6.3 GUARANTEED-SERVICE MODELS

6.3.1 Introduction

Figure 6.6 depicts the supply chain for a digital camera made by Kodak. Each stage represents an activity (as in interpretation (2) from Section 6.1): either a processing activity such as packaging or testing, or an assembly activity such as combining a wafer and an "imager base" to construct an "imager assembly." These activities may occur at different locations or together at the same location. Each stage functions as an autonomous unit that can hold safety stock, place orders to upstream stages, and so on.

The question of interest here is, which stages should hold safety stock, and how much? It may not be necessary for all stages to hold safety stock, but only a few. These stages serve as buffers to absorb all of the demand uncertainty in the supply chain. This problem is a strategic one, since the location of safety stock is a design problem that is costly to change frequently. This problem is therefore known as the *strategic safety stock placement problem* (SSSPP).

The supply chain operates in an infinite-horizon, periodic-review setting, and each stage follows a base-stock policy. Each stage quotes a lead time, or *committed service time* (CST), to its downstream stage(s) within which it promises to deliver each order. As we will see, there is a direct relationship between the CST and the safety stock (and base-stock level) required at each stage. The goal of the strategic safety stock placement model is to choose the CST (and, therefore, the safety stock and base-stock level) at each stage in order to minimize the expected holding cost in each period.

Each stage is required to provide 100% service to its downstream stage(s). In other words, each stage is obligated to deliver every order within the CST *regardless of the size of the order*. In order to enforce this restriction, we will have to assume that the demand is bounded. We will discuss this assumption further in Section 6.3.2.

The guaranteed-service assumption was first used by Kimball in 1955 (later reprinted as Kimball 1988). Simpson (1958) applied it to serial systems and Graves (1988) discussed how to solve the resulting safety stock optimization problem. Inderfurth (1991), Minner (1997), and Inderfurth and Minner (1998) discuss dynamic programming (DP) approaches for distribution and assembly systems. Graves and Willems (2000) extend this to tree systems, and Magnanti et al. (2006) and Humair and Willems (2011) allow general networks that include (undirected) cycles.

We will build gradually to tree networks similar to the one pictured in Figure 6.6, considering first the single-stage case, then serial systems, and finally tree networks. First, we will discuss the demand process.

Throughout Section 6.3, h_i will be used to represent the *local* holding cost at stage i. (In Section 6.2, it represented the echelon holding cost.)

6.3.2 Demand

We assume that the demand *in any interval of time* is bounded. In practice, this is not a terribly realistic assumption (unless the bound is very large), but it is necessary in this model to guarantee 100% service. One way to model the demand is simply to truncate the right tail of the demand distribution. That is, if demand is normally distributed, we simply ignore any demands greater than, say, z_α standard deviations above the mean, for some constant α. This is the approach we will take throughout.

In particular, consider a stage that faces external demand (as opposed to serving other downstream stages). Suppose the demand per period is distributed $N(\mu, \sigma^2)$. Then we will assume that the total demand in any τ periods is bounded by

$$D(\tau) = \mu\tau + z_\alpha\sigma\sqrt{\tau} \tag{6.33}$$

for some constant α. In other words, we assume that the demand in τ consecutive periods is no more than z_α standard deviations above its mean, since the mean demand in τ periods is $\mu\tau$ and the standard deviation is $\sigma\sqrt{\tau}$. This implies that the demand in a single period is bounded by $\mu + z_\alpha\sigma$. The reverse implication, however, is not true: Assuming the single-period demand is bounded by $\mu + z_\alpha\sigma$ implies that the τ-period demand is bounded by $\mu\tau + z_\alpha\sigma\tau$; it does not imply the stronger bound of $\mu\tau + z_\alpha\sigma\sqrt{\tau}$.

If, in actuality, the demand in a given τ-period interval exceeds $D(\tau)$, the excess demands are assumed to be handled in some other manner—say, by outsourcing, scheduling overtime shifts, or by some other method not captured in the model. This allows us to ignore the demands in the tail and pretend the demand never exceeds its bound.

We will use the demand bound in (6.33), but any other bound $D(\tau)$ is acceptable, with suitable changes to the derivations below.

6.3.6 Solution Method

We will solve the SSSPP on a tree system using DP. In principle, the approach is similar to the DP for the serial system in Section 6.3.4, but it is more complicated for two main reasons. First, computing the cost of a given decision is trickier than in the serial system. Second, in the serial system, it is clear which stage follows a given stage, and hence, how the DP recursion should be structured. In this problem, this is less clear, since each stage may have more than one upstream and/or downstream neighbor.

6.3.6.1 *Labeling the Stages*

We will address the second issue first. The DP algorithm requires us to relabel the stages so that each stage (other than stage N) *has exactly one adjacent stage with a higher index*. When we describe the algorithm, it will be clear why this is required. The relabeling is performed using Algorithm 6.1. In the algorithm, L represents the set of stages that have been labeled so far and U represents the set of unlabeled stages.

Algorithm 6.1 Relabel stages

$L \leftarrow \emptyset, U \leftarrow \{1, \dots, N\}$ ▷ Initialization
for $k = 1, \dots, N$ **do** ▷ Labeling stages
    choose $i \in U$ such that i is adjacent to at most one other stage in U
    label i with index k
    $L \leftarrow L \cup \{i\}, U \leftarrow U \setminus \{i\}$
end for
return labels

6.3.6.3 *Dynamic Programming Algorithm*

Algorithm 6.2 gives the pseudocode for the DP algorithm.

Algorithm 6.2 DP algorithm for tree SSSPP

1: **for** $k = 1, \dots, N-1$ **do**
2:     **if** p_k is downstream from k **then**
3:         calculate $\theta_k^o(S)$ for $S = 0, 1, \dots, M_k$
4:     **else**
5:         calculate $\theta_k^i(SI)$ for $SI = 0, 1, \dots, M_k - T_k$
6:     **end if**
7: **end for**
8: $SI^* \leftarrow \text{argmin}_{SI=0,1,\dots,M_N-T_N} \theta_N^i(SI)$
9: **return** $\theta_N^i(SI^*)$

The algorithm returns the optimal objective value, which is equal to the minimum value of $\theta_N^i(SI)$ found in line 8. The optimal solution is found by "backtracking," similar to the Wagner–Whitin algorithm.

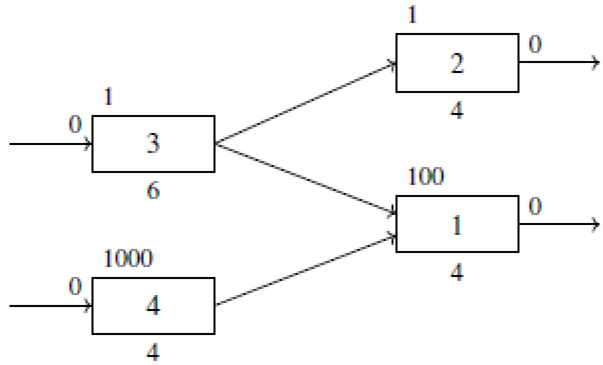
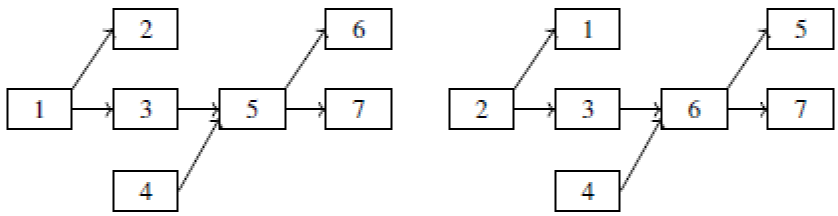


Figure 6.11 A counterexample to the “all-or-nothing” claim for tree systems.



(a) Original network. (b) Relabeled network.

Figure 6.12 Relabeling the network.

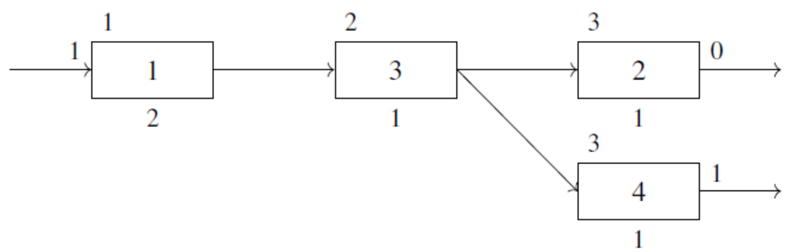


Figure 6.13 Example network for SSSPP DP algorithm for tree systems.

☐ **EXAMPLE 6.5**

We will illustrate the algorithm on the network pictured in Figure 6.13. The numbers below the stages are the processing times T_i. The number on the inbound arrow to stage 1 indicates that $SI_1 = 1$, while the outbound numbers from stages 2 and 4 indicate the fixed CSTs s_i. The holding costs are noted above each stage and are equal to 1 at the first echelon (stage 1), 2 at the second echelon (stage 3), and 3 at the third echelon (stages 2 and 4). Assume $z_\alpha = 1$ at all stages and $\sigma_2 = \sigma_4 = 1$; then $\sigma_1 = \sigma_3 = \sqrt{2}$. Note that the stages have already been relabeled so that each stage has exactly one neighbor with a higher index. Examining the longest path to each node, we get $M_1 = 3$, $M_2 = 5$, $M_3 = 4$, $M_4 = 5$.

Since $p_1 = 3$ is downstream from 1, we first compute $\theta_1^o(S)$ for $S = 0, \ldots, M_1 = 3$. Since 1 is a supply stage, the minimum over SI only considers $SI = 1$.

and

$$\min_{S \le y \le M_2 - T_2} \{\theta_2^i(y)\} = \theta_2^i(S)$$

for all SI and S, and we have:

$$\begin{aligned}
\theta_3^o(0) &= \min_{0 \le SI \le 3} \{c_3(0, SI)\} = 8.28 \\
&c_3(0,0) = 2\sqrt{2}\sqrt{0+1-0} + \theta_1^o(0) + \theta_2^i(0) = 2.83 + 2.45 + 3.00 = 8.28 \\
&c_3(0,1) = 2\sqrt{2}\sqrt{1+1-0} + \theta_1^o(1) + \theta_2^i(0) = 4.00 + 2.00 + 3.00 = 9.00 \\
&c_3(0,2) = 2\sqrt{2}\sqrt{2+1-0} + \theta_1^o(2) + \theta_2^i(0) = 4.90 + 1.41 + 3.00 = 9.31 \\
&c_3(0,3) = 2\sqrt{2}\sqrt{3+1-0} + \theta_1^o(3) + \theta_2^i(0) = 5.66 + 0.00 + 3.00 = 8.66 \\
\theta_3^o(1) &= \min_{0 \le SI \le 3} \{c_3(1, SI)\} = 6.69 \\
&c_3(1,0) = 2\sqrt{2}\sqrt{0+1-1} + \theta_1^o(0) + \theta_2^i(1) = 0.00 + 2.45 + 4.24 = 6.69 \\
&c_3(1,1) = 2\sqrt{2}\sqrt{1+1-1} + \theta_1^o(1) + \theta_2^i(1) = 2.83 + 2.00 + 4.24 = 9.06 \\
&c_3(1,2) = 2\sqrt{2}\sqrt{2+1-1} + \theta_1^o(2) + \theta_2^i(1) = 4.00 + 1.41 + 4.24 = 9.65 \\
&c_3(1,3) = 2\sqrt{2}\sqrt{3+1-1} + \theta_1^o(3) + \theta_2^i(1) = 4.90 + 0.00 + 4.24 = 9.14 \\
\theta_3^o(2) &= \min_{1 \le SI \le 3} \{c_3(2, SI)\} = 7.20 \\
&c_3(2,1) = 2\sqrt{2}\sqrt{1+1-2} + \theta_1^o(1) + \theta_2^i(2) = 0.00 + 2.00 + 5.20 = 7.20 \\
&c_3(2,2) = 2\sqrt{2}\sqrt{2+1-2} + \theta_1^o(2) + \theta_2^i(2) = 2.83 + 1.41 + 5.20 = 9.44 \\
&c_3(3,2) = 2\sqrt{2}\sqrt{2+1-3} + \theta_1^o(2) + \theta_2^i(3) = 0.00 + 1.41 + 6.00 = 7.41 \\
&c_3(3,3) = 2\sqrt{2}\sqrt{3+1-3} + \theta_1^o(3) + \theta_2^i(3) = 2.83 + 0.00 + 6.00 = 8.83 \\
\theta_3^o(4) &= \min_{3 \le SI \le 3} \{c_3(4, SI)\} = 6.71 \\
&c_3(4,3) = 2\sqrt{2}\sqrt{3+1-4} + \theta_1^o(3) + \theta_2^i(4) = 0.00 + 0.00 + 6.71 = 6.71
\end{aligned}$$

Finally, we compute $\theta_4^i(SI)$ for $SI = 0, \ldots, M_4 - T_4 = 4$. Again, 4 is a demand stage, so the minimum ranges only over $S = 1$. However, we need to take greater care with the minimization in (6.50) since $\theta_3^o(x)$ is not monotonic in x.

$$\begin{aligned}
\theta_4^i(0) &= \min_{S=1}\{c_4(S,0)\} = c_4(1,0) = 3\sqrt{0+1-1} + \min_{0 \le x \le 0}\{\theta_3^o(x)\} \\
&= 0.00 + \theta_3^o(0) = 0.00 + 8.28 = 8.28 \\
\theta_4^i(1) &= \min_{S=1}\{c_4(S,1)\} = c_4(1,1) = 3\sqrt{1+1-1} + \min_{0 \le x \le 1}\{\theta_3^o(x)\} \\
&= 3.00 + \theta_3^o(1) = 3.00 + 6.69 = 9.69 \\
\theta_4^i(2) &= \min_{S=1}\{c_4(S,2)\} = c_4(1,2) = 3\sqrt{2+1-1} + \min_{0 \le x \le 2}\{\theta_3^o(x)\} \\
&= 4.24 + \theta_3^o(1) = 4.24 + 6.69 = 10.93 \\
\theta_4^i(3) &= \min_{S=1}\{c_4(S,3)\} = c_4(1,3) = 3\sqrt{3+1-1} + \min_{0 \le x \le 3}\{\theta_3^o(x)\} \\
&= 5.20 + \theta_3^o(1) = 5.20 + 6.69 = 11.89 \\
\theta_4^i(4) &= \min_{S=1}\{c_4(S,4)\} = c_4(1,4) = 3\sqrt{4+1-1} + \min_{0 \le x \le 4}\{\theta_3^o(x)\} \\
&= 6.00 + \theta_3^o(1) = 6.00 + 6.69 = 12.69
\end{aligned}$$

The minimum value is $\theta_4^i(0) = 8.28$, so 8.28 is the optimal cost. The optimal solution has an inbound time of 0 to stage 4, which means $S_3^* = 0$. Since $\theta_3^o(0)$ is minimized when $SI = 0$, the inbound time to stage 3 is 0, hence $S_1^* = 0$. The optimal solution is therefore $\mathbf{S}^* = (0, 0, 0, 1)$. The safety stock at each stage is

$$\begin{aligned} SS_1 &= \sqrt{2}\sqrt{1+2-0} = 2.45 \\ SS_2 &= \sqrt{0+1-0} = 1.00 \\ SS_3 &= \sqrt{2}\sqrt{0+1-0} = 1.41 \\ SS_4 &= \sqrt{0+1-1} = 0.00 \end{aligned}$$

Note that the safety stock is pushed upstream as far as possible: Stage 2 needs to hold *some* safety stock since its processing time is 1 and its CST is 0. Since the holding cost at stages 2 and 4 is high, it is important for stage 3 to quote a CST of 0, so it, too, must hold safety stock. But the bulk of the safety stock is held at stage 1 since the holding cost is smallest there. Stage 1, then, absorbs most of the demand uncertainty by serving as the supply chain's main buffer. □

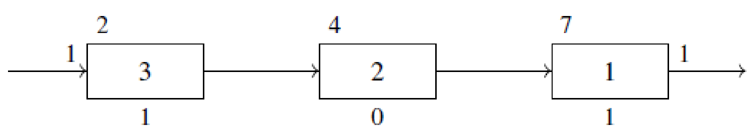


Figure 6.10 Example network for SSSPP DP algorithm for serial systems.

☐ **EXAMPLE 6.3**

Consider the network pictured in Figure 6.10. The numbers below the stages are the processing times T_i. The number on the inbound arrow to stage 3 indicates that $SI_3 = 1$, while the outbound number from stage 1 indicates that the fixed CST $s_1 = 1$. The holding costs at stages 1, 2, and 3 are 7, 4, and 2, respectively, and are noted above each stage. Assume $z_\alpha = \sigma_i = 1$ at all stages.

First note that $SI_N + \sum_{j=k+1}^{N} T_j = 2$ at stages 1 and 2. (SI is fixed to 1 at stage 3.) These are the maximum SI values that we must consider at each stage.

We consider stage $k = 1$ first. From (6.43), $\theta_1(SI) = 7\sqrt{SI}$ for all SI:

$$\begin{aligned}
\theta_1(0) &= 0 \\
\theta_1(1) &= 7 \\
\theta_1(2) &= 7\sqrt{2} = 9.90.
\end{aligned}$$

Next, at stage 2, we use (6.44):

$$\theta_2(0) = \min_{S=0}\{4\sqrt{0+0-S} + \theta_1(S)\} = 0$$

$$\begin{aligned}
\theta_2(1) &= \min_{0\le S\le 1}\{4\sqrt{1+0-S} + \theta_1(S)\} \\
&= \min\{4+0, 0+7\} = 4 \\
\theta_2(2) &= \min_{0\le S\le 2}\{4\sqrt{2+0-S} + \theta_1(S)\} \\
&= \min\{4\sqrt{2}+0, 4+7, 0+7\sqrt{2}\} = 4\sqrt{2} = 5.6.
\end{aligned}$$

Finally, at stage 3, we have only one SI value to consider since SI_3 is fixed at 1:

$$\begin{aligned}
\theta_3(1) &= \min_{0\le S\le 2}\{2\sqrt{1+1-S} + \theta_2(S)\} \\
&= \min\{2\sqrt{2}+0, 2+4, 0+5.6\} \\
&= 2\sqrt{2} = 2.83.
\end{aligned}$$

Since $S = 0$ solved the minimization for $\theta_3(1)$, we have $S_3^* = 0$. Therefore, $SI = 0$ at stage 2, and $S = 0$ solved the minimization for $\theta_2(0)$ as well. Finally, $s_1 = 1$. Therefore, the optimal CSTs are $\mathbf{S}^* = (0, 0, 1)$. These CSTs imply that the NLTs at stages 1, 2, and 3 are 0, 0, and 2, respectively. Therefore, the optimal safety stock levels are as follows:

$$\begin{aligned}
SS_1 &= \sqrt{0} = 0.00 \\
SS_2 &= \sqrt{0} = 0.00 \\
SS_3 &= \sqrt{2} = 1.41
\end{aligned}$$

REFERENCES

1) Fundamentals of Supply Chain Theory – Snyder and Shen