Erratum to Bounds in "Serial Production/Distribution Systems Under Service Constraints"

Tamer Boyaci • Guillermo Gallego • Kevin H. Shang • Jing-Sheng Song
Faculty of Management, McGill University, Montreal, Quebec H3A 1G5, Canada
Department of Industrial Engineering and Operation Research, Columbia University, New York, New York 10027
The Fuqua School of Business, Duke University, Durham, North Carolina 27514
The Fuqua School of Business, Duke University, Durham, North Carolina 27514
tamer.boyaci@mcgill.ca • guillermo.gallego@columbia.edu • khshang@duke.edu • jssong@duke.edu

We noticed an error in the upper bound on the optimal system stock in Boyaci and Gallego (2001). We provide a procedure to compute the correct bound.
(*Inventory/Production; Multistage; Serial; Fill Rate; Base-Stock Policy; Bounds; Solution and Heuristics*)

1. Introduction

Boyaci and Gallego (2001) consider a J-stage serial base-stock system and study the problem of minimizing the expected inventory holding costs subject to fill-rate-type service constraints. They develop bounds on the total system-stock and on base-stock levels of each stage and incorporate these bounds into an algorithm to find an optimal inventory policy. They also present efficient heuristics for the problem. This note identifies and corrects an error in the upper bound on the optimal total system stock. Throughout the note we deal with nonnegative, integer, local (i.e., installation) base-stock policies.

In §2 we provide a brief description of the bound on the total system stock as presented and computed in Boyaci and Gallego. In §3 we provide a procedure to compute a correct upper bound on the optimal total system stock.

2. Upper Bound as Presented and Computed in Boyaci and Gallego (2001)

The upper bound on the total stock s_T in Boyaci and Gallego is built on the fact that the last stage, stage J, will always hold inventory and that the most upstream stage, stage 1, provides the least fill-rate protection. To find the "upper bound" $\bar{s}_T$, the following recursion is presented on Page 46:

$$\underline{s}_J = \min\{s\colon P(D_J < s) \geq \beta\},$$
$$\bar{s}_1 = \min\{s\colon P([D_1 - s_1]^+ + D_2 + \cdots + D_J \leq \underline{s}_J) < \beta\},$$
$$\bar{s}_T = \bar{s}_1 + \underline{s}_J.$$

Note that there are misprints in the definitions of both $\underline{s}_J$ and $\bar{s}_1$. The intended definitions, which are also used in the computations in Boyaci and Gallego, read:

$$\underline{s}_J = \min\{s\colon P(D_2 + \cdots + D_J < s) \geq \beta\}, \quad (1)$$
$$\bar{s}_1 = \min\{s\colon P([D_1 - s]^+ + D_2 + \cdots + D_J < \underline{s}_J) \geq \beta\}. \quad (2)$$

For any given base-stock policy $(s_1, \ldots, s_J)$ let $\beta(s_1, \ldots, s_J)$ denote the fill-rate. Notice that $\underline{s}_J = \min\{s\colon \beta(\infty, 0, \ldots, 0, s) \geq \beta\}$ and $\bar{s}_1 = \min\{s\colon \beta(s, 0, \ldots, 0, \underline{s}_J) \geq \beta\}$. The underlying logic of the upper bound on the total stock $\bar{s}_T$ is to restrict only stages 1 and J to hold inventory. In line with this logic, the last stage's minimum base-stock level is computed taking into account the fact that all intermediate stages do not hold inventory. It can be easily

1523-4614/03/0504/0372
1526-5498 electronic ISSN

seen that $\underline{s}_J$ as given by (1) is the lower bound on the *echelon* base-stock level of stage 2, $\sum_{k=2}^{J} s_k$. Because stage 1 provides the least fill-rate protection, the resulting $\bar{s}_1 + \underline{s}_J$ is an intuitive bound on the total system stock. On closer examination, however, it is possible to verify that there may be no finite s such that $\beta(s, 0, \ldots, 0, \underline{s}_J) \geq \beta$.[1] More importantly, the recursion does not recognize the possible existence of feasible base-stock policies with $\sum_{k=2}^{J} s_k \geq \underline{s}_J$ and $\sum_{k=1}^{J} s_k > \bar{s}_1 + \underline{s}_J$ that are candidates for the optimal solution.

3. Correction on the Upper Bound on s_T

Computing a guaranteed upper bound on s_T requires a more exhaustive search on the feasible base-stock policies. This can be achieved by developing upper and lower bounds on the base-stock levels.

Notice that for any feasible policy $(s_1, s_2, \ldots, s_J)$, it is necessary to have $\beta(\infty, s_2, \ldots, s_J) > \beta$ so that $\beta(s_1, s_2, \ldots, s_J) \geq \beta$ for some finite s_1. Consider the last stage J. The lower bound on s_J can be found as before, assuming $s_1 = s_2 = \cdots = s_{J-1} = \infty$:

$$\underline{s}_J = \min\{s\colon \beta(\infty, \ldots, \infty, s) > \beta\}.$$

Similarly, an upper bound on s_J can be found by assuming $s_1 = s_2 = \cdots = s_{J-1} = 0$:

$$\bar{s}_J = \min\{s\colon \beta(0, \ldots, 0, s) > \beta\}.$$

Consider now a given partial vector of base-stock levels $(-, -, \ldots, -, s_{k+1}, \ldots, s_J)$, which satisfies $\beta(\infty, \ldots, \infty, s_{k+1}, \ldots, s_J) > \beta$. A lower bound on s_k, $k \geq 2$, can be found by assuming that $s_1 = \cdots = s_{k-1} = \infty$:

$$\underline{s}_k(s_{k+1}, \ldots, s_J) = \min\{s\colon \beta(\infty, \ldots, \infty, s, s_{k+1}, \ldots, s_J) > \beta\},$$

and an upper bound on s_k, $k \geq 2$, can be found by assuming $s_1 = \cdots = s_{k-1} = 0$:

$$\bar{s}_k(s_{k+1}, \ldots, s_J) = \min\{s\colon \beta(0, \ldots, 0, s, s_{k+1}, \ldots, s_J) > \beta\}.$$

Notice that given the partial vector $(-, s_2, \ldots, \ldots, s_J)$, there is no need to calculate upper and lower bounds for stage 1. This is because there is a unique, minimum base-stock level s_1 $(s_2, \ldots, s_J)$ that satisfies the fill-rate constraint, i.e.,

$$s_1(s_2, \ldots, s_J) = \min\{s\colon \beta(s, s_2, \ldots, s_J) \geq \beta,$$

and any $s_1 > s_1(s_2, \ldots, s_J)$ results in higher cost.

The bounds on local base-stock levels can be used dynamically in a procedure to find the upper bound $\bar{s}_T$. For any $s_J \in [\underline{s}_J, \bar{s}_J]$, the procedure would first compute the bounds $(\underline{s}_{J-1}(s_J), \bar{s}_{J-1}(s_J))$. Then for any $s_{J-1} \in [\underline{s}_{J-1}(s_J), \bar{s}_{J-1}(s_J)]$, the bounds $(\underline{s}_{J-2}(s_{J-1}, s_J), \bar{s}_{J-2}(s_{J-1}, s_J))$ would be computed and the process would be repeated for $s_{J-2} \in [\underline{s}_{J-2}(s_{J-1}, s_J), \bar{s}_{J-2}(s_{J-1}, s_J)]$ until stage 1 is reached. At this stage, it is possible to compute $s_1(s_2, \ldots, s_J)$ and the resulting total system stock $s_T = s_1(s_2, \ldots, s_J) + \sum_{k=2}^{J} s_k$. Repeating this for all $(s_2, \ldots, s_J)$ within their respective bounds and choosing the highest total system stock would then yield the upper bound $\bar{s}_T$. The procedure Bound(k, s) below presents a formal description of this process. After initializing $\bar{s}_T = 0$, a single call to Bound$(J, 0)$ yields the upper bound $\bar{s}_T$. Notice that this is in essence a search algorithm that evaluates only the feasible policies, in the same spirit as the optimization algorithm in Boyaci and Gallego (only that this procedure is bottom-up as opposed to top-down).

Bound(k, s)
- (1) Calculate $\underline{s}_k$ and $\bar{s}_k$ based on partial vector $(-, -, \ldots, -, s_{k+1}, \ldots, s_J)$. Set $s_k = \underline{s}_k$.
- (2) DO WHILE $s_k \leq \bar{s}_k$
  - IF $k > 1$,
    - Call Bound$(k-1, s)$.
  - ELSE
    - Calculate $s_1(s_2, \ldots, s_J)$ and $s_T = s_1(s_2, \ldots, s_J) + \sum_{k=2}^{J} s_k$.
    - IF $s_T > \bar{s}_T$, THEN
      - $\bar{s}_T = s_T$.

[1] This was not an issue for the problem instances solved in Boyaci and Gallego (2001). This is because when $\underline{s}_J$ is computed as in (1), $\bar{s}_1$ would be infinite only under the unlikely event that $\beta(\infty, 0, \ldots, 0, \underline{s}_J) = \beta$.

```
    ENDIF
  ENDIF
  SET s_k = s_k + 1
  ENDO
```

Computing the bound $\bar{s}_T$ requires considerable computational effort. Given this bound, the algorithm in Boyaci and Gallego (2001) generates an optimal base-stock policy that guarantees the desired fill-rate. Alternatively, the optimization step can be incorporated into the above upper-bound algorithm. For every policy $(s_1(s_2, \ldots, s_J), s_2, \ldots, s_J))$ considered in the algorithm, it is possible also to evaluate the inventory cost. The bottom-up performance evaluation procedure in Shang and Song (2001) can be used for this purpose, as well as for evaluating the fill-rate. Keeping track of the costs would then yield the optimal base-stock policy.

We end with a brief comment concerning feasible policies, which we defined as policies with fill-rates at least β. Because of the discreteness of the policies, it may not be possible to exactly achieve the desired fill-rate. In this event, the inventory manager may try to lower the average holding cost by using a combination of base-stock policies that *on average* have the desired fill-rate. Despite the fact that such savings come at the expense of increased volatility in the fill-rate, this "mixed" policy is indeed the optimal policy form for the studied problem (Axsäter 2003).

References

Axsäter, S. 2003. Note: Optimal policies for serial inventory systems under fill rate constraints. *Management Sci.* **49** 247–253.

Boyaci, T., G. Gallego. 2001. Serial production/distribution systems under service constraints. *Manufacturing Service Oper. Management* **3** 43–50.

Shang, H. K., J. S. Song. 2001. Analysis of serial supply chains with a service constraint. Working paper, Graduate School of Management, University of California, Irvine. http://www.gsm.uci.edu/~song/research.htm#working.

Received: June 30, 2003; accepted: July 3, 2003; Senior Editor: Garrett J. van Ryzin.