# A Decision Support System for Computing Optimal $(R, S)$ Policy Parameters*

S. Armagan Tarim,[1] Brahim Hnich,[2] Roberto Rossi,[3] and Steven Prestwich[3]

Department of Management, Hacettepe University, Ankara, Turkey[1]
armagan.tarim@hacettepe.edu.tr
Faculty of Computer Science, Izmir University of Economics, Turkey[2]
brahim.hnich@ieu.edu.tr
Cork Constraint Computation Centre - CTVR, University College, Cork, Ireland[3]
{r.rossi,s.prestwich}@4c.ucc.ie

*Retail replenishment is a high-value activity. According to the US Commerce Department, \$1.1 trillion in inventory supports \$3.2 trillion in annual US retail sales [...]. Improving distribution centre efficiency of just a few percentage points through advanced automation and real-time replenishment may deliver significant savings and require less capital to be tied up in inventory.*[1]

An interesting class of production/inventory control problems is the one that considers the single-location, single-product case under non-stationary stochastic demand, fixed production/ordering cost and per-unit holding cost. *Exact* and *efficient* approaches for computing optimal production/replenishment decisions are a key factor for achieving profitability in retail business. One of the possible policies that can be adopted to manage stocks is the *replenishment cycle policy* [6]. In this policy the inventory review times are set under a *here-and-now* strategy at the beginning of the planning horizon. These decisions are not affected by the actual demand realized in each period. On the other hand, for each inventory review we observe the actual demand realized in former periods to compute the actual order quantity. This provides an effective means of damping planning instability (deviations in planned orders, also known as *nervousness*) and coping with demand uncertainty. We developed Constraint Programming (CP) [1] models augmented with dedicated *cost-based filtering* [2] strategies in order to improve efficiency. Our approach is now the state-of-the-art technique for computing optimal replenishment cycle policy parameters [7, 4]. An extension considering a stochastic delivery lag is under development [5]. These techniques have been embedded in a **Decision Support System** (DSS) (Fig. 1) developed in Java on the top of Choco [3], an open source CP solver. Our DSS can run both as a web or desktop application. Our system receives the following inputs: a planning horizon of $N$ periods and a demand $d_t$ for each period $t \in \{1, \dots, N\}$, which is a normally distributed random variable with probability density function $g_t(d_t)$. We assume that demands in different periods are independent. A fixed delivery cost $a$ is considered for each order and also a linear holding cost

* Roberto Rossi is 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. 05/IN/I886. S. Armagan Tarim and Brahim Hnich are supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant No. SOBAG-108K027.

[1] "The Future of Retail Replenishment", Manhattan Associates ©, 2006.

$h$ is considered for each unit of product carried in stock from one period to the next. 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. We assume that it is not possible to sell back excess items to the vendor at the end of a period. Our DSS finds a replenishment plan that minimizes the expected total cost, which is composed of ordering costs, holding costs and penalty cost, over the $N$-period planning horizon. A second implementation considers service level constraints instead of penalties for stock-outs. As a service level constraint we require that, with a probability of at least a given value $\alpha$, at the end of each period the net inventory will be non-negative.

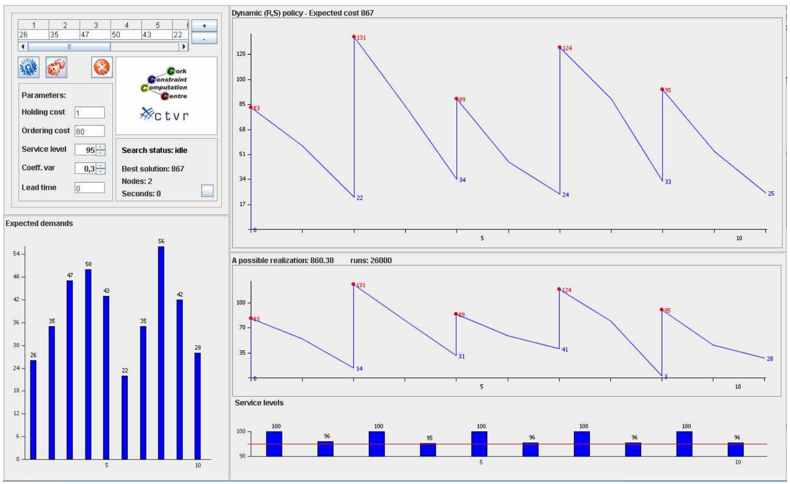


**Fig. 1.** Our Decision Support System. The main window is divided into 4 parts. The top-left window receives the inputs from the user and provides buttons for controlling the DSS. The bottom-left window plots the expected demand as a bar graph. The top-right window plots the optimal replenishment cycle policy: order periods (red dots) and the respective order-up-to-levels (red figures). In the bottom-right window the system is simulated and simulation statistics — number of runs and realized average cost per run — are reported. Finally, for each period the realized average service level per run is also reported.

## References

1. K. Apt. *Principles of Constraint Programming*. Cambridge University Press, Cambridge, UK, 2003.
2. F. Focacci, A. Lodi, and M. Milano. Optimization-oriented global constraints. *Constraints*, 7:351–365, 2002.
3. F. Laburthe and the OCRE project team. Choco: Implementing a cp kernel. Technical report, Bouygues e-Lab, France, 1994.
4. R. Rossi, S. A. Tarim, B. Hnich, and S. Prestwich. Replenishment planning for stochastic inventory systems with shortage cost. In *Proceedings of the International Conference on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimization Problems CP-AI-OR 2007*, pages 229–243. Springer Verlag, 2007. Lecture Notes in Computer Science No. 4510.
5. R. Rossi, S. A. Tarim, B. Hnich, and S. Prestwich. Computing replenishment cycle policy under non-stationary stochastic lead time. Technical report, Cork Constraint Computation Centre, 2008.
6. E. A. Silver, D. F. Pyke, and R. Peterson. *Inventory Management and Production Planning and Scheduling*. John-Wiley and Sons, New York, 1998.
7. S. A. Tarim, B. Hnich, R. Rossi, and S. Prestwich. Cost-based filtering techniques for stochastic inventory control under service level constraints. *Constraints*, 2008 (forthcoming).