An adjustable sample average approximation algorithm for the stochastic production-inventory-routing problem*†

Agostinho Agra, Cristina Requejo, Filipe Rodrigues

Department of Mathematics, University of Aveiro, 3810-193 Aveiro, Portugal.
{aagra, crequejo, fmrodrigues}@ua.pt

Abstract

We consider a stochastic single item production-inventory-routing problem with a single producer, multiple clients and multiple vehicles. At the clients, demand is allowed to be backlogged incurring a penalty cost. Demands are considered uncertain.

A recourse model is presented and valid inequalities are introduced to enhance the model.

A new general approach that explores the sample average approximation (SAA) method is introduced. In the sample average approximation method, several sample sets are generated and solved independently in order to obtain a set of candidate solutions. Then the candidate solutions are tested on a larger sample and the best solution is selected among the candidates. In contrast to this approach, called static, we propose an adjustable approach that explores the candidate solutions in order to identify common structures. Using that information, part of the first-stage decision variables are fixed and the resulting restricted problem is solved for a larger size sample. Several heuristic algorithms based on the mathematical model are considered within each approach.

Computational tests based on randomly generated instances are conducted to test several variants of the two approaches. The results show that the new adjustable SAA heuristic performs better than the static one for most of the instances.

Keywords: Inventory routing; Stochastic programming; Sample average approximation algorithm; Hybrid heuristic; Demand uncertainty; Iterated local search; Adaptive heuristic.

1 Introduction

We consider a single item stochastic production-inventory-routing (SPIR) problem with a single supplier/producer, multiple retailers/clients and multiple vehicles.

*The research of the authors has been partially supported by Portuguese funds through the Center for Research and Development in Mathematics and Applications (CIDMA) and FCT, the Portuguese Foundation for Science and Technology, within project UID/MAT/04106/2013. The research of the third author has been funded by FCT under Grant PD/BD/114185/2016.
†Published in Networks 72 (1), 5–24, 2018. [doi: 10.1002/net.21796]
A vendor managed inventory approach is followed where the supplier monitors the inventory at the retailers and decides on the replenishment policy for each retailer. Inventory aspects are considered at both the supplier and the retailers. Demand is allowed to be backlogged at the retailers and in that case a penalty cost is incurred. Backlog is not allowed at the supplier. Demands are considered uncertain, following a known probability distribution. A constant production capacity at the supplier is assumed. For the distribution plan, multiple vehicles are considered. The decision maker has to decide the production and the distribution plans for a finite time horizon. The production plan consists in defining the production periods and the amount to produce in each one of those periods. The distribution plan defines the retailers that should be visited in each time period, the quantities to deliver to each visited retailer, and the corresponding route for each used vehicle in each time period. Bounds on the delivered quantities and on client inventories are considered. The goal is to minimize the production and the routing cost plus the expected inventory and the penalty costs for backlogged demand.

We assume the production plan and the choice of which clients to visit in each time period (and consequently the routing) are first-stage decisions, that is, decisions that must be taken before the scenario is revealed. The quantities to deliver to each client in each time period as well as the inventory levels can be adjusted to the scenario (known as second-stage decisions). Such assumptions may hold for short planning horizons.

Complex problems combining production, inventory and routing decisions have received much attention in recent years [6, 9, 11, 14, 16, 17, 19, 20, 23, 29, 33, 38, 40]. For a survey on previous publications see [4]. Several such studies have been motivated by real applications [5, 7, 9, 29]. The advantages of coordination and integration of different supply chain decisions are reported by several authors. Hein and Almeder [25] study the benefits of integrated planning for a complex supply production network embedded in a dynamic discrete-type system, and conclude that substantial cost savings are possible with an integrated planning approach compared to a classical sequential approach. Darvish et al. [19] observe that making the right lot-sizing decisions becomes even more significant in interconnected logistic networks and show the benefits of an integrated approach to supply chain decision-making. Integrated approaches for a production-routing problem are also compared against uncoordinated ones in [37].

Most of the solution approaches for such complex problems use heuristics. Aksi et al. [1] and Chitsaz et al. [17] propose two-phase iterative methods where the original inventory-routing problem is split into two subproblems, an inventory problem and a routing problem. Agra et al. [6] also propose the decomposition of the production-inventory-routing problem into a production-inventory subproblem and a routing subproblem using a new Lagrangian decomposition enhanced with valid inequalities. Chitsaz et al. [16] formulate the problem as a MIP and propose a three-phase decomposition matheuristic that relies on the iterative solution of different subproblems. Brahimi and Aouam [14] propose a hybrid heuristic in which a relax-and-fix strategy is combined with a local search strategy. Qiu et al. [33] develop a branch-and-price heuristic by incorporating a column generation formulation based on the Dantzig-Wolfe decomposition. Zhang et al. [41] present an iterative MILP based heuristic that uses restricted sets of routes that are dynamically updated. Russell [38] and Solyal and Süral [40] propose heuristics where routes are computed in advance and, in a second-stage, the predetermined routes are used to simplify the MIP models.
Although, in general, instances from such complex problems are not solved to optimality, exact approaches such as a branch-and-cut \cite{2} and branch-and-cut-and-price algorithms \cite{20, 33} have also been considered. Small benchmark instances can be solved to optimality, see for instance, Avella et al. \cite{12}.

For several related problems, where no backlog is allowed, extended formulations \cite{2, 12, 24, 37} and valid inequalities \cite{11, 37} have been considered.

Uncertainty has also been considered for related problems. Solyah et al. \cite{39} introduce a robust optimization approach for an inventory routing problem with uncertain demands. Aghezzaf \cite{5} considers a variant of the inventory routing optimization problem where customer demand rates and travel times are stochastic but stationary. Stochastic approaches for related inventory routing problems have been also considered \cite{3, 7, 8, 13, 34}. Rahim et al. \cite{34} propose a deterministic equivalent approximation model to the original stochastic problem. Then, using Lagrangian relaxation, the deterministic model is decomposed into an inventory allocation subproblem and a vehicle routing subproblem. Bertazzi et al. \cite{13} propose a hybrid roll-out algorithm for an inventory-routing problem where an order-up-to level policy is followed for each retailer.

A classical approach for handling stochastic problems is the sample average approximation (SAA) method \cite{28, 43}. In this method the expected value of the objective function is approximated by a sample average estimate obtained from a random sample. The resulting SAA problem is solved for different samples in order to obtain a set of candidate solutions. Then, these candidate solutions are tested on a larger sample and the best solution for that sample is chosen. Within related inventory-routing problems the SAA was used in \cite{3, 7, 8}. In a closely related problem, Adulyasak et al. \cite{3} consider two-stage and multi-stage stochastic production routing problems with demand uncertainty. A Benders decomposition approach is used to solve each sample of the SAA method to optimality. However no backlog is allowed in their model. In \cite{7} a practical maritime inventory routing problem is solved using the SAA method. For each sample the stochastic problem was solved using a commercial solver based on a branch-and-cut algorithm with a running time limit. This solution procedure is heuristic since the instances are not solved to optimality. Consequently, the SAA method generates solutions whose objective function value vary significantly from sample to sample. In order to circumvent this lack of stability, Agra et al. \cite{8} propose the use of a local branching heuristic to improve an initial solution for each sample generated within the SAA method. By generating a near local optimal solution for each sample the stability issues were mitigated.

A common approach to deal with inventory models under uncertainty is to consider the possibility of satisfying demands with delay and to allow shortfalls. This assumption aims to use the studied problems to approximate the practical ones faced by companies, and to ensure that a two-stage recourse problem has complete recourse (for each first-stage solution there is a feasible assignment of the second-stage solutions). However, as reported in \cite{8}, penalizing the backlogged demand makes the instance harder to solve by exact methods based on mixed integer formulations, since the integrality gaps of the linear solutions tend to be high. Hence, these SPIR problems tend to be even harder than the deterministic ones when no backlog is allowed.

When the SAA problems are not solved to optimality for each sample, theoretical results on convergence fail and the SAA method acts as a heuristic. Based on this
observation, and on the assumption that one cannot solve efficiently these complex SPIR problems even for small size instances, a natural approach is to follow the SAA method and solve each one of the SAA problems defined for small size samples heuristically. Then, the best solution among the generated solutions is chosen. We call this the static approach. In contrast with this approach, in which the solution to the problem is one of the candidate solutions, we propose a new heuristic approach that explores the knowledge of the several candidate solutions in order to solve the SAA problem for a larger and more representative set of scenarios. Following this approach, a new heuristic named Adjusted Sample Average Approximation (ASAA) algorithm is proposed. In this algorithm, a feasible solution for each sample is generated using a heuristic scheme. Then, these solutions are explored in order to identify first-stage variables that are frequently fixed to zero or to one, and a partial fixed solution is constructed, since only some of those variables are fixed. Finally, the restricted model is solved for a larger sample keeping all the remaining variables free. The proposed adjustable heuristic approach has several advantages in relation to the classical SAA method: i) it does not require solving each sample set to optimality; ii) it does not require the use of large sample sets, and iii) it allows the final solution to be adjusted to a larger sample set since many first-stage variables are kept free. Further, the number of variables that are fixed can be easily controlled.

To the best of our knowledge, the use of the information from several solutions obtained with the SAA method to obtain a partial solution was introduced in a preliminary version of this work [10]. However, algorithms that use information of previous solutions to derive partial solutions are not new. Rochet and Taillard [35] introduce the Adaptive Memory concept to describe a set of good solutions, that are kept in a pool that is dynamically updated taking into account common features of the solutions. Based on this concept, a very general metaheuristic focused on the exploitation of strategic memory components was proposed by Glover [22]. Such a metaheuristic, known as Adaptive Memory Programming (AMP), has been applied over the years to solve several hard combinatorial optimization problems, such as vehicle routing problems [23, 41], stochastic production distribution network design [15] and supplier selection problems [42]. Another closely related concept is the Concentration Set (CS) introduced by Rosing and ReVelle [36] to solve the deterministic p-median problem heuristically. A large number of solutions are generated heuristically. Then, the best \( v \) solutions are selected to the CS and this set is used to set to one (zero) all the variables that take value one (zero) in all the \( v \) solutions. The remaining variables are kept free. Threshold values were also used in [26] for a dynamic facility location problem with capacity levels. Using a Lagrangian relaxation model, some of those levels are eliminated (the corresponding variables are fixed to zero) according to the information provided by the set of solutions.

In this paper we introduce a general approach that explores the SAA in order to handle large size instances of the production-inventory-routing problem. We discuss a mixed integer model and possible enhancements. Based on the enhanced model with valid inequalities, two main approaches are compared: a static approach and an adjustable SAA approach. In order to improve solutions, an iterated local search heuristic based on local branching is presented. The adjustable approach can be easily extended for other two-stage stochastic problems where the corresponding deterministic problems are
already difficult to solve to optimality for realistic size instances.

The rest of this paper is organized as follows. The mathematical model is presented in Section 2. Model improvements, such as the valid inequalities and an extended formulation, are discussed in Section 3. The SAA method, the static heuristic approaches based on the SAA method, and the new ASAA heuristic are introduced in Section 4. Computational tests that compare variants of the static and the adjustable approaches are presented in Section 5. Final conclusions are drawn in Section 6.

2 Problem specifications and mathematical formulation

In this section we introduce the mathematical model for the SPIR problem. We assume the production plan (production periods and quantities to produce) and the routing (which clients to visit in each period) are first-stage decisions. The quantities to deliver, the amount backlogged at each time period as well as the inventory levels are adjusted to the scenario. The goal of the stochastic approach is to find the solution that minimizes the production and the routing cost plus the expected cost of both the inventory and the penalty costs for backlogged demand. Following the SAA method, the true expected cost value is replaced by the mean value of a large random sample \( \Omega = \{\xi_1, \ldots, \xi_s\} \) of scenarios, obtained by the Monte Carlo method. This larger set of scenarios is regarded as a benchmark scenario set representing the true distribution [27].

Consider a graph \( G = (N, A) \) where \( N = \{0, 1, \ldots, n_N\} \) represents the set of nodes and \( A \) represents the set of possible links. Set \( A \) is a subset of set \( N \times N \). Node 0 denotes the producer and \( N_c = \{1, \ldots, n_N\} \) is the set of clients. Set \( T = \{1, \ldots, n_T\} \) is the set of periods and \( K = \{1, \ldots, n_V\} \) is the set of homogeneous vehicles available.

Consider the following parameters: \( d_{it}(\xi) \) is the demand of client \( i \in N_c \) in period \( t \in T \) in scenario \( \xi \in \Omega \), \( I_0/I_i \) is the initial stock at producer/client \( i \), \( \overline{S}_0/\underline{S}_i \) is the inventory capacity at producer/client \( i \), \( \overline{P} \) is the production capacity at each time period, \( Q_{it}^l \) and \( Q_{it}^u \) are the lower and upper limits to the delivery quantity in period \( t \in T \) at client \( i \) and \( L \) is the vehicle capacity. For the objective function, \( S \) is the set up cost for producing in a period, \( P \) is the production cost of a unit of product, \( V \) is the fixed vehicle usage cost, \( C_{ij} \) is the travel cost from node \( i \) to node \( j \), \( (i, j) \in A \), \( H_{it} \) is the holding cost of the product at node \( i \in N \) in period \( t \in T \), \( B_{it} \) is the backlog cost of the product at node \( i \in N \) in period \( t \in T \).

Consider the following first-stage variables: binary variables \( y_{it} \) are the setup variables that indicate whether there is production in period \( t \in T \), variables \( p_k \) give the production level in period \( t \in T \), binary variables \( z_{itk} \) indicate whether there is a visit to node \( i \in N_c \) in period \( t \in T \), by vehicle \( k \in K \), the routing variables \( x_{ijtk} \) indicate whether vehicle \( k \in K \) travels from node \( i \) to node \( j \), \( (i, j) \in A \), in period \( t \in T \), \( v_{it} \) is a binary variable that is one if vehicle \( k \in K \) is used in period \( t \in T \) and zero otherwise; for \( (i, j) \in A \), \( t \in T \), \( k \in K \), \( f_{ijtk} \) is the artificial flow of a single commodity variable used to prevent cycles in the routing problem. Notice that this is not the amount transported from node \( i \) to node \( j \) in period \( t \in T \) by vehicle \( k \) since that quantity depends on the scenario. Such adjustable variables could be used but would imply the use of an unnecessarily large model.

For each scenario \( \xi \in \Omega \), we define the second-stage variables \( q_{itk}(\xi) \) indicating the
quantity delivered at node \( i \in N_c \) in period \( t \in T \) by vehicle \( k \); \( s_{it}(\xi) \) indicating the stock level of node \( i \in N \) at the end of period \( t \in T \); and \( r_{it}(\xi) \) is the quantity backlogged at node \( i \in N_c \) in period \( t \in T \).

The SPIR problem is modeled by the following formulation.

The objective function (1) minimizes the total cost which includes the production setup, the production, the vehicle usage, the traveling costs, and the expected holding and backlog penalty costs. Constraints (2) and (3) are the inventory conservation constraints at the producer and at the clients, respectively. Constraints (4) establish the initial inventory level at the producer and at each client. Constraints (5) impose a maximum capacity at the producer and at each client. Constraints (6) impose a storage at the producer and at the clients, respectively. Constraints (7) impose a maximum capacity during each period of the SPIR problem. Constraints (8) ensure that the quantity delivered at node \( i \) in period \( t \) is within the stock level at node \( i \) in period \( t \). Constraints (9) ensure that the quantity delivered at node \( i \) in period \( t \) is within the maximum capacity at node \( i \) in period \( t \). Constraints (10) ensure that the sum of the quantities delivered at node \( i \) in period \( t \) is equal to the quantity delivered at node \( i \) in period \( t \) minus the quantity backlogged at node \( i \) in period \( t \). Constraints (11) ensure that the sum of the quantities delivered at node \( i \) in period \( t \) is equal to the quantity delivered at node \( i \) in period \( t \). Constraints (12) ensure that the quantity delivered at node \( i \) in period \( t \) is within the maximum capacity at node \( i \) in period \( t \). Constraints (13) ensure that the quantity delivered at node \( i \) in period \( t \) is within the maximum capacity at node \( i \) in period \( t \). Constraints (14) ensure that the quantity delivered at node \( i \) in period \( t \) is within the maximum capacity at node \( i \) in period \( t \). Constraints (15) ensure that the quantity delivered at node \( i \) in period \( t \) is within the maximum capacity at node \( i \) in period \( t \). Constraints (16) ensure that the quantity delivered at node \( i \) in period \( t \) is within the maximum capacity at node \( i \) in period \( t \).
on the production at each period. Constraints (7) impose limits on the delivery quantity at each client in each period. The lower bound imposed ensures that for each scenario the delivered quantity is positive, otherwise there could be second-stage solutions with null delivery quantity. Constraints (8) establish a capacity on the quantity transported by a vehicle. Constraints (9) and (10) are the routing constraints. Constraints (11) guarantee that a vehicle leaves the producer when there are visits to clients. Constraints (12) are the flow balance constraints at clients ensuring that visits are satisfied and constraints (13) guarantee that there is flow between two nodes only if a vehicle travels between these two nodes. Constraints (14), (15) and (16) are the variable domain constraints.

The formulation (1)-(16) is denoted SPIRF.

3 Model tightening

The mixed integer model is weak in the sense that the integrality gap is usually large [2, 6]. In order to tighten the model, we discuss families of valid inequalities and the use of an extended formulation.

3.1 Extended formulation

In this subsection we present an extended formulation which results from adapting ideas from the extended formulations for related problems (see [32] for deterministic lot-sizing problems and [4, 37] for deterministic production-inventory-routing problems without backlog) to the production-inventory-routing problem with multiple scenarios.

One weakness of model SPIRF are the linking constraints (7), especially when the delivered quantities \( q_{ikt}(\xi) \) can be much smaller than the upper bounds \( Q_{it} \).

By decomposing the amount delivered at each client in a given time period into the net demand of the time period destination, tighter linking constraints can be developed.

Let \( nd_{it}(\xi) \) denote the net demand for client \( i \in N_c \) at time period \( t \in T \) when scenario \( \xi \in \Omega \) occurs. Such net demand represents the minimum amount of product that must be delivered to client \( i \) at period \( t \) in order to satisfy the client demand taking into account the initial stock level, and can be computed as follows:

\[
nd_{it}(\xi) = \max\{0, \sum_{\ell=1}^{t} d_{i\ell}(\xi) - I_i - \sum_{\ell=1}^{t-1} nd_{i\ell}(\xi)\}.
\]

Consider the new variables \( u_{it\ell}(\xi) \) indicating the fraction of the net demand of client \( i \in N_c \) at time period \( t \) that is delivered in time period \( \ell \) when scenario \( \xi \) occurs, for \( t \in T, \ell \in T \cup \{n^N + 1\}, \xi \in \Omega \). Notice that when \( \ell = n^T + 1 \), variables \( u_{it\ell}(\xi) \) represent an amount that is not delivered during the considered time horizon.

In order to tighten model SPIRF, the following set of constraints, that we call EF, can be added.
\[
\sum_{k \in K} q_{itk}(\xi) = \sum_{t \in T} nd_{it}(\xi) \ u_{it\ell}(\xi) \quad \forall i \in N_c, t \in T, \xi \in \Omega \tag{17}
\]

\[
u_{it\ell}(\xi) \leq \sum_{k \in K} z_{itk} \quad \forall i \in N_c, t, \ell \in T, \xi \in \Omega \tag{18}
\]

\[
r_{i,t\tau}(\xi) = \sum_{t \in T} nd_{it}(\xi) \ u_{i,t,n+1}(\xi) \quad \forall i \in N_c, \xi \in \Omega \tag{19}
\]

\[
\sum_{\ell \in T \cup \{n+1\}} u_{it\ell}(\xi) = 1 \quad \forall i \in N_c, t \in T, \xi \in \Omega : nd_{it}(\xi) > 0 \tag{20}
\]

\[
u_{it\ell}(\xi) \geq 0 \quad \forall i \in N_c, t \in T, \ell \in T \cup \{n+1\}, \xi \in \Omega \tag{21}
\]

Constraints (17) relate variables \(u_{it\ell}(\xi)\) to the delivery quantity variables \(q_{itk}(\xi)\) and compute the quantity that is delivered to client \(i\) at time period \(\ell\). Constraints (18) ensure that if some fraction of net demand for client \(i\) is delivered at time period \(\ell\), then client \(i\) must be visited in that time period. Constraints (19) compute the demand not satisfied. Constraints (20) guarantee that demand is satisfied by production and/or is unmet during the time horizon (when \(u_{i,t,n+1}(\xi) > 0\)). Constraints (21) define the new variables as nonnegative.

The linking constraints (18) of this extended formulation are tighter than the linking constraints (7) of the original model, leading to a tighter model. Even considering a small number of scenarios, the number of constraints and variables in (17)-(21) becomes too large, which means that it is computationally impractical to use such sets of variables and constraints with all scenarios. In Section 5.1 some strategies are proposed to overcome this problem.

3.2 Valid inequalities

From the underlying fixed-charge network flow structure of the SPIRF one can derive several families of valid inequalities [31]. Since the possible set of inequalities is large, we select several such families of inequalities based on preliminary computational tests and on the knowledge of its relevance to related problems. Some of these inequalities are adapted from deterministic lot-sizing problems [32].

The first family relates the delivered quantity \(\sum_{k \in K} \sum_{i \in N_c} q_{itk}(\xi)\) with the setups. If there is no set up in time period \(t\), then the delivered amount is bounded by the stock in the producer at the end of time period \(t - 1\). Otherwise, \(\sum_{k \in K} \sum_{i \in N_c} q_{itk}(\xi)\) is bounded by the total vehicles capacities \(nV_L\).

\[
\sum_{k \in K} \sum_{i \in N_c} q_{itk}(\xi) \leq nV_L y_t + s_{0,t-1}(\xi) \quad \forall t \in T, \xi \in \Omega \tag{22}
\]

To the best of our knowledge, inequalities (22) are new, even for the deterministic case (when only one scenario is considered). These inequalities are useful to cut linear solutions with a fractional value of \(y_t\). Their impact is greater when \(nV_L\) is small, which happens, for instance, when a single small capacity vehicle is used, see Section 5.1.

The following two families of inequalities link the inventory levels (stock and backlog) with the binary variables representing either setups or the visits to clients. We notice that
the structure of these families of inequalities is quite different from the ones discussed in [4] for a related problem since backlog is not considered there.

The first family is useful to cut fractional solutions when the aggregated value of the binary variables in the RHS of the inequalities is less than 1. This family of inequalities is derived from the uncapacitated lot-sizing with backlogging feasible set (see [32]) defined by inequalities (3), (4) and (7), for a client \( i \), period \( t \) and scenario \( \xi \):

\[
    r_{it}(\xi) \geq \left( \sum_{\ell=1}^{t} d_{i\ell}(\xi) - I_{i} \right)^{+} (1 - \sum_{k \in K} \sum_{\ell=1}^{t} z_{i\ell k}) , \quad \forall i \in N_c, t \in T, \xi \in \Omega \quad (23)
\]

where \( (x)^{+} = \max\{x, 0\} \).

These inequalities force the net demand (demand that cannot be satisfied by the initial stock) at client \( i \) until period \( t \) to be satisfied with backlog in case there is no delivery until period \( t \), that is, if \( \sum_{k \in K} \sum_{\ell=1}^{t} z_{i\ell k} = 0 \). Note that these inequalities are a particular case of the more general family \( s_{i,h-1}(\xi) + r_{it}(\xi) \geq \sum_{\ell=h}^{t} d_{i\ell}(\xi)(1 - \sum_{k \in K} \sum_{\ell=1}^{t} y_{\ell}) \) for \( h,t \in T, h \leq t \).

The second family of inequalities is derived from the uncapacitated lot-sizing with backlogging feasible set obtained by considering the aggregated demands, stocks and backlogs.

\[
    \sum_{i \in N} s_{i,h-1}(\xi) + \sum_{i \in N} r_{it}(\xi) \geq \sum_{\ell=h}^{t} d_{i\ell}(\xi) \left( 1 - \sum_{\ell=1}^{t} y_{\ell} \right) , \quad \forall h,t \in T, h \leq t, \xi \in \Omega \quad (24)
\]

Inequalities (24) ensure that the total demand from period \( h \) to period \( t \) must be satisfied either from stock (at the producer or at the clients) or from backlog at the clients if there is no production in that period.

For the particular case of \( h = 1 \), inequalities (24) can be written in the stronger format

\[
    \sum_{i \in N_c} r_{it}(\xi) \geq D(t,\xi) \left( 1 - \sum_{\ell=1}^{t} y_{\ell} \right) ,
\]

where \( D(t,\xi) = \sum_{i \in N_c} \sum_{\ell=1}^{t} \eta_{i\ell}(\xi) \), is the net demand until period \( t \) for scenario \( \xi \).

Next, we introduce a family of inequalities that can be derived by Mixed Integer Rounding (MIR) [30]. Given a simple mixed integer set

\[
    X = \{(S,Y) \in \mathbb{R}_{+} \times \mathbb{Z} \mid S + aY \geq b\},
\]

where \( a, b \in \mathbb{R}_{+} \) are arbitrary constants, the MIR inequality is given by

\[
    S \geq r(\lfloor b/a \rfloor - Y),
\]

where \( r = b - (\lfloor b/a \rfloor - 1)a \).

Constraint \( \sum_{\ell=1}^{t} y_{\ell} + \sum_{i \in N_c} r_{it}(\xi) \geq D(t,\xi) \) is a relaxation of the feasible solutions set and imposes that the total production capacity plus the backlogged demand must
cover the net demand of all clients until period $t$. Taking $S = \sum_{i \in N_c} r_{it}(\xi)$, $Y = \sum_{t=1}^{T} y_t$, $a = \overline{P}$, and $b = D(t, \xi)$, the following MIR inequality is obtained

$$\sum_{i \in N_c} r_{it}(\xi) \geq R_P(t, \xi) \left( \left\lceil \frac{D(t, \xi)}{\overline{P}} \right\rceil - \sum_{t=1}^{T} y_t \right), \quad \forall t \in T, \xi \in \Omega$$

(25)

where $R_P(t, \xi) = D(t, \xi) - \left( \left\lceil \frac{D(t, \xi)}{\overline{P}} \right\rceil - 1 \right) \overline{P}$. Inequalities (25) state that the total setup capacity plus the backlogged demand cover the net demand of all clients until period $t$.

From preliminary computational experience, we observed that the number of inequalities cutting off the linear relaxation solutions is too large. Hence some strategies need to be taken in order to choose which cuts should be included. Two main strategies, Option (W) and Option (A), that differ in the way the scenarios are considered, were tested.

(W) **Worst case scenario.** In this option, for each family of inequalities and each set of time period and/or client indices (accordingly to the family) at most one inequality is added. The inequality added is the one that, of all the scenarios in $\Omega$, corresponds to the scenario leading to the largest violated inequality.

(A) **Aggregation of scenarios.** In this option, for each family of inequalities and each set of time period and/or client indices (accordingly to the family), the set of inequalities is aggregated for all the scenarios in $\Omega$. It adds the inequality aggregating all the scenarios leading to a violated inequality.

Option (W) tends to add too many cuts since, for a given fractional first-stage solution, the value of the second-stage solution variables, variables $s_{it}(\xi)$ and $r_{it}(\xi)$, tends to be null. Moreover, inequalities for a given scenario do not necessarily affect the solution variables of the remaining scenarios (unless they force the values of the first-stage solutions to change). Option (A) tends to add fewer cuts but these cuts are weaker than the individual inequalities for each scenario. This option resembles the scenario-group cuts approach proposed by Adulyasak et al. [4] where groups of scenarios are considered accordingly to the total demand, and Benders cuts are aggregated for each group. Option (A) corresponds to the case where only one group is created which, in the general framework we are proposing, is reasonable since the number of scenarios considered is small.

In Section 5 we report results derived from the tests conducted to assess the performance of these two options.

4 Solution approaches

In this section we discuss several algorithms to solve the SPIR problem. These algorithms are based on the SPIIRF model which is a two-stage stochastic model.

A common approach to solve stochastic problems is the Sample Average Approximation (SAA) method [28, 43], where the true expected value of the objective function is approximated by the average value obtained for a very large sample of $s$ scenarios. The SAA method generates $M$ separate sample sets $\Omega_m, m \in \{1, \ldots, M\}$, each one containing $\ell \ll s$ scenarios. For each one of the scenarios set, $\Omega_m$, the resulting SAA problem
(where $\Omega$ is replaced by $\Omega_m$ in model SPIRF) is solved generating a candidate solution. Then, for each candidate solution, the first-stage solution is fixed, and the value of the objective function for a very large sample with $s$ scenarios is computed. In the case of the two-stage model SPIRF, this value is computed by solving a pure linear programming problem on the second-stage variables.

Solving the SPIR problem to optimality, even for a very small size scenarios sample, may not be practical, given the difficulty of the problem. Therefore the $M$ subproblems generated by the SAA method can hardly be solved to optimality. Hence, the candidate solutions will be feasible solutions obtained with heuristics.

Next, we describe two approaches to solve the SPIR problem. The first one follows the SAA method and uses heuristics to obtain the $M$ candidate solutions. Then the best one is chosen. This will be named the static approach. The second approach uses the candidate solutions to fix part of the first-stage solution and then solves (heuristically) the resulting restricted problem for a larger sample. Since the final solution can be adjusted to the larger sample we name it the adjustable approach.

4.1 Static approach

A first heuristic consists in solving the SPIRF model by branch-and-bound with a commercial solver and imposing a running time limit. In fact, as we can observe in the computational section, this is a natural option that works well on the easiest instances where the solver can find near optimal solutions within reasonable time. For those cases, imposing a time limit does not have great impact on the quality of the solutions. Since it is well known, a large amount of running time is used to prove optimality, which is not a major relevant issue here given that the solution will be evaluated with another sample.

When the instances are harder to solve, the previous approach can perform quite badly and the quality of the solutions varies from sample to sample, leading to a large variability in the solution values. In order to circumvent this fault we used two heuristic approaches that can be taken separately or combined. One approach is to use an iterative local search method to improve the quality of the candidate solutions. The other approach is to simplify the model by pre-defining a route, making the restricted model easier to solve.

Iterative Local Search method

The quality of any feasible solution can be improved using a Local Search procedure. The local search scheme applies a local branching method based on the method proposed by Fischetti and Lodi [21] for reducing the solution space.

For a given positive integer parameter $\Delta_z$, define the neighborhood $\mathcal{N}(\bar{z}, \Delta_z)$ of $\bar{z}$ as the set of feasible solutions of the SPIRF satisfying the additional local branching constraint:

$$\sum_{i \in N, t \in T, k \in K} z_{itk} + \sum_{i \in N, t \in T, k \in K} (1 - z_{itk}) \leq \Delta_z.$$

Hence, the neighborhood of $\bar{z}$, $\mathcal{N}(\bar{z}, \Delta_z)$, is the set of solutions that differ by a maximum of $\Delta_z$ values from the $z_{itk}$ variables of the current solution $\bar{z}$. In fact, the linear
constraint (26) limits to $\Delta z$ the total number of binary variables $z_{itk}$ flipping their value with respect to the solution $\overline{z}$, either from 1 to 0 or from 0 to 1.

The Local Branching heuristic amounts to adding constraint (26) to SPIRF and running the solver for a time limit of $\alpha$ seconds. If a better solution is found, then $\overline{z}$ is updated and the Local Branching is run again. The complete heuristic is denoted by Iterated Local Search (ILS) and is described in Algorithm 1.

**Algorithm 1: Iterated Local Search.**

1: Solve the SPIRF model, considering a single scenario, for $\alpha$ seconds
2: Save the solution $\overline{z}$
3: Add the remaining scenarios to the SPIRF model
4: repeat
5: Add constraint (26) to the SPIRF model
6: Solve the model for $\beta$ seconds
7: Update the solution $\overline{z}$
8: until No improvement in the objective function is observed

**Using routing precedences from a defined route**

The SPIRF includes the TSP as a subproblem. For each time period, the minimum cost hamiltonian cycle that includes all the visited clients must be determined. The TSP is a NP-hard problem and is often very time-consuming. Combining the TSP formulation with production and inventory makes the model SPIRF quite large. In order to simplify the model, a TSP is initially solved for all clients. The order in which clients are visited on this route is used to define a precedence relation between the clients. These precedences are used to eliminate (set to zero) those variables that violate the precedence relation. Similar heuristic approaches have been used before for deterministic problems [38, 40].

The TSP involving all the clients is solved using the Concorde software [18].

We name this simplification procedure as the Precedence order relation (P) technique.

**4.2 Adjustable approach**

In this subsection we describe the adjustable sample average approximation (ASAA) heuristic. Since this heuristic can easily be extended to other two-stage stochastic problems we introduce it as a general framework procedure.

In contrast with the SAA method, where a set of $M$ candidate solutions are generated by solving $M$ SAA problems and the corresponding first-stage solutions are fixed, we propose a new approach that uses the $M$ candidate solutions to generate a partial solution. The idea is to identify within the first-stage variables those that have the same value in all (or almost all) the $M$ generated solutions. Thus, it is likely that this value coincides with the variable value in the optimal solution. By fixing the value of those first-stage variables the problem is simplified and the solution can be completed by solving a problem with a new larger sample.

It is important that the $M$ candidate solutions result from different solution structures, so that the variable fixing corresponds to common structures under different sample scenarios. Hence, the number of scenarios considered in each sample can be small which
facilitates the generation of the $M$ candidate solutions. Our experience (not reported in the computational tests) show that for the SPIR problem working with only one scenario leads to solutions that are quite different and few first-stage variables are fixed. Thus the number of scenarios should not be too small.

Another important ingredient is to decide which variables to fix. If a binary first-stage variable has the same value in all the candidate solutions it is natural to fix it to that value. However, we may wish to fix variables that have the same value in at least a given percentage of solutions or satisfy another criteria. Moreover, we may wish to assign different weights to each candidate solution either depending on the probability (if the solution is obtained for a scenario with higher probability its weight could be higher) or on its objective function value (a solution with a better objective function value than the others would be given a greater weight).

Hence, we define a function $f_I : \{0, 1\}^M \rightarrow [0, 1]$ that, for particular values $\omega_1, \ldots, \omega_M$ of a first-stage binary variable $\omega$, assigns a value $f_I(\omega_1, \ldots, \omega_M)$ between 0 and 1. The index $I$ represents the vector of instance parameters. Desirably, function $f_I$ should satisfy the following condition:

\textbf{Condition 1:} $f_I(0^M) = 0$ and $f_I(1^M) = 1$.

Vectors $0^M$ and $1^M$ represent the null vector and the vector with all components equal to 1 of dimension $M$, respectively. Condition 1 ensures that if a variable $\omega$ is zero (one) in all the candidate solutions, then the value of the function will be zero (one).

Let $\Xi$ be a subset of binary first-stage decision variables that makes the problem easier to solve when their values are fixed. Thus, this new method can be described in four steps.

**First step: solve $M$ independent problems.** Generate $M$ separate samples $\Omega_m, m \in \{1, \ldots, M\}$, of dimension $\ell_1$. Then, solve the two-stage stochastic model for each sample set, $\Omega_m$, with a time limit of $\alpha$ seconds (either a feasible solution is found within this time limit or it stops after the first feasible solution is found). This gives $M$ candidate solutions, $\varphi_1, \ldots, \varphi_M$.

**Second step: obtain $\hat{M}$ partial solutions.** Use the candidate solutions $\varphi_1, \ldots, \varphi_M$ to obtain partial candidate solutions by fixing the value of some first-stage variables in set $\Xi$ and keeping all the remaining variables free.

Let $\varphi^P_m$ denote the projection of the solution vector $\varphi_m$ into the space of variables in $\Xi$. Define two threshold values, $\gamma_1$ and $\gamma_2$ and generate a new partial solution in the following way: denote by $(\varphi^P_m)_\tau$ the value of a generic first-stage decision variable $\tau$ from set $\Xi$ in its partial solution $\varphi^P_m$, with the value of this variable in the generated partial solution, $\psi^P$, computed as follows

$$(\psi^P)_\tau = \begin{cases} 0, & f_I((\varphi^P_1)_\tau, \ldots, (\varphi^P_M)_\tau) < \gamma_1; \\ 1, & f_I((\varphi^P_1)_\tau, \ldots, (\varphi^P_M)_\tau) \geq \gamma_2. \end{cases}$$

Either considering $\hat{M}$ different functions $f_I$ or varying the parameters $\gamma_1$ and $\gamma_2$, $\hat{M}$ different partial solutions $\psi^{P1}, \ldots, \psi^{P\hat{M}}$ can be obtained.
Third step: complete the $\tilde{M}$ partial solutions. For each partial candidate solution $\psi^p_1, \ldots, \psi^p_M$ constructed and having some of the first-stage variables fixed, solve the two-stage stochastic model using a larger sample set with $\ell_2 \geq \ell_1$ scenarios. Use the optimal solution obtained and fix to its optimal value the first-stage variables not fixed yet. Repeating the process for each partial candidate solution yields $\tilde{M}$ candidate solutions with all the first-stage variables fixed.

Fourth step: solve $\tilde{M}$ simplified problems. With all the first-stage decision variables fixed, the two-stage stochastic model is solved for a very large sample set with $\ell_3 \gg \ell_2$ scenarios. The value of the recourse variables (the second-stage variables) is computed for each scenario and, consequently, the objective function value of each solution is computed. The solution with the lowest average cost is chosen.

The second and the third steps correspond to the adjustable part of the procedure. If we let $M = \tilde{M}$ and remove the second and the third steps, the adjustable approach ASAA coincides with the SAA method.

In the computational tests we considered the following aspects:

- Only one partial solution is derived, that is, $\tilde{M} = 1$.
- The binary first-stage variables considered in set $\Xi$ are the $z_{ijk}$ variables.
- Function $f_I$ is defined as follows: $f_I((\varphi^p_1)_\tau, \ldots, (\varphi^p_M)_\tau) = \sum_{m=1}^{M} w_m(\varphi^p_m)_\tau$ where the weight $w_m$ of each solution $\varphi_m$ is computed according to the corresponding objective function value. Let $c^m$ denote the objective function value of solution $\varphi_m$ (with first-stage variables fixed) computed on a larger sample of dimension $\ell^* \geq \ell_1$. Normalize these weights as follows

$$\bar{c} = \max_{k=1, \ldots, M} \{e^k\} \quad \bar{d} = \min_{k=1, \ldots, M} \{e - c^k | c^k \neq \bar{c}\}.$$

Then, for each solution $\varphi_m$, define its weight as

$$w_m = \frac{\bar{e} + \bar{d} - c^m}{\sum_{k=1}^{M} (\bar{e} + \bar{d} - c^k)}.$$

**Proposition 1:** The function $f_I$ defined above satisfies Condition 1.

**Proof:** Computing the value of the function $f_I$ for the vectors $0^M$ and $1^M$ we have

$$f_I(0, \ldots, 0) = \sum_{m=1}^{M} w_m \times 0 = 0$$

$$f_I(1, \ldots, 1) = \sum_{m=1}^{M} w_m = \sum_{m=1}^{M} \left( \frac{\bar{e} + \bar{d} - c^m}{\sum_{k=1}^{M} (\bar{e} + \bar{d} - c^k)} \right) = \frac{\sum_{m=1}^{M} (\bar{e} + \bar{d} - c^m)}{\sum_{k=1}^{M} (\bar{e} + \bar{d} - c^k)} = 1$$

- In the third step, the model for a large sample is solved by relaxing the routing variables, and all the first-stage variables are fixed, except the routing variables. Then, defining the routing variables as binary, a TSP is solved for each vehicle and each time period.
5 Computational experiments

This section reports the computational experiments carried out to illustrate the performance of the methods described to solve the SPIR problem. All tests were run using a computer with an Intel Core i7-4750HQ 2.00GHz processor and 8GB of RAM, and were conducted using the Xpress-Optimizer 28.01.04 solver with the default options. The performed experiments are based on the instances introduced in [6] which are generated as follows.

The coordinates of the \( n \) clients are randomly generated in a 100 by 100 square grid, and the producer is located in the center of the grid. For each value of \( n \) a complete graph \( G = (N,A) \) is obtained and symmetric traveling costs are associated to the set of arcs. The traveling costs \( C_{ij} \) are the euclidean distance between nodes \( i \) and \( j \) in the grid.

The computational experiments used to test the improvements of both the valid inequalities and the extended formulation are based on instances with \( n^N = 5,15 \) and 25 clients. For the computational experiments conducted to assess the performance of the ASAA, the number of clients used is \( n^N = 10,20,30,40,50,60,70,80 \). In both cases, \( n^T = 5,10 \) time periods are considered.

For each client and each period, the nominal demand value \( d_{it} \) is randomly generated between 40 and 80 units, and the uncertain demands vary in \([0.7d_{it},1.3d_{it}]\).

The initial stock \( I_0 \) at the producer is randomly generated in the interval \([0,30]\), and the initial stock \( I_i \) at client \( i \) is randomly generated between 0 and three times the average demand of client \( i \). The maximum inventory level \( S_i \) is 500 for all \( i \in N \). The production capacity \( P \) is 80% of the average net demand. The number \( n^V \) of available homogeneous vehicles is one and two and their capacity \( L \) is set to 80% and to 40% of the average net demand, respectively. The lower \( Q_{it} \) and upper \( \bar{Q}_{it} \) delivery limits are 1 and \( L \) (the vehicles capacity), respectively. The production set up cost, the unit production cost and the fixed vehicle usage cost are given by \( S = 100, P = 1 \) and \( V = 50 \), respectively. For all \( i \in N, t \in T \), the holding cost \( H_{it} \) is 0.2 and the backlog cost \( B_{it} \) is 0.5, except for \( t = n^T \) where \( B_{i,n^T} \) is 5, since this cost penalizes the demand of client \( i \) that is not satisfied during the planning horizon.

5.1 Model enhancements

In this subsection we report the computational experiments conducted to test the model enhancements, namely the extended formulation and the valid inequalities. Based on these results the final model is chosen.

Inclusion of the extended formulation

First, we evaluate the improvements observed when the extended formulation EF, described in Subsection 3.1, is used. Several strategies can be used to deal with the scenarios in the extended formulation. On the one hand, when the number of scenarios used is large the size of the formulation becomes too large. On the other hand, by considering few scenarios the model becomes weaker. To evaluate advantages and disadvantages of the use of the extended formulation, four strategies were studied:
(i) SPIRF, only SPIRF is used;

(ii) SPIRF+EF, SPIRF is used together with EF considering all the scenarios;

(iii) SPIRF+EF₁, SPIRF is used together with EF for the scenario with the lowest value of total net demand, \( \xi^l = \arg\min_{\xi \in \Omega} \{\sum_{i \in N_c} \sum_{t \in T} nd_t(\xi)\} \);

(iv) SPIRF+EF₂, SPIRF is used together with EF for the scenario with the greatest value of total net demand, \( \xi^g = \arg\max_{\xi \in \Omega} \{\sum_{i \in N_c} \sum_{t \in T} nd_t(\xi)\} \).

We consider instances, as described above, with \( n^N = 5, 15 \) and \( 25 \) clients, \( n^T = 5, 10 \) periods and a single vehicle. For each instance, 10 scenarios were considered and the time limit was set to 600 seconds.

Table 1 summarizes the computational experience. For each one of the four strategies and for each instance, identified by the number \( n^T \) of periods and the number \( n^N \) of clients, Table 1 displays the values of the linear programming relaxation (LR), the lower bound (LB), the upper bound (UB) and two gaps,

\[
\text{Gap}_1 = \frac{\text{BFS} - \text{LB}}{\text{BFS}} \times 100 \quad \text{and} \quad \text{Gap}_2 = \frac{\text{UB} - \text{BFS}}{\text{BFS}} \times 100
\]

where BFS is the value of the best feasible solution obtained among the four strategies mentioned. The values of LB and UB are obtained at the end of the running time.

| \( n^T \) | \( n^N \) | SPIRF | SPIRF+EF | SPIRF+EF₁ | SPIRF+EF₂ |
|---|---|---|---|---|---|
| 5 | 15 | 2196 | 3076 | 3155 | 2.5 | 0.0 | 2275 | 3061 | 3158 | 3.0 | 0.1 |
| 5 | 25 | 5105 | 5841 | 6265 | 6.8 | 0.0 | 5183 | 5682 | 6420 | 9.3 | 2.5 |
| 5 | 10 | 8982 | 9537 | 11395 | 12.9 | 4.1 | 9085 | 9464 | 20972 | 13.6 | 91.6 |
| 5 | 15 | 5001 | 6759 | 6961 | 2.9 | 0.0 | 5026 | 6546 | 6965 | 6.0 | 0.1 |
| 5 | 25 | 13719 | 14884 | 16821 | 11.5 | 0.0 | 13750 | 14625 | 28887 | 13.1 | 71.7 |
| 5 | 10 | 20724 | 21332 | 47110 | 54.7 | 0.0 | 20750 | 21335 | 48339 | 54.7 | 2.6 |

Table 1: Computational comparison of the four strategies regarding the inclusion of the extended formulation EF.

As expected, the formulation SPIRF+EF provides the best quality linear relaxation solutions, however the computational time is higher, and the lower and upper bounds (and consequently the gaps \( \text{Gap}_1 \) and \( \text{Gap}_2 \)), are the worst. The differences are very small between strategies SPIRF, SPIRF+EF₁ and SPIRF+EF₂. Therefore, no clear conclusions can be drawn. Although SPIRF+EF₁ provides better results for small size instances, the results for \( n^N = 25, n^T = 10 \) seem to indicate that for the larger instances
model SPIRF is better. Also, the gains in terms of the linear relaxation value indicate that the improvements obtained by including the complete model EF are minor when the size of the model increases.

Hence, since there is no clear benefit in the use of any one of the strategies SPIRF + EF, SPIRF + EF$_l$ and SPIRF + EF$_g$, mainly for the largest instances, and since the number of constraints and variables tend to increase rapidly as the size of the instances increase, in what follows the extended formulation is not included.

Inclusion of valid inequalities

Now, we assess the improvements observed when SPIRF is tightened with the inclusion of the valid inequalities described in Subsection 3.2.

Since inequalities (22) are new, an independent experiment was conducted on these inequalities for instances with $n^T = 5$ periods, $n^N = 10$ clients and $n^V = 1, 2$ vehicles. The linear gap of SPIRF ($Gap_L$) and the linear gap of SPIRF tightened with inequalities (22) ($Gap_T$) were computed for $L$, the vehicles capacity, which was set to 20%, 40%, 60% and 80% of the average net demand. The average gap reduction obtained by the inclusion of the inequalities, computed by $\frac{Gap_L - Gap_T}{Gap_T} \times 100$, was 27.5, 20.6, 10.6 and 11.1, respectively, for one vehicle, and 26.2, 17.8, 1.4 and 0, respectively, for two vehicles. Thus, the impact of these inequalities is greater for small values of $L$ and for one vehicle.

The valid inequalities (22)-(25) are dynamically added to the SPIRF and three strategies are compared:

(i) SPIRF, only SPIRF is used, no inequalities are included.

(ii) SPIRF-W, corresponds to Option (W) for the inclusion of valid inequalities discussed in Section 3.2.

(iii) SPIRF-A, this strategy corresponds to Option (A) applied to inequalities (22), (23) and (24) and Option (W) applied to inequalities (25). Notice that Option (A), aggregation of the inequalities for all the scenarios, applied to inequalities (25) leads to very weak inequalities.

Twelve sets of instances were used, six considering one vehicle and six considering two vehicles. For each set, five instances were generated considering different samples of demand values. A time limit of 600 seconds was imposed. The obtained results are displayed in Table 2, Figure 1 and Figure 2.

Table 2 displays the results for the three strategies: SPIRF, SPIRF-A, and SPIRF-W. Each line has the results for an instance set identified by its number $n^V$ of vehicles, its number $n^T$ of periods, and its number $n^N$ of clients. Columns identified with $\#vi$ display the average number of valid inequalities added to the SPIRF. Columns identified with $\#nodes$ display the average number of nodes solved within the branch-and-bound algorithm. Columns named Gap display the average gap, computed according to $\frac{UB - LB}{LB} \times 100$, where $UB$ and $LB$ denote the average upper bound value and the average lower bound value, respectively.

Table 2 shows that the average number of inequalities added to the SPIRF in strategy SPIRF-A is lower than the average number of inequalities added to the SPIRF in strategy
SPIRF-W. Also the number of nodes solved is lower. Columns Gap show that the average gap is reduced by the inclusion of valid inequalities and smaller average gaps are always obtained with strategy SPIRF-A. It is also possible to observe that instances with more than one vehicle are more difficult to solve, even for small size instances.

Figure 1 and Figure 2 display the relation between the average linear relaxation, average lower bound, and average upper bound values for the three strategies, for each instance set. Since these values can vary considerably from instance to instance, for better comparison each value was divided by the best (lower) upper bound of the corresponding instance set.

Figure 1 depicts results for the single-vehicle case, while Figure 2 depicts results for the case where two vehicles are available. In both figures, the values obtained by using approaches SPIRF-A and SPIRF-W are associated with the solid and dashed lines, respectively. The dotted lines are associated with SPIRF, the case where no inequalities are added. The circle points (in the first/down set of three lines) represent the ratio between the average linear relaxation and the best upper bound value. Solid points (in the second/middle set of three lines) represent the ratio between the average lower bound and the best upper bound value. Square points (in the third/upper set of three lines) represent the ratio between the average upper bound and the best upper bound value.

Figure 1 and Figure 2 help to explain the gaps computed in Table 2, since the largest average lower bound and smallest average upper bound values are obtained with strategy SPIRF-A, except in one case. Note that, for the three approaches, the average lower bound values are very close.

Since the best results are obtained with strategy SPIRF-A, in what follows, the SPIRF will be used together with the inclusion of inequalities following strategy SPIRF-A.

| $n^V$ | $n^T$ | $n^N$ | SPIRF | SPIRF-A | SPIRF-W |
|-------|-------|-------|-------|---------|---------|
|       |       |       | $\#nodes$ | Gap | $\#vi$ | $\#nodes$ | Gap | $\#vi$ | $\#nodes$ | Gap |
| 5     | 5     | 15    | 2482  | 0.0  | 18     | 2464  | 0.0  | 50     | 2812  | 0.0  |
| 25    | 5     | 15    | 44696 | 2.4  | 65     | 19118 | 1.5  | 82     | 38999 | 1.5  |
| 1     | 10    | 15    | 6772  | 5.9  | 108    | 4039  | 4.1  | 185    | 5885  | 4.5  |
| 5     | 25    | 1      | 35385 | 0.0  | 28     | 30656 | 0.0  | 125    | 30909 | 0.0  |
| 10    | 25    | 1      | 9185  | 2.1  | 90     | 4891  | 1.9  | 215    | 7971  | 2.2  |
| 5     | 25    | 1      | 89    | 11.4 | 210    | 2213  | 7.7  | 299    | 591   | 8.5  |
| 5     | 15    | 5      | 119675| 4.9  | 23     | 82384 | 3.7  | 55     | 129334| 4.7  |
| 25    | 5     | 15    | 2945  | 34.8 | 76     | 7709  | 26.6 | 110    | 7785  | 28.2 |
| 1     | 10    | 15    | 925   | 40.8 | 139    | 1326  | 31.3 | 174    | 724   | 35.7 |
| 2     | 10    | 5     | 10730 | 7.0  | 40     | 12201 | 6.4  | 132    | 17016 | 6.6  |
| 25    | 10    | 15    | 44    | 121.0| 110    | 1792  | 54.2 | 215    | 2147  | 97.8 |
| 25    | 10    | 1      | 74    | 40.3 | 233    | 4079  | 26.6 | 287    | 2695  | 28.9 |

Table 2: Computational results for the three strategies: SPIRF, SPIRF-A and SPIRF-W.
Figure 1: Comparison of average linear relaxation, average lower bound, and average upper bound values for instances using 1 vehicle.

(a) Results for instances with 5 time periods  
(b) Results for instances with 10 time periods

Figure 2: Comparison of average linear relaxation, average lower bound, and average upper bound values for instances using 2 vehicles.

(a) Results for instances with 5 time periods  
(b) Results for instances with 10 time periods
5.2 Computational comparison of the static and adjustable strategies

Extensive computational experiments to assess the performance of the proposed Adjustable Sample Average Approximation method are presented. Three variants of the ASAA, that differ from each other in the strategy used to obtain the initial candidate solutions, are compared. These three variants of the ASAA method are:

(i) $ASAA_{ILS}$, the ILS procedure is used;
(ii) $ASAA_{ILS+P}$, the ILS procedure is used and precedence relations in the routing are considered;
(iii) $ASAA_R$, the routing variables $x_{ijtk}$ are relaxed.

These three variants were tested against the following four static approaches:

(i) $SAA$, the SAA method;
(ii) $SAA_{ILS}$, the SAA with the ILS procedure;
(iii) $SAA_{ILS+P}$, the SAA with the ILS procedure in which precedence relations in the routing are considered;
(iv) $EVP$, the Expected Value Problem (EVP), which corresponds to solving the deterministic problem with all the uncertainty parameters replaced by their expected values.

Whenever the ILS procedure is used, the initial scenario considered is the one that corresponds to the EVP. However, since we are dealing with difficult instances, usually the corresponding EVP problem cannot be solved to optimality within a reasonable amount of running time. Thus a two-stage heuristic procedure to obtain a feasible solution is followed. First, the EVP with the routing variables $x_{ijtk}$ relaxed is solved with a time limit of 300 seconds. Then, all the visit variables $z_{itk}$ are fixed and $n^T$ pure TSP problems are solved by using the software Concorde. Even when the number of visit variables that are allowed to change their value in the local branching constraint (26) is low, the time used to solve each subproblem is large, since all the routing variables remain free. Hence, in order to solve such subproblems faster, we used a local branching constraint similar to (26) for the routing variables $x_{ijtk}$, in which the number of variables that are allowed to flip their value is three times the value used in the local branching constraint associated to the visit variables.

To evaluate the performance of these seven strategies to solve the SPIRF, 32 instances, randomly generated as described before, are used. For each instance and each strategy, except for strategy named $EVP$ (corresponding to solve the EVP), the number of candidate solutions used is $M = 10$ and each candidate solution is obtained by using a sample of $\ell_1 = 10$ scenarios and a time limit of 300 seconds. However, when no integer solution is found within this time limit, the procedure only stops when the first integer solution is obtained. For strategy $EVP$, only one candidate solution is obtained using a single scenario, with a time limit of 3600 seconds (one hour). For all strategies, the dimension of the final sample used is $\ell_3 = 1000$.

For the ASAA variants, a sample of $\ell^* = 25$ scenarios is used to define the weight of each partial candidate solution in the second step and several strategies for fixing
variables $z_{ik}$ are used, corresponding to different choices of the parameters $\gamma_1$ and $\gamma_2$. The threshold values 0, 0.05, 0.15 and 0.25 were tested for $\gamma_1$, while $\gamma_2$ is set to 1 - $\gamma_1$. In the third step, when solving the simplified model, a sample of $\ell_2 = 50$ scenarios is used.

The obtained results are displayed in Table 3 and Table 4. Each strategy is identified in the first line of the table and each instance is identified by its number $n^V$ of vehicles, its number $n^T$ of periods and its number $n^N$ of clients in the first three columns.

In Table 3 the computational time (in seconds) required by each approach to obtain the final solution is displayed in columns named Time. For the adjustable approaches, the computational time values displayed correspond to the time obtained for $\gamma_1$, the one which required more computational time. In columns VSS an approximation of the Value of the Stochastic Solution is displayed for each approach considered. Notice that, as explained before, it is not possible to obtain the exact value of the Expected Value Problem, thus all the values are computed according to the heuristic solution obtained for the EVP. For the adjustable approaches, the approximation for the value of the optimal solution is computed based on the best solution obtained for the different threshold values of $\gamma_1$.

| $n^V$ | $n^T$ | $n^N$ | EVP-Time | SAA-Time | SAA-ILS-Time | ASAA-ILS-Time | SAA-ILS-P-Time | ASAA-ILS-P-Time | VSS-Time |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 4 | 1 | 11 | 2 | 525 | 2 | 138 | 2 | 173 | 2 | 138 | -210 | 86 |
| 20 | 4 | 1 | 3014 | 15 | 3142 | 20 | 1213 | 30 | 1080 | 10 | 963 | 22 | 839 | -120 | 1824 |
| 30 | 3030 | -908 | 3000 | 5 | 1599 | 28 | 1311 | -1 | 1600 | 32 | 1319 | -308 | 3020 |
| 50 | 3054 | -412 | 3544 | -169 | 1908 | 35 | 1592 | 28 | 1879 | 61 | 1540 | -212 | 891 |
| 70 | 3082 | -434 | 3818 | 60 | 2640 | 102 | 1628 | 103 | 2438 | 103 | 1628 | -71 | 1836 |
| 90 | 3076 | -4377 | 4580 | 644 | 3017 | 989 | 1539 | 785 | 3209 | 1034 | 1606 | 1110 | 3060 |
| 100 | 3080 | -3485 | 5035 | 1871 | 3328 | 1007 | 2512 | 2360 | 3329 | 2400 | 3019 | 1586 | 3050 |
| 150 | 3090 | -3485 | 5035 | 1871 | 3328 | 1007 | 2512 | 2360 | 3329 | 2400 | 3019 | 1586 | 3050 |
| 200 | 3090 | -3485 | 5035 | 1871 | 3328 | 1007 | 2512 | 2360 | 3329 | 2400 | 3019 | 1586 | 3050 |

Table 3: Computational time and approximation of the Value of the Stochastic Solution for all the approaches.

The computational times presented in Table 3 for all the approaches are very close and, in general, the lowest ones correspond to the approaches in which precedence relations are used. Furthermore, the additional computational time required to improve the solution in the adjustable approaches is lower than the time required to evaluate all the 10 candidate solutions in the final step of the corresponding static approaches.

The cost of the solutions obtained by each one of the strategies is reported in Table 4.

21
In each set of four columns associated with each variant of the ASAA, the results for the different threshold values of $\gamma_1$ are reported.

The results displayed in Table 4 show that the cost of the solutions obtained by SAA tend to be excessively high for medium and large size instances. However, for the smallest instances, the cost of the solutions obtained by strategy ASAA$_R$, where the structure of the problem changes due to the relaxation of the routing variables, is even higher than the one obtained by SAA.

The solutions obtained by the static strategies that use the ILS procedure, strategies SAA$_{ILS}$ and SAA$_{ILS+P}$, are in general better than the ones obtained by solving the EVP. Furthermore, such solutions can be improved by using the corresponding adjustable strategies, ASAA$_{ILS}$ and ASAA$_{ILS+P}$, mainly when the size of the instances increases. This means that the second and third steps in the adjustable approach improve the quality of the solutions. Notice that these two adjustable approaches iterate from the solutions obtained by the corresponding static version.

The solutions obtained by SAA$_{ILS+P}$ have, in general, a lower cost than the ones obtained by SAA$_{ILS}$ and, in some cases, there is a big difference between the cost of these solutions. The same observation applies when comparing the strategies SAA$_{ILS+P}$ and SAA$_{ILS}$. This means that the use of precedence relations can be very useful for such instances, mainly for the medium and large size instances, since fixing the routing variables \textit{a priori} make the instances easier to solve.

In strategy ASAA$_R$, where the routing variables are relaxed, integer solutions are found quickly. For small size instances, the solutions obtained by this variant are the ones with the worst cost. However, when the size of the instances increases, the computational time required to obtain solutions with the relaxation strategy is kept small and much better solutions than the ones obtained by SAA and EVP are obtained.

The computational results obtained for the adjustable approaches suggest that different good solutions can be obtained from different threshold values, and in some cases, different threshold values may result in the same solution. In fact, there is no specific threshold value that always leads to the best solution. For the large size instances, better results are obtained by considering threshold values different from zero, which corresponds to fixing more variables than the ones that take the same value in all the candidate solutions. However, the threshold value $\gamma_1$ used should not be close to 0.5, since the quality of the solution can be lost when many variables are fixed \textit{a priori}.

From a general point of view, the computational results presented in Table 3 and Table 4 allow us to conclude that in almost all the cases, the solutions obtained by the static approaches can be improved by the second and third steps of the adjustable approaches using a short running time. Hence, the proposed adjustable strategies prove to be effective and efficient, mainly for medium and large instances, since good solutions can be obtained in a short time. Moreover, among the three adjustable strategies tested, strategy ASAA$_{ILS+P}$ is the one that, in general, leads to the best results, solution and time.
| $n^V$ | $n^T$ | $n^N$ | $\text{EVP}$ | $\text{SAA}$ | $\text{SAA}_{ILS}$ | $\text{ASAA}_{ILS}$ | $\text{SAA}_{ILS}+P$ | $\text{ASAA}_{ILS}+P$ | $\text{ASAA}_R$ |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 5151 | 5149 | 5149 | 5149 | 5149 | 5149 | 5149 | 5149 | 5149 |
| 20 | 9069 | 9054 | 9049 | 9049 | 9049 | 9049 | 9049 | 9049 | 9049 |
| 30 | 12336 | 12544 | 12331 | 12331 | 12331 | 12331 | 12331 | 12331 | 12331 |
| 40 | 18468 | 18880 | 18637 | 18449 | 18454 | 18455 | 18411 | 18407 | 18407 |
| 50 | 22495 | 22949 | 22435 | 22303 | 22403 | 22428 | 22397 | 22397 | 22397 |
| 60 | 27308 | 47240 | 27113 | 27060 | 27075 | 27077 | 27077 | 27077 | 27077 |
| 70 | 31701 | 77458 | 31057 | 30740 | 30732 | 30732 | 30732 | 30732 | 30732 |
| 80 | 35152 | 84004 | 33281 | 33151 | 33151 | 33151 | 33151 | 33151 | 33151 |
| 10 | 12936 | 12954 | 12920 | 12920 | 12920 | 12920 | 12920 | 12920 | 12920 |
| 20 | 24010 | 24342 | 23947 | 23850 | 23850 | 23850 | 23850 | 23850 | 23850 |
| 30 | 35527 | 50540 | 34517 | 34438 | 34478 | 34508 | 34508 | 34508 | 34508 |
| 40 | 47017 | 131402 | 46549 | 46538 | 46538 | 46538 | 46538 | 46538 | 46538 |
| 50 | 59861 | 55949 | 56419 | 56419 | 56419 | 56419 | 56419 | 56419 | 56419 |
| 60 | 73463 | 190923 | 68477 | 67458 | 67394 | 67387 | 67387 | 67387 | 67387 |
| 70 | 81309 | 198496 | 75262 | 74030 | 73972 | 73914 | 73914 | 73914 | 73914 |
| 80 | 91870 | 253709 | 88506 | 85353 | 85355 | 85392 | 85389 | 85389 | 85389 |

**Table 4:** Computational results for comparison of several strategies for solving a two-stage stochastic model.
Results not reported here show that each subproblem associated to the determination of each candidate solution is highly influenced by the variation of the demand values. Hence, in order to validate our conclusions, further tests were conducted for the large size instances in which the demand samples are derived from a normal distribution with parameters $d_t$ (mean) and $0.15d_t$ (standard deviation), and for a gamma distribution with parameters $(1/0.15)^2$ (shape) and $1/(0.15^2d_t)$ (rate). For these results we compare the EVP, the best static approach $SAA_{ILS+P}$, and the best adjustable approach $ASAA_{ILS+P}$ using the threshold value 0.15. The results are reported in Tables 5 and 6 respectively.

### Table 5: Computational results obtained for the case where the demands follow the normal distribution.

| $n^V$ | $n^T$ | $n^N$ | **EVP** |  | **SAA_{ILS+P}** |  | **ASAA_{ILS+P}** |  |
|-------|-------|-------|----------|---|----------------|---|----------------|---|
|       |       |       | Time    | Cost | Time    | Cost | VSS    | Time    | Cost | VSS    |
| 60    | 5     | 1     | 3710    | 27285| 2361   | 26802| 483    | 1531   | 26590| 695    |
| 70    | 70    | 10    | 3748    | 31667| 2874   | 30442| 1225   | 1740   | 30204| 1463   |
|       | 80    | 100   | 3789    | 35110| 3207   | 32962| 2148   | 1745   | 32729| 2381   |
| 60    | 10    | 60    | 4014    | 84826| 6718   | 64477| 17379  | 3456   | 67170| 17656  |
| 70    | 70    | 80    | 10191   | 81235| 7215   | 73633| 7602   | 2809   | 73290| 7945   |
| 80    | 10    | 100   | 3973    | 96860| 6250   | 85318| 11542  | 3316   | 84981| 11879  |
| 60    | 5     | 60    | 3708    | 28017| 2534   | 27388| 629    | 1720   | 27237| 780    |
| 70    | 70    | 80    | 3745    | 33008| 3875   | 31139| 1869   | 2765   | 31014| 1994   |
| 80    | 10    | 100   | 3805    | 37015| 3999   | 33866| 3149   | 2409   | 33704| 3311   |
| 60    | 2     | 60    | 4021    | 71910| 6349   | 68873| 3037   | 3031   | 68527| 3383   |
| 70    | 70    | 80    | 10291   | 86527| 7821   | 75803| 10724  | 3415   | 75558| 10969  |
| 80    | 10    | 100   | 3985    | 25019| 5755   | 93426| 156772 | 2725   | 92152| 158046 |

### Table 6: Computational results obtained for the case where the demands follow the gamma distribution.

| $n^V$ | $n^T$ | $n^N$ | **EVP** |  | **SAA_{ILS+P}** |  | **ASAA_{ILS+P}** |  |
|-------|-------|-------|----------|---|----------------|---|----------------|---|
|       |       |       | Time    | Cost | Time    | Cost | VSS    | Time    | Cost | VSS    |
| 60    | 5     | 1     | 3722    | 26495| 2982   | 25958| 537    | 2056   | 25721| 774    |
| 70    | 70    | 10    | 3759    | 30734| 3935   | 29484| 1250   | 2713   | 29246| 1488   |
| 80    | 10    | 60    | 3802    | 34103| 4657   | 31891| 2212   | 3091   | 31625| 2478   |
| 60    | 10    | 80    | 3835    | 82949| 4101   | 65265| 17684  | 2271   | 64960| 17989  |
| 70    | 10    | 100   | 9933    | 78959| 4527   | 71249| 7710   | 2185   | 70858| 8101   |
| 80    | 10    | 60    | 3955    | 91075| 6303   | 82412| 8663   | 3513   | 82045| 9030   |
| 60    | 5     | 60    | 3707    | 27202| 3247   | 26570| 632    | 2441   | 26414| 788    |
| 70    | 70    | 80    | 3747    | 32083| 4333   | 30193| 1890   | 3207   | 30023| 2060   |
| 80    | 10    | 100   | 3791    | 35962| 4550   | 32809| 3153   | 3072   | 32587| 3375   |
| 60    | 2     | 60    | 3829    | 69788| 4772   | 66794| 2994   | 2990   | 66350| 3438   |
| 70    | 70    | 80    | 10034   | 84272| 6013   | 73420| 10852  | 3663   | 73712| 11100  |
| 80    | 10    | 100   | 3990    | 246986| 6014  | 90394| 156592 | 2944   | 89429| 157557 |

Table 5: Computational results obtained for the case where the demands follow the normal distribution.

Table 6: Computational results obtained for the case where the demands follow the gamma distribution.
Both tables show that the static and the adjustable approaches lead to gains when compared to the feasible solution obtained by solving the EVP. Although the running times are similar (which is expected since the first step in both approaches is the same and corresponds to computing the set of candidate solutions, which is the most time consuming step), the adjustable approach is always slightly faster. In relation to the quality of the feasible solution, the adjustable approach is always better than the static one.

6 Conclusion

A stochastic production-inventory-routing problem with recourse is considered. A sample average approximation method is applied where the stochastic problem is solved for several samples and the best solution among the solutions obtained is selected. Since the stochastic problem is very complex and problem instances can hardly be solved to optimality within reasonable amount of running time, two different approaches are proposed: a static approach that consists of using heuristics to generate a solution for the stochastic problem defined for each sample; and a new adjustable heuristic procedure where the generated solutions are used to fix part of the first-stage solution (corresponding to those variables which have common values in the most of the candidate solutions). The partial solution is completed by solving a new restricted problem defined for a larger sample.

Several algorithms were tested for each approach. Computational results have shown that the static procedure performs well on small size instances using as heuristic the (time) truncated branch-and-bound algorithm. For medium size instances the static procedure performs well if an iterated local search heuristic is used to improve the initial candidate solutions. However, such solutions can be improved by using the adjustable approach with a short computational time.

Future research would be to extend the adjustable approach introduced here to other complex two-stage stochastic problems, such as network design problems.

References

[1] N. Absi, C. Archetti, S. Dauzère-Pérès, and D. Feillet, A two-phase iterative heuristic approach for the production routing problem, Transp Sci 49 (2015), 784–795.

[2] Y. Adulyasak, J.F. Cordeau, and R. Jans, Formulations and branch-and-cut algorithms for multi-vehicle production and inventory routing problems, INFORMS J Comput 26 (2014), 103–120.

[3] Y. Adulyasak, J.F. Cordeau, and R. Jans, Benders decomposition for production routing under demand uncertainty, Oper Res 63 (2015), 851–867.

[4] Y. Adulyasak, J.F. Cordeau, and R. Jans, The production routing problem: A review of formulations and solution algorithms, Comput Oper Res 55 (2015), 141–152.
[5] E.H. Aghezzaf, Robust distribution planning for supplier-managed inventory agreements when demand rates and travel times are stationary, J Oper Res Soc 59 (2008), 1055–1065.

[6] A. Agra, A. Cerveira, and C. Requejo, Lagrangian relaxation bounds for a production-inventory-routing problem, Proc Int Workshop on Machine Learning, Optim and Big Data, Vol. 10122 of Lecture Notes in Computer Science, Springer International Publishing, 2016, pp. 236–245.

[7] A. Agra, M. Christiansen, A. Delgado, and L.M. Hvattum, A maritime inventory routing problem with stochastic sailing and port times, Comput Oper Res 61 (2015), 18–30.

[8] A. Agra, M. Christiansen, L.M. Hvattum, and F. Rodrigues, A MIP based local search heuristic for a stochastic maritime inventory routing problem, Comput Logist, Vol. 9855 of Lecture Notes in Computer Science, Springer International Publishing, 2016, pp. 18–34.

[9] A. Agra, M. Christiansen, K.S. Ivarsøy, I.E. Solhaug, and A. Tomasgard, Combined ship routing and inventory management in the salmon farming industry, Ann Oper Res 253 (2017), 799–823.

[10] A. Agra, C. Requejo, and F. Rodrigues, A hybrid heuristic for a stochastic production-inventory-routing problem, Electron Notes Discr Math (2017), To appear.

[11] P. Avella, M. Boccia, and L.A. Wolsey, Single-period cutting planes for inventory routing problems, Transp Sci, DOI: 10.1287/trsc.2016.0729.

[12] P. Avella, M. Boccia, and L.A. Wolsey, Single-item reformulations for a vendor managed inventory routing problem: Computational experience with benchmark instances, Networks 65 (2015), 129–138.

[13] L. Bertazzi, A. Boccia, F. Guerriero, and D. Laganà, A stochastic inventory routing problem with stock-out, Transp Res Part C: Emerging Technologies 27 (2013), 89–107.

[14] N. Brahimi and T. Aouam, Multi-item production routing problem with backordering: a MILP approach, Int J Production Res 54 (2016), 1076–1093.

[15] Y. Cardona-Valdés, A. Álvarez, and J. Pacheco, Metaheuristic procedure for a bi-objective supply chain design problem with uncertainty, Transp Res Part B: Methodological 60 (2014), 66–84.

[16] M. Chitsaz, J.F. Cordeau, and R. Jans, A unified decomposition matheuristic for assembly, production and inventory routing, Les cahiers du GERAD G–2017–03, GERAD HEC Montréal, Montréal (Québec) Canada, 2017.

[17] M. Chitsaz, A. Divsalar, and P. Vansteenwegen, A two-phase algorithm for the cyclic inventory routing problem, Eur J Oper Res 254 (2016), 410–426.
[18] Concorde, *Concorde-03.12.19*, University of Waterloo 2003.

[19] M. Darvish, H. Larrain, and L.C. Coelho, A dynamic multi-plant lot-sizing and distribution problem, Int J Production Res 54 (2016), 6707–6717.

[20] G. Desaulniers, J.G. Rakke, and L.C. Coelho, A branch-price-and-cut algorithm for the inventory-routing problem, Transp Sci 50 (2016), 1060–1076.

[21] M. Fischetti and A. Lodi, Local branching, Math Program 98 (2003), 23–47.

[22] F. Glover, Tabu search and adaptive memory programming – advances, applications and challenges, Interfaces Comput Sci Oper Res: Advances Metaheuristics, Optim, Stochastic Modeling Technologies, Vol. 1, Springer US, 1997, pp. 1–75.

[23] C. Gounaris, P. Repoussis, C. Tarantilis, W. Wiesemann, and C. Floudas, An adaptive memory programming framework for the robust capacitated vehicle routing problem, Transp Sci 50 (2014), 1239–1260.

[24] M. Gruson, J.F. Cordeau, and R. Jans, A computational comparison of formulations for an integrated three-level lot sizing and transportation problem with a distribution structure, Les cahiers du GERAD G-2017-59, GERAD HEC Montréal, Montréal (Québec) Canada, 2017.

[25] F. Hein and C. Almeida, Quantitative insights into the integrated supply vehicle routing and production planning problem, Int J Production Econ 177 (2016), 66–76.

[26] S.D. Jena, J.F. Cordeau, and B. Gendron, Lagrangian heuristics for large-scale dynamic facility location with generalized modular capacities, INFORMS J Comput 29 (2017), 388–404.

[27] M. Kaut and S. Wallace, Evaluation of scenario-generation methods for stochastic programming, Pacific J Optim 3 (2007), 257–271.

[28] A.J. Kleywegt, A. Shapiro, and T.H. de Mello, The sample average approximation method for stochastic discrete optimization, SIAM J Optim 12 (2001), 479–502.

[29] P.L. Miranda, R. Morabito, and D. Ferreira, Optimization model for a production, inventory, distribution and routing problem in small furniture companies, TOP (2017), DOI: 10.1007/s11750-017-0448-1.

[30] G.L. Nemhauser and L.A. Wolsey, Integer and combinatorial optimization, New York: John Wiley & Sons, 1988.

[31] M. Padberg, T.V. Roy, and L. Wolsey, Valid linear inequalities for fixed charge problems, Oper Res 33 (1985), 842–861.

[32] Y. Pochet and L.A. Wolsey, Production planning by mixed integer programming, Springer, New York, 2006.

[33] Y. Qiu, J. Qiao, and P.M. Pardalos, A branch-and-price algorithm for production routing problems with carbon cap-and-trade, Omega 68 (2017), 49–61.
[34] M.K.I.A. Rahim, Y. Zhong, E.H. Aghezzaf, and T. Aouam, Modelling and solving the multiperiod inventory-routing problem with stochastic stationary demand rates, Int J Production Res 52 (2014), 4351–4363.

[35] Y. Rochat and É. Taillard, Probabilistic diversification and intensification in local search for vehicle routing, J Heuristics 1 (1995), 147–167.

[36] K. Rosing and C. ReVelle, Heuristic concentration: Two stage solution construction, Eur J Oper Res 97 (1997), 75–86.

[37] M. Ruokokoski, O. Solyali, J. Cordeau, R. Jans, and H. Süral, Efficient formulations and a branch-and-cut algorithm for a production-routing problem, Les cahiers du GERAD G-2010-66, GERAD HEC Montréal, Montréal (Québec) Canada, 2010.

[38] R.A. Russell, Mathematical programming heuristics for the production routing problem, Int J Production Econ 193 (2017), 40–49.

[39] O. Solyali, J.F. Cordeau, and G. Laporte, Robust inventory routing under demand uncertainty, Transp Sci 46 (2012), 327–340.

[40] O. Solyali and H. Süral, A multi-phase heuristic for the production routing problem, Comput Oper Res 87 (2017), 114–124.

[41] C. Tarantilis, Solving the vehicle routing problem with adaptive memory programming methodology, Comput Oper Res 32 (2005), 2309–2327.

[42] J. Velarde and R. Marti, Adaptive memory programing for the robust capacitated international sourcing problem, Comput Oper Res 35 (2008), 797–806.

[43] B. Verweij, S. Ahmed, A.J. Kleywegt, G. Nemhauser, and A. Shapiro, The sample average approximation method applied to stochastic routing problems: A computational study, Comput Optim Appl 24 (2003), 289–333.

[44] Q. Zhang, A. Sundaramoorthy, I.E. Grossmann, and J.M. Pinto, Multiscale production routing in multicommodity supply chains with complex production facilities, Comput Oper Res 79 (2017), 207–222.