Technical Note: Some Structural Results on Acyclic Supply Chains

Junmin Shi and Yao Zhao

Department of Supply Chain Management and Marketing Sciences

Rutgers University, Newark, NJ

May 5, 2010

Abstract

We consider acyclic supply chains under the full backorder assumption and reveal several simple yet unique properties. Most interestingly, we identify conditions for the best dedicated stocking policy to outperform the best shared stocking policy, and for an acyclic supply chain to be decomposable into a simpler network (e.g., tree). We specify ways to decompose them.

Key words: Acyclic supply chains, dedicated stock, shared stock, decomposition.

1 Introduction

In this paper, we consider acyclic supply chains facing random demand under the assumption that unsatisfied demand at each node is fully backlogged. The objective of this paper is to develop structural results on how to manage inventory in such a supply chain by utilizing a sample-path analysis of the system dynamics for various inventory policies and demand processes.

It is not rare to find an acyclic supply chain in practice because common components may be shared by multiple subassemblies, many of which are assembled into one final product. We refer the reader to Billington, et al. (2004) for a real-world example, and to Zipkin (2000) chapter 5 and Axsater (2006) chapter 8 for more illustrations. More formally, let a node be a unique combination of facility and product, an arc be an immediate supply-demand relationship between two nodes, and a "transit time" be the processing time at a node or the shipping time from one node to the other (Svoronos and Zipkin 1991). An acyclic supply chain is a network of nodes and arcs with no cycles but possibly multiple directed paths between some pairs of nodes. An example is illustrated in Figure 1. This is a four-stage network with one final product, two subassemblies

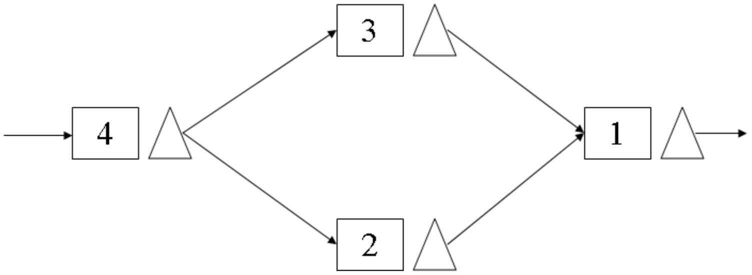

Figure 1: An example of acyclic supply chains.

and one common component. Multiple copies of the component travel by different paths and are eventually assembled into one unit of the final product.

For acyclic supply chains with random demand, so far only guaranteed service-time models are considered in the literature. We refer to Lesnaia (2004), Minner (2000), Magnanti, et al. (2006) and Humair and Willems (2007) for various solution methods, and to Lesnaia (2004) and Graves and Willems (2003) for recent reviews. Under the guaranteed service-time assumption, one does not model what happens when demand exceeds on-hand stock, but rather assumes that the supply chain responds with an extraordinary measure which always ensures fulfillment of the demand at the service time. Thus, the lead times are constant. The inventory positioning problem is then formulated as a mathematical programming problem that minimizes a concave cost function on a polyhedron. It is reported in this research stream that acyclic supply chains are much harder to solve than tree networks.

In this paper, we study the stochastic service-time model for acyclic supply chains with random demand. In this model, all unsatisfied demands are fully backlogged at each node, and the backorder is cleared only when on-hand inventory becomes available. Thus, lead times are random and service times cannot be guaranteed.

Under the stochastic service-time model, Rosling (1989) characterizes the optimal inventory policy for pure assembly systems with constant transit times. For more general assembly systems with stochastic transit times and/or multiple products, the optimal policy is not known but an exact analysis is given for systems with heuristic policies under various assumptions of supply systems and demand processes, see Song and Zipkin (2003), de Kok and Fransoo (2003) and Simchi-Levi and Zhao (2007) for literature reviews.

Acyclic supply chains are combinations of assembly and distribution systems. Consider the four-stage acyclic supply chain in Figure 1, it is clear that the analysis and results of Rosling (1989) do not apply because from node 4, there are two directed paths that lead to node 1. The acyclic supply chain is also different from pure distribution systems because the ordering decisions of nodes 2 and 3 are coupled due to the assembly operation at node 1. The optimal inventory policy for such a system is unknown and at least as complex as those of pure distribution systems. In practice only simple and suboptimal heuristic policies are used, e.g., base-stock policies. However, to our best knowledge, these heuristic policies are not studied for acyclic supply chains with random demand under the stochastic service-time model. System dynamics are not understood.

For a node in an acyclic supply chain, it is common to pool inventory into one stock pile to satisfy demand from all downstream paths on a first-come-first-serve (FCFS) basis. We call such a policy the *shared stocking policy.* This policy allows one to achieve the *risk-pooling effect.* The policy is assumed by all aforementioned work (see Graves and Willems 2000, Eq. 3) and confirmed by reports of industrial practice (see, e.g., Billington, et al. 2004). Alternatively, one can split inventory into multiple stock piles and dedicate each to a unique downstream path. We call such a policy the *dedicated stocking policy.* This policy provides the flexibility of setting different stock levels for different paths, and therefore, better positions inventory along each path. We call such an effect the *stock-positioning effect.*

In this paper, we focus on simple heuristic policies that are widely implemented in practice and extensively studied in the literature. We demonstrate several unique and important properties for acyclic supply chains with a single final outlet by a sample-path analysis and a numerical study (§2-3). Most interestingly, we show that under certain assumptions, for any node serving multiple paths that lead to the final outlet, the best *dedicated stocking policy* always outperforms the best *shared stocking policy.* For example, in the 4-node network of Figure 1, node 4 serves two paths (via nodes 2 and 3) leading to the same outlet, node 1. Because the order processes of nodes 2 and 3 are all driven by node 1, they can be synchronized in both timing and quantity under certain inventory policies. In these cases, the shared stocking policy at node 4 does not achieve the *risk-pooling effect*; but the dedicated stocking policy at node 4 does achieve the *stock-positioning effect* and thus can strictly outperform the former. When the transit times are constant, such an acyclic supply chain can be decomposed into a tree network with an equal or better performance. We discuss extensions of the results to acyclic supply chains with multiple outlets in §4.

2 The Basic Model

Throughout this paper, we assume that unsatisfied demand at each node is fully backordered, outside suppliers have ample stock, and transit times are sequential (Zipkin 2000, Chapter 7) and

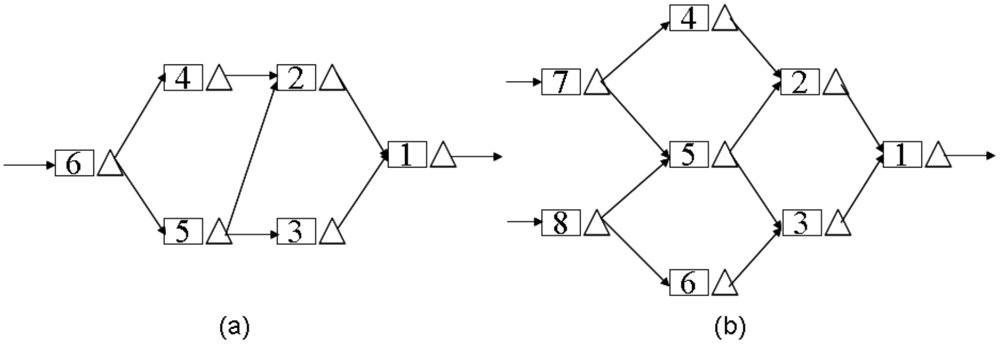

Figure 2: Examples of acyclic supply chains with a single final outlet.

apply to each shipment as a whole. We also assume that demand can be split, i.e., each unit of demand can be satisfied separately, and in assembly operations, a unit can be assembled only if all required components are available.

For the ease of exposition, we define the "basic model" by the following assumptions.

Assumption 1 *The supply chain has a single final outlet; Each node utilizes an installation continuous-time base-stock policy with a non-negative base-stock level; the bills of material (BOM) matrix has unit elements; External demand follows Poisson process, and demand is satisfied on a FCFS basis at each node. Under the shared stocking policy, each node keeps inventory in units of m where m is the number of downstream paths that connect the current node to the final outlet; Transit times are constant.*

In this section, we focus on the basic model for acyclic supply chains (see Figure 2 for more examples with a single final outlet). We will discuss extensions of the results to stochastic, sequential and exogenous transit times, general BOM matrix and negative base-stock levels whenever possible.

Recursive Equations. Consider an acyclic supply chain under the shared stocking policy under Assumption 1. We denote it by $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the node set and $\mathcal{A}$ is the arc set. The nodes in the network are indexed by $1, 2, \ldots, N$, where $N = |\mathcal{N}|$ is the total number of nodes in the network. For node $k \in \mathcal{N}$, there are totally $m_k > 0$ distinct downstream paths that connect it to the final outlet. For node k, let its supplier set be $\mathcal{S}(k) = \{j \in \ \mathcal{N} : (j, k) \in \mathcal{A}\}$, and its customer set be $\mathcal{C}(k) = \{i \in \ \mathcal{N} : (k, i) \in \mathcal{A}\}$.

By Assumption 1, we note that the demand arrival processes at all nodes are identical. While the final outlet faces unit demand, node k faces demand always in the quantity of m_k units. By Assumption 1, node k keeps inventory in units of m_k. Regarding each demand arrival as one flow unit at each node, we let the base-stock level at node k be $m_k s_k$.

For node k, let n_k be the index of demand arrivals, $X_k(n_k)$ be the stockout delay for the demand arrival n_k, and $W_k(n_k)$ be the inventory holding time at node k for the corresponding item that satisfies the demand arrival n_k. For a demand arrival n_k, the corresponding order at node k is placed $T(s_k, n_k)$ time in advance (Simchi-Levi and Zhao 2005), where $T(s_k, n_k)$ is the total time if one starts at the arrival time of demand n_k and counts backwards until the number of demand arrivals at stage k reaches s_k (we drop subscript k from T because all nodes face identical demand arrival processes). The corresponding order is a demand at each supplier of node k, $j \in \mathcal{S}(k)$, which can be indexed by $n_k - s_k$. Finally, let $\tilde{L}_k(n_k)$ be the total replenishment lead time at node k for the order triggered by demand arrival n_k, and $W_{i,k}(n_k)$ be the component i's waiting time at node k before processing (due to the unavailability of other components) for the order triggered by demand arrival n_k.

The recursive equations at node k can be written as follows.

$$\begin{aligned} X_k(n_k) &= [\tilde{L}_k(n_k - s_k) - T(s_k, n_k)]^+ & (1)\\ W_k(n_k) &= [T(s_k, n_k) - \tilde{L}_k(n_k - s_k)]^+ & (2)\\ \tilde{L}_k(n_k - s_k) &= \max_{j \in \mathcal{S}(k)} \{X_j(n_k - s_k) + L_{j,k}\} + L_k & (3)\\ W_{j,k}(n_k - s_k) &= \max_{l \in \mathcal{S}(k)} \{X_l(n_k - s_k) + L_{l,k}\} - [X_j(n_k - s_k) + L_{j,k}],\ j \in \mathcal{S}(k), & (4) \end{aligned}$$

where $L_{i,j}$ is the transit time from node i to node j, and L_i is the transit time at node i.

To illustrate the complexity and unique properties of acyclic supply chains, we apply Eqs. (1)-(4) to the four-stage acyclic supply chain in Figure 1. For node 1,

$$\begin{aligned} X_1(n_1) &= [\tilde{L}_1(n_1 - s_1) - T(s_1, n_1)]^+ & (5)\\ W_1(n_1) &= [T(s_1, n_1) - \tilde{L}_1(n_1 - s_1)]^+ & (6)\\ \tilde{L}_1(n_1 - s_1) &= \max\{X_2(n_1 - s_1) + L_{2,1}, X_3(n_1 - s_1) + L_{3,1}\} + L_1 & (7)\\ W_{i,1}(n_1 - s_1) &= \max\{X_2(n_1 - s_1) + L_{2,1}, X_3(n_1 - s_1) + L_{3,1}\} - X_i(n_1 - s_1) - L_{i,1},\ i = 2, 3. & (8) \end{aligned}$$

For nodes $k = 2, 3$,

$$\begin{aligned} X_k(n_1 - s_1) &= [\tilde{L}_k(n_1 - s_1 - s_k) - T(s_k, n_1 - s_1)]^+ & (9)\\ W_k(n_1 - s_1) &= [T(s_k, n_1 - s_1) - \tilde{L}_k(n_1 - s_1 - s_k)]^+ & (10)\\ \tilde{L}_k(n_1 - s_1 - s_k) &= X_4(n_1 - s_1 - s_k) + L_{4,k} + L_k. & (11) \end{aligned}$$

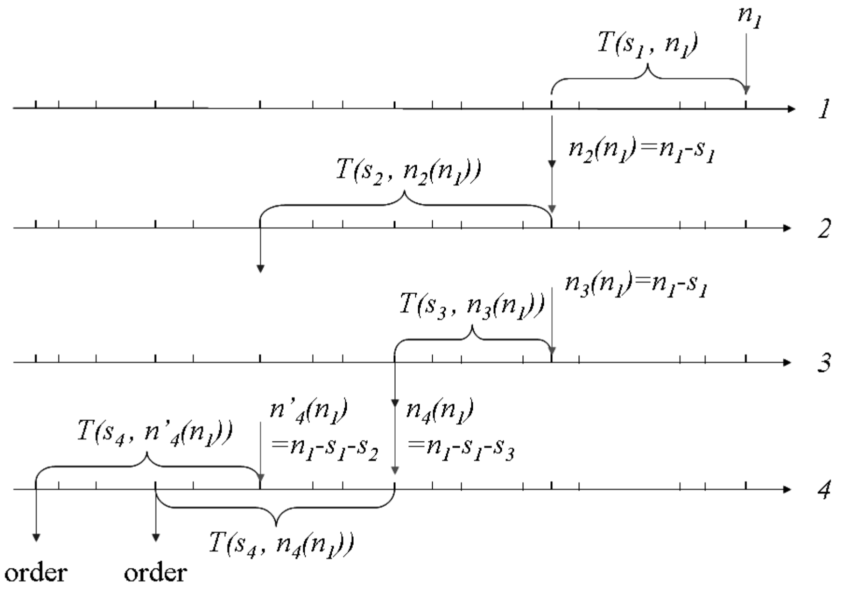

Figure 3: The timeline of the four-stage acyclic supply chain.

Figure 3 provides a visual aid where $n_k(n_1)$ is the demand arrival at node k that corresponds to the external demand n_1. Clearly, $n_k(n_1) = n_1 - s_1$ for $k = 2, 3$. For the ease of exposition, define $n_4(n_1) = n_1 - s_1 - s_3$ and $n'_4(n_1) = n_1 - s_1 - s_2$. For node 4,

$$\begin{aligned} X_4(n_1 - s_1 - s_k) &= [\tilde{L}_4(n_1 - s_1 - s_k - s_4) - T(s_4, n_1 - s_1 - s_k)]^+ \qquad (12) \\ W_4(n_1 - s_1 - s_k) &= [T(s_4, n_1 - s_1 - s_k) - \tilde{L}_4(n_1 - s_1 - s_k - s_4)]^+, \quad k = 2, 3. \qquad (13) \end{aligned}$$

An acyclic supply chain differs from simpler networks, e.g., tree, because to satisfy a demand n_1 at node 1, we need to consider multiple demand arrivals, $n_4(n_1) = n_1 - s_1 - s_3$ and $n'_4(n_1) = n_1 - s_1 - s_2$, at node 4 if $s_2 \neq s_3$. Unlike the simpler networks where n_k is uniquely determined by n_1 for each upstream node k, here n_4 also depends on the path that connects node 4 and node 1. The complexity introduced by such a path dependence is that to characterize $X_1(n_1)$, one needs to estimate the joint probability distribution of consecutive transit times at one node (e.g., $\tilde{L}_4(n_4(n_1) - s_4)$ and $\tilde{L}_4(n'_4(n_1) - s_4)$ at node 4).

Based on the above analysis, we now develop system properties that are unique for acyclic supply chains. In particular, we focus on three issues: stock balancing, shared vs. dedicated stocking policy, and decomposition of an acyclic supply chain to a tree.

Stock Balancing. An important feature of acyclic supply chains is that multiple units of one component can travel by different paths and are assembled into one final product. For instance, consider the acyclic network in Figure 1 where two units of the item at node 4 are assembled into one product at node 1. If $s_2 = s_3$, then $n'_4(n_1) = n_4(n_1)$, and the two units assembled into one product at node 1 are in the same order placed by node 4; otherwise, they are in different orders. If $s_2 > s_3$, then $n'_4(n_1) < n_4(n_1)$. By the non-crossing property of sequential transit times, the corresponding order for demand $n'_4(n_1)$ is placed and replenished at node 4 earlier than the corresponding order for demand $n_4(n_1)$ in any event (see Figure 3). Thus, s_2 and s_3 not only affect $T(\cdot,\cdot)$ in Eqs. (9)-(10), but also affect $\tilde{L}_k(\cdot), k = 2, 3$ because they determine the sequence of the corresponding orders at node 4. The following proposition summarizes the intuition (see Appendix for a proof).

Proposition 1 *Consider the acyclic network in Figure 1 under Assumption 1. Let $s_2 \geq s_3$.*

1. *If $L_{4,2} + L_2 \leq L_{4,3} + L_3$, then in any event,*

$$X_2(n_1 - s_1) \;\leq\; X_3(n_1 - s_1) \tag{14}$$

$$W_2(n_1 - s_1) \;\geq\; W_3(n_1 - s_1). \tag{15}$$

2. *Furthermore, if $L_{2,1} \leq L_{3,1}$, then in any event, reducing s_2 to s_3 results in smaller $W_2(n_1 - s_1)$ and $W_{2,1}(n_1 - s_1)$, and larger $X_2(n_1 - s_1)$, while all other $X's$ and $W's$ remain the same. That is, reducing s_2 to s_3 results in better system performance in any event.*

Intuitively, if $L_{4,2} + L_2 \leq L_{4,3} + L_3$ and $L_{2,1} \leq L_{3,1}$ (i.e., the path through node 3 has a longer transit time), then node 3 should have a greater stock level than node 2. Proposition 1 confirms the intuition and provides stronger "stochastic inequalities." However, if $L_{4,2} + L_2 + L_{2,1} \leq L_{4,3} + L_3 + L_{3,1}$ but $L_{2,1} > L_{3,1}$, part (2) of the proposition does not necessarily hold because $L_1(n_1 - s_1)$ can increase as s_2 decreases. Thus, it is no longer a stock balancing problem among paths but a stock positioning problem for the entire system.

In some real world applications, $L_{2,1}$ can equal $L_{3,1}$ because subassemblies are in the same location as the final assembly. It is worth pointing out that Proposition 1 is based on a sample-path analysis and thus applies to negative base-stock levels and stochastic sequential transit times. Proposition 1 can be extended to acyclic supply chains with multiple subassemblies. For brevity, we omit the details.

Shared vs. Dedicated Stocking Policy.

Consider an acyclic supply chain under Assumption 1. Let *equally dedicated stocking policy* be the special case of dedicated stocking policy with equal base-stock levels for all downstream paths at each node. Let $\mathbf{s}^s$, $\mathbf{s}^d$ and $\mathbf{s}^e$ be the base-stock level vector of the shared stocking policy, dedicated stocking policy and equally dedicated stocking policy, respectively. In particular, $\mathbf{s}^s = (m_1 s_1, m_2 s_2, ..., m_N s_N)$, $\mathbf{s}^d = ([s_{1,1}, ..., s_{1,m_1}], [s_{2,1}, ..., s_{2,m_2}], ..., [s_{N,1}, ..., s_{N,m_N}])$, and $\mathbf{s}^e = ([s_1, ..., s_1], [s_2, ..., s_2], ..., [s_N, ..., s_N])$ where the k-th element of $\mathbf{s}^e$, $[s_k, ..., s_k]$, is an identical vector of m_k dimension. By definition, the feasible space for $\mathbf{s}^e$, $\mathcal{D}^e$, is a subspace of the feasible space for $\mathbf{s}^d$, $\mathcal{D}^d$. Let $U(\cdot)$ be a performance metric of the system as a function of the base-stock level vector. Here, $U(\cdot)$ is generally defined; it can be total on-hand inventory cost or total inventory investment. Without loss of generality, we consider the total inventory cost. Obviously, the dedicated stocking policy always outperforms the equally dedicated stocking policy, that is,

$$\min_{\mathbf{s}^e \in \mathcal{D}^e} U(\mathbf{s}^e) \geq \min_{\mathbf{s}^d \in \mathcal{D}^d} U(\mathbf{s}^d). \tag{16}$$

We are now ready to prove the main theorem of this paper.

Theorem 1 *Consider any acyclic supply chain under Assumption 1. Given everything else being equal, the best dedicated stocking policy always outperforms the best shared stocking policy for the performance metric $U(\cdot)$.*

Proof. For an acyclic supply chain under the shared stocking policy $\mathbf{s}^s = (m_1 s_1, ..., m_N s_N)$, we first group nodes by the number of their downstream paths, m_k. Second, we present an algorithm to split the supply chain into one under the equally dedicated stocking policy following the increasing order of m_k. Thirdly, we prove that in each step of the algorithm, the network performs equally on every sample path before and after splitting. Finally, we show that the resulting supply chain under the equally dedicated stocking policy could not outperform the one under the best dedicated stocking policy.

For the ease of exposition, we first index the final outlet by 1 and define node sets

$$\mathcal{N}_i = \{k \in \mathcal{N} : m_k = i\}, i = 1, 2, ...$$

Second, we present the following "splitting" algorithm:

Step 1. Set $i = 2$;

Step 2. If $\mathcal{N}_i = \emptyset$, skip to Step 3. Otherwise, split inventory at each node in $\mathcal{N}_i$ as follows: Select the most downstream node in $\mathcal{N}_i$ not yet split and denote it by k (if the non-split nodes in $\mathcal{N}_i$ do not have a supply-demand relationship, select any of them). For the selected node k, split its inventory level $m_k s_k$ into m_k stock piles with equal base-stock levels s_k, where each stock pile is dedicated to satisfying demand from a unique downstream path;

Step 3. If $i > \max_{k\in\mathcal{N}}\{m_k\}$, stop. Otherwise, set $i = i+1$ and go back to Step 2.

The algorithm starts with $i = 2$ because we do not need to split the inventory at any node $k \in \mathcal{N}_1$.

Thirdly we prove that in each iteration of the above splitting algorithm, the network performs equally on every sample path before and after splitting. In particular, we prove that in Step 2 above, $X_k(n_k)$ and $W_k(n_k)$ remain the same in any event before and after splitting. To this end, for each sample path, we make the following observations:

1. The demand arrival process at node k from each dedicated stock pile at node $i \in \mathcal{C}(k)$ remains the same before and after splitting. In other words, for any demand n_1 at node 1 (the final outlet), the corresponding demand index at node k, n_k, does not change before and after splitting. This is true because at this iteration, no change is made to $\mathcal{C}(k)$ and their demand streams.

2. The index for the corresponding order placed by node k that satisfies demand n_k remains unchanged before and after splitting. This is true because before splitting, the index equals $n_k - s_k$ (at node k, the stock keeping unit is in m_k), while after splitting, the index also equals $n_k - s_k$ (equally dedicated stocking policy with stock level s_k).

3. For each node $j \in \mathcal{S}(k)$, given a demand index n_j, the corresponding $X_j(n_j)$ and $W_j(n_j)$ remain the same before and after splitting because no change is made to them at this iteration.

By the first two observations above, at node k, $T(s_k, n_k)$ for each dedicated pile equals the one before splitting, which is the sum of interarrival times from the $(n_k - s_k)$-th demand to the n_k-th demand at node 1. By the second and the third observations above, we have that $X_k(n_k)$, $W_k(n_k)$, $\tilde{L}_k(n_k)$ and $W_{j,k}(n_k)$ remain unchanged at node k after splitting (given in Eqs. (1)-(4)).

Applying the same logic to every iteration of the algorithm, we conclude that the shared stocking policy can be replaced by an equally dedicated stocking policy that performs equally in any event. Finally, it follows by Eq. (16) that the resulting equally dedicated stocking policy could not outperform the best dedicated stocking policy. The proof is now completed. ■

The intuition behind this theorem is that in acyclic supply chains, for a node serving multiple downstream paths leading to the same final outlet, the outlet drives the order processes along

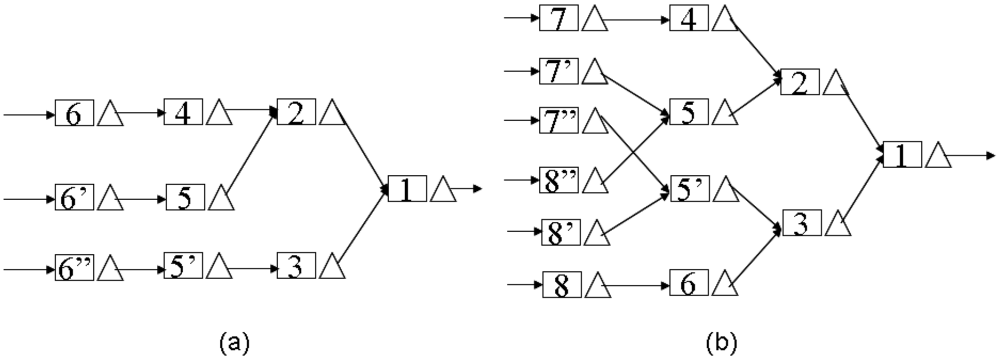

Figure 4: The corresponding tree network for examples in Figure 2.

all paths. Under certain assumptions of demand processes, inventory policies, transit times and network topology, the order processes of these paths are synchronized in both timing and quantity. For such a node, pooling inventory does not achieve the *risk-pooling effect*; but splitting the inventory into dedicated stock piles and allowing the stock-levels to be different does achieve the *stock-positioning effect*. In these cases, the dedicated stocking policy outperforms the shared stocking policy.

By this intuition, we can easily see that Theorem 1 extends to any BOM matrix if we redefine the flow unit along each path appropriately. Theorem 1 also holds for negative base-stock levels, because for any node serving multiple paths leading to the final outlet, the demand from all paths is completely synchronized regardless of the sign of the base-stock levels. Finally, Theorem 1 holds for exogenous and sequential stochastic transit times. To see this, we consider a node with multiple dedicated stock piles. An order placed by this node is the sum of orders triggered at all its stock piles. Because the transit time applies to each order of the node as a whole, the transit time of each order placed by this node does not change before and after "splitting".

In §3, we present numerical examples in which the dedicated stocking policy outperforms the shared stocking policy.

Acyclic Supply Chain vs. Tree. We are now ready to show that under certain conditions, an acyclic supply chain can be decomposed into a tree network without increasing inventory cost (see Appendix for a proof).

Theorem 2 *Under Assumption 1, an acyclic supply chain can always be decomposed into a tree network which performs at least as well as the original system.*

Theorem 2 implies that under certain conditions, acyclic supply chains can be decomposed into tree networks without loss of performance. Tree networks are much easier to solve than acyclic networks, see Simchi-Levi and Zhao (2005) for an algorithm based on dynamic programming and two-moment approximations. Theorem 2 is exemplified by Figure 4 which shows the resulting tree network for each acyclic supply chain in Figure 2. Because the optimal dedicated stocking policy tends to be different for paths heterogeneous in transit time and cost, the performance improvement of the resulting tree networks tends to increase in these cases. Lastly, due to constant transit times, we can apply Rosling (1989) to the resulting tree networks and improve their performance over installation base-stock policies.

It is easily seen that Theorem 2 also holds for any BOM matrix and negative base-stock levels. However, when the transit times are stochastic sequential and exogenous, Theorem 2 does not hold. This is because the transit times for consecutive orders placed by a node are dependent, and therefore, the transit times for orders generated by different stock piles at the same node are also dependent. Hence, the processing facility cannot be split into multiple facilities that operate independently.

3 A Numerical Study

In this section, we provide examples to show that the dedicated stocking policy can perform strictly better than the shared stocking policy. Consider the acyclic supply chain in Figure 1 under Assumption 1. Without loss of generality, we assume $L_{i,j} = 0$ for all arcs. Let the demand arrival rate $\lambda = 1$, $L_1 = 1$ and $h_4 = 1$ where h_i is the installation inventory holding cost at stage $i = 1, 2, 3, 4$. Node 1 is required to provide an immediate fill rate of 95%. In the corresponding tree structure supply chain, the original node 4 is split into two nodes with the new node 4 supplying node 2 and an additional node 4' supplying node 3, while everything else remains unchanged. We use simulation to search for the optimal solution of each network where the simulated fill rate at node 1 must be at least as high as the target fill rate but not exceeding 95.6%, and the difference of the simulated fill rates between the two networks is less than 0.5%.

To build intuition, we first consider the case of $L_2 = L_3$, i.e., the paths are symmetric in transit time. In this case, the shared stocking policy performs as well as the dedicated stocking policy because the optimal dedicated stock levels at nodes 4 and 4' must be identical. When paths are asymmetric in transit time, then in the corresponding tree network, the optimal dedicated stock levels at nodes 4 and 4' may be different, and the dedicated stocking policy can outperform the shared stocking policy. We consider 3 cases.

Case 1: $h_2 = h_3$ but $L_2 > L_3$. We set $(h_1, h_2, h_3, h_4) = (4, 1.5, 1.5, 1)$ and $L_3 = 1$. In examples with combinations of $L_2 \in \{5, 10\}$ and $L_4 \in \{5, 10\}$, the savings range from 0 to 3.58% with the highest saving corresponding to the case with $L_2 = 10$ and $L_4 = 5$. The optimal base-stock levels of the shared (dedicated) stocking policy are $(s_1, s_2, s_3, 2s_4) = (5, 14, 2, 12)$ $((s_1, s_2, s_3, s_4, s_{4'}) = (5, 16, 2, 2, 6))$.

Case 2: $h_2 < h_3$ but $L_2 > L_3$. We set $(h_1, h_2, h_3, h_4) = (6, 1.5, 3, 1)$ and $L_3 = 1, L_4 = 10$. In examples with $L_2 \in \{2, 5, 10\}$, the savings range from 2.87% to 4.45% with the highest saving corresponding to the case with $L_2 = 10$. The optimal base-stock levels of the shared (dedicated) stocking policy are $(s_1, s_2, s_3, 2s_4) = (5, 15, 2, 24)$ $((s_1, s_2, s_3, s_4, s_{4'}) = (5, 14, 1, 12, 14))$.

Case 3: $h_2 > h_3$ but $L_2 > L_3$. We set $(h_1, h_2, h_3, h_4) = (6, 3, 1.5, 1)$ and $L_2 = 10$. In examples with $(L_3, L_4) = (5, 10), (2, 10), (5, 5)$ and $(2, 5)$, the savings range from 0.88% to 3.12% with the highest saving corresponding to the case with $(L_3, L_4) = (5, 5)$. The optimal base-stock levels of the shared (dedicated) stocking policy are $(s_1, s_2, s_3, 2s_4) = (6, 12, 7, 24)$ $((s_1, s_2, s_3, s_4, s_{4'}) = (4, 14, 10, 7, 4))$.

In summary, the savings can be sizable when the transit time is out of balance across paths that lead to the final outlet. Rather than keeping identical stock levels at node 4 for all downstream paths (as in the shared stocking policy), the dedicated stocking policy can allow dedicated stock piles at node 4 to choose different stock levels and therefore optimally position inventory for each path.

4 Extensions

In this section, we discuss the robustness of Theorems 1-2 in more general acyclic supply chains.

General Demand Processes. We first replace the Poisson demand process in Assumption 1 by an arbitrary unit demand process (e.g., renewal unit demand). The recursive Eqs. (1)-(4) still hold. Because Proposition 1 and Theorems 1-2 are based on a sample-path analysis, they hold regardless of the probability law governing the demand arrival process.

We then replace the Poisson demand process in Assumption 1 by a general batch demand process. We observe that for any node serving multiple paths that lead to the final outlet, demands from all downstream paths are still completely synchronized in both timing and quantity. We further assume that such a node utilizes the "virtual allocation" rule (Graves 1996) to satisfy demand. Under this rule, the node satisfies the same proportion of demand from all downstream paths. Consider a demand unit at a node, the exact recursive equation of X and W can be obtained in the same way as Zhao (2007). Furthermore, all the system properties in §2 hold true here because demand can be split, which implies that batch demand processes can be regarded as special cases of unit demand processes where the interarrival times between consecutive demand arrivals can be zero.

Other Inventory Policies. We replace the continuous-time base-stock policy and FCFS rule in Assumption 1 by a periodic review base-stock policy and the "virtual allocation" rule. We also assume that the external demand arrives at the end of each period and the review periods are nested, i.e., whenever a node orders, all of its customer nodes order. We consider two cases: (1) The review periods are identical. In this case, the recursive equations are identical to those under a continuous-time base-stock policy but with batch demand and constant interarrival times. Thus Theorems 1-2 hold by our analysis for general demand process. (2) The review periods are not identical. We observe that for any node serving multiple paths that lead to the final outlet, demand from different downstream paths is still completely synchronized in each period. By the "virtual allocation" rule, Theorems 1-2 hold. However, we must point out that these results may not hold if one discards the assumptions of "virtual allocation" and nested review periods.

For acyclic supply chains with batch ordering policy, Theorems 1-2 do not hold in general because the orders placed by nodes with different batch sizes are not synchronized.

Multiple Final Outlets. Acyclic supply chains with multiple final outlets are much more complex than those with a single final outlet. Although some of these networks can still be decomposed into simpler ones, they require more specific conditions and are case-dependent. We consider a specific example to illustrate the idea: We identify nodes with completely synchronized demand from all downstream paths that lead to each final outlet. This idea can be applied to any acyclic supply chain to check if a reduction of network complexity can be achieved.

The example is shown in Figure 5 (a), which is inspired by a real-world problem, see Billington, et al. (2004), Figure 7 on page 67. Clearly, the timing of the order processes of nodes 3 and 4 are synchronized, i.e., nodes 3 and 4 always place orders at the same time. If assembling one unit at node 1 requests the same amount of component 3 and the same amount of component 4 as assembling one unit at node 2, then properly defining the flow units at node 3 and 4 yields completely synchronized order processes (in both timing and quantity) placed by node 3 and 4.

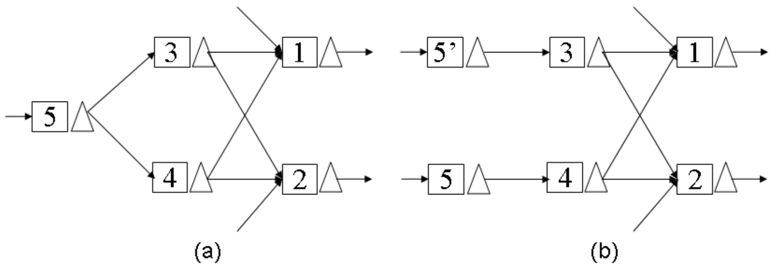

Figure 5: An example of acyclic supply chains with multiple final outlets.

Further assuming Assumption 1 holds except that there are multiple final outlets, then one can replace the acyclic network by a network with at most one directed path between each pair of nodes (Figure 5 (b)), which performs at least as well as the original system.

In conclusion, we shall point out that while an acyclic supply chain may be decomposed into a simpler network (e.g., a tree or a network with at most one-directed path between each pair of nodes) which is easier to solve and plan for, the dedicated stocking policy may increase the complexity of operations and incur extra administrative cost. Furthermore, allowing complexity to increase also opens doors to more complex and possibly better performing inventory policies. It will be interesting for future research to identify such policies.

Acknowledgment We thank the Editor-in-Chief, Professor Awi Federgruen, Associate Editor, and two anonymous referees for their constructive comments and thoughtful suggestions. This research is supported in part by a Faculty Research Grant from Rutgers Business School–Newark and New Brunswick.

Appendix

To prove Proposition 1, we need the following definition.

Definition 1 *The sequential lead-time rule for a one-product inventory system with base-stock level s states that the orders are delivered in the same sequence as they are placed. In other words, if two orders, $n' \leq n''$, are placed at $t' \leq t''$, respectively, then it is always true that*

$$\max\{t' - T(s,n') + \tilde{L}(n'), t'\} \leq \max\{t'' - T(s,n'') + \tilde{L}(n''), t''\}. \tag{17}$$

Note that in Eq. (17), $\max\{t - T(s,n) + \tilde{L}(n), t\}$ is the time at which the n-th order is fulfilled, which is an increasing function of the order arrival time t.

Proof for Proposition 1. Consider a sample path for the demand arrival process. For the ease of exposition, we assume that the orders $n_2 = n_1 - s_1$ and $n_3 = n_1 - s_1$ (to satisfy demand n_1) are placed at time t, and orders $n_4 = n_1 - s_1 - s_2$ and $n_4' = n_1 - s_1 - s_3$ (to satisfy n_2 and n_3) are placed at time t_2 and t_3, respectively, where $t_2 = t - T(s_2, n_1 - s_1)$ and $t_3 = t - T(s_3, n_1 - s_1)$.

To prove part (1), we first note $t_2 \leq t_3$ because $T(s_2, n_1 - s_1) \geq T(s_3, n_1 - s_1)$ by the definition of $T(s_k, n_1 - s_1)$ and the condition $s_2 \geq s_3$. Second, Assumption 1 implies $\tilde{L}_4(n_4 - s_4) = \tilde{L}_4(n_4' - s_4) = L_4$. Then at node 4, the demand n_2 from node 2 is fulfilled at $\max\{t_2 - T(s_4, n_4) + L_4, t_2\}$, while the demand n_3 from node 3 is fulfilled at $\max\{t_3 - T(s_4, n_4') + L_4, t_3\}$. Since $t_2 \leq t_3$, by Eq. (17), we have

$$\max\{t_2 - T(s_4, n_4) + L_4, t_2\} \leq \max\{t_3 - T(s_4, n_4') + L_4, t_3\}. \tag{18}$$

Then at node 2, the time at which the inventory is ready to fulfill order n_2 is

$$R_2 = \max\{t_2 - T(s_4, n_4) + L_4, t_2\} + L_{4,2} + L_2, \tag{19}$$

While at node 3, the time at which the inventory is ready to fulfill order n_3 is

$$R_3 = \max\{t_3 - T(s_4, n_4') + L_4, t_3\} + L_{4,3} + L_3.$$

Combining Eq. (18) and the condition $L_{4,2} + L_2 \leq L_{4,3} + L_3$ yields $R_2 \leq R_3$. Note that $X_k(n_1 - s_1) = [R_k - t]^+$ and $W_k(n_1 - s_1) = [t - R_k]^+$ for $k = 2, 3$. The proof for part (1) is completed by the fact that if $a \leq b$ then $[a]^+ \leq [b]^+$ and $[a]^- \geq [b]^-$.

To prove part (2), we consider s_2 and s_3 such that $s_2 > s_3$. Let $s_2' = s_2 - 1 \geq s_3$. Substituting $t_2 = t - T(s_2, n_1 - s_1)$ into Eq. (19) leads to

$$\begin{aligned} R_2 &= \max\{t - T(s_2, n_1 - s_1) - T(s_4, n_4) + L_4, t - T(s_2, n_1 - s_1)\} + L_{4,2} + L_2 \\ &= t + \max\{-T(s_2 + s_4, n_1 - s_1) + L_4, -T(s_2, n_1 - s_1)\} + L_{4,2} + L_2 \\ &= t - \min\{T(s_2 + s_4, n_1 - s_1) - L_4, T(s_2, n_1 - s_1)\} + L_{4,2} + L_2, \end{aligned} \tag{20}$$

where $T(s_2, n_1 - s_1) - T(s_4, n_4) = T(s_2 + s_4, n_1 - s_1)$ by definition. For the base-stock level s_2', let

$$R_2' = t - \min\{T(s_2' + s_4, n_1 - s_1) - L_4, T(s_2', n_1 - s_1)\} + L_{4,2} + L_2. \tag{21}$$

Then, we have $R_2' > R_2$ because $T(s_2' + s_4, n_1 - s_1) < T(s_2 + s_4, n_1 - s_1)$ and $T(s_2', n_1 - s_1) <$

$T(s_2, n_1 - s_1)$. Consequently,

$$X_2^{'}(n_1 - s_1) = [R_2^{'} - t]^+ \geq [R_2 - t]^+ = X_2(n_1 - s_1), \tag{22}$$

and

$$W_2^{'}(n_1 - s_1) = [t - R_2^{'}]^+ \leq [t - R_2]^+ = W_2(n_1 - s_1).$$

By Eq. (8), we have

$$\begin{aligned} W_{2,1}(n_1 - s_1) &= \max\{0, X_3(n_1 - s_1) + L_{3,1} - [X_2(n_1 - s_1) + L_{2,1}]\} \\ &= X_3(n_1 - s_1) + L_{3,1} - [X_2(n_1 - s_1) + L_{2,1}], \end{aligned} \tag{23}$$

and

$$W_{2,1}^{'}(n_1 - s_1) = X_3(n_1 - s_1) + L_{3,1} - [X_2^{'}(n_1 - s_1) + L_{2,1}], \tag{24}$$

where the second equality in Eq. (23) holds because of the condition $L_{3,1} > L_{2,1}$ and the result obtained in part (1), $X_3(n_1 - s_1) > X_2(n_1 - s_1)$. The similar argument applies to obtain Eq. (24). Combining Eqs. (22)-(24) yields

$$W_{2,1}^{'}(n_1 - s_1) < W_{2,1}(n_1 - s_1).$$

Finally, it follows by Eq. (7) that $\tilde{L}_1(n_1 - s_1)$ remains unchanged, thereby $X_1(n_1)$ and $W_1(n_1)$ remain unchanged by Eqs. (5)-(6). This completes the proof for part (2). ∎

Proof for Theorem 2. Consider a node $k \in \mathcal{N}$ that serves m_k paths leading to the final outlet. By the proof of Theorem 1, we can use the equally dedicated stocking policy for this node without changing performance. Thus, the inventory of such a node is split into m_k stock piles each dedicated for a different path. Furthermore, since the transit times are constant, we can further split the processing facility at such a node into multiple processing facilities each dedicated to a different stock pile without changing the transit times. Doing this sequentially for all such nodes from downstream to upstream results in a network with at most one directed path between each pair of nodes.

We now show that the resulting network has a tree structure by proving that each node in the resulting network has at most one immediate customer node, and thus it is a multi-level pure assembly system. We use contradiction. If there is a node with more than one immediate customer nodes, then the resulting network must be acyclic because there is a unique final outlet. Clearly, this is contradictory to the fact that the resulting network has almost one directed path between each pair of nodes. The proof is now completed. ∎

References

Axsater, S. (2006). *Inventory Control*, 2nd edition. Springer Science+Business Media, LLC.

Billington, C., G. Callioni, B. Crane, J. D. Ruark, J. Unruh Rapp, T. White & S. P. Willems (2004). Accelerating the profitability of Hewlett-Packard's supply chains. *Interfaces*, **34**, 59-72.

de Kok, T. G. & J. C. Fransoo (2003). Planning supply chain operations: definition and comparison of planning concepts. In A.G. de Kok, S.C. Graves (eds.). *Handbooks in Operations Research and Management Science*, Vol. 11: Supply Chain Management. Elsevier, Amsterdam, The Netherlands.

Graves, S. (1996). A Multi-echelon inventory model with fixed replenishment intervals. *Management Science*, **42**, 1-18.

Graves, S. C. & S. P. Willems (2000). Optimizing strategic safety stock placement in supply chains. *Manufacturing and Service Operations Management.* **2**, 68-83.

Graves, S. C. & S. P. Willems (2003). Supply chain design: safety stock placement and supply chain configuration. A. G. de Kok and S. C. Graves, eds, *Handbooks in Operations Research and Management Science Vol. 11, Supply Chain Management: Design, Coordination and Operation.* North-Holland Publishing Company, Amsterdam, The Netherlands.

Humair, S. & S. P. Willems (2007). Optimizing strategic safety stock placement in general acyclic networks. Working Paper, Boston University. Boston, MA.

Lesnaia, E. (2004). Optimizing safety stock placement in general network supply chains. Ph.D. Thesis, MIT Operations Research Center, Cambridge, MA.

Magnanti, T. L., Z. J. M. Shen, J. Shu, D. Simchi-Levi & C. P. Teo (2006). Inventory placement in acyclic supply chain networks. *Operations Research Letters*, **34**, 228-238.

Minner, S. (2000). *Strategic safety stocks in supply chains.* Lecture notes in economics and mathematical systems; 490. Springer-Verlag, Berlin.

Rosling, K. (1989). Optimal inventory policies for assembly systems under random demand. *Operations Research*, **37**, 565-579.

Simchi-Levi, D. & Y. Zhao (2005). Safety stock positioning in supply chains with stochastic leadtimes. *Manufacturing and Service Operations Management*, **7**, 295-318.

Simchi-Levi, D. & Y. Zhao (2007). Three Generic Methods for Evaluating Stochastic Multi-Echelon Inventory Systems. Technical Report, Rutgers Business School.

Song, J. S. & P. Zipkin (2003). Supply chain operations: Assemble-to-order systems. In A.G. de Kok, S.C. Graves (eds.). *Handbooks in Operations Research and Management Science*, Vol. 11: Supply Chain Management. Elsevier, Amsterdam, The Netherlands.

Svoronos, A. & P. Zipkin (1991). Evaluation of one-for-one replenishment policies for multiechelon inventory systems. *Management Science*, **37**, 68-83.

Zhao, Y. (2007). Evaluation and optimization of installation base-stock policies in supply chains with compound Poisson processes. *Operations Research*, **56**, 437-452.

Zipkin, P (2000). *Foundations of Inventory Management.* McGraw Hill, Boston.