



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Branch-price-and-cut algorithms for the vehicle routing problem with stochastic and correlated travel times

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:

Rostami, B., Desaulniers, G., Errico, F., Lodi, A. (2021). Branch-price-and-cut algorithms for the vehicle routing problem with stochastic and correlated travel times. OPERATIONS RESEARCH, 69(2), 436-455 [10.1287/OPRE.2020.2037].

Availability:

This version is available at: <https://hdl.handle.net/11585/905160> since: 2024-03-01

Published:

DOI: <http://doi.org/10.1287/OPRE.2020.2037>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).
When citing, please refer to the published version.

(Article begins on next page)

Submitted to *Operations Research*
manuscript (Please, provide the manuscript number!)

Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and should not be used to distribute the papers in print or online or to submit the papers to another publication.

Branch-price-and-cut algorithms for the vehicle routing problem with stochastic and correlated travel times

Borzou Rostami

Lazaridis School of Business and Economics, Wilfrid Laurier University and
CERC-Data Science for Real-time Decision-making, Polytechnique Montréal, Canada
brostami@wlu.ca

Guy Desaulniers

Department of Mathematics and Industrial Engineering, Polytechnique Montréal and GERAD, Canada
guy.desaulniers@gerad.ca

Fausto Errico

Department of Civil Engineering, École de Technologie Supérieure de Montréal, GERAD and CIRRELT, Canada
fausto.errico@etsmtl.ca

Andrea Lodi

CERC-Data Science for Real-time Decision-making, Polytechnique Montréal and GERAD, Canada
andrea.lodi@polymtl.ca

In this paper we consider a version of the capacitated vehicle routing problem (CVRP) where travel times are assumed to be uncertain and statistically correlated (CVRP-SCT). In particular we suppose that travel times follow a multivariate probability distribution whose first and second moments are known. The main purpose of the CVRP-CST is to plan vehicle routes whose travel times are reliable, in the sense that observed travel times are not excessively dispersed with respect to their expected value. To this scope we adopt a mean-variance approach, where routes with high travel time variability are penalized. This leads to a parametric binary quadratic program for which we propose two alternative set partitioning reformulations and show how to exploit the structure of the correlation matrix when there is correlation only between adjacent links. For each model, we develop an exact branch-price-and-cut algorithm, where the quadratic component is dealt with either in the column generation master problem or in its subproblem. We tested our algorithms on a rich collection of instances derived from well-known datasets. Computational results show that our algorithms can efficiently solve problem instances with up to 75 customers. Furthermore, the obtained solutions significantly reduce the time variability when compared with standard CVRP solutions.

Key words: Vehicle routing, uncertain travel times, correlation, convex quadratic programming, branch-price-and-cut.

1. Introduction

The capacitated vehicle routing problem (CVRP) has been extensively studied in the literature. Given a fleet of identical capacitated vehicles housed in a single depot and a predefined set of customers, each with a given demand volume, the CVRP consists of finding a set of routes with minimum transportation cost in such a way that each customer is visited exactly once and vehicle capacities are not exceeded. For recent surveys on the CVRP and its variants, the reader is referred to Laporte (2009), Baldacci et al. (2012), Toth and Vigo (2014), and Pecin et al. (2017).

Even though the majority of works on the CVRP assumes deterministic parameters (travel times, demand volumes, customer presence, and others), some of these parameters are affected by a certain degree of uncertainty in many real-life applications and are unknown when planning the operations. For this reason, new research focusing on CVRP variants with stochastic parameters developed in the last two decades (see, e.g., Gendreau et al. 2014, 2016, Oyola et al. 2017).

In this paper we consider a variant of the CVRP where travel times between locations are assumed to be stochastic. The duration of a vehicle route, which is the sum of the travel times between the visited locations, is also stochastic and it can be affected by a large variability. In many contexts, however, it may be desirable to attenuate the dispersion of the route duration around its expected value. For example, if a customer expects a visit at a given time, it is beneficial to provide service at a time instant close to the expected one. In this situation, a decision maker might prefer vehicle routes with lower variability at the expense of a slightly higher expected route duration.

Most of the literature on stochastic vehicle routing problems assumes that random variables are statistically independent. This assumption conflicts, however, with real-life contexts, where statistical dependence is rather the norm. With respect to travel times, for example, several empirical studies showed that strong correlation exists, both positive and negative, among the links in a road network (see, for instance, Nicholson 2015, Seshadri and Srinivasan 2012, Chen et al. 2012, Rachtan et al. 2013, Xing and Zhou 2011). Parent and LeSage (2010) develop a dynamic model that relates travel times with highway infrastructure and congestion. They found that the forecast travel time variance may be underestimated by up to 75% when neglecting positive correlation, and overestimated by up to 100% when neglecting negative correlation.

With the goal of suitably accounting for travel time variations, we focus in this work on the CVRP with stochastic and *correlated* travel times (CVRP-SCT). We assume that the first and the second moments of the travel time probability distributions are known and we adopt a mean-variance approach (see Markowitz 1952), i.e., we seek a trade-off between the expected travel time and its variance, where the latter is assumed to be a measure of the travel time reliability. The resulting model is a parametric binary quadratic program for which we propose two types of set partitioning

(SP) reformulations and develop branch-price-and-cut algorithms for each type. Furthermore, we exploit the structure of the covariance matrix in the case where correlation exists only between adjacent links and specialize the proposed models and algorithms.

1.1. Literature Review

Laporte et al. (1992) study the uncapacitated vehicle routing problem with stochastic travel times. The authors assume a deadline on the route travel time and proposed both a chance-constrained model minimizing the routing cost, and a stochastic program with simple recourse penalizing the expected tardiness costs. Lambert et al. (1993) develop heuristic algorithms for a similar problem with hard deadlines. Kenyon and Morton (2003) address the same problem setting as Laporte et al. (1992) and provide two different models in which the objective is to minimize the latest expected completion time, or its tardiness probability. Teng et al. (2004) study a time-constrained traveling salesman problem with stochastic travel and service times. They model the problem as a two-stage stochastic program and solve it using an integer L-shaped method. For the literature on CVRP with time windows and stochastic travel times, we refer the reader to Li et al. (2010), Errico et al. (2016, 2018) and references therein.

To the best of our knowledge, correlation between travel times received little attention in the stochastic CVRP literature. However, in the context of the shortest path problem, it has been extensively studied. According to the method used to quantify travel time variability, the literature can be partitioned into two main groups: probability-based or variance-based approaches. Among the former group, Fan et al. (2005) consider the shortest path problem in stochastic networks with correlated link service levels. The authors restrict the link and node states to either congested or uncongested and applied conditional probabilities to address the correlation between the states of adjacent nodes. Nie and Wu (2009) consider the same problem in a stochastic and time-dependent network, where link travel times are conditional to the state of the tail node, and hence correlated. They use dynamic programming to solve the resulting problem. Dong et al. (2012) study optimal route selection with reliable expected travel time in the stochastic time-dependent networks where the spatial correlation between travel times of adjacent links are taken into account. They develop an approximation method to obtain the reliability of a route travel time based on the Markovian property of the link states, where each link is assumed to experience either congested or uncongested condition. Samaranayake et al. (2011) maximize the probability of arriving on time at a destination given a departure time and a time budget. The authors develop an algorithm to address the problem when the travel time on each link is correlated to the upstream neighbors via which the link is reached.

Probability-based approaches usually need a complete knowledge of the link and path travel time distributions, which can be statistically and computationally challenging. Therefore, many researchers considered variance-based approaches, which only require the knowledge of the probability distribution first and second moments. Ji et al. (2011) formulate spatial correlations as variance-covariance matrix and propose a simulation-based multi-objective genetic algorithm. Chen et al. (2012) address routing optimization under travel time uncertainty in a network derived from the urban area of Hong Kong. They consider the correlation among travel times of neighboring links, model the problem as a multicriteria shortest path problem, and propose a solution method based on some dominant conditions. Xing and Zhou (2011) consider the stochastic shortest path problem in the presence of both independent and correlated travel times. They use the standard deviation to measure the travel time variability, model the problem as a shortest path problem with a nonlinear objective function, and propose a Lagrangian-relaxation-based approach. Zockaie et al. (2014) study a minimum path travel time budget problem with stochastic and correlated times in which the travel time budget is defined as the sum of the mean path travel time and the scaled standard deviation. They use two existing solution methods in the literature (an outer approximation algorithm and a Monte Carlo simulation method) to obtain optimal solutions. Prakash and Srinivasan (2018) address the shortest path problem in stochastic networks with correlated travel times and minimize a weighted combination of the travel time mean and standard deviation. They develop pruning criteria to eliminate nonoptimal subpaths.

The literature on stochastic CVRPs with correlated travel times is somewhat limited. A CVRP with stochastic time-dependent travel times is studied by Lecluyse et al. (2009) where travel times are assumed to follow a lognormal distribution. The authors extend the objective of the CVRP to a weighted sum of expected travel times and their standard deviation and develop a tabu search algorithm where variances are approximated. Letchford and Nasiri (2015) study a version of the Steiner travelling salesman problem in which the road traversal costs are both stochastic and correlated. They use the variance-covariance matrix to represent the correlated uncertain parameters and model the problem as a binary quadratic program. The authors also consider two special cases where the covariance matrix is either a low rank matrix or restricted to adjacent links. They propose some mixed-integer linear programming reformulations and use a known state-of-the-art solver to solve their models. Bakach et al. (2018) study a vehicle routing problem with a makespan objective and stochastic and correlated travel times. The authors present an approach that approximates the expected makespan and the standard deviation based on extreme-value theory. Embedding the approximate algorithm within a vehicle routing algorithm, they develop a heuristic for the creation of routes for a fleet of vehicles.

In routing optimization under uncertainty, representing variance via variance-covariance matrices often leads to quadratic combinatorial optimization problems, which are much more challenging than their linear counterpart (see, for instance, Letchford and Nasiri 2015, Prakash and Srinivasan 2018). The literature on routing optimization with a quadratic objective function is fairly limited; the quadratic shortest path problem, the quadratic traveling salesman problem (QTSP) and the quadratic CVRP (QCVRP) are among those that received more attention (see Fischer and Helmberg 2013, Rostami et al. 2015, 2016, Martinelli and Contardo 2015). In the QTSP and the QCVRP, the interaction among arcs is modeled by a cost component associated with pairs of arcs sharing a node. Fischer and Helmberg (2013) use the polyhedral structure of a linearized integer programming formulation to develop a branch-and-cut algorithm for the QTSP. Martinelli and Contardo (2015) adapt the linearized integer programming formulation and the valid inequalities proposed by Fischer and Helmberg (2013) to the QCVRP, and developed metaheuristics to find feasible solutions.

1.2. Contributions

As mentioned above, we adopt a mean-variance approach (see Markowitz 1952) for a routing problem with correlated uncertain parameters and devise branch-price-and-cut algorithms. Although these ingredients exist in the literature, combining them is challenging and constitutes our primary contribution. More specifically, our contributions are summarized as follows.

- We introduce a new version of the CVRP where the objective function trades off between the route expected travel time and its variance.
- We first formulate the resulting problem as a parametric convex binary quadratic program. We then propose two types of SP reformulations and develop branch-price-and-cut algorithms to solve them. The first reformulation type yields a column generation master problem with a quadratic objective function, while the subproblems are classical elementary shortest path problems with resource constraints (ESPPRCs). In the second reformulation type, the resulting master problem is a mixed-integer linear program (MILP), while subproblems are ESPPRCs with a quadratic objective function.
- In several applications, the travel times on adjacent links are highly correlated and the correlation among non-adjacent arcs can be neglected (Fan et al. 2005, Nie and Wu 2009). We specialize the proposed models and algorithms for this case.
- The second SP reformulation requires the solution of a quadratic ESPPRC subproblem in the column generation algorithm for which the Bellman optimality principle cannot be applied. We therefore develop a novel dominance rule for the labeling algorithm to efficiently solve the subproblem for the adjacent case.

- We perform an extensive computational study. We first build new sets of benchmark instances by suitably modifying well-known datasets from the literature. We then develop two experimental campaigns, one to test the efficiency of the developed algorithms, the other to assess the quality of the computed solutions in terms of their travel time variability. We were able to solve instances with up to 32 customers for general correlation matrices, and up to 75 customers for the adjacent case. Moreover, with respect to the classical CVRP, the obtained solutions display routes with significantly less variance, at the expense of a slight increase in the average travel time.

1.3. Paper structure

The paper is structured as follows. In Section 2, we introduce the CVRP-SCT and present mathematical models for both the general and the adjacent cases. In Section 3, we provide two different types of SP reformulations. In Section 4, we detail the proposed branch-price-and-cut algorithms. Computational results are then reported in Section 5. Section 6 briefly draws some conclusions and discusses future research directions.

2. Problem statement

In this section, we define the CVRP-SCT where the travel times are assumed to be random variables from the class $\mathcal{M}_{(\mu, \mathcal{C})}^m$ of m -variate distributions with mean μ and covariance \mathcal{C} . In Subsection 2.1, we present a model for the general case where every pair of link travel times might be correlated. However, in many real traffic networks, if one link is in a congested situation, it is very likely that neighboring links are also in a similar situation. In Subsection 2.2, we specialize our model for the adjacent case and propose a linearized formulation which exploits the covariance structure.

2.1. General case

Let K be a set of identical vehicles with capacity Q . Let $G = (V, A)$ be a graph with node set $V = \{0, 1, 2, \dots, n\}$ and arc set A with $|A| = m$. Each node $i \in V_0 = V \setminus \{0\}$ represents a customer having a nonnegative demand q_i , while node 0 corresponds to a depot. For simplicity, we say that node 0 has a demand $q_0 = 0$. In A , there exists an arc (i, j) linking node i to node j if the sum of the demands at these nodes does not exceed Q . A feasible route p corresponds to a circuit $(0, v_1, \dots, v_{h-1}, v_h, 0)$ in G such that $\sum_{\ell=1}^h q_{v_\ell} \leq Q$ and $v_i \neq v_j$ for all $i, j \in \{1, \dots, h\}$, $i \neq j$. Whenever suitable, we also identify p by its arc set $\{(0, v_1), (v_1, v_2), \dots, (v_{h-1}, v_h), (v_h, 0)\}$.

Let t_{ij} be a random variable representing the travel time on arc (i, j) with mean μ_{ij} and standard deviation σ_{ij} . Moreover, let ρ_{ijrs} represent the correlation coefficient between travel times on arcs $(i, j), (r, s) \in A$. The entry of the covariance matrix $\mathcal{C} \in \mathbb{R}^{m \times m}$ for the pair of arcs $(i, j), (r, s) \in A$ is given by

$$\mathcal{C}_{ijrs} = \begin{cases} \sigma_{ij}^2 & (i, j) = (r, s) \\ \rho_{ijrs} \sigma_{ij} \sigma_{rs} & (i, j) \neq (r, s), \end{cases} \quad (1)$$

where σ_{ij}^2 is the variance of the travel time on arc (i, j) . Matrix \mathcal{C} is symmetric and positive semidefinite. For a given route p in G , the variance of the travel time on route p is given by $\sigma_p^2 = \sum_{(i,j) \in p} \sum_{(r,s) \in p} \mathcal{C}_{ijrs}$.

The goal of the CVRP-SCT is to find a set of $|K|$ feasible routes where both the total expected travel times and the total variance are minimized such that each customer is visited exactly once. Note that, even though the travel times of different routes are statistically dependent, it turns out that the total variance can be computed independently for each vehicle. In fact, from the practical point of view, the goal is to penalize the time dispersion (variance) of each route around its mean value, as interactions among different routes do not affect the level of the provided service. For each vehicle $k \in K$ and each arc $(i, j) \in A$, we define a binary variable x_{ij}^k which is equal to 1 if arc (i, j) is traversed by vehicle k , and 0 otherwise. The superscript k is needed here because only the arcs traversed by the same vehicle contribute to the variance of the route taken by this vehicle.

Let $\delta^+(i) = \{(i, j) \in A\}$ and $\delta^-(i) = \{(j, i) \in A\}$ denote the sets of outgoing and incoming arcs to node i , respectively. Furthermore, let $\gamma(S)$ be the minimum number of vehicles required to serve the customers in subset $S \subseteq V_0$ according to their demands. Then the CVRP-SCT can be formulated as the following bi-objective integer program:

$$\begin{aligned}
 \text{P: } \quad & \min \quad \begin{cases} \sum_{k \in K} \sum_{(i,j) \in A} \mu_{ij} x_{ij}^k \\ \sum_{k \in K} \sum_{(i,j) \in A} \sum_{(r,s) \in A} \mathcal{C}_{ijrs} x_{ij}^k x_{rs}^k \end{cases} & (2) \\
 \text{s.t.} \quad & \sum_{k \in K} \sum_{(i,j) \in \delta^+(i)} x_{ij}^k = 1 & \forall i \in V_0 & (3) \\
 & \sum_{(0,j) \in \delta^+(0)} x_{0j}^k = 1, & \forall k \in K & (4) \\
 & \sum_{(i,j) \in \delta^+(i)} x_{ij}^k - \sum_{(j,i) \in \delta^-(i)} x_{ji}^k = 0, & \forall k \in K, i \in V_0 & (5) \\
 & \sum_{(i,0) \in \delta^-(0)} x_{i0}^k = 1, & \forall k \in K & (6) \\
 & \sum_{k \in K} \sum_{i \notin S} \sum_{j \in S: (i,j) \in A} x_{ij}^k \geq \gamma(S), & \forall S \subseteq V_0 & (7) \\
 & x_{ij}^k \in \{0, 1\}, & \forall k \in K, (i, j) \in A, & (8)
 \end{aligned}$$

where the objective function (2) minimizes simultaneously the total expected travel time and the total variance. Constraints (3) state that each customer must be visited exactly once. Constraints (4) to (6) ensure that each vehicle is used exactly once and that flow conservation is satisfied at each customer node. The capacity inequalities (7) impose vehicle capacity on each route and ensure that all routes are connected to the depot. Finally, the binary requirements (8) restrict the domain of the variables.

A feasible solution \hat{x} is said mean-variance efficient (see, e.g., Markowitz 1952) if there exists no other feasible solution x such that

$$(\mu x \leq \mu \hat{x} \text{ and } x^T \mathcal{C} x < \hat{x}^T \mathcal{C} \hat{x}) \text{ or } (x^T \mathcal{C} x \leq \hat{x}^T \mathcal{C} \hat{x} \text{ and } \mu x < \mu \hat{x}).$$

In other words, no solution is at least as good as \hat{x} for both objectives and strictly better for at least one.

One of the widely-used methods for bi-objective optimization is the weighted sum method (see, e.g., Zadeh 1963), which replaces the two objective functions by a weighted sum of them. Here, we use the following convex combination of the total travel time mean and the total travel time variance to compute a single mean-variance efficient solution:

$$P_\alpha: \quad \min \quad (1 - \alpha) \sum_{k \in K} \sum_{(i,j) \in A} \mu_{ij} x_{ij}^k + \alpha \sum_{k \in K} \sum_{(i,j) \in A} \sum_{(r,s) \in A} \mathcal{C}_{ijrs} x_{ij}^k x_{rs}^k$$

$$\text{s.t.} \quad (3) - (8).$$

for some $\alpha \in [0, 1]$.

Note that since the covariance matrix \mathcal{C} is positive semidefinite, problem P_α is a convex binary quadratic program for any given $\alpha \in [0, 1]$. Moreover, for $\alpha \neq 1$, we can define a new parameter $\delta = \alpha/(1 - \alpha)$ as a “risk factor,” and we can rewrite the objective function as the conic combination of the sums of the travel time means and variances, which is known as a risk-averse objective function in the literature (Prakash and Srinivasan 2018). This factor shows the amount of decrease in the total mean travel time that is equivalent, in terms of utility, to a unit decrease in the total travel time variance.

To be concise in the sequel, we introduce for a given $\alpha \in [0, 1]$ the following parameters:

$$\begin{aligned} \mathcal{C}_{ijrs}^\alpha &= \alpha \mathcal{C}_{ijrs}, & \forall (i,j), (r,s) \in A, \\ \mu_{ij}^\alpha &= (1 - \alpha) \mu_{ij}, & \forall (i,j) \in A, \\ \tilde{\mu}_{ij}^\alpha &= (\mu_{ij}^\alpha + \mathcal{C}_{ijij}^\alpha), & \forall (i,j) \in A. \end{aligned}$$

2.2. Correlation only between adjacent links

In this section, we consider a special case of the CVRP-SCT where the correlation coefficient between the travel times of any pair of non adjacent arcs is assumed to be zero. This implies a sparse covariance matrix where $\mathcal{C}_{ijrs} = 0$ for all non adjacent arcs $(i,j), (r,s) \in A$. To exploit the covariance structure, we also assume that $\mathcal{C}_{i0j} = 0$ for all adjacent arcs $(i,0), (0,j) \in A$. Note that, if $\mathcal{C}_{i0j} \neq 0$ for some adjacent arcs $(i,0), (0,j) \in A$, one can still use the general model P_α described in Section 2.1. Given the fact that in any feasible solution of the CVRP-SCT, all the routes are elementary, we need to count the covariance for adjacent arc pairs only if these arcs are traversed consecutively. Hence, for a given route $p = (0, v_1, \dots, v_{h-1}, v_h)$ with $v_h = 0$, the variance of the travel time on route p can be computed as

$$\sigma_p^2 = \sum_{(i,j) \in p} \sum_{(r,s) \in p} \mathcal{C}_{ijrs} = \sum_{i=0}^{h-2} 2\mathcal{C}_{i,i+1,i+1,i+2}. \quad (9)$$

According to (9), we only need to take into account the covariance for arcs $(i, j), (r, s) \in A$ if $j \in V_0, j = r$ and these arcs are traversed by some vehicle. Since every pair of consecutive arcs $(i, j), (j, s), j \in V_0$, can only be traversed by the same vehicle, we replace the three-index variables $x_{ij}^k, (i, j) \in A, k \in K$, by two-index variables x_{ij} indicating the presence of arc (i, j) in the computed optimal solution. Accordingly, we can rewrite P_α as follows:

$$P_\alpha^{adj}: \quad \min \quad \sum_{(i,j) \in A} \tilde{\mu}_{ij}^\alpha x_{ij} + \sum_{j \in V_0} \sum_{(i,j) \in A} \sum_{(j,l) \in A} 2C_{ijjl}^\alpha x_{ij} x_{jl} \quad (10)$$

$$\text{s.t.} \quad \sum_{(i,j) \in \delta^+(i)} x_{ij} = 1, \quad \forall i \in V_0 \quad (11)$$

$$\sum_{(i,j) \in \delta^-(j)} x_{ij} = 1, \quad \forall j \in V_0 \quad (12)$$

$$\sum_{(0,j) \in \delta^+(0)} x_{0j} = |K| \quad (13)$$

$$\sum_{(i,0) \in \delta^-(0)} x_{i0} = |K| \quad (14)$$

$$\sum_{i \notin S} \sum_{j \in S: (i,j) \in A} x_{ij} \geq \gamma(S), \quad \forall S \subseteq V_0 \quad (15)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A. \quad (16)$$

The objective function (10) minimizes a convex combination of the total expected travel time and the total variance. Note that, because we only use a single expression $x_{ij}x_{jl}$ ($x_{jl}x_{ij}$ is omitted) for each arc pair $(i, j), (j, l) \in A$, the objective coefficients C_{ijjl}^α are multiplied by 2. Constraints (11) and (12) ensure that exactly one arc enters and leaves each node associated with a customer, respectively. Constraints (13) and (14) impose the degree requirements for the depot. The capacity inequalities (15) play the same role as (7).

Model P_α^{adj} is again a convex binary quadratic program. However, in comparison with P_α , it has much less variables and constraints with a sparse objective function. We can exploit the sparsity of this function by linearizing expressions $x_{ij}x_{jl}$ for all arcs $(i, j), (j, l) \in A$ to produce the following MILP:

$$P_\alpha^{adjL}: \quad \min \sum_{(i,j) \in A} \tilde{\mu}_{ij}^\alpha x_{ij} + \sum_{j \in V_0} \sum_{(i,j) \in A} \sum_{(j,l) \in A} 2C_{ijjl}^\alpha y_{ijl}$$

$$\text{s.t.} \quad (11) - (16)$$

$$\sum_{(l,i) \in \delta^-(i)} y_{lij} = x_{ij}, \quad \forall (i, j) \in A \text{ with } i \neq 0 \quad (17)$$

$$\sum_{(j,l) \in \delta^+(j)} y_{ijl} = x_{ij}, \quad \forall (i, j) \in A \text{ with } j \neq 0 \quad (18)$$

$$y_{ijl} \geq 0, \quad \forall (i, j), (j, l) \in A \text{ with } j \in V_0, \quad (19)$$

where $y_{ijl}, (i, j), (j, l) \in A$ with $j \in V_0$, are new nonnegative variables that are set equal to $x_{ij}x_{jl}$ through constraints (17) and (18). These constraints may be seen as a kind of flow conservation constraints for each arc $(i, j) \in A$: the sum of the flows into $(i, j), i \neq 0$ via arcs (l, i) has to be equal to the sum of the flows out of (i, j) via arcs (j, l) . Moreover, constraints (17) and (18) imply

$$y_{ijl} \geq x_{ij} + x_{jl}, \quad y_{ijl} \leq x_{ij}, \quad \text{and} \quad y_{ijl} \leq x_{jl}, \quad \forall (i, j), (j, l) \in A, \quad (20)$$

as stated in the following theorem.

THEOREM 1. *Model P_α^{adjL} is a valid reformulation of P_α^{adj} and constraints (17) and (18) imply the standard linearization constraints (20).*

Proof. See Appendix A.

Note that model P_α^{adjL} is similar to the QCVRP studied by Martinelli and Contardo (2015). The QCVRP is a version of the CVRP with possible interaction among the consecutive arcs (i.e., a cost component associated with pairs of arcs sharing a node.) The interaction costs are the results of either angle penalties (with application in engineering and logistics) or reload costs (with application in unloading and reloading operations of the cargo).

3. Set partitioning reformulations

In this section, we propose two types of SP reformulations for each model P_α , P_α^{adj} , and P_α^{adjL} described in Section 2. Each SP formulation is obtained by applying the Dantzig-Wolfe decomposition principle (Dantzig and Wolfe 1960) on the corresponding model and contains an exponential number of binary variables, each associated with a feasible route. In the SP formulations of type I, in addition to binary route variables, we also keep the original arc-flow variables and the corresponding variance term in the objective function. In this case, the cost of associated to route variables only accounts for the expected travel time. In the SP formulations of type II, we only consider binary route variables as in the classical SP formulation of the CVRP introduced by Balinski and Quandt (1964), and the cost associated to route variables is a function of the mean and the variance of its travel time.

3.1. Set partitioning reformulations of type I

Consider first P_α . Let \mathcal{R}^k be the set of all feasible routes for vehicle $k \in K$. For each route $p \in \mathcal{R}^k$, let $\mu_r^\alpha = \sum_{(i,j) \in p} \mu_{ij}^\alpha$ represent the expected travel time on route p . Furthermore, for each customer $i \in V_0$ (resp. arc $(i,j) \in A$), let a_{ip} (resp. b_{ijp}) be a binary parameter that takes value 1 if route p visits customer i (resp. traverses arc (i,j)) and 0 otherwise. For each vehicle $k \in K$ and route $p \in \mathcal{R}^k$, we define a binary variable z_p^k that is equal to 1 if vehicle k uses route p and 0 otherwise.

Model P_α can then be reformulated as the following binary quadratic program:

$$SP1_\alpha: \quad \min \quad \sum_{k \in K} \sum_{p \in \mathcal{R}^k} \mu_p^\alpha z_p^k + \sum_{k \in K} \sum_{(i,j) \in A} \sum_{(r,s) \in A} C_{ijrs}^\alpha x_{ij}^k x_{rs}^k \quad (21)$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{p \in \mathcal{R}^k} a_{ip} z_p^k = 1, \quad \forall i \in V_0 \quad (22)$$

$$\sum_{p \in \mathcal{R}^k} b_{ijp} z_p^k = x_{ij}^k, \quad \forall k \in K, (i,j) \in A \quad (23)$$

$$\sum_{p \in \mathcal{R}^k} z_p^k = 1, \quad \forall k \in K \quad (24)$$

$$z_p^k \in \{0, 1\}, \quad \forall k \in K, p \in \mathcal{R}^k \quad (25)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall k \in K, (i,j) \in A. \quad (26)$$

For a given value of α , the first term of the objective function (21) minimizes the total expected travel time, while the second term minimizes the total variance. Constraints (22) ensure that each customer is visited once. Constraints (23) link the original arc-flow variables x_{ij}^k to the new route variables z_p^k . Constraints (24) impose to select only one route for each vehicle. Finally, binary requirements on the variables are expressed through (25) and (26).

Note that if $C_{ijrs} \geq 0$ for all arc pairs $(i, j), (r, s) \in A$, then one can replace the equality sign in constraints (23) by a less-than-or-equal sign because, for any feasible z -solution, there always exists a feasible x -solution for which these inequalities are tight. Note also that the binary requirements (25) and (26) can be replaced by integrality requirements because of constraints (22). Hence, the variables z_p^k and x_{ij}^k do not have to be upper bounded in a continuous relaxation. Similar remarks apply to the models presented below.

For the SP reformulations of P_α^{adj} and P_α^{adjL} , let us denote by \mathcal{R} the set of all feasible routes. Each route $p \in \mathcal{R}$ has an associated expected travel time $\tilde{\mu}_p^\alpha = \sum_{(i,j) \in p} \tilde{\mu}_{ij}^\alpha$. For each route $p \in \mathcal{R}$, we define a binary variable z_p which equals 1 if and only if route p is selected in a solution.

Using this notation, model P_α^{adj} can be reformulated as the following SP model with a quadratic objective function:

$$SP1_\alpha^{adj}: \quad \min \quad \sum_{p \in \mathcal{R}} \tilde{\mu}_p^\alpha z_p + \sum_{j \in V_0} \sum_{(i,j) \in A} \sum_{(j,l) \in A} 2C_{ijrs}^\alpha x_{ij} x_{jl} \quad (27)$$

$$\text{s.t.} \quad \sum_{p \in \mathcal{R}} a_{ip} z_p = 1, \quad \forall i \in V_0 \quad (28)$$

$$\sum_{p \in \mathcal{R}} b_{ijp} z_p = x_{ij}, \quad \forall (i, j) \in A \quad (29)$$

$$\sum_{p \in \mathcal{R}} z_p = |K| \quad (30)$$

$$z_p \in \{0, 1\}, \quad \forall p \in \mathcal{R} \quad (31)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A, \quad (32)$$

The objective function (27) and constraints (28)–(32) play the same role as (21) and (22)–(26), but for the adjacent case.

In a similar way, model P_α^{adjL} can be reformulated as follows:

$$\min \quad \sum_{p \in \mathcal{R}} \tilde{\mu}_p^\alpha z_p + \sum_{j \in V_0} \sum_{(i,j) \in A} \sum_{(j,l) \in A} 2C_{ijjl}^\alpha y_{ijl}$$

$$\text{s.t.} \quad (17) - (19), (28) - (32).$$

Projecting out the variables x_{ij} results in the following SP formulation:

$$SP1_\alpha^{adjL}: \quad \min \quad \sum_{p \in \mathcal{R}} \tilde{\mu}_p^\alpha z_p + \sum_{j \in V_0} \sum_{(i,j) \in A} \sum_{(j,l) \in A} 2C_{ijjl}^\alpha y_{ijl} \quad (33)$$

$$\text{s.t.} \quad (19), (28), (30) - (32)$$

$$\sum_{(l,i) \in \delta^+(i)} y_{lij} - \sum_{p \in \mathcal{R}} b_{ijp} z_p = 0, \quad \forall (i, j) \in A \text{ with } i \neq 0 \quad (34)$$

$$\sum_{(j,l) \in \delta^-(j)} y_{ijl} - \sum_{p \in \mathcal{R}} b_{ijp} z_p = 0, \quad \forall (i, j) \in A \text{ with } j \neq 0, \quad (35)$$

where constraints (34) and (35) link the linearization variables y_{ijl} directly to the routing variables z_p .

3.2. Set partitioning reformulations of type II

For type II, all three models P_α , P_α^{adj} and P_α^{adjL} can yield the same SP reformulation that we expose here. Let \mathcal{R} be the set of feasible routes. For each route $p \in \mathcal{R}$, let μ_p^α and v_p^α be its travel time mean and variance, respectively, i.e.,

$$\begin{aligned}\mu_p^\alpha &= \sum_{(i,j) \in p} \mu_{ij}^\alpha \\ v_p^\alpha &= \sum_{(i,j) \in p} \sum_{(r,s) \in p} c_{ijrs}^\alpha.\end{aligned}$$

Moreover, for each route $p \in \mathcal{R}$, let $c_p^\alpha = \mu_p^\alpha + v_p^\alpha$ be its cost and define a binary variable z_p which takes value 1 if route p is chosen and 0 otherwise.

The models P_α , P_α^{adj} and P_α^{adjL} can be reformulated as the following SP model:

$$\begin{aligned}SP2_\alpha: \quad \min \quad & \sum_{p \in \mathcal{R}} c_p^\alpha z_p \\ \text{s.t.} \quad & (28), (30), (31),\end{aligned}\tag{36}$$

where the objective function minimizes the total routing costs that are now function of the travel time variance. This reformulation is valid for both the general and the adjacent case. Nevertheless, in the following, we use it only for the adjacent case because the best algorithm we designed for the general case is not able to solve large enough instances.

4. Branch-price-and-cut algorithms for solving the SP reformulations

To solve the SP reformulations described in Section 3, we develop branch-price-and-cut algorithms (see Barnhart et al. 1998, Desaulniers et al. 1998, Lübbecke and Desrosiers 2005). Such an algorithm is a branch-and-bound algorithm where the lower bounds are computed by column generation and cuts are added to tighten the continuous relaxations (also called the master problems) encountered throughout the search tree. Column generation is an iterative procedure which solves at each iteration a restricted master problem (RMP), i.e., the master problem restricted to a relatively small subset of the variables, and one or several subproblems. Solving the RMP provides a primal and a corresponding dual solution. By using information available in the dual solution, the role of the subproblems is to verify the optimality of the primal solution with respect to the complete master problem and, if it is not optimal, to provide additional columns (variables) to add to the RMP. In this case, the RMP is updated with these new columns and a new iteration is started. Otherwise, the algorithm stops and the current primal solution yields a lower bound for the current branch-and-bound node.

For the proposed SP reformulations, the master problems correspond to the continuous relaxations of models $SP1_\alpha$, $SP1_\alpha^{adj}$, $SP1_\alpha^{adjL}$, and $SP2_\alpha$, where we relax the upper bounds on all z and x variables as discussed in Section 3.1. The subproblems serve to generate only the z variables. The other variables, that is, the x and y variables, are not generated dynamically; they are all included in all RMPs.

Below, we first describe the subproblems arising for each reformulation type and how they are solved. Then, we discuss some acceleration strategies, the cuts that we apply, and the branching strategies used to derive integer solutions.

4.1. Subproblems for the type I SP reformulations

In this section, we discuss the subproblems arising from $SP1_\alpha$, $SP1_\alpha^{adj}$ and $SP1_\alpha^{adjL}$. Let us consider first the continuous relaxation of $SP1_\alpha$ restricted to a subset of the routes in $\cup_{k \in K} \mathcal{R}^k$. This RMP is a linearly constrained convex quadratic program that can be solved by a commercial software package. Let (\bar{z}, \bar{x}) be an optimal solution of this RMP. To find the associated reduced costs, let us write down the Lagrangian function

$$\begin{aligned} L_\alpha(z, x, \pi, \lambda, \beta, \nu, \theta) = & \sum_{k \in K} \sum_{p \in \mathcal{R}^k} \mu_p^\alpha z_p^k + \sum_{k \in K} \sum_{(i,j) \in A} \sum_{(r,s) \in A} \mathcal{C}_{ijrs}^\alpha x_{ij}^k x_{rs}^k \\ & - \sum_{i \in V_0} \pi_i \left(\sum_{k \in K} \sum_{p \in \mathcal{R}^k} a_{ip} z_p^k - 1 \right) - \sum_{k \in K} \sum_{(i,j) \in A} \lambda_{ij}^k \left(\sum_{p \in \mathcal{R}^k} b_{ijp} z_p^k - x_{ij}^k \right) \\ & - \sum_{k \in K} \beta^k \left(\sum_{p \in \mathcal{R}^k} z_p^k - 1 \right) - \sum_{k \in K} \sum_{p \in \mathcal{R}^k} \nu_p^k z_p^k - \sum_{k \in K} \sum_{(i,j) \in A} \theta_{ij}^k x_{ij}^k. \end{aligned}$$

where π, λ and β are the Lagrangian vectors associated with constraints (22)–(24), respectively, and ν and θ are those associated with the nonnegativity constraints $z \geq 0$ and $x \geq 0$, respectively.

The stationarity conditions with respect to z and x write as:

$$\mu_p^\alpha - \sum_{i \in V_0} \pi_i a_{ip} - \sum_{(i,j) \in A} \lambda_{ij}^k b_{ijp} - \beta^k - \nu_p^k = 0, \quad \forall k \in K, p \in \mathcal{R}^k \quad (37)$$

$$\lambda_{ij}^k - \theta_{ij}^k + \sum_{(r,s) \in A} 2\mathcal{C}_{ijrs}^\alpha x_{rs}^k = 0, \quad \forall k \in K, (i,j) \in A. \quad (38)$$

As a consequence of (38), we can substitute λ_{ij}^k by $\theta_{ij}^k - \sum_{(r,s) \in A} 2\mathcal{C}_{ijrs}^\alpha x_{rs}^k$ for each $k \in K, (i,j) \in A$ and rearrange the Lagrangian function to give:

$$\begin{aligned} L_\alpha(z, x, \pi, \lambda, \beta, \nu, \theta) = & \sum_{k \in K} \sum_{p \in \mathcal{R}^k} \left(\mu_p^\alpha - \sum_{i \in V_0} \pi_i a_{ip} - \sum_{(i,j) \in A} \theta_{ij}^k b_{ijp} + \sum_{(i,j) \in A} \sum_{(r,s) \in A} 2b_{ijp} \mathcal{C}_{ijrs}^\alpha x_{rs}^k - \beta^k - \nu_p^k \right) z_p^k \\ & - \sum_{k \in K} \sum_{(i,j) \in A} \sum_{(r,s) \in A} \mathcal{C}_{ijrs}^\alpha x_{ij}^k x_{rs}^k + \sum_{i \in V_0} \pi_i + \sum_{k \in K} \beta^k. \end{aligned}$$

Taking into account (37) and (38), the Lagrangian dual problem is given by

$$\begin{aligned}
\max \quad & \sum_{i \in V_0} \pi_i + \sum_{k \in K} \beta^k - \sum_{k \in K} \sum_{(i,j) \in A} \sum_{(r,s) \in A} C_{ijrs}^\alpha x_{ij}^k x_{rs}^k \\
\text{s.t.} \quad & \mu_p^\alpha - \sum_{i \in V_0} \pi_i a_{ip} - \sum_{(i,j) \in A} \theta_{ij}^k b_{ijp} + \sum_{(i,j) \in A} \sum_{(r,s) \in A} 2b_{ijp} C_{ijrs}^\alpha x_{rs}^k - \beta^k - \nu_p^k = 0, \\
& \forall k \in K, p \in \mathcal{R}^k \\
& x_{ij}^k \geq 0, \quad \theta_{ij}^k \geq 0, \quad \forall (i,j) \in A, k \in K \\
& \nu_p^k \geq 0, \quad \forall k \in K, p \in \mathcal{R}^k.
\end{aligned}$$

Observe that, for $k \in K$ and $p \in \mathcal{R}^k$, the variables $\nu_p^k \geq 0$ and $\zeta_p^k = \sum_{(i,j) \in A} \theta_{ij}^k b_{ijp} \geq 0$ play the role of slack variables and can thus be removed to yield:

$$D^\alpha: \max \quad \sum_{i \in V_0} \pi_i + \sum_{k \in K} \beta^k - \sum_{k \in K} \sum_{(i,j) \in A} \sum_{(r,s) \in A} C_{ijrs}^\alpha x_{ij}^k x_{rs}^k \quad (39)$$

$$\text{s.t.} \quad \mu_p^\alpha - \sum_{i \in V_0} \pi_i a_{ip} + \sum_{(i,j) \in A} \sum_{(r,s) \in A} 2b_{ijp} C_{ijrs}^\alpha x_{rs}^k - \beta^k \geq 0, \quad \forall k \in K, p \in \mathcal{R}^k \quad (40)$$

$$x_{ij}^k \geq 0, \quad \forall (i,j) \in A, k \in K. \quad (41)$$

At a given column generation iteration, denote by $\mathcal{Q}^k \subseteq \mathcal{R}^k$ the subset of routes already generated for vehicle $k \in K$, i.e., those for which there exists a route variable in the current RMP. Let us consider the solution (\bar{z}, \bar{x}) to the master problem of $SP1_\alpha$ which is built as follows: $z_p^k = 0$ for all routes $p \in \mathcal{R}^k \setminus \mathcal{Q}^k$, $k \in K$, that have not been generated yet; all the other variables are set to their value in the primal solution computed for the current RMP. Verifying if (\bar{z}, \bar{x}) is also optimal for the complete master problem entails searching a route for which the corresponding constraint (40) is violated. The reduced cost $\hat{\mu}_{pk}^\alpha$ of a route $p \in \mathcal{R}^k$, $k \in K$, can thus be expressed as

$$\hat{\mu}_{pk}^\alpha = \mu_p^\alpha - \sum_{i \in V_0} \pi_i a_{ip} + \sum_{(i,j) \in A} \sum_{(r,s) \in A} 2b_{ijp} C_{ijrs}^\alpha \bar{x}_{rs}^k - \beta^k = \sum_{(i,j) \in \mathcal{P}} (\mu_{ij}^\alpha - \pi_i + \sum_{(r,s) \in A} 2C_{ijrs}^\alpha \bar{x}_{rs}^k) - \beta^k, \quad (42)$$

with $\pi_0 = 0$.

For each vehicle $k \in K$, there is one subproblem that consists of minimizing (42) over the set of feasible routes \mathcal{R}^k . It corresponds to an ESPPRC defined on network G , where the cost of arc $(i,j) \in A$ is equal to $\mu_{ij}^\alpha - \pi_i + \sum_{(r,s) \in A} 2C_{ijrs}^\alpha \bar{x}_{rs}^k$. A single resource, namely, a load resource, is required to enforce vehicle capacity.

A similar process can be applied to determine the subproblem arising from model $SP1_\alpha^{adj}$. In this case, there is a single subproblem which is also an ESPPRC but with different arc costs. For $SP1_\alpha^{adjL}$, the master problem is linear and there is a single ESPPRC subproblem that can be derived directly from the reduced cost of a variable in a linear program (see Lübbecke and Desrosiers 2005). For the sake of conciseness, we do not provide the arc costs for these subproblems.

The ESPPRC subproblems can be solved using a standard labeling algorithm (see, e.g., Feillet et al. 2004, Irnich and Desaulniers 2005).

4.2. Subproblem for the type II SP reformulation

Consider the linear master problem arising from $SP2_\alpha$ and denote by π_i , $i \in V_0$, and β the dual variables associated with constraints (28) and (30), respectively. At a given column generation iteration, let $\mathcal{Q} \subseteq \mathcal{R}$ be the subset of feasible routes already generated. Furthermore, let \bar{z} be a solution to the master problem derived from the optimal solution computed for the current RMP, that is, all variables not yet generated are set to 0. This solution is optimal for the complete master problem if and only if there exist no routes with a negative reduced cost. The reduced cost \hat{c}_p^α of a route $p \in \mathcal{R}$ is:

$$\hat{c}_p^\alpha = c_p^\alpha - \sum_{i \in V_0} \pi_i a_{ip} - \beta = \sum_{(i,j) \in p} \tilde{c}_{ij}^\alpha + \sum_{(i,j) \in p} \sum_{\substack{(r,s) \in p \\ (r,s) \neq (i,j)}} C_{ijrs}^\alpha, \quad (43)$$

where

$$\tilde{c}_{ij}^\alpha = \begin{cases} \tilde{\mu}_{ij}^\alpha - \beta & \text{if } i = 0 \\ \tilde{\mu}_{ij}^\alpha - \pi_i & \text{otherwise} \end{cases} \quad \forall (i,j) \in A. \quad (44)$$

The subproblem consists of minimizing (43) over the set of feasible routes. In this case, it corresponds to an ESPPRC with a quadratic objective function. Note that, even without the elementarity and resource constraints, this subproblem is strongly NP-hard as shown by Rostami et al. (2015). Therefore, its complexity is not solely due to the ESPPRC structure but also to the quadratic objective function, which is nonadditive and invalidates the Bellman optimality principle. To solve this complex subproblem, we developed labeling algorithms but none of them were efficient for the general case. Therefore, in what follows, we only present a labeling algorithm for the adjacent case as no computational results for the type II SP reformulation will be reported for the general case.

4.2.1. A labeling algorithm for the adjacent case. We consider the network G with the arc costs (44). Feasible routes are represented by paths in G which start and end at node 0. This node is, therefore, considered as the source and the sink node of G .

In a labeling algorithm (see Irnich and Desaulniers 2005), any partial path starting from the source node and ending at any node $i \in V$ is represented by a vector of information (the state) called a label. This label is attached to the partial path end node. The algorithm starts with an initial label attached to the source node and extends the labels (partial paths) along the arcs using extension functions until reaching the sink node. To avoid enumerating all feasible partial paths, those ending at the same node are compared using a dominance rule, and partial paths that cannot yield to an optimal source-to-sink path are discarded. Let us specialize this algorithm for our subproblem.

Let $p = (0, v_1, \dots, v_{h-1}, v_h = i)$ be a partial route (path) ending at node $i \in V$. We denote by $V(p)$ and $A(p)$ the set of nodes and arcs in p , respectively. Route p is represented by a label $E(p) = (Z(p), L(p), M^1(p), \dots, M^n(p), N(p))$ whose components are defined as follows.

• $Z(p)$: Incomplete reduced cost of route p . According to (43), the reduced cost of partial route p is not well defined because of the quadratic term which depends on arcs to be traversed in an extension of p towards the sink node. Therefore, we define its incomplete reduced cost as:

$$Z(p) = \sum_{(k,l) \in A(p)} \tilde{c}_{kl}^\alpha + \sum_{(k,l) \in A(p)} \sum_{(l,s) \in A(p)} 2\mathcal{C}_{klls}^\alpha; \quad (45)$$

- $L(p)$: Cumulated load in route p ;
- $M^l(p)$: Binary value indicating whether customer node $l \in V_0$ can still be visited in an extension of route p . A node $l \in V_0$ cannot be visited anymore if $L(p) + q_l > Q$ or $l \in V(p)$. In this case, $M^l(p) = 1$;
- $N(p) = v_{h-1}$: Next-to-last node in route p .

In the initial label E_0 , all components are set to 0.

Label $E(p)$ can be extended along an arc $(i, j) \in A$ if $M^j(p) = 0$. If this is the case, the extension produces a route $\bar{p} = (0, v_1, \dots, v_{h-1}, v_h = i, v_{h+1} = j)$ represented by a new label $E(\bar{p}) = (Z(\bar{p}), L(\bar{p}), M^1(\bar{p}), \dots, M^n(\bar{p}), N(\bar{p}))$ whose components are computed using the following extension functions:

$$\begin{aligned} Z(\bar{p}) &= \begin{cases} Z(p) + \tilde{c}_{ij}^\alpha & \text{if } i = 0 \\ Z(p) + \tilde{c}_{ij}^\alpha + 2\mathcal{C}_{N(p)ij}^\alpha & \text{otherwise,} \end{cases} \\ L(\bar{p}) &= L(p) + q_j, \\ M^l(\bar{p}) &= \begin{cases} 1 & \text{if } L(\bar{p}) + q_l > Q \text{ or } l \in V(\bar{p}) \\ 0 & \text{otherwise} \end{cases} \quad \forall l \in V_0, \\ N(\bar{p}) &= i. \end{aligned}$$

Note that, if $j = 0$, then $Z(\bar{p})$ is equal to the complete reduced cost (43) of route p .

To avoid enumerating all feasible routes, dominance is performed as follows. Let p and p' be two feasible partial routes ending at the same node $i \in V$ and represented by the labels $E(p) = (Z(p), L(p), M^1(p), \dots, M^n(p), N(p))$ and $E(p') = (Z(p'), L(p'), M^1(p'), \dots, M^n(p'), N(p'))$, respectively. Label $E(p)$ is said to dominate label $E(p')$ if

C1: any feasible extension of p' along a sequence of arcs is also feasible for p , and

C2: for any such extension $\chi = (w_0 = i, w_1, \dots, w_g)$, $Z(p \oplus \chi) \leq Z(p' \oplus \chi)$ holds, where $Z(p \oplus \chi)$ (resp. $Z(p' \oplus \chi)$) is the incomplete reduced cost of the path $p \oplus \chi$ (resp. $p' \oplus \chi$) resulting from the extension of p (resp. p').

Given the exponential number of feasible extensions, conditions **C1** and **C2** cannot be easily verified. We rather propose the following sufficient conditions. For a partial route p ending at node $i \in V$, we first find the arcs $(i, j) \in A$ that are the most and less correlated with $(N(p), i)$ and record the following values:

$$f^-(p, i) = \min_{(i,j) \in \delta^+(i): M^j(p)=0} \{\mathcal{C}_{N(p)ij}^\alpha\}, \quad (46)$$

$$f^+(p, i) = \max_{(i,j) \in \delta^+(i): M^j(p)=0} \{C_{N(p)ij}^\alpha\}. \quad (47)$$

The sufficient conditions of the dominance rule are stated in the following theorem.

THEOREM 2. *Given two partial routes p and p' ending at the same node $i \in V$ and represented by the labels $E(p) = (Z(p), L(p), M^1(p), \dots, M^n(p), N(p))$ and $E(p') = (Z(p'), L(p'), M^1(p'), \dots, M^n(p'), N(p'))$, respectively. Label $E(p)$ dominates label $E(p')$ if*

$$Z(p) + 2f^+(p, i) \leq Z(p') + 2f^-(p', i), \quad (48)$$

$$L(p) \leq L(p'), \quad (49)$$

$$M^l(p) \leq M^l(p'), \quad \forall l \in V_0. \quad (50)$$

Proof. See Appendix B.

Note that, when $N(p) = N(p')$, the condition (48) can be replaced by $Z(p) \leq Z(p')$. This is correct because in any common extension, the unknown contribution of the quadratic term to the reduced cost would be the same.

In the labeling algorithm, dominated labels are discarded unless two labels dominate each other. In this case, one of the labels is kept.

4.3. Route relaxation, rounded capacity cuts, and branching

In this section, we present the ng -route relaxation as an acceleration technique, the rounded capacity cuts applied to strengthen the continuous relaxations in the branch-and-bound search tree, and the branching strategies to derive integer solutions.

Solving the ESPPRC subproblems is the most time-consuming part of the proposed branch-price-and-cut algorithms. To reduce the computational time spent solving the subproblems, we use the ng -route relaxation introduced by Baldacci et al. (2011). For each node $i \in V$, let $\mathcal{V}_i \subset V$ be a subset of nodes with a priori fixed size (set to 10 in our experiments) that contains node i and its “closest” neighbors. In an ng -route, a customer $i \in V_0$ may be visited more than once if between any two visits to i at least one node $j \in V$ such that $i \notin \mathcal{V}_j$ is visited. With this relaxation, the parameters a_{ip} and b_{ijp} are not binary anymore. They rather counts the number of times that node i and arc (i, j) are traversed in route p , yielding weaker lower bounds. Nevertheless, using this ng -route relaxation accelerates the overall solution process.

To strengthen the continuous relaxations encountered in the search tree, we generate rounded capacity cuts (see Naddef and Rinaldi 2001) dynamically. More precisely, we consider the inequalities (7) and (15) for the general case and the adjacent case, respectively, where $\gamma(S)$ is replaced by the lower bound $\lceil \sum_{i \in S} q_i / Q \rceil$. Then, we replace the corresponding x variables using (23) and (29) and add them to corresponding master problem whenever they are violated. Since the separation

of these inequalities is known to be strongly NP-hard (see, e.g., Naddef and Rinaldi 2001), we use the separation heuristic package of Lysgaard (2003) for our computational experiments.

To derive integer solutions, we branch on the total flow on an arc in A which is not incident to node 0. The total flow is computed by vehicle in model $SP1_\alpha$, that is, for each subproblem. When the arc flow is fractional for several arcs, we branch on the flow on an arc $e \in A$ which is not incident to node 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 A in all subproblems. On the other branch, the flow on e is set to 1 by removing from A all the other arcs with the same tail or head node as e . The columns of the current master problem are then updated accordingly. Finally, the enumeration process applies a best-first search strategy to explore the search tree.

5. Computational results

In this section, we present our computational experiments to evaluate empirically the performance of the proposed algorithms for both the general and the adjacent cases. Moreover, we analyze the efficiency of our proposed parametric models in finding efficient solutions in terms of both the expected travel time and the variance. For simplification, we denote by $BPC-SP1_\alpha$, $BPC-SP1_\alpha^{adj}$, $BPC-SP1_\alpha^{adjL}$, and $BPC-SP2_\alpha^{adj}$ the branch-price-and-cut algorithms applied to models $SP1_\alpha$, $SP1_\alpha^{adj}$, $SP1_\alpha^{adjL}$, and the adjacent case of $SP2_\alpha$, respectively.

In our experiments we tested our algorithms for different choices of $\alpha \in \{0.05, 0.1, 0.3, 0.5\}$. 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, using a single thread. We considered a time limit of 5 hours to solve each instance in the general case and 2 hours in the adjacent case.

To ensure reproducibility of the results produced by our best algorithms, the algorithms $BPC-SP1_\alpha$ for the general case and $BPC-SP2_\alpha^{adj}$ for the adjacent case are made available on the website <https://cgopt.gerad.ca>, which allows to run the instances described below or any instance of limited size (up to around 100 customers). [A user guide and the datasets used in this paper can be found in the online supplement to this paper.](#)

In the following, we describe our test instances in Section 5.1. Then, for the covariance matrices with non-negative entries, we analyze the performance of the proposed algorithms and investigate the effectiveness of our proposed models in terms of the expected travel times and the variance in Sections 5.2 and 5.3, respectively. In Section 5.4, we evaluate the algorithms and the solutions in the presence of negative correlations.

5.1. Test instances

To evaluate and compare the proposed algorithms, we used five groups of CVRP-SCT instances that were created by adapting well-known instances from the literature for the CVRP. Two groups of instances are derived from the A and P CVRP instances originally proposed by Augerat et al. (1998) (and available at <http://vrp.atd-lab.inf.puc-rio.br/index.php/en/new-instances>), where in group A, the customers and the depot are randomly positioned while in group P, they are randomly clustered. The other three groups of instances are derived from the instances of Solomon (1987) for the VRP with time windows (VRPTW) from which we discarded the time windows. In these instances, the geographical locations of the customers are: Clustered (C), Random (R) and a mixed of Random and Clustered (RC). Each instance contains 100 customers that are distributed in a 100×100 square. 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. The three groups contain the instances of various sizes derived from the C101, RC101, and R101 VRPTW instances, respectively. The other VRPTW instances in classes C1, R1, and RC1 were not considered because deleting the time windows make them identical to C101, R101, or RC101. For these instances, we set the number of vehicles $|K|$ to $\lceil \sum_{i \in V_0} q_i / Q \rceil$.

Next, we converted each of the CVRP instances into CVRP-SCT instances by generating suitable mean and covariance matrices. To this end, we set the expected travel time $\mu_{ij} = c_{ij}$, where c_{ij} is the cost of arc (i, j) given in the CVRP instances. To generate a positive semidefinite covariance matrix \mathcal{C} , we need to generate the standard deviations and the correlation coefficients (see equation (1)). Let $M \in \mathbb{R}^m$ be the vector of the standard deviations and $D \in [-1, 1]^{m \times m}$ be the correlation matrix. The covariance matrix can be obtained as $\mathcal{C} = (MM^T) \circ D$ where \circ is the ‘‘Hadamard product’’ operation. Note that, since MM^T is a rank one matrix, and hence positive semidefinite, the resulting covariance matrix \mathcal{C} is a positive semidefinite matrix too. In the following we explain how we generated vector M and matrix D .

- To generate the standard deviation of travel time on each arc (i, j) , we used a coefficient of variation (CV) which is equal to the ratio of the standard deviation to the mean and, thus, provides a measurement of the relative time dispersion (see, for instance, Dong et al. 2012). Given the expected travel time μ_{ij} for each arc $(i, j) \in A$, we compute $M = [\sigma_{ij}]_{(i,j) \in A}$ with $\sigma_{ij} = (CV)_{ij} \times \mu_{ij}$, where $(CV)_{ij}$ is drawn from a uniform distribution in the range $[0.01, 0.2]$.

- To generate the correlation matrix D , we first randomly generate an $m \times n$ full rank matrix \tilde{E} from a normal distribution with a relatively large positive mean and a small standard deviation whose rows and columns correspond to the arcs and the customers, respectively. Any generated negative number is discarded and replaced by another random value. Let E be the matrix resulting

Table 1 Summary results of $BPC-SP1_\alpha$ on the CRCR instances

Instance			$BPC-SP1_\alpha$					
Group	n	$ K $	$Gap_1(\%)$	$Gap_2(\%)$	$RT(s)$	$TT(s)$	Nodes	Opt
C101	15	2	11.7	2.9	10.1	108.5	28	4
C101	20	2	2.4	2.4	60.9	231.8	11	4
C101	23	3	2.8	1.8	230.6	2342.3	106	3
C101	25	3	3.4	1.8	297.0	2883.0	65	3
C101	28	3	3.4	2.2	932.7	6919.3	64	4
C101	30	3	4.3	2.5	1186.8	10752.0	71	3
C101	32	3	3.9	0.6	3056.3	17169.1	69	1
RC101	15	2	1.8	1.3	14.1	59.0	15	4
RC101	20	3	4.9	2.5	100.2	608.2	59	4
RC101	23	3	3.1	1.9	222.4	1885.4	73	4
RC101	25	3	2.0	1.1	361.2	2169.6	57	3
RC101	28	4	8.7	0.5	642.3	11551.1	363	1
RC101	30	4	6.6	0.4	1523.4	9753.5	130	2
RC101	32	4	4.2	0.3	2323.9	13603.0	131	1
R101	15	2	1.9	1.9	17.9	397.1	88	4
R101	20	2	3.0	2.8	83.3	1914.2	73	4
R101	23	2	2.6	2.3	265.1	3871.6	54	4
R101	25	2	2.8	1.7	480.1	5757.4	46	4
R101	28	2	1.3	0.9	1297.7	5129.4	14	3
R101	30	3	-	-	-	-	-	0
R101	32	3	-	-	-	-	-	0

from normalizing the rows of \tilde{E} to have length one. We then set $D = E E^T$ which is a positive semidefinite matrix and its entries lie in $[0, 1]$ because each row of E has length one. To generate negative correlations, we multiply by -1 each generated random number \tilde{E}_{ei} , $e \in A$ and $i \in V_0$, with probability 5%, resulting in a matrix D with all entries in $[-1, 1]$. For the adjacent case, we set $\tilde{E}_{ei} = 0$ for all arcs $e \in A$ and nodes $i \in V_0$ such that $e \notin (\delta^+(i) \cup \delta^-(i))$.

For the sake of conciseness, we will refer to the instances in groups A and P as the AP instances and those in the C101, RC101 and R101 as the CRCR instances.

All these instances and the best solutions found by the proposed algorithms are available at <https://cgopt.gerad.ca/directives> and in the online supplement.

5.2. Algorithmic performance

We conducted computational experiments to evaluate the performance of the proposed branch-price-and-cut algorithms considering only non-negative correlations. Summary computational results are reported in Section 5.2.1 for the general case and in Section 5.2.2 for the adjacent case. Detailed results can be found in the paper online supplement.

5.2.1. The general case. For the general case, we ran experiments with $BPC-SP1_\alpha$ on the CRCR instances with sizes ranging from 15 to 32 customers. Average results are reported in Table 1. In this table, each row gives the average results for the four values of $\alpha \in \{0.05, 0.1, 0.3, 0.5\}$,

Table 2 Aggregated results obtained by $BPC-SP1_{\alpha}^{adj}$ and $BPC-SP1_{\alpha}^{adjL}$ in the adjacent case

Instance		$BPC-SP1_{\alpha}^{adj}$						$BPC-SP1_{\alpha}^{adjL}$					
<i>Group</i>	<i>Total</i>	<i>Gap₁(%)</i>	<i>Gap₂(%)</i>	<i>RT(s)</i>	<i>TT(s)</i>	<i>Nodes</i>	<i>Opt</i>	<i>Gap₁(%)</i>	<i>Gap₂(%)</i>	<i>RT(s)</i>	<i>TT(s)</i>	<i>Nodes</i>	<i>Opt</i>
C101	32	2.9	0.5	463.5	2297.9	36	16	2.1	0.1	47.0	422.0	13	23
RC101	32	3.9	0.6	453.1	1657.8	39	20	2.5	0.0	38.8	39.5	1	24
R101	32	2.7	2.3	155.9	1448.0	61	13	1.3	1.0	31.2	609.6	47	17
A	76	3.0	1.3	369.2	4364.8	229	9	3.1	1.3	31.8	987.9	1658	56
P	60	0.9	0.6	1.1	17.2	72	12	1.7	1.1	50.4	1017.2	1133	31
Total	232						70						151

the average being computed only on the instances solved to optimality within the time limit. The first three columns indicate the instance group (*Group*), the number of customers (n), and the number of required vehicles ($|K|$). The next five columns give the average integrality gaps in percentage (*Gap₁*, and *Gap₂*) before and after applying rounded capacity cuts, the average CPU time in seconds at the root node of the search tree (*RT*), the average total time in seconds to solve the instances to optimality (*TT*), and the average number of explored nodes (*Nodes*). The last column specifies the number of instances solved within the time limit (*Opt*).

We can observe from these results that the lower bounds obtained at the root node are improved significantly by adding rounded capacity cuts. On average, the rounded capacity cuts decrease the integrality gap by 31%, which, in turn, implies smaller search trees. However, the time spent to solve the relaxation at the root node regardless of the number of added rounded capacity cuts (which are only a few) is relatively large and increases with the number of customers. Overall, the algorithm can solve instances with up to 32 customers within the 5-hour time limit. Moreover, the clustered C101 instances appear to be easier to solve than the RC101 and R101 instances. Indeed, the algorithm could solve to optimality 78.5%, 68.7%, and 68.7% of the tested instances in the groups C101, RC101, and R101, respectively.

5.2.2. The adjacent case. In this section, we compare the performances of the three algorithms $BPC-SP1_{\alpha}^{adj}$, $BPC-SP1_{\alpha}^{adjL}$, and $BPC-SP2_{\alpha}^{adj}$ for the adjacent case. To do this comparison, we ran tests on CRCR instances involving between 20 and 65 customers and on AP instances with up to 75 customers.

We first observe that, generally speaking, $BPC-SP1_{\alpha}^{adj}$ is not very efficient. This can be easily inferred by inspecting Table 2 which reports aggregated results by group of instances obtained by $BPC-SP1_{\alpha}^{adj}$, and $BPC-SP1_{\alpha}^{adjL}$. In Table 2, the first two columns indicate the instance group (*Group*) and the total number of instances in this group (*Total*). For each algorithm, the description of the columns is the same as in Table 1. Again, the reported average results are computed over the instances solved to optimality within the time limit. As we can observe from these results, $BPC-SP1_{\alpha}^{adjL}$ clearly outperforms $BPC-SP1_{\alpha}^{adj}$ on all groups of instances. Overall, $BPC-SP1_{\alpha}^{adjL}$ could solve 151 out of the 232 instances within the time limit while $BPC-SP1_{\alpha}^{adj}$ was able to solve

Table 3 Comparison of the algorithms on the CRCR instances in the adjacent case

Instance			$BPC-SP1_{\alpha}^{adjL}$						$BPC-SP2_{\alpha}^{adj}$					
Group	n	$ K $	Gap ₁ (%)	Gap ₂ (%)	RT(s)	TT(s)	Nodes	Opt	Gap ₁ (%)	Gap ₂ (%)	RT(s)	TT(s)	Nodes	Opt
C101	20	2	0.0	0.0	5.8	5.8	1	4	0.0	0.0	22.9	23.1	1	4
C101	25	3	1.1	0.0	6.5	7.4	1	4	0.9	0.0	7.8	8.6	1	4
C101	30	3	2.0	0.0	14.6	16.4	1	4	1.8	0.0	6.3	8.7	1	4
C101	35	4	3.9	0.6	27.7	237.5	63	4	3.8	0.4	3.6	37.0	40	4
C101	40	4	3.9	0.2	49.8	1981.6	7	4	3.8	0.2	7.3	71.1	5	4
C101	50	5	1.9	0.0	138.6	153.1	1	2	1.7	0.0	21.0	42.0	1	4
C101	60	6	0.8	0.0	386.1	405.6	1	1	0.8	0.1	15.7	58.4	1	4
C101	65	6	-	-	-	-	-	0	-	-	-	-	-	0
Solved(%)								71.8						87.5
RC101	20	3	1.7	0.0	1.1	1.2	1	4	1.7	0.0	0.7	0.8	1	4
RC101	25	3	0.6	0.0	5.2	5.4	1	4	0.6	0.0	0.8	1.4	1	4
RC101	30	4	6.1	0.0	8.5	8.9	1	4	6.1	0.0	0.9	1.8	1	4
RC101	35	4	3.7	0.0	26.1	27.4	1	4	3.6	0.0	6.0	10.4	1	4
RC101	40	5	2.8	0.0	37.1	38.4	1	4	2.7	0.0	1.7	4.0	1	4
RC101	50	5	0.3	0.0	154.9	155.9	1	4	0.2	0.0	18.4	23.4	1	4
RC101	60	6	-	-	-	-	-	0	-	-	-	-	-	0
RC101	65	6	-	-	-	-	-	0	-	-	-	-	-	0
Solved(%)								75.0						75.0
R101	20	2	0.8	0.6	7.7	8.2	5	4	0.8	0.5	55.1	56.0	5	4
R101	25	2	1.4	1.0	33.1	45.2	21	4	1.2	0.6	521.7	583.4	16	4
R101	30	3	0.8	0.6	19.4	283.0	11	4	0.7	0.6	12.9	18.9	9	4
R101	35	3	2.2	1.6	46.5	838.6	145	4	2.0	1.3	72.1	277.5	110	4
R101	40	3	1.6	1.1	102.9	5662.5	51	1	1.9	1.2	79.4	215.4	40	4
R101	50	4	-	-	-	-	-	0	2.3	1.1	110.1	607.2	166	4
R101	60	5	-	-	-	-	-	0	2.0	1.0	65.7	544.4	27	1
R101	65	5	-	-	-	-	-	0	1.5	0.7	155.2	1337.5	65	1
Solved(%)								53.1						81.5
Total solved (%)								66.3						81.3

only 70 of them. One possible reason for such a superiority is that the linear programming (LP) relaxations of $SP1_{\alpha}^{adjL}$ provide stronger bounds and are solved more quickly than the continuous relaxations of $SP1_{\alpha}^{adj}$.

Next, we analyze more in depth the behavior of $BPC-SP1_{\alpha}^{adjL}$ and $BPC-SP2_{\alpha}^{adj}$. The results are reported in Tables 3 and 4 for the CRCR instances and the AP instances, respectively. The meaning of the columns is the same as in Table 1. Each row in these tables gives the average results on the 4 values of $\alpha \in \{0.05, 0.1, 0.3, 0.5\}$, except for the solved rows which provide, by instance group and overall, the percentage of instances solved within the time limit. By inspecting these tables, we can observe that $BPC-SP2_{\alpha}^{adj}$ generally outperforms $BPC-SP1_{\alpha}^{adjL}$ in terms of the number of solved instances and total computing times. In fact, using $BPC-SP2_{\alpha}^{adj}$, we were able to solve CRCR instances with up to 65 customers and AP instances with up to 75 customers. Overall, $BPC-SP2_{\alpha}^{adj}$ found optimal solutions for 81.3% of the CRCR instances and 79.1% of the AP instances, while these percentages for $BPC-SP1_{\alpha}^{adjL}$ are 66.3% and 59.4%, respectively. As it can be observed, the main reason for such a superiority is that the LP relaxation of $SP2_{\alpha}$ is solved much more quickly than the LP relaxation of $SP1_{\alpha}^{adjL}$.

Table 4 Comparison of the algorithms on the AP instances in the adjacent case

Instance			$BPC-SP1_{\alpha}^{adjL}$						$BPC-SP2_{\alpha}^{adj}$						
Group	n	K	Gap ₁ (%)	Gap ₂ (%)	RT(s)	TT(s)	Nodes	Opt	Gap ₁ (%)	Gap ₂ (%)	RT(s)	TT(s)	Nodes	Opt	
A	32	5	2.5	0.9	10.0	28.7	128	4	2.4	0.9	0.3	8.1	96	4	
A	32	6	3.3	0.6	7.2	16.6	46	4	3.1	0.6	0.3	4.7	48	4	
A	33	5	4.8	2.4	11.8	403.9	2459	2	4.7	2.8	0.7	3689.9	38727	4	
A	35	5	3.4	1.6	18.4	760.6	2092	4	3.2	1.5	1.6	361.5	1771	4	
A	36	5	2.7	1.2	21.8	457.4	253	4	2.6	1.2	1.7	44.0	233	4	
A	36	6	3.5	2.1	14.7	2616.7	10112	4	3.4	1.9	0.7	705.9	5677	4	
A	37	5	6.4	3.0	27.7	1665.8	5214	4	6.3	2.9	1.2	691.5	4748	4	
A	38	5	2.3	1.2	33.8	1492.2	1814	4	2.1	1.0	1.6	598.0	1616	4	
A	38	6	3.4	1.6	21.2	2540.0	6625	4	3.3	1.3	0.9	773.1	4987	4	
A	43	6	1.1	0.6	45.3	1404.8	67	4	0.9	0.5	1.2	17.1	47	4	
A	44	6	3.6	0.8	62.5	1906.8	80	4	3.5	0.6	2.4	39.2	26	4	
A	44	7	2.6	1.5	48.0	813.3	1733	1	2.4	1.4	1.2	1760.5	7281	2	
A	45	7	1.6	0.3	56.3	95.2	42	4	1.5	0.3	1.8	12.4	32	4	
A	47	7	3.3	1.1	75.4	1105.5	1387	2	3.7	1.6	2.0	3454.3	4633	4	
A	52	7	3.0	2.0	127.8	5948.6	5605	1	2.2	1.3	6.7	3526.6	5341	4	
A	53	7	-	-	-	-	-	0	2.9	0.8	3.8	184.7	107	1	
A	54	9	4.8	1.5	109.3	6850.7	9393	1	4.7	1.3	0.9	461.1	2071	1	
A	59	9	-	-	-	-	-	0	-	-	-	-	-	0	
A	60	9	-	-	-	-	-	0	2.7	1.2	2.2	2379.6	3044	2	
Solved(%)								67.1							81.5
P	15	8	1.6	0.7	0.0	0.1	17	4	1.6	0.7	0.0	0.1	15	4	
P	21	8	0.4	0.2	0.3	0.5	9	4	0.3	0.0	0.0	0.2	1	4	
P	22	8	-	-	-	-	-	0	0.0	0.0	0.0	0.2	1	4	
P	39	5	1.9	1.4	37.0	173.8	356	4	1.7	1.3	1.5	34.3	272	4	
P	44	5	2.3	1.8	70.4	1774.9	3246	4	2.3	1.7	3.0	375.4	2882	4	
P	49	7	1.8	1.3	66.8	1478.0	2105	4	1.8	1.3	0.8	226.2	1759	4	
P	49	8	-	-	-	-	-	0	-	-	-	-	-	0	
P	49	10	1.5	1.2	35.8	880.5	1939	2	1.4	1.1	0.2	399.7	8215	4	
P	50	10	1.4	0.9	46.9	895.9	1767	4	1.3	0.8	0.2	111.8	1598	4	
P	54	7	2.0	0.9	127.1	327.0	107	1	2.1	1.2	1.7	2264.3	6490	2	
P	54	8	1.9	1.2	148.8	4318.3	1543	4	1.7	0.9	1.5	136.2	580	4	
P	54	10	-	-	-	-	-	0	2.4	1.6	0.4	2706.6	26735	2	
P	59	10	-	-	-	-	-	0	1.6	1.1	0.6	1688.6	10573	4	
P	69	10	-	-	-	-	-	0	-	-	-	-	-	0	
P	75	4	-	-	-	-	-	0	1.6	0.8	173.3	3711.6	370	2	
Solved(%)								51.6							76.7
Total solved (%)								59.4							79.1

Table 5 Solution analysis on the CRCR instances in the general case

Instance	$\alpha = 0.05$		$\alpha = 0.1$		$\alpha = 0.3$		$\alpha = 0.5$	
	$\uparrow Exp(\%)$	$\downarrow Var(\%)$	$\uparrow Exp(\%)$	$\downarrow Var(\%)$	$\uparrow Exp(\%)$	$\downarrow Var(\%)$	$\uparrow Exp(\%)$	$\downarrow Var(\%)$
C101	1.3	69.8	2.2	74.0	5.5	85.0	9.8	87.4
RC101	0.4	35.5	1.5	50.0	4.5	64.1	9.9	72.7
R101	3.4	54.9	15.9	70.6	31.7	81.3	37.1	83.4

5.3. Solution analysis in terms of expected travel time and variance

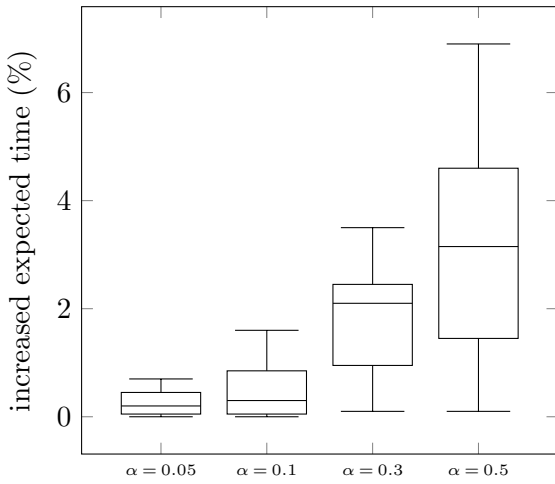
In this section, we investigate the solutions that can be computed using different choices of α , considering again only non-negative correlations. For $\alpha = 0$, the CVRP-SCT reduces to the CVRP, which minimizes the expected total travel time without considering time variability. This can only be interesting for a planner who has a risk-neutral behavior when planning the operations. However, by increasing the value of α , the chosen model can capture the planner's risk attitude to control time variability and yield different routing solutions.

We analyze the solutions from two different perspectives: increased expected travel time ($\uparrow Exp$) with respect to the solutions of the CVRP and the decreased variance ($\downarrow Var$) with respect to the solutions of the CVRP. We use $(Exp(\alpha) - Exp(0))/Exp(\alpha)$ to compute $\uparrow Exp$, and $(Var(0) - Var(\alpha))/Var(0)$ to compute $\downarrow Var$, where $Exp(\alpha)$ and $Var(\alpha)$ are the expected travel time and variance of the optimal routes for a given α . To solve the instances, we used $BPC-SP1_\alpha$ for the general case and $BPC-SP2_\alpha^{adj}$ for the adjacent case. Table 5 reports the average results on the CRCR instances for the general case, while Figures 1 and 2 report the detailed results on the CRCR and the AP instances for the adjacent case. We report the results only for instances solved to optimality within the time limit for the values of $\alpha \in \{0.05, 0.1, 0.3, 0.5\}$. Detailed results for both the general case and the adjacent case can be found in the paper supplement.

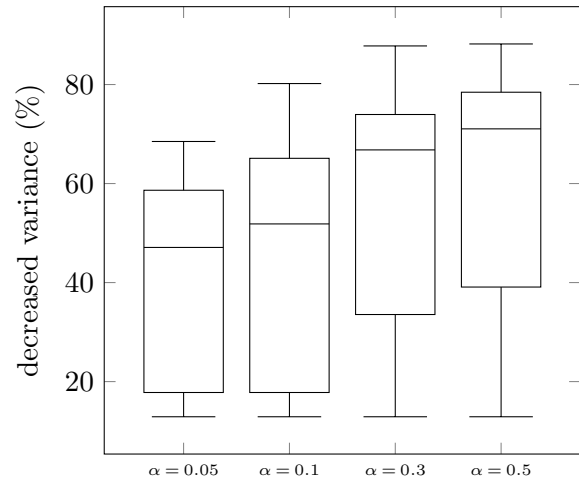
In Table 5, the first column indicates the instance group (*Group*). The next columns represent for each $\alpha > 0$ the average increase in percentage of the expected travel time ($\uparrow Exp(\%)$) and the average decrease in percentage of the total variance ($\downarrow Var(\%)$) with respect to the expected travel time and variance of the optimal routes of the CVRP. Each row gives the average results on the instances solved to optimality within the time limit. As we can observe, solving the CVRP-SCT with $\alpha = 0.05$ can reduce the travel time variance in all three groups C101 (by 69.8%), RC101 (by 35.5%), and R101 (by 54.9%) with a respective increase in the total expected travel time of 1.3%, 0.4% and 3.4%. Even though the behavior of the algorithm on each group of instances differs, the overall results show that the CVRP-SCT can yield solutions with a considerably smaller variance at the expense of a slightly higher expected travel time.

Figures 1 and 2 show the results of the solution analysis on the CRCR and the AP instances in the adjacent case, respectively. For the CRCR instances, the overall behavior of the obtained solutions is similar to the general case for all values of α . For example, for $\alpha = 0.05$, the average decreased variance and increased expected travel time of the CVRP-SCT solutions for the three groups C101, RC101, and R101 are (45.1%, 0.3%), (29.1%, 0.03%), and (38.7%, 1.1%), respectively. Note that for the RC101 instances (part *c* of Figure 1), the routes computed for $\alpha = 0.05$ exhibit the same total expected travel time as in the CVRP in 5 of the 6 instances with an average reduction of 32% of the total variance (part *d* of Figure 1). As expected, by increasing α , we find solutions with smaller variances and higher expected travel times.

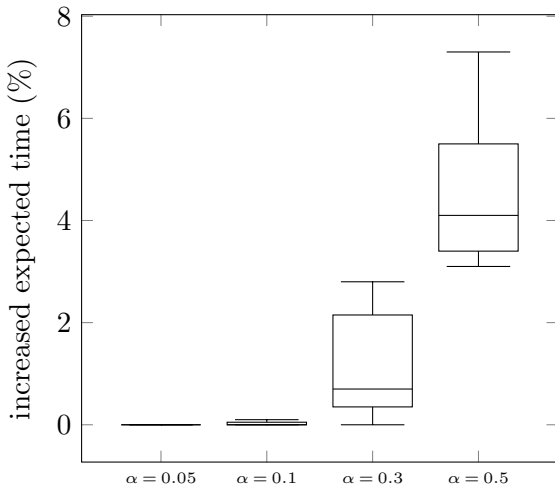
For the AP instances, the results are more interesting (see Figure 2). The CVRP-SCT solution computed with $\alpha = 0.05$ has on average the same total expected travel time as its corresponding CVRP solution, but much less total variance. This highlights the impact of using the CVRP-SCT to find alternative optimal routes with much less variance. The average decrease of the total variance and increase of the total expected travel time with respect to the CVRP optimal solutions for the A instances are (33.8%, 0.07%), (43.1%, 0.22%), (56.7%, 1.08%), and (67.0%, 2.63%) when $\alpha = 0.05$,



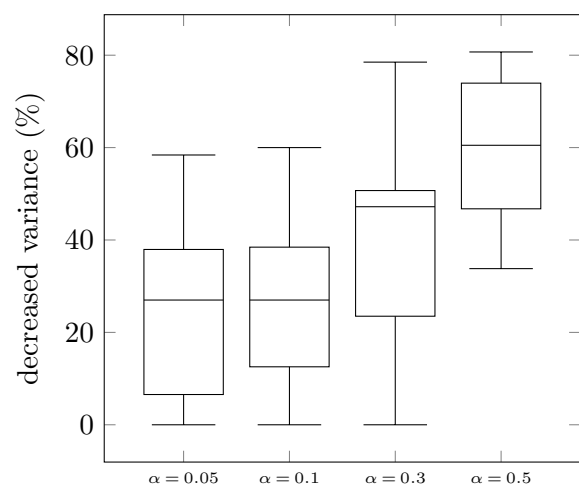
(a) C101: increased expected travel time



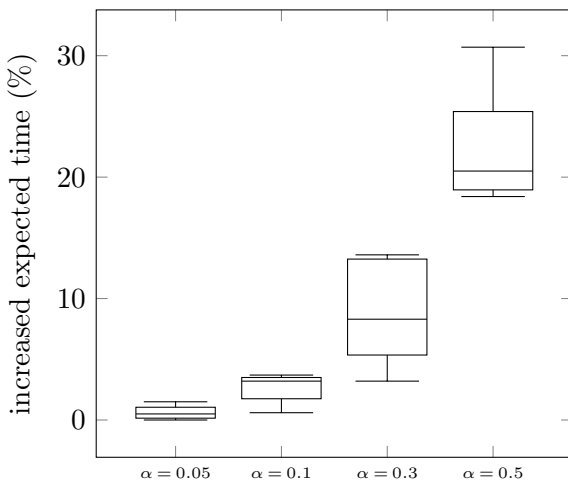
(b) C101: decreased variance



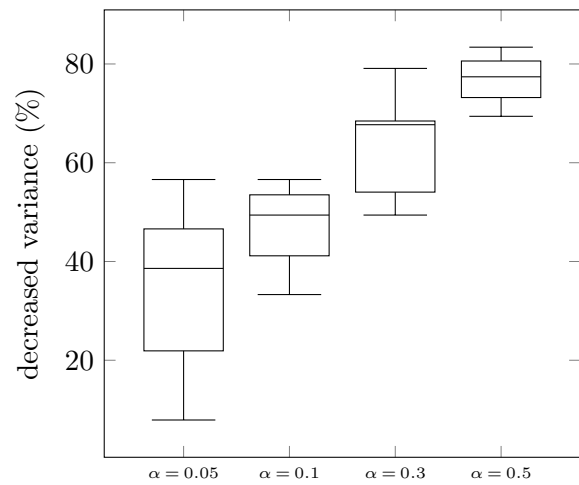
(c) RC101: increased expected travel time



(d) RC101: decreased variance



(e) R101: increased expected travel time



(f) R101: decreased variance

Figure 1 increased expected travel time and decreased variance with respect to the solutions of the CVRP (CRCR instances, adjacent case)

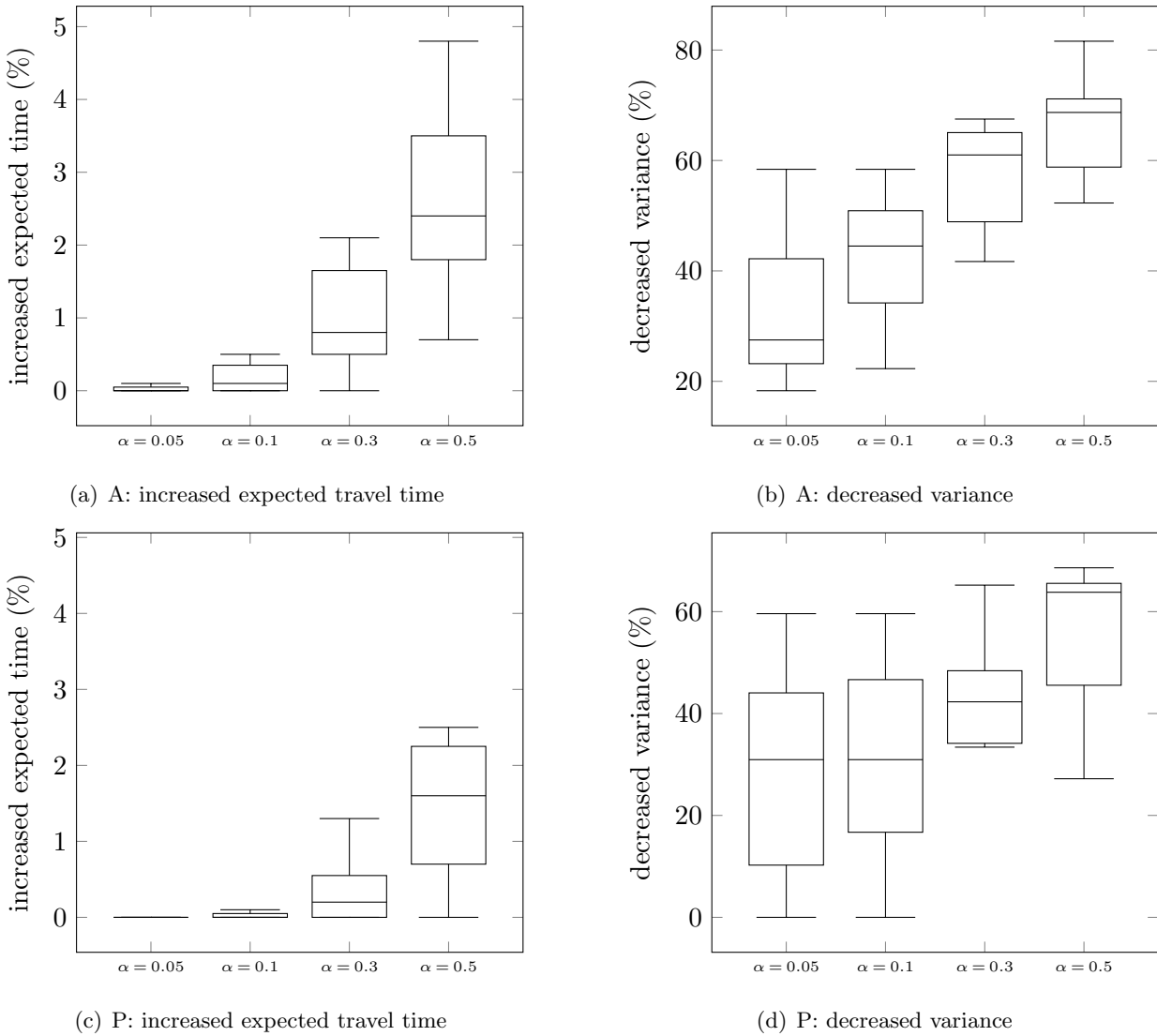


Figure 2 increased expected travel time and decreased variance with respect to the solutions of the CVRP (AP instances, adjacent case)

$\alpha = 0.1$, $\alpha = 0.3$, and $\alpha = 0.5$, respectively. For the group P, we get (30.4%, 0.01%), (33.0%, 0.04%), (42.4%, 0.4%), and (57.3%, 1.8%) for $\alpha = 0.05$, $\alpha = 0.1$, $\alpha = 0.3$, and $\alpha = 0.5$, respectively. We observe again that larger α values yield solutions with much less variability.

Overall, the results indicate the flexibility offered by the CVRP-SCT to compute routes with different values of the total expected travel time and variance, which in turn allows the planner to choose routes according to his risk attitude. Moreover, our results show that, in many cases, the CVRP-SCT provides solutions with the same total expected travel time as the CVRP solutions, but with considerably less variance.

5.4. Effects of negative correlations

In this section, we investigate the effects of negative correlation coefficients on the performance of the algorithms and the properties of the solutions. As mentioned in Section 1, it is possible to have negative correlations between the travel times. For example, negative correlation can occur when there is a bottleneck in one link that restricts the flow in the downstream links. The vehicles on the downstream links can then travel at full speed.

Our computational experiments show that negative correlation coefficients affect the performance of all the algorithms in a similar way. Therefore, here, we only present the results produced by $BPC-SP2_{\alpha}^{adj}$ in the adjacent case for a reduced number of instances, namely, the C101 and A instances. Tables 6 and 7 reports the results for $\alpha = 0.1$ and $\alpha = 0.3$, respectively. The first three columns of these tables give the instance information. The columns $\uparrow Exp$ and $\downarrow Var$ provide the increased expected travel time and the decreased variance of the CVRP-SCT solution compared to the CVRP one. The remaining columns give the same statistics as in Table 1.

As we can see from columns 4 and 5, the computed CVRP-SCT solutions exhibit significantly less travel time variance than the corresponding CVRP solutions, without increasing much the travel time mean, similarly to the positive-only correlation case. For example, for $\alpha = 0.1$, the average expected travel time increase and variance decrease of the CVRP-SCT solutions with respect to the CVRP solutions are 0.68% and 54.7% for the C101 instances, and 0.23% and 48.1% for the A instances against (0.57%, 49.9%) and (0.22%, 43.1%) for the case with only non-negative correlations. On the other hand, from the algorithmic performance point of view, we observe that the instances allowing negative correlations are more difficult to solve. This can be inferred by comparing the columns TT in Tables 6 and 7 with the corresponding ones in Tables 17, 20, and 21. They indicate an increase of the total computing time by average factors of 41.5 and 7.5 for the C101 instances with $\alpha = 0.1$ and $\alpha = 0.3$, respectively, and of 0.9 and 4.2 for the A instances with $\alpha = 0.1$ and $\alpha = 0.3$, respectively. In particular, when $\alpha = 0.3$, $BPC-SP2_{\alpha}^{adj}$ was unable to solve within the time limit C101 instances with more than 30 customers, while instances with up to 60 customers were solved in less than three minutes when considering only non-negative correlations.

6. Conclusions

In this paper, we studied the CVRP-SCT, where correlations between arc travel times are represented by a variance-covariance matrix. To reduce time variability, we introduced a mean-variance approach and considered two cases, namely, the general case where travel time correlation can be observed between any pair of arcs, and the adjacent case where correlation occurs only between adjacent arcs. Exploiting the structure of the covariance matrix in the latter case leads to more

Table 6 Results with negative correlations in the adjacent case for $\alpha = 0.1$

Instance			$\uparrow Exp(\%)$	$\downarrow Var(\%)$	$Gap_1(\%)$	$Gap_2(\%)$	$RT(s)$	$TT(s)$	$Nodes$
<i>Group</i>	n	$ K $							
C101	20	2	0.1	12.9	0.0	0.0	6.3	6.4	1
C101	25	3	1.6	82.5	1.3	0.0	2.3	5.4	1
C101	30	3	0.4	52.9	1.4	0.0	12.9	15.4	1
C101	35	4	0.9	73.5	3.7	0.7	4.6	674.4	65
C101	40	4	0.8	58.5	4.8	0.0	13.2	478.1	1
C101	50	5	0.8	49.2	1.9	0.0	166.7	1360.9	1
C101	60	6	0.2	53.4	0.9	0.0	1899.7	4979.2	1
Avg.			0.7	54.7	2.0	0.1	300.1	1074.2	10
A	31	5	0.3	55.2	2.7	0.8	1.0	8.0	17
A	32	5	0.0	30.1	1.8	0.8	0.4	12.2	131
A	32	6	0.3	64.8	3.8	0.8	0.3	17.8	223
A	33	5	0.3	33.2	5.3	3.6	0.7	6427.9	61183
A	35	5	0.0	32.3	3.0	1.2	1.5	597.2	2323
A	36	5	0.1	44.5	2.5	1.1	2.6	61.5	277
A	36	6	0.3	46.9	3.6	1.7	0.9	1747.0	11669
A	37	5	0.1	54.1	6.3	3.0	1.3	1892.3	9925
A	38	5	0.5	52.2	2.7	1.5	2.1	1253.9	2161
A	38	6	0.5	43.1	4.1	2.2	1.1	1947.4	9457
A	43	6	0.0	44.2	0.8	0.5	1.5	28.5	87
A	44	6	0.4	56.7	3.6	1.0	2.9	165.1	239
A	45	7	0.3	52.0	1.7	0.5	1.8	24.9	59
A	47	7	0.2	64.1	3.1	1.1	1.6	1023.8	1781
Avg.			0.2	48.1	3.2	1.4	1.4	1086.2	7109

Table 7 Results with negative correlations in the adjacent case for $\alpha = 0.3$

Instance			$\uparrow Exp(\%)$	$\downarrow Var(\%)$	$Gap_1(\%)$	$Gap_2(\%)$	$RT(s)$	$TT(s)$	$Nodes$
<i>Group</i>	n	$ K $							
C101	20	2	0.1	12.9	0.0	0.0	16.0	16.1	1
C101	25	3	3.5	89.0	0.8	0.0	20.5	53.3	1
C101	30	3	2.5	66.4	2.4	0.0	157.3	1299.8	1
Avg.			2.0	56.1	1.1	0.0	64.6	456.4	1
A	31	5	1.8	69.6	3.4	2.5	1.1	92.8	379
A	32	5	0.5	39.3	2.5	0.9	0.5	6.8	25
A	32	6	1.1	75.9	2.5	0.7	0.4	6.0	31
A	33	5	2.0	71.6	4.6	1.9	0.9	184.4	1155
A	35	5	1.5	53.1	3.2	1.9	2.2	1919.3	2795
A	36	5	2.2	70.6	2.3	1.6	3.0	279.3	341
A	36	6	1.2	64.5	3.3	2.0	1.1	1105.9	2129
A	37	5	1.1	69.0	6.5	3.1	1.5	2251.5	5121
A	38	5	2.1	67.2	2.2	0.9	3.8	385.8	53
A	38	6	1.7	67.3	2.5	0.6	1.7	53.8	39
A	43	6	0.0	44.2	0.4	0.0	1.6	7.4	1
A	44	6	0.5	59.4	3.4	0.3	4.0	147.1	17
A	45	7	1.4	69.7	1.2	0.2	2.8	18.8	7
A	47	7	0.6	68.3	3.5	1.2	3.4	4130.8	861
Avg.			1.3	63.6	3.0	1.27	2.0	756.4	925

tractable models. We proposed two types of set partitioning formulations and developed branch-price-and-cut algorithms to solve them. Our computational experiments demonstrated the efficiency of the proposed algorithms, especially in the adjacent case, for which we succeeded to solve instances with up to 75 customers. Moreover, we showed that solving the CVRP-SCT can yield

routes with a total expected travel time slightly larger than the one of the routes computed by solving the CVRP, but with significantly less variance.

Several future research avenues stemming from this work can be considered. One that we hope to consider in a near future is to design a heuristic that can tackle efficiently larger instances of the CVRP-SCT. In particular, we believe that the proposed branch-price-and-cut algorithm can be turned into an efficient approximation algorithm by suitably selecting a subset of the covariance coefficients to take into account.

Appendix

A. Proof of Theorem 1

We have to prove that for any feasible solution $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ of P_α^{adjL}

$$\hat{y}_{ijl} = \hat{x}_{ij}\hat{x}_{jl}, \quad \forall (i, j), (j, l) \in A \text{ with } j \in V_0 \quad (51)$$

and that $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ satisfies (20).

First note that the feasibility of $\hat{\mathbf{x}}$ for P_α^{adjL} implies that $\sum_{(j,l) \in \delta^+(j)} x_{jl} = 1$ for each customer $j \in V_0$. Now consider the value \hat{y}_{ijr} for an arc pair $(i, j), (j, r) \in A$ with $j \in V_0$. The constraint (18) associated with (i, j) together with the nonnegativity restriction $\hat{y}_{ijr} \geq 0$ of (19) enforce $\hat{y}_{ijr} \leq \hat{x}_{ij}$. Similarly, constraint (17) for (i, j) and $\hat{y}_{ijr} \geq 0$ enforce $\hat{y}_{ijr} \leq \hat{x}_{jr}$. Consequently,

$$\hat{y}_{ijr} \leq \min \{ \hat{x}_{ij}, \hat{x}_{jr} \}, \quad (52)$$

that is, $\hat{y}_{ijr} = 0$ if either $\hat{x}_{ij} = 0$ or $\hat{x}_{jr} = 0$. In addition, if we subtract constraint $\sum_{(j,l) \in \delta^+(j)} \hat{x}_{jl} = 1$ from equation (18) associated with (i, j) , we obtain

$$\sum_{(j,l) \in \delta^+(j)} (\hat{y}_{ijl} - \hat{x}_{jl}) = \hat{x}_{ij} - 1. \quad (53)$$

From (52) applied to any pair of arcs $(i, j), (j, l)$ with $j \in V_0$, we deduce that $\hat{y}_{ijl} - \hat{x}_{jl} \leq 0$ and, thus, $\hat{y}_{ijr} - \hat{x}_{jr} \geq \sum_{(j,l) \in \delta^+(j)} (\hat{y}_{ijl} - \hat{x}_{jl})$. Combining this inequality with (53) yields

$$\hat{y}_{ijr} \geq \hat{x}_{ij} + \hat{x}_{jr} - 1, \quad (54)$$

giving us that $\hat{y}_{ijr} = 1$ if $\hat{x}_{ij} = \hat{x}_{jr} = 1$. Finally, (52) and (54) imply $\hat{y}_{ijr} = \hat{x}_{ij}\hat{x}_{jr}$.

B. Proof of Theorem 2

From (49), (50) and the extension functions of the M and L components of a label, condition **C1** can be easily proved. Now, we need to show that condition **C2** is also satisfied for a given extension $\chi = (w_0 = i, w_1, \dots, w_g = j)$ ending at node $j \in V$ where $V(\chi)$ and $A(\chi)$ denote its node and arc sets, respectively. We find

$$\begin{aligned}
Z(p \oplus \chi) &= \sum_{(k,l) \in A(p \oplus \chi)} \tilde{c}_{kl}^\alpha + \sum_{(k,l) \in A(p \oplus \chi)} \sum_{(l,s) \in A(p \oplus \chi)} 2C_{klls}^\alpha \\
&= \sum_{(k,l) \in A(p)} \tilde{c}_{kl}^\alpha + \sum_{(k,l) \in A(p)} \sum_{(l,s) \in A(p)} 2C_{klls}^\alpha + \sum_{(k,l) \in A(\chi)} \tilde{c}_{kl}^\alpha + \sum_{(k,l) \in A(\chi)} \sum_{(l,s) \in A(\chi)} 2C_{klls}^\alpha + 2C_{N(p)iiw_1}^\alpha \\
&\leq Z(p) + \sum_{(k,l) \in A(\chi)} \tilde{c}_{kl}^\alpha + \sum_{(k,l) \in A(\chi)} \sum_{(l,s) \in A(\chi)} 2C_{klls}^\alpha + 2f^+(p, i) \\
&\leq Z(p') + \sum_{(k,l) \in A(\chi)} \tilde{c}_{kl}^\alpha + \sum_{(k,l) \in A(\chi)} \sum_{(l,s) \in A(\chi)} 2C_{klls}^\alpha + 2f^-(p', i) \\
&\leq Z(p') + \sum_{(k,l) \in A(\chi)} \tilde{c}_{kl}^\alpha + \sum_{(k,l) \in A(\chi)} \sum_{(l,s) \in A(\chi)} 2C_{klls}^\alpha + 2C_{N(p')iiw_1}^\alpha \\
&= Z(p' \oplus \chi),
\end{aligned}$$

where the first inequality follows from the definition of $f^+(p, i)$ in (46), the second from (48), and the third from the definition of $f^-(p', i)$ in (47). Since both conditions **C1** and **C2** are met, $E(p)$ dominates $E(p')$.

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.

References

- Augerat, P., Belenguer, J., Benavent, E., Corberán, A., Naddef, D. and Rinaldi, G. (1998). Computational results with a branch and cut code for the capacitated vehicle routing problem, *Technical Report R.495*, IASI-CNR.
- Bakach, I., Campbell, A. M., Ehmke, J. F. and Urban, T. L. (2018). Modeling correlation in vehicle routing problems with makespan objectives and stochastic travel times, *Working Paper Series*.
- 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.

- Balinski, M. L. and Quandt, R. E. (1964). On an integer program for a delivery problem, *Operations Research* **12**(2): 300–304.
- Barnhart, C., Johnson, E., Nemhauser, G., Savelsbergh, M. and Vance, P. (1998). Branch-and-price: Column generation for solving huge integer programs, *Operations Research* **46**(3): 316–329.
- Chen, B. Y., Lam, W. H., Sumalee, A. and Li, Z.-l. (2012). Reliable shortest path finding in stochastic networks with spatial correlated link travel times, *International Journal of Geographical Information Science* **26**(2): 365–386.
- Dantzig, G. B. and Wolfe, P. (1960). Decomposition principle for linear programs, *Operations Research* **8**(1): 101–111.
- Desaulniers, G., Desrosiers, J., Ioachim, I., Solomon, M. M., Soumis, F. c. and Villeneuve, D. (1998). A unified framework for deterministic time constrained vehicle routing and crew scheduling problems, in T. G. Crainic and G. Laporte (eds), *Fleet Management and Logistics*, Kluwer, Norwell, MA, pp. 57–93.
- Dong, W., Li, M., Vo, Q. B. and Vu, H. L. (2012). Reliability in stochastic time-dependent traffic networks with correlated link travel times, *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, IEEE, pp. 1626–1631.
- Errico, F., Desaulniers, G., Gendreau, M., Rei, W. and Rousseau, L.-M. (2016). A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times, *European Journal of Operational Research* **249**(1): 55–66.
- Errico, F., Desaulniers, G., Gendreau, M., Rei, W. and Rousseau, L.-M. (2018). The vehicle routing problem with hard time windows and stochastic travel times, *EURO Journal of Transportation and Logistics* **7**(3): 223–251.
- Fan, Y., Kalaba, R. and Moore, J. (2005). Shortest paths in stochastic networks with correlated link costs, *Computers & Mathematics with Applications* **49**(9): 1549–1564.
- 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.
- Fischer, A. and Helmberg, C. (2013). The symmetric quadratic traveling salesman problem, *Mathematical Programming* **142**(1): 205–254.
- Gendreau, M., Jabali, O. and Rei, W. (2014). Stochastic vehicle routing problems, in P. Toth and D. Vigo (eds), *Vehicle Routing: Problems, Methods, and Applications*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp. 213–239.
- Gendreau, M., Jabali, O. and Rei, W. (2016). Future research directions in stochastic vehicle routing, *Transportation Science* **50**(4): 1163–1173.
- Irnich, S. and Desaulniers, G. (2005). Shortest path problems with resource constraints, in G. Desaulniers, J. Desrosiers and M. M. Solomon (eds), *Column Generation*, Springer US, Boston, MA, pp. 33–65.

- Ji, Z., Kim, Y. S. and Chen, A. (2011). Multi-objective α -reliable path finding in stochastic networks with correlated link costs: A simulation-based multi-objective genetic algorithm approach (SMOGA), *Expert Systems with Applications* **38**(3): 1515 – 1528.
- Kenyon, A. S. and Morton, D. P. (2003). Stochastic vehicle routing with random travel times, *Transportation Science* **37**(1): 69–82.
- Lambert, V., Laporte, G. and Louveaux, F. (1993). Designing collection routes through bank branches, *Computers & Operations Research* **20**(7): 783–791.
- Laporte, G. (2009). Fifty years of vehicle routing, *Transportation Science* **43**(4): 408–416.
- Laporte, G., Louveaux, F. and Mercure, H. (1992). The vehicle routing problem with stochastic travel times, *Transportation science* **26**(3): 161–170.
- Lecluyse, C., Van Woensel, T. and Peremans, H. (2009). Vehicle routing with stochastic time-dependent travel times, *4OR: A Quarterly Journal of Operations Research* **7**(4): 363–377.
- Letchford, A. N. and Nasiri, S. D. (2015). The steiner travelling salesman problem with correlated costs, *European Journal of Operational Research* **245**(1): 62 – 69.
- Li, X., Tian, P. and Leung, S. C. (2010). Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm, *International Journal of Production Economics* **125**(1): 137–145.
- Lübbecke, M. E. and Desrosiers, J. (2005). Selected topics in column generation, *Operations Research* **53**(6): 1007–1023.
- Lysgaard, J. (2003). CVRPSEP: A package of separation routines for the capacitated vehicle routing problem.
URL: <http://econ.au.dk/research/researcher-websites/jens-lysgaard/cvrpsep>
- Markowitz, H. (1952). Portfolio selection, *The Journal of Finance* **7**(1): 77–91.
- Martinelli, R. and Contardo, C. (2015). Exact and heuristic algorithms for capacitated vehicle routing problems with quadratic costs structure, *INFORMS Journal on Computing* **27**(4): 658–676.
- Naddef, D. and Rinaldi, G. (2001). Branch-and-cut algorithms for the capacitated VRP, in P. Toth and D. Vigo (eds), *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp. 53–84.
- Nicholson, A. (2015). Travel time reliability benefits: Allowing for correlation, *Research in Transportation Economics* **49**: 14 – 21.
- Nie, Y. M. and Wu, X. (2009). Reliable a priori shortest path problem with limited spatial and temporal dependencies, in W. H. K. Lam, S. C. Wong and H. K. Lo (eds), *Transportation and Traffic Theory 2009: Golden Jubilee*, Springer US, Boston, MA, pp. 169–195.
- Oyola, J., Arntzen, H. and Woodruff, D. L. (2017). The stochastic vehicle routing problem, a literature review, part II: solution methods, *EURO Journal on Transportation and Logistics* **6**(4): 349–388.

- Parent, O. and LeSage, J. P. (2010). A spatial dynamic panel model with random effects applied to commuting times, *Transportation Research Part B: Methodological* **44**(5): 633–645.
- Pecin, D., Pessoa, A., Poggi, M. and Uchoa, E. (2017). Improved branch-cut-and-price for capacitated vehicle routing, *Mathematical Programming Computation* **9**: 61–100.
- Prakash, A. A. and Srinivasan, K. K. (2018). Pruning algorithms to determine reliable paths on networks with random and correlated link travel times, *Transportation Science* **52**(1): 80–101.
- Rachtan, P., Huang, H. and Gao, S. (2013). Spatiotemporal link speed correlations: Empirical study, *Transportation Research Record: Journal of the Transportation Research Board* **2390**: 34–43.
- Rostami, B., Malucelli, F., Belotti, P. and Gualandi, S. (2016). Lower bounding procedure for the asymmetric quadratic traveling salesman problem, *European Journal of Operational Research* **253**(3): 584 – 592.
- Rostami, B., Malucelli, F., Frey, D. and Buchheim, C. (2015). On the quadratic shortest path problem, in E. Bampis (ed.), *Experimental Algorithms*, Springer International Publishing, Cham, pp. 379–390.
- Samaranayake, S., Blandin, S. and Bayen, A. (2011). A tractable class of algorithms for reliable routing in stochastic networks, *Procedia - Social and Behavioral Sciences* **17**: 341 – 363.
- Seshadri, R. and Srinivasan, K. K. (2012). An algorithm for the minimum robust cost path on networks with random and correlated link travel times, in D. M. Levinson, H. X. Liu and M. Bell (eds), *Network Reliability in Practice*, Springer, New York, NY, pp. 171–208.
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints, *Operations Research* **35**(2): 254–265.
- Teng, S., Ong, H. L. and Huang, H. C. (2004). An integer l-shaped algorithm for time-constrained traveling salesman problem with stochastic travel and service times, *Asia-Pacific Journal of Operational Research* **21**(02): 241–257.
- Toth, P. and Vigo, D. (eds) (2014). *Vehicle Routing: Problems, Methods, and Applications*, MOS-SIAM Series on Optimization, 2nd edn, SIAM, Philadelphia, PA.
- Xing, T. and Zhou, X. (2011). Finding the most reliable path with and without link travel time correlation: A lagrangian substitution based approach, *Transportation Research Part B: Methodological* **45**(10): 1660–1679.
- Zadeh, L. (1963). Optimality and non-scalar-valued performance criteria, *IEEE Transactions on Automatic Control* **8**(1): 59–60.
- Zockaie, A., Nie, Y. and Mahmassani, H. (2014). Simulation-based method for finding minimum travel time budget paths in stochastic networks with correlated link times, *Transportation Research Record: Journal of the Transportation Research Board* **2467**(1): 140–148.

Supplementary results

Tables 8 to 23 provide detailed results obtained by $BPC-SP1_\alpha$, $BPC-SP1_\alpha^{adjL}$ and $BPC-SP2_\alpha$ when the covariance matrix \mathcal{C} has only non-negative entries. Tables 24 to 26 show the solution analysis in terms of expected travel time and variance. Tables 27 and 28 report results obtained by $BPC-MP2_\alpha$ when the covariance matrix allows negative entries. The meaning of each column is as follows:

Group : Group (C101, R101, RC101, A or P) of the instance,

n : Number of customers,

$|K|$: Number of vehicles,

α : Weight used in the objective function convex combination

z : Cost of the best feasible solution found,

Exp : Total expected travel time of the solution,

Var : Total variance of the solution,

Gap₁(%) : Integrality gap in percentage before adding the rounded capacity cuts,

Gap₂(%) : Integrality gap in percentage after adding the rounded capacity cuts,

RT(s) : CPU time in seconds at the root node of the search tree,

Nodes : Number of nodes explored in the search tree,

TT(s) : Total time in seconds,

OptGap(%) : Optimality gap in percentage,

$\uparrow Exp$: Increased expected travel time with respect to the solutions of the CVRP,

$\downarrow Var$: Decreased variance with respect to the solutions of the CVRP.

Furthermore, we use the following notation:

TL: Time limit reached,

- : Information is not available.

Table 8 Computational results of $BPC-SP1_\alpha$ on the C101 instances in the general case

Instance				Solution			Root node			Branch-and-bound		
Group	n	K	α	z	Exp	Var	Gap ₁ (%)	Gap ₂ (%)	RT(s)	Nodes	TT(s)	OptGap(%)
C101	15	2	0.05	1352.81	1345.0	1501.3	11.158	0.646	5.8	5	16.8	0.000
C101	15	2	0.1	1352.93	1372.0	1181.3	13.116	1.757	11.1	25	61.8	0.000
C101	15	2	0.3	1197.89	1454.0	600.3	12.199	3.017	10.8	25	94.4	0.000
C101	15	2	0.5	1010.65	1574.0	447.3	10.327	6.271	12.7	59	261.0	0.000
C101	20	2	0.05	1566.49	1598.0	967.8	1.299	1.299	51.8	11	200.4	0.000
C101	20	2	0.1	1527.98	1645.0	474.8	2.220	2.220	60.3	11	203.1	0.000
C101	20	2	0.3	1293.94	1645.0	474.8	1.972	1.972	65.4	13	257.2	0.000
C101	20	2	0.5	1052.90	1688.0	417.8	3.932	3.932	66.0	11	266.4	0.000
C101	23	3	0.05	1794.10	1844.0	846.0	1.646	0.470	165.4	25	610.9	0.000
C101	23	3	0.1	1744.00	1846.0	826.0	1.919	1.078	186.9	45	897.9	0.000
C101	23	3	0.3	1530.70	1891.0	690.0	4.802	3.728	339.6	249	5518.1	0.000
C101	23	3	0.5	-	-	-	-	-	349.6	241	TL	-
C101	25	3	0.05	1864.71	1922.0	776.2	2.537	0.936	218.2	63	1977.5	0.000
C101	25	3	0.1	1794.62	1935.0	531.2	2.518	1.071	257.3	39	1667.4	0.000
C101	25	3	0.3	1507.26	1965.0	439.2	5.088	3.391	415.4	95	5004.2	0.000
C101	25	3	0.5	-	-	-	-	-	449.0	188	TL	-
C101	28	3	0.05	1924.82	1981.0	857.5	1.806	0.648	488.6	63	3920.5	0.000
C101	28	3	0.1	1868.65	1981.0	857.5	1.815	1.149	720.6	29	3235.0	0.000
C101	28	3	0.3	1587.45	2028.0	559.5	3.865	3.151	1255.5	65	8208.7	0.000
C101	28	3	0.5	1231.75	2128.0	335.5	5.957	4.039	1266.0	101	12313.1	0.000
C101	30	3	0.05	2025.83	2054.0	1490.7	2.608	1.532	816.6	49	4434.4	0.000
C101	30	3	0.1	1997.67	2054.0	1490.7	4.111	2.487	1191.2	85	10783.8	0.000
C101	30	3	0.3	1712.01	2241.0	477.7	6.287	3.510	1552.7	79	17037.7	0.000
C101	30	3	0.5	-	-	-	-	-	1945.1	42	TL	-
C101	32	3	0.05	-	0.0	0.0	-	-	1951.8	100	TL	-
C101	32	3	0.1	2430.69	2545.0	1401.9	3.900	0.640	3056.3	69	17169.1	0.000
C101	32	3	0.3	-	-	-	-	-	4027.8	27	TL	-
C101	32	3	0.5	-	-	-	-	-	5159.5	20	TL	-

Table 9 Computational results of $BPC-SP1_\alpha$ on the RC101 instances in the general case

Instance				Solution			Root node			Branch-and-bound		
<i>Group</i>	<i>n</i>	$ K $	α	<i>z</i>	<i>Exp</i>	<i>Var</i>	<i>Gap</i> ₁ (%)	<i>Gap</i> ₂ (%)	<i>RT</i> (s)	<i>Nodes</i>	<i>TT</i> (s)	<i>OptGap</i> (%)
RC101	15	2	0.05	1895.41	1873.0	2321.3	0.304	0.304	9.0	9	20.9	0.000
RC101	15	2	0.1	1884.13	1910.0	1651.3	0.526	0.526	9.7	7	17.0	0.000
RC101	15	2	0.3	1793.79	1968.0	1387.3	2.271	1.594	18.0	25	84.5	0.000
RC101	15	2	0.5	1589.65	2199.0	980.3	4.028	2.832	19.6	21	113.6	0.000
RC101	20	3	0.05	2792.53	2857.0	1567.7	3.148	0.809	66.1	39	271.0	0.000
RC101	20	3	0.1	2726.87	2882.0	1330.7	4.081	1.596	92.1	67	492.9	0.000
RC101	20	3	0.3	2334.71	2966.0	861.7	5.305	3.149	112.0	65	568.3	0.000
RC101	20	3	0.5	1860.85	3151.0	570.7	7.142	4.537	130.8	67	1100.8	0.000
RC101	23	3	0.05	2845.85	2901.0	1798.0	1.709	0.370	187.1	37	828.4	0.000
RC101	23	3	0.1	2772.30	2934.0	1317.0	1.891	0.532	167.2	65	1140.1	0.000
RC101	23	3	0.3	2372.60	3053.0	785.0	3.587	2.357	199.4	83	1917.3	0.000
RC101	23	3	0.5	1919.00	3053.0	785.0	5.015	4.312	336.1	107	3655.9	0.000
RC101	25	3	0.05	2916.71	2981.0	1695.2	0.990	0.360	223.2	51	1561.0	0.000
RC101	25	3	0.1	2841.62	2997.0	1443.2	1.241	0.664	388.7	41	1552.1	0.000
RC101	25	3	0.3	2483.96	3074.0	1107.2	3.720	2.343	471.7	79	3395.7	0.000
RC101	25	3	0.5	2060.60	3291.0	830.2	6.855	3.775	666.5	247	TL	0.126
RC101	28	4	0.05	4051.67	4171.0	1784.4	8.681	0.482	642.3	363	11551.1	0.000
RC101	28	4	0.1	3932.34	4171.0	1784.4	9.144	0.998	916.8	472	TL	0.021
RC101	28	4	0.3	-	-	-	-	-	1899.7	178	TL	-
RC101	28	4	0.5	-	-	-	-	-	2124.6	139	TL	-
RC101	30	4	0.05	4042.48	4167.0	1676.6	6.189	0.271	1338.6	99	7180.1	0.000
RC101	30	4	0.1	3917.96	4167.0	1676.6	6.959	0.561	1708.3	161	12326.8	0.000
RC101	30	4	0.3	-	-	-	-	-	2372.8	165	TL	-
RC101	30	4	0.5	-	-	-	-	-	3328.8	81	TL	-
RC101	32	4	0.05	4053.94	4168.0	1886.8	4.243	0.261	2323.9	131	13603.0	0.000
RC101	32	4	0.1	3939.88	4168.0	1886.8	4.564	0.539	1523.0	133	TL	0.019
RC101	32	4	0.3	-	-	-	-	-	3741.7	29	TL	-
RC101	32	4	0.5	-	-	-	-	-	5823.8	12	TL	-

Table 10 Computational results of $BPC-SP1_\alpha$ on the R101 instances in the general case

Instance				Solution			Root node			Branch-and-bound		
Group	n	$ K $	α	z	Exp	Var	$Gap_1(\%)$	$Gap_2(\%)$	$RT(s)$	$Nodes$	$TT(s)$	$OptGap(\%)$
R101	15	2	0.05	3343.41	2614.0	17202.3	3.425	3.425	15.1	247	1061.4	0.000
R101	15	2	0.1	3796.43	3077.0	10271.3	2.058	2.058	17.6	77	363.6	0.000
R101	15	2	0.3	4459.69	4462.0	4454.3	1.207	1.207	19.0	19	105.8	0.000
R101	15	2	0.5	4458.15	4462.0	4454.3	0.941	0.941	19.9	9	57.7	0.000
R101	20	2	0.05	3437.59	2893.0	13784.8	1.643	1.643	70.6	41	556.8	0.000
R101	20	2	0.1	3825.58	3321.0	8366.8	3.188	2.696	76.7	113	2321.9	0.000
R101	20	2	0.3	4580.54	4019.0	5890.8	3.402	3.036	90.2	89	3007.0	0.000
R101	20	2	0.5	4954.90	4019.0	5890.8	3.861	3.861	95.6	51	1771.1	0.000
R101	23	2	0.05	3718.20	3144.0	14628.1	1.719	1.719	214.6	23	1148.6	0.000
R101	23	2	0.1	4249.61	3379.0	12085.1	3.867	3.428	236.7	139	9237.0	0.000
R101	23	2	0.3	5221.13	3905.0	8292.1	2.495	1.974	262.9	41	3510.9	0.000
R101	23	2	0.5	5611.05	4661.0	6561.1	2.209	2.209	346.3	15	1589.8	0.000
R101	25	2	0.05	4348.96	3474.0	20973.3	2.220	1.613	370.2	45	3720.7	0.000
R101	25	2	0.1	4977.83	4178.0	12176.3	3.005	1.400	453.3	85	9454.0	0.000
R101	25	2	0.3	5783.59	4852.0	7957.3	2.958	1.329	516.1	31	5181.6	0.000
R101	25	2	0.5	6149.15	5585.0	6713.3	3.115	2.335	580.8	23	4673.3	0.000
R101	28	2	0.05	4591.93	3995.0	15933.6	1.719	0.756	983.7	17	3486.9	0.000
R101	28	2	0.1	5158.76	4071.0	14948.6	0.390	0.390	1253.2	5	2046.1	0.000
R101	28	2	0.3	-	-	-	-	-	1382.7	43	TL	-
R101	28	2	0.5	7356.30	5699.0	9013.6	1.941	1.654	1656.1	21	9855.2	0.000
R101 [†]	30	3	0.05	-	-	-	-	-	-	-	-	-
R101 [†]	30	3	0.1	-	-	-	-	-	-	-	-	-
R101 [†]	30	3	0.3	-	-	-	-	-	-	-	-	-
R101 [†]	30	3	0.5	-	-	-	-	-	-	-	-	-
R101 [†]	32	3	0.05	-	-	-	-	-	-	-	-	-
R101 [†]	32	3	0.1	-	-	-	-	-	-	-	-	-
R101 [†]	32	3	0.3	-	-	-	-	-	-	-	-	-
R101 [†]	32	3	0.5	-	-	-	-	-	-	-	-	-

[†]: These instances could not be solved due to CPLEX numerical instability.

Table 11 Computational results of $BPC-SP1_{\alpha}^{adjL}$ on the C101 instances in the adjacent case

Instance				Solution			Root node			Branch-and-bound		
Group	n	$ K $	α	z	Exp	Var	$Gap_1(\%)$	$Gap_2(\%)$	$RT(s)$	$Nodes$	$TT(s)$	$OptGap(\%)$
C101	20	2	0.05	1531.59	1598.0	269.8	0.000	0.000	6.1	1	6.1	0.000
C101	20	2	0.1	1465.18	1598.0	269.8	0.000	0.000	6.6	1	6.6	0.000
C101	20	2	0.3	1199.54	1598.0	269.8	0.000	0.000	4.5	1	4.5	0.000
C101	20	2	0.5	933.90	1598.0	269.8	0.000	0.000	5.8	1	5.8	0.000
C101	25	3	0.05	1837.56	1894.0	765.2	1.680	0.038	6.8	3	8.5	0.000
C101	25	3	0.1	1766.32	1913.0	446.2	1.542	0.000	6.4	1	7.4	0.000
C101	25	3	0.3	1448.26	1951.0	275.2	0.726	0.000	6.2	1	6.9	0.000
C101	25	3	0.5	1112.60	1960.0	265.2	0.255	0.000	6.5	1	6.8	0.000
C101	30	3	0.05	1967.68	2032.0	745.7	1.072	0.000	14.6	1	15.8	0.000
C101	30	3	0.1	1902.07	2035.0	705.7	1.435	0.000	15.1	1	17.8	0.000
C101	30	3	0.3	1608.71	2078.0	513.7	2.451	0.000	14.7	1	16.7	0.000
C101	30	3	0.5	1265.85	2175.0	356.7	3.140	0.000	14.0	1	15.2	0.000
C101	35	4	0.05	2676.61	2777.0	769.1	3.657	0.817	27.6	83	70.5	0.000
C101	35	4	0.1	2572.11	2783.0	674.1	3.675	0.726	27.3	55	782.7	0.000
C101	35	4	0.3	2131.23	2820.0	524.1	3.933	0.045	30.3	5	35.8	0.000
C101	35	4	0.5	1659.05	2855.0	463.1	4.222	0.904	25.5	109	61.1	0.000
C101	40	4	0.05	3186.93	3305.0	943.6	4.863	0.095	53.8	5	2687.2	0.000
C101	40	4	0.1	3064.16	3325.0	716.6	4.761	0.000	50.9	1	2406.1	0.000
C101	40	4	0.3	2518.78	3367.0	539.6	3.590	0.361	48.5	15	2776.8	0.000
C101	40	4	0.5	1940.30	3399.0	481.6	2.347	0.358	46.0	9	56.3	0.000
C101	50	5	0.05	-	-	-	-	-	149.2	1	TL	-
C101	50	5	0.1	3345.55	3602.0	1037.5	1.992	0.000	143.1	1	160.3	0.000
C101	50	5	0.3	2751.25	3667.0	614.5	1.790	0.000	134.2	1	145.8	0.000
C101	50	5	0.5	-	-	-	-	-	165.3	2	TL	-
C101	60	6	0.05	-	-	-	-	-	7249.7	1	TL	-
C101	60	6	0.1	-	-	-	-	-	7232.7	1	TL	-
C101	60	6	0.3	3599.02	4786.0	829.4	0.754	0.000	386.1	1	405.6	0.000
C101	60	6	0.5	-	-	-	-	-	372.4	2	TL	-

Table 12 Computational results of $BPC-SP1_{\alpha}^{adjL}$ on the RC101 instances in the adjacent case

Instance				Solution			Root node			Branch-and-bound		
<i>Group</i>	<i>n</i>	$ K $	α	<i>z</i>	<i>Exp</i>	<i>Var</i>	<i>Gap</i> ₁ (%)	<i>Gap</i> ₂ (%)	<i>RT</i> (s)	<i>Nodes</i>	<i>TT</i> (s)	<i>OptGap</i> (%)
RC101	20	3	0.05	2739.93	2842.0	800.7	2.021	0.000	1.0	1	1.1	0.000
RC101	20	3	0.1	2637.87	2842.0	800.7	2.165	0.000	1.1	1	1.2	0.000
RC101	20	3	0.3	2195.61	2910.0	528.7	1.912	0.011	1.3	3	1.4	0.000
RC101	20	3	0.5	1666.85	3047.0	286.7	0.725	0.000	1.0	1	1.1	0.000
RC101	25	3	0.05	2862.51	2959.0	1029.2	0.649	0.000	5.1	1	5.2	0.000
RC101	25	3	0.1	2759.92	2968.0	887.2	0.636	0.000	5.6	1	5.8	0.000
RC101	25	3	0.3	2287.56	3012.0	597.2	0.634	0.000	5.0	1	5.1	0.000
RC101	25	3	0.5	1775.60	3083.0	468.2	0.585	0.000	5.2	1	5.3	0.000
RC101	30	4	0.05	3995.83	4167.0	743.6	5.852	0.000	8.5	1	8.8	0.000
RC101	30	4	0.1	3824.66	4167.0	743.6	5.684	0.000	8.5	1	8.8	0.000
RC101	30	4	0.3	3139.98	4167.0	743.6	6.065	0.000	9.3	1	9.6	0.000
RC101	30	4	0.5	2409.80	4327.0	492.6	6.823	0.000	7.8	1	8.3	0.000
RC101	35	4	0.05	4550.85	4710.0	1527.1	3.480	0.000	25.4	1	26.5	0.000
RC101	35	4	0.1	4390.31	4715.0	1468.1	3.669	0.000	25.7	1	26.4	0.000
RC101	35	4	0.3	3630.63	4848.0	790.1	3.415	0.000	26.3	1	28.1	0.000
RC101	35	4	0.5	2808.05	4909.0	707.1	4.087	0.000	26.9	1	28.4	0.000
RC101	40	5	0.05	4944.92	5156.0	934.5	2.350	0.000	37.0	1	38.1	0.000
RC101	40	5	0.1	4733.25	5157.0	919.5	2.385	0.000	39.6	1	40.4	0.000
RC101	40	5	0.3	3863.45	5192.0	763.5	2.686	0.000	36.1	1	37.3	0.000
RC101	40	5	0.5	2953.25	5323.0	583.5	3.642	0.017	35.6	3	37.9	0.000
RC101	50	5	0.05	5007.02	5195.0	1435.5	0.197	0.000	151.9	1	152.6	0.000
RC101	50	5	0.1	4819.05	5195.0	1435.5	0.108	0.000	167.7	1	168.7	0.000
RC101	50	5	0.3	4047.35	5234.0	1278.5	0.487	0.000	152.1	1	154.1	0.000
RC101	50	5	0.5	3116.75	5606.0	627.5	0.247	0.000	147.8	1	148.3	0.000

Table 13 Computational results of $BPC-SP1_{\alpha}^{adjL}$ on the R101 instances in the adjacent case

Instance				Solution			Root node			Branch-and-bound		
<i>Group</i>	<i>n</i>	$ K $	α	<i>z</i>	<i>Exp</i>	<i>Var</i>	<i>Gap</i> ₁ (%)	<i>Gap</i> ₂ (%)	<i>RT</i> (s)	<i>Nodes</i>	<i>TT</i> (s)	<i>OptGap</i> (%)
R101	20	2	0.05	2901.24	2810.0	4634.8	1.294	0.849	7.8	11	9.0	0.000
R101	20	2	0.1	2956.88	2893.0	3531.8	0.758	0.433	7.5	3	7.7	0.000
R101	20	2	0.3	2898.14	3038.0	2571.8	0.000	0.000	6.9	1	6.9	0.000
R101	20	2	0.5	2704.40	3576.0	1832.8	1.250	0.931	8.4	7	9.2	0.000
R101	25	2	0.05	3479.82	3378.0	5414.3	1.574	1.228	27.4	37	49.2	0.000
R101	25	2	0.1	3553.83	3474.0	4272.3	0.802	0.552	28.9	7	38.7	0.000
R101	25	2	0.3	3713.49	3474.0	4272.3	2.033	1.432	39.1	31	53.8	0.000
R101	25	2	0.5	3335.15	4763.0	1907.3	1.174	0.694	36.9	9	39.1	0.000
R101	30	3	0.05	3682.93	3616.0	4954.7	1.203	1.143	18.7	21	403.5	0.000
R101	30	3	0.1	3749.87	3616.0	4954.7	1.509	1.322	19.1	21	523.4	0.000
R101	30	3	0.3	3550.11	4158.0	2131.7	0.430	0.018	19.2	3	184.2	0.000
R101	30	3	0.5	3078.85	4463.0	1694.7	0.000	0.000	20.8	1	20.8	0.000
R101	35	3	0.05	4225.41	4099.0	6627.2	2.316	1.698	45.8	181	1066.4	0.000
R101	35	3	0.1	4291.62	4235.0	4801.2	2.497	1.673	46.9	107	2047.8	0.000
R101	35	3	0.3	4179.76	4699.0	2968.2	2.265	1.803	45.1	229	147.8	0.000
R101	35	3	0.5	3669.60	5139.0	2200.2	1.760	1.281	48.3	65	92.3	0.000
R101	40	3	0.05	4808.78	4686.0	7141.7	1.619	1.120	102.9	51	5662.5	0.000
R101	40	3	0.1	-	-	-	-	-	117.2	12	TL	-
R101	40	3	0.3	-	-	-	-	-	115.1	1	TL	-
R101	40	3	0.5	-	-	-	-	-	117.8	1	TL	-
R101	50	4	0.05	-	-	-	-	-	295.3	6	TL	-
R101	50	4	0.1	-	-	-	-	-	281.6	14	TL	-
R101	50	4	0.3	5060.98	5857.0	3203.6	2.251	0.908	275.0	14	TL	0.259
R101	50	4	0.5	4366.30	6478.0	2254.6	2.066	0.992	276.3	44	TL	0.122

Table 14 Computational results of $BPC-SP1_{\alpha}^{adjL}$ on the A instances with $n \leq 38$ in the adjacent case

Instance				Solution			Root node			Branch-and-bound		
Group	n	K	α	z	Exp	Var	Gap ₁ (%)	Gap ₂ (%)	RT(s)	Nodes	TT(s)	OptGap(%)
A	32	5	0.05	632.53	661.0	91.7	2.809	1.105	11.2	61	22.7	0.000
A	32	5	0.1	604.07	661.0	91.7	1.885	0.414	10.3	15	13.6	0.000
A	32	5	0.3	490.21	661.0	91.7	2.562	1.115	9.6	63	19.9	0.000
A	32	5	0.5	370.85	687.0	54.7	3.546	1.790	9.8	421	67.8	0.000
A	32	6	0.05	712.03	744.0	104.6	2.955	0.711	8.9	67	21.5	0.000
A	32	6	0.1	680.06	744.0	104.6	3.889	0.713	7.5	63	20.6	0.000
A	32	6	0.3	550.68	747.0	92.6	2.835	0.674	7.3	41	15.4	0.000
A	32	6	0.5	417.30	757.0	77.6	2.415	0.378	6.7	17	9.8	0.000
A	33	5	0.05	747.29	778.0	163.8	4.587	2.175	15.4	1213	218.7	0.000
A	33	5	0.1	716.58	-	-	5.235	3.806	13.2	29918	7200.4	0.624
A	33	5	0.3	576.44	794.0	68.8	4.736	2.171	12.2	2417	393.4	0.000
A	33	5	0.5	427.90	817.0	38.8	4.894	2.568	11.5	2501	414.5	0.000
A	35	5	0.05	765.55	799.0	130.0	2.984	1.525	19.7	649	210.2	0.000
A	35	5	0.1	732.10	799.0	130.0	3.007	1.216	18.7	2195	847.0	0.000
A	35	5	0.3	594.70	811.0	90.0	3.469	1.989	19.4	2603	844.9	0.000
A	35	5	0.5	447.00	834.0	60.0	4.030	1.841	16.9	1375	503.7	0.000
A	36	5	0.05	643.20	669.0	153.1	2.644	1.614	23.5	215	2558.5	0.000
A	36	5	0.1	616.51	672.0	117.1	2.598	1.301	21.8	441	226.3	0.000
A	36	5	0.3	502.43	683.0	81.1	2.707	1.122	21.7	103	1333.0	0.000
A	36	5	0.5	382.05	683.0	81.1	2.827	1.224	21.7	29	43.9	0.000
A	36	6	0.05	909.45	949.0	158.0	3.417	1.992	18.8	1855	466.2	0.000
A	36	6	0.1	868.40	952.0	116.0	3.666	2.011	14.8	18465	4826.3	0.000
A	36	6	0.3	696.80	956.0	92.0	3.380	1.896	14.2	2405	534.4	0.000
A	36	6	0.5	517.00	972.0	62.0	3.298	2.431	14.9	1113	279.7	0.000
A	37	5	0.05	699.06	730.0	111.2	5.895	2.662	26.3	863	257.1	0.000
A	37	5	0.1	667.22	731.0	93.2	6.314	2.987	29.1	8353	2736.4	0.000
A	37	5	0.3	535.86	738.0	64.2	6.362	3.124	27.3	2815	795.4	0.000
A	37	5	0.5	397.60	745.0	50.2	6.441	2.991	25.3	1337	394.8	0.000
A	38	5	0.05	791.66	826.0	139.3	1.988	0.468	33.3	13	45.2	0.000
A	38	5	0.1	757.33	826.0	139.3	2.712	1.741	34.1	3577	1661.5	0.000
A	38	5	0.3	617.79	840.0	99.3	2.425	0.972	33.9	95	2605.9	0.000
A	38	5	0.5	463.15	851.0	75.3	1.321	0.528	33.0	7	39.8	0.000
A	38	6	0.05	799.01	831.0	191.2	2.423	0.954	23.9	95	58.6	0.000
A	38	6	0.1	766.42	835.0	149.2	4.230	2.322	21.4	13197	5037.2	0.000
A	38	6	0.3	616.56	846.0	81.2	2.725	0.823	22.0	87	55.0	0.000
A	38	6	0.5	463.10	851.0	75.2	2.479	0.840	20.2	21	30.7	0.000

Table 15 Computational results of $BPC-SP1_{\alpha}^{adjL}$ on the A instances with $n \geq 43$ in the adjacent case

Instance				Solution			Root node			Branch-and-bound		
Group	n	K	α	z	Exp	Var	Gap ₁ (%)	Gap ₂ (%)	RT(s)	Nodes	TT(s)	OptGap(%)
A	43	6	0.05	894.83	937.0	93.7	1.070	0.675	54.5	19	3668.5	0.000
A	43	6	0.1	852.67	937.0	93.7	0.827	0.526	46.1	87	2739.2	0.000
A	43	6	0.3	684.01	937.0	93.7	0.802	0.208	46.6	3	50.9	0.000
A	43	6	0.5	514.35	952.0	76.7	1.881	1.108	42.4	92	89.9	0.000
A	44	6	0.05	905.89	945.0	162.8	3.394	0.098	67.4	3	82.2	0.000
A	44	6	0.1	863.58	949.0	94.8	3.561	0.716	66.0	37	3653.1	0.000
A	44	6	0.3	692.74	949.0	94.8	3.260	0.208	60.4	9	79.9	0.000
A	44	6	0.5	520.90	951.0	90.8	4.021	1.678	57.6	239	241.0	0.000
A	44	7	0.05	1095.84	1147.0	123.7	2.393	1.473	49.4	3749	1782.1	0.000
A	44	7	0.1	1044.67	-	-	2.504	1.713	50.8	14906	TL	0.587
A	44	7	0.3	832.51	-	-	2.569	1.688	52.1	15090	TL	0.164
A	44	7	0.5	611.35	1168.0	54.7	2.577	1.530	48.0	1733	813.3	0.000
A	45	7	0.05	875.69	914.0	147.8	1.087	0.000	63.0	1	66.2	0.000
A	45	7	0.1	834.38	917.0	90.8	1.749	0.530	57.3	79	128.0	0.000
A	45	7	0.3	666.54	927.0	58.8	1.402	0.144	57.2	9	67.3	0.000
A	45	7	0.5	492.40	928.0	56.8	1.506	0.000	53.5	1	57.6	0.000
A	47	7	0.05	1027.55	-	-	3.987	1.666	81.6	3021	TL	0.013
A	47	7	0.1	981.50	1074.0	149.0	3.323	1.051	75.4	1387	1105.5	0.000
A	47	7	0.3	794.30	-	-	-	-	75.1	9575	TL	-
A	47	7	0.5	590.50	-	-	4.596	2.400	70.0	9051	TL	0.067
A	52	7	0.05	966.38	-	-	-	-	135.6	20	TL	-
A	52	7	0.1	921.75	-	-	-	-	127.5	18	TL	-
A	52	7	0.3	741.25	-	-	-	-	130.0	90	TL	-
A	52	7	0.5	556.25	1035.0	77.5	3.025	2.015	127.8	5605	5948.6	0.000
A	53	7	0.05	-	-	-	-	-	175.2	4259	TL	-
A	53	7	0.1	-	-	-	-	-	153.8	6196	TL	-
A	53	7	0.3	849.38	-	-	-	-	130.7	94	TL	-
A	53	7	0.5	622.30	-	-	3.094	0.911	143.4	116	TL	0.152
A	54	9	0.05	1026.38	-	-	4.914	1.652	89.1	9246	TL	0.205
A	54	9	0.1	982.75	-	-	-	-	113.0	10972	TL	-
A	54	9	0.3	781.75	-	-	7.079	4.005	112.7	9178	TL	2.450
A	54	9	0.5	574.25	1095.0	53.5	4.836	1.547	109.3	9393	6850.7	0.000

Table 16 Computational results of $BPC-SP1_{\alpha}^{adjL}$ on the P instances in the adjacent case

Instance				Solution			Root node			Branch-and-bound		
Group	n	K	α	z	Exp	Var	Gap ₁ (%)	Gap ₂ (%)	RT(s)	Nodes	TT(s)	OptGap(%)
P	15	8	0.05	428.24	450.0	14.7	1.509	0.540	0.0	9	0.1	0.000
P	15	8	0.1	406.47	450.0	14.7	1.498	0.519	0.0	13	0.1	0.000
P	15	8	0.3	319.41	450.0	14.7	1.698	0.757	0.0	13	0.1	0.000
P	15	8	0.5	231.85	453.0	10.7	1.905	1.090	0.0	29	0.1	0.000
P	21	8	0.05	576.46	603.0	72.3	0.179	0.063	0.3	5	0.4	0.000
P	21	8	0.1	549.23	604.0	56.3	0.000	0.000	0.3	1	0.3	0.000
P	21	8	0.3	437.49	606.0	44.3	0.000	0.000	0.3	1	0.3	0.000
P	21	8	0.5	325.15	612.0	38.3	1.562	0.931	0.3	33	1.1	0.000
P	39	5	0.05	438.97	458.0	77.4	1.862	1.368	44.4	293	168.9	0.000
P	39	5	0.1	419.84	459.0	67.4	1.689	1.020	37.9	243	129.6	0.000
P	39	5	0.3	339.92	464.0	50.4	1.877	1.652	36.3	471	206.4	0.000
P	39	5	0.5	254.20	480.0	28.4	2.279	2.026	36.0	469	229.4	0.000
P	44	5	0.05	487.59	510.0	61.9	2.111	1.343	81.4	557	3676.2	0.000
P	44	5	0.1	465.19	510.0	61.9	1.852	1.458	72.7	3737	1887.2	0.000
P	44	5	0.3	374.67	513.0	51.9	2.357	1.620	67.7	1967	1115.0	0.000
P	44	5	0.5	281.45	523.0	39.9	3.307	2.597	68.3	3543	2210.3	0.000
P	49	7	0.05	529.21	554.0	58.2	1.799	1.421	67.4	1103	3273.2	0.000
P	49	7	0.1	504.42	554.0	58.2	1.826	1.249	68.4	2425	1628.5	0.000
P	49	7	0.3	402.76	556.0	45.2	1.807	1.460	70.2	1975	1498.2	0.000
P	49	7	0.5	298.60	568.0	29.2	1.875	1.428	60.3	1595	1156.7	0.000
P	49	8	0.05	602.40	-	-	-	-	70.6	9999	TL	-
P	49	8	0.1	573.81	-	-	-	-	54.7	9406	TL	-
P	49	8	0.3	459.13	-	-	-	-	54.5	9350	TL	-
P	49	8	0.5	340.05	-	-	-	-	50.5	8775	TL	-
P	49	10	0.05	663.94	697.0	35.9	1.138	0.893	38.4	551	223.4	0.000
P	49	10	0.1	630.89	-	-	1.470	1.161	39.1	15178	TL	0.040
P	49	10	0.3	497.87	698.0	30.9	1.302	1.019	37.8	1747	734.0	0.000
P	49	10	0.5	364.45	701.0	27.9	1.611	1.318	33.8	2131	1026.9	0.000
P	50	10	0.05	706.55	741.0	52.0	1.309	0.862	49.4	387	214.2	0.000
P	50	10	0.1	672.10	741.0	52.0	1.455	1.023	49.8	3371	1671.3	0.000
P	50	10	0.3	532.60	742.0	44.0	1.398	0.879	44.5	255	156.7	0.000
P	50	10	0.5	391.50	752.0	31.0	1.316	0.752	43.6	73	84.4	0.000
P	54	7	0.05	545.89	570.0	87.7	2.490	1.601	140.3	5631	6881.2	0.000
P	54	7	0.1	517.87	-	-	-	-	141.1	5780	TL	-
P	54	7	0.3	414.01	-	-	3.005	2.163	129.3	5885	TL	0.594
P	54	7	0.5	303.35	582.0	24.7	2.019	0.938	127.1	107	327.0	0.000
P	54	8	0.05	549.88	-	-	1.995	1.414	175.7	1986	TL	0.016
P	54	8	0.1	523.77	576.0	53.7	1.675	1.002	155.6	1903	TL	0.000
P	54	8	0.3	419.31	576.0	53.7	1.892	1.204	139.6	1213	1584.8	0.000
P	54	8	0.5	311.35	586.0	36.7	2.268	1.459	144.6	1155	1445.2	0.000
P	54	10	0.05	-	-	-	-	-	73.6	11887	TL	-
P	54	10	0.1	673.84	-	-	-	-	73.6	9418	TL	-
P	54	10	0.3	499.12	-	-	-	-	63.7	8982	TL	-
P	54	10	0.5	365.20	-	-	2.924	2.113	54.3	9893	TL	0.516

Table 17 Computational results of $BPC-SP2_\alpha$ on the C101 instances in the adjacent case

Instance				Solution			Root node			Branch-and-bound		
Group	n	$ K $	α	z	Exp	Var	$Gap_1(\%)$	$Gap_2(\%)$	$RT(s)$	$Nodes$	$TT(s)$	$OptGap(\%)$
C101	20	2	0.05	1531.59	1598.0	269.8	0.000	0.000	3.7	1	3.8	0.000
C101	20	2	0.1	1465.18	1598.0	269.8	0.000	0.000	9.7	1	9.8	0.000
C101	20	2	0.3	1199.54	1598.0	269.8	0.000	0.000	68.7	1	68.9	0.000
C101	20	2	0.5	933.90	1598.0	269.8	0.000	0.000	9.6	1	9.7	0.000
C101	25	3	0.05	1837.56	1894.0	765.2	1.644	0.000	1.4	1	2.2	0.000
C101	25	3	0.1	1766.32	1913.0	446.2	1.459	0.000	1.7	1	2.9	0.000
C101	25	3	0.3	1448.26	1951.0	275.2	0.647	0.000	3.2	1	4.3	0.000
C101	25	3	0.5	1112.60	1960.0	265.2	0.000	0.000	24.9	1	25.1	0.000
C101	30	3	0.05	1967.68	2032.0	745.7	1.030	0.000	5.3	1	7.6	0.000
C101	30	3	0.1	1902.07	2035.0	705.7	1.370	0.000	5.2	1	6.4	0.000
C101	30	3	0.3	1608.71	2078.0	513.7	2.157	0.000	5.7	1	8.7	0.000
C101	30	3	0.5	1265.85	2175.0	356.7	2.780	0.000	9.1	1	12.0	0.000
C101	35	4	0.05	2676.60	2777.0	769.1	3.634	0.729	3.8	77	60.1	0.000
C101	35	4	0.1	2572.11	2783.0	674.1	3.644	0.242	2.8	11	11.9	0.000
C101	35	4	0.3	2131.23	2820.0	524.1	3.876	0.047	4.2	7	19.1	0.000
C101	35	4	0.5	1659.05	2855.0	463.1	4.044	0.525	3.7	67	57.1	0.000
C101	40	4	0.05	3186.93	3305.0	943.6	4.858	0.105	6.2	5	19.0	0.000
C101	40	4	0.1	3064.16	3325.0	716.6	4.749	0.000	5.2	1	14.3	0.000
C101	40	4	0.3	2518.78	3367.0	539.6	3.577	0.185	6.2	9	159.2	0.000
C101	40	4	0.5	1940.30	3399.0	481.6	2.111	0.670	11.5	5	91.8	0.000
C101	50	5	0.05	3473.78	3602.0	1037.5	1.692	0.000	13.0	1	26.4	0.000
C101	50	5	0.1	3345.55	3602.0	1037.5	1.931	0.000	15.0	1	35.7	0.000
C101	50	5	0.3	2751.25	3667.0	614.5	1.658	0.000	30.1	1	59.2	0.000
C101	50	5	0.5	2132.25	3798.0	466.5	1.579	0.000	25.8	1	46.7	0.000
C101	60	6	0.05	4511.07	4679.0	1320.4	0.844	0.000	5.7	1	20.5	0.000
C101	60	6	0.1	4343.14	4679.0	1320.4	0.828	0.210	7.5	3	30.2	0.000
C101	60	6	0.3	3599.02	4786.0	829.4	0.563	0.000	29.4	1	107.6	0.000
C101	60	6	0.5	2804.20	4805.0	803.4	1.017	0.000	20.3	1	75.2	0.000

Table 18 Computational results of $BPC-SP2_\alpha$ on the RC101 instances in the adjacent case

Instance				Solution			Root node			Branch-and-bound		
<i>Group</i>	<i>n</i>	$ K $	α	<i>z</i>	<i>Exp</i>	<i>Var</i>	<i>Gap</i> ₁ (%)	<i>Gap</i> ₂ (%)	<i>RT</i> (s)	<i>Nodes</i>	<i>TT</i> (s)	<i>OptGap</i> (%)
RC101	20	3	0.05	2739.94	2842.0	800.7	2.021	0.000	0.4	1	0.5	0.000
RC101	20	3	0.1	2637.87	2842.0	800.7	2.165	0.000	0.4	1	0.6	0.000
RC101	20	3	0.3	2195.61	2910.0	528.7	1.900	0.000	0.9	1	1.2	0.000
RC101	20	3	0.5	1666.85	3047.0	286.7	0.710	0.000	0.9	1	1.0	0.000
RC101	25	3	0.05	2862.51	2959.0	1029.2	0.647	0.000	0.7	1	1.7	0.000
RC101	25	3	0.1	2759.92	2968.0	887.2	0.617	0.000	0.7	1	1.1	0.000
RC101	25	3	0.3	2287.56	3012.0	597.2	0.624	0.000	0.9	1	1.2	0.000
RC101	25	3	0.5	1775.60	3083.0	468.2	0.547	0.000	1.1	1	1.4	0.000
RC101	30	4	0.05	3995.83	4167.0	743.6	5.846	0.000	0.7	1	1.4	0.000
RC101	30	4	0.1	3824.66	4167.0	743.6	5.677	0.000	0.7	1	1.4	0.000
RC101	30	4	0.3	3139.98	4167.0	743.6	6.024	0.000	1.0	1	2.0	0.000
RC101	30	4	0.5	2409.80	4327.0	492.6	6.773	0.000	1.3	1	2.3	0.000
RC101	35	4	0.05	4550.86	4710.0	1527.1	3.466	0.000	1.9	1	3.1	0.000
RC101	35	4	0.1	4390.31	4715.0	1468.1	3.632	0.000	2.1	1	3.6	0.000
RC101	35	4	0.3	3630.63	4848.0	790.1	3.337	0.000	12.9	1	17.5	0.000
RC101	35	4	0.5	2808.05	4909.0	707.1	3.972	0.000	7.2	1	17.4	0.000
RC101	40	5	0.05	4944.93	5156.0	934.5	2.348	0.000	1.2	1	3.8	0.000
RC101	40	5	0.1	4733.25	5157.0	919.5	2.384	0.000	1.2	1	3.9	0.000
RC101	40	5	0.3	3863.45	5192.0	763.5	2.642	0.000	2.2	1	4.2	0.000
RC101	40	5	0.5	2953.25	5323.0	583.5	3.516	0.000	2.1	1	4.1	0.000
RC101	50	5	0.05	5007.02	5195.0	1435.5	0.165	0.000	3.7	1	8.1	0.000
RC101	50	5	0.1	4819.05	5195.0	1435.5	0.044	0.000	4.3	1	9.5	0.000
RC101	50	5	0.3	4047.35	5234.0	1278.5	0.440	0.000	55.7	1	61.7	0.000
RC101	50	5	0.5	3116.75	5606.0	627.5	0.229	0.000	9.7	1	14.2	0.000

Table 19 Computational results of $BPC-SP2_\alpha$ on the R101 instances in the adjacent case

Instance				Solution			Root node			Branch-and-bound		
<i>Group</i>	<i>n</i>	$ K $	α	<i>z</i>	<i>Exp</i>	<i>Var</i>	$Gap_1(\%)$	$Gap_2(\%)$	<i>RT(s)</i>	<i>Nodes</i>	<i>TT(s)</i>	<i>OptGap(%)</i>
R101	20	2	0.05	2901.24	2810.0	4634.8	1.294	0.849	17.3	11	19.5	0.000
R101	20	2	0.1	2956.88	2893.0	3531.8	0.758	0.433	35.1	3	35.5	0.000
R101	20	2	0.3	2898.14	3038.0	2571.8	0.000	0.000	139.7	1	139.8	0.000
R101	20	2	0.5	2704.40	3576.0	1832.8	1.065	0.743	28.2	5	29.1	0.000
R101	25	2	0.05	3479.82	3378.0	5414.3	1.556	1.124	156.4	45	368.0	0.000
R101	25	2	0.1	3553.83	3474.0	4272.3	0.653	0.236	83.9	5	107.9	0.000
R101	25	2	0.3	3713.49	3474.0	4272.3	1.680	0.969	1001.2	11	1010.0	0.000
R101	25	2	0.5	3335.15	4763.0	1907.3	0.787	0.060	845.3	3	847.7	0.000
R101	30	3	0.05	3682.93	3616.0	4954.7	1.158	1.100	4.9	19	13.9	0.000
R101	30	3	0.1	3749.87	3616.0	4954.7	1.359	1.113	6.4	17	18.9	0.000
R101	30	3	0.3	3550.11	4158.0	2131.7	0.368	0.000	17.8	1	19.6	0.000
R101	30	3	0.5	3078.85	4463.0	1694.7	0.000	0.000	22.5	1	23.0	0.000
R101	35	3	0.05	4225.41	4099.0	6627.2	2.271	1.610	29.6	189	139.7	0.000
R101	35	3	0.1	4291.62	4235.0	4801.2	2.351	1.371	18.1	35	47.8	0.000
R101	35	3	0.3	4179.76	4699.0	2968.2	2.117	1.552	33.8	193	312.9	0.000
R101	35	3	0.5	3669.60	5139.0	2200.2	1.366	0.763	206.8	25	609.5	0.000
R101	40	3	0.05	4808.78	4686.0	7141.7	1.588	1.082	71.3	43	130.9	0.000
R101	40	3	0.1	4906.67	4792.0	5938.7	1.716	1.123	65.7	39	157.9	0.000
R101	40	3	0.3	4850.71	5335.0	3720.7	1.801	1.130	95.5	41	226.9	0.000
R101	40	3	0.5	4387.85	6656.0	2119.7	2.506	1.645	85.1	37	345.9	0.000
R101	50	4	0.05	5444.28	5481.0	4746.6	2.998	1.693	27.6	579	1561.1	0.000
R101	50	4	0.1	5407.56	5481.0	4746.6	2.213	1.111	42.1	27	258.5	0.000
R101	50	4	0.3	5058.68	5768.0	3403.6	1.991	0.671	47.6	23	195.4	0.000
R101	50	4	0.5	4366.30	6478.0	2254.6	1.801	0.735	323.1	37	413.8	0.000
R101	60	5	0.05	5864.22	-	-	3.047	1.698	32.8	1379	TL	0.410
R101	60	5	0.1	5883.15	-	-	3.136	1.688	41.6	363	TL	0.646
R101	60	5	0.3	5499.55	6349.0	3517.5	2.048	1.044	65.7	27	544.4	0.000
R101	60	5	0.5	4829.75	6979.0	2680.5	2.159	0.994	56.2	104	TL	0.010
R101	65	5	0.05	-	-	-	-	-	87.3	1480	TL	-
R101	65	5	0.1	-	-	-	-	-	58.5	456	TL	-
R101	65	5	0.3	6105.50	7076.0	3841.0	1.733	1.024	221.3	397	TL	0.022
R101	65	5	0.5	5356.00	7516.0	3196.0	1.464	0.690	155.2	65	1337.5	0.000

Table 20 Computational results of $BPC-SP2_\alpha$ on the A instances with $n \leq 38$ in the adjacent case

Instance				Solution			Root node			Branch-and-bound		
Group	n	K	α	z	Exp	Var	Gap ₁ (%)	Gap ₂ (%)	RT(s)	Nodes	TT(s)	OptGap(%)
A	32	5	0.05	632.53	661.0	91.7	1.824	0.583	0.3	79	6.9	0.000
A	32	5	0.1	604.07	661.0	91.7	1.838	0.394	0.3	19	2.8	0.000
A	32	5	0.3	490.21	661.0	91.7	2.446	0.992	0.4	65	7.5	0.000
A	32	5	0.5	370.85	687.0	54.7	3.310	1.638	0.4	283	19.3	0.000
A	32	6	0.05	712.03	744.0	104.6	4.221	0.863	0.2	401	24.9	0.000
A	32	6	0.1	680.06	744.0	104.6	3.841	0.689	0.3	65	5.7	0.000
A	32	6	0.3	550.68	747.0	92.6	2.693	0.658	0.3	51	5.3	0.000
A	32	6	0.5	417.30	757.0	77.6	2.000	0.240	0.3	11	2.0	0.000
A	33	5	0.05	747.29	778.0	163.8	4.995	3.063	0.7	47435	4088.3	0.000
A	33	5	0.1	716.58	778.0	163.8	4.980	3.538	0.7	75106	7200.8	0.000
A	33	5	0.3	576.44	794.0	68.8	4.480	1.982	0.7	2855	224.7	0.000
A	33	5	0.5	427.90	817.0	38.8	4.373	2.155	0.9	1841	133.4	0.000
A	35	5	0.05	765.55	799.0	130.0	2.896	1.215	1.2	3171	514.6	0.000
A	35	5	0.1	732.10	799.0	130.0	2.952	1.191	1.6	2393	462.9	0.000
A	35	5	0.3	594.70	811.0	90.0	3.222	1.809	1.7	1265	278.3	0.000
A	35	5	0.5	447.00	834.0	60.0	3.764	1.624	1.4	1033	241.9	0.000
A	36	5	0.05	643.20	669.0	153.1	2.465	1.185	1.4	415	68.7	0.000
A	36	5	0.1	616.51	672.0	117.1	2.578	1.302	1.6	385	65.7	0.000
A	36	5	0.3	502.43	683.0	81.1	2.573	0.944	1.7	75	19.2	0.000
A	36	5	0.5	382.05	683.0	81.1	2.550	1.390	1.9	87	25.6	0.000
A	36	6	0.05	909.45	949.0	158.0	3.741	1.732	0.7	15579	1816.6	0.000
A	36	6	0.1	868.40	952.0	116.0	3.623	1.744	0.8	10051	1180.3	0.000
A	36	6	0.3	696.80	956.0	92.0	3.308	1.971	0.6	1975	330.9	0.000
A	36	6	0.5	517.00	972.0	62.0	3.038	2.067	0.5	631	132.0	0.000
A	37	5	0.05	699.06	730.0	111.2	6.242	2.928	1.1	13569	1886.2	0.000
A	37	5	0.1	667.22	731.0	93.2	6.293	2.975	1.1	7747	1095.1	0.000
A	37	5	0.3	535.86	738.0	64.2	6.252	3.034	1.4	2465	409.9	0.000
A	37	5	0.5	397.60	745.0	50.2	6.245	2.568	1.4	1035	165.9	0.000
A	38	5	0.05	791.66	826.0	139.3	2.969	2.091	1.4	18109	5485.7	0.000
A	38	5	0.1	757.33	826.0	139.3	2.552	1.511	1.5	3215	1178.5	0.000
A	38	5	0.3	617.79	840.0	99.3	2.129	0.694	1.6	31	28.7	0.000
A	38	5	0.5	463.15	851.0	75.3	1.041	0.359	1.7	5	6.3	0.000
A	38	6	0.05	799.01	831.0	191.2	4.152	2.327	0.7	18673	2723.0	0.000
A	38	6	0.1	766.42	835.0	149.2	4.171	2.267	0.8	9957	1539.7	0.000
A	38	6	0.3	616.56	846.0	81.2	2.638	0.453	0.8	31	9.3	0.000
A	38	6	0.5	463.10	851.0	75.2	2.151	0.236	1.0	3	3.6	0.000

Table 21 Computational results of $BPC-SP2_\alpha$ on the A instances with $n \geq 43$ in the adjacent case

Instance				Solution			Root node			Branch-and-bound		
Group	n	$ K $	α	z	Exp	Var	$Gap_1(\%)$	$Gap_2(\%)$	$RT(s)$	$Nodes$	$TT(s)$	$OptGap(\%)$
A	43	6	0.05	894.83	937.0	93.7	0.953	0.740	1.1	229	43.0	0.000
A	43	6	0.1	852.67	937.0	93.7	0.790	0.479	1.1	73	19.6	0.000
A	43	6	0.3	684.01	937.0	93.7	0.557	0.161	1.2	3	5.3	0.000
A	43	6	0.5	514.35	952.0	76.7	1.618	0.862	1.3	41	23.9	0.000
A	44	6	0.05	905.89	945.0	162.8	3.902	0.635	2.2	93	59.2	0.000
A	44	6	0.1	863.58	949.0	94.8	3.509	0.652	2.2	29	37.2	0.000
A	44	6	0.3	692.74	949.0	94.8	3.110	0.400	2.5	11	26.2	0.000
A	44	6	0.5	520.90	951.0	90.8	3.724	0.619	2.5	35	56.4	0.000
A	44	7	0.05	1095.84	-	-	2.369	1.466	0.9	34949	7202.6	0.287
A	44	7	0.1	1044.67	-	-	2.340	1.589	0.9	33073	7202.4	0.284
A	44	7	0.3	832.51	1162.0	63.7	2.450	1.565	1.3	13865	3360.0	0.000
A	44	7	0.5	611.35	1168.0	54.7	2.310	1.281	1.2	697	161.0	0.000
A	45	7	0.05	875.69	914.0	147.8	1.905	0.528	1.9	163	47.5	0.000
A	45	7	0.1	834.38	917.0	90.8	1.684	0.516	1.3	61	17.9	0.000
A	45	7	0.3	666.54	927.0	58.8	1.189	0.130	2.4	5	7.7	0.000
A	45	7	0.5	492.40	928.0	56.8	1.395	0.000	2.3	1	6.2	0.000
A	47	7	0.05	1027.55	1073.0	164.0	3.136	0.883	1.5	1753	589.0	0.000
A	47	7	0.1	981.50	1074.0	149.0	3.259	1.227	1.5	2873	1139.7	0.000
A	47	7	0.3	794.30	1082.0	123.0	4.098	1.942	2.3	9782	7203.3	0.000
A	47	7	0.5	590.50	1113.0	68.0	4.174	1.853	2.9	3005	4334.4	0.000
A	52	7	0.05	966.38	-	-	2.002	1.257	4.7	9534	TL	0.151
A	52	7	0.1	921.75	1011.0	118.5	1.973	1.152	5.2	8929	4875.6	0.000
A	52	7	0.3	741.25	1015.0	102.5	2.210	1.263	5.9	1491	1184.3	0.000
A	52	7	0.5	556.25	1035.0	77.5	2.815	1.730	10.6	2017	3171.0	0.000
A	53	7	0.05	-	-	-	-	-	3.5	11823	TL	-
A	53	7	0.1	-	-	-	-	-	3.2	9759	TL	-
A	53	7	0.3	849.38	-	-	4.076	2.300	4.3	5763	TL	0.278
A	53	7	0.5	622.30	1190.0	54.6	2.903	0.835	3.8	107	184.7	0.000
A	54	9	0.05	1026.38	-	-	4.589	1.346	0.8	35224	TL	0.310
A	54	9	0.1	982.75	-	-	5.097	1.966	0.9	32459	TL	0.762
A	54	9	0.3	781.75	-	-	4.729	1.635	0.9	24407	TL	0.048
A	54	9	0.5	574.25	1095.0	53.5	4.680	1.300	0.9	2071	461.1	0.000
A	60	9	0.05	997.70	-	-	4.391	2.968	3.0	15841	TL	1.763
A	60	9	0.1	943.61	-	-	3.597	2.108	1.9	14606	TL	0.856
A	60	9	0.3	749.03	1043.0	63.1	2.978	1.289	2.5	5483	4135.3	0.000
A	60	9	0.5	553.05	1043.0	63.1	2.499	1.069	2.0	606	623.9	0.000

Table 22 Computational results of $BPC-SP2_\alpha$ on the P instances with $n \leq 49$ in the adjacent case

Instance				Solution			Root node			Branch-and-bound		
Group	n	$ K $	α	z	Exp	Var	$Gap_1(\%)$	$Gap_2(\%)$	$RT(s)$	$Nodes$	$TT(s)$	$OptGap(\%)$
P	15	8	0.05	428.24	450.0	14.7	1.461	0.481	0.0	13	0.1	0.000
P	15	8	0.1	406.47	450.0	14.7	1.498	0.519	0.0	13	0.1	0.000
P	15	8	0.3	319.41	450.0	14.7	1.698	0.757	0.0	11	0.1	0.000
P	15	8	0.5	231.85	453.0	10.7	1.830	1.090	0.0	25	0.1	0.000
P	21	8	0.05	576.46	603.0	72.3	0.000	0.000	0.0	1	0.2	0.000
P	21	8	0.1	549.23	604.0	56.3	0.000	0.000	0.0	1	0.2	0.000
P	21	8	0.3	437.49	606.0	44.3	0.000	0.000	0.0	1	0.2	0.000
P	21	8	0.5	325.15	612.0	38.3	1.167	0.103	0.0	3	0.2	0.000
P	22	8	0.05	503.37	529.0	16.4	0.000	0.000	0.0	1	0.2	0.000
P	22	8	0.1	477.74	529.0	16.4	0.000	0.000	0.0	1	0.2	0.000
P	22	8	0.3	375.22	529.0	16.4	0.000	0.000	0.0	1	0.2	0.000
P	22	8	0.5	272.70	529.0	16.4	0.000	0.000	0.0	1	0.2	0.000
P	39	5	0.05	438.97	458.0	77.4	1.600	1.204	1.3	625	52.2	0.000
P	39	5	0.1	419.84	459.0	67.4	1.667	0.957	1.4	227	24.7	0.000
P	39	5	0.3	339.92	464.0	50.4	1.724	1.387	1.4	403	53.5	0.000
P	39	5	0.5	254.20	480.0	28.4	1.911	1.738	1.7	231	34.2	0.000
P	44	5	0.05	487.59	510.0	61.9	1.789	1.402	2.8	4617	567.0	0.000
P	44	5	0.1	465.19	510.0	61.9	1.816	1.425	2.9	3441	391.4	0.000
P	44	5	0.3	374.67	513.0	51.9	2.253	1.601	3.0	1365	239.7	0.000
P	44	5	0.5	281.45	523.0	39.9	3.138	2.237	3.3	3283	479.3	0.000
P	49	7	0.05	529.21	554.0	58.2	1.761	1.161	0.8	2913	334.0	0.000
P	49	7	0.1	504.42	554.0	58.2	1.807	1.214	0.8	2137	254.6	0.000
P	49	7	0.3	402.76	556.0	45.2	1.798	1.602	0.9	1575	238.8	0.000
P	49	7	0.5	298.60	568.0	29.2	1.792	1.359	0.9	1189	156.7	0.000
P	49	8	0.05	602.40	-	-	3.028	2.423	0.5	48380	7203.9	0.665
P	49	8	0.1	573.81	-	-	3.120	2.454	0.5	46327	7203.7	0.548
P	49	8	0.3	459.13	-	-	3.768	3.033	0.5	35257	7203.8	0.649
P	49	8	0.5	340.05	-	-	4.373	3.544	0.6	29269	7203.7	0.892
P	49	10	0.05	663.94	697.0	35.9	1.506	1.210	0.3	58189	2840.9	0.000
P	49	10	0.1	630.89	697.0	35.9	1.427	1.121	0.2	15527	744.6	0.000
P	49	10	0.3	497.87	698.0	30.9	1.197	0.819	0.2	719	44.8	0.000
P	49	10	0.5	364.45	701.0	27.9	1.511	1.161	0.2	1089	64.8	0.000

Table 23 Computational results of $BPC-SP2_\alpha$ on the P instances with $n \geq 50$ in the adjacent case

Instance				Solution			Root node			Branch-and-bound		
Group	n	$ K $	α	z	Exp	Var	$Gap_1(\%)$	$Gap_2(\%)$	$RT(s)$	$Nodes$	$TT(s)$	$OptGap(\%)$
P	50	10	0.05	706.55	741.0	52.0	1.428	0.969	0.3	8169	618.5	0.000
P	50	10	0.1	672.10	741.0	52.0	1.409	0.964	0.2	3105	209.0	0.000
P	50	10	0.3	532.60	742.0	44.0	1.188	0.758	0.2	143	19.8	0.000
P	50	10	0.5	391.50	752.0	31.0	1.044	0.501	0.2	41	9.4	0.000
P	54	7	0.05	545.89	-	-	3.057	2.411	1.6	29454	TL	1.083
P	54	7	0.1	517.87	-	-	2.543	1.697	1.6	26632	TL	0.256
P	54	7	0.3	414.01	580.0	26.7	2.521	1.665	1.6	12931	4498.5	0.000
P	54	7	0.5	303.35	582.0	24.7	1.748	0.661	1.8	49	30.1	0.000
P	54	8	0.05	549.88	576.0	53.7	1.723	1.343	1.3	12495	2523.1	0.000
P	54	8	0.1	523.77	576.0	53.7	1.633	0.695	1.5	589	120.1	0.000
P	54	8	0.3	419.31	576.0	53.7	1.698	0.976	1.4	625	161.7	0.000
P	54	8	0.5	311.35	586.0	36.7	2.027	1.104	1.4	519	143.1	0.000
P	54	10	0.05	-	-	-	-	-	0.4	82836	TL	-
P	54	10	0.1	673.84	-	-	9.571	9.185	0.4	84071	TL	7.605
P	54	10	0.3	499.12	700.0	30.4	2.401	1.799	0.4	49543	5038.1	0.000
P	54	10	0.5	365.20	702.0	28.4	2.377	1.452	0.4	3927	375.0	0.000
P	59	10	0.05	710.39	744.0	71.9	1.357	0.748	0.5	10575	1571.5	0.000
P	59	10	0.1	676.79	744.0	71.9	1.388	0.825	0.5	7255	1122.9	0.000
P	59	10	0.3	539.77	751.0	46.9	1.693	1.262	0.6	17117	2850.5	0.000
P	59	10	0.5	396.95	759.0	34.9	1.981	1.383	0.7	10665	1658.2	0.000
P	69	10	0.05	-	-	-	-	-	1.7	25424	TL	-
P	69	10	0.1	-	-	-	-	-	1.7	25424	TL	-
P	69	10	0.3	-	-	-	-	-	1.5	19634	TL	-
P	69	10	0.5	440.45	-	-	2.217	1.774	2.3	16137	TL	0.230
P	75	4	0.05	-	-	-	-	-	99.7	1174	TL	-
P	75	4	0.1	-	-	-	-	-	99.7	1174	TL	-
P	75	4	0.3	438.33	597.0	68.1	1.308	0.774	180.5	329	3485.9	0.000
P	75	4	0.5	330.55	601.0	60.1	1.825	0.851	166.2	411	3937.3	0.000

Table 24 Solution analysis on the CRCR instances in the general case

Instance			$\alpha = 0$		$\alpha = 0.05$		$\alpha = 0.1$		$\alpha = 0.3$		$\alpha = 0.5$	
<i>Group</i>	<i>n</i>	$ K $	Exp	Var	$\uparrow Exp(\%)$	$\downarrow Var(\%)$	$\uparrow Exp(\%)$	$\downarrow Var(\%)$	$\uparrow Exp(\%)$	$\downarrow Var(\%)$	$\uparrow Exp(\%)$	$\downarrow Var(\%)$
C101	15	2	1338.0	3113.3	0.5	51.8	2.5	62.1	8.0	80.7	15.0	85.6
C101	20	2	1596.0	5004.8	0.1	80.7	3.0	90.5	3.0	90.5	5.5	91.7
C101	23	3	1815.0	4931.0	1.6	82.8	1.7	83.2	4.0	86.0	-	-
C101	25	3	1882.0	5294.2	2.1	85.3	2.7	90.0	4.2	91.7	-	-
C101	28	3	1939.0	2240.5	2.1	61.7	2.1	61.7	4.4	75.0	8.9	85.0
C101	30	3	2026.0	3410.7	1.4	56.3	1.4	56.3	9.6	86.0	-	-
Avg.					1.3	69.8	2.2	74.0	5.5	85.0	9.8	87.4
RC101	15	2	1873.0	2573.3	0.0	9.8	1.9	35.8	4.8	46.1	14.8	61.9
RC101	20	3	2841.0	2405.7	0.6	34.8	1.4	44.7	4.2	64.2	9.8	76.3
RC101	23	3	2901.0	2969.0	0.0	39.4	1.1	55.6	5.0	73.6	5.0	73.6
RC101	25	3	2954.0	4014.2	0.9	57.8	1.4	64.0	3.9	72.4	10.2	79.3
Avg.					0.4	35.5	1.5	50.0	4.5	64.1	9.9	72.7
R101	15	2	2495.0	49515.3	4.6	65.3	18.9	79.3	44.1	91.0	44.1	91.0
R101	20	2	2810.0	29681.8	2.9	53.6	15.4	71.8	30.1	80.2	30.1	80.2
R101	23	2	3055.0	32861.1	2.8	55.5	9.6	63.2	21.8	74.8	34.5	80.0
R101	25	2	3362.0	38311.3	3.2	45.3	19.5	68.2	30.7	79.2	39.8	82.5
Avg.					3.4	54.9	15.9	70.6	31.7	81.3	37.1	83.4

Table 25 Solution analysis on the CRCR instances in the adjacent case

Instance			$\alpha = 0$		$\alpha = 0.05$		$\alpha = 0.1$		$\alpha = 0.3$		$\alpha = 0.5$	
Group	n	$ K $	Exp	Var	$\uparrow Exp(\%)$	$\downarrow Var(\%)$	$\uparrow Exp(\%)$	$\downarrow Var(\%)$	$\uparrow Exp(\%)$	$\downarrow Var(\%)$	$\uparrow Exp(\%)$	$\downarrow Var(\%)$
C101	20	2	1596.0	309.8	0.1	12.9	0.1	12.9	0.1	12.9	0.1	12.9
C101	25	3	1882.0	2248.2	0.6	66.0	1.6	80.2	3.5	87.8	4.0	88.2
C101	30	3	2026.0	1481.7	0.3	49.7	0.4	52.4	2.5	65.3	6.9	75.9
C101	35	4	2757.0	2439.1	0.7	68.5	0.9	72.4	2.2	78.5	3.4	81.0
C101	40	4	3299.0	1699.6	0.2	44.5	0.8	57.8	2.0	68.3	2.9	71.7
C101	50	5	3602.0	1342.5	0.0	22.7	0.0	22.7	1.8	54.2	5.2	65.3
C101	60	6	4671.0	2711.4	0.2	51.3	0.2	51.3	2.4	69.4	2.8	70.4
Avg.					0.3	45.1	0.6	49.9	2.1	62.3	3.6	66.5
RC101	20	3	2841.0	1096.7	0.0	27.0	0.0	27.0	2.4	51.8	6.8	73.9
RC101	25	3	2954.0	1184.2	0.2	13.1	0.5	25.1	1.9	49.6	4.2	60.5
RC101	30	4	4167.0	743.6	0.0	0.0	0.0	0.0	0.0	0.0	3.7	33.8
RC101	35	4	4710.0	3668.1	0.0	58.4	0.1	60.0	2.8	78.5	4.1	80.7
RC101	40	5	5156.0	1446.5	0.0	35.4	0.0	36.4	0.7	47.2	3.1	59.7
RC101	50	5	5195.0	2414.5	0.0	40.5	0.0	40.5	0.7	47.0	7.3	74.0
Avg.					0.0	29.1	0.1	31.5	1.4	45.7	4.9	63.7
R101	20	2	2810.0	7962.8	0.0	41.8	2.9	55.6	7.5	67.7	21.4	77.0
R101	25	2	3362.0	8446.3	0.5	35.9	3.2	49.4	3.2	49.4	29.4	77.4
R101	30	3	3594.0	10194.7	0.6	51.4	0.6	51.4	13.6	79.1	19.5	83.4
R101	35	3	4088.0	7195.2	0.3	7.9	3.5	33.3	13.0	58.7	20.5	69.4
R101	40	3	4615.0	11637.7	1.5	38.6	3.7	49.0	13.5	68.0	30.7	81.8
R101	50	4	5287.0	10945.6	3.5	56.6	3.5	56.6	8.3	68.9	18.4	79.4
Avg.					1.1	38.7	2.9	49.2	9.8	65.3	23.3	78.1

Table 26 Solution analysis on the AP instances in the adjacent case

Instance			$\alpha = 0$		$\alpha = 0.05$		$\alpha = 0.1$		$\alpha = 0.3$		$\alpha = 0.5$	
Group	n	K	Exp	Var	$\uparrow Exp(\%)$	$\downarrow Var(\%)$	$\uparrow Exp(\%)$	$\downarrow Var(\%)$	$\uparrow Exp(\%)$	$\downarrow Var(\%)$	$\uparrow Exp(\%)$	$\downarrow Var(\%)$
A	32	5	661.0	119.7	0.0	23.4	0.0	23.4	0.0	23.4	3.8	54.3
A	32	6	742.0	251.6	0.3	58.4	0.3	58.4	0.7	63.2	2.0	69.2
A	33	5	778.0	210.8	0.0	22.3	0.0	22.3	2.0	67.4	4.8	81.6
A	35	5	799.0	192.0	0.0	32.3	0.0	32.3	1.5	53.1	4.2	68.7
A	36	5	669.0	211.1	0.0	27.5	0.4	44.5	2.0	61.6	2.0	61.6
A	36	6	949.0	211.0	0.0	25.1	0.3	45.0	0.7	56.4	2.4	70.6
A	37	5	730.0	194.2	0.0	42.7	0.1	52.0	1.1	66.9	2.0	74.2
A	38	5	822.0	266.3	0.5	47.7	0.5	47.7	2.1	62.7	3.4	71.7
A	38	6	831.0	248.2	0.0	23.0	0.5	39.9	1.8	67.3	2.4	69.7
A	43	6	937.0	160.7	0.0	41.7	0.0	41.7	0.0	41.7	1.6	52.3
A	44	6	944.0	223.8	0.1	27.3	0.5	57.6	0.5	57.6	0.7	59.4
A	45	7	914.0	180.8	0.0	18.3	0.3	49.8	1.4	67.5	1.5	68.6
A	47	7	1073.0	315.0	0.0	47.9	0.1	52.7	0.8	61.0	3.6	78.4
A	52	7	1010.0	185.5	0.1	36.1	0.1	36.1	0.5	44.7	2.4	58.2
Avg.					0.1	33.8	0.2	43.1	1.1	56.7	2.6	67.0
P	15	8	450.0	14.7	0.0	0.0	0.0	0.0	0.0	0.0	0.7	27.2
P	21	8	603.0	107.3	0.0	32.6	0.2	47.5	0.5	58.7	1.5	64.3
P	22	8	529.0	28.4	0.0	42.3	0.0	42.3	0.0	42.3	0.0	42.3
P	39	5	458.0	77.4	0.0	0.0	0.2	12.9	1.3	34.9	4.6	63.3
P	44	5	510.0	77.9	0.0	20.5	0.0	20.5	0.6	33.4	2.5	48.8
P	49	7	554.0	86.2	0.0	32.5	0.0	32.5	0.4	47.6	2.5	66.1
P	49	8	631.0	109.1	0.0	45.8	0.0	45.8	0.0	46.7	1.7	65.1
P	49	10	696.0	88.9	0.1	59.6	0.1	59.6	0.3	65.2	0.7	68.6
P	50	10	741.0	68.0	0.0	23.5	0.0	23.5	0.1	35.3	1.5	54.4
P	54	8	576.0	105.7	0.0	49.2	0.0	49.2	0.0	49.2	1.7	65.3
P	59	10	744.0	101.9	0.0	29.4	0.0	29.4	0.9	54.0	2.0	65.8
P Avg.					0.0	30.4	0.0	33.0	0.4	42.4	1.8	57.3

Table 27 Computational results of $BPC-MP2_\alpha$ on the C101 instances in the adjacent case with negative correlations

<i>name</i>	Instance						root node			B-and-B		
	<i>n</i>	<i>k</i>	α	<i>BF</i>	<i>Exp</i>	<i>Var</i>	<i>Gap</i> ₁ (%)	<i>Gap</i> ₂ (%)	<i>rt</i> (s)	<i>nodes</i>	<i>tt</i> (s)	<i>OptGap</i> (%)
C101	20	2	0.1	1465.18	1598.0	269.8	0.000	0.000	6.3	1	6.4	0.000
C101	20	2	0.3	1199.54	1598.0	269.8	0.000	0.000	16.0	1	16.1	0.000
C101	25	3	0.1	1760.12	1912.0	393.2	1.307	0.000	2.3	1	5.4	0.000
C101	25	3	0.3	1439.86	1951.0	247.2	0.754	0.000	20.5	1	53.3	0.000
C101	30	3	0.1	1901.27	2035.0	697.7	1.374	0.000	12.9	1	15.4	0.000
C101	30	3	0.3	1603.91	2078.0	497.7	2.372	0.000	157.3	1	1299.8	0.000
C101	35	4	0.1	2567.61	2781.0	647.1	3.722	0.727	4.6	65	674.4	0.000
C101	35	4	0.3	-	-	-	-	-	1466.1	1	TL	-
C101	40	4	0.1	3062.96	3325.0	704.6	4.833	0.000	13.2	1	478.1	0.000
C101	40	4	0.3	-	-	-	-	-	TL	1	TL	-
C101	50	5	0.1	3336.95	3632.0	681.5	1.941	0.000	166.7	1	1360.9	0.000
C101	50	5	0.3	-	-	-	-	-	TL	1	TL	-
C101	60	6	0.1	4337.34	4679.0	1262.4	0.855	0.000	1899.7	1	4979.2	0.000
C101	60	6	0.3	-	-	-	-	-	TL	1	TL	-

Table 28 Computational results of $BPC-MP2_\alpha$ on the P instances in the adjacent case with negative correlations

Instance							root node			B-and-B		
<i>name</i>	<i>n</i>	<i>k</i>	α	<i>BF</i>	<i>Exp</i>	<i>Var</i>	<i>Gap</i> ₁ (%)	<i>Gap</i> ₂ (%)	<i>rt</i> (s)	<i>nodes</i>	<i>tt</i> (s)	<i>OptGap</i> (%)
A	31	5	0.1	719.56	786.0	121.6	2.744	0.763	1.0	17	8.0	0.000
A	31	5	0.3	583.38	798.0	82.6	3.432	2.483	1.1	379	92.8	0.000
A	32	5	0.1	603.27	661.0	83.7	1.833	0.774	0.4	131	12.2	0.000
A	32	5	0.3	486.61	664.0	72.7	2.498	0.857	0.5	25	6.8	0.000
A	32	6	0.1	678.46	744.0	88.6	3.781	0.766	0.3	223	17.8	0.000
A	32	6	0.3	543.18	750.0	60.6	2.450	0.686	0.4	31	6.0	0.000
A	33	5	0.1	716.08	780.0	140.8	5.350	3.596	0.7	61183	6427.9	0.000
A	33	5	0.3	573.74	794.0	59.8	4.578	1.887	0.9	1155	184.4	0.000
A	35	5	0.1	732.10	799.0	130.0	2.956	1.210	1.5	2323	597.2	0.000
A	35	5	0.3	594.70	811.0	90.0	3.231	1.929	2.2	2795	1919.3	0.000
A	36	5	0.1	614.71	670.0	117.1	2.484	1.147	2.6	277	61.5	0.000
A	36	5	0.3	497.43	684.0	62.1	2.336	1.650	3.0	341	279.3	0.000
A	36	6	0.1	868.00	952.0	112.0	3.625	1.741	0.9	11669	1747.0	0.000
A	36	6	0.3	695.20	961.0	75.0	3.260	1.982	1.1	2129	1105.9	0.000
A	37	5	0.1	666.82	731.0	89.2	6.265	3.024	1.3	9925	1892.3	0.000
A	37	5	0.3	534.66	738.0	60.2	6.535	3.096	1.5	5121	2251.5	0.000
A	38	5	0.1	756.13	826.0	127.3	2.671	1.529	2.1	2161	1253.9	0.000
A	38	5	0.3	614.19	840.0	87.3	2.213	0.852	3.8	53	385.8	0.000
A	38	6	0.1	765.62	835.0	141.2	4.062	2.173	1.1	9457	1947.4	0.000
A	38	6	0.3	615.86	845.0	81.2	2.522	0.644	1.7	39	53.8	0.000
A	43	6	0.1	852.27	937.0	89.7	0.770	0.497	1.5	87	28.5	0.000
A	43	6	0.3	682.81	937.0	89.7	0.411	-0.000	1.6	1	7.4	0.000
A	44	6	0.1	862.88	948.0	96.8	3.552	0.994	2.9	239	165.1	0.000
A	44	6	0.3	691.54	949.0	90.8	3.358	0.272	4.0	17	147.1	0.000
A	45	7	0.1	833.98	917.0	86.8	1.671	0.523	1.8	59	24.9	0.000
A	45	7	0.3	665.34	927.0	54.8	1.225	0.192	2.8	7	18.8	0.000
A	47	7	0.1	978.80	1075.0	113.0	3.122	1.087	1.6	1781	1023.8	0.000
A	47	7	0.3	786.00	1080.0	100.0	3.490	1.233	3.4	861	4130.8	0.000

CGOPT user guide

For the sake of reproducibility, the CGOPT website <https://cgopt.gerad.ca> offers the possibility to run the algorithms $BPC-SP1_\alpha$ (for the general case) and $BPC-SP2_\alpha^{adj}$ (for the adjacent case) on the instances used for this paper or on any other supplied instances. This user guide specifies how this can be accomplished.

The CGOPT website contains the pages *Introduction*, *Input file specifications*, *Your projects*, *Project details*, and *Description of the project output archive*, which are described below.

Introduction

The *Introduction* page gives summary information on the CGOPT website. In particular, it specifies the time limits imposed for solving the instances (two hours for the adjacent case and five hours for the general case), which are the same as those considered in this paper. It also indicates the specifications of the computer on which the algorithms are executed. Note that this computer is slightly slower than the one used in this paper. To ensure that all tests are run in an optimal setting, CGOPT executes one job at a time, using a single thread.

The CGOPT website is open to the research community. However, to run tests, any user must first create an account by clicking on *Sign up* in the top menu and filling out a form to provide his/her name, affiliation, and email address. After validating its researcher status, an email is sent to the user to confirm the creation of his/her account.

Input file specifications

To run tests, a user must submit input files in a project ZIP archive. The *Input file specifications* page describes the content of such an archive.

We refer to an optimization job (of one or more instances) as a project. A project input file consists of a set of input data files bundled into a single, not password protected, ZIP archive. A limit of 100 instances per project is imposed.

The structure of a typical project ZIP archive is as follows.

- It contains a root folder that must be named *input*.
- In this *input* folder, there is one *instance* subfolder for each instance.
- In each *instance* subfolder, there are four data files in text format that must be named *Param.txt*, *Demand.txt*, *Exp_mat.txt*, and *Cov_mat.txt*.

The following rules must be respected.

- Instance folder names must start with an alphanumeric character without diacritics; for the remaining characters, only alphanumeric characters without diacritics, underscores, hyphens and dots are accepted. Spaces are specifically prohibited.

- File/folder names are case sensitive.
- Files/folders other than those mentioned above are strictly prohibited.
- Special file types such as Unix/NTFS symbolic links and Windows shortcuts are not supported and will mistakenly be considered as text files whose content will probably not match the given specifications.

The content of each of the four files for a given instance is described as follows.

Param.txt is a parameter file that includes the following information:

1. The real number $\alpha \in]0,1[$ used to generate the convex combination of the expected travel time and the travel time variance;
2. The instance type: 'A' for the adjacent case and 'G' for the general case.

File example:

```
alpha  0.1
type   A
```

Note that the words alpha and type must be included in the file.

Demand.txt is a file with $n + 2$ rows, where the first row gives the vehicle capacity Q (a positive integer), while rows 2 to $n + 2$ list the customers with their demands (all positive integers). In fact, row 2 represents the depot with demand 0. Note that the first entry of row $j \in \{2, \dots, n + 2\}$ must be $j - 1$.

File example with $n = 3$ customers and $Q = 200$:

```
200
1   0
2  10
3  30
4  10
```

Exp_mat.txt is a matrix of dimensions $(n + 1) \times (n + 1)$ with rows and columns numbered from 0 to n , where the element of row i and column j represents the expected travel time between customers

i and j (a positive integer, except on the main diagonal where it must be 0). Note that row 0 and column 0 correspond to the depot.

File example with $n = 3$ customers:

```

0 22 12 18
21 0 * *
* * 0 *
* * * 0

```

where the * should be replaced by positive integers.

Cov_mat.txt is a covariance matrix \mathcal{C} of dimensions $m \times m$, where $m = n(n-1) + 2n$ is the number of arcs. The ordered list of the m arcs is given by:

$$\begin{array}{cccccc}
 (0, 1), & (0, 2), & \dots, & (0, n-1), & (0, n), \\
 (1, 2), & (1, 3), & \dots, & (1, n), & \\
 (2, 1), & (2, 3), & \dots, & (2, n), & \\
 \vdots & \vdots & \vdots & \vdots & \\
 (n, 1), & (n, 2), & \dots, & (n, n-1), & \\
 (1, n+1), & (2, n+1), & \dots, & (n-1, n+1), & (n, n+1)
 \end{array}$$

where 0 and $n+1$ represent the depot and $\{1, 2, \dots, n\}$ is the customer set.

File example with $n = 3$ customers and, thus, $m = 12$ arcs. The covariance matrix has 144 entries whose rows and columns are indexed by the arcs $(0, 1), (0, 2), (0, 3), (1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2), (1, 4), (2, 4), (3, 4)$:

```

2.1  0.02  0.1  -0.05  0.10.02  0  0.11  0  -0.06  0  0.02  0.3
*    *    *    *    *    *    *    *    *    *    *    *
*    *    *    *    *    *    *    *    *    *    *    *
*    *    *    *    *    *    *    *    *    *    *    *
*    *    *    *    *    *    *    *    *    *    *    *
*    *    *    *    *    *    *    *    *    *    *    *
*    *    *    *    *    *    *    *    *    *    *    *
*    *    *    *    *    *    *    *    *    *    *    *
*    *    *    *    *    *    *    *    *    *    *    *
*    *    *    *    *    *    *    *    *    *    *    *
*    *    *    *    *    *    *    *    *    *    *    *

```

where the * should be replaced by positive integers. This matrix must be symmetric and positive semidefinite.

An example of a project ZIP archive can be downloaded at the bottom of the *Project input file specifications* page, together with project archives containing the instances used in the paper and the corresponding output archives containing the computed solutions.

Your projects

Once a user is logged in, he/she can run instances by submitting a project archive in the *Your projects* page. To submit a project archive, click on *New project*. The user is then redirected to a *Create project* page where he/she must select the file to upload and click on the *Create project* button. Once the file is uploaded, the user returns to the *New project* page where he/she can see the created project in his/her list of projects (including all previously submitted projects that are not yet deleted).

For each project, this list indicates the project ID, when it was created, the name and size of its archive, and its status. The following four statuses are possible:

In queue: The optimization tasks of the project are waiting to be processed. Note that a project may stay a long time in queue if another (long) project is being run.

Running: The optimization tasks of the project are being processed. No results are available before they are all completed.

Complete: The optimization tasks are completed.

Error: The execution ended with an error.

Clicking on the project ID of a project or on the information icon at the end of a project line opens a *Project details* page which is described below. Clicking on the trashcan icon at the end of a project line deletes it.

Project details

The *Project details* page gives information about the execution of a project. This page is available as soon as the project has been submitted, i.e., even when the execution is not finished. It shows the project file name and size, when it was submitted, and the project status which is updated regularly. When available, it also indicates when the optimization process started and ended, and the current running time.

Once the execution process is completed, a summary of this process is provided in a display box, indicating for each instance in the project when the optimization started and ended, and whether it completed with success or not. Error messages, if any, are also displayed in this box.

Moreover, when the process is completed and solutions were computed, an output ZIP archive becomes available at the bottom of this page. This archive is described next.

Description of the project output archive

When unzipping the output file of a project, the user will find a folder for each instance submitted to CGOPT in this project as well as a summary file (Summary.txt) that describes how many

instances were considered and how many failed to be solved adequately. In the folder of an instance, you can either find an `Error.txt` file which reports error messages for the corresponding instance if the process did not complete properly, or the two following two files:

Report.txt reports various statistics on the solution process when completed successfully. The name of these statistics are self-explanatory. Note that the number of branch-and-bound nodes includes the root node and that the total time may exceed the time limit given that this limit is not tested everywhere in the code (e.g., not in the labeling algorithm).

Solution.txt provides the total cost of the best solution found (within the time limit), the number of vehicles used in this solution, and the route assigned to each vehicle. Each route is represented by the list of nodes it traversed. The first and last nodes `_1o` and `_1d` correspond to the (origin or destination) depot. The other nodes are customer nodes (for example, node `_8n` corresponds to customer 8).