trans Model Comput Simul 13(3):276–294
Gross D, Harris CM (1985) Fundamentals of queueing theory. Wiley, New York
Haas CN (1999) On modeling correlated random variables in risk assessment. Risk Anal 19(6):1205–1214
Harris CM, Hoffman KL, Yarrow L- (1995a) Obtaining minimum-correlation Latin hypercube sampling plans using an ip-based heuristic. OR Spektrum 17(2–3):139–148
Harris CM, Hoffman KL, Yarrow L-A (1995b) Using integer programming techniques for the solution of an experimental design problem. Ann Oper Res 58(3):243–260
Hill RR, Reilly CH (1994) Composition for multivariate random variables. In: Proceedings of winter simulation conference. IEEE, pp 332–339
Hill RR, Reilly CH (2000) The effects of coefficient correlation structure in two-dimensional knapsack problems on solution procedure performance. Manag Sci 46(2):302–317
Iman RL, Conover W-J (1982) A distribution-free approach to inducing rank correlation among input variables. Commun Stat Simul Comput 11(3):311–334
Kendall MG, Babington-Smith B (1939) The problem of m rankings. Ann Math Stat 10(3):275–287
Kolev N, Paiva D (2008) Random sums of exchangeable variables and actuarial applications. Insur Math Econ 42(1):147–153
Law AM, Kelton WD (2000) Simulation modeling and analysis, 3rd edn. Mc Graw-Hill, New York
L'Ecuyer P, Meliani L, Vaucher J (2002) Ssj: a framework for stochastic simulation in java. In: Proceedings of the (2002) winter simulation conference. IEEE, Piscataway, NJ, pp 234–242
Legendre P (2005) Species associations: the Kendall coefficient of concordance revisited. J Agric Biol Environ Stat 10(2):226–245
Leschied JR, Mazza MB, Davenport MS, Chong ST, Smith EA, Hoff CN, Ladino-Torres MF, Khalatbari S, Ehrlich PF, Dillman JR (2016) Inter-radiologist agreement for CT scoring of pediatric splenic injuries and effect on an established clinical practice guideline. Pediatr Radiol 46(2):229–236
Levitin G, Xie M (2006) Performance distribution of a fault-tolerant system in the presence of failure correlation. IIE Trans 38(6):499–509
Li ST, Hammond JL (1975) Generation of pseudorandom numbers with specified univariate distributions and correlation coefficients. IEEE Trans Syst Man Cybern 5:557–561
Little RJA, Rubin DB (2019) Statistical analysis with missing data, vol 793. Wiley, New York
Lübbecke ME, Desrosiers J (2005) Selected topics in column generation. Oper Res 53(6):1007–1023
Lurie PM, Goldberg MS (1998) An approximate method for sampling correlated random variables from partially-specified distributions. Manag Sci 44(2):203–218

Medaglia AL, Sefair JA (2009) Generating correlated random vectors using mixed-integer programming. In: Proceedings of the IIE annual conference. Institute of Industrial and Systems Engineers (IISE), 1759

Mildenhall SJ (2005) Correlation and aggregate loss distributions with an emphasis on the Iman–Conover method. http://www.casact.org/pubs/forum/06wforum/06w105.pdf. Part one of "The Report of the Research Working Party on Correlations and Dependencies Among All Risk Sources." Casualty Actuarial Society Forum (Winter 2005)

Mitchell CR, Paulson AS, Beswick CA (1977) Effect of correlated exponential service times on single server tandem queues. Naval Res Logist 24(1):95–112

Moorthy K, Saberi Mohamad M, Deris S (2014) A review on missing value imputation algorithms for microarray gene expression data. Curr Bioinform 9(1):18–22

Morris JA, Gardner MJ (1988) Calculating confidence intervals for relative risk (odds ratios) and standardised ratios and rates. Br Med J 296(6632):1313–1316

Nasr WW, Maddah B (2015) Continuous (s, S) policy with MMPP correlated demand. Eur J Oper Res 246(3):874–885

Oracle (2019) Oracle(@) Crystal Ball reference and examples guide. http://www.hstoday.us/. Release 11.1.2.4, Accessed July 2019

Park CG, Dong WS (1998) An algorithm for generating correlated random variables in a class of infinitely divisible distributions. J Stat Comput Simul 61(1–2):127–139

Park CG, Park T, Shin DW (1996) A simple method for generating correlated binary variates. Am Stat 50(4):306–310

Patuwo BE, Disney RL, McNickle DC (1993) The effect of correlated arrivals on queues. IIE Trans 25(3):105–110

Polge RJ, Holliday EM, Bhagavan BK (1973) Generation of a pseudo-random set with desired correlation and probability distribution. Simulation 20(5):153–158

Pouillot R, Delignette-Muller M-L (2010) Evaluating variability and uncertainty in microbial quantitative risk assessment using two R packages. Int J Food Microbiol 142(3):330–40

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

Reilly CH (2009) Synthetic optimization problem generation: show us the correlations! INFORMS J Comput 21(3):458–467

Rosenfeld S (2008) Approximate bivariate gamma generator with prespecified correlation and different marginal shapes. ACM Trans Model Comput Simul 18(4):16

Schmeiser BW, Lal R (1982) Bivariate gamma random vectors. Oper Res 30(2):355–374

Sefair JA, Méndez CY, Babat O, Medaglia AL, Zuluaga LF (2017) Linear solution schemes for mean-semivariance project portfolio selection problems: an application in the oil and gas industry. Omega 68:39–48

Sheskin DJ (2000) Handbook of parametric and nonparametric statistical procedures, 3rd edn. Chapman and Hall-CRC, Boca Raton

Shin K, Pasupathy R (2010) An algorithm for fast generation of bivariate Poisson random vectors. INFORMS J Comput 22(1):81–92

Shults J (2017) Simulating longer vectors of correlated binary random variables via multinomial sampling. Comput Stat Data Anal 114:1–11

Sigler EA, Tallent-Runnels MK (2006) Examining the validity of scores from an instrument designed to measure metacognition of problem solving. J Gener Psychol 133(2):257–276

Stanfield PM, Wilson JR, King RE (2004) Flexible modelling of correlated operation times with application in product-reuse facilities. Int J Prod Res 42(11):2179–2196

Todd CR, Ng MP (2001) Generating unbiased correlated random survival rates for stochastic population models. Ecol Model 144(1):1–11

Touran A (1993) Probabilistic cost estimating with subjective correlations. J Constr Eng Manag 119(1):58–71

Touran A, Suphot L (1997) Rank correlations in simulating construction cost. J Constr Eng Manag 123(3):297–301

Toyoda Yoshiaki (1975) A simplified algorithm for obtaining approximate solutions to zero-one programming problems. Manag Sci 21(12):1417–1427. https://doi.org/10.1287/mnsc.21.12.1417

Van der Geest PAG (1998) An algorithm to generate samples of multi-variate distributions with correlated marginals. Comput Stat Data Anal 27(3):271–289

Wallis WA (1939) The correlation ratio for ranked data. J Am Stat Assoc 34(207):533–538

Xiao Q (2017) Generating correlated random vector involving discrete variables. Commun Stat Theory Methods 46(4):1594–1605

Yan C, Kung J (2016) Robust aircraft routing. Transp Sci 52(1):118–133

Young DJ, Beaulieu NC (2000) The generation of correlated Rayleigh random variates by inverse discrete Fourier transform. IEEE Trans Commun 48(7):1114–1127

Zhang Yufeng, Khani Alireza (2019) An algorithm for reliable shortest path problem with travel time correlations. Transp Res Part B Methodol 121:92–113. https://doi.org/10.1016/j.trb.2018.12.011