Les Cahiers du GERAD ISSN: 0711–2440

A PCA-based approximation scheme for combinatorial optimization with uncertain and correlated data

R. Borzou, G. Desaulniers,
F. Errico, A. Lodi

G–2018–61

August 2018

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2018
– Bibliothèque et Archives Canada, 2018

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2018
– Library and Archives Canada, 2018

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

A PCA-based approximation scheme for combinatorial optimization with uncertain and correlated data

Borzou Rostami [a,b,c]

Guy Desaulniers [a,d]

Fausto Errico [a,b,e]

Andrea Lodi [a,c]

[a] *GERAD, Montréal (Québec), Canada, H3T 2A7*

[b] *CIRRELT, Montréal (Québec), Canada, H3T 1J4*

[c] *CERC-Data Science for Real-time Decision-making, Polytechnique Montréal, Montréal (Québec) Canada, H3C 3A7*

[d] *Department of Mathematics and Industrial Engineering, Polytechnique Montréal (Québec) Canada, H3C 3A7*

[e] *Departement of Civil Engineering, École de Technologie Supérieure de Montréal, Montréal (Québec) Canada, H3C 1K3*

bo.rostami@gmail.com
guy.desaulniers@gerad.ca
fausto.errico@etsmtl.ca
andrea.lodi@polymtl.ca

August 2018
Les Cahiers du GERAD
G–2018–61

Abstract: This paper addresses combinatorial optimization problems under uncertain and correlated data where the mean-covariance information of the random data is assumed to be known. Such a problem can be modeled as an integer non-linear program. We reformulate the problem using spectral decomposition and apply principal component analysis (PCA) to approximate the reformulation. The quality of the resulting approximate solutions is assessed by determining a worst-case optimality gap. We apply our results to the capacitated vehicle routing problem with uncertain and statistically correlated travel times (CVRP-SCT). The CVRP-SCT seeks vehicle routes whose observed travel times are not excessively dispersed with respect to their expected value. To solve the approximate models to optimality, we develop exact branch-price-and-cut algorithms. Our experimental results on a rich collection of instances show that good quality feasible solutions can be found using the proposed approximate models. In particular, solving the PCA-based model, even with a very small number of components, provides solutions that are optimal for all instances with known optimal values in less computational times than an exact method.

Keywords: Data uncertainty, correlation, convex binary quadratic program, principal component analysis, spectral decomposition, vehicle routing, branch-price-and-cut

Acknowledgments: The authors gratefully acknowledge the financial support of the Natural Science and Engineering Research Council of Canada through grants RGPIN-2017-05683, RGPIN-2015-06289 and the Canada Excellence Research Chair in Data science for real-time decision-making, as well as the Fonds de recherche du Québec - Nature et technologie, through the grant Établissement de nouveaux chercheurs universitaires - 198928.

1 Introduction

Decision-making problems are plagued with uncertain data which affect price, demand, reliability, etc. Identifying the sources of uncertainty, understanding how they contribute to decisions, and the management of uncertainties within the decision-making process are therefore essential. Nowadays impressive capabilities in terms of data collection and analysis enable us to describe the stochastic behavior of many random processes. Decision-making activities have the potential to exploit this information to provide better and more robust operations planning.

In many planning problems, because of some common trigger factors (e.g., weather, congestion, festivals), individual events are likely to be correlated, rather than statistically independent. For example, in passenger and freight transportation, if the costs are proportional to fuel consumption or travel times, then events such as rush hours, road works, and accidents can lead to road correlated costs, since congestion on one road is likely to cause congestion on nearby roads. Other examples include the portfolio selection problem, in which the risk-averse investor has to take into account the correlations among multiple risky assets as well as their individual performances, and the stochastic facility location problem, in which the supplier needs to consider the correlations among demands from different retailers. Given the fact that correlation between the uncertain data makes the decision-making problems computationally challenging, assuming the independence of random variables is very common. This creates a considerable gap between research and reality in situations where significant correlations among the uncertain data parameters may occur (see, for instance, Agrawal et al. 2012).

Incorporating uncertainty in decision-planning leads to the challenging issue of how to model the uncertainty in a proper way such that, on the one hand, it reflects the real-world concerns and, on the other hand, be computationally tractable. There are different solution approaches for dealing with uncertainty. One approach that has been widely used is stochastic programming (see, for instance, Birge and Louveaux 2011, and references therein), where the decision maker optimizes the expected value of an objective function that involves random parameters. This approach is typically applied to the case where uncertainty can be described by known distributions. One alternative approach is robust optimization (RO) (see, e.g., Ben-Tal and Nemirovski 1998, Buchheim and Kurtz 2017) that handles cases where such probability distributions are hard to justify or estimate. Distributionally robust stochastic programming (DRSP) (see, for instance, Delage and Ye 2010, Wiesemann et al. 2014) is an intermediate approach that minimizes the expected cost over the worst joint distribution among all probability distributions consistent with the available information.

Another relevant track of research on decision-making under uncertainty is to develop methodologies to construct the uncertainty sets based on past observations of the random variables and the concept of risk. The "risk" is broadly defined as a quantitative expression of a system of attitudes, or preferences with respect to a set of random outcomes. In utility-based models, the risk attitude determines the curvature of the utility function, thereby influencing the evaluation process of risky choice alternatives. Value-at-Risk (VaR) and the Markowitz mean-variance framework proposed by Markowitz (1952) in the context of portfolio are two popular approaches to address the risk. By definition with respect to a specified probability level β, the β-VaR of a portfolio is the lowest amount α such that, with probability β, the loss will not exceed α. The Markowitz mean-variance framework identifies risk with the volatility (variance) of the random outcome of the decision. This approach evaluates a decision under uncertainties in terms of trade-off between its risk and reward. The computational tractability of this approach, along with its intuitively appealing interpretation, have contributed to widespread its popularity in finance and economics, as well as in operations research, management science, and engineering (see, for instance, Steinbach 2001, and references therein).

The purpose of this paper is to provide approximate models for combinatorial optimization problems under uncertain and correlated data. To model the uncertainty, we use the mean-covariance information about the distributions underlying the random data. Let us consider a cost vector ξ as a random variable from the class $\mathcal{M}^m_{(\mu,\mathcal{C})}$ of $m-$variate distributions with mean μ and covariance $\mathcal{C}$. More precisely, for $i \in \{1,\ldots,m\}$, let ξ_i be a random variable representing the random cost of decision i with mean μ_i and standard deviation σ_i. Moreover, let ρ_{ij} represent the correlation coefficient between ξ_i and ξ_j for $i,j \in \{1,\ldots,m\}$. The entries of $\mathcal{C}$ for each pair of $i,j \in \{1,\ldots,m\}$ are given by $\mathcal{C}_{ij} = \rho_{ij}\sigma_i\sigma_j$ where the principal diagonal

elements represent the variance of random variables ξ_i, $i \in \{1, \ldots, m\}$. Note that for a given data matrix $T \in \mathbb{R}^{p \times m}$, where each column gives the observations of ξ_i, the sample covariance matrix can be computed as $\mathcal{C} = \frac{1}{p}[(T-\mu)(T-\mu)^T] \in \mathbb{R}^{m \times m}$.

Let $\mathbb{B} = \{0, 1\}$, $A \in \mathbb{R}^{n \times m}$ be a real matrix, and $b \in \mathbb{R}^n$ a real vector. For some $0 \leq \alpha < 1$ and $q \in \{1, 2\}$, we consider the following parametric non-linear combinatorial optimization problem:

$$\begin{aligned} \text{PNLCOP:} \quad \min \quad & (1-\alpha)\,\mu^T x + \alpha\,(x^T \mathcal{C} x)^{1/q} \\ \text{s.t.} \quad & x \in \mathcal{F}_x \cap \mathbb{B}^m, \end{aligned} \tag{1}$$

with

$$\mathcal{F}_x = \{x \in [0,1]^m \mid Ax = b\}. \tag{2}$$

Since the covariance matrix $\mathcal{C}$ is positive semidefinite, PNLCOP is a convex binary quadratic program for $q = 1$ and a second-order cone program with binary requirements for $q = 2$. In the first case, the non-linear term of the objective function represents the variance, while in the second case, it denotes the standard deviation.

By defining a new parameter $\delta = \alpha/(1-\alpha)$ as a "risk factor," we can rewrite the objective function as the conic combination of mean and dispersion (variance or standard deviation), which is known as a risk-averse objective function in the literature (see, for instance, Björk et al. 2014, for more details).

1.1 Relationship to other models

Although PNLCOP can be viewed as a mean-dispersion model that evaluates the combinatorial optimization problem under uncertainty in terms of a trade-off between the risk and the cost, it can be shown that different RO, DRSP, and VaR models can be reformulated as PNLCOP.

In particular, PNLCOP with $q = 2$ and $\delta = \alpha/(1-\alpha)$ turns out to be equivalent (see, e.g., Bertsimas and Sim 2004, Buchheim and De Santis 2018) to the following RO problem

$$\begin{aligned} \min \quad & \max_{\xi \in U} \quad \xi^T x \\ \text{s.t.} \quad & x \in \mathcal{F}_x \cap \mathbb{B}^m, \end{aligned} \tag{3}$$

where the worst-case is taken over the uncertainty set U, which is an ellipsoid defined by the mean μ and covariance $\mathcal{C}$ of ξ.

Moreover, when the distribution of random variables ξ is Gaussian, problem PNLCOP with $q = 2$ is equivalent (see, e.g., Ghaoui et al. 2003) to the following VaR problem

$$\begin{aligned} \min \quad & t \\ \text{s.t.} \quad & \boldsymbol{Prob}(\xi^T x \geq t) < \epsilon \\ & x \in \mathcal{F}_x \cap \mathbb{B}^m, \end{aligned} \tag{4}$$

where the VaR is defined as the minimal level t such that the probability that the cost $\xi^T x$ exceeds t is less than ϵ. Note that, in general, the VaR problem, in contrast to PNLCOP, which only requires the knowledge of the first and second moments of the distribution of ξ, assumes that the entire distribution is known.

Furthermore, solving the continuous relaxation of the PNLCOP with $q = 1$ is equivalent (see, for instance, Geoffrion 1967, Popescu 2007) to solving the continuous relaxation of the following DRSP problem

$$\begin{aligned} \min \quad & \max_{\xi} \quad \mathbb{E}[f(\xi^T x)] \\ \text{s.t.} \quad & x \in \mathcal{F}_x \cap \mathbb{B}^m, \end{aligned} \tag{5}$$

where f is an increasing and concave function (i.e., the decision maker is risk averse) with one- or two-point support.

1.2 Scientific contributions

Since the covariance matrix $\mathcal{C}$ is positive semidefinite, the continuous relaxation of PNLCOP is either a convex quadratic program or a second-order cone program, which can be solved by state-of-the-art continuous optimization solvers. However, it is generally very difficult to solve the PNLCOP when it models even small-sized real-life contexts. In fact, not only it is NP-hard, but it has been shown that even some particular cases, such as the shortest path problem with a convex quadratic objective function, are APX-hard (see Rostami et al. 2018a). For an overview of solution methods for the general binary quadratic program, we refer the interested reader to Rostami et al. (2018b) and references therein.

In this paper, we show how PNLCOP beyond providing optimal solutions can be suitably exploited in an approximate solution framework. Our main contributions are summarized as follows:

- We show how the covariance matrix $\mathcal{C}$ can be exploited to reformulate PNLCOP in the projected space of eigenvectors of $\mathcal{C}$. Then, we apply the principal component analysis (PCA) to obtain a family of approximations of the problem and analyze the quality of the approximate solutions in terms of a worst-case optimality gap.
- To demonstrate the effectiveness of the proposed methodology, we apply our results to the capacitated vehicle routing problem with stochastic and correlated travel times (CVRP-SCT, see Rostami et al. 2017), which is known to be a very hard combinatorial optimization problem. We develop a branch-price-and-cut algorithm to solve the approximate formulations. A PCA-based approximation yields a column generation master problem with a quadratic objective function, while the subproblems are classical elementary shortest path problems with resource constraints (ESPPRCs).
- We perform an extensive computational study on known datasets from the literature. Our results indicate the effectiveness of the proposed models with a large number of customers. The PCA-based approximate solutions turn out to be identical to the optimal solutions of the original problem on all instances with known optimal values.
- Results also confirm the benefits of the proposed models in handling variability. Indeed, the obtained approximate solutions display an expected cost that is only marginally higher than that obtained by deterministic models, but with significant less dispersion. Finally, the approximate models provide solutions with different values of expected cost and associated risk, which in turn allows the decision maker (planner) to choose solutions according to her risk attitude.

1.3 Organization of the paper

The remainder of the paper is organized as follows. In Section 2, we detail the PCA-based approximate models and analyze the solution quality. In Section 3, we present the CVRP-SCT and provide the proposed branch-price-and-cut algorithm. In Section 4, we present the data, the experimental settings, the implementation details and the computational results. Conclusions are drawn in Section 5.

2 PCA-based reformulation and approximation

In this section, we present an approximation scheme for PNLCOP. We show how to use the *principal component analysis* technique to find a low-rank approximation of the covariance matrix $\mathcal{C}$ in such a way that the resulting low-rank matrix provides a good approximation of $\mathcal{C}$. Moreover, we analyze the behavior of the algorithm in the worst case.

The PCA is one of the oldest and most widely used techniques to reduce the dimensionality of a dataset while preserving as much variability as possible (see, for instance, Abdi and Williams 2010, Jolliffe and Cadima 2016). The main idea of PCA is to choose a new coordinate system for the dataset by transforming some (possibly) correlated variables into a smaller number of uncorrelated variables, called principal components that are linear functions of the original ones. The greatest variance by any projection of the dataset comes to lie on the first axis (the first principal component), the second greatest variance on the second axis, and so on. If the number of components retained is equal to the number of original variables, all variability will be

accounted for. However, the goal is to keep as few components as possible to explain the substantive amount of variance.

Consider a data matrix T and let T_j represents its jth column. The goal is to find a transformation (direction) $\nu \in \mathbb{R}^m$ such that $\nu^T T = \sum_{j=1}^{m} \nu_j T_j$ captures as much variance of T as possible. Note that the variance of $\nu^T T$ is computed as follows:

$$var(\nu^T T) = \mathbb{E}[(\nu^T (T-\mu))^2] = \mathbb{E}[\nu^T (T-\mu)(T-\mu)^T \nu] = \nu^T \mathcal{C} \nu. \tag{6}$$

It can be shown (see, for instance, Abdi and Williams 2010) that the best transformation is obtained by solving the equation

$$\mathcal{C}\,\nu = \lambda\,\nu \qquad\qquad \text{for some real number } \lambda.$$

Hence, ν must be an eigenvector and λ the corresponding eigenvalue of the covariance matrix $\mathcal{C}$. Therefore, every PCA axis has an eigenvalue associated with it, which is proportional to the amount of variation explained by that axis.

2.1 A reformulation

We present how to exploit the covariance matrix $\mathcal{C}$ to reformulate PNLCOP in the projected space of eigenvectors. Because eigenvectors corresponding to different eigenvalues of $\mathcal{C}$ are orthogonal, it is possible to store all the eigenvectors in an orthogonal matrix (recall that a matrix is orthogonal when the product of this matrix by its transpose is a diagonal matrix). Using spectral decomposition (see, for instance, Golub and Van Loan 2012), we can express matrix $\mathcal{C}$ as

$$\mathcal{C} = V\,\Lambda\,V^T \tag{7}$$

where columns of V represent the normalized eigenvectors $v_1, \ldots, v_m$ of $\mathcal{C}$, and Λ is a diagonal matrix such that the diagonal elements are the eigenvalues $\lambda_1, \ldots, \lambda_m$.

The right-hand side of (7) can be replaced by $U^T U$, where $U \in \mathbb{R}^{r \times m}$ and r is the rank of $\mathcal{C}$. Note that the rank of a matrix can be defined as being the number of non-zero eigenvalues of the matrix. Therefore, if we define $Ux = y$, with $y \in \mathbb{R}^{r \times 1}$, we can rewrite the PNLCOP as

$$\begin{aligned} \text{RPNLCOP:} \quad & \min \quad (1-\alpha)\,\mu^T x + \alpha\,(y^T y)^{1/q} \\ & \text{s.t.} \quad (x,y) \in \mathcal{F}_{xy} \cap (\mathbb{B}^m \times \mathbb{R}^r), \end{aligned} \tag{8}$$

with

$$\mathcal{F}_{xy} = \{(x,y) |\ x \in \mathcal{F}_x,\ Ux = y\}. \tag{9}$$

Note that, if $r \ll m$, model RPNLCOP might be easier to solve than PNLCOP because the number of non-zero terms in the objective function considerably decreases at the cost of introducing only r continuous variables and r linear constraints. However, in general, solving RPNLCOP can be as difficult as PNLCOP.

2.2 An approximate approach

Given the fact that the covariance matrix is positive semidefinite, its eigenvalues are always positive or null, and its eigenvectors are pairwise orthogonal when their eigenvalues are different. The eigenvectors are also composed of real values. Without loss of generality we assume that $\lambda_1 \geq \lambda_2, \geq \ldots \geq \lambda_m$ with corresponding eigenvectors $v_1, v_2, \ldots, v_m$. Our aim, in this section, is to construct an approximation of RPNLCOP in an ℓ-dimensional subspace ($\ell \leq r$) that captures as much variance as possible. The following proposition is a known result providing indications on how to suitably chose directions with higher variance.

Proposition 1 *Let $\mathcal{C}$ be a covariance matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$ and corresponding eigenvectors $v_1, v_2, \ldots, v_m$. The direction ν of maximum variance is v_1 and that of minimum variance is v_m.*

Proof. The results ensue using the Rayleigh quotient $R(\nu) = \frac{\nu^T \mathcal{C} \nu}{\nu^T \nu}$ (see, for instance, Golub and Van Loan 2012), the definition of $\nu^T \mathcal{C} \nu$ provided in (6), and the fact that

$$\max_{\|\nu\|=1} R(\nu) = \max_{\nu \in \mathbb{R}^m \setminus \{0\}} R(\nu) = \lambda_1 \tag{10}$$

$$\min_{\|\nu\|=1} R(\nu) = \min_{\nu \in \mathbb{R}^m \setminus \{0\}} R(\nu) = \lambda_m. \tag{11}$$

□

As a consequence of Proposition 1, we can project RPNLCOP to a subspace spanned by the top ℓ eigenvectors $v_1, v_2, \ldots, v_\ell$. Let $\bar{U}^\ell \in \mathbb{R}^{\ell \times m}$ be the submatrix of U corresponding to the top eigenvectors $v_1, v_2, \ldots, v_\ell$ and define $U^\ell = \left[\bar{U}^\ell \mid 0\right]^T$ with $0 \in \mathbb{R}^{(r-\ell)\times m}$ as a null matrix. We consider the following approximate model:

$$\begin{aligned} \text{RPNLCOP}^\ell: \quad w^\ell = \min \quad & (1-\alpha)\, \mu^T x + \alpha\, (y^T y)^{1/q} \\ \text{s.t.} \quad & (x, y) \in \mathcal{F}^\ell_{xy} \cap (\mathbb{B}^m \times \mathbb{R}^r), \end{aligned} \tag{12}$$

with

$$\mathcal{F}^\ell_{xy} = \{(x, y) |\ x \in \mathcal{F}_x,\ U^\ell x = y\}. \tag{13}$$

Note that, the dimension of y is still r. However, equation $U^\ell x = y$ and the definition of U^ℓ force $y_{\ell+1} = \ldots, y_r = 0$. The following proposition formally shows that solving RPNLCOP$^\ell$ gives a valid lower bound for RPNLCOP.

Proposition 2 *Let $(\hat{x}, \hat{y})$ and $(\bar{x}, \bar{y})$ be optimal solutions of RPNLCOP$^\ell$ and RPNLCOP with objective values of w^ℓ and w, respectively. Then, $w^\ell \leq w$.*

Proof. Feasibility of $(\bar{x}, \bar{y})$ for RPNLCOP implies that $(\bar{x}, \tilde{y})$ with $\tilde{y} = (\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_\ell, 0, \ldots, 0)$ is feasible for RPNLCOP$^\ell$. Since $(\hat{x}, \hat{y})$ is an optimal solution of RPNLCOP$^\ell$, we have

$$\begin{aligned} w^\ell =& (1-\alpha)\, \mu^T \hat{x} + \alpha\, (\hat{y}^T\, \hat{y})^{1/q} \leq (1-\alpha)\, \mu^T \bar{x} + \alpha\, (\tilde{y}^T\, \tilde{y})^{1/q} \\ \leq& (1-\alpha)\, \mu^T \bar{x} + \alpha\, (\bar{y}^T\, \bar{y})^{1/q} = w. \end{aligned}$$

□

Corollary 1 *$w^\ell \leq w^{\ell+1}$ for all $\ell \in \{1, \ldots, r-1\}$.*

One of the main advantages of solving RPNLCOP$^\ell$ is that it includes the original set of constraints $\mathcal{F}_x$. Thus, any feasible solution to RPNLCOP$^\ell$ is also feasible for PNLCOP. Once a solution to RPNLCOP$^\ell$ is found, one can also evaluate the value of this solution for the objective function of PNLCOP to obtain an upper bound to the problem, thereby obtaining an optimality gap.

2.3 Worst-case bound analysis

Given an optimal solution of RPNLCOP$^\ell$, we would like to assess the quality of this solution in terms of the objective function of PNLCOP with respect to the optimal value of PNLCOP. To this end, let us consider the covariance matrix $\mathcal{C}$. Using the decomposition $\mathcal{C} = U^T U$ with $U \in \mathbb{R}^{r \times m}$ as described in Section 2.1 and the definition of U^ℓ given in Section 2.2, we can express the covariance matrix $\mathcal{C}$ as $\mathcal{C} = \mathcal{C}^\ell + \mathcal{C}^{r-\ell}$, where

$$\mathcal{C}^\ell \in \mathbb{R}^{m \times m} = (U^\ell)^T U^\ell \quad \text{and} \quad \mathcal{C}^{r-\ell} \in \mathbb{R}^{m \times m} = (U^{r-\ell})^T U^{r-\ell}. \tag{14}$$

In other words, we partition the covariance matrix with respect to the top ℓ eigenvectors $v_1, v_2, \ldots, v_\ell$ and the remaining $r - \ell$ eigenvectors. Note that we do not consider the eigenvectors corresponding to zero eigenvalues. In the following theorem, we analyze the quality of the optimal solution of RPNLCOP$^\ell$.

Theorem 1 *Let $(\hat{x}, \hat{y})$ and $(\bar{x}, \bar{y})$ be optimal solutions of RPNLCOP$^\ell$ and RPNLCOP, respectively. Moreover, let $val(\hat{x}, \hat{y})$ and $val(\bar{x}, \bar{y})$ represent the values of the objective function of RPNLCOP at $(\hat{x}, \hat{y})$ and $(\bar{x}, \bar{y})$, respectively. Then*

$$val(\hat{x}, \hat{y}) \leq (1 + \varepsilon)\, val(\bar{x}, \bar{y}),$$

where $\varepsilon = \alpha \frac{(m\,\lambda_{\ell+1})^{1/q}}{w^\ell}$ and $\lambda_{\ell+1}$ is the largest eigenvalue of $\mathcal{C}^{r-\ell}$.

Proof. Let $q = 1$. Then,

$$\begin{aligned} val(\hat{x}, \hat{y}) &= (1-\alpha)\,\mu^T\hat{x} + \alpha\,\hat{y}^T\hat{y} = (1-\alpha)\,\mu^T\hat{x} + \alpha\,\hat{x}^T\mathcal{C}\hat{x} \\ &= (1-\alpha)\,\mu^T\hat{x} + \alpha\,\hat{x}^T\mathcal{C}^\ell\hat{x} + \alpha\,\hat{x}^T\mathcal{C}^{r-\ell}\hat{x} \\ &= w^\ell + \alpha\,\hat{x}^T\mathcal{C}^{r-\ell}\hat{x}. \end{aligned}$$

Diving both sides of the above equality by w^ℓ we obtain

$$\frac{val(\hat{x}, \hat{y})}{w^\ell} = 1 + \frac{\alpha\,\hat{x}^T\mathcal{C}^{r-\ell}\hat{x}}{w^\ell}. \tag{15}$$

Using the Rayleigh quotient (see Equations (10) and (11)) and the fact that $\hat{x}^T\hat{x} \leq m$, we obtain

$$\hat{x}^T\mathcal{C}^{r-\ell}\hat{x} \leq \lambda_{\ell+1}\,\hat{x}^T\hat{x} \leq m\lambda_{\ell+1}. \tag{16}$$

According to (15) and (16), and the fact that

$$\frac{val(\hat{x}, \hat{y})}{val(\bar{x}, \bar{y})} \leq \frac{val(\hat{x}, \hat{y})}{w^\ell}$$

the result of the theorem for $q = 1$ follows.

For $q = 2$, we have

$$\begin{aligned} val(\hat{x}, \hat{y}) &= (1-\alpha)\,\mu^T\hat{x} + \alpha\,(\hat{y}^T\hat{y})^{1/2} = (1-\alpha)\,\mu^T\hat{x} + \alpha\,(\hat{x}^T\mathcal{C}\hat{x})^{1/2} \\ &\leq (1-\alpha)\,\mu^T\hat{x} + \alpha\,(\hat{x}^T\mathcal{C}^\ell\hat{x})^{1/2} + \alpha\,(\hat{x}^T\mathcal{C}^{r-\ell}\hat{x})^{1/2} \\ &= w^\ell + \alpha\,(\hat{x}^T\mathcal{C}^{r-\ell}\hat{x})^{1/2}, \end{aligned}$$

where the inequality follows from the fact that $\mathcal{C}^\ell$ and $\mathcal{C}^{r-\ell}$ are positive semidefinite and that $(a+b)^{1/2} \leq a^{1/2} + b^{1/2}$ for $a, b \geq 0$. The rest of the proof is the same as for $q = 1$. □

Note that one can provide a tighter bound than the one proposed in Theorem 1 if the specific problem structure allows us to infer a bound on the value of $\hat{x}^T\hat{x}$ (see, for instance, Section 3.1).

Corollary 2 *Let $lost(\hat{x}) = \hat{x}^T\mathcal{C}^{r-\ell}\hat{x}$ be a function representing the lost dispersion of solution $\hat{x}$. Then,*

$$(m\,\lambda_m)^{1/q} \leq lost(\hat{x}) \leq (m\,\lambda_{\ell+1})^{1/q}.$$

This shows that for small $\lambda_{\ell+1}$, the approximate covariance matrix $\mathcal{C}^\ell$ provides a very good approximation of the original covariance matrix $\mathcal{C}$.

3 Application to the CVRP-SCT

The purpose of this section and the next one is to demonstrate the practical applicability and impact of the proposed PCA-based approximation scheme. To this end, we choose the CVRP-SCT which is of interest for a large community of researchers because accounting for stochasticity can potentially improve the quality of the solutions. While we show in this section how to apply our method to the CVRP-SCT, the computational study is developed in the next section.

The classical CVRP is known to be a very hard combinatorial optimization problem and has been extensively studied in the literature (see, for instance, Baldacci et al. 2012, Toth and Vigo 2014). The CVRP-SCT is a variant of the CVRP where travel times between locations are assumed to be stochastic and statistically correlated. Consequently, the total time of each potential vehicle route, which is the sum of the travel times between each pair of locations visited consecutively is also stochastic and can be affected by a large variability. Taking into account the correlation between the travel times, the objective of the CVRP-SCT consists of planning optimal vehicle routes whose travel times are reliable, in the sense that observed travel times are not excessively dispersed with respect to their expected value.

Let us describe the CVRP-SCT using the set partitioning formulation proposed in Rostami et al. (2017). Let K be a set of identical vehicles with capacity Q and $G = (V, E)$ be a graph with vertex set $V = \{0, 1, 2, \ldots, n\}$ and arc set E, with $|E| = m$. Each vertex $i \in V_c = V \setminus \{0\}$ represents a customer having a nonnegative demand q_i, while vertex 0 corresponds to a depot. In E, there exists an arc (i, j) linking vertex i to vertex j if the sum of the demands at these vertices does not exceed Q. Let $[t_{ij}]_{(i,j)\in E}$ be a vector of random variables (representing the random travel times) from the class $\mathcal{M}^m_{(\mu,\mathcal{C})}$ of $m-$variate distributions with mean μ and covariance $\mathcal{C}$.

The CVRP-SCT consists of finding a set of $|K|$ routes, where both the total expected travel times and the total variance are minimized in such a way that each customer is visited exactly once, each route starts and ends at the depot, and the vehicle capacity is respected for each route. Let $\mathcal{R}^k$ be the set of feasible routes for vehicle $k \in K$. For each $k \in K$, let $\mu^k \in \mathbb{R}^{|\mathcal{R}^k|}$ be the vector of expected travel times for routes in $\mathcal{R}^k$, i.e., $\mu^k_p = \sum_{(i,j)\in p} \mu_{ij}$ for each route $p \in \mathcal{R}^k$. Furthermore, let $A^k \in \mathbb{B}^{|V_c|\times|\mathcal{R}^k|}$ be an incident matrix such that for each customer $i \in V_c$, the entry a^k_{ip} takes value 1 if route $p \in \mathcal{R}^k$ visits customer i and 0 otherwise. In the same way, we also define an incident matrix $B^k \in \mathbb{B}^{m\times|\mathcal{R}^k|}$, where each of its entries b^k_{ijp}, $(i, j) \in E$, $p \in \mathcal{R}^k$, takes value 1 if route p traverses arc (i, j), and 0, otherwise.

For each vehicle $k \in K$, we define two sets of binary variables $z^k = (z^k_1, z^k_2, \ldots, z^k_{|\mathcal{R}^k|})^T$ and $x^k = (x^k_1, x^k_2, \ldots, x^k_m)^T$. In the former, each binary variable z^k_p is equal to 1 if vehicle k uses route p and 0 otherwise, while in the latter, each binary variable x^k_h is equal to 1 if arc $h = (i, j)$ is traversed by vehicle k, and 0 otherwise. The CVRP-SCT is formulated as the following convex binary quadratic mean-variance model:

$$\begin{aligned} \mathrm{P}_{xz}: \quad \min \quad & (1-\alpha) \sum_{k\in K} (\mu^k)^T z^k + \alpha \sum_{k\in K} (x^k)^T \mathcal{C} x^k \\ \text{s.t.} \quad & (x, z) \in \mathcal{F}_{xz}, \end{aligned} \tag{17}$$

with

$$\begin{aligned} \mathcal{F}_{xz} = \Big\{ (x, z) \mid \quad & \\ & \textstyle\sum_{k\in K} A^k_i z^k = 1 && \forall\, i \in V_c && (18)\\ & B^k z^k = x^k && \forall\, k \in K && (19)\\ & (e^k)^T z^k = 1 && \forall\, k \in K && (20)\\ & z^k \in \{0, 1\}^{|\mathcal{R}^k|} && \forall\, k \in K && (21)\\ & x^k \in \{0, 1\}^{|E|} && \forall\, k \in K \Big\}, \end{aligned}$$

where A^k_i is the ith row of matrix A^k and e^k is a vector of ones. For a given value of α ($0 \le \alpha < 1$), the first term of the objective function (17) minimizes the total expected travel time, while the second term

minimizes the total variance. Constraints (18) ensure that each customer is visited once. Constraints (19) link the original arc-flow variables x to the route variables z. Constraints (20) impose to select only one route for each vehicle.

Note that P_{xz} contains a huge number of variables, one binary variable z_p^k for each feasible route $p \in \mathcal{R}^k$, $k \in K$ and $|K| \times |E|$ binary variables x. To solve the problem to optimality, Rostami et al. (2017) develop a branch-price-and-cut algorithm in which the continuous relaxation of P_{xz} is solved by column generation at each node of the search tree.

3.1 Approximate models

Following the PCA approximation described in Section 2.2, we define for each $k \in K$, $Ux^k = y^k$ with $y^k \in \mathbb{R}^{r\times 1}$ and project out variables x^k from P_{xz} to obtain the following reformulation:

$$\begin{aligned} \mathrm{P}_{yz}: \qquad w = \min \quad & (1-\alpha) \sum_{k\in K} \mu^k z^k + \alpha \sum_{k\in K} (y^k)^T y^k \\ \text{s.t.} \quad & (y,z) \in \mathcal{H}_{yz}, \end{aligned} \tag{22}$$

with

$$\mathcal{H}_{yz} = \Big\{(y,z) \mid (18),(20),(21), \quad \text{and} \quad U B^k z^k = y^k \quad \forall k \in K\Big\}.$$

As a consequence of Proposition 1, if we project P_{yz} to a subspace spanned by the top ℓ eigenvectors $v_1, v_2, \ldots, v_\ell$ of $\mathcal{C}$, we obtain the following approximate model:

$$\begin{aligned} \mathrm{P}^\ell_{yz}: \qquad w^\ell = \min \quad & (1-\alpha) \sum_{k\in K} \mu^k z^k + \alpha \sum_{k\in K} (y^k)^T y^k \\ \text{s.t.} \quad & (y,z) \in \mathcal{H}^\ell_{yz}, \end{aligned} \tag{23}$$

with

$$\begin{aligned} \mathcal{H}^\ell_{yz} = \Big\{(y,z) \mid (18),(20),(21), \quad & \text{and} \\ & U^\ell B^k z^k = y^k \quad \forall k \in K\Big\}. \end{aligned} \tag{24}$$

In the following proposition, we show how to obtain a tighter worst-case optimality gap for the solution provided by P^ℓ_{yz}.

Proposition 3 *The value of ε in Theorem 1 for $q = 1$ is given by $\varepsilon = \alpha \frac{\lambda_{\ell+1}(|K|+n)}{w^\ell}$.*

Proof. In the definition of ε in Theorem 1, m is an upper bound for $\hat{x}^T\hat{x}$. Here, we show how to exploit the problem's structure to obtain the exact value of $\hat{x}^T\hat{x}$. We have

$$\hat{x}^T\hat{x} = \sum_{k\in K} (\hat{x}^k)^T \hat{x}^k = \sum_{k\in K} \big((\hat{x}^k_{e_1})^2 + \ldots + (\hat{x}^k_{e_m})^2\big),$$

where $e_1, \ldots, e_m$ represent the m arcs in G. Since routes are elementary, only one arc leaves each customer, but $|K|$ arcs leave the depot. Therefore,

$$\hat{x}^T\hat{x} = |K| + n,$$

and this completes the proof. □

3.2 A branch-price-and-cut algorithm

In this section, we develop a branch-price-and-cut algorithm for solving P^{ℓ}_{yz}, $\ell \leq r$ described in Section 3.1. At any node of the branch-and-bound tree, the continuous relaxation of P^{ℓ}_{yz}, possibly enhanced by generated rounded capacity cuts (see Rostami et al. 2017, for more details), is solved to obtain a lower bound. Given that P^{ℓ}_{yz} has an exponential number of binary variables z, its continuous relaxation cannot be solved directly. Hence, an iterative column generation procedure is applied in such a way that at each iteration a restricted master problem (RMP), i.e., the continuous relaxation of P^{ℓ}_{yz} restricted to a relatively small subset of the z variables, is solved. To verify the optimality of a given solution of the RMP with respect to the original master problem, we then solve one subproblem per vehicle that corresponds to an elementary shortest path problem with resource constraints. The goal of these subproblems is to find negative reduced cost routes. If some are found, the current RMP solution is not optimal and the variables provided by the subproblems are added to the RMP before starting a new iteration. Otherwise, the algorithm stops, and the current primal solution yields a lower bound for the current branch-and-bound node.

Each ESPPRC subproblem can be solved using a standard labeling algorithm (see, for instance, Feillet et al. 2004, Irnich and Desaulniers 2005). However, given that solving an ESPPRC subproblem is very time-consuming, we use the $ng-$route relaxation introduced by Baldacci et al. (2011). For each vertex $i \in V$, let $\mathcal{V}_i \subset V$ be a subset of vertices with *a priori* fixed size (set to 10 in our experiments) that contains vertex i and its "closest" neighbors. In an $ng-$route, a customer $i \in V_c$ may be visited more than once if between any two visits to i at least one vertex $j \in V$ such that $i \notin \mathcal{V}_j$ is visited.

To derive integer solutions, we branch on the total flow of an arc in E that is not incident to vertex 0. The total flow is computed by vehicle, that is, for each subproblem. When the arc flow is fractional for several arcs, we branch on the flow on an arc $e \in E$ which is not incident to vertex 0 and whose total flow is the closest to 0.5. On one branch, the flow on e is set to 0 by simply removing e from E in all subproblems. On the other branch, the flow on e is set to 1 by removing from E all the other arcs with the same tail or head vertex as e. The columns of the current RMP are then updated accordingly. Finally, the enumeration process applies a best-first search strategy to explore the search tree.

3.2.1 Reduced cost computation

Let us consider the RMP of P^{ℓ}_{yz} for a given $\ell \leq r$ and denote by $\bar{\mathcal{R}}^k \subseteq \mathcal{R}^k$ the subset of routes already generated for vehicle $k \in K$, i.e., those for which there exists a route variable z^k_p in the current RMP. Because of the convexity of the objective function, the RMP is a convex quadratic program that can be solved by a state-of-the-art commercial solver. Let $(\bar{z}, \bar{y})$ be an optimal primal solution of the RMP. In order to verify the optimality of $(\bar{z}, \bar{y})$ for the original problem P^{ℓ}_{yz}, one has to compute the reduced cost of variables z^k, $k \in K$. The following theorem formally shows how to compute these reduced costs.

Theorem 2 *Given a solution $(\bar{z}, \bar{y})$ for the continuous relaxation of P^{ℓ}_{yz}, $\ell \leq r$, the vector of reduced costs of variables z^k, $k \in K$ is expressed as*

$$(\bar{\mu}^k)^T = (\mu^k)^T - \frac{1}{1-\alpha}\sum_{i \in V_c} \pi_i A^k_i - \frac{1}{1-\alpha}\beta^k (e^k)^T + \frac{2\alpha}{1-\alpha}(\bar{y}^k)^T U^{\ell} B^k, \tag{25}$$

where π and β are the Lagrangian vectors associated with constraints (18) and (20), respectively.

Proof. Let us consider the continuous relaxation of P^{ℓ}_{yz} with $\ell \leq r$ and suppose $(\pi, \beta, \sigma, \gamma)$ are the Lagrangian vectors corresponding to constraints (18), (20), (24), and constraints $z \geq 0$, respectively. The Lagrangian function is given as follows:

$$\begin{aligned} L(z, y, \pi, \sigma, \beta, \gamma) =& (1-\alpha)\sum_{k \in K} (\mu^k)^T z^k + \alpha \sum_{k \in K} (y^k)^T y^k - \sum_{i \in V_c}\sum_{k \in K} (\pi_i A^k_i z^k - 1) \\ & - \sum_{k \in K} \beta^k ((e^k)^T z^k - 1) - \sum_{k \in K} \sigma^k (U^{\ell} B^k z^k - y^k) - \sum_{k \in K} \gamma^k z^k. \end{aligned}$$

The stationarity conditions with respect to z and y read as

$$(1-\alpha)(\mu^k)^T - \sum_{i\in V_c} \pi_i A_i^k - \beta^k (e^k)^T - \sigma^k U^\ell B^k - \gamma^k = 0 \qquad \forall k \in K \tag{26}$$

$$2\alpha\,(y^k)^T + \sigma^k = 0 \qquad \forall k \in K. \tag{27}$$

By replacing σ^k with $-2\alpha\,(y^k)^T$ in (26) for each $k \in K$ and rearranging the Lagrangian function, the Lagrangian dual problem can be written as follows:

$$\begin{aligned} \mathrm{D}^\ell_{y\pi\beta}: \quad \max \quad & \sum_{i\in V_c} \pi_i + \sum_{k\in K} \beta^k - \alpha \sum_{k\in K} (y^k)^T y^k \\ \text{s.t.} \quad & (1-\alpha)(\mu^k)^T - \sum_{i\in V_c} \pi_i A_i^k - \beta^k (e^k)^T + 2\alpha\,(y^k)^T U^\ell B^k \geq 0 \qquad \forall k \in K. \end{aligned} \tag{28}$$

Notice that, because $\gamma^k \geq 0$ for all $k \in K$, variables γ^k play the role of slack variables in (26), and, hence, have been removed from $\mathrm{D}^\ell_{y\pi\beta}$.

According to (28), the vector of the reduced costs $\bar{\mu}^k$ of variables z^k, $k \in K$ is given by

$$(1-\alpha)\bar{\mu}^k = (1-\alpha)(\mu^k)^T - \sum_{i\in V_c} \pi_i A_i^k - \beta^k (e^k)^T + 2\alpha\,(\bar{y}^k)^T U^\ell B^k \tag{29}$$

which completes the proof. □

Observe that for each $k \in K$, finding a component of $\bar{\mu}^k$ with minimum reduced cost corresponds to solving an ESPPRC defined on graph $G = (V, E)$ with arc costs

$$\mu_{ij}^k - \frac{1}{1-\alpha}\pi_i + \frac{2\alpha}{1-\alpha} \sum_{s=1}^{r} (\bar{y}_s^k) U_{hs}^\ell \qquad \forall h = (i,j) \in E.$$

4 Computational results

In this section, we present our computational experiments to evaluate the performance of the proposed algorithms on CVRP-SCT. We first show how increasing the number of the top principal components affects the lower bounds as well as the worst-case optimality gaps. Then, we compare the PCA-based algorithm with the exact branch-price-and-cut algorithm of Rostami et al. (2017) in terms of the solution quality and computing time. For simplification, we denote by *Exact-BPC* the branch-price-and-cut algorithm of Rostami et al. (2017) applied to P_{xz} and by *Qapp-BPC* the developed branch-price-and-cut algorithm in Section 3.2 applied to P^ℓ_{yz}, $\ell \leq r$.

All the algorithms were coded in C/C++ using CPLEX 12.6 as a solver for the linear and convex quadratic programs and the GENCOL 4.5 library for the implementation of the branch-price-and-cut algorithms. The experiments were performed on a machine running Linux Intel Xeon(R) CPU E3-1270 (2 quad core CPUs with 3.60 GHz) with 64 gigabytes of RAM. To solve each instance, we considered a time limit of 5 hours for the *Exact-BPC* and 2 hours for *Qapp-BPC*.

To compare the algorithms empirically, we used some datasets introduced in Rostami et al. (2017). These datasets are clustered in three groups of instances $C101, RC101$, and $R101$ which are derived from the Solomon (1987) dataset for the vehicle routing problem with time windows. Time windows have been discarded. As common in the literature, we consider instances with reduced size $n < 100$ obtained from the original instances by considering only their first n customers. In all three groups of instances, the expected travel times for each arc $(i,j) \in E$ is equal to the cost of arc (i,j) given in the classical instances, i.e., $\mu_{ij} = c_{ij}$, the standard deviations of the arc travel times is $\sigma_{ij} = cv_{ij} \times \mu_{ij}$, where cv_{ij} is a random coefficient of variation in the range $[0.01, 0.2]$, and the covariance matrix $\mathcal{C}$ is a positive semidefinite matrix with $\mathcal{C}_{ijls} = \rho_{ijls}\sigma_{ij}\sigma_{ls}$ for each pair of arcs $(i,j),(l,s) \in E$ with $\rho_{ijls} \in [0,1]$. We remark that matrices $\mathcal{C}$ are dense and full rank, which makes the resulting optimization problems very difficult.

4.1 Worst-case optimality gap vs. true optimality gap

In this section we investigate the quality of the lower bounds obtained by solving P^{ℓ}_{yz}, $\ell \leq r$ and compare the optimality gaps with the worst-case gaps described in Theorem 1. To this end, we ran *Qapp-BPC* to solve P^{ℓ}_{yz} with $\alpha = 0.1$ and different number of components, i.e., $\ell \in \{1, 3, 5, 10, 20, 30, 40, 50\}$ on all three groups of instances with $n = 25$. Table 1 reports the results. The first column shows the number of components (ℓ). The next columns give the results of *Qapp-BPC* for $C101, R101$, and $RC101$. For each group of instances and for each ℓ, we report the objective value of P^{ℓ}_{yz} (w^{ℓ}), the true gap (*gap*) between the best known feasible solution (UB) and the lower bound w^{ℓ}, and the worst-case gap (ε) computed by the formula described in Theorem 1. The formula used to compute the true gap is $(UB - w^{\ell})/UB$, where UB is set to the optimal value of P_{xz} obtained by *Exact-BPC*.

As we can observe, when the number of components used to approximate the model increases, both the true gaps and worst-case gaps decrease. However, the true gaps are much smaller than the worst-case gaps when ℓ is small and get closer as ℓ increases. Moreover, the results indicate that among all 600 principal components, only a few of the top principal components are required to compute a good solution.

Table 1: Worst-case optimality gaps vs. the true gaps on instances with $n = 25$ and $\alpha = 0.1$.

	C101			R101			RC101		
ℓ	w^{ℓ}	*gap*	ε	w^{ℓ}	*gap*	ε	w^{ℓ}	*gap*	ε
1	1786.42	0.0045	3.619	4956.05	0.0043	6.660	2824.37	0.0060	7.063
3	1786.49	0.0045	3.258	4957.56	0.0040	5.931	2824.61	0.0060	6.742
5	1783.63	0.0044	2.844	4957.88	0.0040	5.387	2825.36	0.0057	5.881
10	1787.41	0.0038	2.320	4963.48	0.0028	3.923	2827.17	0.0050	4.583
20	1789.54	0.0028	1.433	4970.22	0.0015	2.317	2830.18	0.0040	2.891
30	1790.80	0.0021	0.024	4975.91	0.0003	0.011	2836.00	0.0019	0.010
40	1790.90	0.0020	0.016	4977.21	0.0001	0.009	2836.20	0.0019	0.009
50	1793.20	0.0007	0.006	4977.24	0.0001	0.008	2837.60	0.0014	0.006

4.2 PCA as an approximate scheme

In this section, we evaluate the performance of *Qapp-BPC* applied to P^{ℓ}_{yz} with $\ell = 1, 3, 5$, and 10 principal components. We ran experiments on all three groups of instances with sizes ranging from 15 to 60 customers, and compare the results with those of *Exact-BPC* reported in Rostami et al. (2017).

Tables 2 to 4 report the results for $C101$, $R101$, and $RC101$, respectively. Each table gives the results for three values of $\alpha = 0.1, 0.3$ and 0.5. In all tables, the first two columns indicate the number of customers (n) and the number of required vehicles ($|K|$). The next two columns show the optimal value (*Opt.*) and the total time in seconds to solve the instances to optimality using *Exact-BPC*. The next columns give the results obtained by *Qapp-BPC* with $\ell = 1, 3, 5$, and 10 principal components. For each ℓ, we report the value of the objective function of P_{xz} at the solution found by *Qapp-BPC* (UB), the gaps in percentage ($g(\%)$) between the UB and the optimal value of *Qapp-BPC*, and the total time in seconds ($t(s)$) to solve P^{ℓ}_{yz} to optimality. The formula used to compute the gap is $100(UB - w^{\ell})/UB$ where w^{ℓ} is the optimal value of P^{ℓ}_{yz}. We recall that w^{ℓ} is a valid lower bound for CVRP-SCT. We used "–" if the respective algorithm was not able to find any feasible solution within the time limit.

As we can observe from Table 2, *Exact-BPC* is able to solve C101 instances with up to 30 customers when $\alpha = 0.1$ and 0.3, and up to 20 customers when $\alpha = 0.5$. However, *Qapp-BPC* with $\ell = 1, 3, 5$ and 10 is able to solve all instances with up to 50 customers for all values of α within a shorter time limit. Instances with 60 customers and $\alpha = 0.1$ and 0.3 are also solved by *Qapp-BPC* except for $\alpha = 0.3$ and $\ell = 5$ and 10. Overall, the difficulty of solving P^{ℓ}_{yz} is directly proportional to the values of α and ℓ, i.e., by increasing the values of these parameters the problem is more difficult to solve. On the other hand, and from the solution quality perspective, we can observe that *Qapp-BPC* is able to obtain an optimal solution for all instances whose optimal value is known. Interestingly, the upper bounds provided by *Qapp-BPC* for each instance are identical for all values of ℓ, except for two cases marked by asterisks for which the best upper bounds are

Table 2: Results of Exact-BPC and Qapp-BPC on C101 instances.

Instance		Exact-BPC		Qapp-BPC ($\ell = 1$)			Qapp-BPC ($\ell = 3$)			Qapp-BPC ($\ell = 5$)			Qapp-BPC ($\ell = 10$)		
n	$\|K\|$	*Opt.*	$t(s)$	*UB*	$g(\%)$	$t(s)$	*UB*	$g(\%)$	$t(s)$	*UB*	$g(\%)$	$t(s)$	*UB*	$g(\%)$	$t(s)$
$\alpha = 0.1$															
15	2	1352.9	61.8	1352.9	0.5	2.4	1352.9	0.5	4.0	1352.9	0.5	4.6	1352.9	0.3	3.6
20	2	1528.0	203.1	1528.0	0.5	6.3	1528.0	0.4	5.2	1528.0	0.4	5.5	1528.0	0.4	8.0
25	3	1794.6	1667.4	1794.6	0.5	12.8	1794.6	0.5	11.9	1794.6	0.4	15.4	1794.6	0.4	18.2
30	3	1997.7	10783.8	1997.7	0.6	23.0	1997.7	0.6	22.5	1997.7	0.6	20.2	1997.7	0.6	20.4
40	4	–	18,000.0	3107.8	0.5	99.9	3107.8	0.4	149.7	3107.8	0.4	121.9	3107.8	0.4	155.2
50	5	–	18,000.0	3414.4	0.6	758.3	3414.4	0.6	794.7	3414.4	0.6	1194.1	3414.4	0.5	1129.1
60	6	–	18,000.0	4391.9	0.7	5368.8	4391.9	0.6	5089.3	4390.9	0.6	5756.5	4390.9	0.6	6338.6
$\alpha = 0.3$															
15	2	1197.9	94.4	1204.8	1.8	2.6	1204.8	1.8	3.6	1197.9	1.2	4.2	1197.9	0.8	4.0
20	2	1293.9	257.2	1293.9	1.6	3.4	1293.9	1.5	3.1	1293.9	1.4	3.3	1293.9	1.2	4.7
25	3	1507.3	5004.2	1507.3	1.7	11.4	1507.3	1.7	15.4	1507.3	1.6	16.2	1507.3	1.4	18.4
30	3	1712.0	17037.7	1712.0	1.5	15.8	1712.0	1.4	10.9	1712.0	1.4	10.7	1712.0	1.3	14.6
40	4	–	18,000.0	2561.8	1.7	100.5	2561.8	1.6	138.6	2561.8	1.6	97.2	2561.8	1.5	327.1
50	5	–	18,000.0	2836.0	1.9	1636.4	2836.0	1.8	2123.0	2836.0	1.8	1706.7	2836.0	1.7	4737.5
60	6	–	18,000.0	3594.7	2.0	5581.6	3594.7	1.9	6716.5	3594.7	1.9	7200.0	–	–	7200.0
$\alpha = 0.5$															
15	2	1010.6	261.0	1010.6	3.0	1.2	1010.6	2.9	3.5	1010.6	2.8	3.5	1010.6	2.3	2.6
20	2	1052.9	266.4	1052.9	4.0	3.0	1052.9	3.9	1.8	1052.9	3.5	2.9	1052.9	3.0	3.8
25	3	–	18,000.0	1192.6	4.3	21.4	1192.6	4.3	25.6	1192.6	4.3	25.2	1187.1	3.5	30.0
30	3	–	18,000.0	1359.3	3.0	26.1	1359.3	2.9	20.3	1359.3	2.9	24.9	1359.3	2.7	44.7
40	4	–	18,000.0	1955.3	3.6	546.3	1955.3	3.6	390.6	1955.3	3.5	643.0	1955.3	3.4	943.0
50	5	–	18,000.0	2160.8	4.4	2037.8	2160.8	4.3	2031.7	2150.8	3.7	3701.3	2150.8	3.4	3116.6
60	6	–	18,000.0	2733.7	4.0	7200.0	2733.7	4.0	7200.0	–	–	7200.0	–	–	7200.0

obtained when $\ell = 5$ or 10. Overall, *Qapp-BPC* even with $\ell = 1$ performs fairly well on all instances, with small gap values especially when $\alpha = 0.1$ and 0.3.

For the R101 instances, as we can see from Table 3, *Exact-BPC* is able to solve instances with up to 25 customers for all values of α within the time limit of 5 hours. *Qapp-BPC*, on the other hand, is able to solve instances with up to 60 customers for some values of α and obtain an optimal solution for all instances whose optimal value is known. When both algorithms are able to solve an instance within the time limit, *Qapp-BPC* is significantly faster, especially when the number of components is smaller. For the instances solved to optimality by *Qapp-BPC* within the time limit, the upper bounds provided for each instance are identical for all values of ℓ. However, for instances that are not solved to optimality within the time limit, the upper bounds are not always the same. For example, for $n = 60$ and $\alpha = 0.5$, the upper bound provided by *Qapp-BPC* with $\ell = 1$ is better than the one provided by *Qapp-BPC* with $\ell = 3$. Overall, *Qapp-BPC* even with $\ell = 1$ performs fairly well on all instances; with $\ell = 1$ the average percentage gaps on instances with unknown optimal values are 0.47%, 1.1%, and 1.95% for $\alpha = 0.1, 0.3, 0.5$, respectively.

The results for the RC101 instances are not too different from those for C101 and R101. By inspecting the third column of Table 4, we can see that *Exact-BPC* is able to solve instances with up to 30 customers for $\alpha = 0.1$ and up to 25 customers for the other values of α. *Qapp-BPC* again obtains an optimal solution for all instances whose optimal value is known, and faster than *Exact-BPC*. For the instances solved to optimality by *Qapp-BPC* within the time limit, the upper bounds provided for each instance are identical for all values of ℓ, except one case marked by an asterisk for which the best upper bound is obtained when $\ell = 10$.

4.3 PCA solution quality

As we mentioned earlier, one of the main advantages of solving the approximate model P^{ℓ}_{yz}, $\ell \leq r$ is to provide a feasible solution for the original problem P_{xz}. In this section, we investigate the solutions obtained by *Qapp-BPC* with $\ell = 1$ and compare them with those obtained from the deterministic CVRP and a mixed-integer linear approximation of P_{xz}. In the following subsections, we first present this mixed-integer linear approximation, compare its performance with the PCA-based model, and analyze the quality of solutions in terms of *price of robustness* and *reliability benefits*. These criteria will be defined in Subsection 4.3.2.

Table 3: Results of Exact-BPC and Qapp-BPC on R101 instances.

Instance		Exact-BPC		Qapp-BPC ($\ell = 1$)			Qapp-BPC ($\ell = 3$)			Qapp-BPC ($\ell = 5$)			Qapp-BPC ($\ell = 10$)		
n	$\|K\|$	*Opt.*	$t(s)$	UB	$g(\%)$	$t(s)$	UB	$g(\%)$	$t(s)$	UB	$g(\%)$	$t(s)$	UB	$g(\%)$	$t(s)$
$\alpha = 0.1$															
15	2	3796.4	363.6	3796.4	0.8	9.0	3796.4	0.8	6.0	3796.4	0.6	8.0	3796.4	0.5	7.6
20	2	3825.6	2321.9	3825.6	0.4	34.3	3825.6	0.4	30.2	3825.6	0.3	31.2	3825.6	0.3	30.6
25	2	4977.8	18000.0	4977.8	0.4	105.3	4977.8	0.4	111.0	4977.8	0.4	95.4	4977.8	0.3	107.3
30	3	–	18,000.0	4531.1	0.4	40.1	4531.1	0.4	37.5	4531.1	0.4	33.4	4531.1	0.3	53.3
40	3	–	18,000.0	6524.4	0.5	833.9	6524.4	0.5	815.0	6524.4	0.5	940.8	6524.4	0.4	618.3
50	4	–	18,000.0	6519.9	0.5	274.5	6519.9	0.5	232.7	6519.9	0.5	301.0	6519.9	0.4	371.2
60	5	–	18,000.0	7104.2	0.5	7200.0	7129.5	0.5	7200.0	7109.6	0.5	7200.0	–	–	7200.0
$\alpha = 0.3$															
15	2	4459.7	105.8	4459.7	0.8	6.7	4459.7	0.8	3.2	4459.7	0.7	4.3	4459.7	0.4	4.7
20	2	4580.5	3007.0	4580.5	0.9	24.2	4580.5	0.8	30.6	4580.5	0.8	20.1	4580.5	0.7	21.9
25	2	5783.6	5181.6	5783.6	0.9	26.6	5783.6	0.8	57.6	5783.6	0.7	34.5	5783.6	0.5	42.1
30	3	–	18,000.0	4901.5	1.0	14.3	4901.5	1.0	14.8	4901.5	0.9	18.6	4901.5	0.8	34.8
40	3	–	18,000.0	7484.8	1.0	375.0	7484.8	1.0	334.2	7484.8	0.9	298.3	7484.8	0.7	494.5
50	4	–	18,000.0	7032.2	1.3	7200.0	7138.1	1.3	7200.0	7032.2	1.2	7200.0	7032.2	1.1	7200.0
60	5	–	18,000.0	–	–	7200.0	7471.2	1.3	7200.0	7276.0	1.2	7200.0	7245.0	1.2	7200.0
$\alpha = 0.5$															
15	2	4458.1	57.7	4458.1	1.4	6.2	4458.1	1.4	3.0	4458.1	1.2	4.1	4458.1	0.7	4.4
20	2	4954.9	1771.1	4954.9	1.4	19.3	4954.9	1.3	20.8	4954.9	1.2	11.2	4954.9	1.1	14.3
25	2	6149.1	4673.3	6149.1	1.0	19.7	6149.1	0.9	58.6	6149.1	0.8	27.7	6149.1	0.6	42.6
30	3	–	18,000.0	5019.8	1.6	2280.4	5019.8	1.5	1799.5	5019.8	1.7	1767.8	5019.8	1.2	1759.9
40	3	–	18,000.0	7535.4	1.5	421.0	7535.4	1.5	308.5	7535.4	1.4	304.2	7535.4	1.3	355.0
50	4	–	18,000.0	6693.8	2.6	7200.0	6693.8	2.5	7200.0	6693.8	2.4	7200.0	6693.8	2.2	7200.0
60	5	–	18,000.0	6514.2	2.1	7200.0	6592.8	2.0	7200.0	6356.7	2.4	7200.0	6356.7	2.3	7200.0

Table 4: Results of Exact-BPC and Qapp-BPC on RC101 instances.

Instance		Exact-BPC		Qapp-BPC ($\ell = 1$)			Qapp-BPC ($\ell = 3$)			Qapp-BPC ($\ell = 5$)			Qapp-BPC ($\ell = 10$)		
n	$\|K\|$	*Opt.*	$t(s)$	UB	$g(\%)$	$t(s)$	UB	$g(\%)$	$t(s)$	UB	$g(\%)$	$t(s)$	UB	$g(\%)$	$t(s)$
$\alpha = 0.1$															
15	2	1884.1	17.0	1884.1	0.7	0.4	1884.1	0.7	0.5	1884.1	0.6	0.4	1884.1	0.5	0.6
20	3	2726.9	492.9	2726.9	0.6	2.6	2726.9	0.6	3.8	2726.9	0.5	4.2	2726.9	0.5	6.1
25	3	2841.6	1552.1	2841.6	0.6	2.4	2841.6	0.6	3.5	2841.6	0.6	3.8	2841.6	0.5	6.7
30	4	3918.0	12326.8	3918.0	0.5	18.6	3918.0	0.5	18.0	3918.0	0.5	21.9	3918.0	0.5	20.9
40	5	–	18,000.0	4776.9	0.6	97.7	4776.9	0.6	117.6	4776.9	0.6	153.0	4776.9	0.5	154.0
50	5	–	18,000.0	5017.4	0.7	3785.0	5017.4	0.7	3294.4	5017.4	0.7	2943.8	5017.4	0.6	4668.6
$\alpha = 0.3$															
15	2	1793.8	84.5	1793.8	2.3	0.4	1793.8	2.2	0.7	1793.8	2.1	0.7	1793.8	1.4	1.4
20	3	2334.7	568.3	2334.7	1.7	2.4	2334.7	1.7	4.4	2334.7	1.7	3.1	2334.7	1.6	9.3
25	3	2484.0	3395.7	2484.0	2.0	3.0	2484.0	2.0	4.4	2484.0	1.9	4.9	2484.0	1.8	7.3
30	4	–	18,000.0	3281.5	2.0	16.9	3281.5	1.9	20.7	3281.5	1.9	22.1	3281.5	1.8	34.8
40	5	–	18,000.0	3924.6	2.2	389.1	3924.6	2.2	417.6	3924.6	2.1	515.2	3924.6	2.0	790.0
50	5	–	18,000.0	4206.6	1.8	7200.0	4197.4	1.8	7200.0	4252.0	2.1	7200.0	4242.6	2.0	7200.0
$\alpha = 0.5$															
15	2	1589.7	113.6	1592.2	3.2	0.6	1592.2	3.0	0.5	1592.2	2.9	0.6	1589.7	2.3	1.0
20	3	1860.8	1100.8	1860.8	3.3	2.6	1860.8	3.3	3.3	1860.8	3.2	4.3	1860.8	3.1	6.3
25	3	2060.6	18,000.0	2060.6	3.6	7.1	2060.6	3.5	11.7	2060.6	3.4	11.3	2060.6	3.2	16.6
30	4	–	18,000.0	2525.8	4.2	9.4	2525.8	4.2	12.7	2525.8	4.1	15.4	2525.8	3.9	20.0
40	5	–	18,000.0	2991.2	4.7	1003.1	2991.2	4.6	1120.2	2991.2	4.5	1571.2	2991.2	4.2	3004.2
50	5	–	18,000.0	3278.8	3.9	7200.0	3279.8	3.9	7200.0	3279.2	3.9	7200.0	3278.8	3.6	7200.0

4.3.1 A mixed-integer linear approximation

To obtain an alternative feasible solution for P_{xz}, we exploit the convexity of the problem and construct a mixed-integer linear approximation. To this end, let us consider P_{xz} and represent the quadratic part of the objective function by

$$g(x) = \sum_{k \in K} g_k(x^k) = \sum_{k \in K} (x^k)^T \mathcal{C} x^k.$$

Given a feasible solution $(\bar{x}, \bar{z})$ of P_{xz}, because of the convexity, $g(x)$ can be underestimated by a supporting hyperplane in $\bar{x}$, i.e.

$$g(x) \geq \sum_{k \in K} \left(g_k(\bar{x}^k) + \bigtriangledown g_k(\bar{x}^k)(x^k - \bar{x}^k) \right) = \sum_{k \in K} \underline{g}_k(x^k),$$

where for each $k \in K$, $\bigtriangledown g_k(\bar{x}^k) = (2\mathcal{C}\bar{x}^k)^T$ is the gradient of $g(x)$ at $\bar{x}$. This leads to the following approximation of P_{xz}

$$\begin{aligned} \mathrm{P}'_{xz}: \quad \min \quad & (1-\alpha) \sum_{k \in K} (\mu^k)^T z^k + \alpha \sum_{k \in K} \underline{g}_k(x^k) \\ \text{s.t.} \quad & (x, z) \in \mathcal{F}_{xz}, \end{aligned}$$

This problem is a CVRP with a heterogeneous fleet for which exact and heuristic approaches have been proposed in the literature (see, for instance, Baldacci and Mingozzi 2009, Pessoa et al. 2009).

To solve P'_{xz}, we develop a branch-price-and-cut algorithm denoted by *Lapp-BPC* in the sequel. The *Lapp-BPC* is similar to *Qapp-BPC*. Although in *Lapp-BPC*, the master problem is a linear program we still need to solve for each $k \in K$, an ESPPRC subproblem defined on graph $G = (V, E)$ with arc costs

$$\mu^k_{ij} + 2\frac{\alpha}{1-\alpha} \sum_{(r,s) \in E} \mathcal{C}_{ijrs} \bar{x}^k_{rs} - \frac{1}{1-\alpha}\pi_i \qquad \forall (i,j) \in E,$$

where for each $k \in K$, $\bar{x}^k$ is the feasible solution used to construct $\underline{g}_k(x^k)$. For our tests, we set $\bar{x}$ to the computed optimal solution of the deterministic CVRP.

Table 5 reports computational results for the mixed-integer linear approximation on all three groups of instances C101, R101 and RC101. We ran experiments on all three groups of instances with $20, 30, 40$, and 50 customers and compare the results with those of *Qapp-BPC* with $\ell = 1$ in terms of the provided upper bounds and the computational times. The rows labeled by UB and $t(s)$ report the upper bounds and the total times in seconds obtained by *Lapp-BPC*, while rows labeled by $gap_{UB}(Q, L)(\%)$ and $t(Q/L)$ compare the provided upper bounds and the total times of *Lapp-BPC* with those obtained by *Qapp-BPC*. The former shows the gap between the two upper bounds computed as $gap_{UB}(Q, L)(\%) = 100(UB_L - UB_Q)/UB_L$ where UB_Q and UB_L are the upper bounds obtained by *Qapp-BPC* and *Lapp-BPC*, respectively. $t(Q/L)$ shows the ratio between the running times of *Qapp-BPC* and *Lapp-BPC*. A value greater than one means that *Lapp-BPC* was faster than *Qapp-BPC*.

As we expected, since P'_{xz} is a CVRP, *Lapp-BPC* is faster than *Qapp-BPC*. However, one can observe that the upper bounds given by *Lapp-BPC* are not as tight as those provided by *Qapp-BPC*. For the C101 and RC101 instances the average gaps in percentage between upper bounds given by the two algorithms are 1.05% and 0.2%, respectively, but for the R101 instances, the average gap is 8.35%.

Table 5: Results of Lapp-BPC on all three groups of instances.

	C101				RC101				R101			
	20	30	40	50	20	30	40	50	20	30	40	50
UB	1535.0	2052.9	3115.8	3437.0	2726.9	3936.3	4789.2	5017.3	4179.3	4917.9	7256.0	7005.6
$t(s)$	4.9	2.8	9.4	14.4	0.4	1.6	5.2	11.6	13.0	14.1	50.3	168.4
$gap_{UB}(Q, L)(\%)$	0.5	2.7	0.3	0.7	0.0	0.5	0.3	0.0	8.5	7.9	10.1	6.9
$t(Q/L)$	1.3	8.2	10.6	52.6	6.5	11.6	18.8	326.3	2.6	2.8	16.6	1.6

4.3.2 Price of robustness and reliability benefits

In this section, we investigate other characteristics of the solutions obtained by the two algorithms *Qapp-BPC* with $\ell = 1$ and *Lapp-BPC*. For each algorithm, we analyze the solutions from two different perspectives: *price of robustness* and *reliability benefits*. We define *price of robustness* as the increased expected travel time with

respect to the solutions of the CVRP and *reliability benefits* as the decreased variance with respect to the solutions of the CVRP. Note that, for $\alpha = 0$, both models P^{ℓ}_{yz}, $\ell \leq r$, and P'_{xz} reduce to the CVRP, which minimizes the expected total travel time without considering time variability. This can only be interesting for a planner who has a risk-neutral behavior when planning the operations. However, by increasing the value of α, the chosen models can capture the planner's risk attitude to control time variability and yield different routing solutions.

Figures 1 and 2 show the prices of robustness and the reliability benefits on all three groups of instances with $\alpha = 0.1$. In both figures, *Qapp.I* and *Lapp.I* represent the results of algorithms *Qapp-BPC* with $\ell = 1$ and *Lapp-BPC* on instances $I \in \{C101, RC101, R101\}$, respectively. As we can see, the travel time variance of the solutions provided by both algorithms has been significantly reduced with respect to the variance of the optimal routes of the CVRP. It can be seen that *Lapp-BPC* also provides good solutions though with more deviations than the solutions given by *Qapp-BPC*. In terms of price of robustness, *Qapp-BPC* outperforms *Lapp-BPC*. The price of robustness for *Qapp-BPC* and *Lapp-BPC* deviates between 1.5 - 3.0 % and 2.5 - 3.5 % for the C101 instances, between 1.5 - 3.0 % and 2.5 - 3.5 % for the RC101 instances, and between 12.0 - 16.0 % and 10.0 - 27.0 % for the R101 instances.

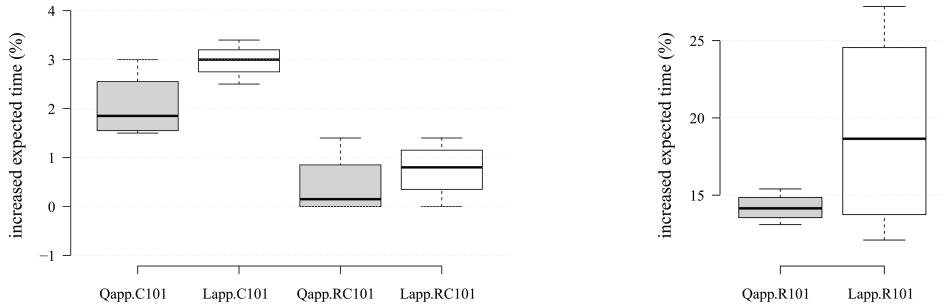

Figure 1: Price of robustness measured as increased expected travel time with respect to the solutions of the CVRP. The results are shown for all three groups of instances with $\alpha = 0.1$. Note that the y-axes in the left and right parts are shown using two different scales.

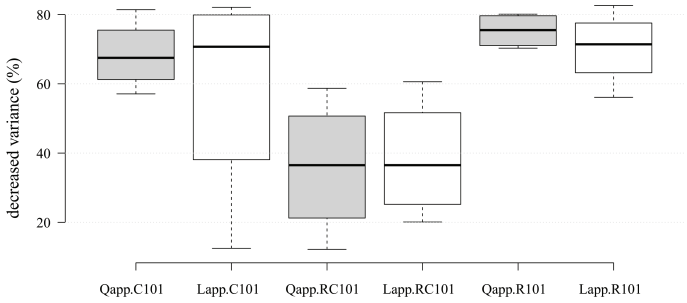

Figure 2: Reliability benefits measured as decreased variance with respect to the solutions of the CVRP. The results are shown for all three groups of instances with $\alpha = 0.1$.

From Figures 1 and 2, one can see that with the current setting of $\alpha = 0.1$, the approximate solutions for the R101 instances appear to be more costly than the approximate solutions for the C101 and RC101 instances. To decrease the price of robustness on R101 instances, we decreased the value of α to 0.01. Figure 3 shows the new results. As we can observe, with the new setting, both algorithms yield reasonably good solutions from both the price of robustness and reliability perspectives. The decreased travel time variance deviates between $34.5 - 47.2\%$ for *Qapp-BPC* and between $29.5 - 43.5\%$ for *Lapp-BPC* with respective increases in the total expected travel time ranging from $0.0 - 2.0\%$ and $0.0 - 2.7\%$, respectively. Overall, the results suggest that *Qapp-BPC* and *Lapp-BPC* offer a good degree of flexibility to compute routes with with different values of the total expected travel time and variance, which in turn allows the planner to choose routes according to her risk attitude.

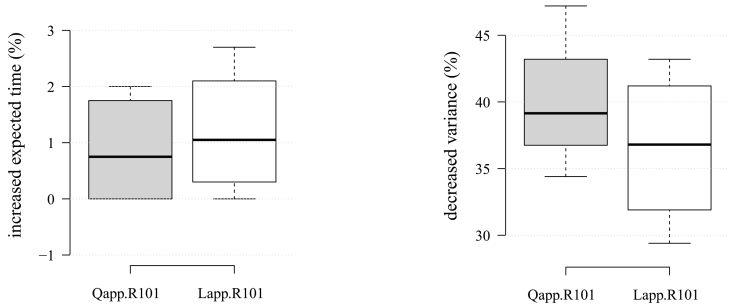

Figure 3: Price of robustness and reliability benefit in terms of increased expected travel time and decreased variance with respect to the solutions of the CVRP on R101 instances with $\alpha = 0.01$.

5 Conclusions

In this paper, we have developed an approximation scheme for combinatorial optimization problems under uncertain and correlated data. To model the uncertainty, we have used the mean-covariance information of the distributions underlying the random data and modeled the problem as a parametric non-linear combinatorial optimization problem. We have proposed a new reformulation based on spectral decomposition of the covariance matrix and applied the PCA to obtain some approximations of the problem. Moreover, we have discussed the quality of the approximate solutions and provided a worst-case optimality gap. To evaluate the robustness and efficiency of our approximate model, we have applied our results to the CVRP-SCT. Our results indicate the effectiveness of the proposed algorithm in solving instances with a large number of customers. In particular, the PCA-based approximate solutions turned out to be identical to the optimal solutions of the original problem on all instances with known optimal values. Solving the approximate model with suitable parameter setting produced routes with a total expected travel time slightly larger than the routes computed by solving the standard CVRP, but with significantly less variance. One possible future research is to combine some polyhedral approximations with the PCA-based approach in an exact solution framework to solve decision-making problems under uncertain and correlated data.

References

Abdi, H. and Williams, L. J. (2010). Principal component analysis. Wiley interdisciplinary reviews: computational statistics, 2(4):433–459.

Agrawal, S., Ding, Y., Saberi, A., and Ye, Y. (2012). Price of correlations in stochastic optimization. Operations Research, 60(1):150–162.

Baldacci, R. and Mingozzi, A. (2009). A unified exact method for solving different classes of vehicle routing problems. Mathematical Programming, 120(2):347–380.

Baldacci, R., Mingozzi, A., and Roberti, R. (2011). New route relaxation and pricing strategies for the vehicle routing problem. Operations Research, 59(5):1269–1283.

Baldacci, R., Mingozzi, A., and Roberti, R. (2012). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. European Journal of Operational Research, 218(1):1–6.

Ben-Tal, A. and Nemirovski, A. (1998). Robust convex optimization. Mathematics of Operations Research, 23(4):769–805.

Bertsimas, D. and Sim, M. (2004). Robust discrete optimization under ellipsoidal uncertainty sets. Technical report, MIT, Cambridge.

Birge, J. R. and Louveaux, F. (2011). Introduction to stochastic programming. Springer Science & Business Media.

Björk, T., Murgoci, A., and Zhou, X. Y. (2014). Mean–variance portfolio optimization with state-dependent risk aversion. Mathematical Finance, 24(1):1–24.

Buchheim, C. and De Santis, M. (2018). An active set algorithm for robust combinatorial optimization based on separation oracles. arXiv preprint arXiv:1804.02239.

Buchheim, C. and Kurtz, J. (2017). Min–max–min robust combinatorial optimization. Mathematical Programming, 163(1–2):1–23.

Delage, E. and Ye, Y. (2010). Distributionally robust optimization under moment uncertainty with application to data-driven problems. Operations Research, 58(3):595–612.

Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. Networks, 44(3):216–229.

Geoffrion, A. M. (1967). Solving bicriterion mathematical programs. Operations Research, 15(1):39–54.

Ghaoui, L. E., Oks, M., and Oustry, F. (2003). Worst-case value-at-risk and robust portfolio optimization: A conic programming approach. Operations Research, 51(4):543–556.

Golub, G. H. and Van Loan, C. F. (2012). Matrix computations, volume 3. JHU Press, Baltimore, MD.

Irnich, S. and Desaulniers, G. (2005). Shortest path problems with resource constraints. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, Column Generation, pages 33–65. Springer US, Boston, MA.

Jolliffe, I. T. and Cadima, J. (2016). Principal component analysis: a review and recent developments. Philosophical transactions. Series A, Mathematical, physical, and engineering sciences, 374(2065):20150202.

Markowitz, H. (1952). Portfolio selection. The Journal of Finance, 7(1):77–91.

Pessoa, A., Uchoa, E., and Poggi de Aragão, M. (2009). A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem. Networks, 54(4):167–177.

Popescu, I. (2007). Robust mean-covariance solutions for stochastic optimization. Operations Research, 55(1):98–112.

Rostami, B., Chassein, A., Hopf, M., Frey, D., Buchheim, C., Malucelli, F., and Goerigk, M. (2018a). The quadratic shortest path problem: complexity, approximability, and solution methods. European Journal of Operational Research, 268(2):473–485.

Rostami, B., Desaulniers, G., Errico, F., and Lodi, A. (2017). The vehicle routing problem with stochastic and correlated travel times. Technical Report DS4DM-2017-016, CERC - Data Science for Real-Time Decision-Making, Montreal.

Rostami, B., Errico, F., and Lodi, A. (2018b). A convex reformulation and an outer approximation for a class of binary quadratic program. Technical Report DS4DM-2018-002, CERC - Data Science for Real-Time Decision-Making, Montreal.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research, 35(2):254–265.

Steinbach, M. C. (2001). Markowitz revisited: Mean-variance models in financial portfolio analysis. SIAM Review, 43(1):31–85.

Toth, P. and Vigo, D., editors (2014). Vehicle Routing: Problems, Methods, and Applications. MOS-SIAM Series on Optimization. SIAM, Philadelphia, PA, 2nd edition.

Wiesemann, W., Kuhn, D., and Sim, M. (2014). Distributionally robust convex optimization. Operations Research, 62(6):1358–1376.