# The Mixed Capacitated General Routing Problem under Uncertainty

**Patrizia Beraldi[a], Maria Elena Bruni[a], Demetrio Laganà[a,*], Roberto Musmanno[a]**

*[a]Department of Mechanical, Energy and Management Engineering, University of Calabria, 87036 Arcavacata di Rende (CS), Italy*

## Abstract

We study the General Routing Problem defined on a mixed graph and with stochastic demands. The problem under investigation is aimed at finding the minimum cost set of routes to satisfy a set of clients whose demand is not deterministically known. Since each vehicle has a limited capacity, the demand uncertainty occurring at some clients affects the satisfaction of the capacity constraints, that, hence, become stochastic. The contribution of this paper is twofold: firstly we present a chance-constrained integer programming formulation of the problem for which a deterministic equivalent is derived. The introduction of uncertainty into the problem poses severe computational challenges addressed by the design of a branch–and–cut algorithm, for the exact solution of limited size instances, and of a heuristic solution approach exploring promising parts of the search space. The effectiveness of the solution approaches is shown on a probabilistically constrained version of the benchmark instances proposed in the literature for the mixed capacitated general routing problem.

*Keywords:* routing problem, mixed graph, neighborhood search, probabilistic constraints

## 1. Introduction

An important operative issue in the context of the distributive logistics consists in planning the delivery routes performed by a fleet of vehicles to satisfy the

*Corresponding author

*Email addresses:* beraldi@deis.unical.it (Patrizia Beraldi), mebruni@deis.unical.it (Maria Elena Bruni), demetrio.lagana@unical.it (Demetrio Laganà), musmanno@unical.it (Roberto Musmanno)

requests of a set of elements of a network, namely required vertices, edges and arcs. In mathematical terms, the problem is modeled as *Mixed Capacitated General Routing Problem* (*MCGRP*): it basically consists in finding a set of routes on a mixed graph, beginning and ending at the same vertex (depot), with minimum total cost, satisfying demands located at links and vertices and with a capacity restriction on the demand satisfied by each route. The *MCGRP* generalizes many vehicle routing problems that have been widely studied in the last forty years and for which hundreds of papers have been written, either to give exact or heuristic procedures for their resolution or to provide lower bounds. Despite the practical importance of the mixed general routing problem, relatively few studies have been published on it. Most works deal with the uncapacitated case. Corberán et al. [1, 2, 3] studied the feasible polyhedron starting from an integer programming formulation solved through an efficient cutting–plane algorithm. Blais and Laporte [4] proposed a different approach based on the transformation of the original problem into an equivalent Traveling Salesman Problem or Rural Postman Problem which are solved in turn through available exact algorithms.

With respect to the capacitated case, Bosco et al. [5] proposed a novel integer programming formulation and a branch-and-cut algorithm (*B&C*) where surrogate inequalities, introduced for the Capacitated Arc Routing Problem, are extended to the *MCGRP* polyhedron.

The aim of this paper is the introduction of the uncertainty issue in this latter and more involved case, where each vehicle has a limited capacity. In effect, most of the real-world applications modeled as *MCGRP* are characterized by some uncertainty which affects the customers' demand. For example, the operational plan of pickup routes in solid waste collection systems implies modeling the service by the means of required arcs or edges whenever the collection points are distributed along the streets, while some vertices are required if the collection is concentrated around specific points (e.g., hospitals, schools, and supermarkets). For generality, we shall also assume that the requests of random elements might be correlated to faithfully represent real situations. For example, in the garbage collection, the geographical nearness of some customers within the same regional district or along the same street suggests to consider a statistical correlation among their garbage productions.

Following these considerations, we bring the stochasticity into the *MCGRP* by adopting the paradigm of the probabilistic constraints defined within the general Stochastic Programming (SP) framework ([6]). This modeling paradigm is appropriate in many situations, where an operational plan is periodically updated over a long planning horizon, and hence, becomes crucial to design a set of a priori routes that will cover the uncertain requests with a high reliability level. In particular, we formally introduce a stochastic formulation of the *MCGRP* where the stochastic

capacity constraints are re-formulated in terms of probabilistic constraints. The explicit inclusion of the uncertainty within an already proved NP-hard problem, poses additional challenges, calling for the design of tailored solution approaches. This represents the second core contribution of the present paper. We develop a branch-and cut ($B\&C$) algorithm for solving small instances and we design a large neighborhood search heuristic for the solution of instances of larger size where the $B\&C$algorithm is used, in turn, to perform an exact local search on a portion of the overall feasible region.

To put our contribution in the right perspective, we should precise that the adoption of the SP framework to model routing problems under uncertainty is not completely new. For an extensive survey, the readers are referred to Dror et al. [7] and Gendreau et al. [8].

Within this stream, most of the contributions rely on the two-stage paradigm and different recourse policies have been proposed in the literature. Bertsimas [9] and Bertsimas and Simchi–Levi [10] focused their researches on simple recourse policies that are separable by vehicle. A different policy has been presented by Ak and Erera [11], that proposed a two-vehicle sharing recourse policy. During the last decades various heuristic and exact optimization approaches have been proposed and analyzed for constructing a set of tours minimizing expected costs given this recourse policy. Gendreau et al. [12] proposed an exact solution for an *a priori* optimization model based on an integer L–shaped method. Laporte et al. [13] presented an improved method where strong lower bounds at the root node contribute significantly to speed up the solution times. Gendreau et al. [14] applied local search concepts embedded into a tabu search scheme to solve the a priori model presented in Gendreau et al. [12]. More recently, Laporte et al. [15] studied the capacitated arc-routing problem with stochastic demands in the context of garbage collection and proposed an adaptive large neighbourhood search heuristic.

Scant attention has been devoted to the formulation of routing problems with probabilistic constraints. Stewart and Golden [16] presented a model able to find minimum cost routes with a threshold constraint on the probability of a route failure, whereas Laporte et al. [17] proposed a chance-constrained model for location–routing problems. A chance constrained version of the vehicle routing problem, solved to optimality by algorithms similar to those developed for the deterministic case, has been presented in Dror et al. [18]

Besides the stochastic programming approach, the robust optimization framework has been adopted to deal with routing problems involving uncertain parameters where the probability distributions are not known. Amongst the recent contributions, we cite Sungur et al. [19], who analyze the case of uncertain customer demands and travel times. The goal is to determine vehicle routes which satisfy the

capacity constraints and the specified time windows if all the uncertain parameters attain the worst case realizations simultaneously. The problem can be simplified to a deterministic model, which is attractive from a computational standpoint. In [20] (see also the references therein), Gounaris et al. investigate the case of capacitated vehicle routing problem. Robust optimization counterparts of several deterministic formulations of the problem are derived and numerically compared. Robust rounded capacity inequalities are developed, which can be separated efficiently for two broad classes of demand supports. Finally, the authors analyze the relation between the robust models and the chance constrained counterparts. Lee et al. [21] considered two types of uncertainty sets for the possible realizations of travel times and demands. The authors propose a column generation algorithm which encapsulates the robustness in the pricing problem cast as a robust version of the shortest path with resource constraints.

In this paper we study the Mixed Capacitated General Routing Problem with Probabilistic Constraints *MCGRPPC*. In Section 2, we introduce the problem and we provide a chance-constrained integer linear programming formulation for the *MCGRPPC*. In Section 3 we define the *B&C*algorithm for solving small instances of the *MCGRPPC*. In Section 4 we present a tailored heuristic search to solve larger *MCGRPPC* instances. In Section 5, we present the results of our computational study. Finally, in Section 6, we give our conclusions and discuss future perspectives in this area.

## 2. Problem description

The *MCGRPPC* is defined over a mixed graph $G = (V, A, E)$, where $V = \{1, \ldots, n\}$ represents the set of vertices, including the depot, and $A = \{(i, j) \subseteq V \times V\}$ is the set of arcs, whereas $E = \{(i, j) \subseteq V \times V : i < j\}$ is the set of edges.

In the following, we shall denote by $L = A \cup E$, the set of *links* and we shall indicate by $c_{ij}$ a non-negative cost coefficient associated with each link $(i, j)$. We assume that the service activity may occur at some vertices $V_R \subseteq V$, named *required vertices*, arcs $A_R \subseteq A$ and/or edges $E_R \subseteq E$, named *required arcs* and *required edges*, respectively. Thus, $L_R = A_R \cup E_R$ denotes the set of *required links* of $G$ and all the required vertices and links will be referred to as required *elements* and indicated by $R$.

For each subset $S \subset V$ of vertices, or its complementary set $\overline{S}$ ($\overline{S} = V \setminus S$), we define the following sets:

*a)* $\delta^+(S) = \left\{(i, j) \in A : i \in S \wedge j \in \overline{S}\right\}$,

*b)* $\delta^-(S) = \left\{(i, j) \in A : i \in \overline{S} \wedge j \in S\right\}$,

*c)* $\delta^+_{A_R}(S) = \left\{(i, j) \in A_R : i \in S \wedge j \in \overline{S}\right\}$,

*d*) $\delta^-_{A_R}(S) = \left\{(i,j) \in A_R : i \in \overline{S} \wedge j \in S\right\}$,
*e*) $\delta(S) = \left\{(i,j) \in E : i \in S \wedge j \in \overline{S},\ \texttt{or}\ i \in \overline{S} \wedge j \in S\right\}$,
*f*) $\delta_{E_R}(S) = \left\{(i,j) \in E_R : i \in S \wedge j \in \overline{S},\ \texttt{or}\ i \in \overline{S} \wedge j \in S\right\}$,
*g*) $\delta_L(S) = \delta^+(S) \cup \delta^-(S) \cup \delta(S)$,
*h*) $\delta_{L_R}(S) = \delta^+_{A_R}(S) \cup \delta^-_{A_R}(S) \cup \delta_{E_R}(S)$,
*i*) $S_R = S \cap V_R$, $A_R(S) = \{(i,j) \in A_R : i \in S \wedge j \in S\}$,
*j*) $A_R(S) = \{(i,j) \in A_R : i \in S \wedge j \in S\}$,
*k*) $E_R(S) = \{(i,j) \in E_R : i \in S \wedge j \in S\}$,
*l*) $R(S) = A_R(S) \cup E_R(S) \cup S_R$.
The previous notation remains valid as long as $S$ is replaced by $v$, and $\overline{S}$ by $\overline{v}$, or $V\backslash\{v\}$. We denote by $G^R$ the graph induced on $G$ by all the required links and vertices. Generally, this graph is non-connected. The vertex sets corresponding to connected components of $G^R$ are called $R$-sets. The subgraphs of $G$ induced by the $R$-sets define the so-called $R$-connected components of $G$. An isolated required vertex represents itself an $R$-connected component of $G$.

In real settings, the service demand associated with all but a subset of required elements is seldom, if ever, known at the time routes have to be designed. Thus, with the aim of more realistically modeling general routing problems, one should deal with the stochastic nature of the input parameters. In the following, we shall assume that the set of required elements is partitioned into two subsets $R_C$ and $R_U$ to differentiate between elements with known and uncertain demands, respectively. Following the stochastic programming modeling framework, we shall assume that the uncertain demands are represented in terms of random variables defined on a given probability space $(\Omega, F, \mathbb{P})$. Thus, we shall denote by $d_i(\omega)$ and $d_{ij}(\omega)$ the random demands associated with the "stochastic" required vertices and links, respectively. Deterministic demands are denoted analogously but without explicating the dependence on $\omega$. The introduction of the random variables within the mathematical formulation makes the classical criterion of feasibility, introduced for the deterministic case, no longer valid, since it may happen that the stochastic capacity constraints are satisfied for some elements $\omega \in \Omega$, but not for all. To deal with this issue, we adopt the paradigm of probabilistic constraints, determining a set of a priori routes of minimum cost satisfying the random demands with a fixed reliability level $\alpha$. In the next subsection, we shall introduce the mathematical formulation.

### *2.1. The probabilistically constrained formulation*

For each required link $(i,j) \in L_R$ and each vehicle $k = 1, \ldots, m$, we denote by $x^k_{ij}$ the binary variable equal to 1 if $(i,j)$ is serviced by vehicle $k$ which travels from vertex $i$ to vertex $j$ and 0 otherwise, and by $y^k_{ij}$ the non-negative variable representing the number of deadheading traversals from vertex $i$ to vertex $j$ by

vehicle $k$, i.e., the number of times that link $(i, j) \in L_R$ is traversed from vertex $i$ to vertex $j$ by vehicle $k$ without being serviced.

Moreover, for each required vertex $i \in V_R$ and each vehicle $k$, we denote by $x_i^k$ the binary variable equal to 1 if $i$ is serviced by $k$ and 0 otherwise.

The following constraints hold:

$$\sum_{k=1}^{m}(x_{ij}^k + x_{ji}^k) = 1, \qquad (i,j) \in E_R \quad (1a)$$

$$\sum_{k=1}^{m} x_{ij}^k = 1, \qquad (i,j) \in A_R \quad (1b)$$

$$\sum_{k=1}^{m} x_i^k = 1, \qquad i \in V_R \quad (1c)$$

$$\mathbb{P}\Big(\sum_{(i,j)\in E_R\cap R_U} d_{ij}(\omega)(x_{ij}^k + x_{ji}^k) + \sum_{(i,j)\in A_R\cap R_U} d_{ij}(\omega)x_{ij}^k + \sum_{i\in V_R\cap R_U} d_i(\omega)x_i^k + \sum_{(i,j)\in E_R\cap R_C} d_{ij}(x_{ij}^k + x_{ji}^k) + \sum_{(i,j)\in A_R\cap R_C} d_{ij}x_{ij}^k + \sum_{i\in V_R\cap R_C} d_i x_i^k \le Q\Big) \ge \alpha, \quad k = 1,\dots,m \quad (1d)$$

$$\sum_{j:(i,j)\in\delta^+_{A_R}(i)} x_{ij}^k + \sum_{j:(i,j)\in\delta^+(i)} y_{ij}^k - \sum_{j:(j,i)\in\delta^-_{A_R}(i)} x_{ji}^k - \sum_{j:(j,i)\in\delta^-(i)} y_{ji}^k = \sum_{j:(i,j)\in\delta_{E_R}(i)} x_{ji}^k + \sum_{j:(i,j)\in\delta(i)} y_{ji}^k - \sum_{j:(i,j)\in\delta_{E_R}(i)} x_{ij}^k - \sum_{j:(i,j)\in\delta(i)} y_{ij}^k, \quad k = 1,\dots,m,\ i \in V \quad (1e)$$

$$\sum_{(i,j)\in\delta^+_{A_R}(S)} x_{ij}^k + \sum_{(j,i)\in\delta^-_{A_R}(S)} x_{ji}^k + \sum_{(i,j)\in\delta_{E_R}(S)} (x_{ij}^k + x_{ji}^k) + \sum_{(i,j)\in\delta^+(S)} y_{ij}^k + \sum_{(j,i)\in\delta^-(S)} y_{ji}^k + \sum_{(i,j)\in\delta(S)} (y_{ij}^k + y_{ji}^k) \ge \begin{cases} 2(x_{uv}^k + x_{vu}^k), & (u,v) \in E_R(S), \\ 2x_{uv}^k, & (u,v) \in A_R(S), \\ 2x_h^k, & h \in S_R, \end{cases} \quad k = 1,\dots,m,\ S \subseteq V\backslash\{1\} \quad (1f)$$

$$x_{ij}^k \in \{0,1\}, \qquad k = 1,\dots,m,\ (i,j) \in A_R \cup E_R \quad (1g)$$

$$x_{ji}^k \in \{0,1\}, \qquad k = 1,\dots,m,\ (i,j) \in E_R \quad (1h)$$

$$y_{ij}^k \in \mathcal{Z}_+, \qquad k = 1,\dots,m,\ (i,j) \in A \cup E \quad (1i)$$

$$y_{ji}^k \in \mathcal{Z}_+, \qquad k = 1,\dots,m,\ (i,j) \in E \quad (1j)$$

$$x_i^k \in \{0,1\}, \qquad k = 1,\dots,m,\ i \in V_R. \quad (1k)$$

Constraints (1a)-(1c) ensure that each request is serviced exactly once by exactly one vehicle (*assignment constraints*). Constraints (1d) are the *probabilistic capacity constraints* imposing, for each route, that the probability of not exceeding the vehicle capacity should be greater than or equal to a reliability level $\alpha$. Inequalities (1e) represent *flow constraints*. They model the symmetry conditions at each vertex. Note that, together with the integrality conditions, such constraints also imply parity conditions at each vertex.

Constraints (1f) are *connectivity constraints*. They impose that for each subset of vertices (excluding the depot) containing a required link or vertex serviced by a vehicle, at least two links incident to the subset must be used to visit it (deadheaded or serviced); they also eliminate subtours not connected with the depot.

The selection of the optimal route is guided by the minimization of the total cost, expressed by the sum of the total service cost $f_1$ and the total deadheading cost $f_2$:

$$\text{Min } f = f_1 + f_2 \tag{2}$$

$$f_1 = \sum_{k=1}^{m} \sum_{(i,j)\in E_R} c_{ij}(x_{ij}^k + x_{ji}^k) + \sum_{k=1}^{m} \sum_{(i,j)\in A_R} c_{ij} x_{ij}^k$$

$$f_2 = \sum_{k=1}^{m} \sum_{(i,j)\in E} c_{ij}(y_{ij}^k + y_{ji}^k) + \sum_{k=1}^{m} \sum_{(i,j)\in A} c_{ij} y_{ij}^k$$

The feasibility region of model (1) can be rewritten in a compact form as:

$$\begin{aligned} X = \{&x \in \{0,1\}^{(|A_R|+2|E_R|+|V_R|)\times m}, y \in \mathbb{Z}_+^{(|A|+2|E|)\times m} \mid \\ &x, y \in X', \mathbb{P}(D(\omega)x^k \le Q) \ge \alpha, k = 1,\dots,m\}. \end{aligned}$$

Here $X'$ is the feasible set defined by the deterministic constraints (1a)-(1f), where $D$ denotes a row vector having the following structure:

$$D = \left( \underbrace{d_{ij}(\omega)\dots}_{(i,j)\in E_R\cap R_U} \quad \underbrace{d_{ij}(\omega)\dots}_{(i,j)\in A_R\cap R_U} \quad \underbrace{d_i(\omega)\dots}_{i\in V_R\cap R_U} \;\middle|\; \underbrace{d_{ij}\dots}_{(i,j)\in E_R\cap R_C} \quad \underbrace{d_{ij}\dots}_{(i,j)\in A_R\cap R_C} \quad \underbrace{d_i\dots}_{i\in V_R\cap R_C} \right).$$

The formulation introduced above imposes individual chance constraints on each vehicle. This condition provides a guarantee that any individual route is feasible (with a reliability level $\alpha$), but it does not account for the performance of the entire fleet. Such an issue could be addressed by means of the joint probabilistic constraints, posing additional theoretical and computational challenges. Our formulation may constitute a safe approximation of the joint case by setting appropriately the values of the reliability parameter. Indeed, by Bonferroni's inequality, a sufficient condition for ensuring feasibility in the joint chance constrained problem, is to divide the joint probability level $\epsilon$ among the $m$ individual chance constraints and letting, for instance, $\alpha = \epsilon/m$ ([22]).

Notwithstanding the individual nature of the chance constraints, the proposed model poses several challenges since it belongs to the class of integer problems under probabilistic constraints, for which the literature is rather scarce. Problems involving discrete random variables, arising either directly or as empirical approximation of the continuous ones, have been studied by [23, 24, 25, 26, 27, 28]. Here both exact and heuristic approaches have been proposed and tested.

The case of general continuous random variables is even less investigated. In [29] deterministic reformulations are analyzed for the case of independently distributed random variables, whereas in [30] the author studies valid inequalities for the problem with individual probabilistic constraints with uncertainty in both sides.

In this paper, we assume that the random variables follow a multivariate normal distribution. While this assumption might appear restrictive, it often provides an accurate approximation of different probabilistic assumption, because of the well-known Central Limit Theorem and its variants. Under the above assumption, the chance constraints can be equivalently rewritten as second-order cone constraints. In the next subsection, we introduce our derivations.

### *2.2. The deterministic equivalent formulation*

Let $d_e$, $e \in R_U$ denote the random demands which assume normally distributed with mean $\mu_e$, variance $\sigma_e^2$, and let $\theta_{e,e'}$ denote the covariance between $e$ and $e'$. The probabilistic constraints can be rewritten as (see, e.g. Section 4 of [29])

$$\sum_{e\in R_U} \mu_e z_e^k + \sum_{e\in R_C} d_e z_e^k + \Phi^{-1}(\alpha) \sqrt{\sum_{(e,e')\in R_U^2} \theta_{e,e'} z_e^k z_{e'}^k} \le Q \tag{3}$$

where

$$
\begin{aligned}
&z_e^k = x_{ij}^k + x_{ji}^k \text{ if } e = (i,j) \in E_R,\\
&z_e^k = x_{ij}^k \text{ if } e = (i,j) \in A_R,\\
&z_e^k = x_i^k \text{ if } e = i \in V_R\}.
\end{aligned} \tag{4}
$$

By assuming that $d_e = \mu_e$ and $\sigma_e = 0$ whenever $e \in R_C$, and $\theta_{e,e'} = 0$ when either $e \in R_C$ or $e' \in R_C$, we can simplify the above relation as follows:

$$
\sum_{e \in R} \mu_e z_e^k + \Phi^{-1}(\alpha) \sqrt{\sum_{(e,e') \in R^2} \theta_{e,e'} z_e^k z_{e'}^k} \leq Q, \tag{5}
$$

that is equivalent to:

$$
\begin{cases}
\sum_{e \in R} \mu_e z_e^k \leq Q, \\
\left[\Phi^{-1}(\alpha)\right]^2 \left(\sum_{e \in R} \sigma_e^2 z_e^k + \sum_{(e,e') \in R^2 | e \neq e'} \theta_{e,e'} z_e^k z_{e'}^k\right) \leq \left(Q - \sum_{e \in R} \mu_e z_e^k\right)^2.
\end{cases}
$$

Since $z_e^k$, are binary variables, the latter inequality can be linearized using classical techniques (see also Section 4 in [25]), so that (3) can be rewritten as:

$$
\begin{aligned}
X^k = &\{z^k \in \{0,1\}^{|R|} \mid \sum_{e \in R} \mu_e z_e^k \leq Q, \\
&\sum_{e \in R} \left(\left[\Phi^{-1}(\alpha)\right]^2 \sigma_e^2 + (2Q - \mu_e)\mu_e\right) z_e^k \leq \\
&Q^2 + \sum_{(e,e') \in R^2 | e \neq e'} \left(\mu_e \mu_{e'} - \left[\Phi^{-1}(\alpha)\right]^2 \theta_{e,e'}\right) w_{e,e'}^k, \\
&w_{e,e'}^k \leq z_e^k,\ w_{e,e'}^k \leq z_{e'}^k,\ w_{e,e'}^k \geq z_e^k + z_{e'}^k - 1,\ (e,e') \in R^2,\ e \neq e', \\
&z_e^k = x_{ij}^k + x_{ji}^k \text{ if } e = (i,j) \in E_R, \\
&z_e^k = x_{ij}^k \text{ if } e = (i,j) \in A_R, \\
&z_e^k = x_i^k \text{ if } e = i \in V_R\}.
\end{aligned} \tag{6}
\tag{7}
$$

where the binary variables $w_{e,e'}^k$ act to linearize $z_e^k z_{e'}^k$. As a result, an equivalent deterministic integer programming formulation of the *MCGRPPC* is obtained by replacing each probabilistic constraint (1d) with the set of additional variables and constraints given by (6). More precisely, it is $\frac{(|R|^2 - |R|)}{2} + |R|$ variables and $2 + \frac{3}{2}(|R|^2 - |R|) + |R|$ constraints. The complexity of the above model increases quickly with the number of vehicles and the number of uncertain required elements.

**Remark** The reformulation (5) and the successive derivations also hold for the significant class of radial distribution ([31]), for which the probability constraints can be converted explicitly into convex second-order cone constraints of type (5). It is worth to highlight also the close relation that exists between the constraints (5) and the explicit deterministic counterparts of distributionally robust chance constraints, which are enforced over an entire family of probability distributions. In particular, for the family composed of all distributions having given mean and covariance, the distributionally robust constraint can be converted, once again, into an explicit second-order cone constraint of the type (5). This broadens the applicability of the approach presented in this paper to several interesting contexts.

In the following section we present a branch-and-cut algorithm to optimally solve instances with a small number of vehicles and a limited number of required elements.

## 3. A branch-and-cut algorithm

In this section, we present the branch and cut algorithm designed to optimally solve the *MCGRPPC*. We mainly focus on the crucial aspects of (i) the determination of an initial feasible solution to use as upper bound and (ii) the definition of valid inequalities. The outline of the algorithm is provided by Algorithm 1 in the Appendix.

### *3.1. Initial solution*

In order to design an efficient branch-and-cut algorithm, it is also important also to have a good initial solution, that we built on the basis of the "partition-first-route-next" paradigm. A first attempt to obtain a good feasible partition relies on solving a probabilistic version of the capacitated concentrator location-based problem (see [32] and [33]) with stochastic demands, in which several required elements are selected as concentrator locations, named "seeds", and the remaining required elements are aggregated around each seed, while respecting, amongst the other deterministic constraints reported in [32] also the probabilistic capacity constraints. The number of required elements selected as concentrators must be equal to $m$, initially set to $\left\lceil \frac{d_T(\alpha)}{Q} \right\rceil$, where $d_T(\alpha)$ represents the $\alpha$-quantile of the random variables representing the total demand of the required elements. If the model is infeasible because $m$ is inadequate, then $m$ is increased to ensure feasibility.

The goal consists in minimizing the total assignment cost of all the required elements to the selected seeds computed as the average of all the costs associated with the shortest paths linking the endpoints of two required elements.

The solution of the above model returns a set of clusters defining a partition of all the required elements. Each cluster represents an instance of the mixed general routing problem that is solved through the *B&C* algorithm.

This partitioning model does not adequately take into account the routing cost associated with each cluster. To improve the routing cost estimation an iterative scheme is designed, in which a set of diversification constraints is added dynamically to the partitioning model with the aim of selecting other seeds around which different clusters are generated and possible better routes are built. Such diversification constraints aim at exploring more promising portions of the search space according to the classical Variable Neighborhood Search (*VNS*) scheme (see, e.g. [34]). More precisely, given a feasible partition of all the required elements, we identify the seed generating the cluster with the highest routing cost and we impose that such a seed cannot be selected for the next $\zeta$ consecutive iterations. We solve again the probabilistic version of the capacitated concentrator location-based problem with this additional constraint and repeat the diversification procedure by exiting with the best cost solution obtained within a maximum number of iterations, say $\mathcal{I}$. Observe that all the required elements that are prevented to become a seed can be selected again as a concentrator after $\zeta$ iterations. In our implementation we set $\mathcal{I}$ as the minimum between the number of all the required elements and a threshold equal to 10, while $\zeta = \left\lceil \frac{\mathcal{I}}{2} \right\rceil$.

Alternatively, a feasible partition can be built heuristically as follows:

a) a set of seeds is defined by identifying at each step the one that is the unclustered required element farthest from the depot and the other seeds;
b) a cluster of required elements closest to each seed is generated in such a way that the overall demand associated with this cluster satisfies the probabilistic capacity constraint.

The routing of the required elements collected in each cluster is obtained by using the *B&C* algorithm. In order to define a starting number of vehicles, we set $m$ equal to the number of vehicles in the minimum cost solution chosen between the best solutions returned by both heuristic procedures.

### *3.2. Valid inequalities*

The initial *LP* relaxation includes the equations (1a), (1b),(1c),(6),(1e), one connectivity inequality (1f) for each *R*–set, and some additional connectivity and *R*–odd cut inequalities that are identified according to the *Sequence of edge cutsets procedure* described by Belenguer and Benavent [35]. The separation problem associated with connectivity inequalities is solvable heuristically through a modification of the heuristic procedure presented by Fischetti et al. [36], and to optimality by means of polynomial time max–flow calculations.

Besides these well-known cuts, we also derive some additional cuts exploiting the reformulation of the problem, which take as a basis the capacity inequalities proposed by Belenguer and Benavent [35].

$$\sum_{(i,j)\in\delta_L(S)} \eta_{ij} \geq 2\left\lceil \frac{D(R(S),\delta_{L_R}(S))}{Q} \right\rceil - |\delta_{L_R}(S)|, \quad S \subseteq V\backslash\{1\}, \tag{8}$$

where $D(R(S),\delta_{L_R}(S)) = \sum_{e\in R(S)\cup\delta_{L_R}(S)} \mu_e$ with

$$\eta_{ij} = \begin{cases} \sum_{k=1}^{m}(y_{ij}^k + y_{ji}^k) & \text{if } (i,j)\in E \\ \sum_{k=1}^{m} y_{ij}^k & \text{if } (i,j)\in A. \end{cases}$$

These inequalities can be adapted to our problem as follows:

$$\mathbb{P}\left(2\left\lceil \frac{D(R(S,\omega),\delta_{L_R}(S,\omega))}{Q} \right\rceil \leq \sum_{(i,j)\in\delta_L(S)} \eta_{ij} + |\delta_{L_R}(S)|\right) \geq \alpha, \quad S \subseteq V\backslash\{1\}, \tag{9}$$

where $D(R(S,\omega),\delta_{L_R}(S,\omega))$ represents the demand associated with the stochastic and deterministic required elements inside $S$ denoted by $R(S,\omega)$, and $\delta_{L_R}(S,\omega)$ indicates the stochastic and deterministic required links with one endpoint in $S$ and the other outside of $S$.

To the best of our knowledge, no polynomial algorithm exists to exactly solve the separation problem (8). However, a max–flow algorithm can be used to solve the separation problem of the so-called *fractional capacity inequalities* (see [35]). In our context,we have to deal with the separation problem of the following probabilistic fractional capacity inequalities:

$$\mathbb{P}\left(2\frac{D(R(S,\omega),\delta_{L_R}(S,\omega))}{Q} \leq \sum_{(i,j)\in\delta_L(S)} \eta_{ij} + |\delta_{L_R}(S)|\right) \geq \alpha, \quad S \subseteq V\backslash\{1\}, \tag{10}$$

that can be transformed in the following way:

$$\sum_{(i,j)\in\delta_L(S)} \eta_{ij} \geq 2\frac{D(R(S),\delta_{L_R}(S))}{Q} - |\delta_{L_R}(S)|, \quad \text{and} \tag{11a}$$

$$\frac{4}{Q^2}\left[\Phi^{-1}(\alpha)\right]^2\left(\sum_{e\in R(S)\cup\delta_{L_R}(S)}\sigma_e^2+\sum_{(e,e')\in(R(S)\cup\delta_{L_R}(S))^2|e\neq e'}\theta_{e,e'}\right)\leq$$

$$2\left(|\delta_{L_R}(S)|-2\frac{D(R(S),\delta_{L_R}(S))}{Q}\right)\sum_{(i,j)\in\delta_L(S)}\eta_{ij}+\left(2\frac{D(R(S),\delta_{L_R}(S))}{Q}-|\delta_{L_R}(S)|\right)^2+$$

$$+\left(\sum_{(i,j)\in\delta_L(S)}\eta_{ij}\right)^2,\quad S\subseteq V\backslash\{1\}\tag{11b}$$

The following property holds.

**Property 3.1.** *For $S\subseteq V\backslash\{1\}$, if the following linear inequalities hold:*

*(l1)* $\sum_{(i,j)\in\delta_L(S)}\eta_{ij}\geq 2\frac{D(R(S),\delta_{L_R}(S))}{Q}-|\delta_{L_R}(S)|$, *and*

*(l2)* $\frac{4}{Q^2}\left[\Phi^{-1}(\alpha)\right]^2\left(\sum_{e\in R(S)\cup\delta_{L_R}(S)}\sigma_e^2+\sum_{(e,e')\in(R(S)\cup\delta_{L_R}(S))^2|e\neq e'}\theta_{e,e'}\right)\leq$
$2\left(|\delta_{L_R}(S)|-2\frac{D(R(S),\delta_{L_R}(S))}{Q}\right)\sum_{(i,j)\in\delta_L(S)}\eta_{ij}+2\left(2\frac{D(R(S),\delta_{L_R}(S))}{Q}-|\delta_{L_R}(S)|\right)^2$,

*then* (10) *is valid for $S$.*

*Proof.* From (*l*1), (*l*2), and for any $S\subseteq V\backslash\{1\}$, it follows that:

$$\frac{4}{Q^2}\left[\Phi^{-1}(\alpha)\right]^2\left(\sum_{e\in R(S)\cup\delta_{L_R}(S)}\sigma_e^2+\sum_{(e,e')\in(R(S)\cup\delta_{L_R}(S))^2|e\neq e'}\theta_{e,e'}\right)\leq$$

$$2\left(|\delta_{L_R}(S)|-2\frac{D(R(S),\delta_{L_R}(S))}{Q}\right)\sum_{(i,j)\in\delta_L(S)}\eta_{ij}+2\left(2\frac{D(R(S),\delta_{L_R}(S))}{Q}-|\delta_{L_R}(S)|\right)^2=$$

$$2\left(|\delta_{L_R}(S)|-2\frac{D(R(S),\delta_{L_R}(S))}{Q}\right)\sum_{(i,j)\in\delta_L(S)}\eta_{ij}+\left(2\frac{D(R(S),\delta_{L_R}(S))}{Q}-|\delta_{L_R}(S)|\right)^2+$$

$$+\left(2\frac{D(R(S),\delta_{L_R}(S))}{Q}-|\delta_{L_R}(S)|\right)^2\leq$$

$$2\left(|\delta_{L_R}(S)|-2\frac{D(R(S),\delta_{L_R}(S))}{Q}\right)\sum_{(i,j)\in\delta_L(S)}\eta_{ij}+2\left(\sum_{(i,j)\in\delta_L(S)}\eta_{ij}\right)^2,\tag{12}$$

that is inequality (11b). □

Observe that (*l*1) and (*l*2) are sufficient but not necessary conditions to ensure that the probabilistic inequality (10) holds. An example of violation of the condition is provided in the Appendix. Observe that, whenever all the demands are deterministic, the probabilistic inequality (10) reduces to the deterministic fractional

capacity inequality expressed by (11a). In such a case, condition ($l1$) is necessary and sufficient to ensure that (10) holds.

The following theorem states that conditions expressed by Property (3.1) are sufficient to ensure that inequality (10) holds. Hence, no violation of (10) occurs if no violations of ($l1$) and ($l2$) are checked.

**Theorem 3.1.** *For any $S \subseteq V\backslash\{1\}$, let $\epsilon = 2\frac{D(R(S),\delta_{L_R}(S))}{Q} - |\delta_{L_R}(S)| \geq 0$ be the fractional minimum number of deadheading traversals in the cutset $\delta_{L_R}(S)$ to service the overall average demand in $R(S) \cup \delta_{L_R}(S)$, then the probabilistic inequality* (10) *is valid for the MCGRPPC if conditions* ($l1$) *and* ($l2$) *are satisfied.*

*Proof.* From condition ($l2$), it follows that:

$$\frac{4}{Q^2}\left[\Phi^{-1}(\alpha)\right]^2\left(\sum_{e\in R(S)\cup\delta_{L_R}(S)}\sigma_e^2 + \sum_{(e,e')\in(R(S)\cup\delta_{L_R}(S))^2|e\neq e'}\theta_{e,e'}\right) \leq -2\epsilon\sum_{(i,j)\in\delta_L(S)}\eta_{ij} + 2\epsilon^2.$$

Condition ($l1$) implies that $\sum_{(i,j)\in\delta_L(S)}\eta_{ij} \geq \epsilon \geq 0$, therefore the following inequalities hold

$$\frac{4}{Q^2}\left[\Phi^{-1}(\alpha)\right]^2\left(\sum_{e\in R(S)\cup\delta_{L_R}(S)}\sigma_e^2 + \sum_{(e,e')\in(R(S)\cup\delta_{L_R}(S))^2|e\neq e'}\theta_{e,e'}\right) \leq -2\epsilon\sum_{(i,j)\in\delta_L(S)}\eta_{ij} + 2\epsilon^2$$

$$\leq -2\epsilon\sum_{(i,j)\in\delta_L(S)}\eta_{ij} + \epsilon^2 + \left(\sum_{(i,j)\in\delta_L(S)}\eta_{ij}\right)^2.$$

Hence, inequality (10) is valid for the *MCGRPPC*. □

If $\epsilon < 0$, then ($l2$) is satisfied for a given value of $\sum_{(i,j)\in\delta_L(S)}\eta_{ij}$.

A heuristic procedure to identify violations of the probabilistic capacity inequalities is applied at every node of the branch and cut tree, consisting in identifying $S^*$ such that ($l1$) is checked for possible violations. If a violation occurs, then $\overline{\eta}_{ij}$, $(i,j) \in \delta_L(S^*)$ is computed according to the *LP* optimal solution, and the condition ($l2$) verified. Hence, the following inequality is added to the *LP* relaxation:

$$\sum_{(i,j)\in\delta_L(S^*)}\eta_{ij} \geq 2\left\lceil\frac{\overline{D}(S^*)}{Q}\right\rceil - |\delta_{L_R}(S^*)|, \tag{13}$$

where

$$\overline{D}(S^*) = \sum_{e\in R(S^*)\cup\delta_{L_R}(S^*)}\mu_e + \Phi^{-1}(\alpha)\sqrt{\sum_{e\in R(S^*)\cup\delta_{L_R}(S^*)}\sigma_e^2 + \sum_{(e,e')\in(R(S^*)\cup\delta_{L_R}(S^*))^2|e\neq e'}\theta_{e,e'}}.$$

As usual in the branch-and-cut method, a cut pool is maintained with some cuts generated so far in the algorithm. When the cut pool has grown to a certain size (50 cuts in our implementation), it is permanently deleted.

## 4. The heuristic approach

In this section we describe a neighborhood search algorithm for the *MCGRPPC*. It relies on an outer diversification scheme, where different *types* of diversification methods are tested for different *rates*, followed by both heuristic and exact intensification phases. The diversification rate indicates the number of required elements modified in the current solution, while the diversification type specifies how to diversify the solution. After a solution is found in the outer loop, new solutions are explored through an intensification phase, after which an exact local search procedure based on the *B&C* algorithm is performed with the aim of finding the best solution in a given neighborhood.

In the next subsections we provide a detailed description of the main procedures briefly discussed above. Algorithm 2 reported in the Appendix provides a pseudocode of the overall heuristic scheme.

### *4.1. The diversification strategy*

The diversification phase consists of removing some required elements from a solution $s$, and reinserting them to the modified solution $\tilde{s}$. The rationale is that the solutions obtained by diversification represent the entry point to explore more promising portion of the feasible space.

A lot of freedom arises in designing the strategy for selecting the customers to remove from the solution, and reallocating them into $\tilde{s}$ (see. e.g. [37], [38], [39], [40]).

We implemented some of the removal and insertion heuristics presented by Pisinger and Ropke [40], but without using statistics from the search to guide the choice. A detailed description of the removal and insertion procedures is provided below:

- randomRemove($p$, $s$) selects $p$% required elements at random and removes them from the solution.
- worstRemove($p$, $s$) removes $p$% of the required elements from solution $s$ by selecting the ones whose removal returns the maximum saving.
- demandOrientedRemove($p$, $s$) removes $p$% of the required elements from solution $s$ according to a distance defined on the basis of the demands of these elements. More precisely, let $\Delta_{ij} = \left\| d_i - d_j \right\|$ be the metric associated with the required elements $i$ and $j$ with demands $d_i$ and $d_j$, respectively. (For random elements we have considered the mean value). The first required

element to remove is randomly chosen, while the second is selected as the nearest to the first, the third as the nearest to the second and so on until $p\%$ of the required elements are removed from $s$.

All the removal procedures return the list $\Gamma$ of the required elements removed from $s$ (see Algorithm 3 in the Appendix).

Concerning the insertion heuristics, the following procedures are used in combination with the removal ones:

- randomInsert($\Gamma$, $\tilde{s}$) randomly inserts in $\tilde{s}$ the elements extracted from $\Gamma$. More precisely, for each element $e \in \Gamma$, a route $r \in \tilde{s}$ is selected randomly among the routes such that the total demand associated with the already serviced required elements satisfies (1d). A random position $\pi$ is chosen inside $r$ where $e$ will be inserted and serviced. Whenever $e$ is already deadheaded in $r$, no attempt is made to select its position.
- regretInsert($\Gamma$, $\tilde{s}$) inserts into $\tilde{s}$ the elements extracted from $\Gamma$, with the aim of maximizing the *regret* associated with each element. We use the regret objective proposed by Pisinger and Ropke [40], with $k = 3$, and verify (1d).

### *4.2. The intensification phase*

Current solutions provided by the diversification phase represent a good starting point to address the search towards the construction of improved *MCGRPPC* solutions. Several neighborhoods are explored according to the basic Variable Neighborhood Descent (VND) strategy described by Hansen and Mladenovic [41] until no further improvement can be found. All these neighborhoods are searched within a mechanism allowing a restart of the search from the first neighborhood at each improving solution found. The main features of the neighborhood moves are summarized as follows:

*m*1) merge: for each couple of routes $r$ and $r'$ in the current solution $s$, an attempt is made in order to merge these routes by servicing all the required elements of $r'$ after the required elements serviced in $r$.

*m*2) $\lambda-$interchange: this move is similar to the one defined by Osman [42], and the CROSS-exchange move proposed by Taillard et al. [43]. In the computational experiments, we used $\lambda \in \{1, 2, 3, 4, 5\}$.

*m*3) interchangeInterRoute: this move is the same introduced by Savelsbergh [44] and consists of reinserting a single required element at a time in an alternate route.

*m*4) interchangeInterRouteNewPath: this move creates a new empty route in which a single required element is inserted.

*m*5) twoOptPlusInterRoute: this move corresponds to the 2-opt$^*$ move defined by Potvin and Rousseau [45].

Each move is applied with the best-accept strategy. Whenever a move generates a solution with one or more empty routes, such routes are dropped from the solution. In order to escape from local optima, we accept also infeasible solutions, penalised in the objective function by a constraint violation penalty factor (see [46] and [47]). More precisely, solutions with a number of vehicles larger than $m$ are $m$-infeasible (alternatively, $m$-feasible), whereas the ones in which one route at least (alternatively, no route) violates (1d) are $Q$-infeasible (alternatively, $Q$-feasible). Given a solution, let $\xi_m = [m(s) - m]^+$ be the number of routes in solution $s$ exceeding $m$, and

$$\xi_Q = \sum_{k=1}^{m(s)} \left( [\mu_k - Q]^+ + \left[ [\Phi^{-1}(\alpha)]^2 \sigma_k^2 - (Q - \mu_k)^2 \right]^+ \right),$$

where $[.]^+ = \max\{0, .\}$, $\mu_k$ and $\sigma_k^2$ are the mean and variance of the random demand associated with the required elements serviced by the $k$th vehicle, respectively. Then, the penalty cost associated with such a solution is defined as $\upsilon_m \xi_m + \upsilon_Q \xi_Q$, where $\upsilon_m$ and $\upsilon_Q$ are the unit penalties associated with the $m$-infeasibility or $Q$-infeasibility, or both. The modified objective function becomes $\tilde{f} = f + \upsilon_m \xi_m + \upsilon_Q \xi_Q$. Penalty $\upsilon_m$ is decreased by setting $\upsilon_m = \max\{\upsilon_m^{min}, \upsilon_m \cdot \upsilon_m^-\}$ if, after $\tau$ consecutive iterations, all the generated solutions are $m$-feasible, whereas it is increased by setting $\upsilon_m = \min\{\upsilon_m^{max}, \upsilon_m \cdot \upsilon_m^+\}$ whenever these solutions are $m$-infeasible. Similarly, penalty $\upsilon_Q$ is decreased by setting $\upsilon_Q = \max\{\upsilon_Q^{min}, \upsilon_Q \cdot \upsilon_Q^-\}$, and it is increased by setting $\upsilon_Q = \min\{\upsilon_Q^{max}, \upsilon_Q \cdot \upsilon_Q^+\}$ in the opposite case. The overall algorithm is detailed in Algorithms 4 and 5 reported in the Appendix.

### *4.3. Exact local search using integer programming*

The *B&C* algorithm presented in Section 3 cannot be used to solve real-life instances directly because of prohibitive computational times, but it can effectively solve instances with a number of vehicles not greater than six. This suggests that the *B&C* algorithm may be used successfully to improve portions of the solution. In particular, a route optimization search relying on integer programming is used as a neighborhood search scheme. With respect to other applications of this idea (see [48] and [49]), in our implementation the *B&C* is used to recover the feasibility of the schedules of some required elements. We select such schedules in a subset of couples and triplet of routes of the solution returned by the intensification phase, and we solve a restricted problem, where the schedules of the remaining routes are kept fixed. Several strategical decisions are adopted to make efficient the proposed approach. How do we select the two or three vehicles whose routes define the neighborhood? Which is the criterion to stop the neighborhood search? We choose a simple and straightforward scheme. Let $L^{dd}(k)$ be the set of links deadheaded by

vehicle $k$ in route $r_k$. Among the subsets of two vehicles that have not been selected before, we select the couple of vehicles $(k_1, k_2)$ with the following rules:

(a) $|L^{dd}(k_1) \cap L^{dd}(k_2)|$ is maximum. In case of multiple couples, we select
(b) the couple for which the maximum violation of (1d) is checked. If no violation of (1d) takes place, then we select the two vehicles with the maximal residual capacity. If more couples still exist, then
(c) we choose randomly the couple of vehicles with the maximum value of $|L^{dd}(k_1) \cap L^{dd}(k_2)|$ and the maximum violation of (1d) or the maximal residual capacity.

We stop after evaluating 30 percent of all the couples. This scheme has been evaluated on a subset of tuning instances with a number of vehicles equal to or less than six, and it was able to find the optimal solution provided by the *B&C* algorithm, or a near optimal solution. However, to deal with instances with a number of vehicles larger than six, we enlarged the neighborhood size by considering the schedules of the required elements serviced by three vehicles, and optimizing the triple selected in accordance with the above rules. To maintain tractability with respect to the computational times, we optimized a triple every $\lceil factor \cdot n_c \rceil$ couples of routes, where $factor = 0.2$ and $n_c$ is the total number of couples to be examined.

## 5. Computational experiments

In order to test the efficiency of the proposed approaches, we used a dataset of randomly generated instances with probability level $\alpha \in \{0.85, 0.95\}$. Each instance is built starting from the corresponding *mggdb* instance with $\beta = 0.25$ of the dataset designed by Bosco et al. [5] for the *MCGRP*. For each required element $e \in R$ with demand $d_e$, a random binary number is extracted to decide if $d_e$ is affected by uncertainty or not. Then, a random value from the discrete uniform distribution over $\{1, \ldots, coef \times d_e\}$, $coef = 5$, is selected as variance of the random variable $d_e(\omega)$. All the stochastic required elements in the neighborhood of the end vertices of each stochastic required link are considered to define a negative or positive correlation with this link. Correlation $\rho$ is selected randomly in the interval $[-1, 1]$. Whenever the correlation between $e, e' \in R$, $e \neq e'$, is not zero, then the relevant entry in the covariance matrix is set as $\theta_{e,e'} = \rho\sigma_e\sigma_{e'}$. Finally, if no non–zero correlation occurs, then an other attempt is made by forcing some stochastic required elements close to the end vertices of the stochastic links to have a covariance with such links. The new instances are named *mggdbsd*, where *sd* stands for stochastic demand.

Computational experiments have been carried out on a PC equipped with 2 Intel Xeon Quad Core CPUs @3.0 GHz, with 6 Gbyte RAM. The heuristic and the *B&C* algorithm have been coded in *java*. The *B&C* has been implemented by

using ILOG CPLEX library, release 12.2, where all the standard CPLEX cuts are activated. We have run each heuristic for 5 times by using the same parameter configuration.

In the tuning phase, we experimented a large range of diversification rates. We removed from 5 up to 90 percent of all the requests in each iteration. Due to the weakness of the insertion heuristics, the improvements obtained with the lowest and highest rates were very limited. Consequently we decided to reduce the range from 15 up to 60. More precisely, we observed that for small instances the best efficiency in removing requests relies on rates chosen in the set $\{15, 25, 35\}$, while for larger instances the best rates are the ones selected in the set $\{50, 60\}$. To ensure consistency, we used the same set of rates for all the instances, that is $\{15, 25, 35, 50, 60\}$.

### *5.1. Results*

The comparison between the results achieved by the proposed heuristic, named *HSGR*, and those provided by the *B&C* algorithm (with a time limit of four hours) is reported in Tables 1 and 2. The column headings are defined as follows:

- Name: instance name;
- $m$: number of vehicles in the upper bound solution;
- $|V|$: number of vertices;
- $|A|$: number of arcs;
- $|E|$: number of edges;
- $|V_R|$: number of required vertices;
- $|A_R|$: number of required arcs;
- $|E_R|$: number of required edges;
- $|V_{R_U}|$: number of vertices with stochastic demand;
- $|A_{R_U}|$: number of arcs with stochastic demand;
- $|E_{R_U}|$: number of edges with stochastic demand;
- UB: upper bound corresponding to the initial solution value;
- CON: number of added connectivity inequalities;
- CAP: number of added capacity inequalities (13);
- cost: cost of the best solution provided by the *B&C* algorithm within a time limit of four hours (optimal values are marked by an asterisk);
- GAP: percentage gap between upper and lower bounds at the termination of the *B&C* algorithm;
- veh.: number of vehicles in the best solution returned by the *B&C*;
- time: computational time (in seconds) of the *B&C* algorithm (the time limit is denoted by $--$);
- avg. cost: average solution value provided by the heuristic (over 5 experiments);

- avg. veh.: average number of vehicles provided by *HSGR* (over 5 experiments).
- best cost: cost of the best solution provided by heuristic (over 5 experiments). Whenever this cost is equal to the optimal cost, it is labeled with an asterisk, while the costs in bold are those outperforming the upper bounds returned by the *B&C* within the time limit.
- best veh.: number of vehicles of the best solution returned by the heuristic (over 5 experiments);
- avg. gap: percentage improving gap computed as $\frac{100[\text{avg. cost}-\text{cost}]}{\text{cost}}$;
- avg. time: computational time (in seconds) spent by *HSGR* (over 5 experiments) to perform one experiment;

The row *Avg.* averages key columns, while *NumB* indicates the number of best solutions found by the *B&C* and *HSGR*, respectively.

Several observations my be pointed out according to different criteria based on the number of vehicles, the prevalence in the total number of required links with respect to the total number of the required vertices, the distribution of the uncertainty within the stochastic required elements. Looking at the number of vehicles only, *HSGR* outperforms the *B&C* when the number of vehicles increases for the growing difficulty in solving the mathematical model. This emerges from instances in which the number of vehicles is greater than 6, as comfirmed by the percentage gaps reported in columns GAP in Tables 1 and 2. From routing point of view, we observe that when the number of required links makes the problem more "arc" than "node" routing based, the performance of the *B&C* is very poor compared with the proposed heuristic, and deteriorates quickly with the increase in the reliability level and the number of vehicles, as shown in instance *mggdbsd2* in which the number of required links is more than twice the number of required vertices, the number of vehicles is greater than 10 and the improvement achieved by *HSGR* increases with $\alpha$. Such a behavior is motivated by the increase in the performance of the exact route optimization strategy that has more chances to improve portions of the feasibility region due to the link–based nearness between the couple of routes to be optimized. Moreover, when the uncertainty affects only links, as for instance *mggdbsd4*, *HSGR* performs much better than the *B&C* regardless the reliability level, confirming our conjecture.

Looking at Tables 1 and 2, only in three instances the *B&C* provides an optimal solution within a limited computational time for the considered reliability levels. This happens when the number of required edges is less than the number of required arcs and vertices, since modeling a required edge through pairs of binary variables increases the difficulty in solving the mathematical model. Finally, we observe that the performance of *HSGR* varies with the distribution of uncertainty within the same number of required elements. For example, if we consider

Table 1: Computational results with $\alpha = 0.95$.

| Name | $m$ | $\|V\|$ | $\|A\|$ | $\|E\|$ | $\|V_R\|$ | $\|A_R\|$ | $\|E_R\|$ | $\|V_{R_U}\|$ | $\|A_{R_U}\|$ | $\|E_{R_U}\|$ | Branch–and–cut | | | | | | | *HSGR* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | UB | CON | PCAP | cost | GAP | veh. | time | avg. cost | avg. veh. | best cost | best veh. | avg. gap | avg. time |
| *mggdbsd1* | 10 | 12 | 34 | 5 | 6 | 12 | 3 | 4 | 4 | 1 | 443 | 408 | 805 | 443 | 0.46 | 10 | – | 421.2 | 10 | **414** | 10 | -4.92 | 1,224.427 |
| *mggdbsd2* | 13 | 12 | 40 | 6 | 6 | 15 | 4 | 3 | 10 | 1 | 566 | 391 | 422 | 566 | 0.46 | 13 | – | 482.6 | 11 | **475** | 11 | -14.73 | 907.76 |
| *mggdbsd3* | 11 | 12 | 34 | 5 | 7 | 12 | 3 | 2 | 7 | 2 | 456 | 37,302 | 6,915 | 456 | 0.46 | 11 | – | 443.4 | 11 | **438** | 11 | -2.76 | 326.35 |
| *mggdbsd4* | 10 | 11 | 30 | 4 | 4 | 11 | 3 | 0 | 8 | 2 | 494 | 11,811 | 2,097 | 494 | 0.45 | 10 | – | 442 | 8 | **442** | 8 | -10.53 | 497.67 |
| *mggdbsd5* | 11 | 13 | 40 | 6 | 5 | 15 | 4 | 3 | 11 | 2 | 566 | 17,566 | 3,598 | 547 | 0.37 | 11 | – | 501.6 | 9 | **495** | 9 | -8.30 | 1,703.24 |
| *mggdbsd6* | 10 | 12 | 34 | 5 | 6 | 12 | 3 | 2 | 5 | 2 | 473 | 937 | 362 | 473 | 0.41 | 10 | – | 472.4 | 10 | **470** | 10 | -0.13 | 288.69 |
| *mggdbsd7* | 8 | 12 | 34 | 5 | 5 | 12 | 3 | 2 | 6 | 3 | 449 | 420 | 181 | 433 | 0.37 | 8 | – | 374 | 7 | **374** | 7 | -13.63 | 364.31 |
| *mggdbsd8* | 17 | 27 | 70 | 11 | 11 | 26 | 8 | 7 | 13 | 6 | 530 | 9,726 | 783 | 530 | 0.51 | 17 | – | 455 | 15 | **448** | 15 | -14.15 | 4,207.79 |
| *mggdbsd9* | 14 | 27 | 78 | 72 | 9 | 29 | 9 | 2 | 13 | 4 | 430 | 7,435 | 2,547 | 430 | 0.43 | 14 | – | 396.8 | 14 | **393** | 14 | -7.72 | 6,913.15 |
| *mggdbsd10* | 6 | 12 | 38 | 6 | 4 | 14 | 4 | 2 | 7 | 2 | 309 | 7,825 | 5,196 | 309 | 0.18 | 6 | – | 291 | 6 | **291** | 6 | -5.83 | 834.79 |
| *mggdbsd11* | 7 | 22 | 68 | 11 | 8 | 25 | 8 | 4 | 12 | 6 | 422 | 14,162 | 7,337 | 422 | 0.24 | 7 | – | 382 | 7 | **380** | 7 | -9.48 | 9.511,25 |
| *mggdbsd12* | 9 | 13 | 36 | 5 | 6 | 13 | 3 | 3 | 7 | 1 | 610 | 13,644 | 6,525 | 610 | 0.36 | 9 | – | 573.4 | 9 | **563** | 9 | -6.00 | 741.50 |
| *mggdbsd13* | 8 | 10 | 42 | 7 | 6 | 15 | 5 | 2 | 8 | 2 | 455 | 355 | 526 | 455 | 0.31 | 8 | – | 423 | 8 | **423** | 8 | -7.03 | 1,037.62 |
| *mggdbsd14* | 6 | 7 | 32 | 5 | 5 | 12 | 3 | 3 | 6 | 0 | 112 | 270,485 | 37,157 | 108 | 0.01 | 6 | – | 108.4 | 6 | 108 | 6 | 0.37 | 137.11 |
| *mggdbsd15* | 5 | 7 | 32 | 5 | 5 | 12 | 3 | 3 | 7 | 1 | 61 | 165 | 19 | 55* | 0.00 | 5 | 4.76 | 55 | 5 | 55* | 5 | 0.00 | 4.50 |
| *mggdbsd16* | 8 | 8 | 42 | 7 | 5 | 15 | 5 | 3 | 5 | 4 | 108 | 148 | 52 | 108 | 0.13 | 8 | – | 100.4 | 8 | **100** | 8 | -7.04 | 314.18 |
| *mggdbsd17* | 7 | 8 | 42 | 7 | 5 | 15 | 5 | 3 | 9 | 3 | 77 | 1761 | 710 | 71* | 0.00 | 7 | 926.41 | 71 | 7 | 71* | 7 | 0.00 | 230.34 |
| *mggdbsd18* | 6 | 9 | 54 | 9 | 6 | 20 | 6 | 3 | 11 | 2 | 169 | 4 | 4 | 169 | 0.21 | 6 | – | 149 | 6 | **149** | 6 | -11.83 | 9.526,75 |
| *mggdbsd19* | 4 | 8 | 18 | 2 | 3 | 6 | 1 | 1 | 1 | 0 | 61 | 1063 | 140 | 61* | 0.00 | 4 | 16.44 | 61 | 4 | 61* | 4 | 0.00 | 7.00 |
| *mggdbsd20* | 6 | 11 | 34 | 5 | 5 | 12 | 3 | 1 | 7 | 1 | 139 | 471 | 11 | 130 | 0.14 | 6 | – | 122.4 | 6 | **121** | 6 | -5.85 | 302.306 |
| *mggdbsd21* | 10 | 11 | 50 | 8 | 7 | 18 | 6 | 5 | 11 | 3 | 172 | 63 | 0 | 168 | 0.22 | 10 | – | 156.4 | 10 | **156** | 10 | -6.90 | 437.14 |
| *mggdbsd22* | 11 | 11 | 66 | 11 | 6 | 24 | 8 | 3 | 11 | 4 | 181 | 197 | 46 | 176 | 0.16 | 11 | – | 168.8 | 11 | **168** | 11 | -4.09 | 2,015.75 |
| *mggdbsd23* | 17 | 11 | 84 | 13 | 8 | 31 | 9 | 4 | 20 | 5 | 208 | 267 | 121 | 208 | 0.16 | 17 | – | 197.8 | 17 | **196** | 17 | -4.90 | 1,090.50 |
| *Avg.* | | | | | | | | | | | | | | 322.70 | 0.26 | | 12,562.94 | 297.77 | | 295.35 | | -6.32 | 1.853,22 |
| *NumB.* | | | | | | | | | | | | | | 3 | | | | | | 23 | | | |

Table 2: Computational results with $\alpha = 0.85$.

| Name | $m$ | $\|V\|$ | $\|A\|$ | $\|E\|$ | $\|V_R\|$ | $\|A_R\|$ | $\|E_R\|$ | $\|V_{R_U}\|$ | $\|A_{R_U}\|$ | $\|E_{R_U}\|$ | Branch–and–cut UB | CON | PCAP | cost | GAP | veh. | time | *HSGR* avg. cost | avg. veh. | best cost | best veh. | avg. gap | avg. time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *mggdbsd1* | 8 | 12 | 34 | 5 | 6 | 12 | 3 | 4 | 4 | 1 | 434 | 47,556 | 12,944 | 414 | 0.43 | 8 | – | 370 | 8 | **368** | 8 | -10.63 | 952.00 |
| *mggdbsd2* | 10 | 12 | 40 | 6 | 6 | 15 | 4 | 3 | 10 | 1 | 483 | 381 | 576 | 483 | 0.38 | 10 | – | 427.4 | 9 | **415** | 9 | -11.51 | 2,553.54 |
| *mggdbsd3* | 9 | 12 | 34 | 5 | 7 | 12 | 3 | 2 | 7 | 2 | 391 | 33,399 | 8,355 | 391 | 0.39 | 9 | – | 364.8 | 8 | **360** | 8 | -6.70 | 618.75 |
| *mggdbsd4* | 8 | 11 | 30 | 4 | 4 | 11 | 3 | 0 | 8 | 2 | 439 | 184,122 | 29,825 | 432 | 0.37 | 8 | – | 373.6 | 6 | **371** | 6 | -13.52 | 964.54 |
| *mggdbsd5* | 9 | 13 | 40 | 6 | 5 | 15 | 4 | 3 | 11 | 2 | 503 | 16,376 | 5,055 | 503 | 0.29 | 9 | – | 483.8 | 9 | **477** | 9 | -3.82 | 2,016.51 |
| *mggdbsd6* | 8 | 12 | 34 | 5 | 6 | 12 | 3 | 2 | 5 | 2 | 430 | 63,895 | 20,708 | 419 | 0.32 | 10 | – | 373.4 | 8 | **367** | 8 | -10.88 | 551.04 |
| *mggdbsd7* | 9 | 12 | 34 | 5 | 5 | 12 | 3 | 2 | 6 | 3 | 450 | 402 | 206 | 433 | 0.35 | 9 | – | 417.6 | 8 | **409** | 8 | -3.56 | 364.31 |
| *mggdbsd8* | 13 | 27 | 70 | 11 | 11 | 26 | 8 | 7 | 13 | 6 | 514 | 19,213 | 3,531 | 453 | 0.46 | 17 | – | 408.8 | 13 | **402** | 13 | -9.76 | 3,068.50 |
| *mggdbsd9* | 13 | 27 | 78 | 72 | 9 | 29 | 9 | 2 | 13 | 4 | 399 | 7,534 | 2,313 | 399 | 0.41 | 13 | – | 363.4 | 13 | **356** | 13 | -8.92 | 6,885.3 |
| *mggdbsd10* | 6 | 12 | 38 | 6 | 4 | 14 | 4 | 2 | 7 | 2 | 319 | 471 | 29 | 302 | 0.18 | 6 | – | 288.8 | 6 | **288** | 6 | -4.37 | 7,937.19 |
| *mggdbsd11* | 6 | 22 | 68 | 11 | 8 | 25 | 8 | 4 | 12 | 6 | 429 | 1,925 | 438 | 402 | 0.21 | 6 | – | 369.2 | 6 | **368** | 6 | -8.16 | 8.278,05 |
| *mggdbsd12* | 8 | 13 | 36 | 5 | 6 | 13 | 3 | 3 | 7 | 1 | 610 | 11,089 | 10,848 | 589 | 0.37 | 8 | – | 529.8 | 8 | **527** | 8 | -10.05 | 1,032.44 |
| *mggdbsd13* | 8 | 10 | 42 | 7 | 6 | 15 | 5 | 2 | 8 | 2 | 456 | 304 | 725 | 455 | 0.21 | 8 | – | 417.6 | 8 | **408** | 8 | -8.22 | 1,668.81 |
| *mggdbsd14* | 6 | 7 | 32 | 5 | 5 | 12 | 3 | 3 | 6 | 0 | 115 | 8,347 | 4,525 | 107* | 0.00 | 6 | 1,164.20 | 108 | 6 | 108 | 6 | 0.93 | 1,669.98 |
| *mggdbsd15* | 5 | 7 | 32 | 5 | 5 | 12 | 3 | 3 | 7 | 1 | 57 | 54 | 15 | 55* | 0.00 | 4 | 3.85 | 55 | 4 | 55* | 4 | 0.00 | 7.43 |
| *mggdbsd16* | 7 | 8 | 42 | 7 | 5 | 15 | 5 | 3 | 5 | 4 | 112 | 60 | 26 | 112 | 0.17 | 7 | – | 100 | 7 | **98** | 7 | -10.71 | 661.97 |
| *mggdbsd17* | 6 | 8 | 42 | 7 | 5 | 15 | 5 | 3 | 9 | 3 | 83 | 1,761 | 710 | 71* | 0.00 | 6 | 926.41 | 71 | 6 | 71* | 6 | 0.00 | 755.68 |
| *mggdbsd18* | 6 | 9 | 54 | 9 | 6 | 20 | 6 | 3 | 11 | 2 | 170 | 20 | 2 | 170 | 0.23 | 6 | – | 146.8 | 5.4 | **144** | 5 | -13.65 | 7.704,14 |
| *mggdbsd19* | 4 | 8 | 18 | 2 | 3 | 6 | 1 | 1 | 1 | 0 | 61 | 362 | 63 | 59* | 0.00 | 4 | 24.03 | 59 | 4 | 59* | 4 | 0.00 | 6.25 |
| *mggdbsd20* | 5 | 11 | 34 | 5 | 5 | 12 | 3 | 1 | 7 | 1 | 136 | 684 | 41 | 117* | 0.00 | 5 | 13,790.76 | 117.6 | 5 | 117* | 5 | 0.51 | 200.42 |
| *mggdbsd21* | 9 | 11 | 50 | 8 | 7 | 18 | 6 | 5 | 11 | 3 | 171 | 177 | 21 | 165 | 0.21 | 9 | – | 154 | 9 | **153** | 9 | -6.67 | 613.96 |
| *mggdbsd22* | 10 | 11 | 66 | 11 | 6 | 24 | 8 | 3 | 11 | 4 | 177 | 235 | 104 | 170 | 0.17 | 10 | – | 167.2 | 10 | **167** | 10 | -1.65 | 876.62 |
| *mggdbsd23* | 15 | 11 | 84 | 13 | 8 | 31 | 9 | 4 | 20 | 5 | 205 | 211 | 101 | 203 | 0.25 | 15 | – | 192 | 14 | **192** | 14 | -5.42 | 409.364 |
| *Avg.* | | | | | | | | | | | | | | 300,17 | 0.24 | | 11,961.27 | 276.47 | | 273.04 | | -6.38 | 2.165,08 |
| *NumB.* | | | | | | | | | | | | | | 5 | | | | | | 23 | | | |

instances *mggdbsd1* and *mggdbsd6* having the same total number of stochastic elements but a different number of stochastic vertices and links, we remark that MMCGRPSD produces better results when the number of stochastic vertices is comparable with the number of stochastic links. This behavior seems reducing when the reliability level decreases as shown in Table 2. We argue that, in general, the reliability levels impact on the hardness of the problem regardless of the distribution of the uncertainty amongst the required elements.

## 6. Conclusions

We studied the mixed capacitated general routing problem under uncertainty and we proposed a probabilistically constrained formulation where capacity constraints are imposed to hold with a given reliability value. We provided a deterministic equivalent formulation under the assumption that the uncertain parameters follow a normal distribution. We designed a *B&C* algorithm for the optimal solution of instances of limited size and an efficient heuristic approach. Extensive numerical experiments have been carried out on a set of stochastic instances of benchmark test problems proposed in the literature for the deterministic case. The analysis of the numerical results has shown the efficiency of the proposed approaches. The proposed approach can be extended to deal with other general routing problems in which operative constraints come into play.

## References


1. Corberán A, Letchford A, Sanchis J. A cutting plane algorithm for the general routing problem. *Mathematical Programming, Ser A* 2001;**90**:291–316.

2. Corberán A, Romero A, Sanchis J. The mixed general routing polyhedron. *Mathematical Programming, Ser A* 2003;**96**:103–37.

3. Corberán A, Mejía G, Sanchis J. New results on the mixed general routing problem. *Operations Research* 2005;**53**:363–76.

4. Blais M, Laporte G. Exact solution of the generalized routing problem through graph transformations. *Journal of the Operational Research Society* 2003;**54**:906–10.

5. Bosco A, Laganà D, Musmanno R, Vocaturo F. Modeling and solving the mixed capacitated general routing problem. *Optimization Letters* 2013; **7**:1451–69.

6. Birge J, Louveaux F. *Introduction to Stochastic Programming*. Berlin: Springer; 1997.

7. Dror M, Trudeau P. Stochastic vehicle routing with modified savings algorithm. *European Journal of Operational Research* 1986;**23**:228–35.

8. Gendreau M, Laporte G, Seguin R. Stochastic vehicle routing. *European Journal of Operational Research* 1996;**88**:3–12.

9. Bertsimas D. A vehicle routing problem with stochastic demand. *Operations Research* 1992;**40**:574–85.

10. Bertsimas D, Simchi-Levi D. The new generation of vehicle routing research: robust algorithms addressing uncertainty. *Operations Research* 1994;**44**:286–304.

11. Ak A, Erera A. A paired-vehicle recourse strategy for the vehicle routing problem with stochastic demands. *Transportation Science* 2007;**41**:222–37.

12. Gendreau M, Laporte G, Seguin R. An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Science* 1995;**29**:143–55.

13. Laporte G, Louveaux F, van Hamme L. An integer l–shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research* 2002;**50**:415–23.

14. Gendreau M, Laporte G, Seguin R. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Operations Research* 1996;**44**:469–77.

15. Laporte G, Musmanno R, Vocaturo F. An adaptive large neighbourhood search heuristic for the capacitated arc-routing problem with stochastic demands. *Transportation Science* 2010;**44**:125–35.

16. Stewart W, Golden B. Stochastic vehicle routing: a comprehensive approach. *European Journal of Operational Research* 1983;**144**:371–85.

17. Laporte G, Louveaux F, Mercure H. Models and exact solutions for a class of stochastic location-routing problems. *European Journal of Operational Research* 1989;**39**:71–8.

18. Dror M, Laporte G, Louveaux F. Vehicle routing with stochastic demands and restricted failures. *ZOR - Methods and Models of Operations Research* 1993;**37**:273–83.

19. Sungur I, Ordóñez F, Dessouky M. A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions* 2008;**40**:509–23.

20. Gounaris C, Wiesemann W, Floudas C. The robust capacitated vehicle routing problem under demand uncertainty. *Operations Research* 2013;**61**:677–93.

21. Lee C, Lee K, Park S. Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *Journal of the Operational Research Society* 2004;**63**:1294–306.

22. Nemirovski A, Shapiro A. Convex approximations of chance constrained programs. *SIAM Journal on Optimization* 2006;**17**:969–96.

23. Beraldi P, Ruszczyński A. A branch and bound method for stochastic integer problems under probabilistic constraints. *Optimization Methods Software* 2002;**17**:359–82.

24. Beraldi P, Ruszczyński A. The probabilistic set-covering problem. *Operations Research* 2002;**50**:956–67.

25. Beraldi P, Ruszczyński A. Beam search heuristic to solve stochastic integer problems under probabilistic constraints. *European Journal of Operational Research* 2005;**167**:35–47.

26. Beraldi P, Bruni M. An exact approach for solving integer problem under probabilistic constraints with random technology matrix. *Annals of Operations Reserach* 2010;**177**:127–37.

27. Beraldi P, Bruni M, Violi A. Capital rationing problems under uncertainty and risk. *Computational Optimization and Applications* 2012;**51**:1375–96.

28. Bruni M, Beraldi P, Laganà D. The express heuristic for probabilistically constrained integer problems. *Journal of Heuristics* 2013;**19**:423–41.

29. Fortz B, Poss M. Easy distributions for combinatorial optimization problems with probabilistic constraints. *Operations Research Letters* 2010;**38**:545–9.

30. Klopfenstein O. Solving chance-constrained combinatorial problems to optimality. *Computational Optimization and Applications* 2010;**45**:607–38.

31. Calafiore G, El Ghaoui L. On distributionally robust chance-constrained linear programs. *Journal of Optimization Theory and Applications* 2006;**130**:1–22.

32. Gouveia L, Saldanha-da Gama F. On the capacitated concentrator location problem: a reformulation by discretization. *Computers & Operations Research* 2006;**33**:1242–58.

33. Correia I, Gouveia L, Saldanha-da Gama F. Discretized formulations for capacitated location problems with modular distribution costs. *European Journal of Operational Research* 2010;**204**:237–44.

34. Fischetti M, Polo C, Scantamburlo M. A local branching heuristic for mixed-integer programs with 2–level variables, with an application to a telecommunication network design problem. *Networks* 2004;**44**:61–72.

35. Belenguer J, Benavent E. A cutting plane algorithm for the capacitated arc routing problem. *Computers & Operations Research* 2003;**30**:705–28.

36. Fischetti M, Salazar J, Toth P. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research* 1997;**45**:378–94.

37. Shaw P. Using constraint programming and local search methods to solve vehicle routing problems. *Lecture Notes in Computer Science* 1998; **1520**:417–31.

38. Russell R. Hybrid heuristics for the vehicle routing problem with time windows. *Transportation Science* 1995;**29**:156–66.

39. Franceschi D, Fischetti M, Toth P. A new ILP-based refinement heuristic for vehicle routing problems. *Mathematical Programming, Ser B* 2006;**105**:471–99.

40. Pisinger D, Ropke S. A general heuristic for vehicle routing problems. *Computers & Operations Research* 2007;**34**:2403–35.

41. Hansen P, Mladenovic N, Gerad LCD. A tutorial on variable neighborhood search. Tech. Rep.; Les Cahiers du GERAD, HEC Montreal and GERAD; 2003.

42. Osman I. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research* 1993;**41**:421–51.

43. Taillard E, Badeau P, Gendreau M, Guertin F, Potvin JY. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science* 1997;**31**:170–86.

44. Savelsbergh M. The vehicle routing problem with time windows: minimizing route duration. *INFORMS Journal on Computing* 1992;**4**:146–54.

45. Potvin JY, Rousseau JM. An exchange heuristic for routing problems with time windows. *Journal of the Operational Research Society* 1995;**46**:1433–46.

46. Cordeau JF, Gendreau M, Laporte G. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* 1997;**30**:105–19.

47. Toth P, Vigo V. The granular tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing* 2003;**15**:333–46.

48. Savelsbergh M, Song JH. An optimization algorithm for the inventory routing problem with continuous moves. *Computers & Operations Research* 2008; **35**:2266–82.

49. Song JH, Furman K. A maritime inventory routing problem: Practical approach. *Computers & Operations Research* 2013;**40**:657–65.