

Strategic Inventory Placement in Multi-Echelon Supply Chains: Three Essays

by

Tor Schoenmeyr

B.S.,M.S., Engineering Physics
Chalmers University of Technology, 2001

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Management
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

© Massachusetts Institute of Technology 2008. All rights reserved.

Author.....

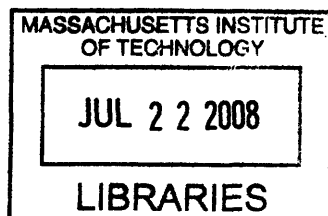
Sloan School of Management
May 2008

Certified by.....

Stephen C. Graves
Abraham J. Siegel Professor of Management Science
Thesis Supervisor

Accepted by.....

Birger Wernerfelt
Professor of Management Science
Chair, Doctoral Program



ARCHIVES

Strategic Inventory Placement in Multi-Echelon Supply Chains: Three Essays

By Tor Schoenmeyr
Submitted to the Sloan School of Management
June, 2008, in partial fulfillment of the
requirement for the degree of
Doctor of Philosophy in Management

Abstract

A central question in supply chain management is how to coordinate activities and inventories over a large number of stages and locations, while providing a high level of service to end customers. One theoretically and practically important methodology for addressing this problem is the guaranteed service (GS) framework, in which the stages of the supply chain operate according to base stock policies, and provide guaranteed service to one another. Demand is assumed to be bounded. Previous work on GS models has established very effective algorithms for finding optimal safety stock placement.

In the first essay of the thesis, we show how these methods can be generalized to handle problems with *capacity constraints*. Furthermore, we investigate orders that are *censored* (reduced so as to prevent deliveries greater than what can be processed). We find safety stock reductions, sometimes even below what was needed in the no-constraint situation.

In the second essay, we investigate a situation in which different parts of the supply chain are controlled by different parties, each of which selfishly applies its own GS optimization. We find that provided that the parties can agree on the right service time between them, it will be in their own interests to maintain the globally optimal solution (i.e., the system is incentive compatible). This suggests that the GS framework is better suited for coordination, than are other frameworks analyzed in the coordination literature.

Finally, in the third essay, we apply the GS framework to a setting where orders are driven by forecasts and schedules, rather than by past demand as in previous GS work. We show precisely how the demand bound can be replaced by a bound on forecast errors, and that existing optimization methods can be used. In a case study, we obtained data from the supply chain of an electronic test system, as well as characterized the forecasting process. We found that incorporating the forecast process led to 25% reduction of safety stocks.

Thesis Supervisor: Stephen C. Graves

Title: Abraham J. Siegel Professor of Management Science

Acknowledgements

First and foremost, I would like to thank Professor Stephen Graves. Steve always let me explore that which I found exciting, yet provided the guidance necessary for turning ideas into research. Uncountable times, Steve pushed me forward when I thought it was impossible, and made me take a second look when I thought there was nothing more to see. Thanks, Steve.

I am also indebted to Professors Gabriel Bitran and David Gamarnik, not only for their valuable input on this thesis, but also for many stimulating and developing discussions on everything from tricky calculations to making the right career choice.

In addition, I would like to thank Professor Jeremie Gallien, who introduced me to the PhD program, and gave me crucial guidance during my initial time at MIT.

A special thanks to Ms. Anna Piccolo for being a friendly and supportive face in the office, and for never hesitating to help out in small and big ways.

I am also grateful to Dennis Mauriello, Nirav Shah, Thomas LeBlanc, Jim Wood, and many others at Teradyne, Inc, for sharing their thoughts and data with me, thus providing motivating questions for this research.

This thesis would not have been possible without funding. I thank the Singapore MIT Alliance (SMA) and Dr. Marcus Wallenberg's Foundation for Education in International Industrial Management for generous support.

Finally, I thank my friends and family, and my fiancé Hong. Thanks for being there for me. We are spread out all over the planet, but you are always close to me.

Content

Strategic Inventory Placement in Multi-Echelon Supply Chains: Three Essays	1
Abstract	3
Acknowledgements	4
List of Figures	8
List of Tables	9
I. Introduction	11
1. The guaranteed service framework and an overview of the thesis	11
2. About the three essays	12
3. A common theme: challenging “optimality” in the GS framework	15
4. Limitations and future directions	16
II. Strategic safety stocks in supply chains with capacity constraints	19
5. Introduction	19
6. Single stage model	22
7. Multiple stage model with base-stock ordering	30
8. Multiple stage model with censored order policy	33
9. Numerical experiments	38
10. Conclusions and discussion	44
Appendix 1: The average backlog	47
Appendix 2: Proofs and derivations	50
III. Coordination of multi-echelon supply chains using the guaranteed service framework	57
11. Introduction	57
12. The optimization problem and its separation	60
13. A proposed structure for contracts and bargaining	66
14. Identifying relevant holding costs	70
15. Numerical examples	75
16. Conclusion	78
Appendix: the concavity of $\mathbf{P}_1(S_B)$	80
IV. Strategic safety stocks in supply chains with evolving forecasts	82

17.	Introduction	82
18.	The forecast model	86
19.	Supply chain model and ordering policy	90
20.	Optimization	95
21.	Case study	102
22.	Numerical examples	109
23.	Conclusions and future directions	112
V.	Addendum	125
24.	Forecast-based ordering with multiple end products	125
25.	Base-stock ordering when demand is not i.i.d.	130
	Bibliography	134

List of Figures

Figure 1: Overview of a single-stage system.....	24
Figure 2: With capacity constraints, for small τ a higher base stock level is necessary, but for $\tau \geq 4$ the capacity constraint does not matter. Plotted parameters are $\mu = \sigma = 4$, $c = 6$ and $z = 2$	29
Figure 3: The cost for the lower part, upper part, and total supply chain, as a function of S_B . In this example, the worst S_B resulted in total costs that were 22.1% higher than the best value.	64
Figure 4: Terms to be specified in a contract.....	67
Figure 5: A contract for situations when downstream holding costs are affected by a non-value added markup m . We note that during steady-state conditions the separation of payment is of no consequence.	74
Figure 6: Schematic view of supply chain for the studied product.....	103
Figure 7: Forecast quality (correlation with what was actually produced) as a function of time into the future, for an electronic test system.....	104
Figure 8: Normalized bound functions for systems with forecasts and with base stock policies.	106
Figure 9: Total safety stock inventory costs for all the nodes with less than a certain lead time	107

List of Tables

Table 1: Example of orders, censored orders, and backlog	34
Table 2: Summary of node properties in supply chain with capacity constraint(s) and censored base stock policy.....	37
Table 3: Alternative structures for supply chain lead-time and cost accumulation.....	39
Table 4: Experiments for base-stock policy, for constant lead time and rate of cost increase	40
Table 5: Total costs for base-stock policy with capacity constraint $c = 45$. In each case, the cost of the constrained problem is given as a percentage of the corresponding unconstrained problem.....	40
Table 6: The cost and optimal solution for scenario with constant lead time and constant cost accumulation.....	41
Table 7: Experiments for censorship, constant lead time and cost accumulation scenarios	42
Table 8: \overline{BL}_k for $\mu = 40, \sigma = 20$, and various censorship values c	48
Table 9: Alternative structures for supply chain lead-time and cost accumulation.....	75
Table 10: System performance when S_B is chosen suboptimally and parties optimize their sections separately. For the different scenarios on the left, the average, and worst cases are stated relative to the optimal one.	76
Table 11: Actual system costs when lower two stages use a superfluous markup in their holding cost.....	77
Table 12: Alternative structures for supply chain lead-time and cost accumulation.....	109
Table 13: Total costs for various supply chains and forecast horizons	110
Table 14: Structure of optimal solution; 1 represents inventory at a node.	111

I. Introduction

1. The guaranteed service framework and thesis overview

A central question in supply chain management is how to coordinate activities and inventories over a large number of stages and locations, while providing a high level of service to end customers. This question is of great practical importance, and is also well suited for theoretical analysis. Accordingly, it has attracted a great deal of attention from both practitioners and scholars.

A multi-echelon supply chain is so complex that one cannot hope to incorporate every conceivable cost, benefit, action and activity into a single, all-inclusive optimization problem. Rather, researchers and practitioners focus on more limited problems within different frameworks - sets of “ground rules” and shared assumptions. During the past decades, numerous investigations within several different frameworks have thus resulted in a rich literature detailing theories, models, and empirical experiences of various supply chain problems and solutions. Nevertheless, many questions remain unanswered, and others are worth revisiting for a deeper examination. Indeed, during field work I found many examples of important, real-world supply chain challenges which have not been adequately solved in the academic literature.

I chose to investigate a subset of these problems taking the guaranteed service (GS) framework as a starting point. The key original assumptions of this framework are that the different stages of the supply chain operate according to local *base-stock policies*. This means that each stage is somehow able to observe end-customer demand in each period and then places a replenishment order equal to the total end-customer demand in each period. Moreover, it is assumed that *demand is bounded*, and that the stages provide *guaranteed service* (for orders within the demand bound). By changing the *service time* between the stages, one can effectively decide the size and location(s) of safety stock(s) in the supply chain. Specifically, one seeks the lowest cost safety stock placement(s), subject to the guaranteed service constraint. I write much more about these assumptions and how they can be justified in the essays themselves.

Previous research on guaranteed service models has established effective optimization methods for many common supply chain topologies. As a consequence, in this thesis I do not explore improvements to these optimization algorithms. Rather, the objective is to improve the *generality* of existing methods, by showing how we can apply the known optimization algorithms to supply chain problems that are, in different ways, more general than those studied to date.

However, in doing so, we will also discover methods that increase supply chain *performance*, even when applied to the original supply chains studied before. By increasing performance, we mean reducing holding costs, without compromising service levels. This performance increase does not come from better optimization algorithms; as noted, this thesis does not investigate optimization algorithms. Rather, the benefits come from expanding the space of investigated solutions. Specifically, we investigate ordering policies that in various ways differ from the original base stock policies. These findings challenge the notion of “optimality” in the guaranteed service framework; we will discuss this topic in greater detail below. First, we will outline the content of the three essays which comprise the thesis.

2. About the three essays

In the first essay, “Strategic safety stocks in supply chains with capacity constraints,” we consider safety stock placement in contexts where there are significant *capacity constraints*. In many real-world situations there are limits to how much goods can be processed, transported, or stored, in a given time frame, but this has not been previously studied in the GS literature. If one operates a supply chain while disregarding such constraints, one may encounter unexpected stock-outs, as deliveries are delayed at system bottlenecks. Not surprisingly, we find that one may sometimes need to add extra inventory downstream of such bottlenecks. However, the extra inventory is not a simple additive term, but rather follows a non-linear pattern. For capacity constraints which are only slightly greater than average demand, the required extra inventory can be significant, indeed, arbitrarily large. On the other hand, we find that if a stage would need significant safety stocks anyway to protect against demand volatility, then a capacity constraint

might *not* necessitate any *additional* safety stocks. By characterizing the necessary safety stock levels as functions of service time, we effectively generalize existing optimization methods so that one can determine the optimal safety stock placement in supply chains with one or many capacity constraint(s).

For the initial analysis we assume that each stage continues to operate with a local base-stock policy. We then relax this assumption and consider a multi-stage system in which stages *cancel* their orders, based on their capacity limits. That is, rather than placing an order larger than the maximum processing capacity, excessive order quantities are delayed so that they will not arrive until sufficient capacity is available. Again we analytically characterize the necessary base stock levels, and develop an extension to the existing dynamic programming algorithms to find the optimal base stock levels and safety stocks. Censored orders are less variable than uncensored orders, so there is less need for safety stocks upstream of the constrained stage. However, that stage will often need extra safety stock (relative to an infinite capacity, no-censorship situation), in order to fulfill the guaranteed service constraint. When both effects are accounted for, we find that the total holding costs for the censored order policy can sometimes be less than that for the corresponding base-stock system without capacity constraints. For this reason, one may want to cancel orders even in systems lacking physical capacity constraints. Numerical simulations suggest that the best way to do this is by canceling far downstream in the supply chain, using a censorship value only slightly larger than average demand.

The second essay in this thesis is “Coordination of multi-echelon supply chains using the guaranteed service framework”. Here, we investigate how the guaranteed service framework can be used when different parts of the supply chain are controlled by different parties. Past work on guaranteed service models has been made from the perspective of a single decision-maker, who controls the entire supply chain. In reality, different parts of the supply chain are often controlled by different companies (or business units within the same company). These different entities might have conflicting and competing interests and objectives. When multiple parties selfishly optimize their own parts of the supply chain, will the entire system function well, or will the parties’ myopic behavior induce inefficiencies? Considerable research efforts have been put into

related questions in the past. However, only limited attention has been paid to coordination of multi-echelon systems, and none at all in the context of guaranteed service models. We find that the GS framework is particularly well suited to facilitate coordination. Specifically, provided that the different parties can agree to use the “right” service time between them, their individual objectives will be aligned with the objective of the entire system. The “right” service time is the service time which coincides with the one obtained from a global optimization. This finding leads us to propose a simple contract structure, in which the supplier agrees to provide guaranteed service for all demand realizations within a specific demand bound, with a service time equal to the globally optimal one. With this contract in place, there is no need for the parties to monitor each others’ internal activities, and the system will behave efficiently. That is, there is *incentive compatibility*. In the essay itself, we discuss several other benefits, and relate the results to Nash’s bargaining model.

Moreover, the simple structure of this agreement stands in sharp contrast to other approaches to coordination in multi-echelon supply chains. For example, in order to coordinate Clark and Scarf’s model, one needs a more complex contract structure that specifies transfer pricing, consignment, shortage reimbursement, and an additional backlog penalty.

The last essay, “Strategic safety stocks in supply chains with evolving forecasts,” differs significantly from the other essays (and from past work on guaranteed service models), in terms of the assumptions on how the stages place their orders. Rather than assuming demand-driven, base-stock orders, we consider a system where orders are placed in response to schedules and forecasts. Such logic is widely used in industry, often facilitated by material resource planning (MRP) software systems. We show that the safety stock problem in this setting is mathematically very similar to the problem from the base stock setting. Specifically, we replace the assumption of bounded demand with an assumption of *bounded forecast errors*. That is, safety stocks are not set up to guarantee service for any demand within a certain bounds. Rather, we specify the safety stocks so as to guarantee service as long as forecast errors stay within certain bounds. An alternative interpretation is that they allow for a maximum quantity of schedule changes. We specify exactly how the bounds can be determined, given a history of forecast data,

and/or a mathematical demand model. This method applies even if demand is non-stationary. Indeed, we highlight how the model is compatible with general state space models of demand (e.g., ARIMA), as well as the well-known forecast evolution (martingale) model. We also discuss how these approaches relate to each other.

Once the forecast error bounds have been determined, the safety stock optimization problem is mathematically very similar to the well-studied problem in the base stock setting. Therefore, we can use the effective optimization algorithms developed by other authors.

In a field study I collected data on the forecasting process and supply chain characteristics of a supply chain at Teradyne, Inc, a manufacturer of electronic test equipment located in North Reading, Massachusetts. I found that incorporating the forecasting process into the safety stock analysis resulted in significant (about 25%) safety stock reductions relative to the optimal base stock solution. Inventory could be reduced, especially downstream in the supply chain, because demand for products in the late stages of completion could be forecasted with good accuracy. The supply chain we analyzed had some 3,866 part-locations, yet the optimization took only about a minute on a mobile computer. We are not aware of any comparable results using other approaches to safety stock optimization in forecast-driven systems.

3. A common theme: challenging “optimality” in the GS framework

In this thesis, the notion of optimality in GS models is repeatedly challenged. We recall that the GS framework, as originally presented, seeks to minimize average inventory costs, *assuming* that the stages operate according to *base stock policies* and provide *guaranteed service* to each other, for demand realizations within the *demand bound*. The assumptions of base stock policies, guaranteed service, and demand bounds, are caveats to the optimality achieved within the GS framework.

A base stock policy relies on observed demand and does not account for any information from a demand forecast. Not surprisingly, one can do better if one modifies the ordering policy in response to useful information about future demand. As noted

above, we explore this in detail in “Strategic safety stocks in supply chains with evolving forecasts”. We also demonstrate the benefits in a case study with real data.

But even without forecast information, we find that (local) base stock policies are not generally optimal. Indeed, in a well-known paper, Clark and Scarf show that echelon-based ordering policies are optimal. These authors look at different optimality criteria; rather than having a guaranteed service constraint, they add a back-order cost component into the objective function. Nevertheless, in our essay “Strategic safety stocks in supply chains with capacity constraints”, we give realistic examples of situations when employing order censorship can lead to improvements. These improvements are with respect to the same performance criteria, and are subject to the same service guarantee constraints, as in the standard GS model. The reason why this is possible is that we have expanded the space of possible solutions; no longer do we limit our optimization to local base stock policies, but rather, we consider more sophisticated ordering policies as well. Evidently, having a downstream stage “smooth out” the variability in a demand process can reduce the necessary safety stocks upstream in the supply chain. Even though this smoothing results in more inventory at the downstream stage, if the conditions are right, global costs can decrease.

Although local base stock policies and guaranteed service are not optimal *per se*, they do have many practical advantages. It is easier to manage a supply chain where every stage can be counted on to perform to some shared criteria. The demand bounds and service times are easy to understand and communicate, and orders can be executed using only local information. We regard the essay “Coordination of multi-echelon supply chains using the guaranteed service framework,” as one formalized argument for why GS models are easy to manage.

4. Limitations and future directions

Each of the three essays answers questions, but also suggests new ones for future research. Some specific follow-up questions are discussed in the respective essays. In all three essays, we develop the initial analysis on serial systems and other simple network topologies. Real-world supply chain structures are often more complex, and may have

multiple customer stages and even cycles. In the work on coordination, it is relatively easy to extend the results to more complex systems. It appears to be more challenging to extend the work on forecasts and capacity constraints. One difficulty is how to establish valid order bounds for stages that serve multiple different customers (or downstream stages). One can always use the simple sum of the downstream bounds as a conservative bound, but this does not capture any savings from statistical economies of scale (or risk pooling), and so the bound may not be tight (and therefore, inefficient). For the forecast-driven orders, we explore this problem in a separate Addendum. Supply chains with both capacity constraints and multiple demand nodes are left for future investigations.

This speaks to a general theme: it is often possible to analytically address various challenges *one at the time*. However, when multiple complications arise it is often very difficult, if not impossible, to mathematically characterize the necessary safety stocks, let alone finding optimal solutions. This situation is common in other areas of operations as well. The resolution may be higher reliance on simulation and numerical solutions. In the addendum we show how bounds on multiple merged forecast processes can be determined by measuring inventory variation directly; this is a step in that direction.

Finally, our discussions on “optimality” touch upon the fact that different frameworks use different performance criteria and make different assumptions about how the supply chain operates. These frameworks also differ in terms of the effectiveness of available optimization algorithms, and in terms of how easily various constraints can be incorporated into the analysis. In this thesis, we reconsider some of the assumptions of the GS framework, and also make connections with completely different supply chain frameworks (for example, MRP). From both a theoretical and a practical perspective, it would be desirable to explore further these types of connections. It would be valuable to investigate in depth how different methods and optimality criteria lead to different solutions, and if there are specific situations for which one framework or the other is particularly well suited.

II. Strategic safety stocks in supply chains with capacity constraints

We generalize the guaranteed-service (GS) model for multi-echelon safety stock placement to include capacity constraints. We first develop an extension of the single-stage base-stock model to include a capacity constraint. We then use this result to model a multi-stage system with a base-stock operating policy. We establish that we can adapt the existing algorithms for the un-constrained case to solve for the safety stocks in a capacitated system. We then consider a multi-stage system in which stages censor their orders, based on their capacity limits. Again we analytically characterize the necessary base stock levels, and develop an extension to the existing dynamic programming algorithms to find the optimal base stock levels and safety stocks. The censored order policy leads to a better solution compared to that for the base-stock policy. Indeed, we find that the total holding costs for the censored order policy can be less than that for the corresponding base-stock system without capacity constraints.

5. Introduction

A central question in supply chain management is how to coordinate activities and inventories over a large number of stages and locations, while providing a high level of service to end customers. Simpson (1958) found that if the individual stages in a serial-system supply chain operate according to local base stock policies with service guarantees, then the globally optimal safety stock strategy is to concentrate inventory to certain key locations, effectively decoupling different parts of the supply chain. Simpson also proposed an enumerative algorithm for determining these locations. Once the globally optimal safety stock strategy has been determined, each stage of the supply chain can operate independently, providing guaranteed service to its downstream customer, and operating according to a simple base stock policy, with a minimum need for communication and coordination between different parts of the supply chain. Graves and Willems (2003) term this framework for supply chain management as the guaranteed service (GS) model.

For a review of work on GS models we cite the overview articles of Inderfurth (1991), Diks et al. (1996) and Graves and Willems (2003). We note in particular that

Graves and Willems (2000) extend Simpson's work to supply chains with spanning tree topology, and formulate a polynomial-time dynamic programming algorithm. Optimizing general networks is an NP-hard problem (Lesnaia et al. 2005); nevertheless, Humair and Willems (2007) have developed very effective algorithms for optimizing the safety stocks in large-scale real-world supply chains. We also note that the GS framework has been deployed successfully in industry (e.g., Billington et al 2004).

However, to our knowledge, all published work on the GS model assumes unlimited capacity for processing and inventory storage at each stage. In reality, there are often limits to the quantity of goods that can be transported, processed or stored in a given time frame. If a stage is unable to process a large order in a short period of time, then this may cause stock-outs that are not anticipated by existing theoretical models. Thus the first goal of this paper is to generalize Kimball's (original manuscript 1955, reprinted in 1988) single-stage base-stock model to account for a capacity constraint (also, see Simpson, 1958). We then show how to combine multiple stages into a network, so that we can optimize the inventory across a multi-stage supply chain with capacity constraints. We show in particular how to extend the decoupling structural property and the effective optimization methods developed for the un-capacitated case to this setting. Secondly, we propose a modification of the base-stock policy, specifically, that a node should propagate an order which is the lesser of its capacity and the order it receives (plus extra quantities to "catch up", as necessary). We refer to this as the "censored" base-stock policy. We show that we can optimize the safety stock inventory in supply chains with censored ordering and capacity constraints, with small modifications to the Graves-Willems' dynamic programming method. We find that the inventory holding costs for the censored base-stock policy are less than for the original base-stock policy, and sometimes even less than that for the corresponding system without capacity constraints. Moreover, for the censored policy the orders still depend only on local information.

Whereas capacity constraints have not been analyzed before within the GS model, there has been some progress for Clark and Scarf's (1960) framework of echelon-based ordering, sometimes referred to as the stochastic service (SS) model. A complete characterization of the optimal solution has not been obtained for capacity constraints for SS models, and, indeed, Speck and van der Wal (1991) show by example that the

echelon-based ordering policy is generally not optimal in multi-echelon systems with capacity constraints. Gallego and Scheller-Wolf (2000) look at a single stage system with fixed ordering costs and capacity constraints, and find that the optimal policy takes (s,S) form. For a single stage, Gallego and Toktay (2004) characterize the optimal policy in the high fixed ordering cost regime, under the assumption that all orders are full capacity orders. Parker and Kapuscinski (2004) provide a detailed analysis in a two-echelon, serial system, and show that a modified echelon base stock (MEDS) is optimal. We will also consider a modified policy, but both the “original” policy and the modification are different. Glasserman and Tayur (1994) consider the stability properties of multi-echelon systems with capacity constraints. They find that inventories and back-logs are stable (i.e., they converge to unique stationary distributions from any initial state) if the mean demand is less than the capacity constraint. In subsequent papers, Glasserman and Tayur assume that an echelon-base stock policy is used in a multi-echelon system, and find optimal order points using simulation and perturbation analysis (Glasserman and Tayur, 1995), and analytical approximations (Glasserman and Tayur, 1996).

Gupta and Selvaraju (2006) develop an approximation for setting echelon-based service levels when the supply chain is modeled as a queueing network, and the stages have exponentially distributed service times. Although we will not deeply explore the relationship between this work and ours, we do in fact also employ queueing theory in order to estimate a certain cost term.

A markedly different approach is taken by Bertsimas and Thiele (2004,2006), who show that capacity constraints can be incorporated into a tractable, robust optimization problem. This approach can handle general networks, and uses echelon-based policies as in the Clark and Scarf (1960) model. A similarity between the robust optimization approach and ours is that there is no need to specify a probability distribution for demand. However, our work differs from *all* of the aforementioned work in that we consider “local” (as opposed to echelon-based) base stock ordering policies, and, as mentioned, guaranteed service constraints rather than back-order costs and stochastic service.

This paper is organized as follows. In §6, we generalize the base-stock model of Kimball (1988) and Simpson (1958) to include a capacity constraint in a single-stage

setting. In §7, we consider optimization of a supply chain with a base-stock policy and potentially any number of capacity constraints. In particular, we find that the optimization procedures and structural results identified by Simpson (1958) and Graves and Willems (2000) carry over to this setting. In §8, we analyze what happens if each stage modifies its order based on its capacity constraint, i.e., censors the order. We find that the structural properties and optimization methods carry over to this setting as well. In §9, we perform a numerical experiment to illustrate our methods and to examine the structure and performance of these policies. We conclude the paper with a discussion on our findings and suggestions for possible future work in §10.

6. Single stage model

In this section we generalize the single-stage model originally developed by Kimball (1988) and Simpson (1958) to include a capacity constraint.

In this model, a stage represents a processing activity that requires one or more inputs and that converts these inputs into an output product or final good. The output can be stored as inventory at the stage, and is used to meet demand from multiple customers or as input into downstream stages. The stage might represent the procurement of a raw material, or the production of a component, or the manufacture of a subassembly, or the assembly and test of a finished good, or the transportation of a finished product from a distribution center to a warehouse.

We let $d(t)$ denote the demand in period t . We assume that the stage provides the same *guaranteed service time* S to each of its customers; this means that the stage guarantees that it will satisfy the demand $d(t)$ by time $t + S$, where S is a non-negative integer.

We also assume that the suppliers to the stage provide a guaranteed service time, which we denote by SI for the inbound service time. Thus, for an order placed at time t , the suppliers will deliver their inputs to the stage at time $t + SI$.

We assume the stage has a capacity limit c and has a known deterministic production lead-time, call it T . Each period the stage can release into production any amount up to the capacity limit of c , assuming that all of the inputs are available and on

hand. The production lead-time is the time from when production is started until production is completed and available to serve demand. The production lead-time includes the waiting and processing time at the stage, plus any transportation time to put the item into inventory.

We assume that the stage operates with a periodic review base-stock replenishment policy with a review period being one time unit (e.g., one day). The timing of events is as assumed by Kimball (1988) and Simpson (1958). In each period t , the stage first observes its demand $d(t)$ and then places an order on each of its upstream suppliers. The stage then receives the earlier order placed at time $t - SI$ from each of the upstream suppliers. Next the stage decides the quantity to release into its process; the stage then completes the process on the release quantity from time $t - T$ and places this quantity into its inventory. Finally the stage serves the demand from period $t - S$, namely $d(t - S)$.

For the base-stock policy without a capacity constraint, Kimball (1988) and Simpson (1958) assume that in each period t , the stage places an order, equal to $d(t)$, on each of its upstream suppliers to replenish the inputs necessary to replenish the observed demand. When these inputs are received by the stage at time $t + SI$, the stage will then initiate production of $d(t)$ units; that is, the release quantity at time $t + SI$ is $d(t)$, which will complete the process and be placed in inventory at time $t + SI + T$.

We now adapt this policy to account for a capacity constraint. We again assume that in each period t , the stage places an order, equal to $d(t)$, on each of its upstream suppliers to replenish the inputs necessary to replenish the observed demand. When these inputs are received by the stage at time $t + SI$, the stage will attempt to initiate production of $d(t)$ units, subject to capacity availability. If the production starts are less than $d(t)$ units due to the capacity limits, then the delayed production will be started as soon as capacity is available. More specifically, we assume that the extra supplier material gets placed into an internal queue $IQ(t)$ while it waits until there is sufficient capacity for processing (i.e., when demand once again falls below c). We illustrate the envisioned arrangement in Figure 1.

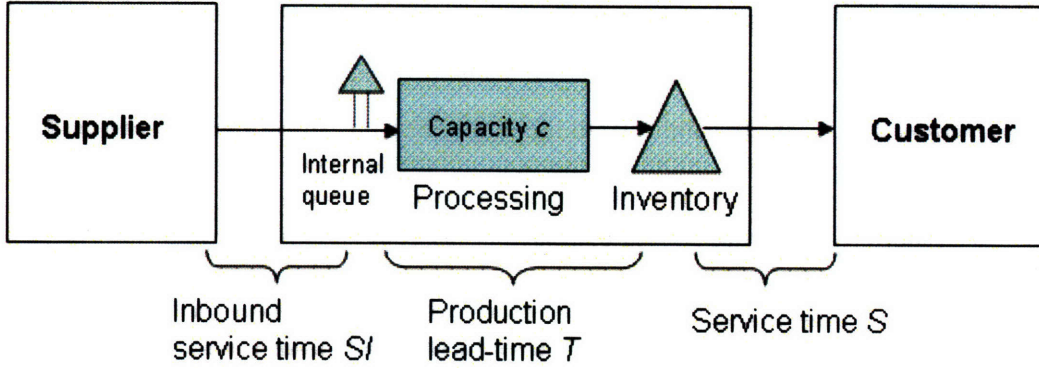


Figure 1: Overview of a single-stage system

We denote the production release at the stage at time t by $R(t)$. Without capacity constraints, we would have $R(t) = d(t - SI)$; with capacity constraints, we have

$$R(t) = \min(c, d(t - SI) + IQ(t - 1)), \quad (1)$$

where we specify the internal queue $IQ(t)$ by the equation:

$$IQ(t) = IQ(t - 1) + d(t - SI) - R(t). \quad (2)$$

The balance equation for the final-good inventory $I(t)$ at the stage is now:

$$I(t) = I(t - 1) + R(t - T) - d(t - S). \quad (3)$$

Combining (2) and (3) we have

$$I(t) + IQ(t - T) = I(t - 1) + IQ(t - T - 1) + d(t - T - SI) - d(t - S). \quad (4)$$

We assume that the system starts at time $t = 0$ with

$d(t) = 0, IQ(t) = 0$, for $t \leq 0$ and $I(0) = B$, where B is the base-stock level. Then for suitably large t we can write the inventory as:

$$I(t) = B - d(t - T - SI, t - S) - IQ(t - T) \quad (5)$$

where we define the notation

$$d(a, b) = \begin{cases} \sum_{i=a+1}^b d(i) & \text{for } a < b \\ 0 & \text{for } a = b \\ -\sum_{i=b+1}^a d(i) & \text{for } a > b \end{cases} \quad (6)$$

In the Appendix we show that we can express the internal queue as:

$$IQ(t) = \max_{n \in Z} \{d(t - SI - n, t - SI) - cn\} \quad (7)$$

where we use Z to denote the set of non-negative integers. We substitute (7) into (5) to obtain:

$$I(t) = B - \max_{n \in Z} \{d(t - SI - T - n, t - S) - cn\}. \quad (8)$$

The base-stock problem is now to determine the minimal value of B that assures that the inventory $I(t)$ is always non-negative, which is sufficient to satisfy the guaranteed service commitment.

We assume that $d(t)$ is non-negative and takes the average value μ where $\mu < c$. Furthermore, as in Kimball (1988) and Simpson (1958), for the purposes of setting the base stock and safety stock levels, we assume that *demand is bounded*. Specifically we assume that there exists a function $D(\tau)$ that bounds demand over any τ consecutive periods. That is,

$$D(\tau) = \max \{d(t, t + \tau)\} \quad \forall t, \tau \geq 0. \quad (9)$$

The combination of guaranteed service and bounded demand constitutes the most significant assumptions in the GS framework. Simply put, we assume that as long as demand stays within certain bounds, there should always be enough safety stock to meet that demand within the service time. This general approach applies well to the typical context in which the implicit and explicit costs of stocking out are perceived to be much greater than the costs of holding inventory. We refer to Graves and Willems (2000) for more discussion and motivation of these assumptions.

We will restrict our attention to demand bounds with certain properties, as follows.

Defintion 1. *A bound function $D(\tau)$ on $\tau \in [0, \infty)$ is said to be valid if $D(0) = 0$, and if it is non-decreasing, and concave. For $\tau < 0$ we define $D(\tau) = 0$.*

These properties hold true for demand bounds that arise in practice. Intuitively, the maximum possible demand over some time period will usually increase with the length of the period, but with diminishing rates. We can now combine (8) and (9) to find that for $I(t) \geq 0$ the minimal base stock is:

$$B(\tau) = \max_{n \in \mathbb{Z}} \{D(\tau + n) - cn\} \quad \text{where } \tau = T + SI - S. \quad (10)$$

In (10) τ denotes the *net replenishment time* for the stage without the capacity limits. We write the base stock in (10) as a function of τ to make explicit its dependence on this parameter. When we optimize the safety stocks across a supply chain, the decision variables will be the service times (S, SI) for each stage, which combine with the given lead time T to determine the un-capacitated net replenishment time τ . We will limit our attention to capacity constraints of certain magnitudes as follows.

Definition 2. A capacity constraint c is said to be valid with respect to a valid bound function $D(\tau)$ if there exist a single point $\tilde{\tau} > 0$ such that $D(\tilde{\tau}) = c\tilde{\tau}$, and that $D(\tau) > c\tau \quad \forall \tau < \tilde{\tau}$ and $D(\tau) < c\tau \quad \forall \tau > \tilde{\tau}$.

The intuitive meaning of this is that we assume demand can exceed the production capacity over some time interval (otherwise the capacity constraint could be omitted from the model), but given sufficiently long time there must be enough capacity to meet any valid demand realization (otherwise guaranteed service is infeasible).

For a given bound function, we can easily find the base stock level for any value of τ . To get some insight into the nature of the solution, let us suppose that the demand bound and the capacity constraints are valid as defined above, and in addition that the demand bound is differentiable. Let us further ignore the integrality restriction on the argument n in (10). Then we can perform the maximization in (10) to get an explicit formula for the base stock level as a function of τ . We define q by $D(q) = c$, i.e., q is the point at which the derivative of the demand bound equals the capacity. Then the base stock is:

$$B(\tau) = \begin{cases} 0 & \text{for } \tau < q - \frac{D(q)}{c} \\ c(\tau - q) + D(q) & \text{for } q - \frac{D(q)}{c} \leq \tau < q \\ D(\tau) & \text{for } \tau \geq q \end{cases} . \quad (11)$$

For a valid demand bound and capacity constraint we note that $q - \frac{D(q)}{c} < 0$; hence we permit the net replenishment time to be negative. That is, unlike in the un-capacitated case, a stage may find it economical to quote a service time S that is longer than the nominal time $SI + T$ that it takes for its inventory to be replenished; due to the capacity constraint, though, the actual replenishment time can exceed this nominal time. We note that the base stock level will be zero at the point $\tau = q - \frac{D(q)}{c}$. For higher values of τ , the necessary base stock level grows linearly at rate c until $\tau = q$; beyond $\tau = q$ the base stock level equals the demand bound function, as is true for the un-capacitated case. For completeness we note that $B(\tau) = 0$ for $\tau < q - \frac{D(q)}{c}$; however, when optimizing the safety stocks in a supply chain, we need not consider net replenishment times in this range, as we can show that any solution in this range can be dominated by another solution with $\tau = q - \frac{D(q)}{c}$ in terms of the inventory requirements.

In practice we often set the bound function analogous to a probabilistic service level with i.i.d. normally distributed demand. That is, we set

$$D(\tau) = \mu\tau + z\sigma\sqrt{\tau}, \quad (12)$$

where σ corresponds to the standard deviation of demand and z is a safety factor. We note that this bound is valid and differentiable, and that any $c > \mu$ will constitute a valid

capacity constraint. With $z = 2$, we have $q = \left(\frac{\sigma}{c - \mu}\right)^2$ for this demand function and we

can find from (11) the base stock level to be:

$$B(\tau) = \begin{cases} 0 & \text{for } \tau < -\left(1 - \frac{\mu}{c}\right) \left(\frac{\sigma}{c - \mu}\right)^2 \\ c\tau + \frac{\sigma^2}{c - \mu} & \text{for } -\left(1 - \frac{\mu}{c}\right) \left(\frac{\sigma}{c - \mu}\right)^2 \leq \tau < \left(\frac{\sigma}{c - \mu}\right)^2 \\ \mu\tau + 2\sigma\sqrt{\tau} & \text{for } \tau \geq \left(\frac{\sigma}{c - \mu}\right)^2 \end{cases} \quad (13)$$

Thus the necessary base stock level for a stage with a capacity constraint takes a rather simple and intuitive form. First, we see that there is a threshold value for the net replenishment time $\tau = \left(\frac{\sigma}{c - \mu}\right)^2$, beyond which the capacity constraint does not matter, in that the base stock is the same as for the un-capacitated system. Second, we see that when the capacity constraint is relevant the base stock depends not just on the demand variability σ and net replenishment time τ but also on the amount of “slack capacity” $c - \mu$. Third, in this range the base stock is a **fixed amount** $\frac{\sigma^2}{c - \mu}$ plus a **variable amount that increases linearly at rate c** in the net replenishment time. We plot the capacitated base stock level (13) in Figure 2, together with the base stock level for the unconstrained case.

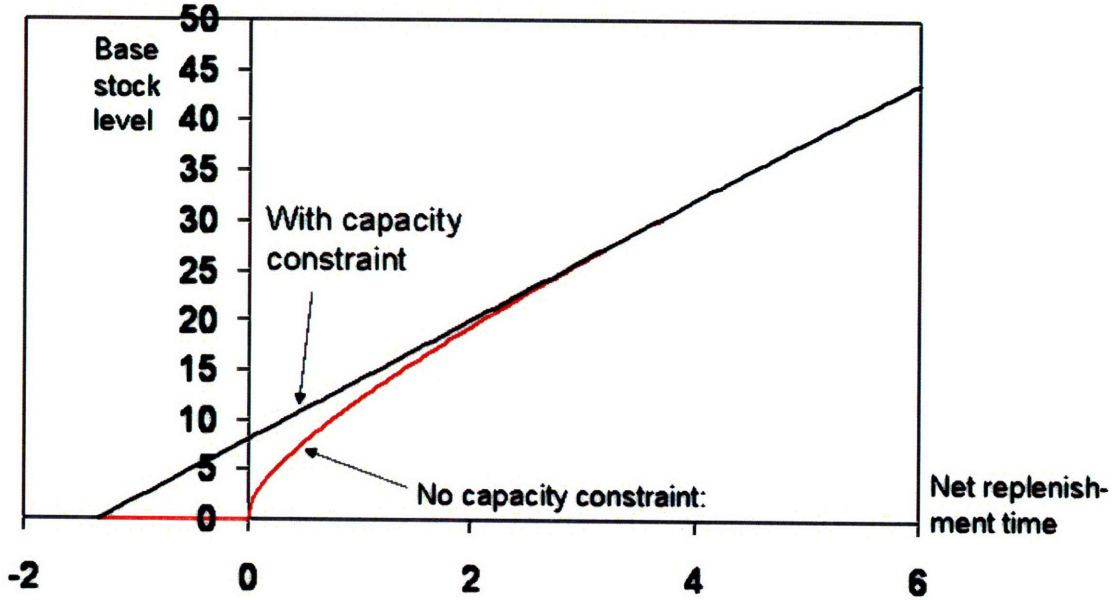


Figure 2: With capacity constraints, for small τ a higher base stock level is necessary, but for $\tau \geq 4$ the capacity constraint does not matter. Plotted parameters are $\mu = \sigma = 4$, $c = 6$ and $z = 2$

To get some additional insight, we use (5) and (13) to find the average inventory in finished goods and the internal queue $E[I(t)] + E[IQ(t-T)]$; as we expect the internal queue to be modest, we use this to approximate the average safety stocks:

$$B(\tau) - \mu\tau = \begin{cases} (c - \mu)\tau + \frac{\sigma^2}{c - \mu} & \text{for } -\left(1 - \frac{\mu}{c}\right)\left(\frac{\sigma}{c - \mu}\right)^2 \leq \tau < \left(\frac{\sigma}{c - \mu}\right)^2 \\ 2\sigma\sqrt{\tau} & \text{for } \tau \geq \left(\frac{\sigma}{c - \mu}\right)^2 \end{cases}. \quad (14)$$

We again note that there is a threshold value for the net replenishment time, beyond which the capacity constraint does not matter. When the capacity constraint is relevant, the safety stock is a **fixed amount** $\frac{\sigma^2}{c - \mu}$ plus a **variable amount that increases linearly** in the net replenishment time. In contrast, for the un-capacitated system, the safety stock is **proportional to the square root** of the net replenishment time. Finally, as the slack capacity goes to zero, the fixed amount of safety stock increases hyperbolically; this is analogous to what happens to the waiting time in a queue as utilization goes to one.

Similar to the traditional square-root formula for safety stocks in an un-capacitated setting, we regard (14) to be a valuable back-of-the-envelope heuristic in that it succinctly displays how the safety stock depends on the slack capacity, the demand variability and the net replenishment time.

7. Multiple stage model with base-stock ordering

In the previous section, we considered the necessary base stock level for a single stage with a capacity constraint. We now investigate a supply chain consisting of multiple stages, each of which may potentially have a capacity constraint. In this network, each stage provides guaranteed service to its customers (downstream neighbors), and operates according to a (local) base-stock policy. We assume that customer demand is immediately propagated up through the system, so that in each period t each node places an order on its suppliers equal to the sum of the customer demand from all its adjacent downstream stages.

In order to describe the network and its characteristics, we index the nodes (or stages) and denote the parameters S_k, T_k, SI_k and c_k specific to node k . To specify the topology of the network, we define the directed edge set A so that $(j, k) \in A$ indicates that node j directly supplies (is upstream of) node k . Customer facing nodes are defined by the set C , and have service times exogenously specified; $S_k = s_k$ for $k \in C$. Other service times (and inbound service times) are *decision variables*. The demand bound $D_k(\cdot)$ is a bound on the demand from the customer node(s) downstream of node k ; see Graves and Willems (2000) for how to combine the bounds from multiple demand streams. To facilitate the derivations to follow, we use operator notation (see e.g. Griffel, 1985) to describe how we determine the base stock level from the capacity constraint and the demand bound. We use the symbol ψ_k to denote the continuous and node-specific version of (10) as follows:

$$(\psi_k D_k)(\tau) = \max_{n \geq 0} \{D_k(\tau + n) - c_k n\}. \quad (15)$$

As before, we set the base stock level to this quantity to ensure guaranteed service:

$$B_k(\tau) = (\psi_k D_k)(\tau). \quad (16)$$

If node k does not have a capacity constraint, we can set $c_k = \infty$ in (15) and find that $B_k(\tau) = D_k(\tau)$. At stage k , the total inventory that is on hand (either as finished inventory or delayed in the internal queue) is $I_k(t) + IQ_k(t)$. We use the node-specific version of equation (5) to characterize the average of this quantity, $\bar{I}_k + \overline{IQ}_k$:

$$\begin{aligned} \bar{I}_k + \overline{IQ}_k &= B_k - \overline{d(t - T_k - SI_k, t - S_k)} - \overline{IQ}_k + \overline{IQ}_k \\ &= (\psi_k D_k)(T_k + SI_k - S_k) - (T_k + SI_k - S_k)\mu \end{aligned} \quad (17)$$

Note that we do not include inventory in process at stage k , because the average level of this pipeline inventory is proportional to the lead time T_k and is not affected by the choice of service times. We will not consider pipeline inventory further. We assume that stage k accrues holding costs proportional to $\bar{I}_k + \overline{IQ}_k$, with the proportionality constant h_k . This is a simplification that we make for tractability. In many contexts one might expect the holding cost for the internal queue inventory to be less than that for the finished good, as holding costs should increase as we add more value to the product by processing. Nevertheless, we expect this difference in holding costs to be modest; moreover, the average internal queue \overline{IQ}_k does not depend on the service times and thus has no impact on the optimal solution.

We are now ready to formulate an optimization problem; the problem is to find the service times that minimize the total inventory holding cost, subject to providing guaranteed service at all nodes, for any demand realization within the bounds.

$$\begin{aligned} \min_{S_k, SI_k} \sum_{k=1}^N h_k & \left((\psi_k D_k)(SI_k + T_k - S_k) - (SI_k + T_k - S_k)\mu \right) \\ S_k, SI_k & \geq 0 \quad \forall k \\ SI_k & \geq S_j \quad \forall (j, k) \in A \\ S_k & = s_k \quad k \in C \\ SI_k + T_k - S_k & \geq q_k - \frac{D_k(q_k)}{c_k} \quad \forall k \end{aligned} \quad (18)$$

The decision variables are the service times, which are non-negative by the first constraint. The second constraint assures that the inbound service time for each node is

greater than or equal to the maximum service time from its supply nodes. The third constraint fixes the service times for customer-facing nodes to the exogenous specifications. The fourth constraint provides a lower bound on the net replenishment time for each node, as discussed in the prior section, where q_k is specified by $D_k'(q_k) = c_k$. Simpson (1958) and Graves and Willems (2000) formulate the un-capacitated version of (18) for a serial system and for general networks, respectively. In both cases, they observe that an optimal solution is on a corner of the solution space, since the problem minimizes a concave objective function over a polyhedral set. We are therefore interested in whether this observation applies here, namely whether the modified function $(\psi_k D_k)(\tau)$ is concave for each node k . Under some reasonable technical conditions, this is in fact the case.

Proposition 1. *Suppose $D(\tau)$ is valid, and c valid with respect to $D(\tau)$. Then*

$$(\psi D)(\tau) = \max_{n \geq 0} \{D(\tau + n) - cn\} \text{ is concave on } \tau \in [q - \frac{D(q)}{c}, \infty).$$

The proofs of all propositions are in the Appendix. Thus, the objective function in (18) is concave as long as we have a valid bound and capacity constraint at each node; hence, the optimal solution will be at an extreme point of the solution space. One practical implication of this is that something similar to the all-or-nothing result identified by Simpson (1958) holds in the capacitated setting as well for serial systems. Specifically, for a serial system either $S_k = 0$, meaning that the stage holds enough inventory to always provide immediate service, or alternatively, $S_k = SI_k + T_k - q_k + \frac{D(q_k)}{c_k}$ in which case the base stock level $B_k = 0$. In the latter case, the net replenishment time is at its lower bound, and the service time S_k exceeds the nominal replenishment time $SI_k + T_k$. Simpson (1958) solved the un-capacitated version of (18) for a serial system by enumeration. Graves and Willems (2000) develop an exact dynamic programming algorithm, which can be used for networks with spanning-tree topology; Lesnaia (2004) shows how to modify and implement this algorithm so that it is polynomial. We will not

review this algorithm here, but will establish that we can modify the Graves-Willems algorithm to solve (18) for spanning-tree networks with capacity constraints. In particular we need to make two modifications. First, instead of using $B_k(\tau) = D_k(\tau)$ for the base stock levels, we use the exact characterization of the base stock level necessary to handle capacity constraints given by (16). Second, the only other change in the problem formulation (18), relative to the un-capacitated problem, is that the lower bound on the net replenishment time is no longer zero, but is given by $q_k - \frac{D_k(q_k)}{c_k}$ for node k .

To account for this, we just need to extend the search space for each iteration of the dynamic program; this can be done with no change to the computational complexity.

8. Multiple stage model with censored order policy

In the prior section we have shown how to generalize the GS framework, and associated optimization methods to encompass capacity constraints. In this section, we will show that certain improvements are possible, if the stages with capacity constraints modify their orders. The basic idea with the censored policy is that a stage should not propagate a full order upstream, if it knows that it will be unable to process such a quantity because of its capacity constraints. Alternatively, we observe that there is no need for an internal queue at a capacitated stage; rather the stage should place orders on its supplier so that these orders arrive at a rate consistent with the stage's capability to process the work.

To simplify the development in this section, we will consider a serial system, where we number the nodes from downstream to upstream; thus node 1 is the customer facing node, and N is the most upstream node. We will briefly discuss censorship in other network structures at the end of this section.

When we censor orders, then we need to distinguish the orders placed by each node, as different nodes will generate different series of orders due to their capacity constraints. We denote the order *received* by node k at time t by $d_k(t)$; we will denote its bound by $D_k(t)$, where

$$d_k(t, t + \tau) \leq D_k(\tau) \quad \forall t, \tau \geq 0. \quad (19)$$

Accordingly, we write $d_1(t) = d(t)$ and $D_1(t) = D(t)$ for customer demand and its bound. Suppose node k has capacity c_k . We assume that it will never order more than its capacity. Thus whenever its demand exceeds its capacity ($d_k(t) > c_k$), it orders less than the demand and creates a shortfall. When demand falls below c_k again, stage k will increase its orders to catch up with the lost quantities. To this end, node k keeps a backlog $BL_k(t)$ of this shortfall, equal to the amount of its demand for which it has yet to place a replenishment order. Node k will add this to its orders as soon as capacity is available. We now specify the orders placed at time t by node k on node $k + 1$ by:

$$d_{k+1}(t) = \min(d_k(t) + BL_k(t-1), c_k). \quad (20)$$

We can specify the backlog $BL_k(t)$ recursively:

$$\begin{aligned} BL_k(t) &= \max\{BL_k(t-1) + d_k(t) - c_k, 0\} \\ &= \max\{\max\{BL_k(t-2) + d_k(t-1) - c_k, 0\} + d_k(t) - c_k, 0\} \\ &= \max_{n \in \mathbb{Z}}\{d_k(t-n, t) - c_k n\}. \end{aligned} \quad (21)$$

Here we assume that the system starts at $t = 0$ with $BL_k(t) = 0, d_k(t) = 0$ for all $t \leq 0$. In each period, stage k adds the difference between its demand and capacity, $d_k(t) - c_k$, to the backlog, subject to keeping the backlog non-negative. In Table 1 below, we illustrate the censored order and the backlog with a simple example, using the capacity constraint $c_k = 8$, and an initial backlog of 0:

Order d_k	6	7	9	9	7	6	6
Censored order d_{k+1}	6	7	8	8	8	7	6
Backlog BL_k	0	0	1	2	1	0	0

Table 1: Example of orders, censored orders, and backlog

We can show that the inventory for node k , $I_k(t)$, is the inventory for the un-capacitated problem, net of the backlog at time $t - SI_k - T_k$. Since the replenishment time is

$SI_k + T_k$, anything in the backlog at node k at time $t - SI_k - T_k$ cannot be available by time t to meet demand at node k . Thus, we have:

$$\begin{aligned}
I_k(t) &= B_k - d_k(t - SI_k - T_k, t - S_k) - BL_k(t - SI_k - T_k) \\
&= B_k - d_k(t - SI_k - T_k, t - S_k) - \max_{n \in \mathbb{Z}} \{d_k(t - n - SI_k - T_k, t - SI_k - T_k) - c_k n\} \\
&= B_k - \max_{n \in \mathbb{Z}} \{d_k(t - n - SI_k - T_k, t - S_k) - c_k n\}
\end{aligned} \tag{22}$$

Except for the fact that the demand d_k is now stage-specific, this is equivalent to the inventory equation for a capacitated stage that does not censor its orders, namely equivalent to (5). That is, when considering the inventory $I_k(t)$ at some stage, it makes no difference whether items are waiting in the internal queue, or whether the orders placed by that stage were temporarily put into a backlog because of censorship. In both cases the quantities that start and finish their processing in each period are the same. Thus if node k has a capacity constraint, we can use equation (16) to determine the base stock level that guarantees service, regardless of whether a censored order policy is employed or not.

Thus we set the base stock level by

$$B_k(\tau) = (\psi_k D_k)(\tau) .$$

where we use the operator ψ_k defined in (15).

There are a couple of immediate implications from the censored order policy. First, in comparison to the base-stock ordering the average inventory will be less (for a fixed base-stock level), because orders never exceed capacity and there is no internal queue. We obtain the total average inventory by taking the average of (22):

$$\bar{I}_k = B_k - \mu(SI_k + T_k - S_k) - \overline{BL}_k \tag{23}$$

Thus in the case of censored orders, we need to calculate the term \overline{BL}_k , in order to determine average inventory levels and costs. The term \overline{BL}_k depends on specific properties of the demand distribution, and is generally difficult to estimate; we will discuss this topic in greater detail in the Appendix. However, we note that \overline{BL}_k does not depend on the decision variables (the service times). Thus we do not need to determine \overline{BL}_k to find the optimal safety stocks; rather, the sole purpose for determining \overline{BL}_k is for the determination of the average inventory level given by (23).

A second implication of the censored order policy is that the censored order is bounded by the capacity at stage k , i.e., $(d_{k+1}(t) \leq c_k)$; thus, a looser capacity constraint at stage $k + 1$ ($c_{k+1} > c_k$) is irrelevant. Indeed, any upstream capacity constraint that is greater than a downstream capacity limit can be ignored.

A third more significant implication of the censorship is that the upstream stages will face a different demand bound, one that is censored by node k 's capacity. We describe next how to determine the bound on the censored orders.

Proposition 2. *Suppose $d_{k+1}(t) = \min(d_k(t) + BL_k(t-1), c_k)$ where BL_k is given by (21), and initialized with $BL_k(t) = 0$ for $t \leq 0$. Assume that D_k is a valid bound for d_k (that is, $d_k(t, t + \tau) \leq D_k(\tau) \quad \forall t, \tau \geq 0$), and that c_k is valid with respect to $D_k(\tau)$. Then*

$$d_{k+1}(t, t + \tau) \leq D_{k+1}(\tau) = (\Phi_k D_k)(\tau) \quad \forall t, \tau \geq 0 \quad (24)$$

where Φ_k is defined by

$$(\Phi_k D)(\tau) = \min(c_k \tau, D(\tau)). \quad (25)$$

This bound is tight, in that for every $\tau \geq 0$ there is some $d_k(t, t + \tau) = D_k(\tau)$.

Thus we have an evaluative model for a serial system with capacity constraints and censored orders. We assume that the demand is propagated up the supply chain by (21), with (22) to account for the backlog of orders. We can then use Proposition 2 to compute the demand bound at each node. Given these demand bounds we can then use (16) to characterize the necessary base stock level for each node. Given the base stock level, we can calculate the average inventory from (23). We summarize these iterative steps in Table 2, with a comparison to the un-capacitated model.

	No capacity constraint at node k	Capacity constraint c_k at node k
Orders placed	$d_{k+1}(t) = d_k(t)$	$d_{k+1}(t) = \min(d_k(t) + BL_k(t-1), c_k)$
Bound on orders placed	$D_{k+1}(\tau) = D_k(\tau)$	$D_{k+1}(\tau) = (\Phi_k D_k)(\tau)$ $= \min(c_k \tau, D_k(\tau))$
Base stock level	$B_k(\tau_k) = D_k(\tau_k)$	$B_k(\tau_k) = (\psi_k D_k)(\tau_k)$ $= \max_{n \geq 0} \{D_k(\tau_k + n) - c_k n\}$
Average inventory	$\bar{I}_k = B_k(\tau_k) - \mu \tau_k$	$\bar{I}_k(t) = B_k(\tau_k) - \mu \tau_k - \overline{BL}_k$

Table 2: Summary of node properties in serial system supply chain with capacity constraint(s) and censored base stock policy; we use τ_k to denote the net replenishment time at node k .

We can now embed this model in an optimization, analogous to (18), to find the best choices for the service times. In the following proposition we establish that the demand bounds and capacity constraints are in fact valid as defined earlier; as these properties were necessary to derive the necessary base stock levels. We also show that the resulting base stock levels $B_k(\tau_k)$ are concave functions.

Proposition 3. *Suppose that end demand $d(t) = d_1(t)$ is bounded by $D(\tau)$ and that $D(\tau)$ is valid. Assume further that some subset of nodes has capacity constraints, and that these are all valid with respect to $D(\tau)$, and that these c_k are decreasing with increasing k . Finally, suppose that each node k places orders $d_{k+1}(t)$ according to (20). Then*

- a) All orders d_k are bounded by $D_k(\tau)$, as specified by (24)
- b) All $D_k(\tau)$ are valid
- c) c_l for nodes with capacity constraints are valid with respect to $D_k(\tau)$, for all $l \geq k$
- d) The base stock levels $B_k(\tau)$ as specified by (16) ensure that $I_k(t) \geq 0$
- e) All $B_k(\tau)$ are concave in τ .

Thus we have shown that in a serial-system supply chain with capacity constraints and censorship, we can compute the demand bounds and necessary base stock levels by recursively applying a sequence of functional operators, as summarized in Table 2. Thus, as for the case of base-stock ordering, we can use the algorithm developed by Graves and Willems to find the optimal service times for a serial-system supply chain with both capacity constraints and censorship, after only small modifications.

We can extend the results in this section to arborescent (assembly tree) supply chain topologies in which each node has a single customer node. The iterative steps laid out in Table 2 apply directly, but with one modest modification: the order placed by node k (given by (21)) is now placed concurrently on each of the suppliers to node k . Again we can use the existing algorithm from Graves and Willems to find the optimal service times and safety stocks.

The extension to supply chains with several end demand nodes (as in a distribution network, for example) is not as immediate. The primary challenge is to determine how to combine multiple demand bounds, each of which may be censored. For un-capacitated supply chains, Graves and Willems (2000) propose how to set bounds if demand streams are independent, and the bound is set analogous to a probabilistic service level, as in (12). They also propose bounds for larger or smaller measures of risk pooling. If one or more bound are generated by a censored order policy, then it is not clear how best to merge bounds from multiple streams. Of course, one can always obtain a valid and conservative bound by simply adding the bounds of downstream stages; we leave for future research the question of how to improve upon this demand bound for supply chains operating with a censored order policy.

9. Numerical experiments

To test the results from previous sections and to get some intuition for the properties of the optimal solution, we performed a number of numerical experiments. We used the same supply chain and cost structures as in Graves and Willems (2008). Specifically, we considered a serial system with $N = 5$ nodes, and with three alternatives for both the cost accumulation and the production lead-time as follows:

Stage	5	4	3	2	1
Increasing	36	28	20	12	4
Constant	20	20	20	20	20
Decreasing	4	12	20	28	36

Table 3: Alternative structures for supply chain lead-time and cost accumulation

The terms “increasing” and “decreasing” should be understood in terms of going upstream starting from the customer facing stage 1. In the case of cost accumulation, the values stated in Table 3 represent the cost added at each stage. For example, for the increasing cost scenario, the cost at stage 5 is 36, the cost at stage 4 is $36 + 28 = 64$, the cost at stage 3 is $36 + 28 + 20 = 84$, etc. For all three scenarios the cost of the finished good at stage 1 is 100.

For the production lead-times, the values for each scenario represent the values for T_i . For each scenario the cumulative lead-time for the supply chain is 100. We assumed that the demand bound was given by $D(\tau) = \mu\tau + z\sigma\sqrt{\tau}$, with the parameters $(\mu, z, \sigma) = (40, 2, 20)$.

We considered a supply chain with no capacity constraint, as well as a single capacity constraint at any one of the 5 nodes. The value of the capacity constraint was taken from the set (42, 45, 50, 60, 70), thus representing $\mu + 0.1\sigma$ up to $\mu + 1.5\sigma$. Moreover, in each test problem, we solved for both the base-stock policy and the censored ordering policy. The alternatives outlined above made for a total of $3 \times 3 \times (1 + 5) \times 5 \times 2 = 540$ test problems. In all cases, we recorded the average total cost of the safety stocks, as well as the optimal solution.

We will first discuss the results for the test problems under the base-stock policy. In Table 4 below, we normalize the total cost of the average supply-chain safety stock to be 1.0 for the unconstrained problem; the results are equivalent for average holding costs provided that the holding costs are proportional to the inventory costs. In this table, we show how the cost increase depends on the size and location of the constraint for the test problems with constant lead time and increasing costs.

	Location of capacity constraint					
c_k	(none)	5	4	3	2	1
42	1.00	1.03	1.07	1.13	1.19	1.01
45	1.00	1.00	1.04	1.12	1.16	1.00
50	1.00	1.00	1.04	1.06	1.08	1.00
60	1.00	1.00	1.02	1.03	1.04	1.00
70	1.00	1.00	1.01	1.02	1.03	1.00

Table 4: Normalized costs for base-stock policy, for constant lead time and increasing cost scenarios.

		Location of capacity constraint					
Cost	Lead time	(none)	5	4	3	2	1
Increasing	Increasing	40.0	102%	111%	117%	114%	100%
	Constant	40.0	106%	113%	117%	119%	100%
	Decreasing	40.0	107%	113%	117%	119%	100%
Constant	Increasing	36.8	100%	100%	102%	102%	100%
	Constant	39.4	100%	104%	112%	116%	100%
	Decreasing	40.0	103%	108%	112%	116%	100%
Decreasing	Increasing	26.8	100%	100%	100%	100%	100%
	Constant	34.6	100%	100%	102%	109%	100%
	Decreasing	39.2	100%	100%	103%	113%	100%

Table 5: Total costs for base-stock policy with capacity constraint $c = 45$. In each case, the cost of the constrained problem is given as a percentage of the corresponding unconstrained problem.

In Table 5 we report the total inventory cost for the un-capacitated cases and then report the relative costs for the capacitated cases for a base-stock policy with capacity constraint $c = 45$. We see from Table 5 that the experiments with increasing (decreasing) costs had higher (lower) total costs, due to the higher holding costs for the non-customer facing stages. Conversely, decreasing lead times implied long lead times downstream (where holding costs are higher) and led to higher holding costs overall.

From both Table 4 and Table 5 we see that the cost of a capacity constraint is greater the smaller the capacity. The location of the capacity constraint matters greatly, but without a simple pattern. To better understand this, we looked more closely at the optimal solution in a few scenarios as listed in Table 6 below.

Net replenishment time at stage	Cost	5	4	3	2	1
No constraint	1.0	20	0	0	0	80
Capacity 45 at stage 1	1.0	20	0	0	0	80
Capacity 45 at stage 3	1.12	0	0	60	0	40

Table 6: The cost and optimal solution for scenario with constant lead time and constant cost accumulation

We see that in the unconstrained problem, the optimal solution is to have one large inventory (and a long net replenishment time) at the first, customer-facing stage, and a small inventory at stage 5. If we introduce a capacity constraint at stage 1, then one may use the same solution - we recall from equation (11) and Figure 2 that for sufficiently large net replenishment times the capacity constraint will not matter. In this case it means no extra inventory even though capacity at stage 1 is quite significantly constrained at $45 = \mu + 0.25\sigma$. As is clear from Table 5, this was in fact the case for all the examples with a capacity constraint of 45 at the first stage. More specifically, from (13) we know that the capacity constraint does not matter when

$$\tau \geq \left(\frac{\sigma}{c - \mu} \right)^2$$

$$c \leq \frac{\sigma}{\sqrt{\tau}} + \mu = \frac{20}{\sqrt{80}} + 40 = 42.2 \quad .$$

Equivalently, we found that the capacity constraint $c = 45$ only matters when the net replenishment time τ is less than 16.

On the other hand, if such a capacity constraint is located at stage 3, then the original solution cannot guarantee service, since stage 3 will need safety stock even if its

net replenishment is zero. The optimal solution is not simply to add extra inventory to stage 3, but rather to completely change the safety stock strategy, at a cost of about 12% of the original optimum.

Thus if we locate a capacity constraint at a node that holds a safety stock in the un-constrained problem, then we find that the cost of the constraint can be quite limited. Informally speaking, one might say that the cost of a capacity constraint depends on whether it is “in harmony” with the un-capacitated optimal solution. Indeed, we found for our entire set of test problems under the base-stock policy that if the stage with the capacity constraint had inventory in the un-constrained problem, then the optimal service times *did not change* when we introduced the constraint, and the average additional cost was only 3.9%. On the other hand, if the constrained stage did not originally have inventory, then in 44.1% of the cases, the optimal service times changed when the constraint was introduced, and the average cost increase was 5.6%.

We also performed corresponding experiments for the censored order policy; the optimal costs relative to the unconstrained problem are listed in Table 7 and Table 8 below.

	Location of capacity constraint					
c_k	(none)	5	4	3	2	1
42	1.00	0.98	0.95	0.91	0.85	0.55
45	1.00	0.98	0.97	0.99	1.01	0.83
50	1.00	0.99	1.02	1.02	1.03	0.94
60	1.00	0.99	1.01	1.01	1.01	0.97
70	1.00	1.00	1.00	1.01	1.01	0.98

Table 7: Normalized costs for censored order policy, constant lead time and cost accumulation scenarios

		Location of capacity constraint					
Cost	Lead time	(none)	5	4	3	2	1
Increasing	Increasing	40.0	98%	102%	104%	100%	85%
	Constant	40.0	102%	104%	106%	107%	87%
	Decreasing	40.0	103%	105%	107%	108%	89%
Constant	Increasing	36.8	98%	93%	90%	84%	69%
	Constant	39.4	98%	97%	99%	101%	83%
	Decreasing	40.0	101%	102%	104%	106%	88%
Decreasing	Increasing	26.8	99%	96%	89%	77%	60%
	Constant	34.6	99%	97%	93%	93%	74%
	Decreasing	39.2	100%	98%	97%	99%	82%

Table 8: Total costs for base-stock policy with capacity constraint $c = 45$. In each case, the cost of the constrained problem is given as a percentage of the corresponding unconstrained problem, for censored orders.

We see that the costs are, especially for smaller capacities in Table 7 significantly reduced relative to the uncensored case in Table 4. Indeed, for the 225 test problems with a capacity constraint, we found an average cost reduction of 8.0% when we replace the base-stock policy with the censored ordering policy. Thus when capacity constraints are present, a strong case can be made for censoring the orders at that stage.

Remarkably, under censorship the costs are often even lower than in the corresponding problem *without capacity constraints*. Indeed, for the 225 test problems, we find that the total cost for the constrained system with a censored order policy is on average 3.6% lower than the total cost for the corresponding unconstrained base-stock system. Nevertheless, as is clear from both Table 7 and Table 8, the effect can be positive or negative depending on the size and location of the constraint. Since the benefits of censorship are mostly found upstream of the capacitated stage, it is not surprising that the best outcome (in terms of total supply chain costs) is when the constraint is downstream in the supply chain, ideally at the customer-facing stage. This is especially true when, as in our examples, the customer-facing stage needs to carry inventory to provide a zero service time to its customers. Indeed, in all the cases we

investigated, for a given capacity value, the best outcome was always when the constraint is at the customer facing-node. In order to make a meaningful impact, the capacity constraint needs to be only slightly higher than average demand; in fact, for all the examples with censorship at the customer-facing stage, the lowest tested capacity $\mu + 0.1\sigma$ always gave the best value. However, we also know from theory, and specifically equation (13), that as the capacity approaches average demand, the necessary base stock level goes to infinity.

When determining the average cost for the censored ordering policy, one must calculate the term \overline{BL}_k in (23). We discuss this topic in greater detail in the Appendix. We note here that this term will depend on specific properties of the demand process, properties which up until this point have not been specified. In the experiments listed here we estimated this term using formula (A28). When we compared these answers to ones obtained using numerical estimates of \overline{BL}_k , the average difference in terms of total costs was only 2.1%, and none of the overall results and conclusions were significantly different from those presented here. We also emphasize that while the value of the term \overline{BL}_k does impact the total cost, it does not affect the optimal solution, nor does a poorly estimated \overline{BL}_k compromise the guaranteed service constraint.

10. Conclusions and discussion

We have analyzed the inclusion of capacity constraints in the context of the GS model for safety stocks in multi-echelon supply chains. We have shown how to extend the single-stage base-stock model to include a capacity constraint. We have used this result to model multi-stage supply chains with capacity constraints. We have characterized the base stock level for two cases that depend on how orders are propagated across the supply chain. In both cases we can extend the structural findings and solution methods that have been developed for the un-capacitated supply chains, e.g., by Simpson (1958) and Graves and Willems (2000).

In general we expect to need more safety stock when we have capacity constraints. Indeed, as is clear from (13), the costs associated with capacity constraints

can be arbitrarily large, as the “slack capacity” goes to zero. On the other hand, for stages with sufficiently high net replenishment time, there may be no additional cost associated with capacity constraints. When optimizing a supply chain the total effect depends significantly on whether the constraints are located at stages that had safety stocks in the corresponding unconstrained problem.

When we use the censored order policy, the costs are not only lower than in the corresponding capacity-constrained problem with base-stock ordering, but frequently even lower than in the unconstrained problem as well. Intuitively, a capacity constraint typically increases the base stock level for the stage with the constraint (Equation (10); Figure 2); however, the resulting increase in the average inventory may not be as great because some of the increase in the base stock ends up as the positive backlog. Furthermore, the censored orders can be much smoother than the original demand process. As a consequence there can be a substantial reduction in the need for inventory at upstream nodes (Proposition 2; Table 7). The total effect may imply higher or lower costs depending on specific parameters.

On a more abstract level, it may seem surprising or even paradoxical that adding a constraint can lead to a better solution. The explanation is that we have in effect expanded the space of possible ordering policies. The original GS model from Simpson (1958) is based under the assumption that all stages operate with a (local) base stock policy. The results from these experiments concretely illustrate that this policy need not be optimal in a multi-echelon supply chain with guaranteed service. Indeed, one might want to introduce a censored order policy even in the absence of capacity constraints in order to get these benefits. In our experiments the best outcome was when the customer-facing node was tightly constrained. This suggests that it may be preferable for the first stage in a supply chain to act as a damper, whereby it absorbs the variability from a demand signal, rather than pass along this variability to the rest of the supply chain.

As we have noted, the costs (but not the optimal solution) of systems with censorship depend on the term \overline{BL}_k , which in turn depends on specific properties of the demand distribution, and which moreover appears difficult to estimate. This suggests opportunities for future research, but also hints at certain fundamental limitations on the ability to predict the performance of censored systems. It is a rare practical situation in

which one can be confident about higher-order properties of the demand distributions (although, in our experiments, the total system costs were quite similar when we estimated \overline{BL}_k in different ways). The challenges of calculating \overline{BL}_k also make it difficult to find “optimal” censorship value(s) and location(s), although surely improvements are possible over the simple brute-force searches we presented in our experiments. Another opportunity for future research is to generalize our results to more complex supply chains, for example those with general network structures.

Appendix 1: The average backlog

Here we consider estimating \overline{BL}_k . We recall that this quantity is necessary to understand the expected inventory costs for a node with capacity constraints and censorship.

However, \overline{BL}_k does not depend on the decision variables (the service times) and it is not needed in order to obtain or implement the optimal solution.

We note that the back log described in (21) behaves rather like a queue - there is a random quantity of arrivals every period, and a fixed, maximum processing rate. Even though BL_k operates in discrete time, we can use continuous-time queuing theory as an approximation. Suppose that we model the internal queue with an M/D/1 queue with arrival rate λ , and deterministic processing time s . We then seek λ and s that agree with the average μ and standard deviation σ of demand per period, or $\frac{\mu}{c}$ and $\frac{\sigma}{\sqrt{c}}$ if we normalize with respect to capacity. That is, we model demand using a continuous-time model with Poisson arrivals, but we ensure that the probability distribution of the number of arrivals agrees in first and second moments to our original process. We are making a second order, continuous-time, approximation:

$$\begin{aligned}\frac{\mu}{c} &= \lambda s \\ \frac{\sigma}{\sqrt{c}} &= \sqrt{\lambda s}\end{aligned}\tag{A26}$$

Conversely, if we already have a mean and standard deviation for demand, we can invert the relations (A26) solve for s and λ .

$$\begin{aligned}s &= \frac{\mu}{\lambda c} = \frac{\mu}{\left(\frac{\sigma^2}{s^2}\right)} = \frac{\mu s^2}{\sigma^2} \\ s &= \frac{\sigma^2}{\mu} \\ \lambda &= \frac{\mu}{sc} = \frac{\mu^2}{c\sigma^2}\end{aligned}\tag{A27}$$

Having calculated s and λ , we can use the Pollaczek-Khintchine formula (with zero processing time variability) for calculating expected number of jobs and the expected waiting time. This formula is exact for Poisson arrivals in continuous time, but for a discrete time system it is only an approximation. Noting that the utilization is simply

$$\rho = \frac{\mu}{c}, \text{ we have:}$$

$$\begin{aligned} \overline{BL} &= \overbrace{\left(\rho + \frac{\rho^2}{2(1-\rho)} \right)}^{\text{Expected number of jobs}} \times \overbrace{s}^{\text{Time per job}} \\ &= \left(\frac{\mu}{c} + \frac{\left(\frac{\mu}{c} \right)^2}{2\left(1 - \frac{\mu}{c}\right)} \right) \left(\frac{\sigma^2}{\mu} \right) \\ &= \left(1 + \frac{\mu}{2(c-\mu)} \right) \left(\frac{\sigma^2}{c} \right) \\ &= \left(\frac{2c-\mu}{c-\mu} \right) \left(\frac{\sigma^2}{2c} \right) \end{aligned} \tag{A28}$$

Finally, we mention that if one wants to estimate \overline{BL}_k for more complex (not i.i.d.) demand processes or if greater precision is desired, it is easy to estimate numerically or using historical data. One can simply evaluate (21),

$$BL_k(t) = \max \{ BL_k(t-1) + d_k(t) - c_k, 0 \}, \text{ for a real or simulated sequence of demand}$$

realizations $d_k(t)$ and calculate the average value \overline{BL}_k . In the context of the numerical experiments in §9, we compared the formula (A28), with the aforementioned numerical estimate, using a normal distribution.

c	42	45	50	60	70
Simulated/ Normal	88.5	29.6	10.6	2.5	0.7
PK-formula/ Poisson	104.8	44.4	24.0	13.3	9.5

Table 9: \overline{BL}_k for $\mu = 40, \sigma = 20$, and various censorship values c .

The continuous-time formula appears to give much higher values of \overline{BL}_k for large capacities, but for smaller capacities the results are reasonably similar. An explanation for this is the continuous time assumption in the Pollaczek-Khintchine formula; there will frequently (and on average) be a queue just after each job arrival, but in the discrete-time system processing effectively happens “after” all the job arrivals in each period. This effect becomes less and less significant as the capacity constraint becomes smaller and smaller and the busy periods become longer and longer.

While these methods yield rather different results, the term \overline{BL}_k is typically only responsible for a small portion of all costs. Fortunately, the different methods are increasingly similar as the cost contribution from the \overline{BL}_k term grows and makes a significant contribution. Thus in the experiments we performed, the average total cost difference between the two methods was, on average, only 2.1%.

Appendix 2: Proofs and derivations

Derivation of (7). First we apply (2) recursively on $IQ_k(t-1)$, $IQ_k(t-2)$, and so on:

$$\begin{aligned} IQ(t) &= \max \{IQ(t-1) + d(t-SI) - c, 0\} \\ &= \max \{ \max \{IQ(t-2) + d(t-SI-1) - c, 0\} + d(t-SI_k) - c, 0 \} \\ &= \max \{IQ(t-2) + d(t-SI) + d(t-SI-1) - 2c, d(t-SI) - c, 0\} \\ &= \dots = \\ &= \max_{n \in \mathbb{Z}} \{d(t-SI-n, t-SI) - cn\} \end{aligned} \tag{A29}$$

Proof of Proposition 1. We define

$$\hat{\tau} = \arg \max_{\tau} \{D(\tau) - c\tau\} \quad (\text{A30})$$

Because $D(\tau)$ is concave (by Definition 1) and crosses $c\tau$ at a single point $\tilde{\tau}$ (by Definition 2), there must be some maximizing positive $\hat{\tau} < \tilde{\tau}$ (if there are multiple maximizing values, any one can be picked as $\hat{\tau}$ for this proof). Now

$$\begin{aligned} B(\tau) &= (\psi D)(\tau) \\ &= \max_{n \geq 0} \{D(\tau + n) - cn\} \\ &= c\tau + \max_{n \geq 0} \{D(\tau + n) - c(\tau + n)\} \\ &= c\tau + \max_{x \geq \tau} \{D(x) - cx\} \end{aligned} \quad (\text{A31})$$

Now if $\tau \leq \hat{\tau}$, then the constraint is irrelevant and x can take the maximizing value $\hat{\tau}$ from (A30). However, if $\tau > \hat{\tau}$, then x will take the smallest value possible (because D is concave, and crosses $c\tau$ at a unique point), which is τ . Thus:

$$B(\tau) = \begin{cases} c\tau + D(\hat{\tau}) - c\hat{\tau} & \text{for } \tau \leq \hat{\tau} \\ D(\tau) & \text{for } \tau > \hat{\tau} \end{cases} \quad (\text{A32})$$

Now we are ready to prove concavity, by definition. For $\tau_1 < \tau_2 \leq \hat{\tau}$ or $\hat{\tau} < \tau_1 < \tau_2$ we must clearly have that

$$B(\lambda\tau_1 + (1-\lambda)\tau_2) \geq \lambda B(\tau_1) + (1-\lambda)B(\tau_2), \quad (\text{A33})$$

since both $c\tau + D(\hat{\tau}) - c\hat{\tau}$ and $D(\tau)$ are individually concave. Now for $\tau_1 < \hat{\tau} < \tau_2$ let us first suppose that $\lambda\tau_1 + (1-\lambda)\tau_2 \leq \hat{\tau}$. Then

$$\begin{aligned} B(\lambda\tau_1 + (1-\lambda)\tau_2) &= c(\lambda\tau_1 + (1-\lambda)\tau_2) + D(\hat{\tau}) - c\hat{\tau} \\ &= \lambda(c\tau_1 + D(\hat{\tau}) - c\hat{\tau}) + (1-\lambda)(c\tau_2 + D(\hat{\tau}) - c\hat{\tau}) \\ &= \lambda B(\tau_1) + (1-\lambda)(c\tau_2 + D(\hat{\tau}) - c\hat{\tau}) \end{aligned} \quad (\text{A34})$$

However, by definition (A30) of $\hat{\tau}$ we must have that $c\tau + D(\hat{\tau}) - c\hat{\tau} \geq D(\tau)$, and hence

$$\begin{aligned} B(\lambda\tau_1 + (1-\lambda)\tau_2) &\geq \lambda B(\tau_1) + (1-\lambda)D(\tau_2) \\ &= \lambda B(\tau_1) + (1-\lambda)B(\tau_2) \end{aligned} \quad (\text{A35})$$

On the other hand, if $\lambda\tau_1 + (1-\lambda)\tau_2 = \tau_3 > \hat{\tau}$, then

$$\begin{aligned}
B(\tau_3) &\geq \\
&= B(\hat{\tau}) + \frac{\tau_3 - \hat{\tau}}{\tau_2 - \hat{\tau}} (B(\tau_2) - B(\hat{\tau})) \\
&= B(\hat{\tau}) \left(1 - \frac{\tau_3 - \hat{\tau}}{\tau_2 - \hat{\tau}} \right) + \frac{\tau_3 - \hat{\tau}}{\tau_2 - \hat{\tau}} B(\tau_2)
\end{aligned} \tag{A36}$$

This holds because $B(\tau)$ is concave by assumption for $\tau > \hat{\tau}$. Now we note that

$$B(\hat{\tau}) = \left((c\tau_2 + D(\hat{\tau}) - c\hat{\tau}) - B(\tau_1) \right) \frac{\hat{\tau} - \tau_1}{\tau_2 - \tau_1} + B(\tau_1) \tag{A37}$$

This is simply describing $B(\hat{\tau})$ as a point on a line between $B(\tau_1)$ and $c\tau_2 + D(\hat{\tau}) - c\hat{\tau}$.

Combining (A36) and (A37) we have

$$\begin{aligned}
&B(\tau_3) \\
&\geq \left(\left((c\tau_2 + D(\hat{\tau}) - c\hat{\tau}) - B(\tau_1) \right) \frac{\hat{\tau} - \tau_1}{\tau_2 - \tau_1} + B(\tau_1) \right) \left(1 - \frac{\tau_3 - \hat{\tau}}{\tau_2 - \hat{\tau}} \right) + \frac{\tau_3 - \hat{\tau}}{\tau_2 - \hat{\tau}} B(\tau_2) \\
&\geq \left((D(\tau_2) - B(\tau_1)) \frac{\hat{\tau} - \tau_1}{\tau_2 - \tau_1} + B(\tau_1) \right) \left(1 - \frac{\tau_3 - \hat{\tau}}{\tau_2 - \hat{\tau}} \right) + \frac{\tau_3 - \hat{\tau}}{\tau_2 - \hat{\tau}} B(\tau_2) \\
&= \left((B(\tau_2) - B(\tau_1)) \frac{\hat{\tau} - \tau_1}{\tau_2 - \tau_1} + B(\tau_1) \right) \left(1 - \frac{\tau_3 - \hat{\tau}}{\tau_2 - \hat{\tau}} \right) + \frac{\tau_3 - \hat{\tau}}{\tau_2 - \hat{\tau}} B(\tau_2) \\
&\equiv H(\tau_3)
\end{aligned} \tag{A38}$$

The second inequality comes from noting the maximizing property of $\hat{\tau}$. On the last line we just defined $H(\tau_3)$ as an affine function of τ_3 . Now we evaluate $H(\cdot)$ at $\hat{\tau}$ and τ_2 .

This gives us

$$\begin{aligned}
H(\hat{\tau}) &= \left((B(\tau_2) - B(\tau_1)) \frac{\hat{\tau} - \tau_1}{\tau_2 - \tau_1} + B(\tau_1) \right) \\
H(\tau_2) &= B(\tau_2)
\end{aligned} \tag{A39}$$

Exactly the same holds if we instead evaluate the function $(B(\tau_2) - B(\tau_1)) \frac{\tau_3 - \tau_1}{\tau_2 - \tau_1} + B(\tau_1)$.

But two affine functions that take the same values at two different points are identical, and so:

$$B(\tau_3) \geq H(\tau_3) = (B(\tau_2) - B(\tau_1)) \frac{\tau_3 - \tau_1}{\tau_2 - \tau_1} + B(\tau_1) \tag{A40}$$

This proves that final case and the proof is complete. \square

Proof of Proposition 2. Let $\tilde{\tau}$ be the point such that $D_k(\tilde{\tau}) = c_k \tilde{\tau}$, which must exist per Definition 2. Clearly, by the ordering mechanism (20), $d_{k+1}(t)$ can never exceed c_k , and so we must have that

$$d_{k+1}(t, t + \tau) \leq c_k \tau \leq D_k(\tau) \quad \tau \leq \tilde{\tau}. \quad (\text{A41})$$

In order to investigate $\tau > \tilde{\tau}$ we note that:

$$\begin{aligned} d_{k+1}(t, t + \tau) &= d_k(t, t + \tau) - BL_k(t + \tau) + BL_k(t) \\ &\leq d_k(t, t + \tau) + BL_k(t) \\ &= d_k(t, t + \tau) + (d_k(t - \tilde{n}, t) - c_k \tilde{n}) \\ &= d_k(t - \tilde{n}, t + \tau) - c_k \tilde{n}. \end{aligned} \quad (\text{A42})$$

where

$$\tilde{n} \equiv \min n \geq 0 : BL_k(t - n) = 0. \quad (\text{A43})$$

That is, $\tilde{n} \geq 0$ is the number of periods that node k has been working at capacity before time t . By the assumption that $BL_k(t) = 0$ for some sufficiently low t , there must always exist such an \tilde{n} . We can replace \tilde{n} defined by (A43) with a maximizing n ; we will still have a valid (although potentially looser) bound:

$$\begin{aligned} d_{k+1}(t, t + \tau) &\leq d_k(t - \tilde{n}, t + \tau) - c_k \tilde{n} \\ &\leq \max_{n \geq 0} d_k(t - n, t + \tau) - c_k n. \end{aligned} \quad (\text{A44})$$

Finally, we invoke the bound on d_k :

$$d_{k+1}(t, t + \tau) \leq \max_{n \geq 0} D_k(\tau + n) - c_k n \quad (\text{A45})$$

However, for $\tau > \tilde{\tau}$ we have, because D_k is concave and $\tilde{\tau}$ is the equality point, that (A45) is maximized for $n = 0$ and hence, for $\tau > \tilde{\tau}$, we have

$$d_{k+1}(t, t + \tau) \leq D_k(\tau) < c_k \tau \quad \tau > \tilde{\tau} \quad (\text{A46})$$

Combining (A41) and gives us the claimed relation. Finally we note that the bound (24) is tight; for example $d_{k+1}(t, t + \tau) = D_{k+1}(\tau)$ is realized if $BL_k(t - 1) = 0$ and

$$d_k(t, t + \tau) = D_k(\tau) \quad \square$$

Proof of Proposition 3: We start by proving a)-c) by induction, noting that they are true by assumption for $k = 0$. The inductive step is trivial if there is no capacity constraint; we therefore consider the case when k does have a capacity constraint. We make the induction hypothesis, that a)-c) are true for some $k-1$, and that node k has a capacity constraint. We can then use Proposition 2 to get that $D_{k+1}(\tau) = \min(c_k \tau, D_k(\tau))$. Thus a) holds for k as well. Moreover, $D_{k+1}(0) = \min(c_k \times 0, D_k(0)) = 0$. Both $c_k \tau$ and $D_k(\tau)$ are non-decreasing and concave, and these properties are preserved under minimization. Hence, if $D_k(\tau)$ is valid then $D_{k+1}(\tau)$ is valid as well, and so b) holds for k .

Suppose now that c) holds for $k-1$, that is, any c_l ($l \geq k - 1$) is valid with respect to $D_k(\tau)$. We need to show that any c_l ($l \geq k$) is valid with respect to $D_{k+1}(\tau)$. By

Definition 2 there is a crossing point such that

$$c_l \tilde{\tau} = D_k(\tilde{\tau}) \tag{A47}$$

By the inductive assumptions a)-c), we can use Proposition 3, and so we have

$$\min(c_k \tilde{\tau}, D_k(\tilde{\tau})) = D_{k+1}(\tilde{\tau}). \tag{A48}$$

Because c_l is decreasing in l , we have

$$c_l \tilde{\tau} = \min(c_k \tilde{\tau}, c_l \tilde{\tau}) \quad \forall l \geq k \tag{A49}$$

Combining (A47)-(A49) gives us

$$c_l \tilde{\tau} = D_{k+1}(\tilde{\tau}) \quad \forall l \geq k \tag{A50}$$

That is, $c_l \tau$ crosses $D_{k+1}(\tau)$ and $D_k(\tau)$ at the same point $\tilde{\tau}$. Furthermore, for $\tau < \tilde{\tau}$ we have $c_l \tau < D_k(\tau)$ and $c_l \tau < c_k \tau$ so $c_l \tau < \min(c_k \tau, D_k(\tau)) = D_{k+1}(\tau)$. For $\tau > \tilde{\tau}$ we have $c_l \tau > D_k(\tau) \geq \min(c_k \tau, D_k(\tau)) = D_{k+1}(\tau)$. Thus, c_l is valid with respect to $D_{k+1}(\tau) = \min(c_k \tau, D_k(\tau))$ as well. Thus c) holds for k as well.

Therefore, we have shown that a)-c) for node $k-1$ imply that a)-c) hold for k as well. Since the base case $k = 1$ is true by assumption, by the induction axiom a)-c) must hold for all k . This means that the necessary assumptions for Propositions 1 and 2 are fulfilled for all k , and this proves d) and e), respectively. \square

III. Coordination of multi-echelon supply chains using the guaranteed service framework

We investigate how the guaranteed-service (GS) framework for multi-echelon safety stock placement can be used when different parts of the supply chain are controlled by different parties. We find that this framework is naturally well suited for decentralized decision-making, and we propose a specific, simple contract structure which facilitates such relationships. This contract is incentive compatible and has several other desirable properties; it is also simpler than contracts proposed for coordination in the stochastic service (SS) framework. We also highlight the role of holding costs, how these should be calculated, and some of the difficulties that this might cause decentralized supply chains.

11. Introduction

A central question in supply chain management is how to coordinate activities and inventories over a large number of stages and locations, while providing a high level of service to end customers. This question is particularly delicate when different parts of the supply chain are controlled by different parties, which may have competing and conflicting interests. In general, selfish behavior in different parts of the supply chain may lead to globally suboptimal behavior, as illustrated for example by a game-theoretic analysis (Corbett and Karmarkar, 2001) of a multi-echelon supply chain controlled by two or more parties.

In this paper, we investigate how coordination works when the different parties operate according to the guaranteed service (GS) framework (so termed by Graves and Willems, 2003), each optimizing his or her own part of the supply chain. In the GS framework, the individual stages operate according to local base stock policies and provide guaranteed service, as long as demand falls within certain bounds. Simpson (1958) found that for a serial system, the globally optimal safety stock strategy is to concentrate inventory to certain key locations, effectively decoupling different parts of the supply chain. Simpson also proposed an enumerative algorithm for determining these locations.

For a review of work on GS models we cite the overview articles of Inderfurth (1991), Diks et al. (1996) and Graves and Willems (2003). We note in particular that Graves and Willems (2000) extend Simpson's work to supply chains with spanning tree topology, and formulate an effective dynamic programming algorithm. Optimizing general networks is an NP-hard problem (Lesnaia et al. 2005); nevertheless, Humair and Willems (2007) have developed very effective algorithms for optimizing the safety stocks in large-scale real-world supply chains. We also note that the GS framework has been deployed successfully in industry (e.g., Billington et al 2004).

In this paper, we consider the problem of how to use GS models in contexts in which two different parties control different parts of a supply chain. The different parties could represent different companies, or different business units with separate performance metrics. If such separate parties apply GS models to their own sections of the supply chain, will the results coincide with those that are obtained from a global optimization? We find that provided that the parties can agree on the right service time for orders between them, the global optimum can be obtained. We propose a simple contract structure that codifies such a relationship, and argue that in addition to optimal safety stocks, this form of cooperation has many properties which are desirable for coordination, and which have been discussed by Lee and Whang (1999). We also show that negotiating over the parameters in such a contract is closely related to Nash's (1950) bargaining problem.

GS models thus lend themselves well to decentralization among self-interested parties. By contrast, consider Clark and Scarf's (1960) well known model for echelon-based ordering, sometimes referred to as a stochastic service (SS) model. In this framework, orders depend on echelon inventory, and so to place an order a stage must obtain information about the inventory positions of all the downstream stages of the supply chain. This can be challenging even if the different stages have a common goal, but it is of course especially problematic if the stages have different priorities; it may not even be in their best interests to truthfully share inventory information with one another. (Nevertheless, as shown by Axsäter and Rosling (1993), echelon-based base stock policies often have an equivalent representation in the form of local base stock policies.) Another complication, which has been pointed out by Lee and Whang (1999), is that in

Clark and Scarf's model, only the customer facing nodes face back-order costs if they run out. Therefore, if the stages are independently managed, upstream stages have no incentives to carry any inventory at all. Thus in order to implement Clark and Scarf's solution in one type of decentralized supply chain, Lee and Whang (1999) propose a rather elaborate scheme involving transfer pricing, consignment, shortage reimbursement, and an additional backlog penalty. Similarly, Berling and Marklund (2006) consider a one-warehouse, multiple-retailer system and obtain a near-optimal solution by enforcing an induced backorder cost. Chu and Leon (2004) study coordination in a system with the same topology, but with fixed ordering costs and deterministic demand. They propose heuristics for decision-making both under local and central information.

We do not here attempt to give a comprehensive overview of the large and growing literature on coordination in supply chain management; instead we refer the reader to Cachon (2003), Chen (2003), and Cachon and Netessine (2004) for overviews. We point out that a distinguishing feature of our work is that we consider coordination of *safety stocks* in *multi-echelon supply chains*. By contrast, most existing work considers coordination of other decision variables (for example, price and production quantities) and/or for simpler, two-echelon supply chains.

The remainder of the paper is organized as follows. In §12, we review the GS optimization problem, and show that we can separate it into sub-problems for the different parts of the supply chain. In §13, we propose and discuss a specific contract which facilitates such arrangements. We also relate bargaining over such a contract to the bargaining theory proposed by Nash (1950). In §14, we discuss how to best determine holding costs (for the purpose of safety stock optimization), and some complications that might arise in decentralized supply chains. In §15 we investigate experimentally the costs of choosing the wrong service time between parties, and of using the wrong holding cost. We conclude the paper in §16.

12. The optimization problem and its separation

The global problem

We assume that the supply chain consists of a number of stages. Although we can generalize most of the results derived in this paper to more complex supply chains, we will limit ourselves to serial systems in derivations and examples. Moreover, we will initially consider the problem facing a single decision-maker, and move on to the coordination problem later in the section.

Thus we index the nodes in a serial system by k , and we designate the customer-facing stage as node 1, and node N as the most upstream stage. A stage might represent the procurement of a raw material, or the production of a component, or the manufacture of a subassembly, or the assembly and test of a finished good, or the transportation of a finished product from a distribution center to a warehouse. Each stage k is a potential location for holding a safety-stock inventory of the item processed at the stage.

For each stage, we assume a known deterministic production lead-time, denoted as T_k . When a stage reorders, the production lead-time is the time from when all of the inputs are available until production is completed and available to serve demand. The production lead-time includes the waiting and processing time at the stage, plus any transportation time to put the item into inventory. We assume that there is no capacity constraint and thus, the lead-time is not affected by the size of the order.

We assume that each stage follows a (local) base stock policy and places an order equal to customer demand $d(t)$ in each period. Thus we assume that in each period, the value of customer demand is immediately propagated through the system. Moreover, we assume that each node k provides a *guaranteed service time* S_k ; that is, an order received from the downstream node (or from the customer, for $k = 1$) must always be met within this time.

Thus if node k places an order on node $k + 1$ at time t , then the order will be met by node $k + 1$ at time $t + S_{k+1}$, at which time node k can use it as input into its process. This order will be available as processed inventory at node k at time $t + T_k + S_{k+1}$. Similarly, an order received by node k from node $k - 1$ at time t must be delivered at time $t + S_k$. We designate the difference between these times

$$T_k + S_{k+1} - S_k = \tau_k \quad (51)$$

to be the *net replenishment time*, which we constrain to be non-negative. We regard the service times to be *decision variables*, except for a pre-determined boundary condition $S_1 = s_1$, where s_1 is an exogenous input for the service time guaranteed to the customer.

Significantly, we assume that *demand is bounded*, that is, we define the function $D(\tau)$:

$$D(\tau) = \max \left\{ \sum_{j=t+1}^{t+\tau} d(j) \right\} \quad \forall t, \tau \geq 0 \quad (52)$$

If we initialize the inventory at each stage to a base stock level, i.e., $I_k(0) = B_k$, and if the system operates as described in this section, then we can show that the inventory $I_k(t)$ at stage k will be

$$I_k(t) = B_k - \sum_{j=t-\tau_k+1}^t d(j). \quad (53)$$

Now by setting

$$B_k = B_k(\tau_k) = D(\tau_k), \quad (54)$$

we ensure that $I_k(t)$ is non-negative for all demand realizations within the bound; that is, it is always possible to guarantee service. If the average demand is μ , we can combine (53) and (54) and take the average to get the average inventory:

$$\bar{I}_k = D(\tau_k) - \tau_k \mu. \quad (55)$$

Finally, if each stage incurs a holding cost proportional to the average inventory with proportionality constant h_k , we get the minimization problem **P**:

$$\begin{aligned} \mathbf{P} \quad & \min_{S_2, \dots, S_{N+1}} \sum_{k=1}^N h_k (D(S_{k+1} + T_k - S_k) - (S_{k+1} + T_k - S_k) \mu) \\ & \text{s.t. } S_k - S_{k+1} \leq T_k \quad \text{for } k \in \{1, 2, \dots, N\} \\ & S_1 = s_1 \\ & S_k \geq 0 \quad \text{for } k \in \{2, \dots, N+1\} \end{aligned} \quad (56)$$

The meaning of this is that we seek the service times and inventories that minimize total holding costs, while at the same time fulfilling the guaranteed service time requirement for all stages and all demand realizations within the bounds. $D(\tau)$ is often assumed to be concave, in which case the optimal solution will be on a corner of the polyhedral constraint set. The practical implication of this is the “all-or-nothing” property identified by Simpson (1958); a stage either has inventory and offers immediate service ($S_k = 0$), or it has no inventory at all ($S_k = T_k + S_{k+1}$).

Separation of the problem

We now assume that different parts of the supply chain are controlled by two different players, player 1 who is downstream, and player 2 who is upstream. Specifically, player 1 controls stages $1, 2, \dots, N_1$, and player 2 controls stages $N_1 + 1, \dots, N$. We assume that, as in the original problem, all stages operate according to periodic review base stock policies, provide guaranteed service, and propagate customer demand. Like before, demand is bounded by $D(\tau)$. The players control service times and inventories in their own parts of the supply chain. Then the service time for orders placed by player 1 and delivered to player 2 is S_{N_1+1} . This service time between the players we shall also denote as $S_B = S_{N_1+1}$. Initially, we will just view S_B as an exogenously specified parameter, and investigate the cost impact of agreeing upon various values of S_B . In §13 we will consider bargaining over S_B .

Under the arrangement described here, the players each face a problem that is very similar to the global optimization problem. Specifically, for a given value of S_B , player 1 will face the optimization problem $P_1(S_B)$ as follows:

$$\begin{aligned}
P_1(S_B) \quad & \min_{S_2, \dots, S_{N_1}} \sum_{k=1}^{N_1} h_k (D(S_{k+1} + T_k - S_k) - (S_{k+1} + T_k - S_k)\mu) \\
s.t. \quad & S_k - S_{k+1} \leq T_k \quad \text{for } j \in \{1, 2, \dots, N_1\} \\
& S_1 = s_1, S_{N_1+1} = S_B \\
& S_k \geq 0 \quad \text{for } k \in \{2, \dots, N_1\}
\end{aligned} \tag{57}$$

Similarly, player 2 will face the problem

$$\begin{aligned}
\mathbf{P}_2(S_B) & \min_{S_{N_1+1}, \dots, S_{N+1}} \sum_{k=N_1+1}^N h_k (D(S_{k+1} + T_k - S_k) - (S_{k+1} + T_k - S_k)\mu) \\
s.t. & S_k - S_{k+1} \leq T_k \quad \text{for } k \in \{N_1 + 1, \dots, N\} \\
& S_{N_1+1} = S_B \\
& S_k \geq 0 \quad \text{for } k \in \{N_1 + 1, \dots, N + 1\}
\end{aligned} \tag{58}$$

In Figure 3 below we plot $\mathbf{P}_1(S_B)$ and $\mathbf{P}_2(S_B)$, and their sum (which represents total system holding costs), for an example system with four stages. Each stage has lead (processing) time equal to 20. The demand bound is $D(\tau) = \mu\tau + z\sigma\sqrt{\tau}$, with parameters $\mu = 20, z = 2, \sigma = 20$. The holding costs parameters are, starting with the customer-facing stage and going upstream, 200, 160, 140, 20 (these particular values were chosen so as to make for an instructive example). We also assume that the two downstream stages and the two upstream stages are controlled by different companies; this implied that $S_B = S_3$. We also assume that $S_1 = S_5 = 0$. Thus, each of the two companies only controls a single decision variable (S_2 and S_4 , respectively), with only two possible solutions for each. Thus for a given S_B it is easy to determine the optimal solution in each part, i.e., $\mathbf{P}_1(S_B)$ and $\mathbf{P}_2(S_B)$.

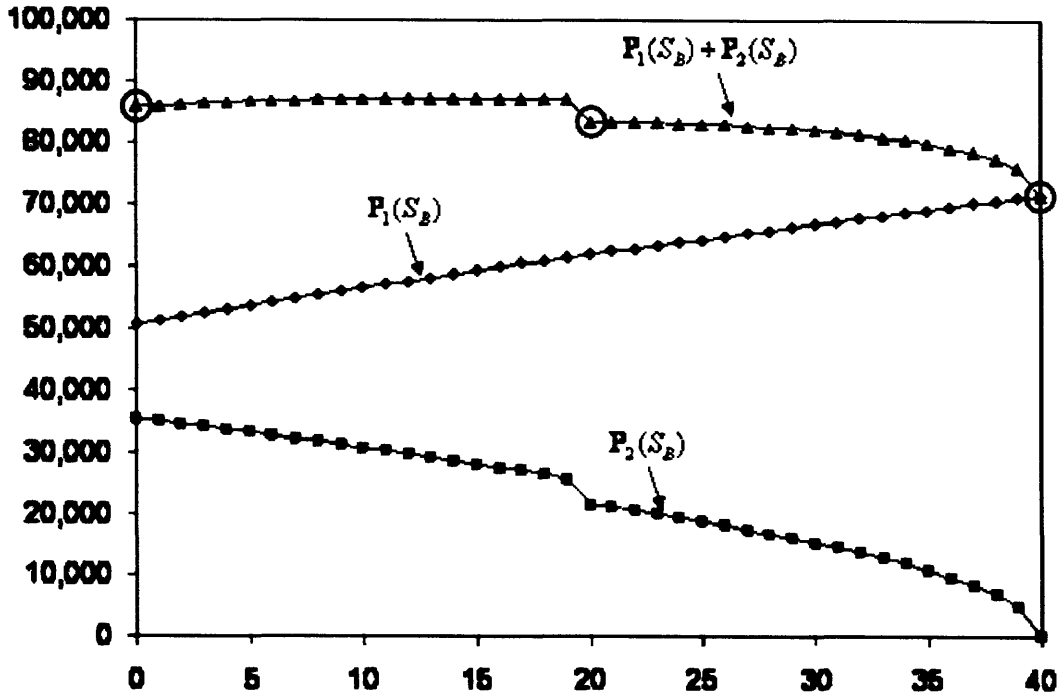


Figure 3: The cost for the lower part, upper part, and total supply chain, as a function of S_B . In this example, the worst S_B resulted in total costs that were 22.1% higher than the best value.

Obviously, $P_1(S_B)$ is increasing in S_B , and $P_2(S_B)$ is decreasing in S_B (player 1 benefits from getting deliveries as quickly as possible, but player 2 would prefer to delay them). We see that $P_1(S_B)$ is here a rather well-behaved function; in fact we show in the Appendix that a concave $D(\tau)$ implies a concave $P_1(S_B)$. It should, however, be apparent from Figure 3 that $P_2(S_B)$ and $P_1(S_B) + P_2(S_B)$ are generally not concave; indeed, the sum can be quite irregular.

Now, suppose that the players would select the value of S_B so that global holding costs were minimized (knowing that they would subsequently solve $P_1(S_B)$ and $P_2(S_B)$, respectively). That is, suppose they sought $\min_{S_B \geq 0} P_1(S_B) + P_2(S_B)$, which is equivalent to the program P , as we shall see in Equation (59) below. In Figure 3 the minimum value is for $S_B = 40$, which means that player 2 does not hold any safety stock at all, but rather delays delivering an order until his or her own inventory has been replenished. We now

proceed to prove the aforementioned equivalence:

$$\begin{aligned}
& \min_{S_B \geq 0} \mathbf{P}_1(S_B) + \mathbf{P}_2(S_B) = \\
& \min_{S_B \geq 0} \left\{ \begin{array}{l} \min_{S_2, \dots, S_{N_1}} \sum_{k=1}^{N_1} h_k (D(S_{k+1} + T_k - S_k) - (S_{k+1} + T_k - S_k)\mu) \\ \text{s.t. } S_k - S_{k+1} \leq T_k \quad \text{for } j \in \{1, 2, \dots, N_1\} \\ S_1 = s_1, S_{N_1+1} = S_B \\ S_k \geq 0 \quad \text{for } k \in \{2, \dots, N_1\} \end{array} \right\} + \\
& \left. \min_{S_B \geq 0} \left\{ \begin{array}{l} \min_{S_{N_1+1}, \dots, S_{N+1}} \sum_{k=N_1+1}^N h_k (D(S_{k+1} + T_k - S_k) - (S_{k+1} + T_k - S_k)\mu) \\ \text{s.t. } S_k - S_{k+1} \leq T_k \quad \text{for } k \in \{N_1+1, \dots, N\} \\ S_{N_1+1} = S_B \\ S_k \geq 0 \quad \text{for } k \in \{N_1+1, \dots, N+1\} \end{array} \right\} \right\} \stackrel{(A)}{=} \\
& \min_{S_B \geq 0} \left\{ \begin{array}{l} \min_{S_2, \dots, S_{N+1}} \sum_{k=1}^N h_k (D(S_{k+1} + T_k - S_k) - (S_{k+1} + T_k - S_k)\mu) \\ \text{s.t. } S_k - S_{k+1} \leq T_k \quad \text{for } k \in \{1, 2, \dots, N\} \\ S_1 = s_1, S_{N+1} = S_B \\ S_k \geq 0 \quad \text{for } k \in \{2, \dots, N+1\} \end{array} \right\} \stackrel{(B)}{=} \\
& \left\{ \begin{array}{l} \min_{S_2, \dots, S_{N+1}; S_B} \sum_{k=1}^N h_k (D(S_{k+1} + T_k - S_k) - (S_{k+1} + T_k - S_k)\mu) \\ \text{s.t. } S_k - S_{k+1} \leq T_k \quad \text{for } k \in \{1, 2, \dots, N\} \\ S_1 = s_1, S_{N+1} = S_B \\ S_k \geq 0 \quad \text{for } k \in \{2, \dots, N+1\} \\ S_B \geq 0 \end{array} \right\} \stackrel{(C)}{=} \\
& \left\{ \begin{array}{l} \min_{S_2, \dots, S_{N+1}} \sum_{k=1}^N h_k (D(S_{k+1} + T_k - S_k) - (S_{k+1} + T_k - S_k)\mu) \\ \text{s.t. } S_k - S_{k+1} \leq T_k \quad \text{for } k \in \{1, 2, \dots, N\} \\ S_1 = s_1 \\ S_k \geq 0 \quad \text{for } k \in \{2, \dots, N+1\} \end{array} \right\} = \mathbf{P} \tag{59}
\end{aligned}$$

Step (A) holds, because for a given S_B , the two programs are independent of each other (that is, changing decision variables in one program does not affect the solution or the constraints in the other). In step (B), we have incorporated the two minimizations into a

single program. Finally, in step (C), we identify the variable S_B as redundant (because it is always equal to S_{N_i+1} , and has no additional constraints), and remove it.

Thus in principle, we can separate the program \mathbf{P} into two components $\mathbf{P}_1(S_B)$ and $\mathbf{P}_2(S_B)$, provided that we can also identify the minimizing value of S_B . As is clear from Figure 3, the global objective function $\mathbf{P}_1(S_B) + \mathbf{P}_2(S_B)$ does in general not take a simple form. On the other hand, if one has information about the constraint set (specifically, the processing times in the upstream part of the supply chain), then one need not investigate the entire space of possible solutions but only look at the corner points (the “all-or-nothing” property). For this problem, these points are $S_B = 0$, $S_B = T_3 = 20$, and $S_B = T_3 + T_4 = 40$; they have been marked with circles in Figure 3.

For the purposes of supply chain coordination, this result is important because it establishes that, assuming that the parties can agree upon the right S_B , it will be in their best interest to manage the supply chain in accordance with the optimal solution. That is, the system is incentive compatible. We will explore this in greater detail in the next section.

Finally, we mention that the derivation (59) naturally generalizes to more complex supply chain structures. We can always separate the global problem into separate sub-problems, plus the problem of finding the right service time in the intersection (or service times, if there are several connections between the parties).

13. A proposed structure for contracts and bargaining

Thus we observe that the optimization problem \mathbf{P} can be separated into sub-problems, which can be managed separately by different players. In Figure 4 below, we propose a specific contract that facilitates such an arrangement.

1. The demand bound $D(\tau)$, within which player 2 commits to meet any demand realization
2. The service time S_b , within which the player 2 must meet all orders
3. The price per unit p , which player 1 agrees to pay for each unit upon placing an order

Figure 4: Terms to be specified in a contract

Next we will argue that it is plausible that the players will reach an agreement in which the optimal and coordinating service time S_b^* is used. Specifically, suppose that player 1, sells finished units at a price P , and that player 2 purchases raw material at a cost C per unit (other exogenous costs or profits could easily be incorporated into these parameters). Under this arrangement, the *utilities*, or average profits, for the two players are

$$\begin{aligned} u_1 &= \mu P - \mu p - P_1(S_b) \\ u_2 &= \mu p - \mu C - P_2(S_b) \end{aligned} \quad (60)$$

Of course, all else being equal, player 1 would prefer for S_b to be zero, and player 2 for S_b to be as large as possible, i.e., $S_b = \sum_{k=N_1+1}^N T_k$. On the other hand, suppose that the parties consider a contract with a service time S_b' which is different from the globally optimal service time S_b^* . Then either party could (assuming that the optimal solution is known) propose a new contract in which both parties are better off. Specifically, if the original proposal was for compensation p' , a service time S_b' , and a utility u_1' , then we can set a new price p such that the gains from greater efficiency is evenly divided between the parties. Mathematically, we have:

$$\begin{aligned} p\mu - P_2(S_b^*) &= p'\mu - P_2(S_b') + \frac{(P_1(S_b') + P_2(S_b')) - (P_1(S_b^*) + P_2(S_b^*))}{2} \\ p &= p' + \frac{P_1(S_b') + P_2(S_b^*) - P_1(S_b^*) - P_2(S_b')}{2\mu} \end{aligned} \quad (61)$$

Thus we can reasonably believe that the players will be able to reach an agreement that uses the efficient solution S_b^* . The resulting arrangement, in which the players follow

the contract from Figure 4, using the service time S_B^* , has many benefits. We highlight some in the list below:

- B1. *Incentive compatibility.* As shown in (59), once the players have agreed on the service time S_B^* , it will be in their own best interests to operate the supply chain according to the globally optimal solution.
- B2. *Informational decentralizability.* Other than passing on orders equal to customer demand, the players do not need to know inventory levels or anything else about the other player's part of the supply chain.
- B3. *Cost conservation property.* Costs can be traced to individual sites, and there is no need for a central authority to hand out subsidies or tariffs to motivate the players to participate (however, an important caveat to this principle is discussed in §14).
- B4. *Simplicity.* The contract only calls for a simple payment for purchased items, without induced backorder costs, shortage reimbursement, etc. The requirement for player 2 to meet orders within the demand bound within a time period S_B is easy to understand, communicate, and monitor. In fact, all the benefits that a single company using a GS model can be enjoyed by both parties.
- B5. *Arbitrary division of profits.* By adjusting the price p , one can achieve an arbitrary division of profits between the players, and hence, the proposed contract structure can be used both in industries where the upstream players have more power, and in industries where the downstream players have more power.

Benefits B1-B3 in particular were highlighted as desirable properties by Lee and Whang (1999).

Predicting the price p with the Nash bargaining model

Above, we argue that it is reasonable to assume that an agreement to use S_B^* can be reached, because for any other S_B , either player could propose an alternative contract under which both players are better off. Is it possible to predict upon which value of the price p the players will agree?

It may be, if we accept some additional assumptions proposed by Nash (1950) in his work on bargaining models. Specifically, let us assume that the players bargain over p and S_B . One could also include the demand bound D into the bargaining process, but for simplicity we will view it as exogenously specified (reflecting the common situation that many firms do not set their customer service levels based on negotiations over vendor delivery terms).

Considering the general problem of bargaining between two players, Nash posited that the outcome ought to be subject to four *axioms* as follows:

1. Invariance to equivalent utility representations
2. Pareto efficiency
3. Symmetry
4. Independence of irrelevant alternatives

These axioms, and some mild technical conditions (e.g., the set of feasible agreements (p, S_B) should be compact) are explained and discussed in for example Osborne and Rubinstein (1994).

Moreover, we assume that if the players disagree and negotiations break down, they will receive utilities $u_1(d)$ and $u_2(d)$, respectively. Nash showed that the *only* solution (the “Nash solution”) that satisfies all four axioms, is the solution that maximizes the product (the “Nash product”) of the players’ net (over the disagreement outcome) utilities $|u_1 - u_1(d)| \times |u_2 - u_2(d)|$. In our setting we have:

$$\begin{aligned} \max_{p, S_B} (P\mu - P_1(S_B) - p\mu - u_1(d)) \times (p\mu - P_2(S_B) - C\mu - u_2(d)) = \\ \max_p (P\mu - P_1(S_B^*) - p\mu - u_1(d)) \times (p\mu - P_2(S_B^*) - C\mu - u_2(d)) \end{aligned} \quad (62)$$

This is true because we know from (61) that only S_B^* fulfills Pareto efficiency. Now differentiating with respect to p , we have

$$\begin{aligned} 0 = -\mu(p^* \mu - P_2(S_B^*) - C\mu - u_2(d)) + \mu(P\mu - P_1(S_B^*) - p^* \mu - u_1(d)) \\ p^* = \frac{P\mu + P_2(S_B^*) + u_2(d) + C\mu - P_1(S_B^*) - u_1(d)}{2\mu} \end{aligned} \quad (63)$$

Thus, under the axioms proposed by Nash, we can predict not only the service time, but also the price, which the parties ought to agree upon. Since the price per unit is paid from player 1 to player 2, any term that has a positive impact on player 1 or a negative impact

on player 2 will increase the price (a positive sign in (63)) that must be paid. For example, if player 2's raw material cost C per unit would increase, then the price paid by player 1 would increase by half that amount, so as to evenly split the loss in profit. We can model asymmetries in this framework by appropriately setting the disagreement utilities $u_1(d)$ and $u_2(d)$. For example, if player 1 is a powerful oligopolist with many alternative suppliers, he or she may enjoy a high disagreement utility $u_1(d)$. As is apparent from (63), this will lead to a lower per-unit price.

Other forms of asymmetries are not modeled in Nash's framework, and indeed, are ruled out by the postulated axioms (however, extensions are discussed in Muthoo, 1999).

14. Identifying relevant holding costs

In this section, we will discuss in greater detail the calculation of holding costs when optimizing safety stocks within the GS framework. This topic is pertinent for any GS implementation (and presumably, for other supply chain theories as well), but it is particularly relevant in settings where different parts of the supply chain are controlled by different parties.

The basic idea is that when calculating holding costs, one should only account for those costs, which are actually affected by the quantity and location of safety stocks. Examples of relevant costs are cost of capital, risk for obsolescence, per-unit physical storage costs, and so on. Many of these costs, for example the cost of capital and the cost of obsolescence are driven by the value of the part. A part that is twice as dear will require twice as much capital and will cause twice the loss if it suddenly becomes obsolete. On the other hand, costs that are not affected by the location and quantity of safety stocks should not be incorporated into the holding costs. For example, for the purpose of determining safety stocks locations, one should not allocate a portion of fixed costs to each part, since those costs are not affected by the choice of service times and safety stocks locations.

However, consider what happens when a company purchases a part from another company; the price charged by the supplier will surely be large enough to cover not only the supplier's variable costs of production, but also the supplier's fixed costs, overhead, markups, and so on. As a consequence, the holding cost at the buyer is based on not just the variable costs in the supply chain, but also on any mark-up that is charged by the supplier to cover its fixed costs.

In general, suppose that the value of a part is the sum of the all prior value adding activities v_k at the upstream stages, and that the holding cost is proportional (with

parameter α) to the value of the part; $h_k = \alpha \sum_{j=k}^N v_j$. Let us further assume that stage

$N_1 + 1$, may add an additional markup m , so that downstream stages use the holding cost

$\tilde{h}_k = \alpha \left(m + \sum_{j=k}^N v_j \right)$ for $k \leq N_1$. We note that there is no guarantee that the two safety-stock

optimization problems with \tilde{h}_k and h_k , (representing with and without markup) have the same optimal solution.

We explore this issue with a small, illustrative example. In §15 below, we will perform a larger set of experiments to better characterize the typical impact of this phenomenon. For now, consider a supply chain with two stages, where each stage just transports the part from one location to another location, each with a transportation time of 10; we assume a demand bound of $\mu\tau + z\sigma\sqrt{\tau}$ and we will see that the parameter values do not matter for this argument. Suppose that the firm purchases the units for \$1 each, and sells them for \$4 each. If no value-adding activities happen at the two stages, the value of the part at each stage will be \$1, (For the example we assume that there is a zero or negligible value added from the transportation steps.) The optimal solution is to have one safety stock location at the downstream stage, with an average inventory of $z\sigma\sqrt{20}$ and average holding cost of $\alpha \times z\sigma\sqrt{20}$ where α is the holding cost rate.

Suppose now instead that the two stages are operated by different parties, and that the downstream firm buys the product for \$3 per unit from the upstream firm. Now if we allocate the values \$3 and \$1 to the two stages, the optimal solution will instead be to have safety stocks at both locations, for an apparent objective value of

$3\alpha \times z\sigma\sqrt{10} + \alpha \times z\sigma\sqrt{10}$. If we were to use this solution with the original holding costs (without the markup), which are α at each stage, we get the cost $\alpha \times z\sigma\sqrt{10} + \alpha \times z\sigma\sqrt{10} > \alpha \times z\sigma\sqrt{20}$. Thus adding a markup at some stage can lead to a suboptimal solution. Note in particular that this problem persists even if the parties share information and coordinate their activities through a contract such as the one proposed in Figure 4. If player 1 pays a markup to player 2 his relevant cost is \$3, so this is not a matter of truthfully sharing information. An intuitive explanation for what is happening is that the extra holding cost αm caused by the markup, is not a system-wide cost but merely a type of transfer from the downstream stage to the upstream stage. Presumably, the upstream stage could earn more interest if the downstream stage added more safety stock, but this effect is presently not captured in the objective function.

Moreover, the optimal solution will typically be different even when we add the markup m to *all* stages of a system. That is, the program (56) is not insensitive to adding a constant term to all of the holding costs parameter (if instead we multiply by a constant, the optimal solution will be the same). One implication of this is that even if the parties could agree to run an optimization with the “real” costs h_k and agree on using the system-optimal service time S_B^* between them, there is no longer *incentive compatibility*. That is, if the downstream player controls multiple stages, the optimal solution for that part of the supply chain will generally depend on the markup m . Critically, for the service time S_B^* , the downstream player will implement the locally optimal solution, which will depend on the markup m and does not always coincide with the global optimum. Intuitively, the downstream player will seek to reduce the interest on the markup αm , even though this might increase the system-wide cost at the expense of player 2.

We close this section by considering possible contract structures that may realign the players’ incentive structures. Obviously, when the supply chain is controlled by different parties, cutting out the profit margin m of the upstream player will generally not be a feasible alternative. Indeed, we highlighted as a benefit B5 precisely the property that a good contract structure should enable an arbitrary distribution of profits. Instead, we should consider contract structures in which markup or profit payments are made in

such a way that they do not affect holding costs. One specific possibility is to divide payments into two parts, one part $p = \sum_{j=N_1+1}^N v_j$ which is paid when ordering, and a second part m , which is paid upon delivery to the end customer. Note that in this case, the downstream player will then use the “real” holding cost $h_k = \alpha \sum_{j=k}^N v_j$ for the purpose of optimizing safety stocks, since the incremental investment to change the downstream safety stock level is proportional to $p = \sum_{j=N_1+1}^N v_j$ and does not depend at all on the markup m .

Under the base stock policy, the orders placed from player 1 to player 2 are actually the same as those placed by the end customer. Therefore, for this contract structure, the payment in period t will be $d(t) \times \sum_{j=N_1+1}^N v_j$ for the order placed on player 2 plus $d(t) \times m$ for the markup on the delivery to the customer. Hence the total payment is $d(t) \times \left(m + \sum_{j=N_1+1}^N v_j \right)$, which is exactly the same as for the traditional payment scheme.

Thus, it might seem that the proposed payment scheme amounts to nothing. However, this equivalence only applies in steady state conditions. Whenever the downstream player makes an adjustment in its safety stock, it will only pay p for each unit of adjustment. In particular, when the firms initialize the supply chain, the downstream player must place a series of orders to establish its safety stock. Similarly, whenever there is a change in the demand bound or in the production times, the downstream player might adjust its safety stocks, and again make payments at the price p . It is during those transient time windows that the payment on orders must not contain a markup, for the system to be coordinated.

The new contract structure is summarized in Figure 5 below.

1. The demand bound $D(\tau)$, within which the player 2 commits to meet any demand realization
2. The service time S_B , within which the player 2 must meet all orders
3. The price per unit $p = \sum_{j=N_1+1}^N v_j$, which player 1 agrees to pay for each unit upon placing an order
4. The additional price per unit m which player 1 agrees to pay for each unit upon delivery to end customer

Figure 5: A contract for situations when downstream holding costs are affected by a non-value added markup m . We note that during steady-state conditions the separation of payment is of no consequence.

The proposed contract structure retains all the benefits B1-B5 highlighted for the contract outlined in Figure 4, with the possible exception of B4, “simplicity”. While it is possible to realign incentives in a system with non-value added markups, the solution has a more complex payment structure. Fortunately the extra complexity only appears when player 1 is making modifications to its safety stock strategy, such as at the initiation of the supply chain.

Finally, we note that negotiating over the contract in Figure 5 is no different from negotiating over the contract in Figure 4. All else being equal, player 2 will find it disadvantageous to sell units at cost in the initial set-up phase, just like he or she will find it disadvantageous to offer fast service. But all things are not equal; player 2 can demand a higher per-unit price as a compensation for agreeing to these globally beneficial terms.

15. Numerical examples

In order to test the significance of coordination, we performed two sets of numerical experiments. In both cases, we used the same supply chain and cost structures as in Graves and Willems (2006). Specifically, we considered a serial system with $N = 5$ nodes, and with three alternatives for both the cost accumulation and the production lead-time as follows:

Stage	5	4	3	2	1
Increasing	36	28	20	12	4
Constant	20	20	20	20	20
Decreasing	4	12	20	28	36

Table 10: Alternative structures for supply chain lead-time and cost accumulation

The terms “increasing” and “decreasing” should be understood in terms of going upstream starting from the customer facing stage 1. In the case of cost accumulation, the values stated in Table 10 represent the cost added at each stage. For example, for the increasing cost scenario, the cost at stage 5 is 36, the cost at stage 4 is $36 + 28 = 64$, the cost at stage 3 is $36 + 28 + 20 = 84$, etc. For all three scenarios the cost of the finished good at stage 1 is 100.

In a first set of experiments, we tested, for each supply chain structure, what would happen if different parts of the supply chain were controlled by two different parties that did not necessarily agree on the system-optimal service time S_B^* . We investigated cases when the first player controlled the most downstream 1,2,3 or 4 stages. For each case we enforced various values of S_B and tested the quality of the optimal solution. As we have seen earlier, if $S_B = S_B^*$ the solution will coincide with the optimal one, even when the different players control their own parts separately, but for different values of S_B the results will generally not be as good. The purpose of this experiment was to see how big a difference it made when the parties agreed on the “wrong” S_B , and subsequently controlled their respective parts of the supply chain as well as possible given S_B as a hard constraint. This S_B was then varied among all integer values between

0 and $\sum_{j=N_1+1}^N T_j$. In each case, we have listed the average and worst (in terms of all the feasible values of S_B) objective values, as compared to the optimal objective value. The results are listed in Table 11 below.

Performance relative best possible (using S_B^*)		Player 2's most downstream stage ($N_1 + 1$)							
		5		4		3		2	
Cost	Lead Time	Avg.	Worst	Avg.	Worst	Avg.	Worst	Avg.	Worst
Increasing	Increasing	105%	106%	108%	116%	112%	125%	113%	126%
	Constant	105%	106%	111%	119%	117%	131%	122%	137%
	Decreasing	103%	105%	112%	117%	121%	130%	128%	139%
Constant	Increasing	103%	104%	105%	107%	106%	110%	107%	113%
	Constant	102%	104%	104%	107%	107%	115%	111%	124%
	Decreasing	101%	102%	106%	108%	111%	117%	118%	128%
Decreasing	Increasing	102%	103%	104%	109%	111%	124%	122%	149%
	Constant	101%	102%	104%	106%	107%	116%	108%	116%
	Decreasing	100%	100%	101%	102%	103%	106%	108%	117%

Table 11: System performance when S_B is chosen suboptimally and parties optimize their sections separately. For the different scenarios on the left, the average, and worst cases are stated relative to the optimal one.

On average, a randomly chosen S_B resulted in an 8.6% higher costs than S_B^* , and the worst choice of S_B resulted in 15.1% higher costs on average (over the 36 cases). In general, the biggest differences were found when Player 2 controlled a larger part of the supply chain, perhaps because of a larger space of feasible (and potentially poorly performing) S_B .

We also performed a set of experiments to investigate the impact of a superfluous 0-50% mark-up of holding cost by stage 3, and thus incurred in the form of larger holding costs by stages 1 and 2 (in addition to their own value-adding activities). In each case, we calculated the optimal solution for the system with the extra cost, and then used that solution in a system without such a markup. That is, we sought to understand to what

extent adding extra costs distorts the “real” optimal solution. The results of this exercise are in Table 12 below.

Cost relative optimal		Extra markup used by lower two stages				
Cost	Lead time	10%	20%	30%	40%	50%
Increasing	Increasing	101.6%	101.6%	101.6%	101.6%	117.0%
	Constant	100.0%	100.0%	100.0%	100.0%	100.0%
	Decreasing	100.0%	100.0%	100.0%	100.0%	100.0%
Constant	Increasing	100.0%	101.7%	101.7%	101.7%	101.7%
	Constant	100.0%	100.0%	100.0%	100.0%	100.0%
	Decreasing	100.0%	100.0%	100.0%	100.0%	100.0%
Decreasing	Increasing	100.0%	100.0%	100.0%	100.0%	100.0%
	Constant	100.0%	100.0%	101.6%	101.6%	101.6%
	Decreasing	100.0%	100.0%	100.0%	100.0%	100.0%
Average		100.2%	100.4%	100.5%	100.5%	102.3%

Table 12: Actual system costs when lower two stages use a superfluous markup in their holding cost

As we can see, the cost of using the wrong holding cost was generally quite modest. In 6 out of the 9 settings, even a 50% markup caused no suboptimality at all. A single outlying data point is the increasing cost, increasing lead time scenario with a 50% markup. In this particular case, the real optimal solution is to have all the inventory at stage 1, but the 50% markup solution is almost the opposite, with a distributed safety stock strategy with significant inventories upstream.

We also performed a similar set of experiments except with the markup added by stage 2 or stage 4, instead of by stage 3. The results for these cases (which we do not show here), were quite similar. We found that the additional average cost associated with a markup was only 0.94%; however, it was uneven, and in 5 out of the 105 cases the cost was 14% or more.

It is always true that increasing holding costs downstream will encourage more inventory to be held upstream. Conversely, when there the “optimal” solution in a coordination problem is to have most inventory upstream, one might find gains by

reviewing the true causes of the holding costs and considering alternative payment structures, as discussed in §14.

16. Conclusion

In summary, the guaranteed service (GS) framework is well suited for distributed decision-making between parties with competing interests. Specifically, we showed in §12, that provided that the parties can agree on the right service time S_B^* between them, they will manage their own parts of the supply chain in alignment with the globally optimal solution. In Figure 4, we propose a specific contract structure that would formalize such an agreement, and in the subsequent discussion we highlight numerous benefits with this type of arrangement.

Noteworthy features of this contract are that it *makes explicit in advance* what is expected from the upstream player during times of high demand. By integrating per-unit price into the same contract, the upstream player can in effect seek compensation for the holding costs that it must incur in order to provide guaranteed service..

The relative ease by which we coordinate the supply chain stands in contrast to supply chains operating according to echelon-based ordering, which need more complex contracts to be coordinated (Lee and Whang, 1999). However, we did note in §14, that if holding costs are driven by markups or other non-value added activities, the supply chain can be misaligned, and in systems with two players where markups are necessary, a more complex payment structure is necessary to coordinate the supply chain. Fortunately, this payment structure is equivalent to the normal one when the system is in steady state; it is only when setting up or changing safety stocks that this becomes an important issue. our numerical tests show that the cost of disregarding this issue altogether is generally low, but there are exceptions. Specifically, one should be concerned about this when the markup is significant, and the solution (when the markup is included in the downstream holding costs) suggests little inventory downstream in the supply chain. In this situation, employing the modified contract proposed in Figure 5 might shift safety stocks downstream and reduce the overall global holding costs.

The results discussed here are not only valid for serial systems, but generalize to more complex supply chain structures as well. We can easily modify derivation (59) to

show that any connected subset of a supply chain will be incentive compatible with the global optimum, provided that the connection (or all the connections, if there are more than one) have service times that coincide with the global optimum.

A more difficult question is whether the parties can reasonably be expected to share the information necessary to calculate the globally optimal solution and the coordinating service time S_B^* . As we have pointed out, and in fact highlighted as a benefit B2, the proposed contract structure does not call for the parties to share any information on an ongoing basis, other than the order quantities for the base-stock policy, which equal end-customer demand. However, in order to find S_B^* and agree on an efficient contract, one needs to have access to the production or processing times and the holding costs across the supply chain. If the parties only have such information for their own parts of the supply chain, it is not clear whether it is in their own best interests to share that information, or even to truthfully reveal the functions $P_1(S_B)$ and $P_2(S_B)$, which are needed to find the optimal solution. While we show in the Appendix that $P_1(S_B)$ is a concave function, $P_2(S_B)$ and $P_1(S_B) + P_2(S_B)$ will generally have no simple structure. Therefore, a sequence of offers and counter-offers cannot be expected to converge towards the optimal solution. In the experiments we performed, using the wrong value of S_B led to an average of 8.6% too high costs, although there was considerable variation. We leave the challenges of information-sharing for future research.

Appendix: the concavity of $P_1(S_B)$

We can view $P_1(S_B)$ and $P_2(S_B)$ as functions of S_B . We first show that, in serial systems, if D is concave, then so is $P_1(S_B)$

First, when D is concave, then the all-or-nothing property holds (for any value of S_B), and we can enumerate all potential optimal solutions, say with index i . Specifically, we can write a binary string with the length equal to the number of stages, and let a 1 indicate that that stage has inventory, and a 0 that that stage does not have inventory. The number of solutions grows exponentially with the number of stages, but this does not matter for our argument. Now let $j(i)$ be the *last* (most upstream) stage which has inventory in solution i , and let C_i be the total cost of the inventory downstream of stage $j(i)$. Then the total cost of solution i can be written as a function of S_B :

$$C_i + h_{j(i)} \left[D(S_B + \sum_{k=j(i)}^{N_1} T_k) - (S_B + \sum_{k=j(i)}^{N_1} T_k) \mu \right] \quad (\text{A64})$$

Under the assumption that D is concave we note that (A64) is a concave function of S_B .

Now we can express the optimal solution as minimization over all of the enumerated solutions

$$P_1(S_B) = \min_i C_i + h_{j(i)} \left[D(S_B + \sum_{k=j(i)}^{N_1} T_k) - (S_B + \sum_{k=j(i)}^{N_1} T_k) \mu \right] \quad (\text{A65})$$

The minimum of a number of concave functions is concave, and then so is $P_1(S_B)$.

IV. Strategic safety stocks in supply chains with evolving forecasts

We examine the placement of safety stocks in a supply chain for which we have an evolving forecast of demand. Under specific assumptions about the forecasts, the demand process, and the supply chain structure, we show that safety stock placement for such systems is effectively equivalent to the corresponding well-studied problem for systems with stationary demand bounds and base stock policies. Hence, we can use existing algorithms to find the optimal safety stocks. We use a case study with real data to demonstrate that there are significant benefits from the inclusion of the forecast process when determining the optimal safety stocks. We also conduct a computational experiment to explore how the placement and size of the safety stocks depend on the nature of the forecast evolution process.

17. Introduction

Most firms plan their supply chain operations based on a forecast of future demand over some planning horizon. Furthermore, firms regularly update and revise these forecasts based on observed sales, advanced orders, and market intelligence. With each forecast revision, a firm will also revise its supply chain plans, in terms of its master schedules for production, procurement, and transportation. Indeed, this update and revision process is central to any supply-chain planning function and is facilitated by the wide-spread deployment of material requirements planning (MRP) systems.

The intent of this paper is to examine the optimal placement of safety stock inventory in a supply chain that is subject to a dynamic, evolving demand forecast. In particular we strive to develop models and algorithms that have the potential to determine safety stocks in real-world supply chains. We assert that the paper makes five contributions.

First, we incorporate a forecast evolution process into the safety stock placement models developed by Simpson (1958) for a serial-system supply chain, and by Graves and Willems (2000) for supply chains with spanning-tree topologies. In particular, we use

the forecast evolution process and model that has been previously used by Graves, Meal et al. (1986), Heath and Jackson (1994) and Graves et al. (1998).

Second, we show for a serial-system supply chain with an evolving forecast that the optimal placement of safety stocks satisfies the all-or-nothing property: that is, each stage either holds a decoupling safety stock or no safety stock. As one consequence of this property, we can determine the optimal safety stocks by a simple enumeration procedure.

Third, for an assembly supply chain with an evolving forecast, we show that its safety-stock optimization problem has the same structure as the safety-stock optimization for an assembly system operating with a base-stock ordering policy. Graves and Willems (2000) have developed a dynamic programming algorithm to determine the optimal safety stocks for this latter system. Thus, we can use this algorithm to solve the safety-stock optimization problem for assembly systems with an evolving forecast. This equivalence also extends to supply chains with spanning-tree topologies; however, the analysis of the supply chains with an evolving forecast requires a bound function on the forecast process, and we do not have a satisfactory way of specifying this bound function for these more general supply chains.

Fourth, based on an industrial study and on a computational experiment, we demonstrate the potential value from incorporating the forecast evolution process into the safety-stock optimization. We find that substantial reductions in inventory are possible, where the size of the reduction depends on how the forecast improves over time; to no surprise, the better the forecast, the less safety stock is required. However, prior safety-stock optimization methods were not able to extract the value from an improving forecast.

Fifth, we demonstrate that we can use our forecast evolution process to model a wide class of demand processes introduced by Aviv (2003); for instance, this class includes autoregressive integrated moving average (ARIMA) processes. The significance of this result is that all of the developments in the paper also apply to a supply chain whose product demand comes from one of these demand processes. For instance consider an assembly system that is subject to demand from an ARIMA (p, d, q) process for any specification of the parameters (p, d, q) ; we can infer a forecast process for this

supply chain and then use this forecast process to determine the supply-chain safety stocks, using the models and methods developed in this paper.

We organize the paper into seven sections. In the remainder of this section, we provide a brief review of related literature. In §18 we introduce the forecast evolution process and show the equivalence between the forecast evolution process and a class of demand processes introduced by Aviv (2003). In §19 we define the ordering policy for a supply chain with an evolving forecast, and then use this to model the inventory dynamics for a serial-system supply chain. We also establish the safety stock required at each stage to satisfy the guaranteed service constraint. In §20 we establish the all-or-nothing property for the optimal solution for a serial-system supply chain, and then show how to determine the safety stocks for an assembly system. We report on an industrial case study in §21 and on a set of computational experiments in §22. We conclude the paper in §23. We also include an Appendix in which we provide the detailed development for several of the results in the paper.

Literature Review

This paper adds to a rich body of work on multi-echelon supply chain management. For a general overview of this research area, we refer to review articles by Axsäter (1993), Federgruen (1993), Inderfurth (1994), and Diks et al. (1996). This paper contributes to three bodies of work in particular.

First, our model uses a dynamic model for forecast evolution, and is related to other work on forecasting and advanced demand information in supply chains. Our forecast evolution model (§18) is a generalization of the one used by Graves, Meal et al. (1986), Heath and Jackson (1994), and Graves et al. (1998). We show that there is a close relationship between this forecast model and popular time-series demand models, such as ARIMA. Therefore this paper is related to the growing body of work that assumes such demand models in supply chains. In particular, the demand model we use is based on the framework introduced by Aviv (2003). We refer the reader to Zhang (2004) for results and references on supply chain dynamics, Aviv (2004) for an overview of forecasts and

collaboration, and Gallego and Özer (2001) and Karaesmen et al. (2002) for results on the value of advanced demand information in the supply chain.

Second, the underlying supply chain model (§19) and optimization procedure (§20) follow closely the work of Simpson (1958) and Graves and Willems (2000). These authors assume that each stage or node of the supply chain operates under a base-stock policy and that demand is bounded. They then find the least cost service times and inventory placement that are guaranteed to meet any demand realization within these bounds. This approach provides a way to find the optimal strategic safety stocks in quite general supply chains, and it has successfully been deployed to industry (e.g., Billington et al. 2004).

Our work shares many important aspects with this line of work, but one significant difference is that we assume that each stage places orders in response to changes in schedules and forecasts of future demand. The aforementioned work assumes that the stages operate according to (local) base stock policies and place orders in response to realized demand at the customer-facing stages. Because of these differences, the new safety stock strategies are more applicable for firms that already operate in a forecast- or schedule-driven way, and who seek a comprehensive safety stock strategy.

Thirdly, our assumed ordering policy is similar to that for an MRP system; thus, we can relate our work to the research literature on safety stocks in MRP systems. For an overview of MRP literature in general, see Baker (1993). Guide and Srivastava (2000) have reviewed buffering in particular, and list a comprehensive table of various approaches and results.

More specifically, Lambrecht et al. (1984) consider exact solutions for small systems, and heuristics for serial systems. Buzacott and Shanthikumar (1994) analyze and compare safety stocks and safety times in a single-stage system. Yano and Carlson (1987) consider a two-stage system under either fixed or flexible scheduling; the current analysis corresponds most closely to flexible scheduling. Lagodimos and Anderson (1993) consider the maximum service level achievable, for a given safety stock quantity. Mollinder (1997) studies a number of systems using simulation, and finds optimal solutions with simulated annealing.

Most of the aforementioned work is limited to small systems, typically one or two stages. The lack of solutions for larger systems in particular has been highlighted in the overview paper by Guide and Srivastava (2000). Moreover, *none* of the aforementioned authors model dynamically evolving forecasts and non-stationary demand.

We note that the ordering policy in the current paper is a special case of the class of policies considered by Graves et al. (1998), who model a supply chain with a dynamically evolving forecast and with an objective to smooth operations to reduce the variability of production. However, Graves et al. (1998) does not attempt to optimize the supply-chain safety stocks, which is the primary point of the current paper. Similar to Graves et al. (1998), Aviv (2007) develops a model of a two-stage supply chain with a dynamically evolving forecast; he also incorporates production smoothing and schedule/forecast changes into his objective function. But the primary intent of this work is to understand the benefits from collaborative forecasting. In contrast, we assume an ordering policy and accept the resulting variability from the induced schedule changes, and seek only to reduce the safety stock costs across the supply chain.

18. The forecast model

We use a forecast evolution model based on Graves et al. (1986) and Heath and Jackson (1994). In period t we denote the forecast for period $t+i$ as $f_i(t+i)$ for

$i \in \{1, 2, \dots, H\}$ where H is the forecast horizon. By convention we set $f_i(t) = D_t$ where D_t is the demand in period t . We will not make any notational distinction between D_t for future times, which are random variables, and for past times, which are realized scalar values. We assume that in each period t we make an initial forecast for the demand in period $t+H$, that is $f_i(t+H)$; we also assume that each period we revise the nearer-term forecasts, where we define the *forecast revision* as

$$\Delta f_i(t+i) = f_i(t+i) - f_{i-1}(t+i) \text{ for } i \in \{0, 1, \dots, H-1\}.$$

We can express demand as follows

$$D_t = f_{t-H}(t) + \sum_{i=1}^H \Delta f_{t-H+i}(t). \quad (66)$$

We let $\underline{\Delta f}_t$ be the vector of H forecast revisions. We assume that $\underline{\Delta f}_t$ is a random, i.i.d. vector with $E[\Delta f_t(j)] = 0$ for all t and j . With these assumptions Graves et al. (1986), Heath and Jackson (1994) and Graves et al. (1998) have established several properties for this forecast evolution model: $f_t(t+i)$ is a martingale; $f_t(t+i)$ is an unbiased estimate of D_{t+i} ; and the variance of the forecast error $(D_{t+i} - f_t(t+i))$ increases in i . Furthermore, they show that the variance of the random variable D_t is the trace of the covariance matrix for $\underline{\Delta f}_t$, which we denote by Σ .

The prior work assumes that the initial forecast is $f_t(t+H) = \mu$ for all t . Under this assumption the demand process D_t has mean μ and variance given by the trace of Σ . We depart from the earlier work in that we do not make *any* assumptions about $f_t(t+H)$. In particular, we permit $f_t(t+H)$ to be generated by a non-stationary process of arbitrary complexity, or to be user-specified.

Thus we can apply this forecast model to contexts in which the initial forecast $f_t(t+H)$ contains information of future orders or advanced demand information. For instance, consider the planning process used at Teradyne Inc., a manufacturer of semiconductor test equipment with which we have worked (see also Abhyankar and Graves (2001) for more about Teradyne's planning process). For many of its product lines, Teradyne is a make-to-order operation. But the supply chain lead-time (the longest procurement time for a piece part plus the internal assembly and test lead-times) exceeds the customer lead-time (the delivery lead-time requested by customers). Hence, Teradyne must plan much of its procurement and upstream production activities prior to receiving an order. Teradyne does this by means of a master production schedule (MPS) that covers a planning horizon that corresponds to the length of the supply chain lead-time. In effect this MPS is its demand forecast. At any point in the time, the master schedule consists of a mix of open orders, identified orders and booked orders. An open order corresponds to a traditional forecast of what the sales force plans to sell, an identified order is associated with a potential customer and is based on some preliminary discussions with the customer, and a booked order is a firm customer order. As time moves forward, an open order gets converted into an identified order as the sales force obtains tentative

commitments and product specifications from a customer. Similarly, an identified order gets converted into a booked order once (and if) the product specifications and due date become a firm order.

From our experiences, this process is descriptive of many other make-to-order companies as well. In these cases, the initial forecast is based on the progress at identifying customers and in securing advanced orders. Subsequently, the forecast revisions correspond to changes of the master schedule, which reflect the success at converting the forecast (open orders) into demand (booked orders).

Relationship with demand models

In the previous section we defined a forecast process, and discussed how the framework arises in practice. Moreover, since $D_t = f_t(t)$, defining a forecast process also gives us a demand process; that is, the specification of the covariance matrix Σ and the initial forecast $f_t(t + H)$ determines a demand process D_t .

In this section, we start with a demand model D_t and show that we can infer a forecast process by setting the forecast to the expected value of demand. That is, for a given demand model D_t , we set¹

$$f_t(t + s) \equiv E[D_{t+s} \mid D_t, D_{t-1}, D_{t-2}, \dots], \text{ and}$$

$$\Delta f_t(t + s) \equiv E[D_{t+s} \mid D_t, D_{t-1}, D_{t-2}, \dots] - E[D_{t+s} \mid D_{t-1}, D_{t-2}, \dots].$$

A question of interest is whether the forecast revisions generated in this way are i.i.d. and have zero mean, since these assumptions were made in the forecast evolution model, and in fact are necessary for the supply chain work to follow.

We find that these i.i.d. and zero mean properties hold quite generally. In particular, suppose that we model demand by the general state space framework proposed by Aviv (2003):

¹ We remind the reader that we use D_t to denote both the random variable for future demand as well as the realized history for past demand.

$$\begin{aligned}
X_t &= FX_{t-1} + V_t \\
\Psi_t &= HX_t \\
D_t &= \mu + R\Psi_t
\end{aligned}
\tag{67}$$

where X_t is the state vector (with a dimension that depends on the complexity of the demand model), Ψ_t the vector of observations, F , H and R constant matrices, and V_t an i.i.d., multivariate random variable with zero mean. Demand can in general also be a vector (if there are multiple demand streams), but presently we will consider the case where demand is scalar and R is a row vector. Assume further that the system is *observable*, which loosely speaking means that the system state X_t can be inferred from the observations Ψ_t , or more specifically for the model (67) that $E[X_t | \Psi_t] = X_t$. Then, we show in the Appendix that

$$\begin{aligned}
\Delta f_t(t+s) &= E[D_{t+s} | \Psi_t, \Psi_{t-1}, \dots] - E[D_{t+s} | \Psi_{t-1}, \Psi_{t-2}, \dots] \\
&= RHF^{s+1}V_t
\end{aligned}
\tag{68}$$

Since V_t is i.i.d. with zero mean, so is $\Delta f_t(t+s)$ and thus, Δf_t is i.i.d. with zero mean. Given the covariance matrix for V_t , we easily find from (68) the covariance matrix for the forecast revision Δf_t . We note that most common time series models of demand, including ARIMA models, can be written in this state space framework, and are observable. Thus we find that even demand models that are quite complex and non-stationary often have i.i.d. forecast revisions. Also we will see that whereas the initial forecast $f_t(t+H) = E[D_{t+H} | D_t, D_{t-1}, D_{t-2}, \dots]$ might be quite complex, this has no bearing on our safety stock analysis.

This equivalence between the forecast evolution model and this broad class of demand models means that we can apply our results, for example, in make-to-stock supply chain with a time-series demand model. We refer the reader to the specialized literature (for example Hamilton 1994) for how to estimate a time series demand model based on historical data. Once this is done one can use (68) to find the properties of $\Delta f_t(t+s)$, which are needed for the safety stock optimization to follow.

19. Supply chain model and ordering policy

We assume that the supply chain consists of a number of stages. For convenience, we will derive most of our results for serial systems; we discuss the extension to assembly systems and to spanning-tree topologies in the optimization section. For a serial system, we index the nodes by k and we designate the customer-facing stage as node 1, and node N as the most upstream stage. A stage might represent the procurement of a raw material, or the production of a component, or the manufacture of a subassembly, or the assembly and test of a finished good, or the transportation of a finished product from a distribution center to a warehouse. Each stage k is a potential location for holding a safety-stock inventory of the item processed at the stage.

For each stage, we assume a known deterministic production lead-time, denoted as T_k . When a stage reorders, the production lead-time is the time from when all of the inputs are available until production is completed and available to serve demand. The production lead-time includes the waiting and processing time at the stage, plus any transportation time to put the item into inventory. We assume that there is no capacity constraint and thus, the lead-time is not affected by the size of the order.

We assume that each stage places an order in each period, and that each node provides a *guaranteed service time* S_k ; that is, an order received from the downstream node (or from the customer, for $k = 1$) must always be met within this time.

Thus if node k places an order on node $k + 1$ at time t , then the order will be met by node $k + 1$ at time $t + S_{k+1}$, at which time node k can use it as input into its process.

This order will be available as processed inventory at node k at time $t + T_k + S_{k+1}$.

Similarly, an order received by node k from node $k - 1$ at time t must be delivered at time $t + S_k$. We designate the difference between these times

$$T_k + S_{k+1} - S_k = \tau_k \quad (69)$$

to be the *net replenishment time*, which we constrain to be non-negative. We regard the service times to be *decision variables*, except for the pre-determined boundary conditions $S_1 = S_{N+1} = 0$. We do this particular assignment without loss of generality, because we

can model a nonzero S_1 (customer lead-time) by shifting the forecast process (in particular, the forecast is perfect over the time between customer order and delivery). We can add a nonzero S_{N+1} into the production lead-time T_N of node N .

The assumptions to this point are identical to those made by Simpson (1958) and Graves and Willems (2000). These authors assume that each stage uses a base-stock ordering policy; that is, in each period each stage observes and orders the customer demand:

$$P_k^{\text{Base stock}}(t) = f_i(t) = D_i \quad \forall k \quad (70)$$

By contrast, in this paper we assume that each stage places an order based on the forecast of future demand. Specifically, we define

$$L_k = S_{k+1} + \sum_{j=1}^k T_j = \sum_{j=1}^k \tau_j \quad (71)$$

to be the cumulative lead-time for node k . This represents the shortest time for an order on stage k to reach the final stage and become available to meet customer demand. Given the cumulative lead-time L_k we denote the order placed by stage k at time t as $P_k(t)$ and specify it as follows:

$$P_k(t) = f_i(t + L_k) + \sum_{i=0}^{L_k-1} \Delta f_i(t + i). \quad (72)$$

We note that this ordering mechanism assumes that in each period the forecast is shared among all the nodes. We will term (72) to be the *forecast-based ordering policy*. Intuitively, if forecasts were perfect ($\Delta f_i = 0$) then $P_k(t) = f_i(t + L_k)$; in each period each node of the supply chain places an order so as to push forward exactly what is necessary to meet customer demand in the future, and there is no need for safety stocks. We can view the base-stock policy as a special case of the forecast-based ordering policy (72) in which the initial forecast $f_i(t + H) = \mu$, there are no forecast revisions until the period of demand realization, and then $\Delta f_i(t) = f_i(t) - f_{i-1}(t) = D_i - \mu$. With these assumptions, it is easy to show that for each stage (72) reduces to the base-stock policy $P_k(t) = D_i$.

An alternate characterization of (72) is that each node k in each period t places an order so as to keep the expected inventory at node k at time $t + T_k + S_{k+1}$ constant (we prove this in the Appendix). We also note that the forecast-based ordering policy (72) is the multi-node extension of the installation-based order-up-to policy from Aviv (2003).

Moreover, the forecast-based ordering policy is analogous to what one might expect in practice, as it represents the orders that would be generated by applying MRP logic to a serial system with no lot-sizing and no yield uncertainties. In particular, if we denote node k 's on-hand inventory at the end of time t with $I_k(t)$, we show in the Appendix that we can write (72) recursively as

$$\begin{aligned}
 P_0(t) &= D_t \\
 P_k(t) &= \underbrace{\sum_{i=1}^{T_k+S_{k+1}} E_t [P_{k-1}(t+i-S_k)]}_{\text{scheduled downstream demand}} - \underbrace{\sum_{i=1}^{T_k+S_{k+1}-1} P_k(t-i)}_{\text{inventory on order}} - \underbrace{I_k(t)}_{\text{inventory on hand}} + \underbrace{I_k^0}_{\text{desired safety stock}} \quad (73)
 \end{aligned}$$

The first term on the right hand side represents the order schedule that node k needs to fulfill over its replenishment lead time. The second term represents what is currently on order to node k , namely the inbound orders from node $k+1$ and the orders currently in process at node k . The third term is inventory on hand. The final term I_k^0 is a constant safety stock target, which is set to maintain an inventory buffer for the eventuality of higher than expected demand. Thus, from (73) we see that the order placed by stage k in time t equals the forecast of requirements on stage k over its replenishment lead-time, net of the inventory that it will have available over this time period, plus the safety stock target.

In some simple settings, we can show that the forecast-based ordering policy is optimal with respect to certain criteria. As noted, the policy is optimal when the forecasts are perfect, since the inventory variation is zero in this case, and no safety buffers are needed. Moreover, Aviv (2003) shows by induction that each stage should follow such a policy in order to deliver on orders received and minimize a quadratic cost function. These results all assume that the service times are zero; the present contribution is to consider non-zero service times in a global optimization problem.

Inventory dynamics

For the given assumption of a forecast-based ordering policy, we now proceed to investigate the dynamics of the inventories $I_k(t)$. For guaranteed service times, we have the inventory balance equations:

$$I_k(t+1) = I_k(t) - P_{k-1}(t+1 - S_k) + P_k(t+1 - S_{k+1} - T_k) \quad (74)$$

We show in the Appendix that by combining (72) and (74), we have:

$$I_k(t + T_k + S_{k+1}) = I_k^0 - \sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k} \Delta f_i(j), \quad (75)$$

where we choose the time $t + T_k + S_{k+1}$ on the left hand side for ease of exposition, and I_k^0 is the target safety stock. The expression (75) shows that the current inventory level is a function of recent forecast revisions. Indeed, to get some insight on the required safety stock, we can re-express the forecast revision summation in terms of the forecast errors:

$$\begin{aligned} \sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) &= \sum_{i=t+1}^{t+L_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) - \sum_{i=t+\tau_k+1}^{t+L_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) \\ &= \sum_{j=t+1}^{t+L_k} \sum_{i=t+1}^j \Delta f_i(j) - \sum_{j=t+\tau_k+1}^{t+L_k} \sum_{i=t+\tau_k+1}^j \Delta f_i(j) \\ &= \sum_{j=t+1}^{t+L_k} (D_j - f_t(j)) - \sum_{j=t+\tau_k+1}^{t+L_k} (D_j - f_{t+\tau_k}(j)) \end{aligned} \quad (76)$$

The first term on the right-hand side of (76) is the cumulative forecast error for the forecast made at time t for the next L_k periods. The second term is the cumulative forecast error for the forecast made at time $t + \tau_k$ for the next $L_{k-1} = L_k - \tau_k$ periods. Thus, from (76) we need set the safety stock target to cover the forecast revisions that are made to the L_k -period cumulative forecast over the next $\tau_k = L_k - L_{k-1}$ periods.

We now assume that we have a bound $B(L_{k-1}, L_k)$ on the forecast revisions to the L_k -period cumulative forecast over the next $\tau_k = L_k - L_{k-1}$ periods. That is, we identify $B(L_{k-1}, L_k)$ such that

$$\sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) \leq B(L_{k-1}, L_k) \quad \forall t. \quad (77)$$

If we set

$$I_k^0 \leftarrow B(L_{k-1}, L_k), \quad (78)$$

then it is clear from (75) that the inventory is non-negative, and thus we fulfill the guaranteed service constraint.

A natural question is how to determine the bound function. We might obtain this bound based on historical data; if we have enough observations of the forecast revisions, then we can develop an empirical distribution for the left-hand side of (77) and use this to determine bounds for setting the safety stocks.

An alternate way to obtain the bound is to suppose that management specifies that the safety stock is to protect against some maximum level of forecast error. (For a discussion and justification of this perspective, see Simpson, 1958 and Graves and Willems, 2000) In particular, suppose we can measure the standard deviation of the cumulative forecast error for each possible cumulative lead-time. Then, for the purposes of setting the safety stocks, we might set the maximum forecast error analogous to a service level bound $F(L)$ on the cumulative forecast error:

$$F(L) = z\sigma \left(\sum_{j=t+1}^{t+L} (D_j - f_i(j)) \right) \quad (79)$$

where z is a safety factor and $\sigma(\)$ denotes the standard deviation. Thus, we want the safety stock to provide 100% protection as long as the forecast errors are within $F(L)$ for all lead-times L .

With this specification, we show in the Appendix that the bound function is simply

$$B(L_{k-1}, L_k) = z\sigma \left(\sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) \right) = \sqrt{F^2(L_k) - F^2(L_{k-1})}. \quad (80)$$

Hence, if we are given the maximum allowable level of forecast errors (79) for each possible L , we can then determine the bound function (80); from the bound function we can determine the safety stock level (78) that is necessary to assure the guaranteed service for all forecast/demand realizations within the maximum forecast errors.

We note that (80) is a fairly simple and workable form. We just need to characterize the variance of the cumulative forecast error over all relevant time horizons. From this function, we can directly compute the bound function as given by (80). In the next section we show how we use this bound function to choose the optimal service times S_k (and consequently L_k) to minimize the total inventory costs.

20. Optimization

Given the bound function on the forecast revisions, we can formulate the optimization problem. The objective is to minimize the expected inventory holding costs in the system. From (75) we observe that the expected inventory level at each stage is given by:

$$E[I_k] = I_k^0.$$

We assume that we set the safety stock target I_k^0 according to (78), and thus:

$$E[I_k] = I_k^0 = B(L_{k-1}, L_k).$$

Finally, we assume that each stage k incurs holding costs at a rate h_k proportional to the average inventory level I_k^0 . In addition to this safety stock, the supply chain has pipeline inventory, which is directly proportional to the production lead-times. We do not consider pipeline inventory in the optimization as this inventory does not depend at all on the choice of service times.

By changing the service times S_k we can find different safety stock configurations; we seek the least cost solution. The optimization problem for a serial supply chain is:

$$\begin{aligned} \min_{S_k} \quad & \sum_{k=1}^N h_k B(L_{k-1}, L_k) \\ \text{s.t.} \quad & S_{k+1} + T_k \geq S_k \quad \forall k \\ & L_k = S_{k+1} + \sum_{j=1}^k T_j \quad \forall k \\ & S_k \geq 0 \quad \forall k \quad S_1, S_{N+1}, L_0 = 0 \end{aligned} \tag{81}$$

The first set of constraints assures that the net replenishment time is non-negative for each stage; the second set of constraints defines the cumulative lead-times for each stage.

Simpson (1958) posed and analyzed a similar problem for a serial system operating with a base-stock policy. He assumes that for any time interval $(0, \tau]$ there is a bound on the demand given by:

$$B(\tau) = \mu \times \tau + z\sigma\sqrt{\tau}$$

where μ is the average demand rate and σ corresponds to the standard deviation of demand. We can interpret his assumptions and his analysis as a special case of optimization problem (16) in which the bound function for each stage k is given by

$$B(L_{k-1}, L_k) = z\sigma\sqrt{L_k - L_{k-1}} = z\sigma\sqrt{T_k + S_{k+1} - S_k} \quad \forall k \quad (82)$$

Thus, for the base-stock policy, the objective function is a sum of terms, each of which is concave in the service times. As a consequence, the solution is found on the corners of the solution space, which implies “all-or-nothing” solutions: either a node keeps no safety stock ($S_{k+1} + T_k = S_k$), or it keeps so much safety stock that it is decoupled from the downstream supply chain ($S_k = 0$). It also means that we can find the optimal solution for a serial system through enumeration.

Now suppose we assume an evolving forecast and the forecast-based ordering policy, with the bound given by (80). We will demonstrate that the optimization (81) for the general case is no more difficult than that solved by Simpson for the special case of a base-stock policy. For ease of notation, we define

$$g(L) = F^2(L) = z^2 \text{var} \left[\sum_{j=t+1}^{t+L} (D_j - f_t(j)) \right]. \quad (83)$$

We can re-write the optimization problem:

$$\begin{aligned} \min_{S_k} & \sum_{k=1}^N h_k \sqrt{g(L_k) - g(L_{k-1})} \\ \text{s.t.} & \quad S_{k+1} + T_k \geq S_k \quad \forall k \\ & \quad L_k = S_{k+1} + \sum_{j=1}^k T_j \quad \forall k \\ & \quad S_k \geq 0 \quad \forall k \quad S_1, S_{N+1}, L_0 = 0 \end{aligned}$$

Without loss of generality we add $\sum_{j=1}^{k-1} T_j$ (a constant) to both sides of the first set of constraints and to the non-negativity constraints:

$$\begin{aligned}
& \min_{S_k} \sum_{k=1}^N h_k \sqrt{g(L_k) - g(L_{k-1})} \\
& \text{s.t. } S_{k+1} + T_k + \sum_{j=1}^{k-1} T_j \geq S_k + \sum_{j=1}^{k-1} T_j \quad \forall k \\
& L_k = S_{k+1} + \sum_{j=1}^k T_j \quad \forall k \\
& S_k + \sum_{j=1}^{k-1} T_j \geq \sum_{j=1}^{k-1} T_j \quad \forall k \\
& S_1, S_{N+1}, L_0 = 0
\end{aligned}$$

Now we can rewrite everything, including the decision variable, in terms of L_k

$$\begin{aligned}
& \min_{L_k} \sum_{k=1}^N h_k \sqrt{g(L_k) - g(L_{k-1})} \\
& \text{s.t. } L_k \geq L_{k-1} \quad \forall k \\
& L_k \geq \sum_{j=1}^k T_j \quad \forall k \\
& L_0 = 0
\end{aligned}$$

At this point, we need to make a mild technical assumption: that the variance of the cumulative forecast error $g(L)$ is strictly increasing in L . Then, we can apply $g(\cdot)$ to both sides of the constraint equations:

$$\begin{aligned}
& \min_{L_k} \sum_{k=1}^N h_k \sqrt{g(L_k) - g(L_{k-1})} \\
& \text{s.t. } g(L_k) \geq g(L_{k-1}) \quad \forall k \\
& g(L_k) \geq g\left(\sum_{j=1}^k T_j\right) \quad \forall k \\
& g(L_0) = 0
\end{aligned}$$

Finally, we define $Z_k = g(L_k)$, and use this as a scalar decision variable. We can do this, because by assumption, $g(\cdot)$ is strictly increasing, and hence it is a bijective (one-to-one) mapping. Every solution in terms of Z_k corresponds to a unique solution in terms of L_k

and S_k , and the feasibility and objective value are unaffected by the mapping (note that

$g(\sum_{j=1}^k T_j)$ is a constant). The final program is:

$$\begin{aligned}
 & \min_{Z_k} \sum_{k=1}^N h_k \sqrt{Z_k - Z_{k-1}} \\
 & \text{s.t. } Z_k \geq Z_{k-1} \quad \forall k \\
 & \quad Z_k \geq g\left(\sum_{j=1}^k T_j\right) \quad \forall k \\
 & \quad Z_0 = 0
 \end{aligned} \tag{84}$$

This program has the same concavity properties as found by Simpson, which implies that the optimal solution is found on a corner of the feasible region: for each stage

either $Z_k = g(\sum_{j=1}^k T_j)$ or $Z_k = Z_{k-1}$, corresponding to $S_{k+1} = 0$ or $S_{k+1} + T - S_k = 0$,

respectively. Thus, *the all-or-nothing property of the optimal solution still holds when there is an evolving forecast and the forecast-based ordering policy*. Moreover, we can find the solution by enumeration, although, as we will see next, faster dynamic programming methods are available as well.

Assembly system supply-chain topologies

Graves and Willems (2000) introduce a dynamic programming algorithm to solve the safety-stock optimization problem for systems with base-stock ordering. This approach is not only faster than Simpson's, but it can also be used on supply chains with more general spanning-tree topology. Moreover, it does not rely on concavity properties of the bound function $B(L_{k-1}, L_k)$.

We can extend Graves and Willems' algorithm to supply chains with spanning-tree topology and with forecast-based ordering policy. We first show how to do this for an assembly system; this case is simpler, as we have only one end item and thus one bound function, which applies to all nodes. We will then briefly describe the development required for more general settings in which we have multiple end items, each with a bound function.

For the assembly system, we let $k = 1$ be the customer-facing node as before, and introduce the function $a(k)$ to denote the node that is immediately downstream (after) of k . We set $a(1) = 0$. Each node can now have multiple upstream supply nodes. Since processing at a node cannot start until material from *all* of its supply nodes is available, we define the inbound service time SI_k as the longest service time from the set of supply nodes:

$$SI_k = \max_{\{j: a(j)=k\}} S_j \quad \forall k \quad (85)$$

We then can define the cumulative lead-time for each stage by the recursion:

$$\begin{aligned} L_0 &= 0 \\ L_k &= SI_k + T_k - S_k + L_{a(k)} \end{aligned}$$

Given the cumulative lead-time L_k , we assume that in each period t each node k places an order on its supply nodes for delivery at time $t + SI_k$ with the forecast-based ordering policy, namely:

$$P_k(t) = f_t(t + L_k) + \sum_{i=0}^{L_k-1} \Delta f_t(t + i).$$

Analogous to (73), we can re-express the ordering policy in the following form:

$$\begin{aligned} P_0(t) &= D_t \\ P_k(t) &= \underbrace{\sum_{i=1}^{T_k+SI_k} E_t [P_j(t+i-S_k)]}_{\text{scheduled downstream demand}} - \underbrace{\sum_{i=1}^{T_k+SI_k-1} P_k(t-i)}_{\text{inventory on order}} - \underbrace{I_k(t)}_{\text{inventory on hand}} + \underbrace{I_k^0}_{\text{desired safety stock}} \end{aligned} \quad (86)$$

where $j = a(k)$. (For any supply node i for which $S_i < SI_k$ and $k = a(i)$, we delay each order from node k by $SI_k - S_i$ periods so as to avoid early delivery and excess inventory.)

As in Graves and Willems, the inventory dynamics at each node k depend on the node's outbound and inbound service times, namely S_k, SI_k . By following the same development as for the serial system we can express the expected inventory at each stage k as:

$$E[I_k] = I_k^0 = B(L_{a(k)}, L_k)$$

We can then write the optimization problem:

$$\begin{aligned}
& \min_{S_k, SI_k} \sum_{k=1}^N h_k B(L_{a(k)}, L_k) \\
& s.t. \quad SI_k + T_k \geq S_k \quad \forall k \\
& \quad \quad L_k = L_{a(k)} + SI_k + T_k - S_k \quad \forall k \\
& \quad \quad SI_k \geq S_j \quad \forall j, k = a(j) \\
& \quad \quad S_1, L_0 = 0 \\
& \quad \quad SI_k, S_k \geq 0 \quad \forall k
\end{aligned} \tag{87}$$

As for the optimization for a serial system, the first set of constraints assures that the net replenishment time is non-negative and the second set specifies the cumulative lead-time; for the assembly systems we need add a third set of constraints to relate the inbound service time for each stage to the outbound service times for the adjacent upstream stages.

The optimization problem (87) is the very same problem as solved by Graves and Willems' algorithm for base-stock system, except that we now have a different bound function. Graves and Willems use a demand bound in their objective function, whereas here we use the bound function on the forecast revisions, given by (77); that is, we set the bound function (with modifications for the assembly system) as

$$\sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) \leq B(L_{a(k)}, L_k) \quad \forall t \tag{88}$$

where the net replenishment time is $\tau_k = L_k - L_{a(k)}$. Since Graves and Willems' dynamic programming algorithm does not rely on any special properties of the bound function, we can solve (87) by their algorithm.

Graves and Willems' approach applies to not just assembly systems but to more general spanning-tree topologies as well. For supply chains with evolving forecasts, we can in principle apply the same methods to find the optimal safety stock placements, provided that one can construct the bound function for each stage. For serial systems and for assembly systems we only have one end item and one evolving forecast; we thus only need to develop one bound function, given by (77), which applies to all stages in the supply chain. However, with general spanning-tree networks with multiple end items, we can have upstream stages that supply more than one end item. Each end item has a forecast process for which we would need to specify a bound on the forecast revisions,

similar to (77). The forecast process for upstream stages can now be a combination of the forecast processes for multiple end items; however, it is not clear how best to create a bound function on the forecast revisions for the combined forecast processes. This is a critical step in extending the model to these types of supply chains, as the bound function determines the safety stock requirements. Of course, one can use the simple sum of the bounds on the individual forecast revisions as a conservative bound. However, we leave the development of more economical bounds for future research.

21. Case study

In order to test the results from the previous sections, we performed a case study of the supply chain for an electronic testing system manufactured by Teradyne, Inc. At the time Teradyne had large safety stocks and a high service level, but was looking at ways to reduce inventories. It was thus a good match for our research. This test case also allowed us to develop some intuition for how the new method manifests itself in terms of the locations and quantities of safety stocks. To do these things, we implemented Graves and Willems' dynamic programming algorithm in the PERL programming language, after modifying it with the new bound function.

The supply chain produces a family of semiconductor test equipment. The actual product that is sold to a customer is customized to meet the requirements of the customer's application. This customization is accomplished by the selection of options from a large set of alternatives, where there is an electronic subassembly for each option. Nevertheless, except for this choice of options, the rest of the product is standard for all customers. For our test, we consider only the standard bill-of-material and the corresponding supply chain, which is common for all products. This supply chain entails 3,866 stages or nodes and a single end item, where each node represents one specific part at one specific location. The supply chain extends over multiple locations. Many of the production steps are not done by Teradyne, but by subcontractors. Because of close cooperation and strong relationships with the suppliers, Teradyne has considerable influence over safety stocks at their locations as well. We used real data from the bill of material to characterize the different parts and locations. We assumed that the holding cost was directly proportional to the value of the parts, which were already calculated by Teradyne. We plot the supply chain topology in Figure 6.

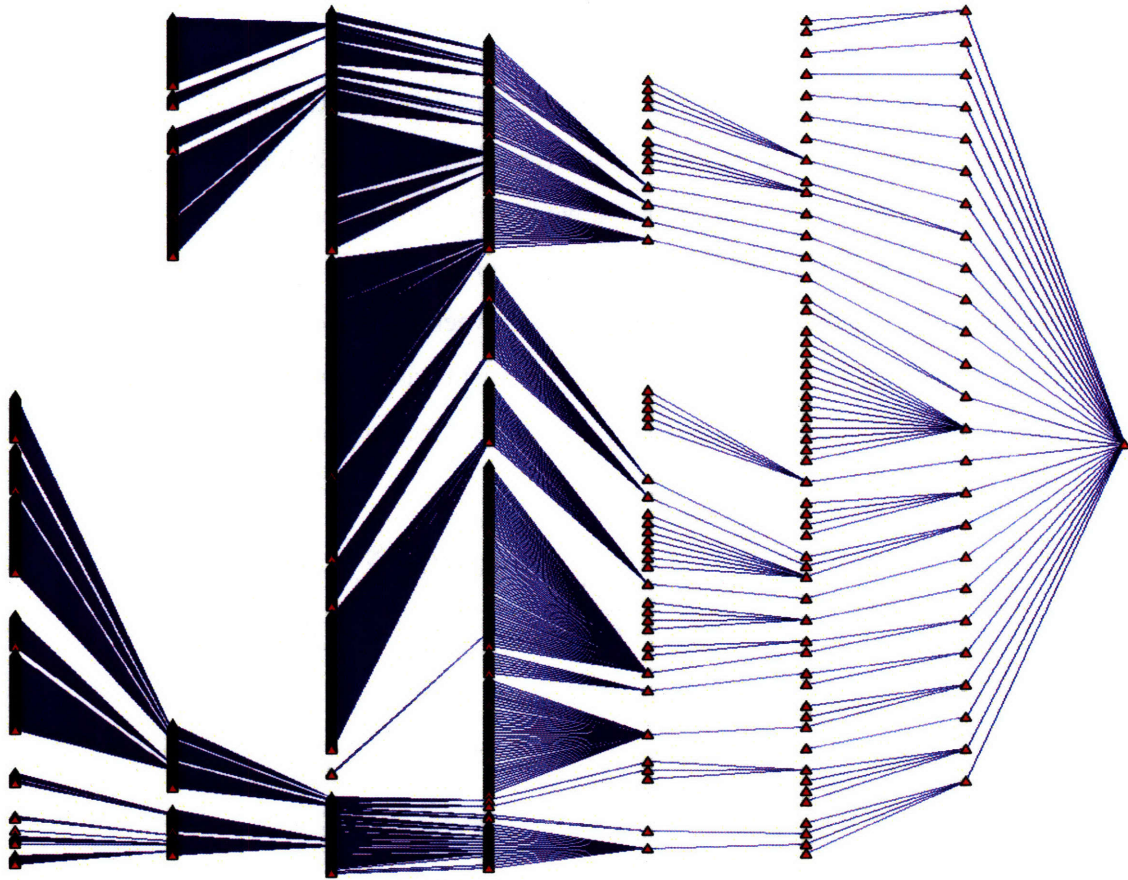


Figure 6: Schematic view of supply chain for the studied product.

Teradyne forecasts the demand for future weeks in a master schedule. Orders are first entered as open or “preliminary” orders, representing perhaps an early discussion with an interested customer or a sales target. Eventually the customer has to commit and the order becomes booked. In this way, the master schedule can be seen as a forecast. The upcoming few weeks are quite accurate (booked orders cannot be cancelled, and new orders are usually not allowed), whereas further into the future the schedule is bound to undergo more changes and hence it is less reliable.

We collected data on schedules and their revisions for one year, and compared the schedules with actual demand. For each week, we had data for the forecasts that were made for a sixteen-week horizon into the future. In total we had about 50 observations for each of the sixteen forecasts in the forecast horizon; that is, we had fifty observations

for the one-week ahead forecast, for the two-week ahead forecast, up to the sixteen-week forecast.

As shown in Figure 7, we measured the correlation between each forecast and the demand and found that this correlation decreased approximately linearly over the upcoming ten weeks. Beyond ten weeks, we found that there was effectively no correlation, which implies that the forecast had no predictive power. In the subsequent experiments we use the linear fit for the first ten weeks, and then assume zero correlation beyond that. Under the assumption of i.i.d. forecast revisions we find that the forecast correlation is equivalent to the standard deviation of the forecast, normalized with respect to the standard deviation of demand:

$$\begin{aligned} \rho(D_t, f_i(t)) &= \frac{\text{cov}(D_t, f_i(t))}{\sigma(D_t)\sigma(f_i(t))} = \frac{\text{cov}(f_i(t) + \sum_{j=i+1}^t \Delta f_j(t), f_i(t))}{\sigma(D_t)\sigma(f_i(t))} \\ &= \frac{\text{cov}(f_i(t), f_i(t)) + \text{cov}(\sum_{j=i+1}^t \Delta f_j(t), f_i(t))}{\sigma(D_t)\sigma(f_i(t))} = \frac{\sigma^2(f_i(t)) + 0}{\sigma(D_t)\sigma(f_i(t))} = \frac{\sigma(f_i(t))}{\sigma(D_t)} \end{aligned} \quad (89)$$

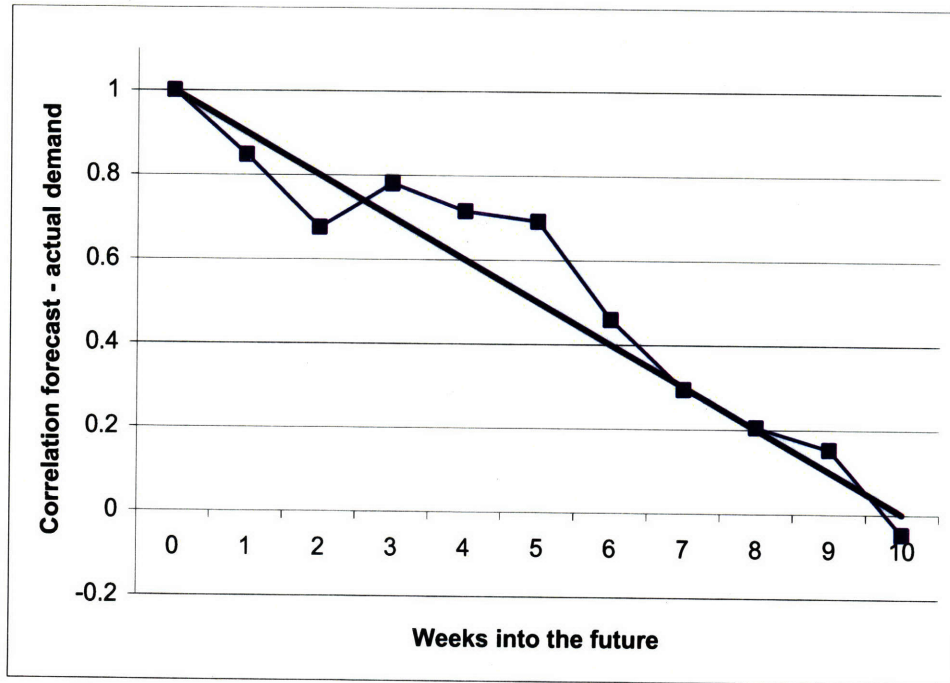


Figure 7: Forecast quality (correlation with what was actually produced) as a function of time into the future, for an electronic test system.

Also we can relate this forecast quality measure to the variance of the forecast error, which we use to calculate the bound function in equation (80); we show in the Appendix that:

$$\text{var}(D_t - f_i(t)) = (1 - \rho^2(D_t, f_i(t))) \text{var}(D_t) \quad (90)$$

To make the initial test simple, we assume that $\Delta f_i(t + j)$ are independent for different j (in addition to the weaker assumption of independence over t , which we make throughout the paper). With this additional assumption, and equations (90) and (82), we have (see Appendix):

$$\begin{aligned} B(L_{a(k)}, L_k) &= z\sigma(D_t) \sqrt{L_k - L_{a(k)} - \sum_{j=t+L_{a(k)}+1}^{t+L_k} \rho^2(D_j, f_i(j))} \\ &= z\sigma(D_t) \sqrt{T_k + SI_k - S_k - \sum_{j=t+L_{a(k)}+1}^{t+L_k} \rho^2(D_j, f_i(j))} \end{aligned} \quad (91)$$

This is similar to the expression for the base stock problem (82) with $SI_k = S_{k+1}$, but now the square root term has an additional term. In particular, we reduce the net replenishment time for node k ($\tau_k = L_k - L_{a(k)}$) by a measure of the forecast quality over the time window $(t + L_{a(k)}, t + L_k]$.

We note that in both cases the bound is proportional to the term $z\sigma(D_t)$; this can be seen as a constant with which the objective function is multiplied, but which does not affect the optimal solution or the relative performance between the base stock policy and the forecast-based policy. Making the arbitrary assignment $z\sigma(D_t) \leftarrow 1$, we compared the bound functions for the base stock ordering policy versus the forecast-based ordering policy, using the straight regression line from the correlation terms measured at Teradyne and illustrated in Figure 7. In Figure 8 we plot the bound $B(0, L)$ for both the base-stock system (82) and for the forecast system (91). Since we assume $S_1 = 0$, the bound function $B(0, L)$ equals the required safety stock for node $k = 1$ if its net replenishment time were $L = T_1 + SI_1$.

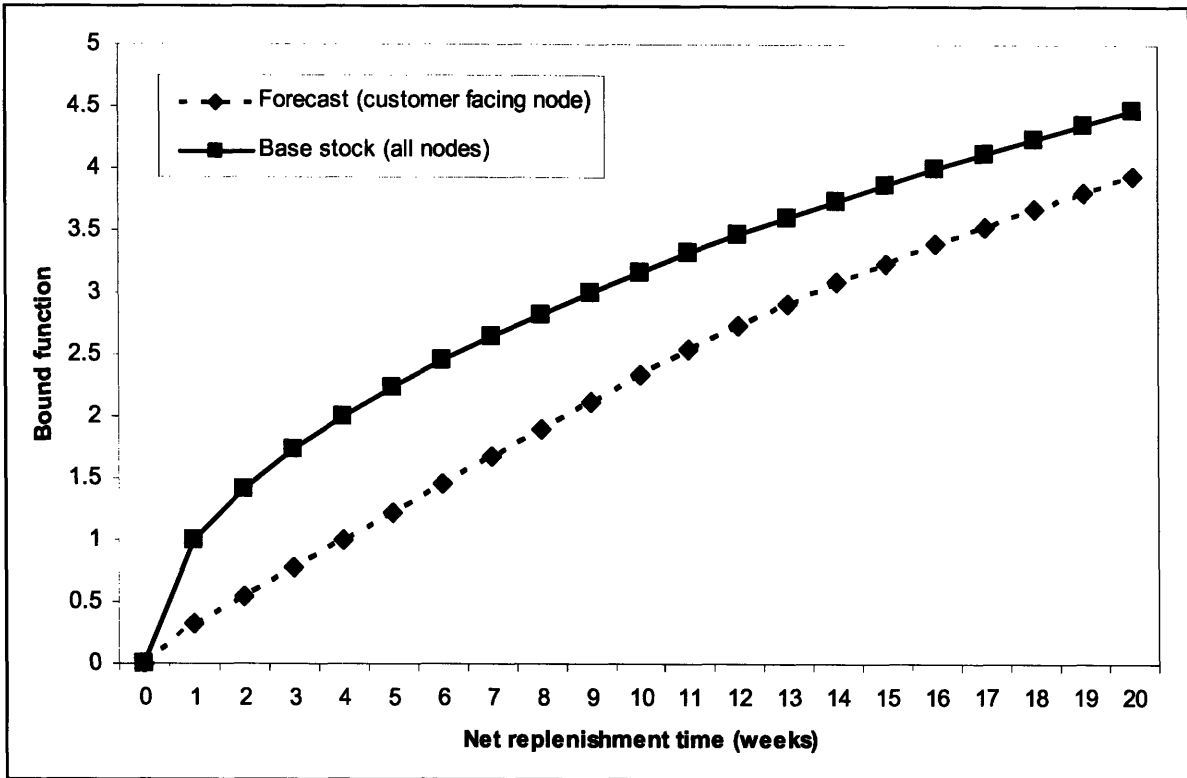


Figure 8: Normalized bound functions for systems with forecasts and with base stock policies.

We see that for node 1, the forecast-based ordering policy results in significantly less inventory than the base-stock ordering policy, especially when its net replenishment time is only a few weeks. This is because the forecasts are relatively accurate in the short term. As the net replenishment time increases, the inventory savings decline as the value of the forecast decreases.

We solved the optimization problem (87) for four cases. For three cases, we assume a forecast-based ordering policy but with different forecast properties. Specifically, we assumed that the correlation between the forecast and actual demand drops linearly from one to zero over a five week, ten week or twenty week horizon, and then remains at zero beyond this horizon. As illustrated in Figure 7, the ten week horizon closely matches the actual situation at Teradyne and is our base case. The five week and twenty week were hypothetical cases included for comparative purposes. For the fourth case, we assume a base-stock ordering policy, i.e, the Graves-Willems optimization; this case ignores the evolving forecast.

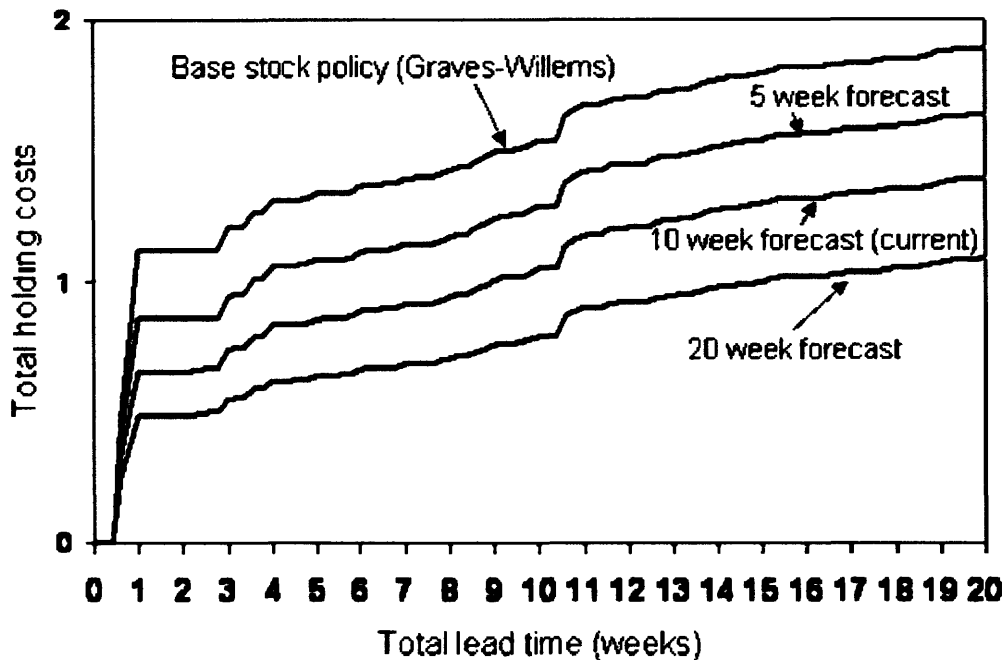


Figure 9: Total safety stock inventory costs for all the nodes with less than a certain lead time

The safety-stock holding costs for the forecast-based ordering policy with a ten-week horizon are 25% less than that for the base-stock ordering policy. Thus, for this supply chain, there seems to be substantial benefit from accounting for the forecast evolution when setting the safety stocks.

We also find that the safety stocks depend on the quality of the forecast process. For the actual case the forecast improves steadily over a ten-week period as seen in Figure 7. In comparison, we considered the supply chain assuming that the forecast improves over a twenty-week horizon. This is a higher quality forecast as all of the forecasts in a twenty-week horizon are more accurate relative to the ten-week case; when we optimized this case, we find that there is a reduction of 21% in the safety stock holding costs, relative to the optimal solution for the ten-week case. Similarly, we considered a lower quality forecast in which the forecast improves over a five-week horizon; here the optimal safety stock holding costs were 17% higher than for the case with a ten week horizon.

To get some intuition for how the four solutions differ, we calculated how the inventory was distributed in the supply chain for each case. In order to make this comparison, we defined the *minimal cumulative lead time* \tilde{L}_k to be the cumulative lead time L_k when $SI_k = 0$. That is:

$$\begin{aligned}\tilde{L}_1 &= T_1 \\ \tilde{L}_j &= \tilde{L}_k + T_j \quad k = a(j), \forall j\end{aligned}\tag{92}$$

By construction, \tilde{L}_k is a property of each node that does not depend on the service times, and so it serves as a measure by which we can compare the inventory placement for different solutions.

In Figure 9, we plot the total holding costs (on the y axis) for all nodes k such that $\tilde{L}_k \leq x$ (on the x axis). For example, at ten weeks on the x axis, the curve represents the total inventory holding cost for all inventory that can, in theory, be processed into finished products within 10 weeks. We only plot the holding costs for the first twenty weeks, which accounts for 97% of the total holding cost of the supply chain for the base stock case. Beyond twenty weeks the cumulative holding costs for the four cases grow at the same rate.

From Figure 9, we see that the difference in the solutions is found primarily in those parts of the supply chain whose distance is less than the effective range of the forecasts. For example, the safety stock for the base-stock policy initially grows much faster than that for the ten week case. As explanation, the ten week case requires very little safety stock in the downstream parts of the supply chain, as it can take advantage of the accuracy of the short-term forecasts. In contrast, the base-stock ordering policy does not use these forecasts and so cannot realize this gain; this policy must have inventories commensurate with the temporal variations of demand, which are considerable. However, beyond ten weeks, the safety stocks for the ten week case and for the base-stock case grow at approximately the same rate; this is because the stages with cumulative lead-times greater than ten weeks use the same bound functions for both the forecast case and the base-stock case and hence, the forecast case has no advantage over the base-stock case.

The computations for each of these supply chain problems took about one minute on a mobile computer (Intel® Core™2 CPU, 2.33GHz, 1GB RAM); no doubt this time can be reduced by implementation in a compiled programming language.

It is somewhat difficult to compare the solution directly to the actual situation at Teradyne, since at the time there were multiple layers of buffering and it was not clear what the actual service level was. As Teradyne had not performed a global optimization to set its stocks, we suspect that the optimal solution for the base-stock ordering policy is a fairly conservative proxy for their current state. Hence we believe there is substantial opportunity for improvement from the consideration of the evolving forecast.

22. Numerical examples

To examine the impact of an evolving forecast for various supply chains, we performed a number of numerical experiments. We used the same supply chain and cost structures as in Graves and Willems (2006). Specifically, we considered a serial system with $N = 5$ nodes, and with three alternatives for both the cost accumulation and the production lead-time as follows:

Stage	5	4	3	2	1
Increasing	36	28	20	12	4
Constant	20	20	20	20	20
Decreasing	4	12	20	28	36

Table 13: Alternative structures for supply chain lead-time and cost accumulation

The terms “increasing” and “decreasing” should be understood in terms of going upstream starting from the customer facing stage 1. In the case of cost accumulation, the values stated in Table 13 represent the cost added at each stage. For example, for the increasing cost scenario, the cost at stage 5 is 36, the cost at stage 4 is $36 + 28 = 64$, the cost at stage 3 is $36 + 28 + 20 = 84$, etc. For all three scenarios the cost of the finished good at stage 1 is 100.

For the production lead-times, the values for each scenario represent the values for T_k . For each scenario the cumulative lead-time for the supply chain is the 100.

We assume that the length of the forecast horizon is 100 periods and we consider five forecast processes. For each forecast process we use (26) as a bound on the forecast revisions, where we assume $z = 2, \sigma = 20$. Similar to the Teradyne example, we assume that the correlation between the forecast and realized demand goes linearly from 0 to 1 over a horizon of 0,25,50,75, or 100 periods. The first case thus represents no useful forecasts and is equivalent with Graves-Willems optimization. The 25, 50, 75 and 100 period cases represent increasing improvements in the quality of the forecast.

The combination of 3 cost structures, 3 lead-time structures, and 5 forecast horizons results in a total of 45 experiments, listed in Table 14 below. We state the optimal holding cost for the zero-horizon case, which corresponds to the base-stock policy. For the other forecast-horizon cases, we report the optimal cost as a percentage of the zero-horizon case.

		Forecast horizon				
Cost	Lead-Time	0	25	50	75	100
Increasing	Increasing	40.0	96.0%	90.8%	84.5%	78.3%
	Constant	40.0	96.0%	91.6%	86.9%	82.0%
	Decreasing	40.0	96.0%	91.6%	86.9%	82.0%
Constant	Increasing	36.8	87.2%	79.7%	72.2%	66.0%
	Constant	39.4	95.4%	90.3%	84.8%	79.0%
	Decreasing	40.0	96.0%	91.6%	86.9%	82.0%
Decreasing	Increasing	26.8	79.2%	66.7%	58.2%	52.0%
	Constant	34.6	93.9%	85.0%	76.6%	69.7%
	Decreasing	39.2	95.5%	90.5%	85.2%	79.4%

Table 14: Total costs for various supply chains and forecast horizons

We see that in all cases, one can reduce the safety-stock costs significantly if one can incorporate a high-quality forecast into the planning process.

Not surprisingly, the “decreasing” cost scenario leads to the lowest overall costs, because under this scenario the upstream stages have much lower holding costs compared

to the other cost scenarios. The inventory savings from the forecast are the largest for the decreasing cost scenario, relative to the other cost scenarios.

The “increasing” lead-time scenario also results in lower costs; in this scenario, the shortest lead-times are downstream, nearest to the customer, where holding costs are the highest. Similarly, the forecast provides the greatest savings for this case relative to the other lead-time scenarios.

The combination of decreasing costs and increasing lead times results in particularly low total costs, especially in combination with forecasts which are useful over the planning horizon.

We also report the structure of the optimal solution in Table 15. We denote a solution by a binary code, whereby a “1” in the k^{th} position denotes a decoupling inventory at stage k , while a “0” denotes no inventory. For example, “00001” represents inventory at stage 1 only.

		Forecast horizon				
Cost	Lead Time	0	25	50	75	100
Increasing	Increasing	00001	00001	10001	10001	10001
	Constant	00001	00001	00001	00001	00001
	Decreasing	00001	00001	00001	00001	00001
Constant	Increasing	01001	10011	10011	10101	10101
	Constant	10001	10001	10001	10001	10001
	Decreasing	00001	00001	00001	00001	00001
Decreasing	Increasing	11101	11011	11111	11111	11111
	Constant	11001	11001	10101	10101	10101
	Decreasing	11001	11001	11001	11001	10101

Table 15: Structure of optimal solution; 1 represents inventory at a node.

For this set of test problems there is a wide variety of optimal solutions. However, for a particular cost and lead-time scenario, the structure of the solution (i.e, where we place safety stocks) is relatively stable across the different forecast scenarios. This is important as it suggests that the optimal location of safety stocks depends primarily on

how the holding costs and lead-times are set across the supply chain, rather than on the specifics of the forecast process.

23. Conclusions and future directions

In spite of the ubiquity of forecast-based planning systems (e.g., MRP systems), the analysis of safety stocks has been limited to simple special cases, such as one or two nodes, i.i.d. demand processes or perfect forecasts (Guide and Srivastava 2000). In this paper we develop an approach that extends the framework of Simpson (1958) and Graves and Willems (2000) to include an evolving forecast. We are then able to apply the dynamic programming algorithm from Graves and Willems (2000) to solve for the safety stocks in assembly-system supply chains. Furthermore we demonstrate that accounting for the forecast evolution process results in less safety stock, where the magnitude of the savings depends on the quality of the forecasts. We expect that our approach is computationally fast enough to solve supply chains of any size likely to arise in practice.

In the literature, there is debate over where to place safety stocks in MRP systems, or the type of buffer to use (Guide and Srivastava 2000). If one accepts the specific assumptions made in this paper, then we note that the optimal placement of supply-chain safety stocks is driven by three different (and sometimes conflicting) principles. The first two points are the same as for systems with base-stock ordering policies.

- Statistical economies of scale, as manifested in the (strict) concavity of the bound functions, encourage the use of fewer, larger, safety-stock buffers.
- Value-adding activities (holding costs that increase downstream in the supply chain) encourage the use of more numerous, smaller, and distributed safety stocks
- When using a forecast-based ordering policy (e.g., MRP logic), the overall size of the safety stocks depends on the size of the forecast errors, rather than the variability of demand. To the extent that we have a meaningful forecast, we expect the forecast errors to be smaller than demand variability, resulting in less safety stock. These reductions in safety stock will primarily be downstream in the supply chain, at the stages whose cumulative lead-times correspond to the horizon of useful forecasts.

Finally we remind the reader of the limitations of this work, and the opportunities that this suggests. We do not have a good solution for supply chains with spanning-trees topologies. We believe that the crux of the problem is to find a practical way to determine the bound function on the forecast revisions; nevertheless, there might be completely different and better ways to approach and analyze this type of supply chain. We also leave even more general (cyclic) networks for future research. Another limitation is the simplicity of the ordering policy. We do not consider many features that are typically incorporated in MRP systems, such as lot sizing, capacity constraints, and supply uncertainty; we do not account for these considerations in the present framework for strategic safety stocks. Finally we note our assumption of deterministic procurement and production lead-times; it would be most valuable to determine how to extend this approach to accommodate stochastic lead-times, as is common in practice.

Appendix

In this Appendix, we provide proofs and detailed derivations for some of the claims made throughout the text. The order is the same as that used in the text.

Derivation of (68)

$$\begin{aligned}
\Delta f_t(t+s) &= E[D_{t+s} | \Psi_t, \Psi_{t-1}, \dots] - E[D_{t+s} | \Psi_{t-1}, \Psi_{t-2}, \dots] \\
&= E[\mu + RHX_{t+s} | \Psi_t, \Psi_{t-1}, \dots] - E[\mu + RHX_{t+s} | \Psi_{t-1}, \Psi_{t-2}, \dots] \\
&= RHE[X_{t+s} | \Psi_t, \Psi_{t-1}, \dots] - RHE[X_{t+s} | \Psi_{t-1}, \Psi_{t-2}, \dots] \\
&= RHE[FX_{t+s-1} + V_{t+s-1} | \Psi_t, \Psi_{t-1}, \dots] - RHE[FX_{t+s-1} + V_{t+s-1} | \Psi_{t-1}, \Psi_{t-2}, \dots] \\
&= RHF E[X_{t+s-1} | \Psi_t, \Psi_{t-1}, \dots] - RHF E[X_{t+s-1} | \Psi_{t-1}, \Psi_{t-2}, \dots] \\
&= \dots = RHF^s E[X_t | \Psi_t, \Psi_{t-1}, \dots] - RHF^s E[X_t | \Psi_{t-1}, \Psi_{t-2}, \dots] \\
&= RHF^s (X_t - E[X_t | \Psi_{t-1}, \Psi_{t-2}, \dots]) \\
&= RHF^s (FX_{t-1} + V_t - E[FX_{t-1} + V_t | \Psi_{t-1}, \Psi_{t-2}, \dots]) \\
&= RHF^s (FX_{t-1} + V_t - E_{t-1}[FX_{t-1} + V_t | \Psi_{t-1}, \Psi_{t-2}, \dots]) \\
&= RHF^{s+1} V_t
\end{aligned} \tag{A93}$$

Proof that (72) holds if and only if $E_t[I_k(t+T_k + S_{k+1})]$ constant

In this proof, and the ones to follow, we will use the notation $E_t[\cdot]$ to indicate expectation conditional on all events (specifically, all realizations of the forecast revision process) up to time t , inclusive. First we note that for i in $t < i < t + T_k + S_{k+1}$ it is impossible to hold $E_t[I_k(i)]$ constant since any control only affects $I_k(t)$ after a leadtime $T_k + S_{k+1}$. Furthermore, if $E_t[I_k(t+T_k + S_{k+1})]$ is held constant every period, then $E_t[I_k(i)]$ for $i > t + T_k + S_{k+1}$ will automatically be held constant as well. Hence, keeping $E_t[I_k(t+T_k + S_{k+1})]$ constant is in some sense the best we can do in terms of keeping expected future inventory constant. We now proceed with proving the claim. If: This will be shown in the derivation of (75) below.

Only if: We need to assume that $E_t[I_k(t+T_k+S_{k+1})]=I_k^0$, and show that this implies the policy (72). In what follows we use the identity $E_t[P_s^k]=P_s^k$ for $t \geq s$:

$$\begin{aligned}
I_k^0 &= E_t[I_k(t+T_k+S_{k+1})] \\
&= E_{t-1}[I_k(t+T_k+S_{k+1}-1)] + (E_t[I_k(t+T_k+S_{k+1})] - E_{t-1}[I_k(t+T_k+S_{k+1}-1)]) \\
&= I_k^0 + \left(I_k(t-1) - \sum_{i=t-S_k}^{t+T_k+S_{k+1}-S_k} E_t[P_i^{k-1}] + \sum_{i=t-T_k-S_{k+1}}^{i=t} P_i^k \right) - \left(I_k(t-1) - \sum_{i=t-S_k}^{i=t+T_k+S_{k+1}-S_k-1} E_{t-1}[P_i^{k-1}] + \sum_{i=t-T_k-S_{k+1}}^{i=t-1} P_i^k \right) \\
&= I_k^0 + \left(-E_t[P_{t+T_k+S_{k+1}-S_k}^{k-1}] - \sum_{i=t-S_k}^{t+T_k+S_{k+1}-S_k-1} E_t[P_i^{k-1}] + \sum_{i=t-T_k-S_{k+1}}^{i=t-1} P_i^k + P_t^k \right) - \\
&\quad \left(- \sum_{i=t-S_k}^{i=t+T_k+S_{k+1}-S_k-1} E_{t-1}[P_i^{k-1}] + \sum_{i=t-T_k-S_{k+1}}^{i=t-1} P_i^k \right) \\
&= I_k^0 - E_t[P_{t+T_k+S_{k+1}-S_k}^{k-1}] + \sum_{i=t-S_k}^{t+T_k+S_{k+1}-S_k-1} (E_{t-1}[P_i^{k-1}] - E_t[P_i^{k-1}]) + P_t^k \\
&= I_k^0 - E_t[P_{t+T_k+S_{k+1}-S_k}^{k-1}] + \sum_{i=t-S_k}^{t-1} (E_{t-1}[P_i^{k-1}] - E_t[P_i^{k-1}]) + (E_{t-1}[P_t^{k-1}] - E_t[P_t^{k-1}]) \\
&\quad + \sum_{i=t+1}^{t+T_k+S_{k+1}-S_k-1} (E_{t-1}[P_i^{k-1}] - E_t[P_i^{k-1}]) + P_t^k \\
&= I_k^0 - E_t[P_{t+T_k+S_{k+1}-S_k}^{k-1}] + \sum_{i=t-S_k}^{t-1} (P_i^{k-1} - P_i^{k-1}) + (E_{t-1}[P_t^{k-1}] - P_t^{k-1}) \\
&\quad + \sum_{i=t+1}^{t+T_k+S_{k+1}-S_k-1} (E_{t-1}[P_i^{k-1}] - E_t[P_i^{k-1}]) + P_t^k \\
&= I_k^0 - E_t[P_{t+T_k+S_{k+1}-S_k}^{k-1}] + (E_{t-1}[P_t^{k-1}] - P_t^{k-1}) + \sum_{i=t+1}^{t+T_k+S_{k+1}-S_k-1} (E_{t-1}[P_i^{k-1}] - E_t[P_i^{k-1}]) + P_t^k
\end{aligned}$$

Comparing the first and last line, and rewriting in terms of P_t^k , we have

$$P_t^k = E_t[P_{t+T_k+S_{k+1}-S_k}^{k-1}] + P_t^{k-1} - E_{t-1}[P_t^{k-1}] + \sum_{i=t+1}^{t+T_k+S_{k+1}-S_k-1} (E_t[P_i^{k-1}] - E_{t-1}[P_i^{k-1}]) \quad (\text{A94})$$

For the special case $k=1$, we have that $P_t^0 = D_t$ and $E_t[D_{t+j}] \equiv f_t(t+j)$. Inserting these identities in (A94) gives us (72). Hence the claim is true for $k=1$, we now proceed to prove it for greater k , by induction. Suppose that the policy (72) is used for some $k-1$, and that $I_k^0 = E_t[I(t+T_k+S_{k+1})]$ for all k . Then we have:

$$\begin{aligned}
P_t^k &= E_t[P_{t+T_k+S_{k+1}-S_k}^{k-1}] + P_t^{k-1} - E_{t-1}[P_t^{k-1}] + \sum_{i=t+1}^{t+T_k+S_{k+1}-S_k-1} (E_i[P_i^{k-1}] - E_{i-1}[P_i^{k-1}]) \\
&= f_t(t+L_{k-1}+T_k+S_{k+1}-S_k) + P_t^{k-1} - f_{t-1}(t+L_{k-1}) + \sum_{i=t+1}^{t+T_k+S_{k+1}-S_k-1} (f_t(i+L_{k-1}) - f_{i-1}(i+L_{k-1})) = \\
&= f_t(t+L_k) + \left(f_t(t+L_{k-1}) + \sum_{i=t}^{t+L_{k-1}-1} \Delta f_t(i) \right) - f_{t-1}(t+L_{k-1}) + \sum_{i=t+L_{k-1}+1}^{t+L_k-1} \Delta f_t(i) = \\
&= f_t(t+L_k) + \sum_{i=t}^{t+L_k-1} \Delta f_t(i)
\end{aligned} \tag{A95}$$

Since we showed that the claim was true for $k=1$, we have that $I_k^0 = E_t[I_k(t+T_k+S_{k+1})]$ for all k implies policy (72) for all k .

Proof that (72) and (73) are equivalent

From the inventory balance equation (74) we can describe $I_k(t+T_k+SI_k)$ in terms of $I_k(t)$ and incoming and outgoing orders.

$$I_k(t+T_k+SI_k) = I_k(t) - \sum_{i=1}^{T_k+SI_k} P_{k-1}(t+i-S_k) + \sum_{i=1}^{T_k+SI_k} P_k(t+i-T_k-SI_k) \tag{A96}$$

Taking the expectation at time t gives us

$$E_t[I_k(t+T_k+SI_k)] = I_k(t) - \sum_{i=1}^{T_k+SI_k} E_t[P_{k-1}(t+i-S_k)] + \sum_{i=1}^{T_k+SI_k} E_t[P_k(t+i-T_k-SI_k)].$$

We note that in the rightmost sum $t+i-T_k-SI_k \leq t$ and so the orders

$P_k(t+i-T_k-SI_k)$ have already been realized and the expectation operator is redundant. If we furthermore simplify the last summation index and separate out the last term we get

$$\begin{aligned}
&I_k(t) - \sum_{i=1}^{T_k+SI_k} E_t[P_{k-1}(t+i-S_k)] + \sum_{i=1}^{T_k+SI_k} E_t[P_k(t+i-T_k-SI_k)] \\
&= I_k(t) - \sum_{i=1}^{T_k+SI_k} E_t[P_{k-1}(t+i-S_k)] + \sum_{i=1}^{T_k+SI_k-1} P_k(t-i) + P_k(t)
\end{aligned}$$

So far we have not used any properties of the policy itself, only of the generic inventory balance equations. Now suppose that the order $P_k(t)$ is determined by (73). Then:

$$\begin{aligned}
& I_k(t) - \sum_{i=1}^{T_k+SI_k} E_t [P_{k-1}(t+i-S_k)] + \sum_{i=1}^{T_k+SI_k-1} P_k(t-i) + P_k(t) \\
&= I_k(t) - \sum_{i=1}^{T_k+SI_k} E_t [P_{k-1}(t+i-S_k)] + \sum_{i=1}^{T_k+SI_k-1} P_k(t-i) + \\
&\quad \left(\sum_{i=1}^{T_k+SI_k} E_t [P_{k-1}(t+i-S_k)] - \sum_{i=1}^{T_k+SI_k-1} P_k(t-i) - I_k(t) + I_k^0 \right) \\
&= I_k^0
\end{aligned}$$

Thus when (73) is used $E_t[I_k(t+T_k+SI_k)] = I_k^0$ and is thus constant. But we have already shown that this property uniquely characterizes the policy (72), and so (72) and (73) must be equivalent.

Derivation of (75)

$$\begin{aligned}
I_k(t+T_k+S_{k+1}) &= I_k(t+S_k) - \sum_{i=t+1+S_k-S_k}^{t+T_k+S_{k+1}-S_k} P_i^{k-1} + \sum_{i=t+1+S_k-(T_k+S_{k+1})}^{t+(T_k+S_{k+1})-(T_k+S_{k+1})} P_i^k \\
&= I_k(t+S_k) - \sum_{i=t+1}^{t+\tau_k} P_i^{k-1} + \sum_{i=t+1-\tau_k}^t P_i^k \\
&= I_k(t+S_k) - \sum_{i=t+1}^{t+\tau_k} \left(\sum_{j=i}^{i+L_{k-1}-1} \Delta f_i(j) + f_i(i+L_{k-1}) \right) + \sum_{i=t+1-\tau_k}^t \left(\sum_{j=i}^{i+L_k-1} \Delta f_i(j) + f_i(i+L_k) \right) \\
&= I_k(t+S_k) - \sum_{i=t+1}^{t+\tau_k} \left(\sum_{j=i}^{i+L_{k-1}-1} \Delta f_i(j) + \cancel{f_{i-\tau_k}(i+L_{k-1})} + \sum_{l=i-\tau_k+1}^i \Delta f_l(i+L_{k-1}) \right) \\
&\quad + \sum_{i=t+1-\tau_k}^t \left(\sum_{j=i}^{i+L_k-1} \Delta f_i(j) + \cancel{f_i(i+L_k)} \right) \\
&= I_k(t+S_k) - \sum_{i=t+1}^{t+\tau_k} \left(\sum_{j=i}^{i+L_{k-1}-1} \Delta f_i(j) + \sum_{l=i-\tau_k+1}^i \Delta f_l(i+L_{k-1}) \right) + \sum_{i=t+1-\tau_k}^t \sum_{j=i}^{i+L_k-1} \Delta f_i(j) \\
&= I_k(t+S_k) - \sum_{i=t+1}^{t+\tau_k} \left(\sum_{j=i}^{i+L_{k-1}-1} \Delta f_i(j) + \sum_{l=t+1}^i \Delta f_l(i+L_{k-1}) \right) + \\
&\quad + \left(\sum_{i=t+1-\tau_k}^t \sum_{j=i}^{i+L_k-1} \Delta f_i(j) - \sum_{i=t+1}^{t+\tau_k} \sum_{l=i-\tau_k+1}^i \Delta f_l(i+L_{k-1}) \right) \\
&= I_k(t+S_k) - \sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{i+L_{k-1}-1} \Delta f_i(j) - \sum_{j=t+1}^{t+\tau_k} \sum_{i=t+1}^j \Delta f_i(j+L_{k-1}) \\
&\quad + \left(\sum_{i=t+1-\tau_k}^t \sum_{j=i}^{i+L_k-1} \Delta f_i(j) - \sum_{j=t+1}^{t+\tau_k} \sum_{i=j-\tau_k+1}^i \Delta f_i(j+L_{k-1}) \right) \\
&= I_k(t+S_k) - \sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{i+L_{k-1}-1} \Delta f_i(j) - \sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+\tau_k} \Delta f_i(j+L_{k-1}) + \\
&\quad + \left(\sum_{i=t+1-\tau_k}^t \sum_{j=i}^{i+L_k-1} \Delta f_i(j) - \sum_{i=t+1-\tau_k}^t \sum_{j=t+1}^{i+\tau_k-1} \Delta f_i(j+L_{k-1}) \right) \\
&= I_k(t+S_k) - \sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{i+L_{k-1}-1} \Delta f_i(j) - \sum_{i=t+1}^{t+\tau_k} \sum_{j=i+L_{k-1}}^{t+L_k} \Delta f_i(j) + \left(\sum_{i=t+1-\tau_k}^t \sum_{j=i}^{i+L_k-1} \Delta f_i(j) - \sum_{i=t+1-\tau_k}^t \sum_{j=t+1+L_{k-1}}^{i+L_k-1} \Delta f_i(j) \right) \\
&= I_k(t+S_k) - \sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) + \left(\sum_{i=t+1-\tau_k}^t \sum_{j=i}^{t+L_{k-1}} \Delta f_i(j) \right)
\end{aligned}$$

We note that $\sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k} \Delta f_i(j)$ is the change of inventory caused by new forecast revisions in the time window $[t+1, t+\tau_k]$, and $\sum_{i=t+1-\tau_k}^t \sum_{j=i}^{t+L_{k-1}} \Delta f_i(j)$ is the replenishment for the forecasts that happened during $[t+1-\tau_k, t]$. These expressions are identical except for a translation of τ_k . If there were no change of forecasts during $[t+1, t+\tau_k]$, $\sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) = 0$, then the replenishment would bring the safety stock back to its default value which we denote

I_k^0 . Mathematically, we have that $I_k(t+S_k) + \left(\sum_{i=t+1-\tau_k}^t \sum_{j=i}^{t+L_{k-1}} \Delta f_i(j) \right) = I_k^0$. Hence

$I_k(t+T_k+S_{k+1}) = I_k^0 - \sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k} \Delta f_i(j)$. Note also that I_k^0 is the average inventory level:

$$E[I_k(t+T_k+S_{k+1})] = E[I_k^0] - E \left[\sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) \right] = I_k^0 \quad (\text{A97})$$

Derivation of (80)

For the bound to be valid, we need to show that

$$z\sigma(I_k(t)) = \sqrt{F^2(L_k) - F^2(L_{k-1})}. \quad (\text{A98})$$

We have that

$$\begin{aligned}
& \text{var}[I_k(t)] \\
&= \text{var} \left[\sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) \right] \\
&= \sum_{i=t+1}^{t+\tau_k} \text{var} \left[\sum_{j=i}^{t+L_k} \Delta f_i(j) \right] \\
&= \sum_{i=t+1}^{t+L_k} \text{var} \left[\sum_{j=i}^{t+L_k} \Delta f_i(j) \right] - \sum_{i=t+\tau_k+1}^{t+L_k} \text{var} \left[\sum_{j=i}^{t+L_k} \Delta f_i(j) \right] \\
&= \text{var} \left[\sum_{i=t+1}^{t+L_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) \right] - \text{var} \left[\sum_{i=t+\tau_k+1}^{t+L_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) \right].
\end{aligned} \tag{A99}$$

We note that since $\Delta f_i(j)$ are i.i.d., we can add or deduct any constant from i and j , and still preserve the variance:

$$\begin{aligned}
& \text{var} \left[\sum_{i=t+1}^{t+L_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) \right] - \text{var} \left[\sum_{i=t+\tau_k+1}^{t+L_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) \right] \\
&= \text{var} \left[\sum_{i=t+1}^{t+L_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) \right] - \text{var} \left[\sum_{i=t+\tau_k+1}^{t+L_k} \sum_{j=i}^{t+L_k} \Delta f_{i-\tau_k}(j-\tau_k) \right]
\end{aligned} \tag{A100}$$

Because $L_k = L_{k-1} + \tau_k$, we have

$$\begin{aligned}
& \text{var} \left[\sum_{i=t+1}^{t+L_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) \right] - \text{var} \left[\sum_{i=t+\tau_k+1}^{t+L_k} \sum_{j=i}^{t+L_k} \Delta f_{i-\tau_k}(j-\tau_k) \right] \\
&= \text{var} \left[\sum_{i=t+1}^{t+L_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) \right] - \text{var} \left[\sum_{i=t+\tau_k+1}^{t+L_k} \sum_{j=i-\tau_k}^{t+L_k-\tau_k} \Delta f_{i-\tau_k}(j) \right] \\
&= \text{var} \left[\sum_{i=t+1}^{t+L_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) \right] - \text{var} \left[\sum_{i=t+\tau_k-\tau_k+1}^{t+L_k-\tau_k} \sum_{j=i}^{t+L_k-\tau_k} \Delta f_i(j) \right] \\
&= \text{var} \left[\sum_{i=t+1}^{t+L_k} \sum_{j=i}^{t+L_k} \Delta f_i(j) \right] - \text{var} \left[\sum_{i=t+1}^{t+L_{k-1}} \sum_{j=i}^{t+L_{k-1}} \Delta f_i(j) \right] \\
&= \text{var} \left[\sum_{j=t+1}^{t+L_k} \sum_{i=t+1}^j \Delta f_i(j) \right] - \text{var} \left[\sum_{j=t+1}^{t+L_{k-1}} \sum_{i=t+1}^j \Delta f_i(j) \right] \\
&= \text{var} \left[\sum_{j=t+1}^{t+L_k} (f_j(j) - f_t(j)) \right] - \text{var} \left[\sum_{j=t+1}^{t+L_{k-1}} (f_j(j) - f_t(j)) \right] \\
&= \text{var} \left[\sum_{j=t+1}^{t+L_k} (D_j - f_t(j)) \right] - \text{var} \left[\sum_{j=t+1}^{t+L_{k-1}} (D_j - f_t(j)) \right] \tag{A101}
\end{aligned}$$

Thus

$$\text{var}[I_k(t)] = \text{var} \left[\sum_{j=t+1}^{t+L_k} (D_j - f_t(j)) \right] - \text{var} \left[\sum_{j=t+1}^{t+L_{k-1}} (D_j - f_t(j)) \right], \tag{A102}$$

which combined with the definition (79) of F , gives the claimed relation.

Derivation of (90)

$$\begin{aligned}
& \text{var}(D_t - f_i(t)) \\
&= \text{var}\left(\sum_{j=i+1}^t \Delta f_j(t)\right) \\
&= \text{var}\left(\sum_{j=i+1}^t \Delta f_j(t)\right) + \text{var}(f_i(t)) - \text{var}(f_i(t)) = \\
&= \text{var}\left(\sum_{j=i+1}^t \Delta f_j(t) + f_i(t)\right) - \text{var}(f_i(t)) = \\
&= \text{var}(D_t) - \text{var}(f_i(t)) \\
&= \left(1 - \frac{\text{var}(f_i(t))}{\text{var}(D_t)}\right) \text{var}(D_t)
\end{aligned} \tag{A103}$$

Finally, using (89), we have:

$$\begin{aligned}
&= \left(1 - \frac{\text{var}(f_i(t))}{\text{var}(D_t)}\right) \text{var}(D_t) \\
&= (1 - \rho^2(D_t, f_i(t))) \text{var}(D_t)
\end{aligned} \tag{A104}$$

Derivation of (91) under additional independence assumption

We have from (79) and (80) that:

$$B(L_{k-1}, L_k) = z \sqrt{\text{var}\left(\sum_{j=i+1}^{t+L_k} (D_j - f_t(j))\right) - \text{var}\left(\sum_{j=i+1}^{t+L_{k-1}} (D_j - f_t(j))\right)} \tag{A105}$$

If we now make the additional independence assumption (as stated), then $(D_j - f_t(j))$

are independent for different j . Then

$$z \sqrt{\text{var}\left(\sum_{j=i+1}^{t+L_k} (D_j - f_t(j))\right) - \text{var}\left(\sum_{j=i+1}^{t+L_{k-1}} (D_j - f_t(j))\right)} = z \sqrt{\sum_{j=i+L_{k-1}+1}^{t+L_k} \text{var}[D_j - f_t(j)]} \tag{A106}$$

And finally, using (90)

$$\begin{aligned}
& z \sqrt{\sum_{j=t+L_{k-1}+1}^{t+L_k} \text{var}[D_j - f_t(j)]} \\
&= z\sigma(D_t) \sqrt{\sum_{j=t+L_{k-1}+1}^{t+L_k} (1 - \rho^2(D_j, f_t(j)))} \\
&= z\sigma(D_t) \sqrt{\sum_{j=t+L_{k-1}+1}^{t+L_k} 1 + \sum_{j=t+1}^{t+L_k} (-\rho^2(D_j, f_t(j))) - \sum_{j=t+1}^{t+L_{k-1}} (-\rho^2(D_j, f_t(j)))} \\
&= z\sigma(D_t) \sqrt{T_k + S_{k+1} - S_k - \sum_{j=t+L_{k-1}+1}^{t+L_k} \rho^2(D_j, f_t(j))}
\end{aligned} \tag{A107}$$

V. Addendum

24. Forecast-based ordering with multiple end products

In the essay “Strategic safety stocks in supply chains with evolving forecasts,” we extended the applicability of the guaranteed service framework to a setting in which orders were placed in response to changes in forecasts and schedules. We showed that one can use the Graves-Willems optimization algorithm, which was developed under the assumption of base-stock ordering, in the forecast setting after some limited modifications. In particular, the key modification is to replace the demand bound with a forecast error bound.

Our work on forecasts was limited to supply chains with a single demand stage. Here, we discuss how we can construct bounds on the forecast errors for multiple merged order streams. Specifically, we define the set A of demand nodes. We also define the binary (indicator) variable δ_{kl} , for $l \in A$, so that $\delta_{kl} = 1$ if the demand node l is downstream of the node k , and zero otherwise. We index the demand and forecast processes by the relevant demand node, so that $d^l(t)$ is the demand in period t at stage l , and $f_i^l(t+j)$ is the forecast made at time t for the demand at node l , at time $t+j$. We assume that for each l , the forecasts f_i^l and forecasts revisions Δf_i^l fulfill all the condition specified in the single customer case.

We assume that a stage k which (directly or indirectly) serves multiple customer stages, provides the same service time S_k to all of them (this assumption is shared with Graves and Willems). We define $C(k,l)$ to be the set of stages that are downstream of k but upstream of l (inclusive); we can then define the cumulative lead time

$$L_k^l = SI_k + \sum_{j \in C(k,l)} T_j. \quad (108)$$

That is, the cumulative lead time is specified for a duplet (k, l) , which is the time it takes from the point where k places an order, until that order has been delivered all the way to the demand node l . With this notation in place, we can generalize the inventory formula

(10) from the essay “Strategic safety stocks in supply chains with evolving forecasts”, as follows:

$$I_k(t + T_k + SI_k) = I_k^0 - \sum_{l \in A} \delta_{kl} \left(\sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k^l} \Delta f_i^l(j) \right). \quad (109)$$

Assuming, analogously to the single product case, that Δf_i^l are normally distributed, we can specify a probabilistic service level by setting

$$B_k(S_k, SI_k) = z\sigma[I_k(t)] = z\sigma \left[\sum_{l \in A} \delta_{kl} \left(\sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k^l} \Delta f_i^l(j) \right) \right], \quad (110)$$

where z is a service level parameter and $\sigma[]$ indicates standard deviation. It is thus quite straightforward to *formulate* the equations for inventory and forecast error bounds, even when there are multiple demand nodes. The challenge is to determine the standard deviation term in practical applications.

If the forecast (and demand) processes for different end products are independent, we have from elementary probability that

$$B_k(S_k, SI_k) \underset{\text{independent } l}{=} z \sqrt{\sum_{l \in A} \delta_{kl} \text{var} \left[\sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k^l} \Delta f_i^l(j) \right]} \quad (111)$$

In principle, one could envision specifying a multivariate demand and forecast process, and then determining (110) analytically. However, this approach may require an extraordinary amount of data and computation. Specifically, if there are M demand nodes and a forecast horizon of length H , then the forecast revision process $\Delta f_i^l(t+i)$ consists of MH different values, and would have a covariance matrix with M^2H^2 values. Such a matrix can be challenging to store, let alone compute from historical data.

Instead, we propose that we estimate $I_k(t)$ and $z\sigma[I_k(t)]$ directly from historical data. That is, rather than completely characterizing the (multi-dimensional) forecast revision process, one can save a lot of memory and computation by directly calculating only the resulting inventory variations for relevant service times. In each period we observe the forecast revisions for each product. If we have access to a sufficient quantity

of such data, we can determine the empirical distribution of $\sum_{l \in A} \delta_{kl} \left(\sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k^l} \Delta f_i^l(j) \right)$, for any parameter values needed by the optimization. Specifically, suppose we need to evaluate (111) for given values of k , S_k and SI_k , and that we have data on past Δf_i^l . Suppose further that for the purpose of finding asymptotic complexity, we disregard the factor $\sum_{l \in A} \delta_{kl}$. Then if we evaluate the term for T different values of t , the computational cost will be $O(\tau_k L_k^l T)$, or $O(H^2 T)$ if we consider that the first two terms are bounded by the forecast horizon H . In order to perform Graves-Willems' optimization, (111) must be evaluated for all N stages, and for each stage for $O(H^2)$ different combinations of S_k and SI_k . Thus the total computational requirement for this naïve approach is $O(NH^4 T)$.

We can improve on this by noting that the sums in $\sum_{i=t+1}^{t+\tau_k} \sum_{j=i}^{t+L_k^l} \Delta f_i^l(j)$ will be

calculated many times, often with only small changes in indices. Recalculating the whole sum from scratch every time may not be the most economical approach. Indeed, we will next outline a faster method, starting by defining the stage-specific matrices a_k and b_k as follows:

$$a_k(i, m) = \sum_{l \in A} \delta_{kl} \left(\sum_{j=i}^{m+1 + \sum_{j \in C(k, l)} T_j} \Delta f_i^l(j) \right) \quad (112)$$

$$b_k(n, m) = \sum_{i=1}^n a_k(i, m)$$

The matrices a_k and b_k and their individual elements do not necessarily have any particular intuitive meaning, rather they are devices which help us store partially computed sums. a_k has the dimension $(T+H) \times (T+H)$, and b_k has the dimension $H \times (T+H)$. We note that a_k and b_k can be calculated recursively:

$$\begin{aligned} a_k(i, m) &= a_k(i, m-1) + \sum_{l \in A} \delta_{kl} \Delta f_i^l(m+1 + \sum_{j \in C(k, l)} T_j) \\ b_k(n, m) &= b_k(n-1, m) + a_k(n, m) \end{aligned} \quad (113)$$

Hence, we can use the following algorithm to calculate a_k and b_k for all relevant values, using only $O(NH(T+H)) = O(NHT)$ operations. The symbol \leftarrow stands for assignment.

1. For $i \in (1, \dots, T+H)$

$$\text{a. } a_k(i, i) \leftarrow \sum_{l \in A} \delta_{kl} \left(\sum_{j=i}^{i+1 + \sum_{j \in C(k, l)} T_j} \Delta f_i^l(j) \right)$$

b. For $m \in (i+1, \dots, i+H)$

$$a_k(i, m) \leftarrow a_k(i, m-1) + \sum_{l \in D} \delta_{kl} \Delta f_i^l(m+1 + \sum_{j \in C(k, l)} T_j)$$

2. For $m \in (1, \dots, T+H)$

$$\text{a. } b_k(1, m) \leftarrow a_k(1, m)$$

b. For $n \in (2, \dots, H)$

$$b_k(n, m) \leftarrow b_k(n-1, m) + a_k(n, m)$$

Now using (110) we have that

$$\begin{aligned} \sigma[I_k(t)] &= \\ \sigma \left[\sum_{l \in A} \delta_{kl} \left(\sum_{i=t+1}^{t+T_k+SI_k-S_k} \sum_{j=i}^{t+SI_k + \sum_{j \in C(k, l)} T_j} \Delta f_i^l(j) \right) \right] &= \\ = \sigma \left[\sum_{i=t+1}^{t+T_k+SI_k-S_k} a_k(i, t+SI_k) \right] &= \\ = \sigma [b_k(t+T_k+SI_k-S_k, SI_k) - b_k(t, SI_k)] & \end{aligned} \quad (114)$$

And therefore, for a service level type bound, we have that

$$B_k(S_k, SI_k) = z_k \sqrt{\frac{1}{T} \sum_i (b_k(t+T_k+SI_k-S_k, SI_k) - b_k(t, SI_k))^2} \quad (115)$$

If b has been calculated in advance and stored in a matrix, (115) can be calculated with $O(T)$ operations whenever $B_k(S_k, SI_k)$ is called, which happens $O(NH^2)$ times in the Graves-Willems algorithm. Thus the total time complexity for first calculating a and b from historical data and then solving for the optimal solution is $O(NHT + NH^2 \times T) = O(NH^2T)$. This represents an increase by only a factor T over the original base-stock algorithm.

As we saw in the case study in “Strategic safety stocks in supply chains with evolving forecasts”, Graves-Willems algorithm enables us to quickly optimize even a very large supply chain on a mobile computer. This suggests that the increase of a factor T will be affordable in many practical situations.

How large does T need to be in order to establish reliable estimates of $B_k(S_k, SI_k)$? We recall that $B_k(S_k, SI_k)$ is a measure of the maximum cumulative forecast errors. The forecast horizon H is the greatest time period for which we need to estimate these cumulative errors. Therefore, it is desirable to have T many times larger than H , so that we have many independent cumulative forecast error realizations. For example, in a supply chain with a maximum lead time of two months, one may want at least two years worth of data, to have 12 independent observations of cumulative forecast errors over two months. To get a more quantitative understanding of estimation errors, one can determine confidence intervals. It must be noted, however, that 2 years of data will generate a very large number of separate but overlapping observations that can be used for example (111). However, these sums are not independent and must not be treated as such for the purposes of calculating confidence intervals. We leave a deeper exploration of this topic for future research.

25. Base-stock ordering when demand is not i.i.d.

In this section we will make some remarks about demand that is not i.i.d. in guaranteed service models. We recall that guaranteed service models do not make any explicit assumptions about demand (other than the existence of an average demand level μ). However, the demand bounds can be seen as being consistent with certain demand distributions. In particular, Simpson (1958) proposed using the demand bound $\mu\tau + \sigma z\sqrt{\tau}$, where σ is the standard deviation of demand, z a service level parameter, and τ the net replenishment time. This bound is consistent with providing a certain service level for a demand process that is i.i.d., and normally distributed. The bound has been extensively used in theory and practice. More recently, Graves and Willems (2000) suggest that the bound $\mu\tau + \sigma z(\tau)^{1/p}$ can be used, and indeed has been used, in situations with non-i.i.d. demand. Although Graves and Willems (2000) provide no theoretical justification for this particular form, one might choose the value of p by fitting to historical data. A $p < 2$ implies that the demand bound grows *faster* over time than in the i.i.d. case, which is consistent with positive period-to-period correlations in demand.

Here, we remark that given any model for a stationary demand process, it should in fact be possible to derive an exact form for the cumulative demand over some window τ . We illustrate this by looking at an auto-regressive, 1st order model, AR(1). We hope that this example will be useful in its own right, as well as motivate investigations of other demand distributions.

We recall that the AR(1) model is:

$$d(t+1) = \alpha d(t) + \mu + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma), |\alpha| < 1 \quad (116)$$

where α, μ and σ are parameters, and ε_t are i.i.d. random variables. We wish to derive equations for inventory for a stage that has the net replenishment time τ and faces such demand. First we note that

$$d(t+\tau) = \alpha d(t+\tau-1) + \mu + \varepsilon_{t+\tau} = \dots = \alpha^\tau d(t) + \sum_{j=0}^{\tau-1} \alpha^j (\mu + \varepsilon_{t+\tau-j}) \quad (117)$$

Now, using (117) we have

$$\begin{aligned}
\sum_{i=1}^{\tau} d(t+i) &= \sum_{i=1}^{\tau} \left(\alpha^i d(t) + \sum_{j=0}^{i-1} \alpha^j (\mu + \varepsilon_{t+i-j}) \right) \\
&= \sum_{i=1}^{\tau} \alpha^i d(t) + \sum_{i=1}^{\tau} \sum_{j=0}^{i-1} \alpha^j (\mu + \varepsilon_{t+i-j}) \\
&= \alpha \frac{1-\alpha^{\tau}}{1-\alpha} d(t) + \sum_{i=1}^{\tau} \frac{1-\alpha^i}{1-\alpha} \mu + \sum_{i=1}^{\tau} \sum_{j=0}^{i-1} \alpha^j \varepsilon_{t+i-j} \\
&= \alpha \frac{1-\alpha^{\tau}}{1-\alpha} d(t) + \frac{\mu}{1-\alpha} \left(\tau - \frac{1-\alpha^{\tau+1}}{1-\alpha} \right) + \sum_{i=1}^{\tau} \varepsilon_{t+i} \sum_{j=0}^{\tau-i} \alpha^j \\
&= \alpha \frac{1-\alpha^{\tau}}{1-\alpha} d(t) + \frac{\mu}{1-\alpha} \left(\tau - \frac{1-\alpha^{\tau+1}}{1-\alpha} \right) + \sum_{i=1}^{\tau} \frac{1-\alpha^{\tau-i+1}}{1-\alpha} \varepsilon_{t+i}
\end{aligned} \tag{118}$$

To proceed, we need to consider the distribution of $d(t)$, assuming that the demand process has reached steady state and we have no prior demand observations. We use (117) and note that ε_t is i.i.d.:

$$d(t) \sim N\left(\frac{\mu}{1-\alpha}, \sqrt{\sum_{i=0}^{\infty} (\alpha^i \sigma)^2}\right) = N\left(\frac{\mu}{1-\alpha}, \sqrt{\frac{\sigma^2}{1-\alpha^2}}\right) \tag{119}$$

We recall that in a base stock setting, we can state the inventory $I_k(t)$ at stage k at time t as:

$$I_k(t) = B_k - \sum_{j=t-S_k+1}^{t+T_k+S_k} d(j), \tag{120}$$

where B_k is the (constant) base stock level. Combining (118), (119) and (120) we can find the inventory variance as follows:

$$\begin{aligned}
\text{var}[I_k(t)] &= \\
&= \text{var} \left[\alpha \frac{1-\alpha^\tau}{1-\alpha} d(t) + \frac{\mu}{1-\alpha} \left(\tau - \frac{1-\alpha^{\tau+1}}{1-\alpha} \right) + \sum_{i=1}^{\tau} \frac{1-\alpha^{\tau-i+1}}{1-\alpha} \varepsilon_{t+i} \right] \\
&= \left(\alpha \frac{1-\alpha^\tau}{1-\alpha} \right)^2 \text{var}[d(t)] + \sum_{i=1}^{\tau} \left(\frac{1-\alpha^{\tau-i+1}}{1-\alpha} \right)^2 \text{var}[\varepsilon_{t+i}] \\
&= \left(\alpha \frac{1-\alpha^\tau}{1-\alpha} \right)^2 \frac{\sigma^2}{1-\alpha^2} + \sum_{i=1}^{\tau} \left(\frac{1-\alpha^{\tau-i+1}}{1-\alpha} \right)^2 \sigma^2 \\
&= \frac{\sigma^2}{(1-\alpha)^2} \left(\frac{\alpha^2(1-\alpha^\tau)^2}{1-\alpha^2} + \sum_{i=1}^{\tau} (1-2\alpha^{\tau-i+1} + \alpha^{2(\tau-i+1)}) \right) \\
&= \frac{\sigma^2}{(1-\alpha)^2} \left(\frac{\alpha^2(1-2\alpha^\tau + \alpha^{2\tau})}{1-\alpha^2} + \left(\tau - 2\alpha \frac{1-\alpha^\tau}{1-\alpha} + \alpha^2 \frac{1-\alpha^{2\tau}}{1-\alpha^2} \right) \right) \\
&= \frac{\sigma^2}{(1-\alpha)^2} \left(\frac{\alpha^2(2-2\alpha^\tau)}{1-\alpha^2} + \left(\tau - 2\alpha \frac{1-\alpha^\tau}{1-\alpha} \right) \right) = \\
&= \frac{\sigma^2}{(1-\alpha)^2} \left(\frac{\alpha^2(2-2\alpha^\tau) - 2\alpha(1+\alpha)(1-\alpha^\tau)}{1-\alpha^2} + \tau \right) \\
&= \frac{\sigma^2}{(1-\alpha)^2} \left(\frac{\alpha^2(2-2\alpha^\tau) - 2\alpha(1+\alpha-\alpha^\tau - \alpha^{\tau+1})}{1-\alpha^2} + \tau \right) \\
&= \frac{\sigma^2}{(1-\alpha)^2} \left(\frac{-2\alpha(1-\alpha^\tau)}{1-\alpha^2} + \tau \right)
\end{aligned} \tag{121}$$

Thus if we want to set safety stock so that it corresponds to a service level, we have:

$$\bar{I}_k = z \frac{\sigma}{1-\alpha} \sqrt{\frac{-2\alpha(1-\alpha^\tau)}{1-\alpha^2} + \tau} \tag{122}$$

This term should replace the term $z\sigma\sqrt{\tau}$ used in the i.i.d. case, which incidentally is recovered in the special case $\alpha = 0$ (this can be confirmed both by looking at the demand distribution itself, (116) and the resulting safety stock equation (122)).

We think that (122) can be used as a refinement of Simpson's formulation in practical cases. The parameter α is in fact equal to the period-to-period demand correlation ρ :

$$\begin{aligned}
\text{cov}(D(t), D(t+1)) &= \\
&= \text{cov}(D(t), \alpha D(t) + \mu + \varepsilon_{t+1}) \\
&= \text{cov}(D(t), \alpha D(t)) + \text{cov}(D(t), \mu + \varepsilon_{t+1}) \\
&= \alpha \text{var}(D(t))
\end{aligned} \tag{123}$$

Rearranging for α gives us:

$$\begin{aligned}
\alpha &= \frac{\text{cov}(D(t), D(t+1))}{\text{var}(D(t))} \\
&= \frac{\text{cov}(D(t), D(t+1))}{\sqrt{\text{var}(D(t)) \text{var}(D(t+1))}} \\
&= \rho(D(t), D(t+1))
\end{aligned} \tag{124}$$

Of course there is no guarantee that demand follows exactly the AR(1) process; but it would seem like this is a sensible refinement of Simpson's model, which in effect amounts to the assumption that the period to period demand correlation is zero. The formulation (122) requires that one more parameter $\alpha = \rho$ be estimated but will presumably result in better inventory allocation when demand is not i.i.d. Estimating $\alpha = \rho$ from historical data can easily be done with standard spreadsheet functions.

Bibliography

- Abhyankar H. and Graves S.C. 2001. Creating an inventory hedge for Markov-modulated Poisson demand: An application and model. *Manufacturing & Service Operations Management*. Vol. 3, No. 4, Fall, 306-320
- Aviv Y. 2003. A time-series framework for supply-chain inventory management. *Operations Research*. Vol. 51, No. 2, March–April, 210–227
- 2004. Collaborative forecasting and its impact on supply chain performance. Simchi-Levi D., Wu S.D., Shen Z. J., Handbook of quantitative supply chain analysis. Kluwer Academic Publishers, Chapter 10
- 2007. On the benefits of collaborative forecasting partnerships between retailers and manufacturers. *Management Science*. Vol. 53, No 5, 777-794
- Axsäter S. and Rosling K. 1991. Notes: Installation vs. Echelon Stock Policies for Multilevel Inventory Control. *Management Science*, Vol. 39, No. 10. 1274-1280.
- Baker, K. R. 1993. Requirements planning. S. C. Graves, A. H. Rinnooy Kan, P. H. Zipkin, eds. Handbooks in operations research and management science. Vol. 4., Logistics of production and inventory. North-Holland Publishing Company, Amsterdam, The Netherlands, Chapter 3.
- Berling P. and Marklund J. 2006. Heuristic Coordination of Decentralized Inventory Systems Using Induced Backorder Costs *Production and Operations Management*. Vol. 15, No. 2. 294-310.
- Bertsimas D. and Thiele A. 2004. A robust optimization approach to supply chain management. Integer Programming and Combinatorial Optimization, Springer Berlin / Heidelberg, 86-100
- Bertsimas D. and Thiele A. 2006. A robust optimization approach to inventory theory. *Operations Research*. Vol. 54, No. 1, 150-168
- Billington C., Callioni G., Crane B., Ruark J. D., Rapp J. U., White T. and Willems S. P. 2004. Accelerating the profitability of Hewlett-Packard's supply chains. *Interfaces*. Vol. 34, No. 1., January-February, 59-72
- Buzacott J. A., Shanthikumar J.G. 1994. Safety stock versus safety time in MRP controlled production systems. *Management Science*. Vol. 40, No. 12., December.
- Cachon G.P. 2003. Supply Chain Coordination with Contracts S. C. Graves, A. G. de Kok, eds. *Handbooks in Operations Research and Management Science, Vol 11., Supply Chain Management: Design, Coordination and Operation* Elsevier, Amsterdam, The Netherlands, Chapter 6.
- ., Netessine S. 2004. Game Theory in Supply Chain Analysis. Simchi-Levi D., Wu S.D., Shen Z. J., *Handbook of quantitative supply chain analysis* Kluwer Academic Publishers, Chapter 2
- Carlson, R.C., and Yano C.A. 1986. Safety stocks in MRP-Systems with emergency setups for components. *Management Science*. Vol 32. No. 4, April.
- Chen F. 2003. Information Sharing and Supply Chain Coordination S. C. Graves, A. G. de Kok, eds. *Handbooks in Operations Research and Management Science, Vol 11., Supply Chain Management: Design, Coordination and Operation* Elsevier, Amsterdam, The Netherlands, Chapter 7.

- Chu , C-L., and Leon V.J. 2004 .Distributed Inventory Coordination for Multi-Echelon Distribution Systems. *Proceedings of 13th Annual Institute of Industrial Engineers Research Conference*. Houston, Texas, May 15-19.
- Clark A.J. and Scarf H. 1960. Optimal policies for a multi-echelon inventory problem, *Management Science* Vol. 6, 475-490
- Corbett C. J., Karmarkar U. S., 2001. Competition and Structure in Serial Supply Chains with Deterministic Demand, *Management Science* Vol. 47, No. 7, July 2001 pp. 966–97
- Diks, E. B., de Kok A. G., Lagodimos A. G. 1996. Multi-echelon systems: A service measure perspective. *European Journal of Operations Research*. Vol. 95, 241–263.
- Federgruen, A. 1993. Centralized planning models for multi-echelon inventory systems under uncertainty. S. C. Graves, A. H. Rinnooy Kan, P. H. Zipkin, eds. Handbooks in operations research and management science. Vol 4., Logistics of Production and Inventory. North-Holland Publishing Company, Amsterdam, The Netherlands, Chapter 3.
- Gallego G. and Scheller-Wolf A. 2000. Capacitated inventory problems with fixed order costs: Some optimal policy structure. *European Journal of Operations Research*. Vol. 126. 603-613
- , Toktay B.L. 2004. All-or-Nothing ordering under a capacity constraint. *Operations Research*. Vol. 52. November-December. 1001-1002.
- , Özer Ö. 2001. Integrating replenishment decisions with advance demand information *Management Science*. Vol. 47, No. 10, 1344-1360
- Glasserman P. and Tayur S. 1994 The stability of capacitated, multi-echelon production-inventory system under base-stock policy *Operations Research* Vol. 42, No. 5, September-October 1994
- ,1995 Sensitivity analysis for base-stock levels in multiechelon production-inventory systems. *Management Science*. Vol. 41. 263-281
- 1996. A Simple Approximation for a Multistage Capacitated Production-Inventory System. *Naval Research Logistics*, Vol. 43, Issue 1 (p 41-58)
- Graves S.C., Meal H.C., Dasu S., Qiu Y.1986. Two-stage production planning in a dynamic environment. S. Axsäter, C. Schneeweiss and E. Silver, eds. Multi-Stage Production Planning and Inventory Control. Springer.
- , D. B. Kletter, W. B. Hetzel. 1998. A dynamic model for requirements planning with application to supply chain optimization. *Operations Research*. Vol. 46, S35–S49.
- , Willems S. P. 2000. Optimizing strategic safety stock placement in supply chains. *Manufacturing & Service Operations Management*. Vol. 2, No. 1, Winter, 68-83.
- , Willems S. P. 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.
- , Willems S. P. 2008, Strategic Inventory Placement in Supply Chains: Non-Stationary Demand, *Manufacturing & Service Operations Management*, Vol. 10, No. 2, Spring, 278–287
- Griffel D.H. 1985. Applied functional analysis. John Wiley & Sons, New York.

- Grimmett G. and Stirzaker D. 2001. Probability and random processes. Oxford University Press.
- Guide V.D.R. Jr and Srivastava R. 2000. A review of techniques for buffering against uncertainty in MRP systems. *Production Planning and Control*. Vol. 11, No. 3, 223-233.
- Gupta D. and Selvaraju N. 2006. Performance evaluation and stock allocation in capacitated serial systems. *Manufacturing & Service Operations Management*, Vol. 8, No. 2, Spring, 169–191
- Hamilton J. D. 1994 Time series analysis. Princeton University Press.
- Heath, D.C., and P.L. Jackson 1994. Modeling the evolution of demand forecasts with application to safety stock analysis in production/distribution systems. *IIE Transactions*. Vol. 26, No. 3, 17-30.
- Humair S. and Willems S. P. 2007. Optimizing strategic safety stock placement in general acyclic networks. Working paper.
- Inderfurth, K. 1991. Safety stock optimization in multi-stage inventory systems. *International Journal of Production Economics*. Vol. 24. 103-113.
- Karaesmen F., Buzacott J.A., Dallery Y. 2002. Integrating advance order information in make-to-stock production systems. *IIE Transactions*. Vol. 34, 649–662
- Kimball, G. E. 1988. General Principles of Inventory Control. *Journal of Manufacturing and Operations Management*. Vol. 1. 119-130.
- Lagodimos A.G., and Anderson E.J. 1993. Optimal positioning of safety stocks in MRP. *International Journal of Production Research*. Vol. 31, No.8, 1797-1813
- Lambrecht M. R., Muckstadt J.A., and Luyten R. 1984. Protective stocks in multi-stage production systems. *International Journal of Production Research*. Vol. 22, No.6, 1001-1025
- Lee H. and Whang J. 1999. Decentralized Multi-Echelon Supply Chains: Incentives and Information *Management Science* Vol. 45, No. 5., 633-640
- Lesnaia E. 2004 Optimizing Safety Stock Placement in General Network Supply Chains. PhD Thesis, Massachusetts Institute of Technology
- Lesnaia, E., Vasilescu, I., & Graves, S. C. 2005. The complexity of safety stock placement in general-network supply chains. *Proceedings of the 2005 SMA Conference*. Singapore. 5.
- Molinder A. 1997. Joint optimization of lot-sizes, safety stocks, and safety lead times in an MRP system. *International Journal of Production Research*. Vol. 35, No.4, 983-994
- Muthoo A. 1999. *Bargaining Theory with Applications*. Cambridge, United Kingdom.
- Nash, J. F. 1950. The Bargaining Problem. *Econometrica*. 18. 155-162.
- Osborne M. J. and Rubinstein A. 1994. *A Course in Game Theory*. MIT Press. Cambridge, Massachusetts.
- Parker R. P., Kapuscinski R., 2004. Optimal policies for a capacitated two-echelon inventory system. *Operations Research*. Vol. 52. September-October. 739-755
- Simpson, K. F. 1958. In-process inventories. *Operations Research*. Vol. 6, 863–873.
- Yano C. A., Carlson R. C. 1987. Interaction between frequency of rescheduling and the role of safety stock in material requirements planning systems. *International Journal of Production Research*. Vol. 25, No.2, 221-232

Zhang X. 2004. Evolution of ARMA Demand in Supply Chains. *Manufacturing & Service Operations Management*. Vol. 6, No. 2, Spring, 195–198