CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Computing optimal (R,s,S) policy parameters by a hybrid of branch-and-bound and stochastic dynamic programming
Authors	Visentin, Andrea;Prestwich, Steven D.;Rossi, Roberto;Tarim, S. Armagan
Publication date	2021-01-13
Original Citation	Visentin, A., Prestwich, S., Rossi, R. and Tarim, S. A. (2021) 'Computing optimal (R,s,S) policy parameters by a hybrid of branch-and-bound and stochastic dynamic programming', European Journal of Operational Research, 294(1), pp.91-99. https://doi.org/10.1016/j.ejor.2021.01.012
Type of publication	Article (peer-reviewed)
Link to publisher's version	https://www.sciencedirect.com/science/article/pii/S037722172100014X - 10.1016/j.ejor.2021.01.012
Rights	© 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/) - http://creativecommons.org/licenses/by/4.0/
Download date	2026-09-10 04:17:06
Item downloaded from	https://hdl.handle.net/10468/11078

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

European Journal of Operational Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Production, Manufacturing, Transportation and Logistics

Computing optimal (R, s, S) policy parameters by a hybrid of branch-and-bound and stochastic dynamic programming

Andrea Visentin[a,*], Steven Prestwich[a], Roberto Rossi[b], S. Armagan Tarim[c]

[a] Insight Centre for Data Analytics, University College Cork, Ireland
[b] University of Edinburgh Business School, Edinburgh, UK
[c] Cork University Business School, University College Cork, Ireland

ARTICLE INFO

Article history:
Received 18 December 2019
Accepted 8 January 2021
Available online xxx

Keywords:
Inventory
(R,s,S) policy
Demand uncertainty
Stochastic lot sizing

ABSTRACT

A well-known control policy in stochastic inventory control is the (R, s, S) policy, in which inventory is raised to an order-up-to-level S at a review instant R whenever it falls below reorder-level s. To date, little or no work has been devoted to developing approaches for computing (R, s, S) policy parameters. In this work, we introduce a hybrid approach that exploits tree search to compute optimal replenishment cycles, and stochastic dynamic programming to compute (s, S) levels for a given cycle. Up to 99.8% of the search tree is pruned by a branch-and-bound technique with bounds generated by dynamic programming. A numerical study shows that the method can solve instances of realistic size in a reasonable time.

© 2021 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

1. Introduction

Single-item, single-stocking location, stochastic inventory systems have long been investigated under various operational assumptions, and the associated literature is large. Scarf's seminal paper, Scarf (1959), addressed this problem over a finite planning horizon comprising discrete time periods, non-stationary stochastic demands, a fixed ordering cost, and linear holding and shortage costs. Scarf proved that the (s, S) policy (more precisely the (s_t, S_t) policy) is cost-optimal. In this policy the decision-maker checks the current inventory position at review epochs (the start of each time period) and if the inventory position is at or below s_t an order is placed to raise it to S_t. For a planning horizon of T periods the optimal (s_t, S_t) policy requires $2T$ policy parameters, computed in a here-and-now fashion at the start of the planning horizon. Actual replenishment timings and associated order quantities are instead determined in a wait-and-see manner.

In this paper, we address a more general form of the inventory control problem described by Scarf. According to Silver (1981) the (R, s, S) policy is one of the most commonly adopted inventory control strategies (also called (T, s, S) or (s, S, T) in the literature, Babai, Syntetos, and Teunter (2010); Lagodimos, Christou, and Skouri (2012a)). In an (R_t, s_t, S_t) system the inventory level is checked only at review epochs R_t, which are policy parameters that are fixed at the start of the planning horizon. After a review, an order may be placed to raise the inventory level up to S_t if it is at or below s_t.

Two special cases of the (R, s, S) policy naturally arise. Firstly, it reduces to the (s, S) case if there is no explicit cost involved in carrying out inventory reviews. Inventory review (also known as stock-taking) is costly in practice, so we consider the case in which a fixed *system control* cost Silver (1981) is incurred when the inventory is reviewed, e.g. Christou, Skouri, and Lagodimos (2020); Fathoni, Ridwan, and Santosa (2019). The (R, s, S) policy relaxes the cost accounting assumption that the fixed cost of replenishment covers both review and delivery costs, and separates the fixed cost of conducting a review from the fixed ordering cost. One practical implication of this relaxation is that the order cancellation and relevant costs can be explicitly incorporated into inventory planning.

Secondly, the (R, s, S) reduces to the (R, S) policy (the *replenishment cycle policy*) if reorder levels s_t are equal to the order-up-to-levels S_t. In a replenishment cycle policy, the replenishment periods are fixed at the beginning of the planning horizon and the

* Corresponding author. Tel.: +353 834888847.
E-mail addresses: andrea.visentin@insight-centre.org (A. Visentin), s.prestwich@cs.ucc.ie (S. Prestwich), roberto.rossi@ed.ac.uk (R. Rossi), armagan.tarim@ucc.ie (S.A. Tarim).

https://doi.org/10.1016/j.ejor.2021.01.012
0377-2217/© 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

Please cite this article as: A. Visentin, S. Prestwich, R. Rossi et al., Computing optimal (R, s, S) policy parameters by a hybrid of branch-and-bound and stochastic dynamic programming, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2021.01.012

replenishment orders are placed only in these periods after period demands so far have been observed.

Although (R, s, S) is one of the most general and frequently used inventory policies, as pointed out by Silver (1981) *the determination of the exact best values of the three parameters is extremely difficult*. To the best of our knowledge no approach to computing them has been presented in the literature. We fill this important gap in the literature by making the following contributions:

- we introduce an efficient hybrid of branch-and-bound and stochastic dynamic program (SDP) to compute optimal policy parameters;
- we improve the branch-and-bound by using tighter bounds computed through a separate dynamic programming (DP) method;
- we show empirically that the new algorithm performs significantly better than a baseline method and that it is able to solve problems of realistic size in a reasonable time.

The paper is structured as follows. Section 2 surveys related literature. Section 3 provides a problem description. Section 4 introduces a simple SDP formulation. Section 5 introduces a branch-and-bound strategy. Section 6 carries out a comprehensive numerical study. Finally, Section 7 concludes the paper.

2. Literature review

The problem of computing policy parameters for an inventory control system under stochastic demand has received a great deal of attention. In this section, we survey the relevant literature on the classic stochastic inventory control problem. We then survey different versions of the problem. Finally, we survey (R, s, S) real-world applications.

An important class of these problems is the single-item single-location non-stationary stochastic lot-sizing under linear holding costs, penalty costs and both linear and fixed ordering costs. Different policies can be used to determine the size and timing of the orders on such setting. In his seminal work, Scarf characterises the structure of the optimal policy for such a problem. The framework proposed by Bookbinder and Tan (1988) divides the policies into three classes: static uncertainty, dynamic uncertainty and static-dynamic uncertainty. These classes differ in the moment at which the decisions are taken. Since then, numerous research works tackled the computation of policy parameters under demand uncertainty, mainly focusing on the (s, S) and the (R, S) policies that have a flexible order quantity. According to the strategies categorisation presented in Powell (2019), these works can be divided into two types: *deterministic/special structure solutions* or *sample models*. The first category, that comprises this study, includes a wide variety of approaches based on: dynamic programming Özen, Doğru, and Tarim (2012); Rossi, Tarim, Hnich, and Prestwich (2011); Scarf (1959), mixed-integer linear programming Tarim and Kingsman (2004); Tunc, Kilic, Tarim, and Rossi (2018); Xiang, Rossi, Martin-Barragan, and Tarim (2018), approximations Gutierrez-Alcoba, Rossi, Martin-Barragan, and Hendrix (2017), and constraint programming Rossi, Tarim, Hnich, and Prestwich (2012). The sample models category includes two-stage stochastic programming, that has been applied to inventory policy computation in Cunha, Raupp, and Oliveira (2017); Fattahi, Mahootchi, Moattar Husseini, Keyvanshokooh, and Alborzi (2015); dos Santos and Oliveira (2019). Different comparison studies have been conducted recently to benchmark different aspects of these policies: Kilic and Tarim (2011) extends a measure of planning instability for the non-stationary stochastic lot-sizing; Dural-Selcuk, Rossi, Kilic, and Tarim (2019a) compares different policies performances in the receding horizon, Sani and Kingsman (1997) and Babai et al. (2010) are comparative studies on the performances of (s, S) heuristics.

Modifications on the original inventory model have been proposed to allow a closer representation of real-world problems. Dillon, Oliveira, and Abbasi (2017) proposes an (R, S) policy solution to manage the blood supply chain, that includes perishable products. Alvarez, Buijs, Kilic, and Vis (2020)'s model considers both quantity and quality decay of the inventory product; the quality can be improved by mixing it with a higher quality product. A set of heuristics for the lot-sizing problem with remanufacturing of returned products is presented in Kilic and Tunc (2019). All-units discount (s, S) policy has been analysed in Wang, Yu, Zhang, and Hua (2019). Uncertainty can involve other aspects of the inventory system; for example, Bashyam and Fu (1998); Rossi, Tarim, Hnich, and Prestwich (2010) considers a stochastic lead time. Different supply chain configuration can be considered; for example, a two-echelon inventory system Schneider and Rinks (1991); Schneider, RNKS, and Kelle (1995). Ma, Rossi, and Archibald (2019) provides an updated review on stochastic inventory control algorithms, while Bushuev, Guiffrida, Jaber, and Khan (2015) presents a broader picture of the state-of-the-art in lot sizing.

The (R, s, S) policy parameters computation has been tackled in the literature under the stationary, continuous time setting. With this configuration, only three parameters have to be optimised since the demand does not change over time. This problem has been solved to optimality by Lagodimos et al. (2012a). In Christou et al. (2020); Lagodimos, Christou, and Skouri (2012b) a batch version of the policy is considered.

None of the surveyed methods can be easily adapted to compute the (R, s, S) policy parameters under the finite horizon and discrete time setting, since it has three sets of decision variables making the previous models not applicable. While other policies can be used for the same problem, the (R, s, S) is a more generic model and has better cost performances, being the (s, S) and the (R, S) specific case of an (R, s, S) policy. The introduction of the review cost makes no difference in the (s, S) and (R, S) policy computation; in the (s, S) the cost is charged in every period, while in the (R, S) every review coincides with an order. A static policy would also have poor performance because it can not react to the demand realisations Dural-Selcuk et al. (2019a).

The (R, s, S) policy is widely used by practitioners, usually not independently but as a component of complex supply chains, and here we survey some recent models. Due to the complexity of the determination of its parameters, in the surveyed papers, the value of R is considered to be constant across the time horizon. Bijvank and Vis (2012a) describe an inventory control system for point-of-use location. They compare the performance of (R, s, Q) policies (with fixed order quantities) against (R, s, S) under stationary stochastic demand. Because of stationarity, the policy parameters were constant throughout the horizon. Ahmadi, Mahootchi, and Ponnambalam (2018); Monthatipkul and Yenradee (2008) tackle a capacitated two-echelon inventory system with one warehouse and multiple retailers. They use a heuristic based on Schneider et al. (1995) for the (R, s_t, S_t) policy. Cabrera et al. (2013) consider a similar two-level supply chain in which a single plant serves a set of warehouses, which in turn serve a set of end customers or retailers. The warehouses model is based on (R, s, S) and they develop a heuristic to solve an inventory location model with this configuration. The same problem has been tackled by Araya-Sassi, Miranda, and Paredes-Belmar (2018) using Lagrangian relaxation and the subgradient method. Bijvank and Vis (2012b) analysed lost-sales inventory control policies with service level. They define an optimal policy starting from the (s, S) SDP introduced by Scarf (1959). They present a value-iteration algorithm to find the (R, s, S) parameters that minimise the inventory cost subjected to service constraints. As the parameters are fixed, their solution is unsuitable for a non-stationary setting.

The state-of-the-art analysis confirms the novelty of our solution and the practitioners' interest in the (R, s, S) policy usage in a stochastic environment.

3. Problem description

We consider the single-item, single-stocking location, stochastic inventory control problem over a T-period planning horizon. Without loss of generality, we assume that orders are placed at the start of each period and that the lead time is zero, as is common in the literature Bollapragada and Morton (1999); Scarf (1959); Tarim and Kingsman (2004). An inventory control policy defines the timing and quantities of orders over the planning horizon. We define a review moment, or review period, as a period in which the level of the inventory is assessed and an order can be placed. A replenishment cycle is represented by the interval between two review moments. We denote by Q_t the quantity of the order placed in period t, and an inventory review cost by W. Ordering costs are represented by a fixed value K and a linear cost, but we shall assume without loss of generality that the linear cost is zero. The extension of our solution to the case of a non-zero production/purchasing cost is straightforward, as this cost can be reduced to a function of the expected closing inventory level at the final period Tarim and Kingsman (2004). At the end of each period, a linear holding cost h is charged for every unit carried from one period to the next.

Demands d_t in each period t are independent random variables with known probability distributions. Backlogging of excess demand is assumed, so if the demand in a period exceeds on-hand inventory the rest of the demand is carried to the next period; a linear penalty cost b is incurred on any unit of back-ordered demand at the end of a period.

Under the non-stationarity assumption the (R, s, S) policy takes the form (R_t, s_t, S_t) where R_t denotes the length of the t^{th} replenishment cycle, while parameters s_t and S_t denote the reorder-level and order-up-to-level associated with the t^{th} inventory review. We consider the problem of computing the (R, s, S) policy parameters that minimize the expected total cost over the planning horizon.

4. Stochastic dynamic programming formulation

In this section, we provide a simple technique to compute the optimal (R, s, S) policy parameters. It can be considered the state-of-the-art in computing such parameters in the presence of stochastic non-stationary demand. Moreover, it constitutes the basis of the branch-and-bound technique introduced later.

We represent the replenishment moments by binary variables γ_t $(t = 1, \ldots, T)$ which take value 1 if a review is placed in period t and 0 otherwise. We assume $Q_t = 0$ if $\gamma_t = 0$ so no order will be placed outside a review moment. The optimal (R, s, S) policy for our problem is represented by the parameters γ_t, s_t, S_t that minimize the expected total cost.

Consider an arbitrary review cycle plan with γ_t as a parameter, not a decision variable. We denote the closing inventory level for each period by I_t, and the given initial inventory level by I_0. We assume that the orders are placed at the beginning of each time period and delivered instantaneously. The problem can be formulated and solved to optimality as an SDP (Bellman, 1966). The expected immediate cost combining ordering, review, holding and penalty costs, given action Q_t:

$$f_t(I_{t-1}, Q_t) = \gamma_t W + K\mathbb{1}\{Q_t > 0\} + E[h\max(I_{t-1} + Q_t - d_t, 0) + b\max(d_t - I_{t-1} - Q_t, 0)] \quad (1)$$

Let $C_t(I_{t-1})$ represent the expected total cost of an optimal policy over periods $t, \ldots, n$ and $\mathbb{1}$ is the indicator function. These variables are the states of the DP formulation. We model the problem with the functional equation:

$$C_t(I_{t-1}) = \min_{0 \le Q_t \le M\gamma_t} (f_t(I_{t-1}, Q_t) + E[C_{t+1}(I_{t-1} + Q_t - d_t)]) \quad (2)$$

where M is a sufficiently large number. The boundary condition is:

$$C_{T+1}(I_T) = 0 \quad (3)$$

$C_1(I_0)$, where I_0 is the initial inventory level, contains the expected cost for the optimal (s, S) policy associated with the γ assignment. To reduce the computational time we can exploit the property of K-convexity (Scarf, 1959) when solving the SDP.

Let $\hat{C}_1(I_0)$ represent the expected total cost of the optimal (R, s, S) policy, given the initial inventory level I_0 at period 1. We can define it as:

$$\hat{C}_1(I_0) = \min_{\gamma_1, \ldots, \gamma_T} (C_1(I_0)) \quad (4)$$

Evaluating the optimal (s, S) policy for all possible assignments of $\gamma_1, \ldots, \gamma_T$ yields the optimal (R, s, S) policy. The model works with every possible demand distribution, as long as it is finite and discretisable. This is our baseline method on which we aim to improve.

4.1. Unit cost

The algorithm can be extended to model the per unit ordering cost. There are two options: reducing it to a function of the expected closing inventory, e.g. Tarim and Kingsman (2004); or including it in the immediate cost function.

Let v be the per unit ordering/production cost, Eq. (1) is replaced by:

$$f_t(I_{t-1}, Q_t) = \gamma_t W + K\mathbb{1}\{Q_t > 0\} + vQ_t + E[h\max(I_{t-1} + Q_t - d_t, 0) + b\max(d_t - I_{t-1} - Q_t, 0)] \quad (5)$$

4.2. Lost sales

Complete backlogging of the demand is a limiting assumption in many real-world settings. Studies analysing customer behaviour show that in case of a stock out, only a minority delay the purchase (Verhoef & Sloot, 2006). According to Bijvank and Vis (2012a), the lost sales configuration is underrepresented in the lot-sizing literature, even if it is more appropriate to model customers' behaviour. Approximating a lost sales model with a backlog model results in a non-negligible increase in costs (Zipkin, 2008).

The SDP formulation can be extended to model lost sales using the partial backorder configuration presented in dos Santos and Oliveira (2019). They define as β ($\beta \in [0, 1]$) the fraction of the unmet demand that is carried on to the next period and the reminder is lost. This parameter gives the flexibility to model both backlog ($\beta = 1$), lost sales ($\beta = 0$) or a combination of the two. The functional Eq. (2) becomes:

$$C_t(I_{t-1}) = \min_{0 \le Q_t \le M\gamma_t} (f_t(I_{t-1}, Q_t) + E[C_{t+1}(\max(I_{t-1} + Q_t - d_t, \beta(I_{t-1} + Q_t - d_t))]) \quad (6)$$

4.3. Example

We use a simple example to illustrate the application of our method, with a 3-period planning horizon. We assume an initial inventory level of zero and a Poisson distributed demand for each period with averages $\bar{d} = [20, 30, 40]$. We consider an ordering cost value $K = 30$, a review cost $W = 10$, and holding and penalty costs of $h = 1$ and $b = 10$ per unit per period respectively.

The algorithm must choose replenishment moments $\gamma = \langle \gamma_1, \gamma_2, \gamma_3 \rangle$ that minimize the expected cost of the policy. Table 1

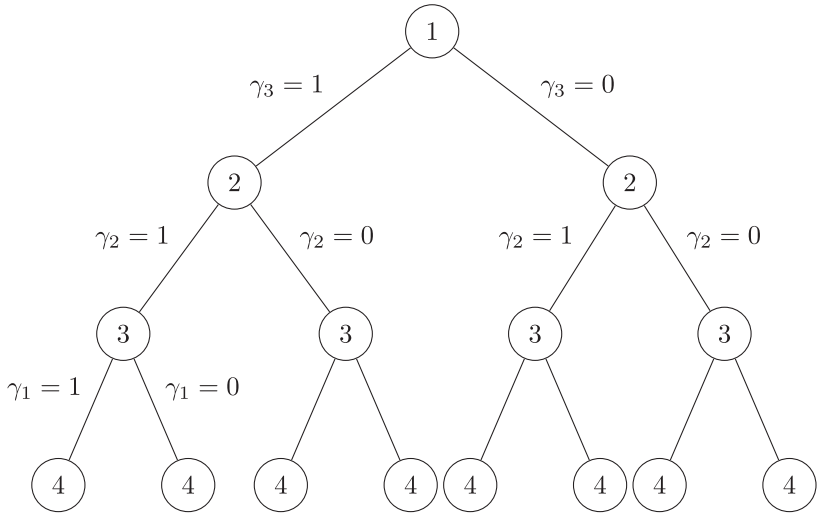

Fig. 1. Search tree for a 3-period instance: nodes contains level numbers.

Table 1
Optimal expected cost for the 3-period example.

γ_1	γ_2	γ_3	Expected cost
0	0	0	1600.0
0	0	1	751.8
0	1	0	304.7
0	1	1	302.0
1	0	0	185.0
1	0	1	**142.7**
1	1	0	153.1
1	1	1	150.4

shows the expected cost of each (s, S) policy computed with different review periods. The optimal solution is $\gamma = \langle 1, 0, 1\rangle$ with expected cost 142.7. However, exhaustive search becomes impractical as the planning horizon grows so in Section 5 we develop a more efficient method.

5. A hybrid of branch-and-bound and SDP

In this section, we present a hybrid technique that combines SDP and branch-and-bound. The algorithm obtains optimal (R, s, S) policies associated with specific review plans at leaf nodes. The search tree (defined in Section 5.1) is explored by depth-first search (DFS). The subproblems associated with the nodes are defined in Section 5.2. Section 5.3 introduces the pruning condition and lower bound computed with DP. Finally, Section 5.4 presents the node resolution process.

5.1. Search tree

The branch-and-bound goal is to find the review plan with the minimum expected cost. During branching of γ_t, the value is fixed to 1 or 0. The search tree has $T + 1$ levels, and the branching at its root fixes the value of γ_T. At level ℓ branching involves the variable $\gamma_{T-\ell+1}$. The path from the root to a node at level ℓ represents a fixed assignment of the suffix $\langle \gamma_{T-\ell+2}, \ldots, \gamma_T\rangle$. A leaf node represents a complete assignment of the γ values. Fig. 1 shows the search tree of a 3-period problem, as in the example presented in the previous section.

5.2. Subproblems

Given the period t and the partial assignment of a suffix of the review moments $\langle \gamma_t, \ldots, \gamma_T\rangle$, the problem at a node is to find the $\langle \gamma_1, \ldots \gamma_{t-1}\rangle$ that minimizes the expected cost of the optimal policy. We denote this problem as BnB-SDP$(t, \langle \gamma_t, \ldots, \gamma_T\rangle)$. For each subproblem using Eq. (2) we can compute the expected cost of the optimal policy starting at period t with inventory level i. This is possible because all review moments after period t are fixed, and because of the SDP stage structure presented in Section 4.

5.3. Bounds and pruning

If all the solutions in the subtree rooted in a node are suboptimal then we can prune that node without compromising optimality.

Proposition 1. *Given a fixed assignment of γ:*

$$\min_I(C_t(I)) \geq \min_I(C_{t-1}(I)) \tag{7}$$

From the functional Eq. (2) it is clear that C_t is equal to the expected value of C_{t+1} plus some non-negative costs, so the minimum cost in each stage increases monotonically with tree depth.

During tree search $\bar{C}$ records the expected cost of the best plan computed so far, that is the minimum $C_1(I_0)$ among all leaves already computed. This is used as an upper bound for the expected cost of the optimal plan as follows. Considering the subproblem BnB-SDP$(t, [\gamma_t, \ldots, \gamma_T])$ with the associated $C_t(i)$ expected costs:

Proposition 2. *If*

$$\min_i(C_t(i)) \geq \bar{C} \tag{8}$$

then because of the monotonicity of the cost function: (7):

$$\min_i(C_1(i)) \geq \bar{C} \tag{9}$$

Finally, since the expected cost associated with a plan $(C_1(I_0))$ is part of C_1:

$$C_1(I_0) \geq \bar{C} \tag{10}$$

Hence if (8) is true the subproblem BnB-SDP$(t, [\gamma_t, \ldots, \gamma_T])$ is not part of an optimal solution and the search tree can be pruned.

However, this pruning condition makes no assumption on the costs faced on periods $1, \ldots, t-1$, and a lower bound on the costs in those periods leads to more effective pruning. Let $MC_t(I_t)$ represent a lower bound on the cost faced in periods $1, \ldots, t$ with a closing inventory of I_t in period t. The pruning condition (8) can be refined to:

$$\min_{I_t}(C_t(I_{t-1}) + MC_{t-1}(I_{t-1})) \geq \bar{C} \tag{11}$$

Having a bound independent from the review periods allows us to compute it only once before the branch-and-bound algorithm.

The bounds can be computed by a DP with stages and states equivalent to the SDP presented in Section 4 and functional equation:

$$MC_t(I_t) = \min \begin{cases} f_t(I_t, 1) + \min_{j<I_t}(MC_{t-1}(j)) \\ f_t(I_t, 0) + \min_{j\geq I_t}(MC_{t-1}(j)) \end{cases} \quad (12)$$

where, as defined in Section 4, I_t is the current inventory level, and $f_t(I_t, Q_t)$ is the ordering-holding-penalty cost. In the first case, an order has been placed in period t so the inventory level in the previous period was less than or equal to the current level. In the second case, an order has not been placed so the previous inventory level was greater than or equal to the current level. The boundary condition is:

$$MC_1(I_1) = \begin{cases} W + K + f_1(I_1) & \text{if } I_1 > I_0 \\ f_1(I_1) & \text{if } I_1 \leq I_0 \end{cases} \quad (13)$$

where I_0 is the initial inventory. Considering finite demand, the DP has an amount of states equal to the number of periods multiplied by the maximum inventory level. Each state requires a single computation of Eq. (12), that is pseudo-polynomial in relation to the maximum inventory. The overall complexity of a DP is the number of states multiplied by the complexity required to solve one of them, so the overall complexity is pseudo-polynomial.

5.4. Node computation

Algorithm 1 summarises the branch-and-bound procedure. In line 1, the SDP stage t is solved. In line 7, the pruning condition is evaluated: if a pruning occurs the branching phase is skipped. In lines 8 and 9 DFS recursively continues. Lines 3–6 relate to leaf nodes: if the policy represented by the leaf is better than the best found so far, the value of $\bar{C}$ is updated. The algorithm starts by invoking BnB-SDP$(T+1, \emptyset)$, and at the end, the expected cost of the optimal policy is given by $\bar{C}$.

Algorithm 1 BnB-SDP$(t, [\gamma_t, \ldots, \gamma_T])$.

Data: the current upper bound $\bar{C}$, the $C_{t+1}(i)$computed at the parent node, and the bounds $MC(i)$.

1: Compute C_t using Equation 2
2: **if** $t = 1$ **then**
3: **if** $C_1(I_0) < \bar{C}$ **then**
4: $\bar{C} \leftarrow C_1(I_0)$
5: Save $[\gamma_1, \ldots, \gamma_T]$ as incumbent review plan
6: **else**
7: **if** $\min(C_t(i) + MC_{t-1}(i)) \geq \bar{C}$ **thenreturn**
8: BnB-SDP$(t-1, [0, \gamma_t, \ldots, \gamma_T])$
9: BnB-SDP$(t-1, [1, \gamma_t, \ldots, \gamma_T])$

The algorithm always branches by assigning first $\gamma_t = 0$, but its performance can be improved by randomisation. If, during each branching phase, we randomly order lines 8–9 we obtain a better solution earlier, leading to a stronger pruning of the search tree. We evaluate the effect of this randomisation in Section 6.

5.5. Guided tree search

The random descent can become trapped in inferior branches of the search tree. It takes a considerable time to obtain a reasonable review plan, and a good cost upper bound in these cases. Computing a near-optimal review plan and using it to guide the search leads to the immediate computation of a policy with a low expected cost. This tighter bound increases the number of nodes proved sub-optimal by the pruning condition.

A reasonable review plan can be computed using the (R_t, S_t) policy. As mentioned in the introduction, this policy places an order at each review moment. The replenishment cycles (R_t) can be used as a review plan, while the order-up-to-levels S_t can be ignored. During the first descent of the branch-and-bound search tree, the gamma values are selected following this review plan; thus, the first leaf to be computed is the one that has R_t as review moments. This leaf represents the optimal (R_t, s_t, S_t) policy for that review plan, and it should have a low expected cost. After computing the first leaf of the tree, the search proceeds in the replenishment plan's neighbourhood using a randomised approach.

The experimental section shows the improvement in pruning efficacy and computational time. Good computational performance and implementation simplicity make the MILP formulation presented in Rossi, Kilic, and Tarim (2015) a practical solution to compute the (R_t, S_t) policy; this formulation is used in the experimental section.

5.6. Example

The search tree with the DP bounds for the example of Section 4.3 is represented in Fig. 2. Each internal node contains the value of the pruning condition with the DP bounds (11). An internal node is underlined if the pruning occurs in that node. Each leaf is in bold if it contains an improvement compared to the previous best solution $\bar{C}$. Pruned nodes are indicated by an asterisk $(^*)$.

We define *pruning percentage* as the percentage of nodes that are proved to be suboptimal by the pruning condition during tree search. In this example, the number of computed nodes is 10 and 4 nodes have been pruned, so the pruning percentage is $4/14 = 28.57\%$.

6. Computational study

In this section, we evaluate the new methods, including an assessment of the effects of branching randomisation and problem parameters (costs) empirically. We conduct two sets of experiments as follows. In Section 6.1, we analyse the scalability of the new approaches by increasing the number of periods until no method is able to solve the problem within a 1-hour time limit consistently. In Section 6.2 we fix the planning horizon to 10 and 20 periods and vary the cost parameters. For the experiments, we use three (R, s, S) policy solvers:

- **SDP**, the SDP technique described in Section 4 which we consider the current state-of-the-art.
- **BnB**, the branch-and-bound solution introduced in Section 5.
- **BnB-Rand**, branch-and-bound with randomised branching.
- **BnB-Guided**, branch-and-bound with a guided tree search, Section 5.5.

We compare these in terms of computational time, pruning percentage and average number of review periods (but not expected costs because the solutions are optimal in each case). All experiments are executed on an Intel(R) Xeon E5620 Processor (2.40GHz) with 32 Gb RAM. For the sake of reproducibility, we made the code available[1].

We base our numerical studies on the set of instances originally proposed by Berry (1972) and widely used in the literature Dural-Selcuk, Rossi, Kilic, and Tarim (2019b); Rossi, Tarim, Hnich, and Prestwich (2008); Xiang et al. (2018). A Poisson variable represents the demand in each period.

[1] https://github.com/andvise/RsS-EJOR

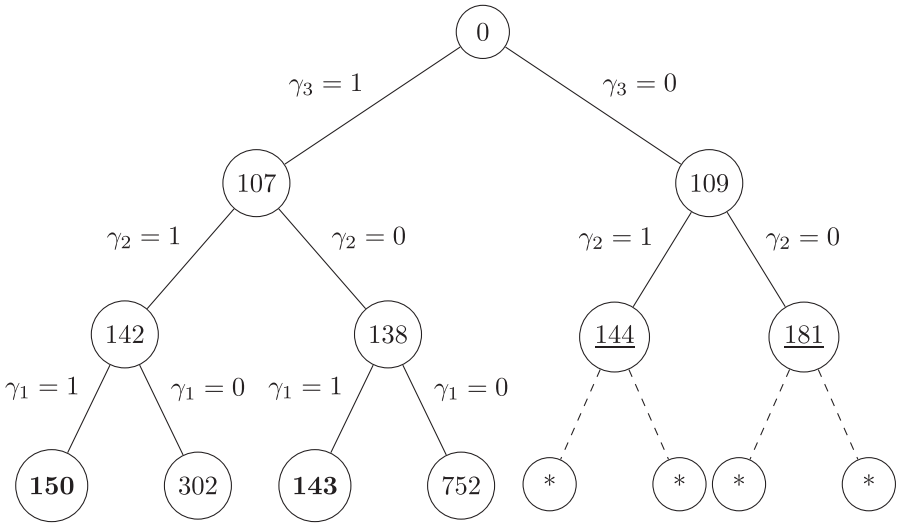

Fig. 2. Branch-and-bound technique applied to the toy problem.

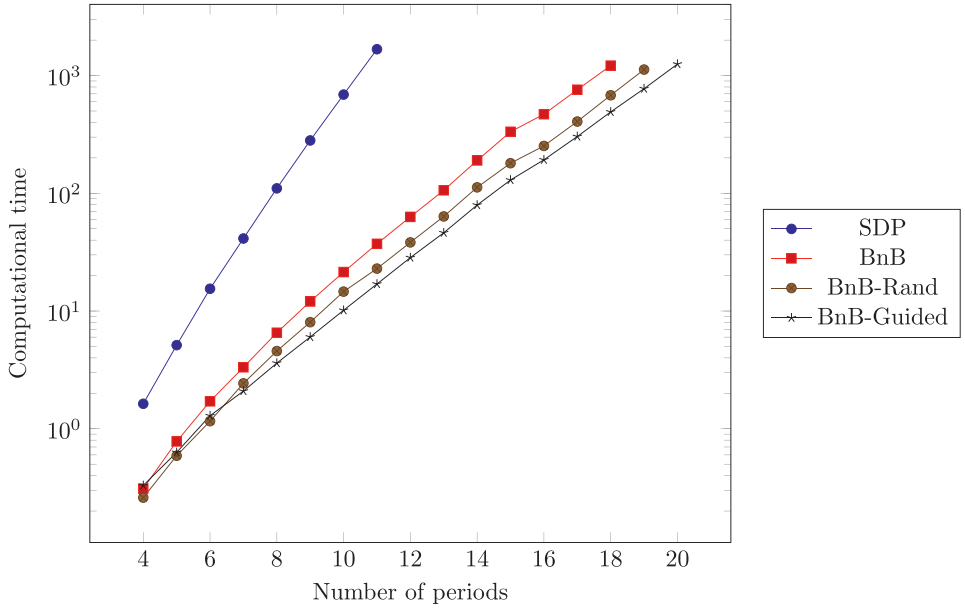

Fig. 3. Average computational time of the 100 instances over the number of periods. Time limit 1 hour.

6.1. Scalability

This experiment aims to assess the improvement provided by the branch-and-bound approach compared to what we can consider as the state-of-the-art. Furthermore, we aim to assess how the randomisation and the guided search affect the computational performances and the pruning percentage. For the scalability analysis, we use randomly generated parameter values and progressively increase the number of periods. We fix the holding cost per unit at $h = 1$, but the other cost parameters are uniform random variables: ordering cost is in the range $K \in [80, 320]$, review cost is in the range $W \in [80, 320]$ and penalty cost per unit is in the range $b \in [4, 16]$. Demands per period are uniform random variables in the range [30,70]. We generate 100 different instances and for planning horizons in the range 4–20 periods.

Fig. 3 shows the average computational time over the 100 instances. The y-axis is logarithmic to show the exponential behaviour of the solutions. The new method is able to solve instances almost twice as large in a reasonable time. Though it still has exponential behaviour, its slope is considerably less than that of the SDP. The randomisation reduces the computational effort needed. The guided search requires the computation of an (R, S) policy before the BnB approach. For small instances, the added computational effort is higher than the improvement provided by a higher pruning percentage. However, for medium/big instances, the improvement is considerable.

Fig. 4 shows the range of the minimum and maximum computational times for increasing planning horizon lengths; we omitted BnB-Rand to improve the readability of the plot. The SDP solution has a low variability in the required computational time. BnB-Guided presents the highest variability among the different solutions. This is due to the fact that in some instances, the precomputed replenishment plan is the optimal one and leads to a strong pruning of the tree that reduces the computational time considerably.

Fig. 5 shows the pruning percentage (Section 5.6) of the branch-and-bound approaches. The pruning becomes more effective for longer planning horizons. A high value means that the approach

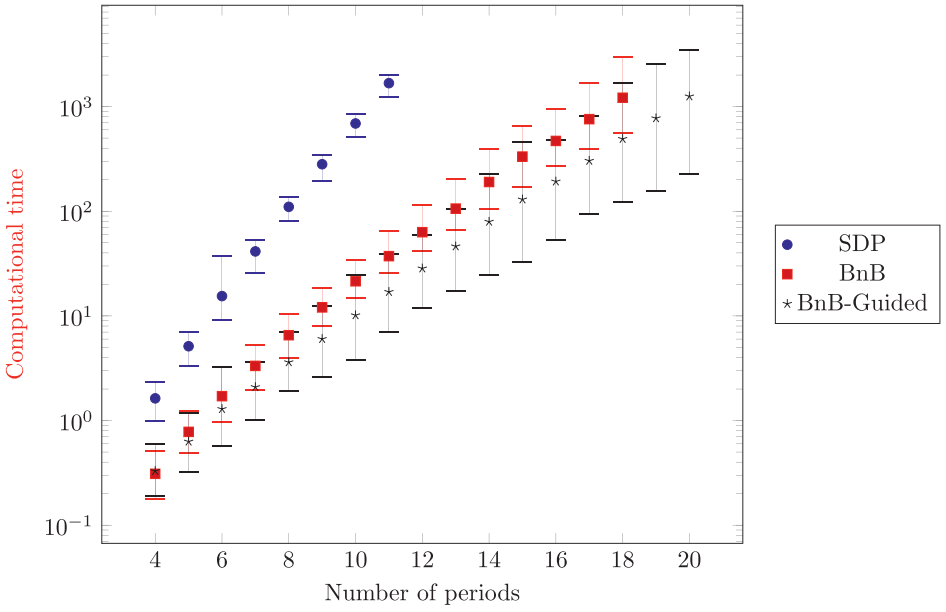

Fig. 4. Range of computational time over the number of periods. Time limit 1 h.

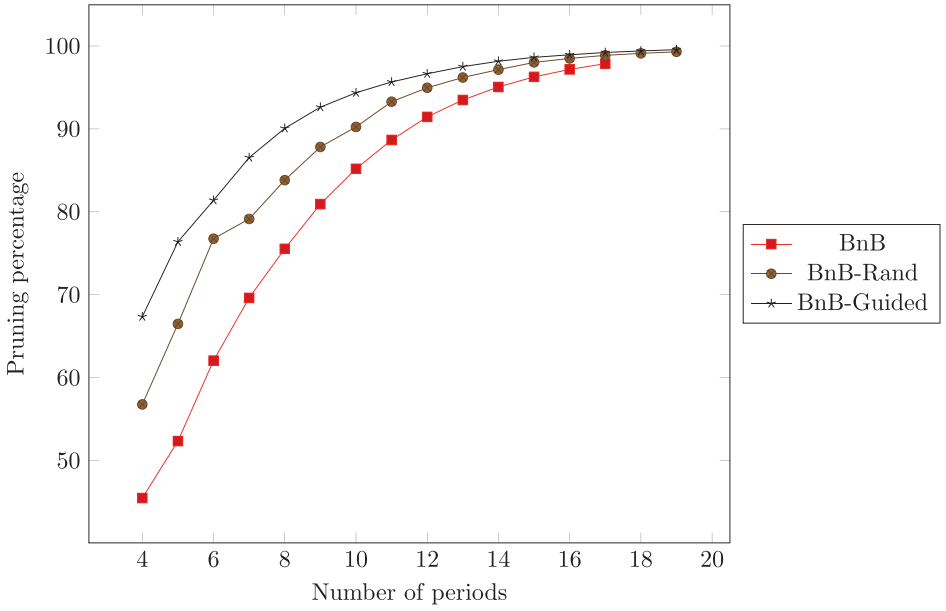

Fig. 5. Average percentage of nodes pruned over the 100 instances in relation to the number of periods.

finds a good policy earlier in the search process; the cost of this policy provides a tighter bound for the pruning condition (Eq. (11)). We can see that BnB-Guided provides considerable improvement.

6.2. Instance type analysis

In the parameter value analysis, we aim to understand how the cost parameters affect the computational effort required to find the policy and the pruning percentage. We use a testbed of 324 instances. To generate the average demand values we use seasonal data with different trends:

- **(STA)** stationary case: $\tilde{d}_t = 50$
- **(INC)** positive trend case: $\tilde{d}_t = \lceil 100t/(n-1) \rceil$
- **(DEC)** negative trend case: $\tilde{d}_t = \lceil 100 - 100t/(n-1) \rceil$
- **(LCY1)** life-cycle trend 1 case: this pattern is a combination of the first 3 trends. The first third of positive trend up to an average demand of 75, a central stationary one and the last negative third. If the number of periods is not a multiple of 3, the central period is extended.
- **(LCY2)** life-cycle trend 2 case: this pattern is a combination of INC and DEC trends. Positive trend for the first half of the planning horizon and negative trend for the second half.
- **(RAND)** erratic: $\tilde{d}_t = \lceil U(1, 100) \rceil$

All the patterns have an average demand of 50 per period. For the cost parameters we use all possible combinations of ordering cost values $K \in \{80, 160, 320\}$, review costs $W \in \{80, 160, 320\}$ and penalty costs $b \in \{4, 8, 16\}$, with holding cost fixed at $h = 1$. We use all combinations of cost parameters and the six demand patterns presented above for a full factorial experiment. We analyze the results for the 10-periods and 20-periods instances.

Since the baseline (Eq. (4) in Section 4) is too computationally expensive (it takes approximately 45 days to solve a 20 periods in-

Table 2
Computational times (in minutes), pruning percentage and number of reviews for 10 periods instances. Between brackets is the standard deviation of the pruning percentage.

		Computational time			Pruning %		
		Base	BnB	BnB-Guided	BnB	BnB-Guided	Nr. reviews
K values	80	14.62	0.55	0.3	82.15(3.37)	91.51(3.94)	3.0
	160	14.83	0.56	0.28	81.79(3.38)	92.06(4.07)	2.56
	320	14.97	0.61	0.32	80.31(4.74)	91.06(5.37)	2.06
W values	80	14.83	0.68	0.46	78.36(4.37)	86.94(4.08)	3.0
	160	14.79	0.56	0.27	81.94(2.84)	92.48(2.48)	2.56
	320	14.8	0.48	0.17	83.96(1.99)	95.2(1.76)	2.06
b values	4	14.89	0.57	0.28	81.48(3.32)	92.06(4.21)	2.39
	8	14.81	0.57	0.29	81.69(3.72)	91.89(4.27)	2.56
	16	14.71	0.58	0.33	81.09(4.7)	90.68(4.93)	2.67
Pattern	STA	10.76	0.37	0.21	82.87(2.87)	91.58(4.21)	2.63
	INC	17.3	0.69	0.46	81.27(3.27)	88.67(3.96)	2.7
	DEC	17.39	0.66	0.27	81.5(2.63)	93.7(3.6)	2.33
	LCY1	15.03	0.65	0.32	78.61(3.49)	90.72(4.52)	2.59
	LCY2	16.56	0.71	0.36	78.99(3.25)	90.57(5.01)	2.48
	RAND	11.79	0.35	0.17	85.27(3.86)	94.01(3.19)	2.48
Average		14.81	0.57	0.3	81.42(3.96)	91.54(4.52)	2.54

Table 3
Computational times (in minutes), pruning percentage and number of reviews for 20 periods instances. Between brackets the standard deviation of the pruning percentage.

		Computational time			Pruning %		
		Base	BnB	BnB-Guided	BnB	BnB-Guided	Nr. reviews
K values	80	65366.67	105.12	47.57	98.56(0.76)	99.34(0.52)	6.04
	160	65470.02	109.35	49.78	98.53(0.92)	99.33(0.59)	5.17
	320	66070.17	115.98	51.9	98.47(1.04)	99.32(0.68)	4.13
W values	80	66737.03	181.66	100.37	97.61(0.9)	98.67(0.56)	6.04
	160	64772.93	96.12	36.09	98.68(0.45)	99.5(0.23)	5.17
	320	65396.9	52.67	12.8	99.27(0.23)	99.83(0.1)	4.13
b values	4	65851.88	96.59	41.92	98.7(0.76)	99.45(0.53)	4.78
	8	65847.45	108.37	48.75	98.56(0.83)	99.35(0.55)	5.2
	16	65207.52	125.49	58.59	98.3(1.07)	99.21(0.69)	5.35
Pattern	STA	43447.11	73.24	38.54	98.51(0.84)	99.23(0.66)	5.3
	INC	72449.66	110.73	69.98	98.69(0.62)	99.18(0.53)	5.41
	DEC	72706.98	141.49	48.06	98.29(1.05)	99.43(0.57)	4.7
	LCY1	62607.87	139.2	62.58	98.05(0.96)	99.13(0.72)	5.19
	LCY2	69243.25	141.35	56.22	98.22(0.82)	99.3(0.57)	5.04
	RAND	73358.85	54.88	23.13	99.36(0.31)	99.74(0.23)	5.04
Average		65635.62	110.15	49.75	98.52(0.91)	99.33(0.6)	5.11

stance) we replace it with an estimate in the 20-periods instances. The estimate is computed by solving 100 times the SDP for different γ assignments and averaging it over all the possible assignments.

Tables 2 and 3 give an overview of the computational time, the pruning percentage and the average number of reviews of the methods for the 10- and 20-period experiments. They show that SDP is not strongly affected by the cost parameters and that the main difference is caused by the demand patterns. This is due to the maximum average demand per period being lower for STA, LCY1 and RAND. The stationary case is faster to compute as its maximum is 50, the second-fastest is the first life cycle with a maximum of 75, and the erratic pattern is slowest. All the other patterns have a maximum of 100.

The pruning percentage gives an indication of the efficacy of the branch-and-bound. Our algorithms perform particularly well on high review costs. For instance, with 20 periods and $W = 320$ the pruning percentage reached a high average of 99.83% for BnB-Guided, solving one instance in less than 13 minutes on average, while the baseline is expected to take more than six weeks. For the BnB, the percentage is 98.52%, so it visits more than twice the nodes compared to the guided version. The randomised search (not shown in the table for the sake of readability) reaches an average of 98.92%. We note that the penalty cost also affects performance: a higher penalty cost reduces pruning.

The average number of review moments of the optimal policies decreases as the ordering and the review increase. Also, a higher penalty cost leads to more frequent reviews, which reduces the probability of demand excess and mitigates the uncertainty of the inventory level. We observe that the decreasing pattern requires fewer review periods than the others, due to its decreasing tail that reduces the number of orders needed.

Our best-proposed method outperforms the baseline by factors of 50 and 1300 on 10- and 20-period instances, respectively.

7. Conclusion and future work

In this paper, we considered the single-item single-stocking location inventory lot-sizing problem with non-stationary stochastic demand, fixed and linear ordering cost, review cost, holding cost and penalty cost. We present the first algorithm to compute optimal (R, s, S) policy parameters. This policy has a high practical value, but the computation of optimal or near-optimal parameters has been considered extremely difficult. Our proposed technique is a hybrid of branch-and-bound and stochastic dynamic programming, enhanced by ad hoc bounds computed with dynamic programming, and by a randomised depth-first exploration of the search tree.

In an extensive numerical study, we first investigated the scalability of the technique under increasing time horizon, analysing both computational time and the efficacy of the bounding technique. We then tested the performance of the method for different cost parameters. Our technique performs best on low penalty costs

and high review costs. On 20-period instances, it outperforms a baseline method by three orders of magnitude.

This technique opens up multiple research directions on the determination of (R, s, S) policy parameters. It can lead to new optimal solutions for the same problem, and it can be improved with tighter bounds. It is also useful for computing optimality gaps of new heuristics.

In future studies, the approach presented herein can be extended to overcome some of the limitations of the problem setting. Considering multiple items with joint shipping or modelling a more complex supply chain with multiple echelons are generalisations of this problem, and they would increase the applicability of the (R, s, S) policy.

Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 12/RC/2289-P2, which is co-funded under the European Regional Development Fund.

References

Ahmadi, T., Mahootchi, M., & Ponnambalam, K. (2018). Optimal randomized ordering policies for a capacitated two-echelon distribution inventory system. *Computers & Industrial Engineering, 124*, 88–99.
Alvarez, J. A. L., Buijs, P., Kilic, O. A., & Vis, I. F. (2020). An inventory control policy for liquefied natural gas as a transportation fuel. *Omega, 90*, 101985.
Araya-Sassi, C., Miranda, P. A., & Paredes-Belmar, G. (2018). Lagrangian relaxation for an inventory location problem with periodic inventory control and stochastic capacity constraints. *Mathematical Problems in Engineering, 2018*.
Babai, M. Z., Syntetos, A. A., & Teunter, R. (2010). On the empirical performance of (t, s, s) heuristics. *European Journal of Operational Research, 202*(2), 466–472.
Bashyam, S., & Fu, M. C. (1998). Optimization of (s, S) inventory systems with random lead times and a service level constraint. *Management Science, 44*(12-part-2), S243–S256.
Bellman, R. (1966). Dynamic programming. *Science, 153*(3731), 34–37.
Berry, W. L. (1972). Lot sizing procedures for requirements planning systems: A framework for analysis. *Production and Inventory Managements, 13*(2), 19–34.
Bijvank, M., & Vis, I. F. (2012a). Inventory control for point-of-use locations in hospitals. *Journal of the Operational Research Society, 63*(4), 497–510.
Bijvank, M., & Vis, I. F. (2012b). Lost-sales inventory systems with a service level criterion. *European Journal of Operational Research, 220*(3), 610–618.
Bollapragada, S., & Morton, T. E. (1999). A simple heuristic for computing nonstationary (s, s) policies. *Operations Research, 47*(4), 576–584.
Bookbinder, J. H., & Tan, J.-Y. (1988). Strategies for the probabilistic lot-sizing problem with service-level constraints. *Management Science, 34*(9), 1096–1108.
Bushuev, M. A., Guiffrida, A., Jaber, M. Y., & Khan, M. (2015). A review of inventory lot sizing review papers. *Management Research Review, 38*(3), 283–298.
Cabrera, G., Miranda, P. A., Cabrera, E., Soto, R., Crawford, B., Rubio, J. M., & Paredes, F. (2013). Solving a novel inventory location model with stochastic constraints and inventory control policy. *Mathematical Problems in Engineering, 2013*.
Christou, I., Skouri, K., & Lagodimos, A. (2020). Fast evaluation of a periodic review inventory policy. *Computers & Industrial Engineering*, 106389.
Cunha, P., Raupp, F. M., & Oliveira, F. (2017). A two-stage stochastic programming model for periodic replenishment control system under demand uncertainty. *Computers & Industrial Engineering, 107*, 313–326.
Dillon, M., Oliveira, F., & Abbasi, B. (2017). A two-stage stochastic programming model for inventory management in the blood supply chain. *International Journal of Production Economics, 187*, 27–41.
Dural-Selcuk, G., Rossi, R., Kilic, O. A., & Tarim, S. A. (2019a). The benefit of receding horizon control: Near-optimal policies for stochastic inventory control. *Omega*, 102091.
Dural-Selcuk, G., Rossi, R., Kilic, O. A., & Tarim, S. A. (2019b). The benefit of receding horizon control: Near-optimal policies for stochastic inventory control. *Omega*, 102091.
Fathoni, F. A., Ridwan, A. Y., & Santosa, B. (2019). Development of inventory control application for pharmaceutical product using abc-ved cycle counting method to increase inventory record accuracy. In *Proceedings of the international conference on industrial enterprise and system engineering (ICOIESE 2018)*. Atlantis Press.
Fattahi, M., Mahootchi, M., Moattar Husseini, S., Keyvanshokooh, E., & Alborzi, F. (2015). Investigating replenishment policies for centralised and decentralised supply chains using stochastic programming approach. *International Journal of Production Research, 53*(1), 41–69.
Gutierrez-Alcoba, A., Rossi, R., Martin-Barragan, B., & Hendrix, E. M. (2017). A simple heuristic for perishable item inventory control under non-stationary stochastic demand. *International Journal of Production Research, 55*(7), 1885–1897.
Kilic, O. A., & Tarim, S. A. (2011). An investigation of setup instability in non-stationary stochastic inventory systems. *International Journal of Production Economics, 133*(1), 286–292.
Kilic, O. A., & Tunc, H. (2019). Heuristics for the stochastic economic lot sizing problem with remanufacturing under backordering costs. *European Journal of Operational Research, 276*(3), 880–892.
Lagodimos, A., Christou, I., & Skouri, K. (2012a). Computing globally optimal (s, s, t) inventory policies. *Omega, 40*(5), 660–671.
Lagodimos, A. G., Christou, I., & Skouri, K. (2012b). Optimal (r, nq, t) batch-ordering policy under stationary demand. *International Journal of Systems Science, 43*(9), 1774–1784.
Ma, X., Rossi, R., & Archibald, T. (2019). Stochastic inventory control: A literature review. *IFAC-PapersOnLine, 52*(13), 1490–1495.
Monthatipkul, C., & Yenradee, P. (2008). Inventory/distribution control system in a one-warehouse/multi-retailer supply chain. *International Journal of Production Economics, 114*(1), 119–133.
Özen, U., Doğru, M. K., & Tarim, S. A. (2012). Static-dynamic uncertainty strategy for a single-item stochastic inventory control problem. *Omega, 40*(3), 348–357.
Powell, W. B. (2019). A unified framework for stochastic optimization. *European Journal of Operational Research, 275*(3), 795–821.
Rossi, R., Kilic, O. A., & Tarim, S. A. (2015). Piecewise linear approximations for the static–dynamic uncertainty strategy in stochastic lot-sizing. *Omega, 50*, 126–140.
Rossi, R., Tarim, S. A., Hnich, B., & Prestwich, S. (2008). A global chance-constraint for stochastic inventory systems under service level constraints. *Constraints, 13*(4), 490–517.
Rossi, R., Tarim, S. A., Hnich, B., & Prestwich, S. (2010). Computing the non-stationary replenishment cycle inventory policy under stochastic supplier lead-times. *International Journal of Production Economics, 127*(1), 180–189.
Rossi, R., Tarim, S. A., Hnich, B., & Prestwich, S. (2011). A state space augmentation algorithm for the replenishment cycle inventory policy. *International Journal of Production Economics, 133*(1), 377–384.
Rossi, R., Tarim, S. A., Hnich, B., & Prestwich, S. (2012). Constraint programming for stochastic inventory systems under shortage cost. *Annals of Operations Research, 195*(1), 49–71.
Sani, B., & Kingsman, B. G. (1997). Selecting the best periodic inventory control and demand forecasting methods for low demand items. *Journal of the Operational Research Society, 48*(7), 700–713.
dos Santos, F. S. P., & Oliveira, F. (2019). An enhanced l-shaped method for optimizing periodic-review inventory control problems modeled via two-stage stochastic programming. *European Journal of Operational Research, 275*(2), 677–693.
Scarf, H. (1959). The optimality of (s, s) policies in the dynamic inventory problem.
Schneider, H., & Rinks, D. B. (1991). Empirical study of a new procedure for allocating safety stock in a wholesale inventory system. *International Journal of Production Economics, 24*(1–2), 181–189.
Schneider, H., RNKS, D. B., & Kelle, P. (1995). Power approximations for a two-echelon inventory system using service levels. *Production and Operations Management, 4*(4), 381–400.
Silver, E. A. (1981). Operations research in inventory management: A review and critique. *Operations Research, 29*(4), 628–645.
Tarim, S. A., & Kingsman, B. G. (2004). The stochastic dynamic production/inventory lot-sizing problem with service-level constraints. *International Journal of Production Economics, 88*(1), 105–119.
Tunc, H., Kilic, O. A., Tarim, S. A., & Rossi, R. (2018). An extended mixed-integer programming formulation and dynamic cut generation approach for the stochastic lot-sizing problem. *INFORMS Journal on Computing, 30*(3), 492–506.
Verhoef, P. C., & Sloot, L. M. (2006). Out-of-stock: reactions, antecedents, management solutions, and a future perspective. In *Retailing in the 21st century* (pp. 239–253). Springer.
Wang, H., Yu, Y., Zhang, W., & Hua, Z. (2019). Procurement strategies for lost-sales inventory systems with all-units discounts. *European Journal of Operational Research, 272*(2), 539–548.
Xiang, M., Rossi, R., Martin-Barragan, B., & Tarim, S. A. (2018). Computing non-stationary (s, s) policies using mixed integer linear programming. *European Journal of Operational Research, 271*(2), 490–500.
Zipkin, P. (2008). Old and new methods for lost-sales inventory systems. *Operations Research, 56*(5), 1256–1263.