

# Constraint programming for stochastic inventory systems under shortage cost

Roberto Rossi · S. Armagan Tarim · Brahim Hnich · Steven Prestwich

Published online: 30 July 2011

© The Author(s) 2011. This article is published with open access at Springerlink.com

**Abstract** One of the most important policies adopted in inventory control is the replenishment cycle policy. Such a policy provides an effective means of damping planning instability and coping with demand uncertainty. In this paper we develop a constraint programming approach able to compute optimal replenishment cycle policy parameters under non-stationary stochastic demand, ordering, holding and shortage costs. We show how in our model it is possible to exploit the convexity of the cost-function during the search to dynamically compute bounds and perform cost-based filtering. Our computational experience show the effectiveness of our approach. Furthermore, we use the optimal solutions to analyze the quality of the solutions provided by an existing approximate mixed integer programming approach that exploits a piecewise linear approximation for the cost function.

**Keywords** Inventory control · Constraint programming · Decision making under uncertainty · Replenishment cycle policy · Non-stationary demand · Shortage cost

---

This paper is an extended version of the work presented in Rossi et al. (2007).

R. Rossi (✉)

Logistics, Decision and Information Sciences, Wageningen UR, Wageningen, The Netherlands  
e-mail: [roberto.rossi@wur.nl](mailto:roberto.rossi@wur.nl)

S.A. Tarim

Department of Management, Hacettepe University, Ankara, Turkey  
e-mail: [armagan.tarim@hacettepe.edu.tr](mailto:armagan.tarim@hacettepe.edu.tr)

B. Hnich

Faculty of Computer Science, Izmir University of Economics, Izmir, Turkey  
e-mail: [brahim.hnich@ieu.edu.tr](mailto:brahim.hnich@ieu.edu.tr)

S. Prestwich

Cork Constraint Computation Centre, University College, Cork, Ireland  
e-mail: [s.prestwich@4c.ucc.ie](mailto:s.prestwich@4c.ucc.ie)

## 1 Introduction

Much of the inventory control literature concerns the computation of optimal replenishment policies under demand uncertainty. One of the most important policies adopted is the  $(R, S)$  policy (also known as the *replenishment cycle* policy). A detailed discussion on the characteristics of  $(R, S)$  can be found in de Kok (1991). In this policy a replenishment is placed every  $R$  periods to raise the inventory position to the order-up-to-level  $S$ . This provides an effective means of damping planning instability—deviations in planned orders, also known as *nervousness* (de Kok and Inderfurth 1997, Heisig 2002)—and coping with demand uncertainty. As pointed out by Silver et al. (1998, pp. 236–237),  $(R, S)$  is particularly appealing when items are ordered from the same supplier or require resource sharing. In these cases all items in a coordinated group can be given the same replenishment period. Periodic review also allows a reasonable prediction of the level of the workload on the staff involved, and is particularly suitable for advanced planning environments and risk management (Tang 2006). For these reasons  $(R, S)$  is a popular inventory policy.

As pointed out by Graves (1999) one major theme in the continuing development of inventory theory is to incorporate more realistic assumptions about product demand into inventory models. In most industrial contexts, demand is uncertain and hard to forecast. Many demand histories behave like random walks that evolve over time with frequent changes in their directions and rates of growth or decline. Furthermore, as product life cycles get shorter, the randomness and unpredictability of these demand processes have become even greater. In practice, for such demand processes, inventory managers often rely on forecasts based on a time series of prior demand, such as a weighted moving average. Typically these forecasts are predicated on a belief that the most recent demand observations are the best predictors for future demand. An important class of stochastic production/inventory control problems therefore assumes a non-stationary demand process (see e.g. Bayraktar and Ludkovski 2010). Under this assumption the  $(R, S)$  policy takes the non-stationary form  $(R_n, S_n)$  where  $R_n$  denotes the length of the  $n$ th replenishment cycle and  $S_n$  the corresponding order-up-to-level.

To compute the *near* optimal  $(R_n, S_n)$  policy parameters, Tarim and Kingsman (2006) proposed a mixed integer programming (MIP) formulation using a piecewise linear approximation to a complex cost function. Tarim and Kingsman assume a fixed procurement cost each time a replenishment order is placed, whatever the size of the order, and a linear holding cost on any unit carried over in inventory from one period to the next. Instead of including a service level constraint—the probability that at the end of every time period the net inventory is not negative set at least to a certain value (see Bookbinder and Tan 1988; Tarim and Kingsman 2004; Tarim and Smith 2008; Tarim et al. 2009b; Rossi et al. 2008; Tempelmeier 2007; Pujawan and Silver 2008; and Rossi et al. 2011 for  $(R_n, S_n)$  under a service level constraint)—their model employs a shortage cost scheme. So far no exact approach exists for computing  $(R_n, S_n)$  policy parameters under a shortage cost scheme. In fact, as shown in Tarim and Kingsman (2006), the cost structure is complex in this case and it differs significantly from the one under a service level constraint. Tarim and Smith (2008) proposed a constraint programming (CP) model under a service level constraint. In their work it was shown that not only CP is able to provide a more compact formulation than the MIP one, but that it is also able to perform faster and to take advantage of dedicated pre-processing techniques that reduce the size of decision variable domains. Moreover dedicated cost-based filtering techniques were proposed in Tarim et al. (2009b) for the same model, these techniques are able to improve the performance of several orders of magnitude.

In this paper, instead of employing a piecewise linear approximation to the expected total cost function, we give an *exact* CP formulation of the  $(R_n, S_n)$  inventory control problem under shortage cost scheme. Furthermore we propose a dedicated *cost-based filtering* method (Focacci and Milano 1999) to improve the performance of the search. Our contribution is two-fold: we can now efficiently obtain provably optimal solutions for the  $(R_n, S_n)$  inventory control problem under shortage costs and we can gauge the accuracy of the piecewise linear approximation proposed by Tarim and Kingsman (2006). Our computational experience shows the effectiveness of our approach.

This work is structured as follows. In Sect. 2 we introduce the problem of interest and we develop the mathematical background that will be employed in order to develop our CP model. In Sect. 3 we introduce CP. In Sect. 4 we discuss our CP model for computing optimal  $(R_n, S_n)$  policy parameters. In Sect. 5 we introduce cost-based filtering techniques for improving the efficiency of the search process. In Sect. 6 we present our computational experience. In Sect. 7 we draw conclusions.

## 2 Problem definition and $(R_n, S_n)$ policy

We consider the single stocking location, single product inventory problem over a finite planning horizon of  $N$  periods. The demand  $d_t$  in period  $t$  is assumed to be a normally distributed random variable with known probability density function (PDF)  $g_t(d_t)$ . Demand is assumed to occur instantaneously at the beginning of each period. The mean rate of demand may vary from period to period. Demands in different time periods are assumed to be independent. Demands occurring when the system is out of stock are assumed to be back-ordered and satisfied as soon as the next replenishment order arrives.

A fixed positive holding cost  $h$  is incurred on any unit carried over in inventory from one period to the next. A fixed positive shortage cost  $s$  is incurred for each unit of demand that is back-ordered. A fixed positive procurement (ordering or set-up) cost  $a$  is incurred each time a replenishment order is placed, whatever the size of the order. In addition to the fixed ordering cost, a positive proportional direct item cost  $v$  is incurred.

For convenience, and without loss of generality, the initial inventory level is set to zero. We assume that the delivery lead-time is zero, and thus orders are delivered immediately. It is assumed that negative orders are not allowed, so that if the actual stock exceeds the order-up-to-level for that review, this excess stock is carried forward and does not return to the supply source. However, such occurrences are regarded as rare events and accordingly the cost of carrying the excess stock is ignored. The above assumptions hold for the rest of this paper.

The general multi-period production/inventory problem with stochastic demands can be formulated as finding the timing of the stock reviews and the size of non-negative replenishment orders,  $X_t$  in period  $t$ , minimizing the expected total cost over a finite planning horizon of  $N$  periods:

$$\min E\{TC\} = \int_{d_1} \int_{d_2} \dots \int_{d_N} \sum_{t=1}^N (a\delta_t + vX_t + hI_t^+ + sI_t^-) \\ \times g_1(d_1) \dots g_N(d_N) d(d_1) \dots d(d_N) \quad (1)$$

$$\text{subject to, for } t = 1, \dots, N,$$

$$X_t > 0 \Rightarrow \delta_t = 1 \quad (2)$$

$$I_t = \sum_{i=1}^t (X_i - d_i) \tag{3}$$

$$I_t^+ = \max(0, I_t) \tag{4}$$

$$I_t^- = -\min(0, I_t) \tag{5}$$

$$X_t, I_t^+, I_t^- \in \mathbb{R}^+ \cup \{0\}, \quad I_t \in \mathbb{R}, \quad \delta_t \in \{0, 1\} \tag{6}$$

where

- $d_t$ : the demand in period  $t$ , a normal random variable with PDF  $g_t(d_t)$ ,
- $a$ : the fixed ordering cost,
- $v$ : the proportional direct item cost,
- $h$ : the proportional stock holding cost,
- $s$ : the proportional shortage cost,
- $\delta_t$ : a  $\{0,1\}$  variable that takes the value of 1 if a replenishment occurs in period  $t$  and 0 otherwise,
- $I_t$ : the inventory level at the end of period  $t$ ,  $-\infty < I_t < +\infty$ ,  $I_0 = 0$
- $I_t^+$ : the excess inventory at the end of period  $t$  carried over to the next period,
- $I_t^-$ : the shortages at the end of period  $t$ , or magnitude of negative inventory,
- $X_t$ : the replenishment order placed and received in period  $t$ ,  $X_t \geq 0$ .

The proposed non-stationary  $(R,S)$  policy consists of a series of review times and associated order-up-to-levels. Consider a review schedule which has  $m$  reviews scheduled, respectively, at  $\{T_1, T_2, \dots, T_m\}$ , where  $T_j > T_{j-1}$ , over the  $N$ -period planning horizon. For convenience let  $T_1 = 1$  and  $T_{m+1} = N + 1$ . In Tarim and Kingsman (2006), the decision variable  $X_{T_i}$  is expressed in terms of a new variable  $S_i \in \mathbb{Z}$ , according to the relation  $X_{T_i} = \max(0, S_i - I_{-1})$ .  $S_i$  may be interpreted as the opening stock level for period  $t$ , if there is no review scheduled in this period (i.e.  $t \notin \{T_1, T_2, \dots, T_m\}$  and, clearly,  $X_t = 0$ ), and as the order-up-to-level for the  $i$ -th review period  $T_i$ , if there is a review scheduled in period  $t$  (i.e.  $t \in \{T_1, T_2, \dots, T_m\}$  and  $X_t \geq 0$ ). Let  $D_{1,t_2} = \sum_{j=1}^{t_2} d_j$ . According to the above variable change the expected cost function, (1), is written as the summation of  $m$  intervals,  $T_i$  to  $T_{i+1}$  for  $i = 1, \dots, m$ ,

$$\begin{aligned} \min \quad E\{TC\} &= \sum_{i=1}^m \left( a\delta_{T_i} + \sum_{t=T_i}^{T_{i+1}-1} E\{C_{T_i,t}\} \right) \\ &+ vI_N + v \int_{D_{1,N}} D_{1,N} \times g_{1,N}(D_{1,N})d(D_{1,N}), \end{aligned} \tag{7}$$

where  $g_{i,j}(D_{i,j})$  is the PDF of  $D_{i,j}$  and  $E\{C_{T_i,t}\}$  is defined as

$$\int_{-\infty}^{S_{T_i}} h(S_{T_i} - D_{T_i,t})g(D_{T_i,t})d(D_{T_i,t}) - \int_{S_{T_i}}^{\infty} s(S_{T_i} - D_{T_i,t})g(D_{T_i,t})d(D_{T_i,t}). \tag{8}$$

The term  $v \int_{D_{1,N}} D_{1,N} \times g(D_{1,N})d(D_{1,N})$  is constant and can therefore be ignored in the optimization model.  $E\{C_{T_i,t}\}$  is the expected cost function of a single-period inventory problem where the single-period demand is  $D_{T_i,t}$ . Since  $S_{T_i}$  may be interpreted as the order-up-to-level for the  $i$ -th review period  $T_i$  and  $S_{T_i} - D_{T_i,t}$  is the end of period inventory for the “single-period” with demand  $D_{T_i,t}$ , the expected total sub-costs  $E\{C_{T_i,t}\}$  are the sums of

single-period inventory costs where the demands are the cumulative demands over increasing periods.

By dropping the  $T_i$  and  $t$  subscripts in (8) we obtain the following well-known expression for the expected total cost of a single-period newsvendor problem (Hadley and Whitin 1964, pp. 297–299),

$$E\{TC\} = h \int_{-\infty}^S (S - D)g(D)d(D) - s \int_S^{\infty} (S - D)g(D)d(D). \tag{9}$$

In (9) we consider two cost components: holding cost on the positive end of period inventory and shortage cost for any back-ordered demand.

Let  $G(\cdot)$  be the cumulative distribution function of the demand in our single-period newsvendor problem. A known result in inventory theory (Hadley and Whitin 1964) is convexity of (9). The so-called *Critical Ratio*,  $\frac{s}{s+h}$ , can be seen as the service level  $\beta$  (i.e. probability that at the end of the period the inventory level is non-negative) provided to customers when we fix the order-up-to-level  $S$  to the optimal value  $S^*$  that minimizes expected holding and shortage costs (9). By assuming  $G(\cdot)$  to be strictly increasing, we can compute the optimal order-up-to-level as  $S^* = G^{-1}(\frac{s}{s+h})$ .

### 2.1 Stochastic cost component in single-period newsvendor

We now aim to characterize the cost of the policy that orders  $S^*$  units to meet the demand in our single-period newsvendor problem. Such a problem has been widely studied in the inventory control literature (Silver et al. 1998).

Since the demand  $D$  is assumed to be normal with mean  $\mu$  and standard deviation  $\sigma$ , then we can write  $D = \mu + \sigma Z$ , where  $Z$  is a standard normal random variable. Let  $\Phi(z) = \Pr(Z \leq z)$  be the cumulative distribution function of the standard normal random variable. Since  $\Phi(\cdot)$  is strictly increasing,  $\Phi^{-1}(\cdot)$  is uniquely defined. Let  $z_\beta = \Phi^{-1}(\beta)$ , since  $\Pr(D \leq \mu + z_\beta\sigma) = \Phi(z_\beta) = \beta$ , it follows that  $S^* = \mu + z_\beta\sigma$ . The quantity  $z_\beta$  is known as the safety factor and  $S^* - \mu = z_\beta\sigma$  is known as the safety stock. It can be shown (Hadley and Whitin 1964) that

$$\int_{S^*}^{\infty} (S^* - D)g(D)d(D) = E\{D - S^*\}^+ = \sigma E\{Z - z_\beta\}^+ = \sigma[\phi(z_\beta) - (1 - \beta)z_\beta], \tag{10}$$

where  $\phi(\cdot)$  is the PDF of the standard normal random variable. Let  $E\{S^* - D\}^+ = \int_{-\infty}^{S^*} (S - D)g(D)d(D)$ , it follows

$$\begin{aligned} E\{TC(S^*)\} &= h \cdot E\{S^* - D\}^+ + s \cdot E\{D - S^*\}^+ \\ &= h \cdot (S^* - \mu) + (h + s)E\{D - S^*\}^+ \\ &= h z_\beta \sigma + (h + s)\sigma E\{Z - z_\beta\}^+ \\ &= h z_\beta \sigma + (h + s)\sigma[\phi(z_\beta) - (1 - \beta)z_\beta] \\ &= (h + s)\sigma\phi(z_\beta). \end{aligned} \tag{11}$$

The last expression  $(h + s)\sigma\phi(z_\beta)$  holds only for the optimal order-up-to-level  $S^*$  that provides the service level  $\beta = \frac{s}{s+h}$  computed from the *critical ratio* (CR). Instead, expression

$$h z_\alpha \sigma + (h + s)\sigma[\phi(z_\alpha) - (1 - \alpha)z_\alpha] \tag{12}$$

can be used to compute the expected total cost for any given level  $S$  such that  $\alpha = \Phi(\frac{S-\mu}{\sigma})$ .

### 2.2 Stochastic cost component in multiple-period newsvendor

The considerations in the former sections refer to a single-period problem, but they can be easily extended to a replenishment cycle  $R(i, j)$  that covers the period span  $i, \dots, j$ . In Levi et al. (2006) it is possible to find a discussion on multi-period newsvendor problems and a sampling-based heuristic approach to find near-optimal solutions. In contrast the approach we propose is exact.

Consider a normally distributed demand in each period with PDF  $g_i(d_i), \dots, g_j(d_j)$ . The cost for a multi-period replenishment cycle, when ordering costs are neglected, can be expressed as

$$E\{TC\} = \sum_{k=i}^j \left( h \int_{-\infty}^S (S - D_{i,k}) g_{i,k}(D_{i,k}) d(D_{i,k}) - s \int_S^{\infty} (S - D_{i,k}) g_{i,k}(D_{i,k}) d(D_{i,k}) \right). \tag{13}$$

Since demands are independent and normally distributed in each period, the term  $g_{i,j}(D_{i,j})$  can be easily computed (Fortuin 1980) once the demand in each period  $d_i, \dots, d_j$  is known. It is now easy to apply the same rule as before and compute the second derivative of this expression:

$$\frac{d^2}{dS^2} E\{TC\} = \sum_{k=i}^j (h \cdot g_{i,k}(S) + s \cdot g_{i,k}(S)), \tag{14}$$

which is again a positive function of  $S$ , since  $g_{i,k}(S)$  are PDFs and both holding and shortage cost are assumed to be positive. The expected cost of a single replenishment cycle therefore remains convex in  $S$  regardless of the periods covered.

Unfortunately, it is not possible to compute the CR as before, by using a simple algebraic expression to obtain the optimal  $S^*$  which minimizes the expected cost. Nevertheless, since the cost function is convex, it is still possible to compute  $S^*$  efficiently. In fact, (12) can be extended in the following way to compute the cost for the replenishment cycle  $R(i, j)$  as a function of the opening inventory level  $S$ :

$$\sum_{k=i}^j (h z_{\alpha(i,k)} \sigma_{i,k} + (h + s) \sigma_{i,k} [\phi(z_{\alpha(i,k)}) - (1 - \alpha(i, k)) z_{\alpha(i,k)}]), \tag{15}$$

where  $G_{i,k}(\cdot)$  and  $\sigma_{i,k}$  denote, respectively, the cumulative distribution function (CDF) and the standard deviation of  $d_i + \dots + d_k$ ;  $\alpha(i, k) = G_{i,k}(S)$ ; and  $z_{\alpha(i,k)} = \Phi^{-1}(\alpha(i, k))$ . Therefore we have  $j - i + 1$  cost components: the holding and shortage costs at the end of period  $i, i + 1, \dots, j$ . For each possible replenishment cycle we can efficiently compute the optimal  $S^*$  that minimizes our cost function by using gradient based methods for convex optimization such as Newton’s method. Note that the complete expression for the cost of replenishment cycles that start in period  $i \in \{1, \dots, N\}$  and end in period  $N$  is

$$\sum_{k=i}^N (h z_{\alpha(i,k)} \sigma_{i,k} + (h + s) \sigma_{i,k} [\phi(z_{\alpha(i,k)}) - (1 - \alpha(i, k)) z_{\alpha(i,k)}]) + v \left( S - \sum_{k=i}^N d_k \right). \tag{16}$$

In fact, for this set of replenishment cycles we must also consider the unit cost component. Once  $S^*$  is known, by subtracting the expected demand over the replenishment cycle  $R(i, j)$  we obtain the optimal *expected buffer stock level*  $b(i, j) = S^* - \sum_{t=1}^j \tilde{d}_t$  required for such a replenishment cycle in order to minimize holding and shortage cost. Note that every other choice for buffer stock level will produce a higher expected total cost for  $R(i, j)$ .

### 2.3 Upper-bound for opening inventory levels

We now propose an *upper bound* for the value of the opening inventory level in each period  $t \in \{1, \dots, N\}$ . Firstly we ignore the direct item cost  $v$ , in fact from (7) it is trivial to see that  $v$  may only decrease the opening inventory level for the last replenishment cycle scheduled. We consider a single replenishment cycle covering the whole planning horizon. If we relax the original problem formulation and we ignore holding and shortage cost components at the end of each period  $t \in \{1, \dots, N - 1\}$ , the resulting model will reflect a single period newsvendor problem. In this problem we incur holding and shortage cost only at the end of the last period  $N$  and the stochastic demand is given by the sum of the demand distributions in each period of our planning horizon. The optimal buffer stock  $b(1, N)$  required to optimize the convex cost for this problem can be easily computed, as seen, by means of the critical ratio. It is easy to see that, since we relaxed holding and shortage costs for each period  $t \in \{1, \dots, N - 1\}$ , then for each period  $t \in \{1, \dots, N\}$ , every value in the domain of  $S_t$  greater than  $\sum_t \tilde{d}_t + b(1, N)$  is suboptimal and should be removed from the domain. In fact, since we assume positive shortage and holding costs, opening inventory levels for this replenishment cycle may only be decreased by the additional cost components in the original model.

Moreover the upper bounds computed are still valid if the planning horizon is covered by more than a single replenishment cycle. The reason is the following. If the planning horizon is covered by a number of replenishment cycles, again it is possible to apply a similar reasoning and it is possible to reduce each replenishment cycle  $R(i, j)$ , covering periods  $\{i, \dots, j\}$ , to a single period newsvendor problem, by ignoring holding and shortage costs for each period  $t \in \{i, \dots, j - 1\}$  and by considering only the cost component of the last period  $j$ . Then for each replenishment cycle  $R(i, j)$  we will easily obtain a buffer stock  $b(i, j)$ , by means of the critical ratio. Since  $b(i, j)$  is increasing, that is  $b(i, j) \leq b(i, j + 1)$ , as shown in Tarim and Smith (2008), obviously opening inventory levels computed in this case will be lower than those computed for the former case where a single replenishment cycle covers the whole planning horizon. Furthermore, we recall that also in this case opening inventory levels may only be decreased when the additional holding and shortage cost components for other periods are reintroduced in the model. It directly follows that the upper bounds computed are valid for the original model.

### 2.4 Lower-bound for expected closing inventory levels

A *lower bound* for the value of the expected closing inventory level in each period  $t \in \{1, \dots, N\}$ , that is opening inventory level minus expected demand, can be computed by considering every possible buffer stock  $b(i, j)$  required to optimize the convex cost of a single replenishment cycle  $R(i, j)$ , independently of the other cycles that are planned. The lower bound will be the minimum value among all these possible buffer values for  $j \in \{1, \dots, N\}$  and  $i \in \{1, \dots, j\}$ .

### 3 Constraint programming

Before introducing our CP model for the problem discussed, we now provide some formal background on CP. CP (Rossi et al. 2006) is a declarative programming paradigm in which relations between decision variables are stated in the form of constraints. Informally speaking, constraints specify the properties of a solution.

Formally, a Constraint Satisfaction Problem (CSP) (Apt 2003; Rossi et al. 2006) is a triple  $\langle V, C, D \rangle$ , where  $V$  is a set of decision variables each with a discrete domain of values  $D(V_k)$ , and  $C$  is a set of constraints stating allowed combinations of values for subsets of variables in  $V$ . Finding a solution to a CSP means assigning values to variables from the domains without violating any constraint in  $C$ . We may also be interested in finding a feasible solution that minimize (maximize) the value of a given objective function over a subset of the variables. In this case, the problem is denoted as Constraint Optimization Problem (COP).

The constraints used in CP are of various kinds: logic constraints (i.e. “ $x$  or  $y$  is true”, where  $x$  and  $y$  are boolean decision variables), linear constraints, and *global constraints* (Regin 2003).

A global constraint captures a relation among a non-fixed number of variables. One of the most well known global constraints is the `alldiff` constraint (Regin 1994), that can be enforced on a set of decision variables in order to guarantee that no two variables are assigned the same value.

With each constraint, CP associates a *filtering algorithm* able to remove provably infeasible or suboptimal values from the domains of the decision variables that are constrained and, therefore, to enforce some degree of *consistency* (see Rossi et al. 2006). These filtering algorithms are repeatedly called until no more values are pruned. This process is called *constraint propagation*.

*Example 1* Consider the following CSP. Decision variables  $x_1, x_2, x_3$  with domains  $D_1 = \{0, 1, 2, 3\}$ ,  $D_2 = \{0, 1, 2, 3, 4, 5\}$ ,  $D_3 = \{0, 1\}$ , and

$$x_1 \geq 3, \quad (17)$$

$$x_1 + x_2 = 8, \quad (18)$$

$$(x_2 > 0) \leftrightarrow (x_3 = 1). \quad (19)$$

We apply constraint propagation until no new deduction can be made:

$$\begin{array}{l} x_1 \in \{0, 1, 2, 3\} \\ x_2 \in \{0, 1, 2, 3, 4, 5\} \\ x_3 \in \{0, 1\} \end{array} \quad (17) \quad \longrightarrow \quad \begin{array}{l} x_1 \in \{3\} \\ x_2 \in \{0, 1, 2, 3, 4, 5\} \\ x_3 \in \{0, 1\} \end{array} \quad (18) \quad \longrightarrow \quad \begin{array}{l} x_1 \in \{3\} \\ x_2 \in \{5\} \\ x_3 \in \{0, 1\} \end{array} \quad (19) \quad \longrightarrow \quad \begin{array}{l} x_1 \in \{3\} \\ x_2 \in \{5\} \\ x_3 \in \{1\}. \end{array}$$

In addition to constraints and filtering algorithms, constraint solvers also feature some sort of heuristic *search engine* (e.g. a backtracking algorithm). The search algorithm first selects a variable and then decides how to explore its domain. The process is known as “branching” and it is guided by *variable* and *value ordering heuristics*. The ordering imposed by these heuristics guides the search process and has a great impact on it. During the search, the constraint solver exploits filtering algorithms in order to proactively prune parts of the search space that cannot lead to a feasible or to an optimal solution.



*Example 2* We present the solution process of CP, using COP  $P_0$ :

$$\begin{aligned} x_1 &\in \{3, 8\}, & x_2 &\in \{0, 1, 2, 3, 4, 5\}, & x_3 &\in \{0, 1\}, & z &\in \{0, \dots, 10\}, \\ \text{minimize } & z, \\ z &= x_1 + 6x_3, \\ x_1 &\geq 3, \\ x_1 + x_2 &= 8, \\ (x_2 > 0) &\leftrightarrow (x_3 = 1). \end{aligned}$$

To build a search tree, we apply the lexicographic variable and value ordering heuristic. As a search strategy we use Depth First Search. At each node we apply constraint propagation.

The CSP  $P_0$  is the root of our search tree. We first apply constraint propagation to  $P_0$ . It follows that

$$x_1 \in \{3, 8\}, \quad x_2 \in \{0, 5\}, \quad x_3 \in \{0, 1\}, \quad z \in \{8, 9\}.$$

We select the lexicographically least variable,  $x_1$ , and the lexicographically least value in its domain, i.e. 3, and generate the descendant  $P_1$ , where  $P_1 = P_0 \cup x_1 \in \{3\}$ , that is the union of  $P_0$  and the “partial” assignment  $x_1 \in \{3\}$ .

We descend to node  $P_1$  and apply constraint propagation. It follows that

$$x_1 \in \{3\}, \quad x_2 \in \{5\}, \quad x_3 \in \{1\}, \quad z \in \{9\}.$$

We have found a solution with  $z = 9$ . Hence we add to all CSPs the constraint  $z < 9$ .

Next, we backtrack to  $P_0$ , we select the next lexicographically least value not yet considered in the domain of  $x_1$ , i.e. 8, we generate the descendant  $P_2$ , where  $P_2 = P_0 \cup x_1 \in \{8\}$ , we descend to  $P_2$  and apply constraint propagation. It follows that

$$x_1 \in \{8\}, \quad x_2 \in \{0\}, \quad x_3 \in \{0\}, \quad z \in \{8\}.$$

We have found a solution with  $z = 8$ . Hence we add to all CSPs the constraint  $z < 8$ .

Next we backtrack to  $P_0$  and stop because all its descendants have been visited. Finally, we return the optimal solution we have found in leaf  $P_2$ .

In CP, *optimization-oriented global constraints* constitute an important class of constraints. These global constraints embed an optimization component, representing a proper relaxation of the constraint itself, into a global constraint (Focacci et al. 2002). The relaxation employed can be a continuous relaxation, as in the examples provided in Focacci et al. (2002), a DP relaxation, as discussed in Focacci and Milano (2001), or it can be any other suitable relaxation.

The optimization component provides three pieces of information: (a) the optimal solution of the relaxed problem; (b) the optimal value of this solution representing an upper bound on the original problem objective function; (c) a *gradient function*  $\text{grad}(V, v)$ , which returns for each variable-value pair  $(V, v)$  an optimistic evaluation of the profit obtained if  $v$  is assigned to  $V$ . These pieces of information are exploited both for propagation purposes and for guiding the search.

CP has recently proved to be a very effective technique for decision making under uncertainty (see e.g. Walsh 2002; Hnich et al. 2009; Tarim et al. 2009a). A complete survey of CP approaches for decision making under uncertainty is given in Hnich et al. (2011).

### 4 Deterministic equivalent CP formulation

Building upon the considerations in Sect. 2 it is easy to construct a *deterministic equivalent* CP formulation for the non-stationary  $(R_n, S_n)$  policy under stochastic demand, ordering cost, holding and shortage cost. (For a detailed discussion on deterministic equivalent modeling in stochastic programming see Birge and Louveaux 1997.)

In order to correctly compute the expected total cost for a replenishment cycle  $R(i, j)$  with opening inventory level  $S_i$ , we must build a special-purpose constraint `objConstraint(·)` that dynamically computes such a cost by means of an extended version of (15)

$$C(S_i, i, j) = a + \sum_{k=i}^j (hz_{\alpha(i,k)}\sigma_{i,k} + (h + s)\sigma_{i,k}[\phi(z_{\alpha(i,k)}) - (1 - \alpha(i, k))z_{\alpha(i,k)}]) \tag{20}$$

that considers the ordering cost  $a$  associated with a replenishment cycle.

Intuitively, within this constraint the expected total cost for a certain replenishment plan will be computed as the sum of all the expected total costs for replenishment cycles in the solution, plus the respective ordering costs. `objConstraint(·)` also computes the optimal expected buffer stock level  $b(i, j)$  for every replenishment cycle  $R(i, j)$  identified by a partial assignment for  $\delta_{k \in \{1, \dots, N\}}$  variables.

A *deterministic equivalent* CP formulation is

$$\min E\{TC\} = C \tag{21}$$

subject to

$$\text{objConstraint}(C, \tilde{I}_1, \dots, \tilde{I}_N, \delta_1, \dots, \delta_N, d_1, \dots, d_N, a, h, s) \tag{22}$$

and for  $t = 1, \dots, N$

$$\tilde{I}_t + \tilde{d}_t - \tilde{I}_{t-1} \geq 0 \tag{23}$$

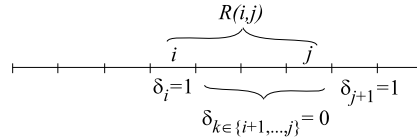
$$\tilde{I}_t + \tilde{d}_t - \tilde{I}_{t-1} > 0 \implies \delta_t = 1 \tag{24}$$

$$\tilde{I}_t \in \mathbb{R}, \quad \delta_t \in \{0, 1\}. \tag{25}$$

Each  $\tilde{d}_t$  represents the expected value of the demand in a given period  $t$  according to its PDF  $g_t(d_t)$ . The binary decision variables  $\delta_t$  state whether a replenishment is fixed for period  $t$  ( $\delta_t = 1$ ) or not ( $\delta_t = 0$ ). Each decision variable  $\tilde{I}_t$  represents the expected closing inventory level at the end of period  $t$ .

In the above model the domain of  $\tilde{I}_t$  ranges over  $\mathbb{R}$ . According to the discussion in Sects. 2.3 and 2.4 it is possible to enforce an upper and a lower bound for the domain of each decision variables  $\tilde{I}_t$ . Nevertheless, the interval obtained still contains an infinite number of possible real values. According to our discussion in Sect. 3, in CP decision variables are assumed to range over finite domains. This assumption is related to the fact that a standard CP solver cannot branch on a domain containing an infinite amount of possible values. For this reason, a key observation is needed in order to make sure that our model can be processed by a CP solver. It should be noted that—according to the discussion in the rest of this section—when all the  $\delta_t$  variables have been assigned, the optimal value for each  $\tilde{I}_t$  can be

**Fig. 1** A replenishment cycle  $R(i, j)$  is identified by the current partial assignment for  $\delta_i$  variables



immediately computed within our `objConstraint`. For this reason, the search procedure can simply branch on the  $\delta_t$  variables and does not have to branch on the continuous variable, which are therefore excluded from the variable selection heuristic. Bound propagation can be employed in order to correctly propagate constraints (23) and (24), and—as we will see in the rest of this section—within `objConstraint`, since they all involve continuous variables.

Equation (23) enforces a no-buy-back condition, which means that received goods cannot be returned to the supplier. As a consequence of this the expected inventory level at the end of period  $t$  must be no less than the expected inventory level at the end of period  $t - 1$  minus the expected demand in period  $t$ . Equation (24) expresses the replenishment condition. We have a replenishment if the expected inventory level at the end of period  $t$  is greater than the expected inventory level at the end of period  $t - 1$  minus the expected demand in period  $t$ . This means that we received some extra goods as a consequence of an order. The objective function (21) minimizes the expected total cost over the given planning horizon. `objConstraint(·)` dynamically computes buffer stocks and it assigns to  $C$  the expected total cost related to a given assignment for replenishment decisions, depending on the demand distribution in each period and on the given combination for problem parameters  $a, h, s$ .

In order to propagate `objConstraint(·)`, during the search we wait for a partial assignment involving some or all the  $\delta_t$  variables. In particular, we look for an assignment where there exists some  $i$  s.t.  $\delta_i = 1$ , some  $j > i$  s.t.  $\delta_{j+1} = 1$  and for every  $k, i < k \leq j, \delta_k = 0$ . This will uniquely identify a replenishment cycle  $R(i, j)$  (Fig. 1).

There may be more replenishment cycles associated to a partial assignment. If we consider each  $R(i, j)$  identified by the current assignment, it is easy to minimize the convex cost function already discussed, and to find the optimal expected buffer stock  $b(i, j)$  for this particular replenishment cycle independently on the others. By independently computing the optimal expected buffer stock  $b(i, j)$  for every replenishment cycle identified, two possible situations may arise:

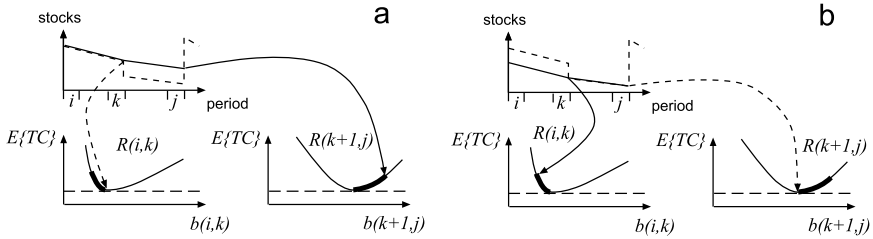
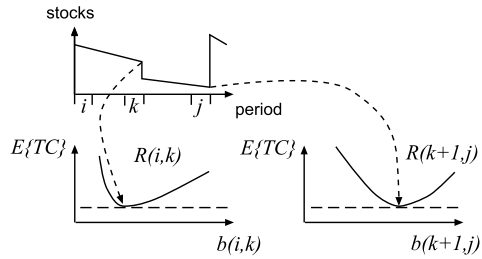
- the buffer stock configuration obtained satisfies every inventory conservation constraint (23) or
- for some pair of subsequent replenishment cycles this constraint is violated (Fig. 2). In other words, we observe an expected negative order quantity.

If the latter situation arises, we can adopt a fast convex optimization procedure to compute a feasible buffer stock configuration with minimum cost. The key idea is to identify two possible limit situations:

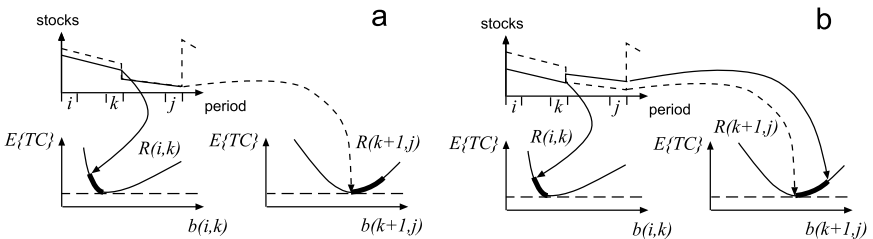
- we increase the opening inventory level of the second cycle, thus incurring a higher overall cost for it, to preserve optimality of the first cycle (Fig. 3a), or
- we decrease the buffer stock of the first replenishment cycle, thus incurring a higher overall cost for it, to preserve optimality of the second cycle cost (Fig. 3b).

A key observation is that, when negative order quantity scenarios arise, at optimality the expected closing inventory levels of the first and the second cycle lie in the interval delimited

**Fig. 2** The expected total cost of both replenishment cycles is minimized, but the inventory conservation constraint is violated between  $R(i, k)$  and  $R(k + 1, j)$



**Fig. 3** Feasible limit situations when negative order quantity scenarios arise



**Fig. 4** Infeasible (a) and suboptimal (b) plans realized when the opening inventory level of the second cycle is not equal to the expected closing inventory level of the first cycle

by the two situations described. This directly follows from the convexity of both the cost functions. Moreover, the expected closing inventory level of the first cycle must be equal to the opening inventory level of the second cycle. In fact, if they are not, then either the first cycle has an expected closing inventory level higher than the opening inventory level of the second cycle and the solution is not feasible (Fig. 4a), or the first cycle has an expected closing inventory level smaller than the opening inventory level of the second cycle. In the latter case we can decrease the overall cost by choosing a smaller opening inventory level for the second cycle (Fig. 4b).

The algorithm for computing optimal buffer stock configurations in presence of negative order quantity scenarios simply exploits the linear dependency between the opening inventory level of the second cycle and the expected closing inventory level of the first cycle. Due to this dependency the overall cost is still convex in  $b(i, k)$  (or equivalently in  $b(k + 1, j)$ , since they are linearly dependent) and we can apply any convex optimization technique to find the optimal buffer stock configuration. Note that this reasoning still holds in a recursive process. Therefore, we can optimize buffer stock for two subsequent replenishment cycles, then we can treat these as a new single replenishment cycle, since their buffer stocks are

linearly dependent, and repeat the process in order to consider the next replenishment cycle if a negative order quantity scenario arises.

Once buffer stocks are known, we can apply (20) to the opening inventory level  $S_i = \tilde{d}_i + \dots + \tilde{d}_j + b(i, j)$  and compute the cost  $C(S_i, i, j)$  associated to a given replenishment cycle. Since the cost function in (20) is convex and we handle negative order quantity scenarios, a lower bound for the expected total cost associated to the current partial assignment for  $\delta_t, t = 1, \dots, N$  variables is now given by the sum of all the cost components  $C(S_i, i, j)$ , for each replenishment cycle  $R(i, j)$  identified by the assignment. Furthermore this bound is tight if all the  $\delta_t$  variables have been assigned. `objConstraint(·)` exploits this property in order to incrementally compute a lower bound for the cost of the current partial assignment for  $\delta_t$  variables. Note that *only* when every  $\delta_t$  variable is ground the lower bound for the cost becomes tight and buffer stocks computed for each replenishment cycle identified can be assigned to the respective  $\tilde{I}_t$  variables. Finally, in order to consider the unit variable cost  $v$  we must add the term  $v \cdot \tilde{I}_N$  to the cycle cost  $C(S_i, i, N)$  for  $i \in \{1, \dots, N\}$ . Therefore the complete expression for the cost of replenishment cycles that start in period  $i \in \{1, \dots, N\}$  and end in period  $N$  is:

$$C(S_i, i, N) = a + \sum_{k=i}^N (hz_{\alpha(i,k)}\sigma_{i,k} + (h + s)\sigma_{i,k}[\phi(z_{\alpha(i,k)}) - (1 - \alpha(i, k))z_{\alpha(i,k)}]) + v \left( S_i - \sum_{k=i}^N \tilde{d}_k \right). \tag{26}$$

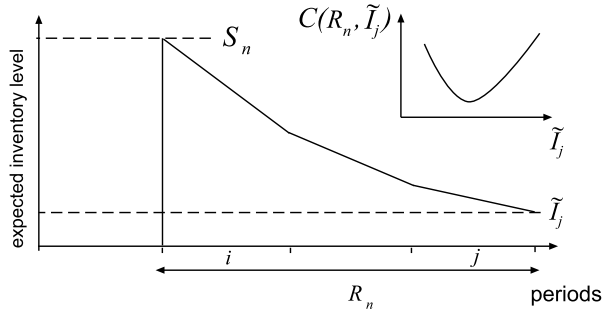
In the following section we will discuss how it is possible to incorporate in our CP model a dedicated cost-based filtering method (Focacci and Milano 1999) based on a *dynamic programming relaxation* (Tarim 1996) that is able to generate good bounds during the search. Such a technique has been already employed under a service level constraint (Tarim et al. 2009b). It should be noted that due to the non-linearity of the cost function induced by the shortage cost scheme, the version of the problem we consider is significantly more complicated than the one under a service level constraint. Nevertheless, despite the non-linearity of the cost function, we will see that the convexity of the cost function can be exploited to define a relaxation similar to the one employed in Tarim et al. (2009b).

### 5 Cost-based filtering by relaxation

Cost-based filtering is an elegant way of combining techniques from CP and Operations Research (OR) (Fahle and Sellmann 2002, Focacci and Milano 1999). OR-based optimization techniques are used, typically within optimization oriented global constraints, to remove values from variable domains that cannot lead to better solutions. This type of domain filtering can be combined with the usual CP-based filtering methods and branching heuristics, yielding powerful hybrid search algorithms.

In Tarim et al. (2009b), the authors adopt a relaxation proposed by Tarim (1996) for the CP model that computes  $(R_n, S_n)$  policy parameters under service level constraints. When the relaxed model is solved it provides good bounds for the original problem. Furthermore the relaxed problem is a Shortest Path Problem that can be solved in polynomial time. Therefore it is easy to obtain good bounds at each node of the search tree. In the same work the authors also explain how it is possible to take into account a partial assignment for replenishment decisions  $\delta_1, \dots, \delta_N$  and for expected closing inventory levels  $\tilde{I}_1, \dots, \tilde{I}_N$  when the relaxed

**Fig. 5** Convexity of the expected total cost associated to a given replenishment cycle  $R_n$  covering periods  $\{i, \dots, j\}$ . The expected total cost is a function of the expected closing inventory level  $\tilde{I}_j$



problem is constructed, so that the effect of these assignments is reflected on the bound that is obtained by solving the relaxed problem.

As shown in Tarim et al. (2009b), the CP model proposed for computing  $(R_n, S_n)$  policy parameters under service level constraints can be reduced to a Shortest Path Problem if the inventory conservation constraint and the replenishment condition constraint, that is constraints (23) and (24) in our model under shortage cost scheme, are relaxed for replenishment periods. That is for each possible pair of replenishment cycles  $\langle R(i, k - 1), R(k, j) \rangle$  where  $i, j, k \in \{1, \dots, N\}$  and  $i < k \leq j$ , the relationship between the opening inventory level of  $R(k, j)$  and the expect closing inventory level of  $R(i, k - 1)$ , i.e. (23) for  $t = k$ , is not considered. The same approach can be translated to the CP model for  $(R_n, S_n)$  under shortage cost scheme.

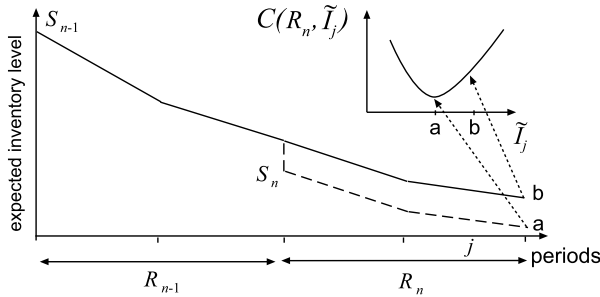
In Sect. 4, we provided a general function  $C(S_i, i, j)$  to compute the expected total cost of replenishment cycle  $R(i, j)$ , when the order-up-to-level is set to  $S_i$ . It should be noted that if  $\tilde{I}_j$  denotes the expected closing inventory level at period  $j$ , then the following linear relation exists between  $S_i$  and  $\tilde{I}_j$ ,  $S_i - \sum_{t=i}^j \tilde{d}_t = \tilde{I}_j$ . It follows that  $C(S_i, i, j)$  may be easily expressed in terms of  $\tilde{I}_j$  and that such a function is also convex in  $\tilde{I}_j$  (Fig. 5). Let us denote as  $C(i, j, \tilde{I}_j)$ ,  $C(S_i, i, j)$  expressed as a function of  $\tilde{I}_j$ . In explaining our cost-based filtering strategy, we perform this variable substitution because our CP model is expressed in terms of  $\tilde{I}_j$  and not of  $S_i$ .

If we consider each replenishment cycle  $R(i, j)$  independently, we can efficiently compute the optimal expected closing inventory level that minimizes the expected total cost associated to such a cycle using gradient based methods for convex optimization. This way we obtain a set  $S$  of  $N(N + 1)/2$  possible replenishment cycles and respective order-up-to-levels. Our new problem is to find an optimal set  $S^* \subset S$  of consecutive disjoint replenishment cycles that covers our planning horizon at the minimum cost.

In Tarim et al. (2009b) it was shown that the optimal solution to this relaxation is given by the shortest path in a graph from a given initial node to a final node where each arc represents a specific cost. We now adapt their approach to our model that employs a shortage cost scheme.

If  $N$  is the number of periods in the planning horizon of the original problem, we introduce  $N + 1$  nodes. By assuming, without loss of generality, that an order is always placed at period 1, we take node 1, which represents the beginning of the planning horizon, as the initial one. Node  $N + 1$  represents the end of the planning horizon. For each possible replenishment cycle  $R(i, j - 1)$  such that  $i, j \in \{1, \dots, N + 1\}$  and  $i < j$ , we introduce an arc  $(i, j)$  with associated cost

$$Q(i, j) = C(i, j - 1, \tilde{I}_{j-1}^*), \tag{27}$$



**Fig. 6** The optimal expected closing inventory level for replenishment cycle  $R_n$  considered alone is  $a$ , this minimizes the convex cost associated to replenishment cycle  $R_n$ . In order to meet the inventory conservation constraint for the stocks carried over from cycle  $R_{n-1}$ , the minimum expected closing inventory level required is  $b$ . Such a value produces a higher expected total cost for  $R_n$

where  $\tilde{I}_{j-1}^*$  is the expected closing inventory level that minimizes the convex cost of replenishment cycle  $R(i, j - 1)$ . Since we are dealing with a one-way temporal feasibility problem (Wagner and Whitin 1958), when  $i \geq j$ , we introduce no arc.

The cost of the shortest path from node 1 to node  $N + 1$  in the given graph is a valid lower bound for the original problem, as it is a solution of the relaxed problem. In fact, the expected total cost function for each replenishment cycle is convex in the expected closing inventory level held at the end of the cycle. Therefore in order to meet the violated inventory conservation constraints, if any exists, we will incur an overall higher expected total cost for a given group of replenishment cycles (Fig. 6).

Furthermore, it is easy to map the optimal solution for the relaxed problem, that is the set of arcs participating to the shortest path, to a solution for the original problem by noting that each arc  $(i, j)$  represents a replenishment cycle  $R(i, j - 1)$ . The feasibility of such a solution with respect to the original problem can be checked by verifying that it satisfies every relaxed constraint. If no inventory conservation constraint is violated, it is easy to see that the computed cost is optimal for the given replenishment plan.

We will now show how to exploit this lower bound in an optimization oriented global constraint able to dynamically produce good bounds when a partial solution is provided. Cost-based filtering can be performed by simply noting that the costs stored in the connection matrix can be adjusted to reflect the current partial assignment for decision variables  $\delta_i$  and  $\tilde{I}_i$  exactly the way shown for the service level constrained model (Tarim et al. 2009b). More specifically:

- $\delta_k = 0$ : if in a given partial solution a decision variable  $\delta_k, k \in \{1, \dots, N\}$  has been already set to 0, then we can remove from the graph every inbound arc to node  $k$  and every outbound arc from node  $k$ . This prevents node  $k$  from being part of the shortest path, and hence prevents period  $k$  from being a replenishment period. In this modified graph, the cost of the shortest path will provide a valid lower bound for the cost of an optimal solution incorporating the decision  $\delta_k = 0$ . Furthermore, an assignment for decision variables is associated with the shortest path. If this assignment is feasible for the original problem, then it is optimal with respect to the decision  $\delta_k = 0$ .
- $\delta_k = 1$ : on the other hand, if in a given partial solution a decision variable  $\delta_k, k \in \{1, \dots, N\}$  has been already set to 1, then we can remove from the graph every arc connecting a node  $i$  to a node  $j$ , where  $i < k < j$ . This forces the shortest path to pass through node  $k$ , and hence forces period  $k$  to be a replenishment period. In this modified graph,

the cost of the shortest path will provide a valid lower bound for the cost of an optimal solution incorporating the decision  $\delta_k = 1$ . Furthermore, an assignment for decision variables is associated with the shortest path. If this assignment is feasible for the original problem, then it is optimal with respect to the decision  $\delta_k = 1$ .

- $\tilde{I}_t$  assigned: if a given  $\tilde{I}_t$ ,  $t \in \{i, \dots, j - 1\}$  is assigned a value, the expected closing inventory level ( $\tilde{I}_{j-1}$ ) for the replenishment cycle  $R(i, j - 1)$ , which covers period  $t$ , is uniquely determined and therefore the expected total cost for such a replenishment cycle—that is the cost of arc  $(i, j)$ —can be directly computed from  $C(i, j, \tilde{I}_j)$ , provided that the current partial assignment for  $\delta_t$  decision variables uniquely identifies  $R(i, j - 1)$ .

Building upon these three cases it is easy to obtain a *gradient function* that returns, for each variable-value pair, an optimistic evaluation for the cost of an optimal solution.

## 6 Computational experience

In what follows, we present our computational experience. In Sect. 6.1, we analyze the computational efficiency of our approach for a large number of instances and realistic planning horizon lengths ranging from 20 to 38 periods. In Sect. 6.2, we compare the cost and the structure of the solutions obtained with our exact approach with those obtained with the heuristic approach proposed by Tarim and Kingsman (2006) for a set of 8-period instances discussed in their work.

### 6.1 Computational performances

We now discuss the computational efficiency of our approach. A single problem is considered and the period demands are generated from seasonal data with no trend:  $\tilde{d}_t = 50[1 + \sin(\pi t/6)]$ . In addition to the “no trend” case (P1) we also consider three others:

- (P2) positive trend case,  $\tilde{d}_t = 50[1 + \sin(\pi t/6)] + t$
- (P3) negative trend case,  $\tilde{d}_t = 50[1 + \sin(\pi t/6)] + (52 - t)$
- (P4) life-cycle trend case,  $\tilde{d}_t = 50[1 + \sin(\pi t/6)] + \min(t, 52 - t)$

In each test we assume an initial null inventory level and a normally distributed demand for every period with a coefficient of variation  $\sigma_t/\tilde{d}_t$  for each  $t \in \{1, \dots, N\}$ , where  $N$  is the length of the considered planning horizon. We performed tests using four different ordering cost values  $a \in \{50, 100, 150, 200\}$  and two different  $\sigma_t/\tilde{d}_t \in \{1/3, 1/6\}$ . The planning horizon length  $N$  takes even values in the range  $[20, 38]$ . The holding cost used in these tests is  $h = 1$  per unit per period. Our tests also consider two different shortage cost values  $s = 15$  and  $s = 25$ . Direct item cost is  $v = 2$  per unit produced.

All the experiments were performed on an Intel(R) Centrino(TM) CPU 1.50 GHz with 500 Mb RAM. The solver used is Choco (1), an open-source solver developed in Java. The cost-based filtering techniques presented are implemented as dedicated constraints within Choco.

The variable selection heuristic branches first on the  $\delta_t$  variables in lexicographic order. Note that when all the  $\delta_t$  variables have been assigned, the global constraint `objConstraint(·)` will immediately produce an assignment for the remaining decision variables. Values in decision variable domains are selected in lexicographic order.

Tables 1 and 2 show the time (in seconds) required by our CP model, enhanced with the cost-based filtering technique described in the former section, to obtain an optimal solution



**Table 1** Time (in seconds) required by our CP model, enhanced with the cost-based filtering technique described in Sect. 5, to obtain an optimal solution for test set P1 and P2. In our test results “–” means that, within the time limit of 5 seconds, our CP approach could not find an optimal solution

<i>a</i>	<i>N</i>	Test set P1				Test set P2			
		$\sigma_t/\tilde{d}_t = 1/3$		$\sigma_t/\tilde{d}_t = 1/6$		$\sigma_t/\tilde{d}_t = 1/3$		$\sigma_t/\tilde{d}_t = 1/6$	
		<i>s</i> = 15	<i>s</i> = 25	<i>s</i> = 15	<i>s</i> = 25	<i>s</i> = 15	<i>s</i> = 25	<i>s</i> = 15	<i>s</i> = 25
50	20	0.150	0.030	0.020	0.020	0.030	0.040	0.050	0.050
	22	–	–	0.020	0.030	0.040	0.030	0.060	0.060
	24	0.031	0.040	0.030	0.030	0.060	0.040	0.080	0.070
	26	0.040	0.070	0.040	0.040	0.060	0.050	0.120	0.120
	28	0.050	0.080	0.060	0.050	0.070	0.060	0.170	0.121
	30	0.080	0.090	0.060	0.050	0.080	0.081	0.161	0.161
	32	0.100	0.090	0.070	0.081	0.120	0.141	0.180	0.150
	34	–	–	0.060	0.070	0.140	0.080	0.180	0.160
	36	0.210	0.111	0.080	0.081	0.161	0.090	0.230	0.180
	38	0.171	0.100	0.090	0.080	0.140	0.120	0.210	0.241
100	20	0.030	–	0.020	0.030	0.040	0.030	0.020	0.020
	22	0.030	0.030	0.030	0.030	0.040	0.030	0.031	0.030
	24	0.030	0.040	0.040	0.030	0.040	0.041	0.040	0.030
	26	0.040	0.040	0.040	0.040	0.080	0.050	0.050	0.050
	28	0.060	0.070	0.050	0.050	0.060	0.071	0.060	0.051
	30	0.061	0.060	0.060	0.060	0.071	0.080	0.061	0.080
	32	0.080	–	0.070	0.070	0.081	0.090	0.071	0.070
	34	0.070	0.060	0.070	0.070	0.090	0.080	0.231	0.070
	36	0.080	0.101	0.071	0.071	0.101	0.100	0.090	0.090
	38	0.080	0.101	0.090	0.091	0.110	0.120	0.100	0.101
150	20	0.020	0.020	0.030	0.021	0.030	0.020	0.020	0.030
	22	0.030	0.030	0.030	0.020	0.030	0.030	0.030	0.030
	24	0.040	0.040	0.030	0.030	0.040	0.040	0.040	0.030
	26	0.040	0.040	0.040	0.040	0.040	0.050	0.050	0.061
	28	0.050	0.050	0.050	0.041	0.060	0.061	0.050	0.050
	30	0.070	0.071	0.050	0.061	0.070	0.070	0.060	0.070
	32	0.070	4.306	0.060	0.071	0.080	0.080	0.070	0.070
	34	0.070	0.070	0.060	0.070	0.100	0.080	0.070	0.071
	36	0.080	0.080	0.070	0.080	0.090	0.110	0.080	0.090
	38	0.090	0.100	0.100	0.080	0.110	0.120	0.110	0.121
200	20	0.030	0.030	0.030	0.020	0.031	0.040	0.030	0.020
	22	0.030	0.220	0.030	0.030	0.030	0.041	0.030	0.030
	24	0.030	0.040	0.030	0.040	0.040	0.040	0.030	0.041
	26	0.040	0.040	0.040	0.040	0.050	0.051	0.041	0.050
	28	0.050	0.050	0.051	0.060	0.080	0.060	0.060	0.050
	30	0.070	0.060	0.060	0.060	0.070	0.070	0.070	0.070
	32	0.080	0.080	0.060	0.060	0.080	0.090	0.070	0.070
	34	0.070	–	0.070	0.070	0.090	0.080	0.080	0.081
	36	0.080	0.081	0.070	0.070	0.110	0.101	0.090	0.110
	38	0.100	0.090	0.091	0.090	0.121	0.100	0.110	0.110

**Table 2** Time (in seconds) required by our CP model, enhanced with the cost-based filtering technique described in Sect. 5, to obtain an optimal solution for test set P3 and P4. In our test results “–” means that, within the time limit of 5 seconds, our CP approach could not find an optimal solution

<i>a</i>	<i>N</i>	Test set P3				Test set P4			
		$\sigma_t/\tilde{d}_t = 1/3$		$\sigma_t/\tilde{d}_t = 1/6$		$\sigma_t/\tilde{d}_t = 1/3$		$\sigma_t/\tilde{d}_t = 1/6$	
		<i>s</i> = 15	<i>s</i> = 25	<i>s</i> = 15	<i>s</i> = 25	<i>s</i> = 15	<i>s</i> = 25	<i>s</i> = 15	<i>s</i> = 25
50	20	0.321	0.170	0.330	0.160	0.070	0.030	0.050	0.061
	22	0.480	0.300	0.370	0.341	0.030	0.040	0.060	0.060
	24	0.581	0.310	0.531	0.421	0.050	0.040	0.110	0.071
	26	1.222	0.501	0.791	0.531	0.070	0.060	0.090	0.090
	28	2.224	0.661	1.142	0.741	0.140	0.070	0.120	0.160
	30	2.013	0.722	1.052	0.751	0.100	0.060	0.130	0.170
	32	1.812	0.941	1.182	0.801	0.121	0.080	0.180	0.140
	34	1.883	0.862	1.312	0.952	0.120	0.090	0.190	0.150
	36	2.093	0.981	1.472	1.152	0.121	0.110	0.210	0.180
	38	3.636	1.131	1.803	1.512	0.120	0.100	0.251	0.200
100	20	0.030	0.040	0.060	0.070	0.040	0.030	0.030	0.020
	22	0.040	0.040	0.070	0.071	0.040	0.030	0.030	0.030
	24	0.040	0.050	0.090	0.080	0.050	0.030	0.040	0.040
	26	0.050	0.281	0.100	0.100	0.050	0.050	0.050	0.040
	28	0.070	0.070	0.131	0.120	0.061	0.060	0.060	0.060
	30	0.070	0.070	0.140	0.130	0.070	0.070	0.070	0.060
	32	0.080	0.080	0.150	0.160	0.080	0.080	0.070	0.070
	34	0.090	0.090	0.161	0.210	0.090	0.081	0.080	0.070
	36	0.100	0.110	0.240	0.180	0.090	0.090	0.090	0.080
	38	0.141	0.130	0.211	0.250	0.100	0.100	0.110	0.100
150	20	0.040	0.030	0.060	0.060	0.030	0.030	0.030	0.030
	22	0.040	0.040	0.071	0.070	0.030	0.030	0.030	0.030
	24	0.050	0.041	0.140	0.080	0.040	0.030	0.040	0.030
	26	0.060	0.050	0.100	0.160	0.060	0.040	0.050	0.040
	28	0.070	0.070	0.120	0.170	0.060	0.060	0.060	0.050
	30	0.070	0.070	0.130	0.140	0.070	0.060	0.070	0.060
	32	0.090	0.090	0.160	0.220	0.080	0.080	0.070	0.070
	34	0.091	0.090	0.171	0.170	0.080	0.090	0.080	0.080
	36	0.100	0.110	0.181	0.250	0.100	0.100	0.090	0.090
	38	0.140	0.120	0.220	0.260	0.100	0.101	0.100	0.100
200	20	0.071	0.030	0.060	0.070	0.040	0.030	0.030	0.030
	22	0.090	0.070	0.070	0.120	0.030	0.030	0.030	0.030
	24	0.090	0.130	0.080	0.080	0.050	0.040	0.030	0.040
	26	0.110	0.110	0.100	0.171	0.050	0.050	0.051	0.050
	28	0.130	0.170	0.181	0.130	0.070	0.070	0.060	0.050
	30	0.210	0.150	0.150	0.151	0.060	0.070	0.070	0.060
	32	0.210	0.090	0.150	0.221	0.070	0.070	0.070	0.080
	34	0.210	0.241	0.180	0.160	0.080	0.080	0.080	0.081
	36	0.250	0.210	0.241	0.190	0.090	0.100	0.080	0.090
	38	0.221	0.271	0.260	0.210	0.140	0.110	0.100	0.131

**Table 3** Expected demand values from Tarim and Kingsman (2006)

Period	1	2	3	4	5	6	7	8
$\tilde{d}_t$	200	100	70	200	300	120	50	100

for test set P1, P2, P3 and P4. In our test results “–” means that, within the time limit of 5 seconds, our CP approach could not find an optimal solution.

When the cost-based filtering method we proposed in Sect. 5 is not used, the pure CP approach is never able to provide an optimal solution within the given running time limit of 5 seconds for every instance. The CP approach employing cost-based filtering generally requires only a fraction of a second to produce the optimal solution. Only in 7 instances the optimal solution was not produced within the given time limit of 5 seconds. The worst case running time for our CP approach employing cost-based filtering over the whole test bed was 6.77 minutes. Therefore even in the few cases in which an optimal solution is not found in less than 5 seconds, our CP model provides a reasonable running time.

## 6.2 Comparison between exact CP and approximate MIP approaches

In this section, we compare the cost and the structure of the solutions obtained with our exact approach with those obtained with the heuristic approach proposed by Tarim and Kingsman (2006) for a set of 8-period instances discussed in their work.

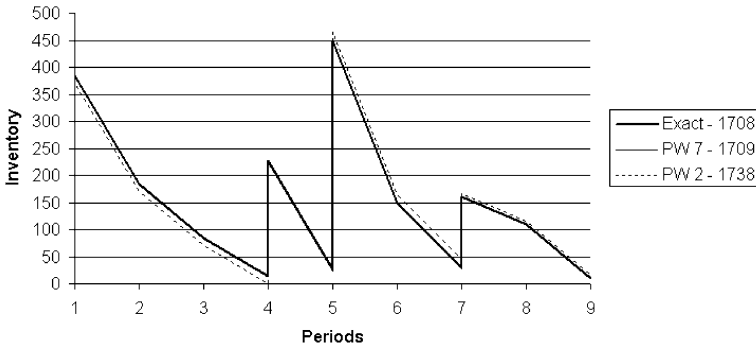
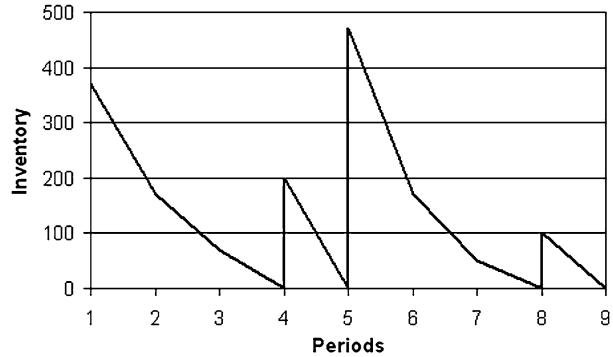
Tarim and Kingsman (2006) proposed a piecewise linear approximation of the cost function for the single-period newsvendor type model under holding and shortage costs, which we analyzed in this work. They were able to build a MIP model approximating an optimal solution for the multi-period stochastic lot-sizing under fixed ordering, holding and shortage costs. They gave a few examples to show the effect of higher noise levels (uncertainty in the demand forecasts) on the order schedule. By using the same examples we shall compare the policies obtained using our exact CP approach with their approximation.

Depending on the number of segments used in Tarim and Kingsman’s piecewise approximation, the quality of the solutions obtained can be improved. We shall consider approximations with two and seven segments.

Tarim and Kingsman considered a set of instances defined on a planning horizon comprising  $N = 8$  periods. The expected demand  $\tilde{d}_t$  they considered for each period  $t = 1, \dots, N$  is given in Table 3. The demand in each period is normally distributed about the forecast value with the same coefficient of variation  $\tau$ . Thus the standard deviation of demand in period  $t$  is  $\sigma_t = \tau \cdot \tilde{d}_t$ . In all cases, initial inventory levels, delivery lead-times and salvage values are set to zero.

In Fig. 7–11 optimal replenishment policies obtained with our CP approach are compared for four different instances, with respect to  $\tau$ ,  $v$ ,  $a$  and  $s$ , with the policies provided by the 2-segment (PW-2) and 7-segment (PW-7) approximations. For each instance we compare the expected total cost provided by the exact method with the expected total cost provided by the policies found using approximate MIP models. Since the cost provided by PW-2 and PW-7 is an approximation, it often differs significantly from the real expected total cost related to policy parameters found by these models. It is therefore not meaningful to compare the cost provided by the MIP model with that of the optimal policy obtained with our CP model. To obtain a meaningful comparison we computed the real expected total cost by applying the exact cost function ((20), (26)) discussed above to the  $(R_n, S_n)$  policy parameters obtained

**Fig. 7** The optimal replenishment policy for  $h = 1$ ,  $a = 250$ ,  $s = 10$ ,  $v = 0$ ,  $\tau = 0.0$



**Fig. 8** The optimal replenishment policy for  $h = 1$ ,  $a = 250$ ,  $s = 10$ ,  $v = 0$ ,  $\tau = 0.1$

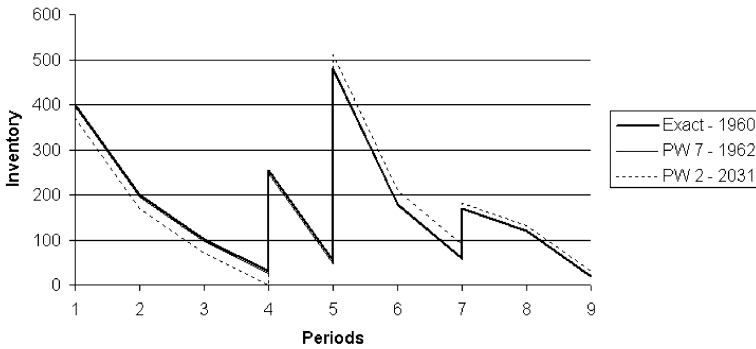
through PW-2 and PW-7. It is then possible to assess the accuracy of approximations in Tarim and Kingsman (2006).

Figure 7 shows the optimal replenishment policy for the deterministic case ( $\tau = 0.0$ ). The direct item cost ( $v$ ) is taken as zero. Four replenishment cycles are planned. The  $(R_n, S_n)$  policy parameters are  $R = [3, 1, 3, 1]$  and  $S = [370, 200, 470, 100]$ . The total cost for this policy is 1460.

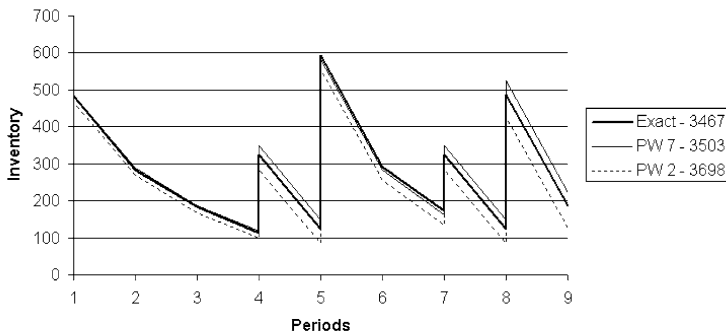
Figure 8 shows an instance where we consider low levels of forecast uncertainty ( $\tau = 0.1$ ). In this case both PW-2 and PW-7 perform well compared to our exact CP solutions. Since forecast uncertainty must be considered, all the models introduce buffer stocks. The optimal  $(R_n, S_n)$  policy parameters found by our CP approach are  $R = [3, 1, 2, 2]$  and  $S = [384, 227, 449, 160]$ . The PW-2 solution is 1.75% more costly than the exact solution, while the PW-7 solution is slightly more costly than the exact solution.

Figure 9 shows that as the level of forecast uncertainty increases ( $\tau = 0.2$ ), the quality of the PW-2 solution deteriorates, in fact it is now 3.62% more costly than the exact solution. The optimal  $(R_n, S_n)$  policy parameters found by our CP approach are  $R = [3, 1, 2, 2]$  and  $S = [401, 253, 479, 170]$ . In contrast the PW-7 solution is still only slightly more costly than the exact solution.

As noted in Tarim and Kingsman (2006) the quality of the approximation decreases for high ratios  $s/h$ . In Fig. 10 we consider  $s/h = 50$  and a different demand pattern, in which the expected demand is particularly high in the last two periods. This modified demand pattern is suitable for showing the impact of the direct item cost  $v$  on the structure of the



**Fig. 9** The optimal replenishment policy for  $h = 1, a = 250, s = 10, v = 0, \tau = 0.2$



**Fig. 10** The optimal replenishment policy for  $h = 1, a = 350, s = 50, v = 0, \tau = 0.3$

**Table 4** Expected demand values for our modified demand pattern

Period	1	2	3	4	5	6	7	8
$\tilde{d}_t$	200	100	70	200	300	120	200	300

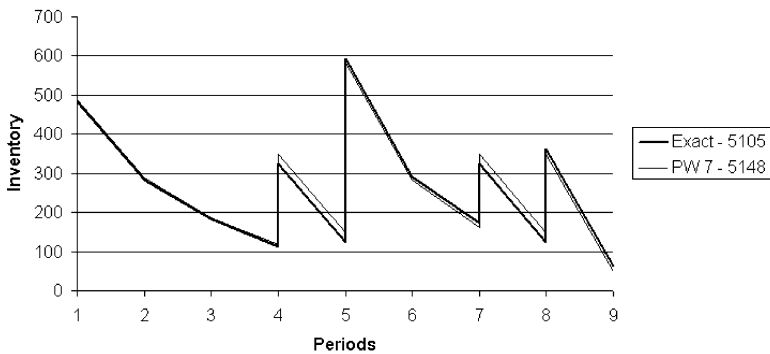
optimal policy. The forecast of demand in each period are given in Table 4. Now the PW-2 solution is 6.66% more costly than the exact approach, while the PW-7 solution is 1.03% more costly. The optimal  $(R_n, S_n)$  policy parameters found by our CP approach are  $R = [3, 1, 2, 1, 1]$  and  $S = [483, 324, 592, 324, 486]$ .

In Fig. 11 we consider the same instance but a direct item cost is now incurred ( $v = 15$ ). The buffer stock held in the last replenishment cycle is affected by this parameter, and is decreased from 186 to 63. The PW-7 policy is now 0.84% more costly than the exact one.

We can conclude that, for these instances, seven segments for the piecewise linear approximation usually provide a solution with a cost reasonably close to optimal.

### 7 Conclusions

We presented a CP approach that finds optimal replenishment cycle policy parameters under non-stationary stochastic demand and a shortage cost scheme. We have exploited the



**Fig. 11** The optimal replenishment policy for  $h = 1$ ,  $a = 350$ ,  $s = 50$ ,  $v = 15$ ,  $\tau = 0.3$

convexity of the cost function in concert with an existing relaxation strategy to dynamically generate bounds and perform cost-based filtering during search. Our computational experience shows the effectiveness of our approach. We are now able to solve problems over a planning horizon up to 38 periods, typically in a fraction of a second and in the worst case in a few minutes. This means that our approach can now be applied to problems of a realistic size. By using our approach it is now possible to evaluate the quality of a previously published MIP-based approximation method. For a set of problem instances we showed that a piecewise approximation with seven segments usually provides good quality solutions, while using only two segments can yield solutions that differ significantly from the optimal.

**Acknowledgements** R. Rossi has received funding from the European Community's Seventh Framework Programme (FP7) under grant agreement no. 244994 (project VEG-i-TRADE). S.A. Tarim and Brahim Hnich are supported by the Scientific and Technological Research Council of Turkey (TUBITAK). S.A. Tarim is supported by Hacettepe University-BAB.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

## References

- Apt, K. (2003). *Principles of constraint programming*. Cambridge: Cambridge University Press.
- Bayraktar, E., & Ludkovski, M. (2010). Inventory management with partially observed nonstationary demand. *Annals of Operation Research*, *176*(1), 7–39.
- Birge, J. R., & Louveaux, F. (1997). *Introduction to stochastic programming*. New York: Springer.
- Bookbinder, J. H., & Tan, J. Y. (1988). Strategies for the probabilistic lot-sizing problem with service-level constraints. *Management Science*, *34*, 1096–1108.
- de Kok, A. G. (1991). *Basics of inventory management. Part 2. The (R, S)-model*. Research memorandum, FEW 521. Department of Economics, Tilburg University, Tilburg, The Netherlands.
- de Kok, T., & Inderfurth, K. (1997). Nervousness in inventory management: comparison of basic control rules. *European Journal of Operational Research*, *103*, 55–82.
- Fahle, T., & Sellmann, M. (2002). Cost-based filtering for the constrained knapsack problem. *Annals of Operation Research*, *115*, 73–93.
- Focacci, F., Lodi, A., & Milano, M. (1999). Cost-based domain filtering. In *Lecture notes in computer science: Vol. 1713. Proceedings of the 5th international conference on the principles and practice of constraint programming* (pp. 189–203). Berlin: Springer.
- Focacci, F., Lodi, A., & Milano, M. (2002). Optimization-oriented global constraints. *Constraints*, *7*(3–4), 351–365.

- Focacci, F., & Milano, M. (2001). Connections and integrations of dynamic programming and constraint programming. In *Proceedings of the international workshop on integration of AI and OR techniques in constraint programming for combinatorial optimization problems CP-AI-OR 2001*.
- Fortuin, L. (1980). Five popular probability density functions: a comparison in the field of stock-control models. *The Journal of the Operational Research Society*, 31(10), 937–942.
- Graves, S. C. (1999). A single-item inventory model for a non-stationary demand process. *Manufacturing & Service Operations Management*, 1, 50–61.
- Hadley, G., & Whitin, T. M. (1964). *Analysis of Inventory Systems*. New York: Prentice Hall.
- Heisig, G. (2002). *Planning stability in material requirements planning systems*. New York: Springer.
- Hnich, B., Rossi, R., Tarim, S. A., & Prestwich, S. D. (2009). Synthesizing filtering algorithms for global chance-constraints. In *Lecture notes in computer science: Vol. 5732. Proceedings of principles and practice of constraint programming (CP 2009)* (pp. 439–453). Berlin: Springer.
- Hnich, B., Rossi, R., Tarim, S. A., & Prestwich, S. (2011). A survey on CP-AI-OR hybrids for decision making under uncertainty. In P. van Hentenryck & M. Milano (Eds.), *Springer optimization and its applications: Vol. 45. Hybrid optimization* (pp. 227–270). New York: Springer. Chap. 7.
- F. Laburthe and the OCRE Project Team (1994). *Choco: Implementing a CP kernel*. Technical report, Bouygues e-Lab, France.
- Levi, R., Roundy, R. O., & Shmoys, D. B. (2006). Provably near-optimal sampling-based algorithms for stochastic inventory control models. In *Proceedings of the thirty-eighth annual ACM symposium on theory of computing (STOC '06)* (pp. 739–748). New York: ACM Press.
- Pujawan, I. N., & Silver, E. A. (2008). Augmenting the lot sizing order quantity when demand is probabilistic. *European Journal of Operational Research*, 127(3), 705–722.
- Regin, J.-C. (1994). A filtering algorithm for constraints of difference in csps. In *American Association for Artificial Intelligence* (pp. 362–367). Seattle, Washington.
- Regin, J.-C. (2003). Global constraints and filtering algorithms. In M. Milano (Ed.), *Constraints and integer programming combined*. Dordrecht: Kluwer Academic.
- Rossi, F., van Beek, P., & Walsh, T. (2006). *Handbook of constraint programming. Foundations of artificial intelligence*. New York: Elsevier.
- Rossi, R., Tarim, S. A., Hnich, B., & Prestwich, S. (2007). Replenishment planning for stochastic inventory systems with shortage cost. In *Lecture notes in computer science: Vol. 4510. Proceedings of the international conference on integration of AI and OR techniques in constraint programming for combinatorial optimization problems CP-AI-OR 2007* (pp. 229–243). Berlin: Springer.
- Rossi, R., Tarim, S. A., Hnich, B., & Prestwich, S. D. (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. (2011). A state space augmentation algorithm for the replenishment cycle inventory policy. *International Journal of Production Economics*, 113(1), 377–384.
- Silver, E. A., Pyke, D. F., & Peterson, R. (1998). *Inventory management and production planning and scheduling*. New York: Wiley.
- Tang, C. S. (2006). Perspectives in supply chain risk management. *International Journal of Production Economics*, 103, 451–488.
- Tarim, S. A. (1996). *Dynamic lotsizing models for stochastic demand in single and multi-echelon inventory systems*. Ph.D. thesis, Lancaster University.
- 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, 105–119.
- Tarim, S. A., & Kingsman, B. G. (2006). Modelling and computing ( $R^n, S^n$ ) policies for inventory systems with non-stationary stochastic demand. *European Journal of Operational Research*, 174, 581–599.
- Tarim, S. A., & Smith, B. (2008). Constraint programming for computing non-stationary ( $R, S$ ) inventory policies. *European Journal of Operational Research*, 189, 1004–1021.
- Tarim, S. A., Hnich, B., Prestwich, S. D., & Rossi, R. (2009a). Finding reliable solution: event-driven probabilistic constraint programming. *Annals of Operation Research*, 171(1), 77–99.
- Tarim, S. A., Hnich, B., Rossi, R., & Prestwich, S. D. (2009b). Cost-based filtering techniques for stochastic inventory control under service level constraints. *Constraints*, 14(2), 137–176.
- Tempelmeier, H. (2007). On the stochastic uncapacitated dynamic single-item lotsizing problem with service level constraints. *European Journal of Operational Research*, 127(1), 184–194.
- Wagner, H. M., & Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management Science*, 5, 89–96.
- Walsh, T. (2002). Stochastic constraint programming. In *Proceedings of European conference on artificial intelligence (ECAI'2002)* (pp. 111–115).