

# Benders Decomposition for Production Routing Under Demand Uncertainty

Yossiri Adulyasak, Jean-François Cordeau, Raf Jans

HEC Montréal and CIRRELT, 3000 chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7, Canada,  
{yossiri.adulyasak, jean-francois.cordeau, raf.jans}@hec.ca

The production routing problem (PRP) is a generalization of the inventory routing problem and concerns the production and distribution of a single product from a production plant to multiple customers using capacitated vehicles in a discrete and finite time horizon. In this study, we consider the stochastic PRP with demand uncertainty in two-stage and multi-stage decision processes. The decisions in the first stage include production setups and customer visit schedules, while the production and delivery quantities are determined in the subsequent stages. We introduce formulations for the two problems which can be solved by a branch-and-cut algorithm. To handle a large number of scenarios, we propose a Benders decomposition approach which is enhanced through lower bound lifting inequalities, scenario group cuts and Pareto-optimal cuts. For the multi-stage problem, we also use a warm-start procedure that relies on the solution of the simpler two-stage problem. Finally, we exploit the reoptimization capabilities of Benders decomposition in a sample average approximation method for the two-stage problem and in a rollout algorithm for the multi-stage problem. Computational experiments show that instances of realistic size can be solved to optimality for both the two-stage and multi-stage problems, and that Benders decomposition provides significant speedups compared to a classical branch-and-cut algorithm.

*Key words:* Production routing; demand uncertainty; two-stage and multi-stage stochastic programs with recourse; Benders decomposition.

---

## 1. Introduction

Demand uncertainty is a major issue in supply chain management because some of the critical information required for decision making is often known only approximately in the form of forecasts. In these conditions, solving a deterministic model using point estimates can lead to wrong and costly decisions. One should thus explicitly take the uncertainty into account in the decision process. This paper introduces a novel approach to solve the production routing problem (PRP), a generalization of the inventory routing problem (IRP), under demand uncertainty. We consider the problem with

a single product in a discrete and finite time horizon, where the distribution network consists of a production plant and multiple customers. At the beginning of each period, the plant can perform a setup and make the product using a limited production capacity. The plant can then dispatch a given number of capacitated vehicles to deliver the product to the customers. Each customer must carry sufficient inventory to satisfy its demand in every period and backlogging is not allowed. If some demand is left unmet at the end of a period, a unit penalty cost has to be paid. This penalty can be viewed in general as an opportunity cost related to lost sales or as the cost of outsourcing the production and delivery of the product. The objective of the problem is to minimize production costs, which consist of fixed setup and unit costs, inventory holding costs at both the plant and the customers, cost of unmet demand, and routing costs for the dispatched vehicles. Note that the PRP reduces to the IRP if the production setup and quantity decisions are fixed. Many authors have addressed the deterministic PRP and IRP. Due to the complexity of these problems, however, most studies have used heuristics (see, e.g., Adulyasak et al. 2014b). We refer to Andersson et al. (2010), Coelho et al. (2014b) and Adulyasak et al. (2014c) for recent reviews of algorithms for the IRP and PRP.

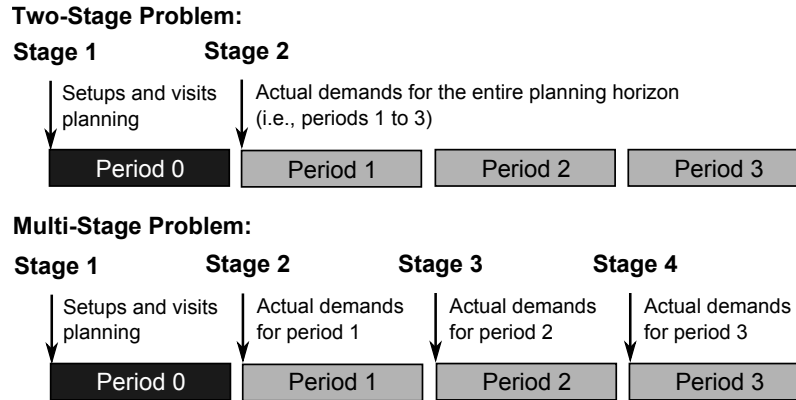
Taking demand uncertainty into account in the PRP or IRP makes the problem very hard to solve (Hvattum and Løkketangen 2009, Solyali et al. 2012). To the best of our knowledge, the PRP with demand uncertainty has not been addressed before. There are, however, a few studies that have introduced heuristics for the stochastic IRP (SIRP). Federgruen and Zipkin (1984) considered the SIRP with random demands but in a single-period planning horizon. The authors showed that their approach could provide 6-7% savings compared to the solution obtained by solving a deterministic vehicle routing problem (VRP). A different SIRP involving long term planning horizons was addressed by Jaillet et al. (2002). In this study, a repeated distribution pattern is used and the problem is solved in a rolling horizon framework by using approximations of the direct shipment delivery costs. The IRP with a discrete time infinite horizon was addressed in several studies. Kleywegt et al. (2002a) considered the SIRP with direct deliveries while Kleywegt et al. (2004) extended the problem to include multiple customers in the same route. Since the state space

is too large to compute, the authors employed approximate dynamic programming techniques. Adelman (2004) focused on the same problem and proposed a price-directed approach where the future costs of current actions are approximated using optimal dual prices. Hvattum et al. (2009) and Hvattum and Løkketangen (2009) used heuristics based on finite scenario trees to solve the same problem. For the IRP with a discrete finite planning horizon, Bertazzi et al. (2013) addressed the stochastic problem with the order-up-to level (OU) policy. They developed a heuristic rollout algorithm using an approximate cost-to-go and a branch-and-cut algorithm. Solyalı et al. (2012) addressed the single-vehicle IRP with demand uncertainty in a discrete and finite planning horizon, where the distribution of demand is unknown and backlogging is allowed. This problem is called the robust inventory routing problem (RIRP) and is solved using a branch-and-cut algorithm. Finally, Coelho et al. (2014a) considered a dynamic and stochastic variant of the IRP in which demands are gradually revealed over time. They proposed a heuristic and assessed the value of demand forecasts and transshipments between customers.

In this study, we consider the stochastic PRP (SPRP) under demand uncertainty in two-stage and multi-stage decision processes, where the distribution of the demand is assumed to be known. In the first stage of both problems, production setup and customer visit decisions, as well as the assignment of vehicles to customers, must be determined. This is in line with real-world practice, where some higher level decisions such as those associated to production planning are made in advance and these plans remain fixed in order to avoid large disruptions (Hopp and Spearman 2000). Planned visits must also be communicated in advance to the customers and to the drivers to prepare the required workforce, equipment and materials. Routing decisions made in the first stage consist of constructing a tour for each vehicle to visit the set of customers assigned to it, which can be done regardless of the demand realizations. The subsequent stages involve production, inventory and delivery quantity decisions which are made when the demand becomes known. The major difference between the two-stage and the multi-stage problems is the timing of the actual demand realizations. In the two-stage problem, the demands for the entire planning horizon become known once the first-stage decisions are made. In the multi-stage problem, the demands for a given stage

become known only after the decisions for the previous stage have been made. These two problems are illustrated in Figure 1. The two-stage SPRP and the multi-stage SPRP will be referred to as 2-SPRP and M-SPRP, respectively.

**Figure 1** Illustration of the two-stage and the multi-stage PRP.



To the best of our knowledge, neither the 2-SPRP nor the M-SPRP has been addressed before. Our study makes five main contributions. First, we introduce two-stage and multi-stage stochastic PRP formulations. Second, we compare two strategies to solve these problems: a classical branch-and-cut algorithm and a Benders decomposition approach. Third, we propose several computational enhancements for the Benders decomposition algorithm: the use of a single branching tree for the master problem, lower bound lifting inequalities, and scenario group cuts. Fourth, we develop an effective rollout heuristic to obtain good feasible solutions to the multi-stage problem. Fifth, we demonstrate the benefits of the reoptimization capabilities of the Benders decomposition in a sample average approximation method for the 2-SPRP and in the rollout heuristic for the multi-stage problem.

The rest of the paper is organized as follows. Section 2 introduces notation and mathematical formulations for the SPRP. Section 3 then describes the Benders decomposition algorithms. This is followed by the computational experiments in Sections 4 and 5, and by the conclusion.

## 2. Mathematical Formulations

This section first introduces the notation used throughout the paper. It then describes the two-stage and multi-stage formulations of the SPRP.

### 2.1. Notation

Let  $\Omega$ , indexed by  $\omega$ , denote the finite set of demand scenarios, and let  $\rho_\omega$  be the probability of scenario  $\omega \in \Omega$ . The two-stage SPRP can be defined on a complete undirected graph  $G = (N, E)$ , where  $N = \{0, \dots, n\}$  is the set of nodes and  $E = \{(i, j) : i, j \in N, i < j\}$  is the set of edges. Node 0 represents the plant while  $N_c = N \setminus \{0\}$  is the set of customers. Let  $E(S)$  be the set of edges  $(i, j) \in E$  such that  $i, j \in S$ , where  $S \subseteq N$  is a given set of nodes, and let  $\delta(S)$  be the set of edges incident to a node set  $S$ , i.e.,  $\delta(S) = \{(i, j) \in E : i \in S, j \notin S \text{ or } i \notin S, j \in S\}$ . For simplicity, we also write  $\delta(i)$  to represent the set  $\delta(\{i\})$  of edges incident to node  $i$ . We denote by  $T = \{1, \dots, l\}$  the set of time periods and by  $d_{it\omega}$  the demand of customer  $i$  in period  $t$  under scenario  $\omega$ . A unit cost  $\sigma_i$  is incurred if some demand of customer  $i$  is left unmet at the end of a period. At the beginning of the planning horizon, an initial inventory  $I_{i0}$  is available at node  $i$ . In each period, a production capacity of  $C$  is available and a unit production cost  $u$  and a fixed production setup cost  $f$  are incurred if production takes place. A set  $K = \{1, \dots, m\}$  of identical vehicles of capacity  $Q$  can be dispatched from the plant and a transportation cost  $c_{ij}$  applies when a vehicle travels between nodes  $i$  and  $j$ . Inventory can be kept at both the plant and customers but the inventory level cannot exceed  $L_i$  at node  $i$ , and a unit inventory holding cost  $h_i$  is incurred. Let also  $I_{i0\omega} = I_{i0}, \forall \omega \in \Omega$ ,  $M_{t\omega} = \min \left\{ C, \sum_{j=t}^l \sum_{i \in N_c} d_{ij\omega} \right\}$  and  $M'_{it\omega} = \min \left\{ L_i, Q, \sum_{j=t}^l d_{ij\omega} \right\}$ .

The following decision variables are used to formulate the SPRP. The variable  $y_t$  is equal to 1 iff production takes place in period  $t$  and  $z_{ikt}$  is equal to 1 iff node  $i$  is visited by vehicle  $k$  in period  $t$ . The variable  $x_{ijkt}$  represents the number of times vehicle  $k$  travels directly between node  $i$  and node  $j$  in period  $t$ . Finally, we denote by  $p_{t\omega}$  the production quantity in period  $t$  under scenario  $\omega$ ,  $I_{it\omega}$  the inventory at node  $i$  at the end of period  $t$  under scenario  $\omega$ ,  $q_{ikt\omega}$  the quantity delivered to

customer  $i$  with vehicle  $k$  in period  $t$  under scenario  $\omega$ , and  $e_{it\omega}$  the amount of unmet demand at customer  $i$  in period  $t$  associated with scenario  $\omega$ .

## 2.2. Two-Stage SPRP Formulation

We first present a two-stage SPRP formulation, which is an extension of the formulation for the deterministic problems studied by Archetti et al. (2007, 2011), Solyalı and Süral (2011) and Adulyasak et al. (2014a). The SPRP can be formulated as follows:

$$\min \sum_{t \in T} \left( f y_t + \sum_{(i,j) \in E} \sum_{k \in K} c_{ij} x_{ijkt} + \sum_{\omega \in \Omega} \rho_{\omega} \left( u p_{t\omega} + \sum_{i \in N} h_i I_{it\omega} + \sum_{i \in N_c} \sigma_i e_{it\omega} \right) \right) \quad (1)$$

$$\text{s.t.} \quad I_{0,t-1,\omega} + p_{t\omega} = \sum_{i \in N_c} \sum_{k \in K} q_{ikt\omega} + I_{0t\omega} \quad \forall t \in T, \forall \omega \in \Omega \quad (2)$$

$$I_{i,t-1,\omega} + \sum_{k \in K} q_{ikt\omega} + e_{it\omega} = d_{it\omega} + I_{it\omega} \quad \forall i \in N_c, \forall t \in T, \forall \omega \in \Omega \quad (3)$$

$$I_{0t\omega} \leq L_0 \quad \forall t \in T, \forall \omega \in \Omega \quad (4)$$

$$I_{it\omega} + d_{it\omega} \leq L_i \quad \forall i \in N_c, \forall t \in T, \forall \omega \in \Omega \quad (5)$$

$$p_{t\omega} \leq M_{t\omega} y_t \quad \forall t \in T, \forall \omega \in \Omega \quad (6)$$

$$\sum_{i \in N_c} q_{ikt\omega} \leq Q z_{0kt} \quad \forall k \in K, \forall t \in T, \forall \omega \in \Omega \quad (7)$$

$$q_{ikt\omega} \leq M'_{it\omega} z_{ikt} \quad \forall i \in N_c, \forall k \in K, \forall t \in T, \forall \omega \in \Omega \quad (8)$$

$$\sum_{k \in K} z_{ikt} \leq 1 \quad \forall i \in N_c, \forall t \in T \quad (9)$$

$$\sum_{(j,j') \in \delta(i)} x_{jj'kt} = 2 z_{ikt} \quad \forall i \in N, \forall k \in K, \forall t \in T \quad (10)$$

$$\sum_{(i,j) \in E(S)} x_{ijkt} \leq \sum_{i \in S} z_{ikt} - z_{ekt} \quad \forall S \subseteq N_c : |S| \geq 2, \forall e \in S, \forall k \in K, \forall t \in T \quad (11)$$

$$e_{it\omega}, p_{t\omega}, I_{it\omega}, q_{ikt\omega} \geq 0 \quad \forall i \in N, \forall k \in K, \forall t \in T, \forall \omega \in \Omega \quad (12)$$

$$y_t, z_{ikt} \in \{0, 1\} \quad \forall i \in N, \forall k \in K, \forall t \in T \quad (13)$$

$$x_{ijkt} \in \{0, 1\} \quad \forall (i, j) \in E : i \neq 0, \forall k \in K, \forall t \in T \quad (14)$$

$$x_{0jkt} \in \{0, 1, 2\} \quad \forall j \in N_c, \forall k \in K, \forall t \in T. \quad (15)$$

The objective function (1) minimizes the cost of the first stage decisions and the expected cost of the second stage decisions. Constraints (2) and (3) enforce the inventory flow balance for each scenario at the plant and customers. The maximum inventory level is imposed by constraints (4) and (5). Constraints (6) allow a positive production quantity only if a setup is made, and this quantity cannot exceed the minimum of the capacity and the total demand in the remaining periods. The delivery quantity in each vehicle cannot exceed the vehicle capacity (constraints (7)) and a positive delivery quantity is allowed only if the customer is visited (constraints (8)). Each customer cannot be visited more than once per period following constraints (9). Constraints (10) require the number of incident edges to be 2 if the node is visited and constraints (11) eliminate subtours for each vehicle. We remark that constraints (5) impose the inventory capacity at customers by assuming that the delivery is made prior to demand consumption. These constraints can also be written as  $I_{i,t-1,\omega} + \sum_{k \in K} q_{ikt\omega} + e_{it\omega} \leq L_i$ .

Note that this formulation becomes the vehicle-index formulation for the deterministic PRP in Adulyasak et al. (2014a) when the number of scenarios is equal to one and the amount of unmet demand is forced to be zero. To strengthen the routing part, the following inequalities can be added to the formulation (see Archetti et al. 2007, 2011):

$$z_{ikt} \leq z_{0kt} \quad \forall i \in N_c, \forall k \in K, \forall t \in T \quad (16)$$

$$x_{ijkt} \leq z_{ikt} \text{ and } x_{ijkt} \leq z_{jkt} \quad \forall (i, j) \in E(N_c), \forall k \in K, \forall t \in T. \quad (17)$$

Constraints (16) allow a vehicle to visit customers only if it is dispatched from the plant and constraints (17) allow an edge incident to a customer node only if that customer is visited. When addressing the multi-vehicle aspect, Adulyasak et al. (2014a) showed that the following valid vehicle symmetry breaking constraints can significantly improve algorithmic performance:

$$z_{0kt} \geq z_{0,k+1,t} \quad \forall 1 \leq k \leq m-1, \forall t \in T \quad (18)$$

$$\sum_{i=1}^j 2^{(j-i)} z_{ikt} \geq \sum_{i=1}^j 2^{(j-i)} z_{i,k+1,t} \quad \forall j \in N_c, \forall 1 \leq k \leq m-1, \forall t \in T. \quad (19)$$

Model (1)-(19) will be referred to as the 2-BF.

### 2.3. Multi-Stage SPRP Formulation

In the multi-stage problem, one has to make the decisions in stage  $t$  without knowing the demand of future periods. An independent set of possible realizations is thus considered for each time period and the full set of scenarios can be represented by a scenario tree. Denote by  $\Omega_t$ , indexed by  $\omega_t$ , the set of possible demand realizations at period  $t$ . Each  $l$ -period scenario  $\omega$  can be described by a path from the root node of the tree in period 1 to a leaf node in period  $l$ , i.e.,  $\omega = \{\omega_1, \dots, \omega_l\}$ . Unlike the 2-SPRP where scenarios are independent from each other, some  $l$ -period scenarios in the M-SPRP have common elements in their trajectories from the first period to a certain period  $t < l$ . This requires the introduction of so-called non-anticipativity constraints to ensure the consistency of production, inventory and delivery quantity decisions between scenarios. Denote by  $\mathcal{H}^t(\omega)$  the index of the scenario node in the scenario tree at period  $t$  associated with scenario  $\omega$ . Let also  $p'_{t, \mathcal{H}^t(\omega)}$ ,  $I'_{it, \mathcal{H}^t(\omega)}$ ,  $q'_{ikt, \mathcal{H}^t(\omega)}$  and  $e'_{it, \mathcal{H}^t(\omega)}$  denote the variables  $p$ ,  $I$ ,  $q$  and  $e$  associated with the scenario node  $\mathcal{H}^t(\omega)$ , respectively. The M-SPRP can be formulated by adding the following non-anticipativity constraints:

$$p_{t\omega} = p'_{t, \mathcal{H}^t(\omega)} \quad \forall t \in T, \forall \omega \in \Omega \quad (20)$$

$$I_{it\omega} = I'_{it, \mathcal{H}^t(\omega)} \quad \forall i \in N, \forall t \in T, \forall \omega \in \Omega \quad (21)$$

$$e_{it\omega} = e'_{it, \mathcal{H}^t(\omega)} \quad \forall i \in N_c, \forall t \in T, \forall \omega \in \Omega \quad (22)$$

$$q_{ikt\omega} = q'_{ikt, \mathcal{H}^t(\omega)} \quad \forall i \in N_c, \forall k \in K, \forall t \in T, \forall \omega \in \Omega. \quad (23)$$

Model (1)-(23) will be referred to as the M-BF.

### 3. Benders Decomposition

We now introduce exact algorithms based on Benders decomposition (Benders 1962) to solve the 2-SPRP and the M-SPRP. In Benders decomposition, the original problem is partitioned into a *master problem* and a number of *subproblems* which are typically easier to solve than the original problem. By using linear programming duality, all the variables that belong to the subproblems are projected out and the master problem contains the remaining variables and an artificial variable



representing a lower bound on the total cost of the subproblems. The resulting model is solved by a cutting plane algorithm in which, at each iteration, the values of the master problem variables are first determined and the subproblems are solved with these variables fixed. If the subproblems are feasible and bounded, an *optimality cut* is added to the master problem, otherwise a *feasibility cut* is added. An upper bound can be computed from feasible subproblems and a lower bound is obtained if the master problem is solved to optimality. The process continues until an optimal solution is found or the optimality gap is smaller than a given threshold.

### 3.1. Two-Stage SPRP Benders Reformulation

In the two-stage SPRP, we observe that the integer variables are the first-stage decisions while the continuous variables belong to the second stage. If the first-stage decisions are fixed, the resulting subproblem is a network flow problem which can be decomposed by scenario. This follows the original idea of applying Benders decomposition to stochastic integer programs, also known as the *L-shaped method* (see Van Slyke and Wets 1969, Birge and Louveaux 2011).

We let  $\bar{\mathbf{x}}$ ,  $\bar{\mathbf{y}}$  and  $\bar{\mathbf{z}}$  denote the vectors of fixed  $x_{ijkt}$ ,  $y_t$  and  $z_{ikt}$  variables, respectively. The second-stage decisions are independent of  $\bar{\mathbf{x}}$ , i.e., independent of the order of the visits in a route. The expected total cost of the second-stage decisions, denoted by  $v(\bar{\mathbf{y}}, \bar{\mathbf{z}})$ , can be calculated as  $v(\bar{\mathbf{y}}, \bar{\mathbf{z}}) = \sum_{\omega \in \Omega} \rho_{\omega} v_{\omega}(\bar{\mathbf{y}}, \bar{\mathbf{z}})$ , where  $v_{\omega}(\bar{\mathbf{y}}, \bar{\mathbf{z}})$  is the total second-stage cost of scenario  $\omega$ , which can itself be obtained by solving the following *primal flow subproblem (PFS)*:

$$v_{\omega}(\bar{\mathbf{y}}, \bar{\mathbf{z}}) = \min \sum_{t \in T} \left( up_{t\omega} + \sum_{i \in N} h_i I_{it\omega} + \sum_{i \in N_c} \sigma_i e_{it\omega} \right) \quad (24)$$

s.t. (12) and

$$I_{0,t-1,\omega} + p_{t\omega} = \sum_{i \in N_c} \sum_{k \in K} q_{ikt\omega} + I_{0t\omega} \quad \forall t \in T \quad (25)$$

$$I_{i,t-1,\omega} + \sum_{k \in K} q_{ikt\omega} + e_{it\omega} = I_{it\omega} + d_{it\omega} \quad \forall i \in N_c, \forall t \in T \quad (26)$$

$$I_{0t\omega} \leq L_0 \quad \forall t \in T \quad (27)$$

$$I_{it\omega} + d_{it\omega} \leq L_i \quad \forall i \in N_c, \forall t \in T \quad (28)$$

$$p_{t\omega} \leq M_{t\omega} \bar{y}_t \quad \forall t \in T \quad (29)$$

$$\sum_{i \in N_c} q_{ikt\omega} \leq Q \bar{z}_{0kt} \quad \forall k \in K, \forall t \in T \quad (30)$$

$$q_{ikt\omega} \leq M'_{it\omega} \bar{z}_{ikt} \quad \forall i \in N_c, \forall k \in K, \forall t \in T. \quad (31)$$

Due to the presence of the variables  $e_{it\omega}$ , the PFS is always feasible because the demand can be left unmet. Furthermore, since the cost parameters  $u$ ,  $h_i$  and  $\sigma_i$  are finite and due to constraints (25)-(28), any feasible solution of the PFS must be bounded. As a consequence, the dual of PFS is feasible and bounded. We let  $\alpha = (\alpha_{i\omega} | \forall t \in T, \forall \omega \in \Omega)$ ,  $\beta = (\beta_{it\omega} | \forall i \in N_c, \forall t \in T, \forall \omega \in \Omega)$ ,  $\gamma = (\gamma_{t\omega} \geq 0 | \forall t \in T, \forall \omega \in \Omega)$ ,  $\theta = (\theta_{it\omega} \geq 0 | \forall i \in N_c, \forall t \in T, \forall \omega \in \Omega)$ ,  $\delta = (\delta_{t\omega} \geq 0 | \forall t \in T, \forall \omega \in \Omega)$ ,  $\kappa = (\kappa_{kt\omega} \geq 0 | \forall k \in K, \forall t \in T, \forall \omega \in \Omega)$  and  $\zeta = (\zeta_{ikt\omega} \geq 0 | \forall i \in N_c, \forall k \in K, \forall t \in T, \forall \omega \in \Omega)$  denote the vectors of the dual variables associated with constraints (25)-(31), respectively, and let also  $\alpha_{l+1,\omega} = 0$  and  $\beta_{i,l+1,\omega} = 0$ . The dual of the primal subproblem for each scenario  $\omega$ , called the *dual flow subproblem (DFS)*, can be formulated as follows:

$$v_\omega(\bar{\mathbf{y}}, \bar{\mathbf{z}}) = \max \quad -I_{00}\alpha_{1\omega} + \sum_{i \in N_c} (d_{i1\omega} - I_{i0})\beta_{i1\omega} + \sum_{i \in N_c} \sum_{t=2}^l d_{it\omega} \beta_{it\omega} - \sum_{t \in T} L_0 \gamma_{t\omega} \\ - \sum_{t \in T} \sum_{i \in N_c} (L_i - d_{it\omega}) \theta_{it\omega} - \sum_{t \in T} M_{t\omega} \bar{y}_t \delta_{t\omega} - \sum_{t \in T} \sum_{k \in K} Q \bar{z}_{0kt} \kappa_{kt\omega} - \sum_{t \in T} \sum_{k \in K} \sum_{i \in N_c} M'_{it\omega} \bar{z}_{ikt} \zeta_{ikt\omega} \quad (32)$$

$$\text{s.t.} \quad (\alpha, \beta, \gamma, \theta, \delta, \kappa, \zeta) \in \Delta_\omega, \quad (33)$$

where  $\Delta_\omega$  denotes the polyhedron defined by the constraints of the problem.

We define the set  $\Delta = \bigcup_{\omega \in \Omega} \Delta_\omega$  and we let  $P_\Delta$  denote the set of extreme points of  $\Delta$ . To obtain the *Benders master problem*, we further define  $\pi_\omega(\alpha, \beta, \gamma, \theta) = -I_{00}\alpha_{1\omega} + \sum_{i \in N_c} (d_{i1\omega} - I_{i0})\beta_{i1\omega} + \sum_{i \in N_c} \sum_{t=2}^l d_{it\omega} \beta_{it\omega} - \sum_{t \in T} L_0 \gamma_{t\omega} - \sum_{t \in T} \sum_{i \in N_c} (L_i - d_{it\omega}) \theta_{it\omega}$  and we introduce an extra variable  $\eta$  representing the expected total flow cost. The model 2-BF can be reformulated as follows:

$$\min \sum_{t \in T} \left( f y_t + \sum_{(i,j) \in E} \sum_{k \in K} c_{ij} x_{ijkt} \right) + \eta \quad (34)$$

s.t. (9)-(11), (13)-(15), (16)-(19) and

$$\sum_{\omega \in \Omega} \rho_\omega \left( - \sum_{t \in T} M_{t\omega} \delta_{t\omega} y_t - \sum_{t \in T} \sum_{k \in K} Q \kappa_{kt\omega} z_{0kt} - \sum_{t \in T} \sum_{k \in K} \sum_{i \in N_c} M'_{it\omega} \zeta_{ikt\omega} z_{ikt} + \pi_\omega(\alpha, \beta, \gamma, \theta) \right) \leq \eta \\ \forall (\alpha, \beta, \gamma, \theta, \delta, \kappa, \zeta) \in P_\Delta. \quad (35)$$

This formulation will be referred to as 2-BRF.

Observe that 2-BRF contains a large number of Benders cuts (35) as well as the SECs (11). Thus, a natural solution approach is to start from a relaxed 2-BRF where these constraints are dropped. Next, violated constraints are detected and iteratively added to the problem. A solution approach to handle this reformulation will be explained in Section 3.3.

### 3.2. Multi-Stage SPRP Benders Reformulation

As explained in Section 2.3, the M-SPRP is basically the 2-SPRP with the additional non-anticipativity constraints. Therefore, one obtains a similar reformulation as for the 2-BRF by projecting out the continuous variables. We first consider the original model M-BF (1)-(23). Denote by  $\lambda_{t\omega}^p, \lambda_{it\omega}^I, \lambda_{it\omega}^e$  and  $\lambda_{ikt\omega}^q$  the dual variables associated with constraints (20)-(23), respectively. One obtains the following Benders reformulation:

$$\min \sum_{t \in T} \left( f y_t + \sum_{(i,j) \in E} \sum_{k \in K} c_{ij} x_{ijkt} \right) + \eta \quad (36)$$

s.t. (9)-(11), (13)-(15), (16)-(19) and

$$\sum_{\omega \in \Omega} \rho_{\omega} \left( - \sum_{t \in T} M_{t\omega} \delta_{t\omega} y_t - \sum_{t \in T} \sum_{k \in K} Q \kappa_{kt\omega} z_{0kt} - \sum_{t \in T} \sum_{k \in K} \sum_{i \in N_c} M'_{it\omega} \zeta_{ikt\omega} z_{ikt} \right) + \pi'(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta}, \boldsymbol{\lambda}^p, \boldsymbol{\lambda}^I, \boldsymbol{\lambda}^e, \boldsymbol{\lambda}^q) \leq \eta \quad \forall (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta}, \boldsymbol{\delta}, \boldsymbol{\kappa}, \boldsymbol{\zeta}, \boldsymbol{\lambda}^p, \boldsymbol{\lambda}^I, \boldsymbol{\lambda}^e, \boldsymbol{\lambda}^q) \in Q_{\Delta}, \quad (37)$$

where  $Q_{\Delta}$  denotes the set of extreme points of the polyhedron defined by the constraints of the resulting multi-period subproblem, and the function  $\pi'(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta}, \boldsymbol{\lambda}^p, \boldsymbol{\lambda}^I, \boldsymbol{\lambda}^e, \boldsymbol{\lambda}^q)$  is the part of the dual subproblem objective function that does not depend on the first stage variables  $(x, y, z)$ . This expression is equivalent to the function  $\pi_{\omega}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta})$  defined in Section 3.1 given that the right-hand-side of constraints (20)-(23) is equal to 0. This formulation will be referred to as M-BRF. Note that the subproblem is no longer separable by scenario because of the non-anticipativity constraints.

### 3.3. Benders-Based Branch-and-Cut Algorithm

Because Benders cuts can be generated from any master problem solution and not just from an optimal integer solution (McDaniel and Devine 1977), one can solve the Benders reformulation

in a standard branch-and-cut framework where the subproblems are solved and Benders cuts are generated at any node of the branch-and-bound tree for the master problem. This approach is sometimes called Benders-based branch-and-cut (BBC) (see, e.g., Naoum-Sawaya and Elhedhli 2013). Other implementations of BBC were discussed by Codato and Fischetti (2006), Fortz and Poss (2009) and de Camargo et al. (2011), among others. In our algorithm, to avoid generating a large number of Benders cuts, these cuts are added to the master problem only at the root node and when an incumbent solution of the Benders master problem is found. Our tests showed that this strategy significantly improves performance compared to adding cuts at every node of the branch-and-bound tree.

Since our master problem also contains subtour elimination constraints (SECs), a separation procedure is applied to detect violated SECs at each node of the branch-and-bound tree and these cuts are added to the problem together with the Benders cuts, if any. We use the minimum  $s - t$  algorithm of the Concorde callable library (Applegate et al. 2011) as the separation algorithm.

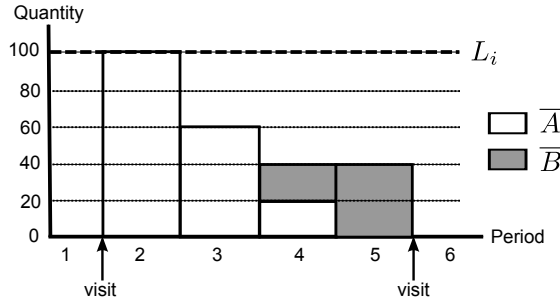
### 3.4. Computational Enhancements

We now describe computational enhancements that help improve the convergence of the algorithm.

**3.4.1. Lower Bound Lifting Inequalities.** Because parts of the objective function (1) are projected out in the Benders reformulations, the optimality gap may be large in the initial stages of the algorithm due to the low quality of the lower bound. A large number of Benders cuts are thus needed to close the gap. To address this issue, one can lift the lower bound of the Benders master problem by using initial cuts, called lower bound lifting inequalities (LBL), that contain some information about the parts of the original objective function that were removed. In particular, one can add cuts to represent a lower bound on the flow costs, i.e., unit production, inventory and penalty costs. We observe that, for the periods between two consecutive deliveries to a customer, the minimum flow cost can be calculated by considering the quantity that must be supplied between the two visits to satisfy the demand. Figure 2 illustrates the inventory level when customer  $i$  is visited in period 2 and then in period 6, while the demand in each period is equal

to 40 so that the minimum total quantity required to satisfy the demand between these periods is equal to 160. Given the maximum inventory capacity  $L_i = 100$ , this total demand quantity can be separated into two parts, i.e., the amount  $\bar{A} = 100$  under  $L_i$  which can be supplied or can also be left unmet if this results in a lower cost, and the amount  $\bar{B} = 60$  which cannot be supplied and has to be left unmet due to the inventory capacity limit.

**Figure 2** Inventory level corresponding to the two consecutive visits in periods 2 and 6.



We define the periods 0 and  $l+1$  as dummy periods at the beginning and the end of the planning horizon (used for calculation purposes) and  $d_{i0\omega} = d_{i,l+1,\omega} = 0, \forall i \in N_c, \forall \omega \in \Omega$ . Let  $\lambda_{ivt}$  be a binary variable equal to one if customer  $i$  is visited in period  $v < t$  and the next visit is in period  $t$ , and the parameter  $\phi_{ivt}$  be the minimum possible sum of unit production, inventory and penalty costs over the period  $v$  to  $t-1$  associated with the variable  $\lambda_{ivt}$ , calculated as

$$\phi_{ivt} = \sum_{\omega \in \Omega} \rho_{\omega} \left( \sum_{s=v}^{t-1} \min \{ H_{ivs\omega}^P, H_{ivs\omega}^{\sigma} \} + c_{ivt}^h + \sigma_i \left( \sum_{s=v}^{t-1} d_{is\omega} - \phi_{iv} \right)^+ \right)$$

$$\text{where } H_{ivs\omega}^P = \begin{cases} (h_i(s-1) + u) \min \left\{ d_{is\omega}, \left( \phi_{i0} - \sum_{w=1}^{s-1} d_{iww} \right)^+ \right\} & \text{if } v = 0 \\ (h_i(s-v) + u) \min \left\{ d_{is\omega}, \left( \phi_{iv} - \sum_{w=v}^{s-1} d_{iww} \right)^+ \right\} & \text{if } v > 0, \end{cases}$$

$$H_{ivs\omega}^{\sigma} = \sigma_i \min \left\{ d_{is\omega}, \left( \phi_{iv} - \sum_{w=v}^{s-1} d_{iww} \right)^+ \right\},$$

$$\phi_{iv} = \begin{cases} I_{i0} & \text{if } v = 0 \\ L_i & \text{if } 0 < v < t \leq l+1 \end{cases} \quad \text{and } c_{ivt}^h = \begin{cases} h_i(t-1) \left( I_{i0} - \sum_{w=1}^{t-1} d_{iww} \right)^+ & \text{if } v = 0 \\ h_i(t-v) \left( I_{i0} - \sum_{w=1}^{t-1} d_{iww} \right)^+ & \text{if } 0 < v < t \leq l+1. \end{cases}$$

Note that  $c_{ivt}^h$  is the inventory cost at customer  $i$  incurred over the period  $v$  to  $t-1$  for the part

of the initial inventory that is not used up at the end of period  $t - 1$ . The following cuts can be added to the master problem:

$$\sum_{t=1}^{l+1} \sum_{v=0}^{t-1} \sum_{i \in N_c} \phi_{ivt} \lambda_{ivt} - u \sum_{i \in N_c} I_{i0} \leq \eta \quad (38)$$

$$\sum_{v=0}^{t-1} \lambda_{ivt} = \sum_{k \in K} z_{ikt} \quad \forall i \in N_c, \forall t \in T \quad (39)$$

$$\sum_{v=0}^{t-1} \sum_{s=t}^{l+1} \lambda_{ivs} = 1 \quad \forall i \in N_c, \forall t \in T \cup \{l+1\} \quad (40)$$

$$\lambda_{ivt} \in \{0, 1\} \quad \forall i \in N_c, \forall 0 \leq v < t \leq l+1. \quad (41)$$

Constraint (38) provides a lower bound for the flow cost. Constraints (39) link the  $\lambda_{ivt}$  and  $z_{ikt}$  variables and constraints (40) enforce that one replenishment plan  $\lambda_{ivt}$  be selected in each period. One can observe that when all the  $z_{ikt}$  variables are fixed, the variables  $\lambda_{ivt}$  can be easily set by inspection.

**3.4.2. Scenario Group Cuts.** In the 2-BRF, the Benders subproblem decomposes into many independent subproblems. Hence, many Benders cuts, one for each subproblem, can be added at once to accelerate convergence (Birge and Louveaux 1988). However, adding too many cuts at each iteration can lead to a worse performance because of the time taken to solve the master problem (de Camargo et al. 2008). In the SPRP, there is a subproblem for each scenario and the number of Benders cuts generated at each iteration can be huge. To overcome this issue, one can instead create groups of scenarios and aggregate the Benders cuts in each group to reduce the size of the master problem. To create scenario groups, we first define the number of scenario groups  $n_G \leq |\Omega|$  and groups of scenarios  $G(g)$ , indexed by  $g$ . The range between the maximum and minimum total demand among all scenarios is calculated and separated equally into  $n_G$  groups. Each scenario is then assigned to a group according to its total demand. Denote by  $\eta_g$  the expected total flow cost corresponding to group  $g$ . The variable  $\eta$  in the objective function (34) is replaced by  $\sum_{g=1}^{n_G} \eta_g$  and constraints (35) are replaced by the following constraints:

$$\sum_{\omega \in G(g)} \rho_\omega \left( - \sum_{t \in T} M_{t\omega} \delta_{t\omega} y_t - \sum_{t \in T} \sum_{k \in K} Q_{k\omega} z_{0kt} - \sum_{t \in T} \sum_{k \in K} \sum_{i \in N_c} M'_{it\omega} \zeta_{ikt\omega} z_{ikt} + \pi_\omega(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta}) \right) \leq \eta_g \quad \forall 1 \leq g \leq n_G, \forall (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta}, \boldsymbol{\delta}, \boldsymbol{\zeta}, \boldsymbol{\kappa}) \in P_\Delta. \quad (42)$$

**3.4.3. Pareto-Optimal Cuts.** The performance of a Benders decomposition algorithm often depends on the quality of the cuts being generated. This is especially true when the dual subproblem admits multiple optimal solutions, each providing a potentially different cut. To identify strong cuts, we employ the approach of Magnanti and Wong (1981) which ensures the generation of a Pareto-optimal cut whenever there are multiple optimal dual solutions. More details on our implementation of this approach are provided in the Online Supplement.

**3.4.4. Warm Start for the M-BRF.** Since the 2-SPRP is a relaxation of the M-SPRP, one can warm-start the algorithm for the M-BRF by first solving the 2-BRF to generate an initial set of cuts. These cuts are valid for the M-BRF because the primal subproblem for the 2-BRF is a relaxation for that of the M-BRF obtained by omitting the non-anticipativity constraints. Hence, the dual subproblem for the 2-BRF is a restriction of the dual subproblem for the M-BRF (since the former contains only a subset of the variables of the latter). As a result, a dual solution identified by solving the 2-BRF subproblem is guaranteed to be feasible for the M-BRF as well. This solution may not correspond to an extreme point of the polyhedron but it will nonetheless provide a valid cut. The idea of solving a relaxed primal subproblem to warm-start a Benders decomposition algorithm was used before, e.g., by Cordeau et al. (2001). Following this observation, one can solve the 2-BRF to generate cuts using the same set of scenarios as for the M-SPRP but without the non-anticipativity constraints. Moreover, it is known that the optimal solution of the 2-BRF also provides a valid lower bound for the M-SPRP (Birge and Louveaux 2011). Finally, since the LBL cuts are valid for 2-BRF, one can add these cuts to the M-BRF before the algorithm starts.

## 4. Computational Experiments

We have performed experiments using the PRP instances introduced by Adulyasak et al. (2014a), which were themselves generated from the instances of Archetti et al. (2011). The test set consists of instances with  $n = 5, 10, 15, 20, 25$  and 30 customers while the number of periods  $l$  is between 3 and 6. For simplicity, we designate by  $\mathcal{P}$  the set of instances with  $n \leq 20$  for the 2-SPRP and with

$n \leq 15$  for the M-SPRP. We have generated scenarios by a Monte-Carlo simulation in which the demand in each period is independent and varies in the range  $[\bar{d}_{it}(1 - \epsilon), \bar{d}_{it}(1 + \epsilon)]$ , where  $\bar{d}_{it}$  is the demand of the nominal case from the original test set. We assume that the demands are uniformly distributed. For the M-SPRP, the scenarios for each period  $\Omega^t$  are generated separately and the set  $\Omega$  contains all the trajectory paths of the scenarios at the leaf nodes in the scenario tree. The penalty cost is set to  $\sigma_i = \hat{\alpha} [u + f/C + 2c_{0i}/Q]$ , where  $\hat{\alpha}$  is a predefined penalty level. Unless stated otherwise, we use  $\hat{\alpha} = 5$  for the default setting when solving the instances. In all tables, columns *#Opt*, *Gap*, *CPU* and *B.Cuts* show the number of instances solved to optimality, the average optimality gap (%), the average CPU time in seconds and the average number of generated Benders cuts, respectively. The experiments were performed on a workstation with an Intel Xeon 2.67GHz processor and 6GB of RAM under Scientific Linux 6.1 using CPLEX 12.5.1. The algorithms were coded in C and C# on MonoDevelop 3.0. The maximum CPU time per instance is set to two hours unless otherwise indicated. To prevent memory problems, we also set the maximum number of branch-and-bound nodes to 200,000. In all experiments, branching priority was given first to the  $y$  variables and then to  $z$  and  $x$  variables.

#### 4.1. Impact of the Computational Enhancements

Tables 1 and 2 report the performance of the BBC algorithms when the enhancements of Section 3.4 are applied for the 2-BRF and M-BRF, respectively. The tests were performed on the instances of set  $\mathcal{P}$  with three periods, one vehicle and  $\epsilon = 0.2$ . For each value of  $n$ , we solved four different instances with different cost parameters and the maximum CPU time per instance is set to one hour. For the 2-BRF, although we have tested several different choices for the number of groups in the scenario group cuts, we report results only for  $b = 5$  because a change of  $b$  in a small range (e.g., less than 10) has little impact on performance. Columns *None* indicate the BBC without any enhancement and other columns show the results when different computational enhancements are used. The abbreviations LBL, PO and SG5 correspond to the lower bound lifting inequalities,



**Table 1** Average results of the BBC algorithms using the computational enhancements for the 2-BRF

T	$\Omega$	#Ins	None				LBL				LBL\PO				LBL\PO\SG5				
			#Opt	Gap	CPU	B.Cuts	#Opt	Gap	CPU	B.Cuts	#Opt	Gap	CPU	B.Cuts	#Opt	Gap	CPU	B.Cuts	
3	100	16	13	0.5	714.3	2683	15	0.0	323.6	2157	16	0.0	115.6	39	16	0.0	50.9	314	
3	500	16	13	0.8	530.9	3713	13	0.1	523.9	3266	16	0.0	145.5	38	15	0.0	153.6	361	
3	1000	16	12	0.7	961.5	1931	13	0.1	697.1	1897	15	0.0	323.7	37	16	0.0	207.8	424	
Total			48	38	0.6	735.5	2776	41	0.0	514.9	2440	47	0.0	194.9	38	47	0.0	137.4	366

Pareto-optimal cuts, and scenario group cuts with 5 groups, respectively. We use WS to indicate the warm-start process using cuts generated from the 2-BRF.

The results indicate that the LBL significantly reduce the average CPU time. The generation of Pareto-optimal cuts results in an increase in the computing time per iteration but this is largely compensated by a reduction in the number of Benders cuts being generated. Finally, choosing an appropriate number of groups can lead to better performance and combining the scenario group cuts and Pareto-optimal cuts provides the best overall results compared to the other options. Thus, we chose the combination of LBL, PO and SG5 as the preferred setting of the BBC in all remaining experiments for the 2-SPRP.

**Table 2** Average results of the BBC algorithms using the computational enhancements for the M-BRF

T	$\Omega_t$	$\Omega$	#Ins	None				LBL				LBL\WS			
				#Opt	Gap	CPU	B.Cuts	#Opt	Gap	CPU	B.Cuts	#Opt	Gap	CPU	B.Cuts
3	6	216	12	7	20.6	1214.7	1847	9	0.8	645.8	1015	9	0.6	767.1	1042
3	8	512	12	5	32.3	2754.4	1794	9	0.9	1420.6	858	10	0.7	1254.1	717
Total			24	12	26.5	1984.6	1820	18	0.9	1033.2	936	19	0.6	1010.6	880

For the M-BRF, the results indicate that the LBL can significantly improve the computational performance and applying the LBL together with the warm start provides additional improvements. The average optimality gap decreased from 26.5% to 0.6% while the average CPU was reduced by approximately 50% with the LBL\WS. We thus used the LBL\WS as the default setting for the M-BRF.

## 4.2. Comparisons with Branch-and-Cut

In this section, we compare the results of the BBC with a branch-and-cut algorithm (BC). To this end, we adapted the branch-and-cut algorithm for the vehicle-index formulation of Adulyasak et al. (2014a), which provided the best results for the deterministic problem. This BC algorithm is applied to solve the formulations 2-BF and M-BF directly. In the BC, the SECs (11) are relaxed and generated by calling the separation procedure. The results are shown in Tables 3 and 4. Columns *Nodes* and *N.Cplex* show the number of branching nodes and the number of CPLEX cuts generated in the branch-and-bound tree, respectively. To better illustrate the differences between the two approaches, we performed the tests on two instance sets: one with 3 periods and 5 to 30 customers, and the other with 6 time periods and 5 to 20 customers, while setting the number of scenarios  $|\Omega|$  equal to 100, 500 or 1000 for the 2-SPRP. For the M-SPRP, we used one instance set with 3 periods and 5 to 20 customers, and another set with 4 periods and 5 to 15 customers. In both cases, we set  $|\Omega^t| = 5, 6$  and  $7$ , which results in  $|\Omega| = 125, 216$  and  $343$  and  $|\Omega| = 625, 1296$  and  $2401$  for the instances with  $|T|=3$  and  $|T|=4$ , respectively. The number of vehicles in each instance varies between 1 and 3 while the sum of vehicle capacities remains constant. Finally, we set  $\epsilon = 0.2$ . Note that the average optimality gap was computed only on the instances where a feasible solution was found. In addition to these results, we also report the expected value of perfect information (EVPI) and the value of the stochastic solution (VSS) as a percentage of the best solution cost. Note that when an instance is not solved to optimality, the calculated values provide a lower bound on the VSS and an upper bound on the EVPI.

For the 2-SPRP, when the number of scenarios is not large (i.e.,  $|\Omega|=100$ ) the BC algorithm is superior to the BBC. However, the performance of the BC algorithm deteriorates sharply when the number of scenarios, the number of periods and the number of vehicles increase. The BBC algorithm, on the other hand, is less sensitive to these parameters and still provides acceptable gaps for the most difficult instances tested. For the M-SPRP, the BC algorithm is superior to the BBC on the instances with 3 periods but the performance deteriorates significantly on instances with

**Table 3** Average results of the BC and the BBC algorithms on 2-SPRP instances

T	K	Ω	#Ins	BC						BBC						%EVPI	%VSS
				#Opt	Gap	CPU	Nodes	N.Cplex	#Opt	Gap	CPU	Nodes	N.Cplex	B.Cuts			
3	1	100	6	6	0.0	29.6	8	8109	4 <sup>[1]</sup>	0.7	1577.4	71637	25	1932	0.5%	1.0%	
3	2	100	6	6	0.0	171.4	31	11388	4 <sup>[1]</sup>	0.8	2475.3	86963	86	1522	0.6%	0.7%	
3	3	100	6	6	0.0	2029.2	270	12615	3	1.2	3734.7	83591	89	2287	0.5% <sup>(1)</sup>	1.2%	
3	1	500	6	6	0.0	427.0	12	37274	4 <sup>[1]</sup>	0.6	1882.1	61211	27	1753	0.5%	1.1%	
3	2	500	6	5	16.7	2606.5	28	48654	3 <sup>[2]</sup>	0.8	2815.1	83476	92	1540	0.7%	0.7%	
3	3	500	6	2	50.7	5539.7	107	34441	2 <sup>[1]</sup>	1.8	4699.9	78370	70	2448	0.6% <sup>(2)</sup>	1.1%	
3	1	1000	6	6	0.0	1837.7	10	72148	4 <sup>[1]</sup>	0.6	1975.1	70452	27	1373	0.5%	1.2%	
3	2	1000	6	3	33.5	5270.1	23	46625	3 <sup>[2]</sup>	1.0	2884.0	88063	77	1362	0.6%	0.9%	
3	3	1000	6	1	60.2	6948.6	31	27001	2	2.0	4877.8	72765	75	2088	0.6% <sup>(2)</sup>	1.0%	
Total			54	41	17.9	2762.2	58	33139	29 <sup>[9]</sup>	1.1	2991.3	77392	63	1812	0.6% <sup>(5)</sup>	1.0%	
6	1	100	4	4	0.0	162.7	92	8475	2 <sup>[1]</sup>	0.7	3675.2	95218	14	4600	0.6%	0.6%	
6	2	100	4	4	0.0	2301.2	1611	11564	1	2.1	5454.9	79983	83	7938	0.8%	1.3%	
6	3	100	4	2	14.0	4580.6	265	15965	1	3.4	7102.5	72114	67	9508	0.4% <sup>(3)</sup>	2.4%	
6	1	500	4	4	0.0	3830.9	91	45367	2	0.8	3881.7	72483	21	4028	0.6%	1.0%	
6	2	500	4	0	68.3	7200.0	34	28942	1	2.3	5607.7	46994	103	7145	0.9%	1.1%	
6	3	500	4	0	78.6	7200.0	20	15841	0	4.1	7200.0	42926	64	7720	n/a <sup>(4)</sup>	0.6%	
6	1	1000	4	1	45.2	6840.5	31	62624	2	1.0	4158.5	40382	17	3865	0.7%	0.4%	
6	2	1000	4	0	94.4	7200.0	0	19720	1	2.5	5846.8	24817	87	6490	0.1% <sup>(3)</sup>	1.1%	
6	3	1000	4	0	100.0	7200.0	0	9771	0	4.5	7200.0	21100	72	5950	n/a <sup>(4)</sup>	0.9%	
Total			36	15	44.5	5168.4	238	24252	10 <sup>[1]</sup>	2.4	5569.7	55113	58	6360	0.5% <sup>(14)</sup>	1.1%	

<sup>[a]</sup>Number of instances for which the maximum number of brand-and-bound nodes was reached.

<sup>(a)</sup>Number of instances for which the EVPI could not be computed in 4 hours of CPU time.

4 periods since the number of scenarios in the M-SPRP increases as the number of time periods increases and BC is very sensitive to this change. It should also be noted that a large number of CPLEX cuts is generated when solving the BC and the majority of them, approximately 90%, are *flow cover* cuts that strengthen the network structure of the problem. The performance of the algorithms without CPLEX cuts was reported in Adulyasak (2012). Without these cuts, the performance of the BC algorithm decreases significantly.

### 4.3. Sensitivity Analyses

In this section, we evaluate the solution quality and the performance of the algorithm when the parameters and settings are changed. In the first part of the analysis, we use the instances with different cost parameters as in the setting described by Archetti et al. (2011) and Adulyasak et al. (2014a). The details of the instance groups are given in Table 5.

**Table 4** Average results of the BC and the BBC algorithms on M-SPRP instances

T	K	$\Omega_t$	$\Omega$	#Ins	BC					BBC					%EVPI	%VSS	
					#Opt	Gap	CPU	Nodes	N.Cplex	#Opt	Gap	CPU	Nodes	N.Cplex			B.Cuts
3	1	6	216	4	4	0.0	32.4	23	4977	3	1.7	2088.9	24758	25	1123	1.6%	1.9%
3	1	7	343	4	4	0.0	60.7	28	7597	3	1.7	2419.4	22849	18	1301	1.8%	1.7%
3	1	8	512	4	4	0.0	109.1	20	11075	2	3.4	4337.7	24397	24	1839	1.8%	1.7%
3	2	6	216	4	4	0.0	158.7	70	8219	1	3.4	5471.7	65247	52	2965	1.6%	1.1%
3	2	7	343	4	4	0.0	313.5	83	11490	1	4.5	5509.5	27246	42	2211	1.8%	1.1%
3	2	8	512	4	4	0.0	615.0	74	20022	1	4.3	5642.7	35267	43	2532	1.8%	1.1%
3	3	6	216	4	4	0.0	1741.1	515	10440	2	4.2	4301.9	62287	60	2739	1.6%	2.1%
3	3	7	343	4	3	0.3	3560.6	345	15357	1	4.9	5976.9	63326	62	3033	1.9%	2.1%
3	3	8	512	4	2	1.3	4060.4	201	22196	1	5.1	6210.4	58347	49	2628	2.1%	2.0%
Total				36	33	0.2	1183.5	151	12375	15	3.7	4662.1	42636	42	2263	1.8%	1.7%
4	1	6	1296	3	3	0.0	773.9	48	26321	1	5.9	5800.0	7834	16	993	1.4%	0.7%
4	1	7	2401	3	3	0.0	2571.0	70	50489	1	6.0	5962.5	4523	17	594	1.1%	1.1%
4	1	8	4096	3	2	33.3	5874.6	52	82980	1	6.0	7200.0	2821	19	373	1.1%	0.9%
4	2	6	1296	3	2	23.4	4280.4	155	41392	0	8.1	7200.0	15494	46	1489	1.5%	0.5%
4	2	7	2401	3	0	34.4	7200.0	104	67994	0	8.4	7200.0	6362	31	754	1.6%	0.8%
4	2	8	4096	3	0	61.9	7200.0	1	63486	0	12.2	7200.0	3644	38	372	0.6% <sup>(1)</sup>	1.2%
4	3	6	1296	3	0	35.3	7200.0	151	41985	0	10.2	7200.0	13982	28	1216	0.9% <sup>(1)</sup>	1.2%
4	3	7	2401	3	0	83.3	6249.5 <sup>[1]</sup>	5	35585	0	14.6	7200.0	7935	44	612	0.8% <sup>(2)</sup>	1.3%
4	3	8	4096	3	0	87.5	6910.9 <sup>[1]</sup>	0	26560	0	15.4	7200.0	2816	48	281	0.6% <sup>(2)</sup>	1.0%
Total				27	10	39.9	5362.1 <sup>[2]</sup>	65	48533	3	9.7	6906.9	7268	32	743	1.1% <sup>(6)</sup>	1.0%

<sup>[a]</sup>Number of instances for which CPLEX was terminated due to the memory limit.

<sup>(a)</sup>Number of instances for which the EVPI could not be computed in 4 hours of CPU time.

**Table 5** Descriptions of the four instance groups

No.	Descriptions
G1	Standard instance (as used in the previous experiments)
G2	High production unit cost, $u^{(G1)} \times 10$
G3	Large transportation costs, coordinates <sup>(G1)</sup> $\times 5$
G4	No customer inventory costs

We solved the instances with the BBC algorithm and chose the instance set  $\mathcal{P}$  with  $|T|=3$  and  $m=1$  for both the 2-SPRP and M-SPRP. However, since not all the instances of size  $n=15$  for the M-SPRP were solved to optimality, we only reported the results of the instances with  $n=5, 10$  for this problem. The results are shown in Tables 6 and 7. The number of scenarios, penalty factors and the demand variation levels are shown in the tables.

For the 2-SPRP, one can see that the instance groups G3 and G4 are more difficult to solve than the standard instances and the stochastic solutions for these instances appear to provide small

**Table 6** Average results on different 2-SPRP instance groups

$\Omega$	$\hat{\alpha}$	$\epsilon$	#Ins	Solution cost				CPU				B.Cuts				%VSS			
				G1	G2	G3	G4	G1	G2	G3	G4	G1	G2	G3	G4	G1	G2	G3	G4
100	5	0.2	4	16979	121682	23777	13539	196.6	68.7	151.8	915.5	254	144	158	10177	1.0%	2.8%	0.2%	0.5%
500	5	0.2	4	16973	121658	23875	13371	235.1	38.2	921.3	926.7	252	150	4230	10219	1.1%	2.8%	0.6%	1.1%
1000	5	0.2	4	16961	121504	23762	13530	326.5	128.0	936.2	937.3	318	136	4255	3017	1.2%	3.0%	0.6%	0.4%
Total			12	16971	121615	23805	13480	252.8	78.3	669.8	926.5	274	143	2881	7804	1.1%	2.9%	0.5%	0.7%
100	10	0.2	4	17213	123677	24362	14051	203.2	24.0	208.2	217.3	243	135	216	308	2.5%	5.4%	0.1%	0.6%
500	10	0.2	4	17198	123606	24360	13700	185.2	55.6	545.5	945.9	160	120	2306	10412	2.6%	5.4%	1.8%	1.6%
1000	10	0.2	4	17220	123568	24064	13676	203.9	112.2	548.6	943.8	194	158	2308	10246	2.5%	5.4%	1.7%	2.8%
Total			12	17211	123617	24262	13809	197.4	63.9	434.1	702.3	199	137	1610	6989	2.5%	5.4%	1.2%	1.7%
100	5	0.4	4	17035	121933	24134	13841	182.9	111.8	281.1	394.6	221	165	231	1342	3.1%	5.6%	0.6%	0.5%
500	5	0.4	4	17061	122167	24148	13931	197.6	132.7	257.3	786.3	236	145	206	6914	3.1%	5.7%	1.1%	0.3%
1000	5	0.4	4	17033	121826	24097	13888	259.7	131.7	222.7	1331.9	290	167	179	8555	3.2%	6.0%	0.2%	0.4%
Total			12	17043	121975	24126	13887	213.4	125.4	253.7	837.6	249	159	205	5604	3.1%	5.8%	0.6%	0.4%
100	10	0.4	4	17277	123770	25346	14372	171.7	125.8	189.8	169.7	220	150	243	311	6.7%	11.2%	0.6%	3.5%
500	10	0.4	4	17264	123791	25359	14314	155.3	68.3	384.3	189.7	206	132	331	404	6.9%	11.5%	0.6%	3.8%
1000	10	0.4	4	17262	123684	25191	14303	187.8	135.0	430.2	635.6	186	163	354	1544	6.8%	11.5%	0.9%	3.7%
Total			12	17268	123748	25299	14330	171.6	109.7	334.8	331.7	204	148	309	753	6.8%	11.4%	0.7%	3.6%

**Table 7** Average results on different M-SPRP instance groups

$\Omega_t$	$\Omega$	$\hat{\alpha}$	$\epsilon$	#Ins	Solution cost				CPU				B.Cuts				%VSS			
					G1	G2	G3	G4	G1	G2	G3	G4	G1	G2	G3	G4	G1	G2	G3	G4
6	216	5	0.2	2	12872	95390	18841	10976	1373.1	404.5	28.7	62.1	2623	707	80	144	2.9%	5.7%	0.1%	0.1%
8	512	5	0.2	2	12936	96280	18970	11070	1576.3	732.9	70.3	188.9	1162	546	76	166	2.7%	5.1%	0.0%	0.1%
Total			4	12904	95835	18905	11023	1474.7	568.7	49.5	125.5	1893	626	78	155	2.8%	5.4%	0.1%	0.1%	
6	216	10	0.2	2	13473	100635	19970	11936	302.9	489.6	16.9	530.4	562	792	44	1015	5.8%	10.0%	0.1%	1.3%
8	512	10	0.2	2	13520	100997	19925	11946	1077.8	365.3	61.1	1720.4	763	300	74	1096	5.5%	9.4%	0.3%	0.9%
Total			4	13496	100816	19948	11941	690.4	427.4	39.0	1125.4	662	546	59	1055	5.7%	9.7%	0.2%	1.1%	
6	216	5	0.4	2	12834	94015	19324	11268	416.8	832.4	25.0	105.7	844	1459	69	1589	7.1%	12.0%	0.1%	1.1%
8	512	5	0.4	2	12902	95520	19474	11839	2116.7	279.1	69.3	12.7	1556	235	73	12	6.9%	10.6%	0.1%	0.1%
Total			4	12868	94767	19399	11554	1266.7	555.7	47.2	59.2	1200	847	71	800	7.0%	11.3%	0.1%	0.6%	
6	216	10	0.4	2	13555	100992	21210	12181	328.1	123.3	55.0	182.1	619	232	146	391	12.7%	18.5%	0.1%	6.8%
8	512	10	0.4	2	13673	101905	21050	12334	682.0	838.1	104.6	89.9	528	644	126	109	11.3%	16.7%	0.1%	4.4%
Total			4	13614	101448	21130	12258	505.0	480.7	79.8	136.0	573	438	136	250	12.0%	17.6%	0.1%	5.6%	

cost saving compared to solving an expected value problem (as measured by %VSS). The instance group G2, however, could be solved more efficiently and provide the highest savings among all instance groups. For the M-SPRP, the instance groups G1 and G4 are the most difficult in general. The stochastic solutions for the instance group G3, as for the 2-SPRP, provide the smallest cost savings while those of the instance groups G1 and G2 provide the highest cost savings. We can also

observe that the %VSS increases when the penalty factor or the demand uncertainty level increase.

In the Online Supplement, we report the results of additional computational experiments to evaluate the sensitivity of the models.

## 5. Reoptimization Capabilities of the Benders Decomposition Algorithm for Two-Stage and Multi-Stage SPRPs

The purpose of this section is to study the reoptimization capabilities offered by the Benders decomposition algorithm. The general idea is that changes to the customer demands affect only the right-hand-side of the primal subproblem and, thus, only the objective function of the dual subproblem. Since the dual subproblem polyhedron is unaffected, extreme points identified when solving a given problem instance remain valid after demand changes and one can generate cuts for a new instance directly from these points. A new optimal solution is typically obtained in a few iterations. If one instead employs a branch-and-cut algorithm, it must start from scratch every time a change is made, which can be very time consuming. We now discuss how this reoptimization approach can be used to warm start the algorithm and expedite the solution process in two practical settings: a sample average approximation for the 2-SPRP and a rollout algorithm for the M-SPRP.

In both cases, the procedure starts by solving the first problem from scratch using the BBC and keeping the dual solutions generated during the solution process. When the problem is reoptimized, Benders cuts corresponding to the demand scenarios in the new problem are computed using the dual solutions that were already generated in the previous replications. These cuts are added to the Benders master problem before the Benders algorithm starts. To avoid adding too many cuts, we use a preprocessing step to select the cuts. This process starts by solving the Benders master without any Benders cuts to obtain an initial solution along with the values of  $\eta_g$ . Then, the cuts with a non-negative left-hand-side value for the initial solution are used as initial Benders cuts. The number of initial Benders cuts is limited to 5000. If the number of cuts exceeds the limit, those with the largest left-hand-side value are selected first.

### 5.1. Sample Average Approximation for the 2-SPRP

Sample average approximation (SAA) (Kleywegt et al. 2002b) is a Monte-Carlo simulation-based sampling method developed to solve problems where the number of scenarios is very large. It can also be applied to problems with continuous distributions or with an infinite number of scenarios. Given a large scenario set, denoted by  $\Omega'$ , which is intractable, the SAA consists of solving a number  $\mathcal{M}$  of smaller and tractable problems with a sample of size  $|\Omega| \ll |\Omega'|$ . In the SAA process, one can calculate the SAA gap which is the estimated difference between the solution obtained by solving the  $\mathcal{M}$  replications of the sample size  $|\Omega|$  and a statistical lower bound on the optimal value for the large scenario set  $\Omega'$ . This gap can be determined by a sample average function. In practice, one can choose a sample size  $|\Omega|$  and the number of replications  $\mathcal{M}$  that are most appropriate for the problem in terms of solution quality and computing time. In the SPRP, where different demand scenarios are considered in each replication, one can take advantage of the reoptimization capabilities of Benders decomposition to solve the problem more efficiently. We provide more details on the SAA scheme and steps to compute the SAA gap in the Online Supplement.

Tests were performed with sample sizes  $|\Omega| = 100, 200, 500$  and  $1000$  and a number of replications  $\mathcal{M} = 100, 50, 20$  and  $10$ , respectively, which makes the total number of evaluated scenarios equal to  $10,000$  for every sample size. We performed the tests on instances with  $n = 10, 15$ ,  $l = 3$  and  $m = 1$  to avoid excessive computing times. In addition to the uniform distribution, experiments were also performed using normal and gamma distributions with parameters chosen so that the current demand range  $[\bar{d}_{it}(1 - \epsilon), \bar{d}_{it}(1 + \epsilon)]$  corresponds approximately to a 99.5% confidence interval. The size of the large scenario set was chosen equal to  $|\Omega'| = 10,000$  to evaluate the SAA gap.

Scenarios were generated a priori and all the algorithms were tested on the same scenario sets to ensure a fair comparison. In the first set of experiments, we compared the results provided by different algorithms. The results are provided in Table 8. The results obtained by the BBC algorithm with reoptimization are shown in column *BBC-ReOpt* and the average percentage of the number of initial Benders cuts in each replication is shown in column *%I.Cuts*. Column *T.CPU*

indicates the average total time spent to solve all replications of the same instance and boldface is used to indicate the smallest time. Columns *CPU*, *Nodes* and *B.Cuts* are the same as in the previous section and they show the average results per replication.

**Table 8** Performance of the algorithms using the SAA method

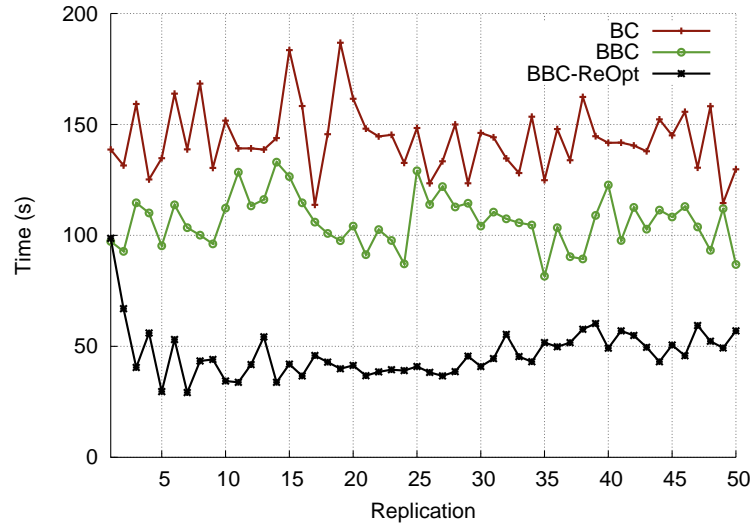
Distribution	$\Omega$	$\mathcal{M}$	BC		BBC				BBC-ReOpt					
			T.CPU	CPU	Nodes	T.CPU	CPU	Nodes	B.Cuts	T.CPU	CPU	Nodes	B.Cuts	%I.Cuts
Uniform	100	100	2740.4	27.4	35.6	3797.8	38.0	763.3	274.3	<b>2379.0</b>	23.8	332.2	2853.5	97.2
	200	50	4361.3	87.2	44.0	3413.3	68.3	771.8	280.3	<b>1633.3</b>	32.7	345.6	2166.2	96.1
	500	20	9484.7	474.2	51.4	2969.6	148.5	780.8	282.2	<b>1284.3</b>	64.2	388.7	1147.5	91.5
	1000	10	19265.6	1926.6	53.5	2877.3	287.7	814.2	303.7	<b>1332.1</b>	133.2	441.8	793.9	84.2
Average			8963.0	628.9	46.1	3264.5	135.6	782.5	285.1	<b>1657.2</b>	63.5	377.1	1740.3	94.5
Normal	100	100	2706.6	27.1	35.8	3275.3	32.8	670.2	238.3	<b>1960.9</b>	19.6	276.0	2610.8	97.4
	200	50	4182.0	83.6	43.4	2949.8	59.0	690.3	245.5	<b>1363.7</b>	27.3	293.6	1866.6	96.3
	500	20	9256.3	462.8	55.3	2605.8	130.3	710.9	251.4	<b>1066.5</b>	53.3	324.0	1015.8	91.7
	1000	10	18912.6	1891.3	58.5	2405.1	240.5	708.8	259.3	<b>1213.7</b>	121.4	374.7	741.4	84.2
Average			8764.4	616.2	48.2	2809.0	115.6	695.0	248.6	<b>1401.2</b>	55.4	317.1	1558.7	94.6
Gamma	100	100	2838.7	28.4	45.1	3171.5	31.7	694.7	227.6	<b>1997.7</b>	20.0	302.3	2526.5	97.3
	200	50	4375.9	87.5	55.4	2892.0	57.8	728.9	237.9	<b>1342.1</b>	26.8	329.7	1736.3	96.1
	500	20	9309.3	465.5	64.8	2544.8	127.2	742.1	245.4	<b>1110.1</b>	55.5	364.8	1001.8	91.6
	1000	10	19088.2	1908.8	63.4	2453.9	245.4	807.2	264.1	<b>1132.9</b>	113.3	384.0	697.5	84.4
Average			8903.0	622.5	57.2	2765.6	115.5	743.2	243.7	<b>1395.7</b>	53.9	345.2	1490.5	94.5
Total Average			8876.8	622.5	50.5	2946.4	122.3	740.3	259.2	<b>1484.7</b>	57.6	346.4	1596.5	94.5

These results clearly indicate the benefits of reoptimization when solving the SPRP within the SAA method. The BBC algorithm using reoptimization could reduce the average total computing time by approximately 50% and 83% compared to the BBC without reoptimization and BC, respectively. The average number of Benders cuts generated by the BBC-ReOpt is significantly larger than for the BBC. However, the majority of them (94.5%) are the initial cuts generated from previous replications and only 5.5% of the number of Benders cuts or 87.7 cuts on average were newly generated at each replication. To further demonstrate the benefits of reoptimization, Figure 3 shows the average computing time spent in each replication to solve the instances with  $n = 15$ ,  $\epsilon = 0.10$ ,  $|\Omega| = 200$  and a uniform distribution. The computing times after the first replication are significantly reduced when using the BBC-ReOpt.

To evaluate the impact of the sample size, we performed further experiments and compared the



**Figure 3** Computing time spent in each replication in the SAA method.



solution quality obtained with different sizes using the BBC-ReOpt. These results are provided in the Online Supplement. We observed that the largest sample size  $|\Omega| = 1000$  can provide the best average SAA gap with the least variation, while the sample size  $|\Omega| = 500$  is generally the best in terms of trade-off between solution quality and computing time.

## 5.2. Rollout Algorithm for the M-SPRP

Unlike the 2-SPRP where the upper bound of a problem can be computed in a straightforward manner when the first stage decisions are known, in the M-SPRP one must construct an implementable and feasible policy to obtain a valid upper bound (Shapiro 2003). More importantly, in the M-SPRP, since the size of the problem grows exponentially with the number of scenarios in each stage, this task becomes very challenging for instances involving a large number of time periods. In this section, we introduce a rollout algorithm (Bertsekas et al. (1997)) that exploits the reoptimization capability of the Benders decomposition to obtain an upper bound for the M-SPRP. In a rollout algorithm, decisions are made sequentially by using a heuristic to approximate the impact of these decisions in a look-ahead mechanism. In our case, this heuristic consists of solving the two-stage problem on a shortened planning horizon, which serves as an approximation of the multi-stage problem. Similarly to a rolling horizon approach, our algorithm divides the full planning

horizon into a series of shorter and overlapping planning intervals. For each interval, we then solve the 2-SPRP to make the (binary) planning decisions and we solve the resulting M-SPRP subproblem to derive the optimal (continuous) recourse decisions for the multi-stage problem. After fixing the decisions for the first few periods in this interval, these steps are repeated for the next interval. We refer to the overall heuristic as a multi-stage rollout algorithm (M-RO). An important aspect of this heuristic is that one needs to compute the inventory levels that become initial conditions for the next interval. These initial inventory levels must be passed through the scenario tree of the M-SPRP during the process. Figure 4 illustrates the idea of the rollout algorithm for the M-SPRP with intervals of two periods and an overlap of one period.

The heuristic consists of two main steps at each iteration. In the first step, we solve a modified 2-BRF model with the scenarios defined at the end of the current interval to determine the first-stage integer decisions. To expedite this process, we can also apply the reoptimization technique by using initial Benders cuts generated from the dual solutions in the previous intervals with the updated demand scenarios and initial inventory levels of the current interval. In the second step, the multi-stage recourse decisions are determined by solving a M-BRF subproblem. Then the algorithm continues until the end of the full planning horizon. We denote by  $\hat{\tau}$  the length of the time intervals considered in each problem and by  $\hat{r} < \hat{\tau}$  the number of overlapping time periods.

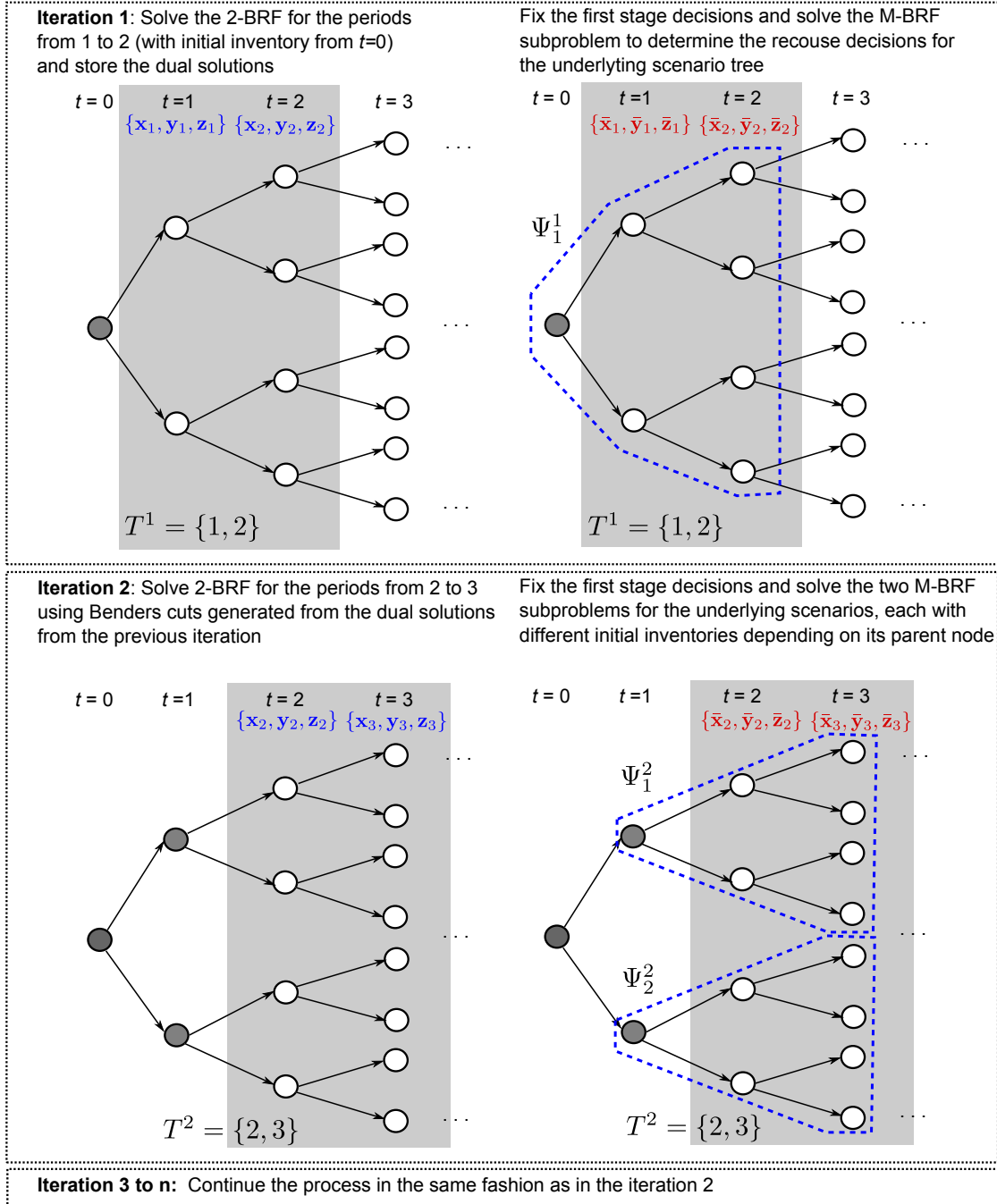
The M-RO can be described as follows:

1. Compute the number  $\mathcal{M} = \left\lceil \frac{|T| - \hat{\tau}}{\hat{\tau} - \hat{r}} \right\rceil + 1$  and construct a set of time intervals  $\mathcal{T}^1, \dots, \mathcal{T}^{\mathcal{M}}$ , each of size  $|\mathcal{T}^m| = \hat{\tau}$ ,  $m = 1, \dots, \mathcal{M} - 1$ , except the last one whose size is  $|\mathcal{T}^{\mathcal{M}}| = |T| - (\mathcal{M} - 1) \cdot (\hat{\tau} - \hat{r})$ . Create the set  $\mathcal{S}^m$  of scenarios associated with each interval, create an empty pool of dual solutions  $\Phi$ , and set the initial solution cost  $UB = 0$ .
2. For  $m = 1, \dots, \mathcal{M}$ , denote by  $t_f$  and  $t_t$  the first and the last period at the current iteration  $m$ , and by  $t'_f$  the first period of the next iteration  $m + 1$ . Do the following:
  - (a) **Determine the first stage decisions:**
    - i. If  $t_t - (t_f - 1) = \hat{\tau}$ , retrieve the dual solutions from the pool  $\Phi$  based on the selection criteria to create initial Benders cuts.

- ii. Parameterize the unit costs by dividing all the first stage cost parameters by  $|\Omega|/|\mathcal{S}^m|$ .

This ensures that the probability associated with each scenario remains constant, i.e.,  $\rho_\omega = \frac{1}{|\Omega|}, \forall \omega \in \mathcal{S}^m$ , throughout the process, so that the dual solutions generated in a given iteration can be used to generate Benders cuts in subsequent iterations.

Figure 4 Illustration of the multi-stage rollout algorithm.



- iii. Solve the 2-SPRP on the set of scenarios  $\mathcal{S}^m$  for the period  $t_f$  to  $t_t$  by using Benders cuts generated from the retrieved dual solutions to obtain the  $(\bar{\mathbf{y}}, \bar{\mathbf{z}}, \bar{\mathbf{x}})$ . Add the dual solutions associated with the new Benders cuts to the pool  $\Phi$ .
- (b) **Determine the multi-stage recourse decisions:** For each scenario node  $j = 1, \dots, |J|$  of the first period in  $\mathcal{T}^m$ , do the following:
- i. Construct a set of succeeding scenarios associated with  $j$ , denoted by  $\Psi_j$ .
  - ii. Solve the M-SPRP<sup>LP</sup> $(\bar{\mathbf{y}}, \bar{\mathbf{z}}, \bar{\mathbf{x}})$  to obtain the solution vectors  $(\bar{\mathbf{p}}, \bar{\mathbf{I}}, \bar{\mathbf{q}}, \bar{\mathbf{e}})$ .
  - iii. To update the solution cost, multiply the total cost of the period  $t_f$  to  $t'_f$  by  $|\mathcal{S}^m|/|\Omega|$  and add to the current solution cost  $UB$ .

Table 9 shows the results of using the M-RO approach compared to the exact branch-and-cut algorithm for instances with 5, 10 or 20 customers, 5 or 6 time periods and 2 or 5 scenarios per time period (which results in the total number of scenarios reported in Column  $|\Omega|$ ). Columns *Best* show the best results obtained by the exact algorithms (either the BC or BBC). Columns *M-RH (BC)*, *M-RH (BBC)* and *M-RH (BBC and reopt)* show the results obtained by the M-RH approach when the 2-SPRP is handled by the BC, BBC and BBC (with reoptimization), respectively. Column *Gap* shows the average optimality gap (%) of the heuristic compared to the best lower bound of the exact algorithms. One can see the BBC with reoptimization has a much improved performance, both in terms of the upper bound and the average CPU time. Using a larger overlap  $\hat{r}$  also improves the solution quality. The optimality gap of the upper bound produced by the BBC with reoptimization with 3 periods and 2 overlapping periods is approximately 6.9%, while the optimality gap of the best exact procedure is 10.9%. Furthermore, the optimality gap of the heuristic solution is only 0.6% when comparing with the solutions of the 18 instances that were solved to optimality.

## 6. Conclusions

We have addressed demand uncertainty in the production routing problem within two-stage and multi-stage decision processes. To solve these problems, we have proposed two different classes of solution algorithms: branch-and-cut and Benders decomposition. The BBC for the two-stage

**Table 9** Performance of the multi-stage rollout algorithm

$n$	$ T $	$ \Omega^t $	$ \Omega $	Best (BC and BBC)		M-RH (BC)				M-RH (BBC)				M-RH (BBC and reopt)						
						$\hat{r} = 1$		$\hat{r} = 2$		$\hat{r} = 1$		$\hat{r} = 2$		$\hat{r} = 1$		$\hat{r} = 2$				
				LB	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU
5	5	2	32	17444	0.0	2.2	0.7	1.2	0.7	1.7	0.7	3.8	0.7	4.8	0.7	4.8	0.7	5.6		
5	5	5	3125	17681	0.0	1662.4	0.6	3013.6	0.8	2930.2	0.6	221.4	0.8	210.0	0.6	296.8	0.8	274.3		
10	5	2	32	26170	0.0	6.5	0.1	2.6	0.1	3.6	0.1	8.0	0.1	11.4	0.1	8.2	0.1	14.7		
10	5	5	3125	26024	1.5	6348.9	2.2	7223.2	2.2	7375.6	2.2	144.3	2.2	278.3	2.2	405.0	2.2	332.3		
15	5	2	32	35466	0.0	37.0	0.2	6.4	1.1	11.7	0.2	86.1	1.1	198.3	0.2	152.4	1.1	283.0		
15	5	5	3125	33018	10.7	7200.0	46.2	7265.5	56.8	7544.0	9.1	679.4	9.9	3328.0	9.1	879.1	9.9	3038.9		
20	5	2	32	41282	0.0	57.6	0.3	11.8	0.3	17.4	0.5	618.9	0.5	1369.4	0.4	503.3	0.4	1118.3		
20	5	5	3125	40664	4.0	7200.0	44.1	7268.4	16.0	7726.6	3.7	5819.0	4.0	8980.2	3.3	3122.6	4.0	4971.5		
5	6	2	64	22027	0.0	11.1	1.1	1.7	0.7	3.1	1.1	5.4	0.7	12.2	1.1	7.0	0.7	20.5		
5	6	5	15625	17072	32.2	7200.0	42.7	10406.7	70.3	11823.2	24.4	1177.0	25.2	1281.0	24.4	599.4	23.9	1387.4		
10	6	2	64	30828	0.0	16.5	6.0	5.1	0.1	5.1	6.0	9.4	0.1	24.9	6.0	8.6	0.1	34.7		
10	6	5	15625	23720	46.7	7200.0	64.3	14502.8	67.9	21887.4	28.6	605.9	24.9	1673.2	28.6	518.3	25.5	1768.7		
15	6	2	64	42636	0.0	67.1	3.7	12.3	2.9	28.5	3.7	32.1	1.2	404.0	3.7	41.3	1.2	258.2		
15	6	5	15625	35026	47.7	7200.0	69.1	14529.6	66.9	22152.5	22.3	2911.7	21.2	9279.7	22.3	1136.6	21.2	6391.1		
20	6	2	64	48882	0.0	222.4	3.2	17.5	0.3	45.7	3.2	158.8	0.2	1587.3	3.2	94.0	0.1	1189.4		
20	6	5	15625	41893	32.1	7200.0	66.3	18281.8	66.3	22650.0	19.5	3803.6	19.1	17215.9	19.5	2524.3	18.2	7582.1		
Total				31239	10.9	3227.0	21.9	5159.4	22.1	6512.9	7.9	1017.8	7.0	2866.2	7.8	643.9	6.9	1791.9		

problem is further improved by several computational enhancements, namely, lower bound lifting inequalities, scenario group cuts and Pareto-optimal cuts. These enhancements can be used for the multi-stage problem which also benefits from a warm start. The computational results show that, for both the two-stage and the multi-stage problems, the BC is efficient in handling small instances with a small number of scenarios while the BBC with the enhancements outperforms the BC on larger instances and particularly on instances with a large number of scenarios. To handle practical instances, we have further discussed reoptimization capabilities of the Benders decomposition in the context of the sample average approximation for the two-stage problem and of a rollout algorithm for the multi-stage problem.

## Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada under grants 227837-09 and 342182-09, and by the Fonds de recherche du Québec – Nature et technologies. This support is gratefully acknowledged. The authors also thank the RQCHP for providing computing facilities for our experiments. Finally, we are grateful to the Associate Editor and to two anonymous referees for their valuable

comments on an earlier version of the paper.

## References

- Adelman, D. 2004. A price-directed approach to stochastic inventory/routing. *Oper. Res.* **52** 499–514.
- Adulyasak, Y. 2012. Models and solution algorithms for production routing problems. Ph.D. thesis, Logistics and Operations Management Department, HEC Montréal.
- Adulyasak, Y., J.-F. Cordeau, R. Jans. 2014a. Formulations and branch-and-cut algorithms for multi-vehicle production and inventory routing problems. *INFORMS J. Comput.* **26** 103–120.
- Adulyasak, Y., J.-F. Cordeau, R. Jans. 2014b. Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Sci.* **48** 20–45.
- Adulyasak, Y., J.-F. Cordeau, R. Jans. 2014c. The production routing problem: A review of formulations and solution algorithms. *Comput. Oper. Res.* Forthcoming.
- Andersson, H., A. Hoff, M. Christiansen, G. Hasle, A. Løkketangen. 2010. Industrial aspects and literature survey: Combined inventory management and routing. *Comput. Oper. Res.* **37** 1515–1536.
- Applegate, D., R. Bixby, V. Chvátal, W. Cook. 2011. The Concorde TSP Solver web site. Available at <http://www.math.uwaterloo.ca/tsp/concorde.html>.
- Archetti, C., L. Bertazzi, G. Laporte, M. G. Speranza. 2007. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Sci.* **41** 382–391.
- Archetti, C., L. Bertazzi, G. Paletta, M. G. Speranza. 2011. Analysis of the maximum level policy in a production-distribution system. *Comput. Oper. Res.* **38** 1731–1746.
- Benders, J. F. 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* **4** 238–252.
- Bertazzi, L., A. Bosco, F. Guerriero, D. Laganà. 2013. A stochastic inventory routing problem with stock-out. *Transport. Res. C-Emer.* **27**.
- Bertsekas, D., J. Tsitsiklis, C. Wu. 1997. Rollout algorithms for combinatorial optimization. *J. Heuristics* **3** 245–262.
- Birge, J. R., F. V. Louveaux. 1988. A multicut algorithm for two-stage stochastic linear programs. *Eur. J. Oper. Res.* **34** 384–392.

- Birge, J. R., F. V. Louveaux. 2011. Two-stage recourse problems. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering, Springer New York, NY, USA, 181–263.
- Codato, G., M. Fischetti. 2006. Combinatorial Benders' cuts for mixed-integer linear programming. *Oper. Res.* **54** 756–766.
- Coelho, L. C., J.-F. Cordeau, G. Laporte. 2014a. Heuristics for dynamic and stochastic inventory-routing. *Comput. Oper. Res.* Forthcoming.
- Coelho, L.C., J.-F. Cordeau, G. Laporte. 2014b. Thirty years of inventory routing. *Transportation Sci.* **48** 1–19.
- Cordeau, J.-F., F. Soumis, J. Desrosiers. 2001. Simultaneous assignment of locomotives and cars to passenger trains. *Oper. Res.* **49** 531–548.
- de Camargo, R. S., G. de Miranda Jr., R. P.M. Ferreira. 2011. A hybrid outer-approximation/Benders decomposition algorithm for the single allocation hub location problem under congestion. *Oper. Res. Lett.* **39** 329–337.
- de Camargo, R.S., G. de Miranda Jr., H.P. Luna. 2008. Benders decomposition for the uncapacitated multiple allocation hub location problem. *Comput. Oper. Res.* **35** 1047–1064.
- Federgruen, A., P. Zipkin. 1984. A combined vehicle routing and inventory allocation problem. *Oper. Res.* **32** 1019–1037.
- Fortz, B., M. Poss. 2009. An improved Benders decomposition applied to a multi-layer network design problem. *Oper. Res. Lett.* **37** 359–364.
- Hopp, W., M. Spearman. 2000. *Factory Physics*. 2nd ed. Irwin/McGraw-Hill, NY, USA.
- Hvattum, L. M., A. Løkketangen. 2009. Using scenario trees and progressive hedging for stochastic inventory routing problems. *J. Heuristics* **15** 527–557.
- Hvattum, L. M., A. Løkketangen, G. Laporte. 2009. Scenario tree-based heuristics for stochastic inventory-routing problems. *INFORMS J. Comput.* **21** 268–285.
- Jaillet, P., J. F. Bard, L. Huang, M. Dror. 2002. Delivery cost approximations for inventory routing problems in a rolling horizon framework. *Transportation Sci.* **36** 292–300.

- Kleywegt, A. J., V. S. Nori, M. W. P. Savelsbergh. 2002a. The stochastic inventory routing problem with direct deliveries. *Transportation Sci.* **36** 94–118.
- Kleywegt, A. J., V. S. Nori, M. W. P. Savelsbergh. 2004. Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Sci.* **38** 42–70.
- Kleywegt, A. J., A. Shapiro, T. Homem de Mello. 2002b. The sample average approximation method for stochastic discrete optimization. *SIAM J. Optim.* **12** 479–502.
- Magnanti, T. L., R. T. Wong. 1981. Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Oper. Res.* **29** 464–484.
- McDaniel, D., M. Devine. 1977. A modified Benders' partitioning algorithm for mixed integer programming. *Management Sci.* **24** 312–319.
- Naoum-Sawaya, J., S. Elhedhli. 2013. An interior-point Benders based branch-and-cut algorithm for mixed integer programs. *Ann. Oper. Res.* **210** 33–55.
- Shapiro, A. 2003. Monte Carlo sampling methods. A. Ruszczyński, A. Shapiro, eds., *Stochastic Programming, Handbooks in Operations Research and Management Science*, vol. 10. Elsevier, Amsterdam, 353–425.
- Solyali, O., J.-F. Cordeau, G. Laporte. 2012. Robust inventory routing under demand uncertainty. *Transportation Sci.* **46** 327–340.
- Solyali, O., H. Süral. 2011. A branch-and-cut algorithm using a strong formulation and an a priori tour-based heuristic for an inventory-routing problem. *Transportation Sci.* **45** 335–345.
- Van Slyke, R., R. Wets. 1969. *L*-shaped linear programs with applications to optimal control and stochastic programming. *SIAM J. Appl. Math.* **17** 638–663.