A Global Constraint for Computing Exact Buffer Stock Levels in Stochastic Inventory Control

Student: Roberto Rossi
Supervisor: Steven Prestwich

Cork Constraint Computation Centre, University College, Cork, Ireland
{r.rossi,s.prestwich}@4c.ucc.ie

Abstract. Inventory theory provides methods for managing and controlling inventories under different constraints and environments. We consider a class of production/inventory control problems that has a single product and a single stocking location, when a stochastic demand with a known non-stationary probability distribution is given. A control policy for this type of inventory system is the one where the objective is to find the optimal number of replenishments, their timings and their respective order-up-to-levels that meet customer demands to a required service level. Two different models have been presented so far to solve this problem to optimality: a MIP model and an efficient CP model. In both these models 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 not returned to the supply source. Since this event is assumed to be rare, in both the models its effect is ignored. We present a global constraint that lets us to compute exact buffer stock levels for the CP model by considering the effect that carrying excess stock has on the service level in each period of our planning horizon. An extended version of this paper hasn't been submitted to the technical programme.

1 Introduction

We consider the class of production/inventory control problems that refers to the single-location, single-product case under non-stationary stochastic demand. In this problem the following inputs are given: a planning horizon of N periods; and a demand d_t for each period $t \in \{1, \ldots, N\}$, which is a random variable with probability density function $g_t(d_t)$. We will assume without loss of generality that these variables are normally distributed and that the demand occurs instantaneously at the beginning of each time period. The demand we consider is non-stationary, that is it can vary from period to period, and demands in different periods are assumed to be independent. A fixed delivery cost a is considered for each order and also a linear holding cost h is considered for each unit of product carried in stock from one period to the next. We assume that it is not possible to sell back excess items to the vendor at the end of a period. As a service level constraint we require the probability that at the end of each and every period the net inventory will not be negative set to be at least a given value α. Our aim is to minimize the expected total cost (ordering costs and holding costs) over the N-period planning horizon, satisfying the service level constraints.

Different inventory control policies [4] can be adopted to cope with the described problem. A policy states the rules to decide when orders have to be placed and how to compute the replenishment lot-size for each order. One of the possible policies that can be adopted is the replenishment cycle policy (R, S). Under the non-stationary demand assumption this policy takes the non-stationary form (R^n, S^n), where R^n denotes the length of the nth replenishment cycle, and S^n the order-up-to-level values for each replenishment. In order to provide a solution for our problem under the (R^n, S^n) policy we must populate both the sets R^n and S^n.

The first complete solution method for this problem was introduced by Tarim & Kingsman [2], who proposed a certainty-equivalent Mixed Integer Programming (MIP) formulation for computing (R^n, S^n) policy parameters. Tarim & Smith [1] introduced a more compact and efficient Constraint Programming (CP) formulation of the same problem that showed a significant computational improvement over the MIP formulation.

Both the MIP and the CP formulation assume 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 not returned to the supply source, but since this event is assumed to be rare, in both the models its effects are ignored:

- The cost of carrying excess stock is ignored, therefore the actual cost of a policy can be higher than the one provided by the model.
- The event of carrying excess stock can have a significant impact on the service level of the next periods, in particular it could be possible to exploit excess stock to provide the required service level keeping lower buffer stocks.

This paper extends the CP model presented in [1] by dropping its service level constraint, which is based on a matrix computed a-priori, and by expressing such a constraint as a global constraint able to dynamically compute exact buffer stock levels during the search. It has to be noticed that while in CP the former assumption can be relaxed using a dedicated global constraint, this is not possible in MIP, where buffer stocks have to be pre-computed. CP is therefore not only a more efficient way than MIP for dealing with stochastic inventory control as shown in [1], but it is also a mandatory choice if we want to compute the optimal solution for the (R^n, S^n) policy. The paper is organized as follows. In Section 2 we describe the CP model for the (R^n, S^n) policy. In Section 3 our global constraint that computes exact buffer stock levels is introduced. In Section 4 we give an example that shows the effectiveness of our approach.

2 A CP model

In this section we review the CP formulation for the (R^n, S^n) policy proposed in [1]. For a detailed discussion of Constraint Programming see [5]. The CP formulation presented in [1] for the (R^n, S^n) policy is as follows:

$$\min \ E\{TC\} = \sum_{t=1}^{N} \left(a\delta_t + h\tilde{I}_t \right) \tag{1}$$

subject to

$$\tilde{I}_t + \tilde{d}_t - \tilde{I}_{t-1} \geq 0 \tag{2}$$

$$\tilde{I}_t + \tilde{d}_t - \tilde{I}_{t-1} > 0 \Rightarrow \delta_t = 1 \tag{3}$$

$$\tilde{I}_t \geq \Phi \left[t, \max_{j \in \{1,\ldots,t\}} j\delta_j \right] \tag{4}$$

$$\tilde{I}_t \in \mathbf{Z}^+ \cup \{0\}, \quad \delta_t \in \{0,1\} \tag{5}$$

for $t = 1 \ldots N$ where $\Phi[i,j]$ is defined by

$$\Phi[i,j] = G^{-1}_{d_j+d_{j+1}+\ldots+d_i}(\alpha) - \sum_{k=j}^{i} \tilde{d}_k$$

and is implemented using the `element` constraint, and $G_{d_j+d_{j+1}+\ldots+d_i}$ is the cumulative probability distribution function of $d_j + d_{j+1} + \ldots + d_i$. It is assumed that G is strictly increasing, hence G^{-1} is uniquely defined.

Each decision variable $\tilde{I}_t$ represents the expected inventory level at the end of period t. The binary decision variables δ_t state whether a replenishment is fixed for period t ($\delta_t = 1$) or not ($\delta_t = 0$). The objective function (1) minimizes the total expected cost over the given planning horizon. Two terms contribute to the overall expected cost: ordering costs and inventory holding costs. Constraint (2) enforces a no-buy-back condition, which means that received goods cannot be returned to the supplier. As a consequence of this the expected net inventory at period t must be no less than the expected net inventory in period $t+1$ plus the expected demand in period t. Constraint (3) expresses the replenishment condition. We have a replenishment if the expected net inventory at period t is greater than the expected net inventory in period $t+1$ plus the expected demand in period t. This means that we received some extra goods as a consequence of an order. Constraint (4) enforces the required service level α. This is done by specifying the minimum buffer stock required for each period t in order to assure that, at the end of each and every time period, the probability the net inventory will not be negative is at least α. Buffer stock levels in matrix Φ are computed a-priori for each replenishment cycle, therefore they don't take into account the effect that excess stock from former periods has (Fig. 1).

It is not possible to compute a-priori such an effect, because for each buffer stock related to a given replenishment cycle, this effect directly depends on the length of former replenishment cycles. In order to take this effect into account we can implement a global constraint that dynamically computes buffer stock levels depending on the current partial assignment of the δ_i variables.

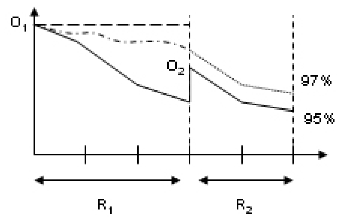

Fig. 1. Effect of excess stock: the buffer stock computed a-priori assures a 95% service level, but the combined effect of excess stock in the given policy produce a higher actual service level

3 A new service level global constraint

In this section, in order to describe our global constraint, we will exploit the following property of buffer stocks.

Property 1. The buffer stock for any replenishment cycle depends only on the length of former replenishment cycles and not on subsequent cycle lengths.

We will consider now a two replenishment cycle case (Fig. 1) in an N period planning horizon, then we will give an idea about how it is possible to extend it in a recursive fashion to the case of M subsequent replenishment cycles.

The planning horizon is made up of two consecutive replenishment cycles, let us call them R_1 and R_2. Let O_i be the opening inventory level for R_i. We assume that O_1 is known (Property 1). We define $P(H)$ as the probability of the event "observing a demand higher than $O_1 - O_2$ during R_1". $P(D_x)$ is the probability of the event "observing a demand equal to x, where $x \in \{0, ..., O_1 - O_2\}$, during R_1". S_y is the service level at the end of R_2 if a buffer stock y is hold. Then the correct buffer stock for R_2 can be computed as the minimum value b s.t.

$$P(H) \cdot S_b + \frac{\sum_{i=0}^{O_1-O_2} (P(D_i) - P(D_{i-1})) \cdot S_{b+O_1-O_2-i}}{\sum_{i=0}^{O_1-O_2} (P(D_i) - P(D_{i-1}))} \cdot (1 - P(H)) \geq \alpha \quad (6)$$

where $O_2 = \tilde{d}_2 + b$. For the two replenishment cycles case, this can be rewritten using the following extended form

$$(1 - G^{-1}(\frac{Q}{\sigma_1})) \cdot G^{-1}(\frac{b}{\sigma_2}) + \frac{\sum_{i=-d_1}^{Q} (G^{-1}(\frac{i}{\sigma_1}) - G^{-1}(\frac{i-1}{\sigma_1})) \cdot G^{-1}(\frac{b-(i-Q)}{\sigma_2})}{\sum_{i=-d_1}^{Q} (G^{-1}(\frac{i}{\sigma_1}) - G^{-1}(\frac{i-1}{\sigma_1}))} \cdot G^{-1}(\frac{Q}{\sigma_1}) \geq \alpha \quad (7)$$

where $Q = O_1 - O_2 - \tilde{d}_1$, G^{-1} is the inverse normal cumulative distribution function, σ_i is the standard deviation of the demand for the replenishment cycle i. Notice that if the opening inventory level of R_1 is smaller than the opening inventory level of R_2, obviously the former cycle has no influence on the buffer stock and Condition 6 becomes $S_b \geq \alpha$. Furthermore, if the computed b is s.t. $R_2 \leq R_1 - \tilde{d}_1$, we just set the buffer stock to the minimum value allowed, that is $R_1 - \tilde{d}_1 - \tilde{d}_2$. Finally we should observe that, since we are using the standard normal distribution function and not a truncated normal distribution, the right term in the sum has to be normalized to $\frac{(1-P(H))}{\sum_{i=0}^{O_1-O_2}(P(D_{x+1})-P(D_x))}$.

Let us see how to apply this computation during the search. At each node of the search tree if at least a decision variable δ_i, $i \in \{1, ..., N\}$ that has not been assigned yet exists, we don't enforce any service level constraint. Otherwise if $\exists \delta_i$ s.t. $\delta_i = 1$, we know that a replenishment cycle starts in period i and it covers subsequent periods till the minimum j, $j \geq i$ s.t. $\delta_{j+1} = 1$ or $j + 1 > N$. We now exploit Property 1, which assures that we can consider subsequent replenishment cycles in our planning horizon in a sequential fashion. Therefore we can generate the buffer stock for the first replenishment cycle, which is not affected by any other replenishment period. Then we can generate the buffer stock for the second, which is only affected by the first one, whose buffer stock is known, etc. Each time we compute the buffer stock level b for a replenishment cycle we can remove from the domain of I_t, where t is the last period in the replenishment cycle, every value smaller than b.

We now describe briefly how it is possible to extend this computation to the case of M replenishment cycles. The key idea is that the buffer stock of a replenishment cycle R_j will be affected only by former replenishment cycles, until the first R_{i-1}, $i-1 < j$, whose opening inventory level is smaller than the one of R_i. Let O_i be the opening inventory level of R_i. Now $P(H)$ is the probability of the event "observing a demand higher than $O_i - O_j$ during $\{R_i, ..., R_j\}$", while $P(D_x)$ is the probability of the event "observing a demand equal to x, where $x \in \{0, ..., O_i - O_j\}$, during $\{R_i, ..., R_j\}$". It is now easy to extend Condition 6 to compute the buffer stock b for R_j.

$$P(H) \cdot S_b + \frac{\sum_{i=0}^{O_i-O_j}(P(D_i) - P(D_{i-1})) \cdot S_{b+O_i-O_j-i}}{\sum_{i=0}^{O_i-O_j}(P(D_i) - P(D_{i-1}))} \cdot (1 - P(H)) \geq \alpha \tag{8}$$

4 An example

We now present an example to compare the solution provided when our global constraint is used and the one provided by the original CP model. We assume an initial null inventory level and a normally distributed demand with a coefficient of variation $\sigma_t/\tilde{d}_t$ for each period $t \in \{1, \ldots, 4\}$. The expected values for the demand in each period are $\{120, 70, 50, 40\}$. The other parameters are $a = 150$, $h = 1$, $\sigma_t/\tilde{d}_t = 0.4$, $\alpha = 0.8(z_{\alpha=0.8} = 0.8414)$. In Table 1 the optimal solution found when our global constraint is used to dynamically generate buffer stock levels is compared with the one obtained using the matrix Φ. It is possible to see that by computing correct buffer stock levels we obtained a less costly policy, still meeting the required service level of 80%. In fact were able to keep lower stocks in the last two periods exploiting the effect of excess stocks carried on from former periods.

Original buffer stock computation			Dynamic buffer stock computation		
Policy cost: 548			Policy cost: 542		
I_1 117	δ_1 1	Service 99.2	I_1 117	δ_1 1	Service 99.2
I_2 47	δ_2 0	Service 80.4	I_2 47	δ_2 0	Service 80.4
I_3 62	δ_3 1	Service 100	I_3 59	δ_3 1	Service 99.8
I_4 22	δ_4 0	Service 82.8	I_4 19	δ_4 0	Service 80.1

Table 1. Optimal solution comparison

5 Conclusions

In [1] Tarim and Smith showed that CP is a more natural and efficient way, compared to mathematical programming, for expressing constraints for lot-sizing under the (R^n, S^n) policy. In this paper we showed that in CP it is also possible to dynamically consider during the search the effect of excess stocks, which are carried from former replenishment cycle, on the buffer stock level of a given period. When the service level constraint is expressed using the global constraint we presented, we showed that the CP model can provide a better solution than the one produced by the MIP model or by the original CP model presented in [1]. Unfortunately the current implementation is not as efficient as the original CP model. As a future extension we plan to incorporate cost based filtering methods in the model in order to improve the search process.

Acknowledgements This work was supported by Science Foundation Ireland under Grant No. 03/CE3/I405 as part of the Centre for Telecommunications Value-Chain-Driven Research (CTVR) and Grant No. 00/PI.1/C075.

References

1. S. A. Tarim, B. Smith. Constraint Programming for Computing Dynamic (R,S) Inventory Policy With Non-Stationary Stochastic Demand Under Service Level Constraints. Cork Constraint Computation Center, UCC, Ireland, 2005. (Submitted for publication.)
2. S. A. Tarim, B. G. Kingsman. The Stochastic Dynamic Production/Inventory Lot-Sizing Problem With Service-Level Constraints. *International Journal of Production Economics* 88:105–119, 2004.
3. H. M. Wagner, T. M. Whitin. Dynamic Version of the Economic Lot Size Model. *Management Science* 5:89–96, 1958.
4. R. Peterson, E. Silver, D. F. Pyke. Inventory Management and Production Planning and Scheduling. John Wiley and Sons, New York, 1998.
5. K. Apt. Principles of Constraint Programming. Cambridge University Press, Cambridge, UK, 2003.
6. L. Fortuin. Five Popular Probability Density Functions: a Comparison in the Field of Stock-Control Models. *Journal of the Operational Research Society* 31(10):937–942, 1980.
7. I. J. Lustig, J.-F. Puget. Program Does Not Equal Program: Constraint Programming and its Relationship to Mathematical Programming. *Interfaces* 31:29–53, 2001.