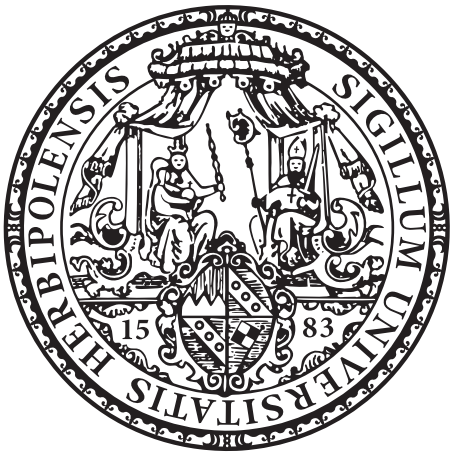
Julius-Maximilians-Universität Würzburg

Wirtschaftswissenschaftliche Fakultät



Inaugural Dissertation

To obtain the academic degree

Doctor rerum politicarum (Dr. rer. pol.)

Allocation Planning in Sales Hierarchies

Konstantin Kloos

December 17, 2019

Author:
Konstantin Frederic Pascal Kloos, M. Sc.

Friedenstraße 30
97072 Würzburg

1. Supervisor:
Prof. Dr. R. Pibernik

2. Supervisor:
Prof. Dr. C. Flath

Acknowledgments

My first thanks go to my doctoral advisor, Prof. Dr. Richard Pibernik. Channeling my creativity into four structured and scientific rigorous studies was surely a demanding task and I cannot thank Prof. Pibernik enough for his persistent support. I enjoy looking back to our various projects and want to thank him for the trust and the confidence he put in me. I am also very grateful for getting the repeated opportunity to meet and exchange with researchers from all over the world.

Secondly, I want to thank Prof. Dr. Christoph Flath for serving as my second supervisor and for the challenging and always interesting debates we enjoyed on many occasions.

I want to thank Dr. Benedikt Schulte for his outstanding support in my first paper. His input has drastically improved the paper and I have learned a lot from our intense exchange on various topics. My special thanks goes to my dear colleagues at the Chair of Logistics and Quantitative Methods—they are the best colleagues one can imagine. I will miss our joint lunches, numerous coffee breaks and our conversations on both research-related and personal subjects. I hope that we all will stay in touch. Next, I want to thank my student assistant, Niels Westphal, for his valuable help in structuring and implementing my numerical experiments in Python.

I also want to thank my research project team, Dr. Jaime Cano-Belmán, Prof. Dr. Moritz Fleischmann, Prof. Dr. Herbert Meyr, and Maryam Nouri. Working with you was a pleasure. I enjoy looking back to our inspiring and productive project meetings and our joint trips to the OR conferences. At this point, I also want to thank the Deutsche Forschungsgemeinschaft (DFG), which sponsored my research project under grant PI 438/5-1.

Finally, I want to thank my beloved wife. Bin, your love, care and support was what kept me going. You were always encouraging and never doubted me in my academic journey.

Contents

List of Tables

List of Figures

List of Abbreviations

AAE average allocation efficiency

aATP allocated available-to-promise

AGO absolute gap to optimality

APS advanced planning systems

arpg average relative profit gap

ATP available-to-promise

B2B business to business

CAP customer allocation problem

cdf cumulative distribution function

CV coefficient of variation

DAP deterministic allocation policy

DE decentralization error

DF demand fulfillment

FCFS first-come-first-served

HAP hierarchy allocation problem

iid independent identically distributed

LP linear program

LSO local sales organization

M-DP myopic approach with dynamic penalty approach

M-P myopic approach with penalty-based approach

M-PC myopic approach with per commit

M-SDP myopic approach with smoothed dynamic penalty approach

MC central myopic allocation policy

MPAP myopic penalty-based allocation policy

MSAP myopic stochastic allocation policy

MSLAP myopic service-level-based allocation policy

MTO make-to-order

MTS make-to-stock

NAP normalized additional penalty

PC per commit allocation

pdf probability density function

rae relative allocation error

RAGO relative average gap to optimality

RDAP randomized deterministic allocation policy

RLP randomized linear program

RM revenue management

rpg relative profit gap

SL service level

STAP stochastic time-aggregated allocation policy

Deutschsprachige Zusammenfassung

Lieferanten nutzen Service Levels, um ihre Leistung gegenüber ihren Kunden zu überwachen. Um der Unterschiedlichkeit der Kunden gerecht zu werden, streben sie häufig unterschiedliche Leistungsstufen gegenüber ihren Kunden an. So erhalten beispielsweise Kunden mit einer hohen strategischen Bedeutung für den Lieferanten oder einer höheren Zahlungsbereitschaft einen besseren (d.h. zuverlässigeren) Service als andere. Diese Kundendifferenzierung ist insbesondere wichtig, wenn die Kapazitäten oder Bestände knapp werden und der Lieferant entscheiden muss, welche Kunden er noch bedient und welche Aufträge er verschiebt oder gar ablehnt.

Diese Differenzierung und Priorisierung erfolgt in Unternehmen meist durch eine Allokationsplanung, bei der geplante Kapazitäten/Bestände einzelnen Kunden oder Kundengruppen zugeordnet werden. Bestellungen der Kunden werden dann solange angenommen und erfüllt, bis die dem jeweiligen Kunden zugeordnete Allokation aufgebraucht ist. In der Praxis erfolgt diese Allokationsplanung zumeist nicht durch einen zentralen Planer, sondern entlang der hierarchischen Struktur der Vertriebsorganisation. So werden die gesamt verfügbaren Kapazitäten schrittweise weiter heruntergebrochen bis schlussendlich die Allokationen zu den Kunden bestimmt werden. Diese schrittweise und dezentrale Planung erfolgt üblicherweise mittels einfacher Regeln, auch da die meisten der in der Literatur vorgeschlagenen Modelle nicht für diese dezentrale Planung geeignet sind.

Motiviert von dieser Implementierungslücke, befasst sich diese Dissertation mit der dezentralen Allokationsplanung in Unternehmen. In vier abgeschlossenen Ka-

piteln werden verschiedene Aspekte dieser unternehmerischen Planungsaufgabe beleuchtet.

In Kapitel 2 wird zunächst ein einperiodiges Model untersucht, in dem der Lieferant seine Kunden durch Alpha-Service Level differenziert. Für dieses Model wird gezeigt, wie optimale Allokationen zentral und dezentral errechnet werden können. Da die dezentrale optimale Lösung sich jedoch als impraktikabel herausstellt, werden die heute weit verbreiteten einfachen Allokationsregeln untersucht. Mathematische Untersuchungen ergeben, dass diese einfachen Regeln typischerweise zu suboptimalen Allokationen führen. Daher werden zwei neue Ansätze entwickelt, für die analytisch und numerisch gezeigt wird, dass ihre Allokationen nahe dem zentralen Optimum sind.

Kapitel 3 untersucht ein ähnliches Model; hier unterscheiden sich die Kunden jedoch in ihrer Zahlungsbereitschaft. Der Fokus der Analyse in Kapitel 3 liegt auf der Art von Informationen, die innerhalb der Organisation geteilt werden müssen, um gute Allokationen zu erhalten. Durch eine weitgreifende numerische Analyse kann gezeigt werden, dass Informationen über die Heterogenität der Zahlungsbereitschaft der Kunden und Informationen über die Nachfrageunsicherheit besonders relevant sind. Es werden zwei neue Methoden entwickelt, die „Stochastische Theil-index Methode“ und die „Clustering Methode“, welche auf eben diese Informationen zurückgreifen und deren Allokationen nahezu optimal sind.

Kapitel 4 betrachtet die Allokationsplanung für Lieferanten, die einen bestimmten Typ von Service-Level-Vertrag mit ihren Kunden abgeschlossen haben, welcher besonders in der Industrie verbreitet ist. Hierbei einigen sich die Vertragspartner auf ein bestimmtes Leistungsniveau, das der Lieferant über einen definierten Zeitraum (Berichtszeitraum) erreichen muss. Unterschreitet der Lieferant das vereinbarte Leistungsniveau, wird eine Strafzahlung fällig, die sich nach der Höhe der Abweichung richtet. Nach der Modellierung des Problems wird die optimale Allokationspolitik für den Lieferanten als dynamisches Programm charakterisiert. Wegen der großen Komplexität und des einsetzenden „Curse of Dimensionality“ kann diese Politik jedoch nicht numerisch berechnet werden. Dennoch können aus der Analyse Anforderungen abgeleitet werden, die eine gute Politik erfüllen sollte. Anhand dieser Anforderungen werden bisherige Ansätze aus Literatur und Praxis evaluiert, insgesamt vier neue Allokationspolitiken entwickelt und in einem Simulationsexperiment auf ihre Leistung überprüft.

Während Kapitel 4 noch von einem zentralen Planer ausgeht, wird in Kapitel 5 eine dezentrale Planung entlang der Vertriebsorganisation angenommen. Es

wird gezeigt, dass sich das Problem in zwei hierarchisch verbundene Teilprobleme separieren lässt. Das eine Teilproblem umfasst die Planung auf der untersten Ebene der Vertriebshierarchie, wo die Zuordnung der Kapazität zu einzelnen Kunden geschieht. Hier können die in Kapitel 4 entwickelten Methoden direkt angewendet werden. Das zweite Teilproblem umfasst die Planung auf höheren Ebenen der Hierarchie bei der noch keine direkte Zuordnung zu Kunden geschieht. Für dieses Problem wird gezeigt, wie mit einigen Modifikationen die in Kapitel 3 entwickelten Methoden für die Planung verwendet werden können. Die entstehenden Allokationssysteme (bestehend aus den Lösungsansätzen für beide Teilprobleme) werden erneut in einem Simulationsexperiment überprüft. Die Resultate zeigen, dass bei Verwendung von geeigneten Ansätzen auf höheren Ebenen ähnliche Allokationen wie unter zentraler Planung erreicht werden können.

Kapitel 6 ordnet die Ergebnisse der einzelnen Kapitel ein und gibt Anregungen für weitere Untersuchungen.

Chapter 1

Introduction

Many suppliers use service levels to monitor their performance towards their customers. Because customers differ in their service preferences, profitability and/or strategic importance for the supplier, a supplier will oftentimes pursue different service levels for different customers. Especially in situations where supply or capacity is scarce, these customer-specific service-level targets allow the supplier to differentiate its customer-faced performance and prioritize demands of customers with higher importance.

In state-of-the-art advanced planning systems (APS) this prioritization in demand fulfillment is realized by a two-stage process: In the first step, "allocation planning," the supplier allocates the planned supply from master planning and/or production planning to individual customers or groups of customers (cf. Kilger and Meyr, 2015). These dedicated allocations are the input to the second step, "order promising" (Ball et al., 2004). During order promising the customers' orders arrive and are fulfilled by the supplier until the corresponding allocations are consumed. Consequently, allocation planning crucially affects the supplier's performance towards its customers: If an allocation is chosen too high, valuable supply may be left unconsumed and is missing for fulfilling the demands of other customers; choosing an allocation too small leads to unfilled demand that results in profit losses, penalty payments, or, at least, unsatisfied customers.

The literature on allocation planning can be broadly divided in profit-based and service-level-related literature. Profit-based models reflect the differences in customer importance by customer-specific unit-profits and, thus, the objective is to

maximize the supplier's revenue. Ball et al. (2004), who refer to allocation planning as "Push-based ATP," propose first stochastic and deterministic models for profit-based allocation planning. They point out, that stochastic allocation planning can be regarded as a special case of quantity based revenue management. Meyr (2009) shows on an ordered data set from the lighting industry that deterministic allocation planning significantly increases profits as compared to fulfilling orders first-come-first-served. Quante et al. (2009a) consider stochastic demand and propose a stochastic dynamic programming model that allocates supply to customers who differ in their unit profits. They benchmark their approach against that of Meyr and show that it leads to significantly higher profits at higher levels of demand uncertainty. Eppler (2015) extends the approach to incorporate nesting among different customer classes.

Although most research on allocation planning focuses on profit-maximization, some researchers also consider service-levels. For instance, Pibernik and Yadav (2008) and Pibernik and Yadav (2009) determine allocations for a planner that wants to ensure a minimum service level for a high priority customer group.

The literature on inventory rationing shares some similarities with allocation planning: both address the question of how to optimally allocate supply to groups of customers. (See, i.e., Deshpande et al., 2003; Arslan et al., 2007; Schulte and Pibernik, 2016.) The difference between the two streams is in the way supply is treated: In inventory rationing supply is a decision variable and can be freely adjusted by the planner; in allocation planning supply is given by master planning/production planning and is, thus, fixed in the short term and cannot be adjusted.

A common assumption across publications on allocation planning is that there is a single planner with the ability to decide on the allocations to all customers simultaneously. We refer to this as the central allocation planning problem. In many companies, however, there does not exist a central planner and, instead, allocation planning is a decentral process aligned with the company's multi-level hierarchical sales organization ("sales hierarchy") (Kilger and Meyr, 2015). For instance, in a company with a geography-based hierarchy, first, high-level sales managers will decide how to share the supply among different regions (i.e., America, Europe, Asia). Then regional planners may further disaggregate supply and allocate it to country managers (responsible for, i.e., USA or Canada). At some point, planners in local sales organizations (LSOs) assign the received supply to individual

customers or groups of customers. We refer to this as the decentral allocation planning problem.

This gap between the central planning assumed in the literature and the decentral planning in companies has first been observed by Roitsch and Meyr (2015) in a case study from the oil industry. Currently, companies typically resort to simple rules for allocation planning. An example for such a simple rule is per commit, under which supply is shared evenly among the customers based on their expected demand (Kilger and Meyr, 2015). While the approach is easy to use, it does not prioritize and suppliers can typically not achieve the desired service differentiation. Vogel and Meyr (2015) are among the first to address this problem of decentral allocation planning in sales hierarchies. For a single-period setting with deterministic demand they show that the simple rules used in practice lead to a detrimental performance. Consequently they propose a new allocation approach that uses the Theil index (a measure of income inequality from the economic literature) to measure the profit heterogeneity between the customer groups. This leads to a better prioritization of demands from customers with higher profits. Recently, Cano-Belmán and Meyr (2019) have extended the approach to consider multiple planning periods.

While Vogel and Meyr (2015) and Cano-Belmán and Meyr (2019) provide valuable insights into decentral allocation planning, their approaches are still limited in that they only consider a deterministic setting. This dissertation is part of a larger DFG-sponsored project with research teams from the University of Mannheim and the University of Hohenheim to analyze decentral allocation planning in-depth and under different objectives. In particular, this dissertation addresses allocation planning in multi-level sales hierarchies as described above with the objective of achieving customer-specific service-level targets.

Our objectives are to 1) develop new decentral allocation approaches for our specific setting, 2) quantify the performance gap of the simplistic approaches currently applied in APS and of our new-developed allocation approaches compared to central planning, and 3) identify the information crucial to good decentralized allocation decisions. We pursue these objectives in two steps: Chapter 2 and 3 address decentral allocation planning in a simpler single-period setting. In contrast to previous research, botch chapters consider the stochasticity of customer demand. The reduced complexity of the single-period setting allows us to obtain analytical results on the performance of different allocation approaches and structural insights on which information is required for obtaining good allocations. Chap-

ter 4 and 5 analyze a more realistic setting, where allocation planning is carried out over multiple-periods and the supplier is engaged in so-called service-level contracts with its customers.

More specifically, Chapter 2 or Kloos et al. (2018) analyzes the decentral allocation problem in a single-period setting where planners aim to achieve customer-specific alpha-service-level targets. We show how to compute optimal (central) allocations and prove that the optimal allocation can also be achieved by decentral planning. However, the extensive information sharing required for the decentral optimal allocation makes it infeasible for most practical applications. Based on our insights from the central problem, we analyze the allocations generated by conventional allocation rules and develop two new allocation approaches, the "hybrid approach" and the "service-level aggregation approach." Based on a rigorous analytical and numerical analysis, we find that two types of customer heterogeneity are decisive for the performance of the allocation approaches. With "within heterogeneity" we describe the heterogeneity that occurs between customers within the same sub-tree, while with "between heterogeneity" we account for the differences between the individual sub-trees of the sales hierarchy. If between and within heterogeneity are low, the per commit rule we described earlier already leads to close-to-optimal allocations; if only between heterogeneity is low, our newly developed hybrid approach shows the best performance; under low within heterogeneity the service-level aggregation approach leads to allocations that are almost optimal. In hierarchies that exhibit both high within and high between heterogeneity our new approaches still perform significantly better than conventional approaches but decision makers may want to use allocation approaches that guarantee optimal allocations.

Based on our collaboration within the research project we identified a strong structural similarity between the problem analyzed in Chapter 2 and the problem of maximizing the suppliers profit in the single-period setting. Consequently Chapter 3 or Fleischmann et al. (2019) is joint work with the team from University of Mannheim. While it addresses maximizing customers' profits, the focus of the analyses presented in this chapter is on the information sharing required by different allocation approaches and their respective performance. Besides, again, analyzing per commit as a minimal-information benchmark, we develop and evaluate three more advanced allocation approaches: the "deterministic Theil approach" from Vogel and Meyr (2015), the "stochastic Theil approach," a stochastic extension of the approach developed by Vogel and Meyr and the "clustering approach," in

which the customers are clustered according to their respective profits. By comparing the approaches numerically, we identify four types of information relevant for obtaining "good" allocations: mean demand, demand uncertainty, profits and profit heterogeneity. Both the stochastic Theil approach and our clustering approach use this information and, consequently, their performance is close to optimal independent of the specific setting. Especially our results on the clustering approach are highly relevant: They suggest that a relatively simple clustering logic with three clusters (high, average and low profit customers) and some additional information on the aggregated demand distributions of the clusters is sufficient to obtain virtually the same profit as a centralized full-information approach, which is typically not feasible in practice.

As Sieke et al. (2012) explain, service-level contracts become more and more important in the B2B relationships between manufacturers and their customers. So instead of implicit service-level targets that result from the suppliers evaluation of its customers' importance, under a service-level contract the supplier and its customer explicitly agree on the desired performance level, specify a time period over which the performance is reviewed and penalties for deviations. While service-level contracts have been addressed in the literature before, they have so from an inventory management perspective and as such, supply is part of the planner's decision. Chapter 4 and 5 analyze service-level contracts from the perspective of allocation planning, where supply is essentially fixed and cannot be adjusted.

Chapter 4 or Kloos and Pibernik (2020) analyzes allocation planning under service-level contracts in a central setting, where there is a single planner with complete information on all the customers' past performance and service-level contracts. We formalize the problem as a stochastic dynamic program and characterize its optimal solutions. Due to the ensuing "curse of dimensionality," however, computing the optimal policy is infeasible. Nonetheless, our analytical insights allow us to identify the factors that affect the optimal decision: The customers' fill-rate targets and penalties, the companies performance toward the customer in the past and the customers' demand distribution in the current and all future allocation periods. Based on these factors we propose several deterministic and stochastic policies, compare them with myopic approaches that have been suggested in the relevant literature and evaluate their performance numerically. Our results show that a stochastic myopic allocation typically outperforms competing approaches and leads to relatively small gaps compared to an ex-post optimization.

In Chapter 5 or Kloos (2019) we extend the problem from Chapter 4 to a hierarchical setting with decentral decision. To this end, we decompose the allocation problem into two interrelated sub-problems: The first problem is that of the LSOs at the lowest level of the sales hierarchy which have complete information on their customers' service-level contracts but are constrained in their allocations by the supply they receive from higher levels of the hierarchy. We call this the customer allocation problem (CAP) and we can straightforwardly apply the approaches from Chapter 4. The second problem is that of the planners in the hierarchy, who have only aggregate information on the service-level contracts and only decide on the allocations to the LSOs. We call this the hierarchy allocation problem (HAP). As the objectives differ, the approaches developed in Chapter 2 and 3 cannot be readily applied to the HAP. We are, however, able to extract relevant information from the service-level contracts and develop two approaches that allow to use the clustering method (from Chapter 3) for the HAP. We evaluate the performance of the resulting allocation systems (comprised of the approaches applied for the CAP and the HAP) in two settings. In the first setting, we assume that leftover inventory or backlog are cleared centrally. In this setting, our results in-line with our observations from Chapter 3: When applying a suitable decentral allocation approach, there is little to no difference between decentral and central allocation planning. In the second setting, the LSOs are responsible for clearing inventory and backlog. Then, we see a significant difference between central and decentral planning as inventories and backlog build up in individual LSOs.

The results presented in the four main chapters of the dissertation have immediate relevance to suppliers performing decentral allocation planning: Decision makers can learn when the currently applied simple rules have a negative impact on the performance of the company and which alternative rules they can use that promise a superior performance. In addition, Chapter 4 is relevant to suppliers having entered service-level contracts and provides decision makers with policies that can drastically reduce the penalties resulting from service-level deviations.

Table 1.1 provides an overview of the scientific contribution of the four main chapters in this dissertation.

Table 1.1: Overview of scientific contribution

	Alloc. Plan. in Sales Hierarchies w. Stoch. Demand and SL-Targets	**Single-Period Stoch. Demand Fulfillment in Cust. Hierarchies**	**Alloc. Plan. under SL-Contracts**	**Mang. SL-Contracts in Sales Hierarchies**
	Chapter 2, p. 13	*Chapter 3, p. 63*	*Chapter 4, p. 95*	*Chapter 5, p. 145*
Analytical model	• Minimize deviations from alpha-service-level targets • Single-period • Multi-level hierarchy	• Maximize profit • Single-period • Multi-level hierarchy	• Minimize penalties from service-level contracts • Multi-period • Central Planning	• Minimize penalties from service-level contracts • Multi-period • Multi-level hierarchy
Methodological contribution	• First study to address hierarchical allocation planning under service-level contraints. • Analytical analysis of popular allocation rules. • Two new allocation heuristics that improve performance.	• Extends hierarchical allocation planning to stochastic demand. • Formalizes information sharing in decentral planning settings. • Two new allocation methods that only require limited information sharing.	• First study to address allocation planning under service-level contracts. • Characterization of the optimal allocation policy. • New myopic allocation policy significantly reduces expected penalties.	• Decomposition of the hierarchical allocation problem in two sub-problems. • Numerical analysis of the performance losses from decentral planning for several allocation approaches.
Conceptional findings	• Optimal allocations can be achieved with decentral planning • Two forms of heterogeneity (*between* and *within* heterogeneity) are decisive for the performance of decentral allocation approaches.	• Information on the customers' heterogeneity and demand stochasticity are decisive to obtain "good" allocations. • The *clustering approach* leads to close to optimal allocations independent of the specific setting.	• Information on the customers demand distribution and their penalty heterogeneity are most important for good allocations. • Our *myopic stochastic allocation policy* significantly reduces expected penalties.	• Decentral inventory clearing critically affects the performance of decentral planning • Under central inventory clearing, decentral alloc. planning performs similar to central planning.

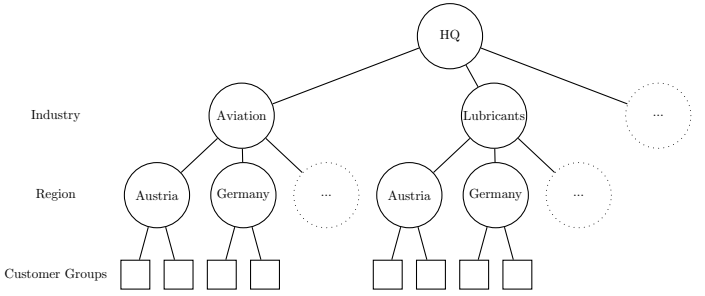
Chapter 2

Allocation Planning in Sales Hierarchies with Stochastic Demand and Service-Level Targets[1]

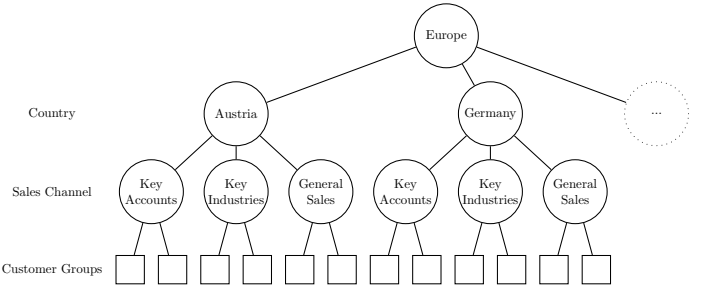
2.1 Introduction

Optimally matching supply with demand is a challenging task in many manufacturing environments, especially when purchasing and production must be planned with sufficient lead time, demand is uncertain, overall supply may not suffice to fulfill all of the projected demand, and customers differ in their level of importance. The particular structure of the sales organization frequently adds another layer of complexity, as sales organizations often have multi-level hierarchical structures (in brief, "sales hierarchy") that may include multiple geographic sales regions, distribution channels, customer groups, and individual customers (e.g., key accounts), each with its own priorities and profitabilities. The structure and

[1] This chapter was published in *OR Spectrum* as Kloos et al. (2018) and is co-authored by Benedikt Schulte and Richard Pibernik.



(a) Oil and gas company



(b) Technology company

Figure 2.1: Examples of sales hierarchies.

composition of such sales hierarchies can vary substantially. Figure 2.1 illustrates two examples.

Example (a) in Figure 2.1 (cf. Roitsch and Meyr, 2015) refers to an international oil and gas company that faces raw-material lead times that are much longer than the planning interval, along with inflexible capacities in its refineries. Demand and supply planning take place in a four-level hierarchy that involves industry segments (e.g., aviation, lubricants) on the second level, countries on the third level, and individual customers (with differing characteristics in terms of demand levels, demand uncertainty, service-level targets, etc.) on the fourth level. Example (b) refers to a large multinational technology company that structures its sales organizations by country on the second level. Within each country are sales teams, each of which is assigned to a particular market segment: one for key accounts, one targeting certain industries (e.g., the automotive industry), and one for "general

sales"—that is, for everyone else. While the number of customers and the characteristics of the segments (e.g., profitabilities, service-level targets) vary significantly among the various markets, the characteristics of each of the market segments are similar in each country.

For demand and supply planning, the two companies in Figure 2.1 follow a three-step approach, which is common among companies with hierarchical sales organizations: First, available supply is planned based on aggregate demand forecasts and available resources for the medium term (master production planning). Then, in the "allocation planning" step (Stadtler et al., 2015), the planned supply is allocated throughout the hierarchical multi-level sales organization. Finally, the allocated supply is consumed on the lowest level of the sales organization (e.g., in a particular customer region) as actual customer orders materialize ("order promising," cf. Ball et al., 2004).

The focus of the research presented in this paper is on allocation planning in companies with hierarchical sales organizations and whose customers have heterogeneous demand characteristics and differ in their level of importance (reflected by their individual service-level targets). In particular, we address settings in which supply is scarce—that is, supply is not sufficient to match the service-level targets of all of the company's customer groups. As most multi-national companies have hierarchically structured sales organizations, work with individualized service-level targets for their various customer groups, and face periods of supply scarcity, such a setting is often encountered in practice.

In theory, it is relatively easy to solve the hierarchical allocation planning problem if we assume an omniscient central planner who has full information about the demand distributions and well-defined preferences over the (realized) service levels for different customer groups. In such a case, an optimal allocation plan can be derived by solving a stochastic knapsack problem.

However, allocation planning is typically not performed by an omniscient central planner who has all information he or she needs. Instead, companies have iterative planning processes in which information about customer demand is gradually aggregated and then shared from lower to higher-level planners in the sales hierarchy before the aggregated supply is gradually broken down and allocated from higher to lower levels. For instance, in Example (a) (Figure 2.1), each country organization reports demand forecasts and service-level targets, which are first aggregated at the industry level before being communicated to the company's headquarters. Subsequently, the planner in the headquarter plans the available supply

and allocates it to the industry clusters. Then planners in the industry organizations split their allocations among the country organizations for which they are responsible, and planners in local sales organizations perform yet another allocation to the customer groups or individual customers in their respective countries.

This iterative and largely decentralized process is supported by Advanced Planning Systems (APS), such as SAP's APO. The systems support the planners on the various levels of the sales hierarchy by aggregating demand information (bottom-up) and providing recommendations for the allocations (top-down). As Kilger and Meyr (2015) explain, APS usually employ relatively simple allocation approaches (rules) that are based on limited information about the levels of the sales hierarchy. The most commonly used allocation rules are *per commit* and *rank based*. Under *per commit* supply is allocated proportionally to the demand forecast. Under *rank based* the customer groups are first ranked according to some priority measure; supply is then allocated in ascending order of the rank. (See Section 2.5 for a formal description and discussion.) The key benefits of these rules are that they are easy to understand, communicate, and put into practice, and they require only limited information to be shared throughout the hierarchy. However, these benefits come at a cost, as these allocation rules usually lead to significantly lower performance as compared to the theoretical optimum that would be achieved by an omniscient central planner.

Therefore, the first objective of our research is to determine when these conventional allocation rules lead to optimal (or at least acceptable) results and to characterize their optimality gap relative to the theoretical optimum. Our analysis suggests that the conventional allocation rules lead to optimal results only under very restrictive conditions and that the loss in optimality is often substantial. This result leads us to pursue our second objective: to find alternative (decentral) allocation approaches that generate acceptable performance under conditions in which the conventional allocation rules lead to poor results. Based on our results and findings, we develop two advanced allocation approaches that exploit more of the relevant information about the customer groups. We term them *hybrid* approach and *service level aggregation* approach. We provide a formal characterization of the two approaches and show under what conditions they lead to optimal allocations. Based on numerical analyses, we find that these alternative approaches outperform the conventional allocation rules, independent of the conditions under which they are used.

Certainly, our advanced allocation approaches are not as simple and intuitive as the conventional allocation rules are. Practitioners can benefit from knowing when it is "safe" to rely on conventional allocation rules and when it is worth using our more sophisticated approaches. Our analytical and numerical results shed light on this question: We find that two forms of customer heterogeneity in the sales hierarchy are decisive. We term these "between heterogeneity" and "within heterogeneity." In Example (a) in Figure 2.1, heterogeneity occurs mostly on the highest level, where the priority and (service-level) requirements of different industry segments differ widely, while differences between countries (in the same industry segment) are less pronounced. Thus, the individual sub-trees of the hierarchy vary substantially in terms of their heterogeneity (they exhibit high between heterogeneity), while each sub-tree is relatively homogeneous in itself (low within heterogeneity). On the other hand, in Figure 2.1's Example (b), heterogeneity occurs mostly on the lowest level (e.g., key accounts are more important than general sales). In this case, there is a hierarchy with multiple similar sub-trees—that is, low between heterogeneity—but very heterogeneous customers in the sub-tree—that is, high within heterogeneity. In our numerical analyses we systematically vary these two types of customer heterogeneity and compare the results of the allocation approaches to the theoretical optimum. Our results suggest that the conventional *per commit* rule should be employed only under conditions of low (overall) heterogeneity, while the *hybrid* approach (the *service level aggregation* approach) works well when there is low between (low within) heterogeneity. Although our new allocation rules perform reasonably well in settings that are characterized by high within and high between heterogeneity, the company may, under certain conditions, want to use an approach that guarantees optimal allocations in order to further increase the performance of allocation planning. Therefore, we also show how optimal allocations can be determined in a decentralized fashion without having to assume an omniscient central planner. This, however, comes at the expense of extensive information-sharing and involved computations on the various levels of the sales hierarchy.

The remainder of this paper is organized as follows: Section 2.2 provides an overview of the relevant literature and positions our contribution in relation to previous work. We introduce our basic analytical model in Section 2.3. In Section 2.4 we explain how to compute optimal allocations in the case of an omniscient central planner and when allocation planning is carried out in a decentralized fashion. In Section 2.5 we use our previous results to analyze when the conventional alloca-

tion rules lead to optimal allocations and then develop two advanced allocation approaches (the *hybrid* approach and the *service level aggregation* approach), which are optimal under less restrictive conditions. In Section 2.6, we investigate the performance of these rules outside their domain of optimality using an extensive numerical analysis. Based on our analytical and numerical analyses, we develop recommendations on when to employ which approach for allocation planning. Section 2.7 summarizes our research contributions and discusses the limitations of our research as well as possible avenues for further research.

2.2 Literature Review

According to Kilger and Meyr (2015), demand fulfillment can be broadly characterized as a three-step process: demand planning, where forecasts are generated at lower levels of the sales hierarchy and then aggregated on higher levels; allocation planning, where aggregated supply is disaggregated and allocated from the top of the hierarchy to the lowest planning levels; and order promising/order fulfillment where the allocations on the lowest planning level are used to fulfill customer orders based on pre-defined rules.

A number of authors propose and discuss rules for allocation planning in flat hierarchies (that is, without considering the multi-level hierarchal structure of the sales organization). These approaches are frequently referred to as allocated Available to Promise (aATP) or Push-based ATP—see Ball et al. (2004) and Quante et al. (2009b) for reviews and Pibernik (2005) and Framinan and Leisten (2010) for classifications. Meyr (2009) uses a linear programming approach to allocate ATP to higher-profit customers and shows in an extensive simulation experiment that the approach can lead to close to optimal results compared to an ex-post optimization. Meyr finds that allocation planning is most beneficial if customers' profits are heterogeneous and demand forecasts are accurate.

To the best of our knowledge, Kilger and Schneeweiss (2000) are the first to describe the problem of allocation planning in a multi-level sales hierarchy, the problem that lies at the heart of the research we present in this paper. They consider allocation planning as a core part of the overall demand-fulfillment process. Kilger and Meyr (2015) discuss allocation rules that are common in industry practice (*per commit*, *rank based*, and *fixed split*). Based on Kilger and Meyr's work, Roitsch and Meyr (2015), from whom we borrowed Example (a) in Figure 2.1, characterize

the demand-fulfillment process in the downstream supply chain of a company in the oil industry. As master planning is performed on a high level of aggregation, and the profitabilities of individual customer groups vary widely, they stress the importance of allocation planning in matching the planned supply with (uncertain) demand, especially when supply is scarce. In their case, allocation planning takes place in a four-level hierarchy in which allocations are determined top-down and level-by-level such that, on each level, the customer groups with the highest (average) profits are prioritized.

Motivated by Roitsch and Meyr (2015), Vogel and Meyr (2015) are the first to focus solely on the problem of allocation planning in sales hierarchies. They show for a single-period setting with deterministic demand that a decentralized profit-based allocation leads to a significant loss in total profit compared to the global optimum. The loss in profit occurs because the decentralized allocation approach averages customers' profits on each level, leading to a loss of relevant information. As a consequence, the approach fails to prioritize customers with higher profits. Using a measure of income inequality from the economic literature, the Theil index, to capture the profit heterogeneity of customer groups, they develop an allocation approach to mitigate the aforementioned problem. Based on numerical analyses, the authors show that their approach leads to close to optimal allocations and robustly outperforms the conventional rules, at least when demand is deterministic.

While our work is similar to that of Vogel and Meyr (2015) in that we consider allocation planning in a multi-level sales hierarchy, it differs in three primary respects. First, we focus on allocation planning under uncertain demand. Second, we assume that allocation planning is carried out with the objective of meeting pre-defined service-level targets for customer groups, which is in line with the current industry practice, where companies implicitly or explicitly promise their customers certain service levels. Third, we account for various hierarchy setups—that is, how customers are structured in the hierarchy—and show how the set-up affects the performance of allocation rules. Doing so allows us to identify and study the effects of within and between heterogeneity.

We also provide five methodical contributions. First, we formalize the problem of allocation-planning in a multi-level sales hierarchy with uncertain customer demand and characterize the optimal solution for an omniscient central planner. Second, we show how an optimal allocation can be obtained in a decentralized fashion and discuss its feasibility in practical settings. This approach can be viewed as a stochastic extension of Vogel and Meyr (2015), although we do not approximate a

profit function. Third, we establish the conditions under which the conventional allocation rules lead to optimal allocations. Fourth, we use these insights to derive and analyze two new approaches, which are relatively easy to implement but whose results are superior to those of the conventional rules because they exploit more of the relevant information about the individual customer groups. Finally, we carry out numerical analyses to obtain comprehensive insights into when the conventional rules and our new approaches perform well relative to the optimal solution of an omniscient central planner. As a result, we can derive useful recommendations for practitioners by showing under what conditions certain allocation approaches can be used without substantial losses in performance.

2.3 The Model

Consider a company that supplies a single product during a single sales period to a set of diverse customers who are part of a sales hierarchy (as explained in Section 2.1). A sales hierarchy can be represented by a rooted and balanced[2] mathematical tree with a set of nodes, $\mathcal{N}$. Each node belongs to one of K levels. Level $k = 1$ contains only the root node, $0 \in \mathcal{N}$; all levels $k > 1$ contain at least one node; and $\mathcal{I}_k$ denotes the set of nodes on level k. For each intermediate node, $n \in \mathcal{N} \setminus \mathcal{I}_K$, the set of immediate successor nodes is denoted by $\mathcal{S}_n$. The root node represents the highest level of the sales hierarchy (e.g., the company's headquarters), and intermediate nodes represent sales regions or sales divisions. Each leaf node $l \in \mathcal{I}_K$ (i.e., a node without successor node on level K) represents a homogeneous customer group. Figure 2.2 provides an illustration of a generic sales hierarchy.

Section 2.1 used the notion of a sub-tree as part of the overall sales hierarchy. The root node of a sub-tree is an intermediate node $n \in \mathcal{N} \setminus \mathcal{I}_K$; the sub-tree $\mathcal{N}_n \subset \mathcal{N}$ consists of the (sub-tree) root node n, its successors $m \in \mathcal{S}_n$, their successors $m' \in \mathcal{S}_m$ for all $m \in \mathcal{S}_n$, and any further successors until the leaf nodes. (See Figure 2.2 for a visualization.) In particular, the leaf nodes of the sub-tree to n are $\mathcal{L}_n = \mathcal{N}_n \cap \mathcal{I}_K$.

The nodes on level $K - 1$ represent the lowest level of the sales organization, which is responsible for forecasting and fulfilling the customer groups' individual

[2] Assuming the tree is balanced (i.e., all leaf nodes have the same distance to the root) is without loss of generality; any unbalanced tree can be transformed into a balanced tree by adding "dummy nodes."

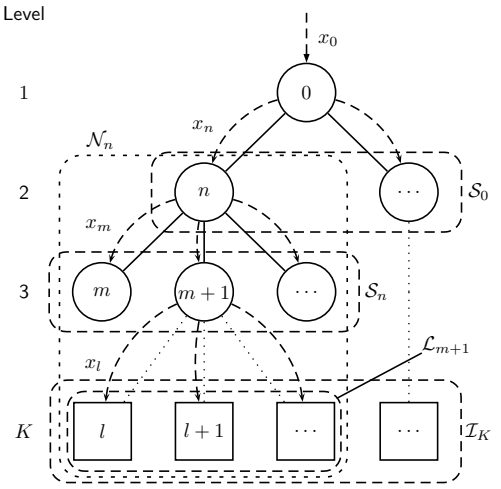


Figure 2.2: Formal representation of a general sales hierarchy.

demands (represented by leaf nodes $l \in \mathcal{I}_K$). D_l denotes the uncertain demand of customer group $l \in \mathcal{I}_K$, and we assume that it follows a demand distribution with cumulative distribution function (cdf) G_l, a mean μ_l, and a coefficient of variation (CV) CV_l. We also assume that an alpha-service-level target is defined for each customer group, which we denote by α_l.

We assume the supply for the planning period, denoted by x_0, is deterministic and given by the master production plan. Prior to the sales period (during allocation planning), the company allocates this supply to the nodes of the sales hierarchy. We denote by x_n the amount of supply that is allocated to node $n \in \mathcal{N}\backslash\{0\}$. In a sales hierarchy, each node's supply allocation is an upper bound of the sum of allocations to its successor, which leads to Definition 2.1. Here and in the following we use bold characters to represent vectors.

Definition 2.1 (Feasible allocation)**.** *An allocation $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{N}\backslash\{0\}|}$ is a feasible allocation if $x_n \geq 0$ for all $n \in \mathcal{N}\backslash\{0\}$ and $\sum_{m\in\mathcal{S}_n} x_m = x_n$ for all $n \in \mathcal{N}\backslash\mathcal{I}_K$. The set of feasible allocations is $\mathcal{F} = \{\boldsymbol{x} \in \mathbb{R}^{|\mathcal{N}\backslash\{0\}|} \mid \boldsymbol{x} \text{ is feasible}\}$.*

The feasibility condition $\sum_{m\in\mathcal{S}_n} x_m = x_n$ implies that all initial supply x_0 is allocated and that there are no other sources of supply at lower levels of the hierarchy. (In a single period model there is no reason to retain unallocated supply.) In particular, any feasible allocation fulfills $\sum_{l\in\mathcal{I}_K} x_l = x_0$, which follows from a

straightforward induction. Reflecting that a sales hierarchy typically involves dispersed geographies and separate business divisions with individual profits and losses, supply that is allocated to a customer group $l \in \mathcal{I}_K$ can be consumed only by this customer group. In other words, we do not consider transshipments or other forms of inventory/supply-sharing (e.g., nesting).

For a given allocation $\boldsymbol{x}$, we are able to compute basic performance measures:

Proposition 2.1 (Basic performance measures)**.** *Denoting expected fulfilled demand for customer group l by $\hat{x}_l$, expected unfulfilled (lost) demand by L_l, and the corresponding alpha-service level by $\hat{\alpha}_l$, we have*

1. $\hat{x}_l(x_l) = \int_0^{x_l} (1 - G_l(t))\, \mathrm{d}t$
2. $L_l(x_l) = \int_{x_l}^{\infty} (1 - G_l(t))\, \mathrm{d}t$
3. $\hat{\alpha}_l(x_l) = G_l(x_l)$

The proof of Proposition 2.1 and the subsequent proofs can be found in Appendix A.1.

Based on Definition 2.1 and the results stated in Proposition 2.1, we can express the company's allocation planning problem as:

Problem 2.1a (Allocation planning problem)**.** For a given vector of service-level targets $\boldsymbol{\alpha} = (\alpha_l)_{l \in \mathcal{I}_K}$, determine the allocation $\boldsymbol{x} \in \mathcal{F}$ that minimizes $[\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}(\boldsymbol{x})]^+$, where $[\boldsymbol{v}]^+ = (\max\{v_1, 0\}, \max\{v_2, 0\}, \ldots)$.

It is straightforward to compute the allocation required at each node in order to meet the corresponding customer groups' service-level targets. We denote this allocation by x_l^r. For $l \in \mathcal{I}_K$, $x_l^r = G_l^{-1}(\alpha_l)$, where $G_l^{-1} : [0,1] \to \mathbb{R}$ denotes the left inverse of G_l. For $n \notin \mathcal{I}_K$, the corresponding x_n^r can be computed inductively by summing the allocations of all successor nodes $l \in \mathcal{S}_n$. In particular, the required allocation to fulfill the service-level targets of all customer groups—that is, $x_0^r = \sum_{l \in \mathcal{I}_K} x_l^r$—depends only on the demand distributions and service-level targets of the customer groups and is not influenced by the hierarchy's structure.

Because it is trivial to solve Problem 2.1a for $x_0 \geq x_0^r$, we focus on situations in which supply is scarce (i.e., $x_0 < x_0^r$). Determining an optimal allocation is challenging for several reasons whenever supply is scarce. First, because $[\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}(\boldsymbol{x})]^+$ is an $|\mathcal{I}_K|$-dimensional objective function, Problem 2.1a is a multi-objective decision-making problem. Second, while a solution to Problem 2.1a is typically determined

iteratively in a decentralized fashion, in practice, the allocation vector and, therefore, the service-level deviations result from the entirety of all of the local planners' decisions. For each intermediate node $n \in \mathcal{N} \setminus \mathcal{I}_K$ there is a dedicated planner who receives allocation x_n from the parent node (the root node "receives" x_0), which the planner then splits, based on his or her individual preferences, between the successor nodes in $\mathcal{S}_n$ (cf. Figure 2.2).[3] Therefore, in this case, Problem 2.1a is a decentralized multi-objective planning problem in which multiple planners make (local) allocation decisions based on their individual preferences with respect to the service-level deviations of their successor nodes.

While we can identify the set of pareto-optimal allocations for Problem 2.1a, it is not clear which of the pareto-optimal allocations to choose if we do not know the planners' preferences regarding service-level deviations $[\alpha_l - \hat{\alpha}_l(x_l)]^+$ for the customer groups $l \in \mathcal{I}_K$. We cannot compare the performance of the various methods for allocation planning unless we transform the multi-objective problem into a scalar-valued problem. To this end, we first state an alternative formulation of Problem 2.1a.

Problem 2.1b (Allocation planning problem with expected shortfall)**.** For a given vector of required allocations $\boldsymbol{x}^r = (x_l^r)_{l \in \mathcal{I}_K}$, determine the allocation $\boldsymbol{x} \in \mathcal{F}$ that minimizes $\left(\left[L_l(x_l) - L_l(x_l^r) \right]^+ \right)_{l \in \mathcal{I}_K}$.

$\left[L_l(x_l) - L_l(x_l^r) \right]^+$ is the additional expected shortfall of customer group l, which is calculated as the difference between the expected shortfall when customer group l receives x_l^r, based on the service-level target α_l, and the expected shortfall when it receives an allocation x_l. From Proposition 2.1 (parts 1 and 2) we know, without the need for additional proof, that the set of pareto-optimal solutions to Problem 2.1a contains all pareto-optimal solutions to Problem 2.1b.

Working with additional expected shortfalls instead of service levels is convenient for multiple reasons. It allows us to capture the highly non-linear relationship between allocations and service levels, and it is reasonable to assume that each unit of additional expected shortfall for a particular customer group l induces the same negative consequences for the company. With this assumption, we can introduce constant weights w_l that reflect the planner's preferences regarding the additional expected shortfalls $[L_l(x_l) - L_l(x_l^r)]^+$ of the customer

[3] The planner's decision is typically based on a selection of information that is specific to node n. (We refrained from formalizing this specific information to avoid unnecessary and potentially confusing notation.) In Section 2.5, we describe examples of such selections of information.

groups l, and derive a (scalar-valued) surrogate objective function that takes the form $W(\boldsymbol{x}) = \sum_{l \in \mathcal{I}_K} w_l \left[L_l(x_l) - L_l(x_l^r) \right]^+$. Based on this function, we can formulate the following surrogate (single-objective) optimization problem:

Problem 2.1c (Single-objective allocation planning problem)**.** For a given vector of required allocations $\boldsymbol{x}^r = (x_l^r)_{l \in \mathcal{I}_K}$, determine the allocation $\boldsymbol{x} \in \mathcal{F}$ that minimizes $W(\boldsymbol{x})$.

Clearly, this problem formulation prompts the question concerning how to determine the weights w_l. In theory, one could use well-established methods for eliciting the planner's preferences or, in the decentralized case, the preferences of multiple planners to obtain these weights. For example, one could derive weights from pairwise comparison matrices. Apart from the apparent problems and difficulties (effort, potential inconsistencies across planners), deriving weights in this way would have a methodological drawback. As the service-level targets α_l already contain information regarding the relative importance of customer groups, eliciting essentially the same information using, for example, pairwise comparison matrices, is not only redundant but also likely to lead to inconsistencies. Therefore, we exploit the information contained in the target service levels α_l for customer groups l to determine the weights w_l. In traditional inventory theory a (single) service-level target reflects a ratio of (per-unit) overage and underage costs. Recall the derivation of the optimal alpha-service level in the standard newsvendor model: one obtains $c^{overage} \cdot \alpha = c^{underage} \cdot (1 - \alpha)$ by equating expected marginal overage costs and expected marginal underage costs. This gives the well-known identity $\alpha = c^{underage} / (c^{overage} + c^{underage})$ (cf. Chopra and Meindl, 2010). However, in our model, the service-level targets for the customer groups contain information beyond the ratio of underage and overage costs for a single customer group. If the service-level targets α_l are set correctly, they also inform us about differences in underage and overage costs between the customer groups l. At first, one may think that the overage costs are the same across all customer groups such that an unsold unit has the same negative consequences, regardless of the customer group to which it was allocated or the available supply x_0. In this case, the differences in α_l could be attributed to differences in underage costs, and it would be simple to infer a per-unit weight w_l that reflects the negative consequences of being one unit short for customer group l. Section 2.4.1 shows that for scarce supply (i.e. when $x_0 < x_0^r$) this inference holds true only in the optimum. Otherwise, the expected marginal overage costs that are associated with allocating one incremental unit to a

customer group are equal to the reduction in expected marginal underage costs of allocating this unit to another group. Thus, they depend on the current allocation, and we cannot directly employ the logic described above. However, in the optimum, the marginal weighted expected shortfalls of allocating an additional unit are equal across all customer groups as long as $x_0 \leq x_0^r$. Therefore, assuming constant weights w_l, we need only one optimal solution to derive the weights w_l. We know that, for $x_0 = x_0^r$, the optimal solution is $\boldsymbol{x} = \boldsymbol{x}^r$, and we can show that this optimal solution will be obtained only if we set $w_l = 1/(1-\alpha_l)$. (See Lemma 2.2 in Section 2.4.1.) Therefore, we use this definition of w_l to derive an optimal solution to Problem 2.1c—which will serve as a best-case benchmark—in Section 2.4 and to compare the various heuristics for decentralized allocation planning in Section 2.5 and Section 2.6.

2.4 Optimal Allocation of Supply

We first solve Problem 2.1c from the perspective of an omniscient central planner and, most important, derive structural insights that guide our further analyses (Section 2.4.1). In Section 2.4.2, we use these insights to develop an approach to achieve optimal allocations even when allocation planning is carried out in a decentralized fashion.

2.4.1 Optimal Central Allocation

In this section we consider the case of an omniscient central planner, located at the root node, who decides upon the allocations to all intermediate nodes and leaf nodes in the sales hierarchy. Assuming that the planner's preferences can be represented by weights w_l, the planner solves Problem 2.1c. The optimal solution for this setting serves as a benchmark throughout the remainder of our upon analyses and reveals important insights into what makes an allocation optimal (or at least good).

As a first step in developing a solution approach for Problem 2.1c, we prove the convexity of the objective function.

Lemma 2.1 (Convexity of the objective function)**.**
$W(\boldsymbol{x}) = \sum_{l \in \mathcal{I}_K} w_l \left[L_l(x_l) - L_l(x_l^r)\right]^+$ *is convex in* $\boldsymbol{x}$*.*

Lemma 2.1 ensures that local extrema of W are minima, so using (non-linear) Lagrangian methods (cf. Ruszczyński, 2006), we can describe each solution for Problem 2.1c by means of a Lagrangian multiplier λ, as Theorem 2.1 shows.

Theorem 2.1 (Characterization of solutions). *$\boldsymbol{x}^* \in \mathcal{F}$ is optimal if and only if there exists a value $\lambda > 0$, such that, when using*

$$A_\lambda = \{l \mid l \in \mathcal{I}_K, \lambda \geq w_l(1 - G_l(0))\}, \tag{2.1}$$

the following hold

$$-\lambda/w_l \in \partial L_l(x_l^*) \qquad \text{for all } l \in \mathcal{I}_K \setminus A_\lambda \tag{2.2}$$

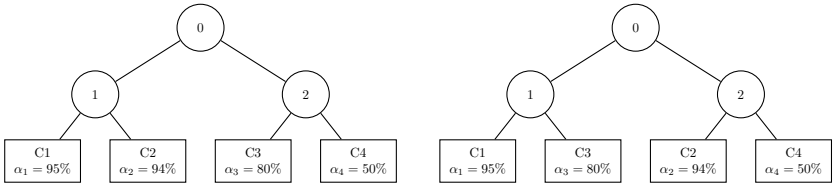
$$x_l^* = 0 \qquad \text{for all } l \in A_\lambda. \tag{2.3}$$

In addition, for any optimal $\boldsymbol{x}^*$

$$\sum_{l \in \mathcal{I}_K} x_l^* = x_0. \tag{2.4}$$

Allow us a technical note before we discuss the intuition behind Theorem 2.1. (2.2) specifies that $-\lambda/w_l$ is part of the sub-differential of L_l at the point x_l^*. Using the concept of sub-differentiability allows us to also cover instances in which L_l is not differentiable (e.g., discrete demand distributions). Whenever L_l is differentiable in x_l^*, (2.2) reduces to $\lambda/w_l = -L_l'(x_l^*) = 1 - G_l(x_l^*)$, as we discuss in more detail in the proof of Corollary 2.1, below. In addition, using Proposition 2.1 part 3 and the fact that L_l is real-valued, we can formulate (2.2) as $\lim_{x \nearrow x_l^*} G_l(x) \leq 1 - \frac{\lambda}{w_l} \leq \lim_{x \searrow x_l^*} G_l(x)$, where $\lim_{x \searrow x_l^*} G_l(x) = G_l(x)$ as a consequence of G_l's being càdlàg.

In Theorem 2.1 the set of customer groups is split into two sets: the set of customer groups ($\mathcal{I}_K \setminus A_\lambda$) who receive an allocation and the set of customer groups (A_λ) who do not receive an allocation. For the first set the incremental reduction in weighted shortfalls per additional unit of supply is equal across all customer groups. This is best observable when G_l is differentiable: Then, (2.2) is equivalent to $\lambda = \left[1 - G_l(x_l^*)\right] \cdot w_l$, where $1 - G_l(x_l^*)$ is the probability that an additional marginal unit of supply is consumed. Consequently, the Lagrangian multiplier λ is the marginal reduction in weighted shortfalls at the point at which the entire supply is allocated. Similar results have been obtained by Allen (1985) who showed that (for homogeneous customer groups) supply should be allocated such that all customer groups have the same probability that an additional unit of supply is



(a) High between and low within heterogeneity **(b)** Low between and high within heterogeneity

Figure 2.3: Examples of sales hierarchies.

consumed. Avrahami et al. (2014) obtained similar structural results for a problem of allocating supply to different retailers. Because we are using weighted shortfalls the customer groups' maximal marginal reduction in weighted shortfalls differs. As a consequence, customer groups whose maximal shortfall reductions are lower than λ receive no supply (2.1). As more supply becomes available λ decreases, more and more customers receive an allocation (and the cardinality of the set A_λ decreases).

Our setting is also very close to a special case of a divergent multi-echelon inventory system where intermediate stockpoints may not hold inventories, lead times are zero, backorder costs are linear and holding costs are zero. Diks and de Kok (1998) studied the optimal control of divergent multi-echelon inventory systems and obtained similar structural results as those presented in Theorem 2.1.

It is worth noting that the optimal allocations as characterized by Theorem 2.1 only depend on properties of the customer groups (leaf nodes). Thus, the optimal allocations are independent of how the customer groups are arranged in the sales hierarchy. Consider, for instance, the example illustrated in Figure 2.3, which is inspired by our the example shown in Figure 2.1. The leaf nodes (customer groups) are the same in both sales hierarchies, but they are grouped differently into subtrees. In both hierarchies the optimal allocations to the customer groups are the same. However, we will see later that this is not the case for some decentralized planning approaches, where the hierarchy's structure affects the outcomes.

Figure 2.4 illustrates the allocation logic for a sample problem instance, showing the sequential nature of the allocation. While at first only customer group 1 (the one with the highest service-level target and weight) receives an allocation, the other customer groups are gradually added and any additional supply is shared

between the customer groups.[4] Typically, a customer group that has just been added (i.e., removed from A_λ) receives the majority of the additional allocation—that is, the slope of the allocation curve for this group is the steepest—since the probability that an additional unit will be consumed is high when the allocation is still small.

In addition, the part of Figure 2.4 which displays the value of λ (i.e., the marginal weighted shortfall reduction) depending on the available supply, is highly intuitive. When only a few units of supply are available, the probability that an allocated unit is consumed is high, so the weighted shortfall reduction is close to the weight. However, the more is allocated to a particular customer group, the less likely it is that an additional unit allocated to this customer group will be consumed, so the expected (marginal) shortfall reduction decreases. At the point at which the expected (marginal) shortfall reduction is as low as the initial expected reduction of the weighted shortfall of the next lower customer class (which equals the corresponding weight w_l if $G_l(0) = 0$), this class also receives an allocation.

While Theorem 2.1 might still seem abstract, assuming that all G_l are continuous and strictly increasing allows us to translate the intuition developed above into a set of formulae for straightforward computation of an optimal solution.

Corollary 2.1 (Continuous and strictly increasing cdf)**.** *Assume that G_l is continuous and strictly increasing on $\{G_l < 1\}$ for all $l \in \mathcal{I}_K$; then there is a single optimal solution of Problem 2.1c that is defined by*

$$x_l^*(\lambda) = \begin{cases} 0 & \text{if } \lambda \geq [1 - G_l(0)] \cdot w_l \\ G_l^{-1}(1 - \dfrac{\lambda}{w_l}) & \text{else.} \end{cases}$$

Section 2.3 provided a loose explanation for how to derive weights w_l from service-level targets α_l. Theorem 2.1 and its proof confirm that, in the optimum, an incremental unit's marginal benefit is equal across all customer groups, equaling $\lambda(x_0)$. As simultaneous scaling of all weights—and, thus, $\lambda(x_0)$—does not impact

[4] The points at which a customer group l receives the first allocation (the kinks in the graphs) can be computed up front: At the point at which customer group l receives the first allocation, we have $\lambda^l = [1 - G_l(0)] \cdot w_l$. The corresponding allocations are $x_k^l = 0$ if $\lambda^l \geq [1 - G_k(0)] \cdot w_k$ and $x_k^l = G_k^{-1}(1 - \lambda^l / w_k)$ in all other cases. The corresponding total supply is $x_0^l = \sum_{k \in \mathcal{I}_K} x_k^l$. The possibility to calculate x_0^l and λ^l for all $l \in \mathcal{I}_K$ helps to determine the order in which customer groups are served (i.e., in order of decreasing λ^l) and to compute optimal solutions for Problem 2.1c. More precisely, if $x_0^l \leq x_0 \leq x_0^k$, then $\lambda^l \leq \lambda^* \leq \lambda^k$. We use this property in our numerical algorithm in Section 2.6.

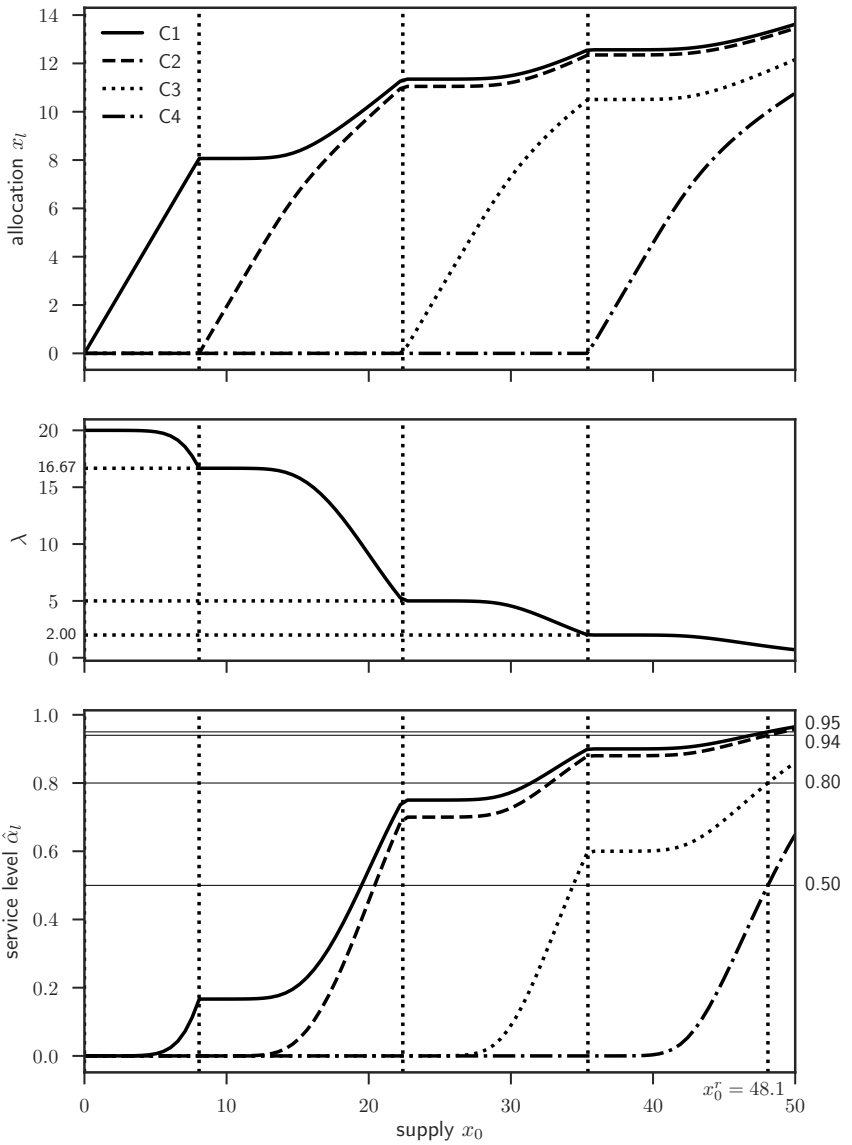


Figure 2.4: Allocations, the corresponding λ and the resulting service levels for optimal allocations in case of normal distributed demand (with $\mu_l = 10$, $\sigma_l = 2$) and weights $w_1 = 20, w_2 = 16.\overline{6}, w_3 = 5, w_4 = 2$.

the results, we can assume without loss of generality that $\lambda(x_0^r) = 1$. Therefore, $x_l^r = G_l^{-1}(1 - 1/w_l)$, which leads to $w_l = 1/(1 - \alpha_l)$, which, following the reasoning described in Section 2.3, is the natural way to convert given service-level targets to weights.

Lemma 2.2 confirms that this conversion fulfills the conditions one would naturally posit: the weights for customer groups should be chosen such that customers with higher service-level targets achieve higher service levels for all levels of supply, and when supply is sufficient, all service-level targets are fulfilled.

Lemma 2.2 (Linking service-level targets and weights)**.** *Assume that $G_l(0) = 0$ for all $l \in \mathcal{I}_K$ and set $w_l = 1/(1 - \alpha_l)$. Then*

1. *For any $l, k \in \mathcal{I}_K$ and for any $x_0 > 0$, $\alpha_l \geq \alpha_k \Rightarrow \hat{\alpha}_l \geq \hat{\alpha}_k$.*
2. *If $x_0 = x_0^r$, then the optimal allocation fulfills $\hat{\alpha}_l = \alpha_l$.*

Lemma 2.2 shows that the conversion $\alpha_l \mapsto w_l = 1/(1 - \alpha_l)$ preserves the main information of the service-level targets—that is, the relative importance of the customer segments and the optimal allocation when there is sufficient supply.[5] Figure 2.4 shows the expected service levels if this conversion is used when $\alpha_1 = 0.95, \alpha_2 = 0.94, \alpha_3 = 0.8$, and $\alpha_4 = 0.5$ (i.e., the case that corresponds to the setting depicted in Figure 2.3). Clearly, the relative importance of the customer groups is reflected in the expected service levels.

2.4.2 Optimal Decentral Allocation

In a decentralized planning regime there is a planner at the root node and at each intermediate node. These planners determine the allocations to their immediate successor nodes. Vogel and Meyr (2015) show for a deterministic profit-maximization problem that the optimal central solution can also be achieved in decentralized planning regimes when each planner knows the (piecewise-linear) profit functions of his or her successor nodes. In the following, we extend this logic to our stochastic setting, formalize a fully decentralized version of Problem 2.1c and then use dynamic programming techniques to show that it is sufficient for the

[5] Although doing so is not only intuitive but also convenient from a technical perspective, assuming $G_l(0) = 0$ for all $l \in \mathcal{I}_K$ in Lemma 2.2—is more restrictive than necessary. In order to prove part 1 it would, for instance, be sufficient to assume that $G_k(0) = 0$ for the one k in question or to assume that x_0 is large enough to warrant $x_l^* > 0$ for all $l \in \mathcal{I}_K$. In order to prove part 2, an alternative condition could be, for instance, $\alpha_l > G_l(0)$.

planners to know the (nonlinear) objective functions of their successor nodes. Finally, building on these insights, we discuss whether and how such a decentralized approach could be implemented in practice.

A planner at node $n \in \mathcal{N} \setminus \mathcal{I}_K$ seeks to find the allocations x_m to his or her successor nodes $m \in \mathcal{S}_n$ that minimize the planner's total expected weighted shortfall. Suppose the planner knows the expected weighted shortfall functions $W_m(x_m)$ for each successor node $m \in \mathcal{S}_n$. Then he or she can determine the optimal allocation to the successor nodes by solving Problem 2.2, a non-linear knapsack-problem where $\boldsymbol{x}^n = (x_m)_{m \in \mathcal{S}_n}$ denotes the allocation vector and $\mathbb{R}_{\geq 0}^{|\mathcal{S}_n|}$ the set of non-negative real-valued $|\mathcal{S}_n|$ dimensional vectors.

Problem 2.2 (Decentral allocation planning problem)**.**

$$\min_{\boldsymbol{x}^n \in \mathbb{R}_{\geq 0}^{|\mathcal{S}_n|}} \sum_{m \in \mathcal{S}_n} W_m(x_m)$$

subject to

$$\sum_{m \in \mathcal{S}_n} x_m \leq x_n.$$

The weighted shortfall functions W_m in Problem 2.2 for level K can be directly computed from the loss functions and the weights of the corresponding customer groups. At higher levels of the sales hierarchy they must be derived iteratively from the successor nodes by solving Problem 2.2 for all possible allocations x_m, which results in the following definition of the weighted shortfall functions:

$$W_m(x_m) = \begin{cases} w_m \left[L_m(x_m) - L_m(x_m^r)\right]^+ & \text{if } m \in \mathcal{I}_K \\ \min\limits_{\boldsymbol{y} \in \mathbb{R}_{\geq 0}^{|\mathcal{S}_m|}; \|y\|_1 \leq x_m} \sum_{l \in \mathcal{S}_m} W_l(y_l) & \text{else.} \end{cases}$$

Let $\boldsymbol{x}^d$ denote a solution to the decentralized allocation problem. Then $\boldsymbol{x}^d$ can be determined using Algorithm 2.1.

Algorithm 2.1 Decentral allocations

$x_0^d \leftarrow x_0$
for $i \in \mathcal{I}_k, k = 0, \ldots, K-1$ **do**
  Solve Problem 2.2 with $n = i$ and $x_i = x_i^d$
  $x_m^d \leftarrow x_m^i$ for all $m \in \mathcal{S}_i$
end for
return $\boldsymbol{x}^d$

As Problem 2.2 is a subproblem to Problem 2.1c, reformulating Bellman's (1957) principle of optimality gives rise to Lemma 2.3.

Lemma 2.3. *Assume $x_n = x_n^*$ and let $\boldsymbol{y}^* \in \mathbb{R}_{\geq 0}^{|\mathcal{S}_n|}$ be a solution of Problem 2.2; then there exists a solution $\boldsymbol{x}^*$ of Problem 2.1c with $y_m^* = x_m^*$ for all $m \in \mathcal{S}_n$.*

Proposition 2.2, as a straightforward consequence of Lemma 2.3, shows that $\boldsymbol{x}^d$ is optimal.

Proposition 2.2. *$\boldsymbol{x}^d$ is an optimal solution to Problem 2.1c, i.e. $\boldsymbol{x}^d = \boldsymbol{x}^*$.*

We observe that the hierarchical allocation problem can be solved optimally in a decentralized fashion—that is, without an omniscient central planner. More precisely, the joint decisions of dedicated planners for each node, each of which has information only about the allocation a node receives from its predecessor node and the weighted shortfall functions of its successor nodes, lead to an optimal allocation for the entire sales hierarchy.

However, this approach is mostly theoretical and for two primary reasons is unlikely to be implemented in practice. First, the information that must be communicated from each node to its predecessor is an entire real-valued function without an explicit expression. While it is possible to communicate an approximation of this function (e.g., based on the Lorenz-curve approximation used by Vogel and Meyr, 2015), the transmitted information will be difficult to interpret, especially compared to the relative ease of current methods that require communication of only one or two values (e.g., mean demand and service level) that are easy to understand. Second, Problem 2.2 must be solved for each node, even when the objective function is approximated. While doing so is computationally feasible, solving a non-linear knapsack problem is a difficult task that requires involved computations and may result in low levels of acceptance from the planners.

The next section builds on the insights developed in this section and develops two new decentralized allocation approaches that are easier to implement but still provide acceptable results (if they are used properly).

2.5 Heuristic Allocation Rules

In this section we first address the optimality of the conventional allocation rules *per commit* and *rank based* (Kilger and Schneeweiss, 2000). As these rules are optimal only under very restrictive conditions, we propose two advanced allocation approaches, show under what conditions they lead to optimal results, and derive first insights into how they perform relative to the conventional rules.

2.5.1 Conventional Allocation Rules

In this section we discuss the *per commit* and *rank based* rules. In particular, we present a formalization for each of the rules, the information they require, an example of their use, and an analysis (based on Theorem 2.1) of when these allocations are optimal.

Per Commit Under *per commit*, scarce supply is allocated proportionally to forecasted demand, as formalized by Definition 2.2.

Definition 2.2 (Per commit)**.** *The per commit allocation from intermediate node* $n \in \mathcal{N}\backslash\mathcal{I}_K$ *to a successor node* $m \in \mathcal{S}_n$ *is* $x_m^{PC} = x_n \mu_m / \mu_n$*, where* $\mu_m = \sum_{m' \in \mathcal{S}_m} \mu_{m'}$ *and* $\mu_n = \sum_{m \in \mathcal{S}_n} \mu_m$ *are the total mean demand of node m and n, respectively.*

Per commit has a property that is relevant to our further analysis. From Definition 2.2 we infer that the allocation to each customer group $l \in \mathcal{I}_K$ is $x_l^{PC} = \frac{\mu_l}{\mu_0} x_0$, where $\mu_0 = \sum_{l \in \mathcal{I}_K} \mu_l$. Thus, the allocations are independent of the structure of the hierarchy. We also see that *per commit* requires that only limited information is shared in the hierarchy, as it is sufficient for each node n to know only the total demands μ_m of its successors $m \in \mathcal{S}_n$.

To provide more insight regarding the allocation logic of *per commit*, Figure 2.5 plots the expected service levels of a *per commit* allocation for various levels of supply and compares them with the optimal allocation for the examples from Figure 2.3. As all customer groups have the same demand, and *per commit* does not prioritize, each customer group receives the same allocation and, thus, the same

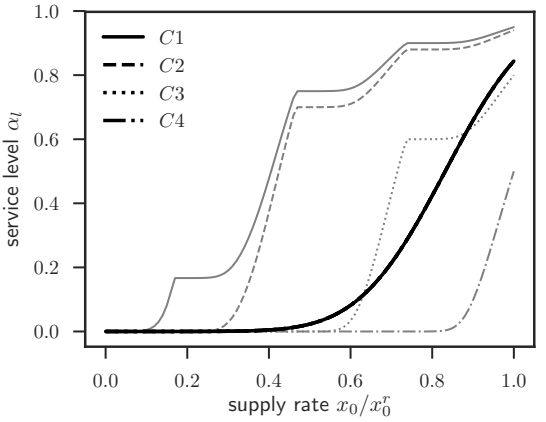


Figure 2.5: Expected service level per customer group of *per commit* (black) compared to optimal allocation (gray).

service level. For this reason, the individual service levels in Figure 2.5 cannot be distinguished.

Because allocations are based only on the mean demand and do not depend on other characteristics, the service levels in our example differ substantially from the target service levels. More important, even if supply is sufficient, some of the service-level targets are not met, which raises the question concerning whether and under what conditions *per commit* allocates optimally.

Proposition 2.3 (Optimality of per commit allocations)**.** *Assume that G_l is continuous and strictly increasing in $[0, x_l^r]$ for all $l \in \mathcal{I}_K$. A per commit allocation is optimal—that is, $x_l^{PC} = x_l^*$ for all $x_0 \in [0, x_0^r]$—if and only if*

$$\frac{w_{l'}}{w_{l''}} = \frac{1 - G_{l''}(\mu_{l''}\chi)}{1 - G_{l'}(\mu_{l'}\chi)} \qquad \text{for all } l', l'' \in \mathcal{I}_K, \chi \in \left(0, \frac{x_{l'}^r}{\mu_{l'}}\right]. \tag{2.5}$$

An instance in which Condition (2.5) is fulfilled is when the service-level targets and demand distributions are identical for all customer groups. However, it is not sufficient that the demand of the customer groups follow the same type of distribution, even when all weights are identical. Assume, for instance, that $G_{l'}$ and $G_{l''}$ are z-transformations of a common distribution, with cdf $G(z)$ (i.e., $G_{l'}(x) = G\left(\frac{x/\mu_{l'}-1}{CV_{l'}}\right)$ and $G_{l''}(x) = G\left(\frac{x/\mu_{l''}-1}{CV_{l''}}\right)$); then Condition (2.5) is fulfilled only if all customers have the same CV.

Hence, assuming that demand distributions belong to the same class, *per commit* allocates optimally only when there is no heterogeneity among the service levels, and demand forecasts have the same accuracy across customer groups—that is, when there is no forecast heterogeneity. We can assume that increasing heterogeneity in terms of service-level targets and/or forecast accuracy degrades the performance of *per commit*.

Under *per commit*, a customer group typically cannot fulfill its service-level targets even if there is sufficient supply. Therefore, we introduce a largely straightforward modification of the conventional *per commit* rule that remedies this problem and term this modified version *extended per commit*. *Extended per commit* is also a building block for one of our new allocation approaches that we propose in Section 2.5.2. Definition 2.3 provides a formal characterization.

Definition 2.3 (Extended per commit)**.** *The extended per commit allocation from intermediate node $n \in \mathcal{N} \setminus \mathcal{I}_K$ to successor node $m \in \mathcal{S}_n$ is $x_m^{ePC} = x_n \cdot x_m^r / x_n^r$, where x_m^r is the required allocation of node m and $x_n^r = \sum_{m \in \mathcal{S}_n} x_m^r$.*

In essence, *extended per commit* uses the required allocation x_l^r of a customer group l instead of its mean demand μ_l in determining the group's allocation. On level $K-1$ *extended per commit* uses information about the demand distributions of the customer groups and their service-level targets, so it exploits more information than conventional *per commit* does. However, only a single value, the aggregated required allocation $x_m^r = \sum_{l \in \mathcal{S}_m} x_l^r$, is shared with the predecessor node. Because *extended per commit* follows the same general allocation logic as conventional *per commit*, its results are independent of the structure of the hierarchy.

Proposition 2.4 (Optimality of extended per commit allocations)**.** *Assume G_l is continuous and strictly increasing in $[0, x_l^r]$ for all $l \in \mathcal{I}_K$; then the following hold:*

1. *If $x_0 = x_0^r$, then $x_l^{ePC} = x_l^*$ for all $l \in \mathcal{I}_K$.*
2. *$x_l^{ePC} = x_l^*$ for all $l \in \mathcal{I}_K$ and all $x_0 \in [0, x_0^r]$ if and only if*

$$\frac{w_{l'}}{w_{l''}} = \frac{1 - G_{l''}(\mu_{l''}\chi)}{1 - G_{l'}(\mu_{l'}\chi)} \qquad \text{for all } l', l'' \in \mathcal{I}_K, \chi \in \left[0, \frac{x_{l'}^r}{\mu_{l'}}\right]. \tag{2.6}$$

Part 2 of Proposition 2.4 shows that *extended per commit* is optimal, independent of the level of supply if and only if (2.6) holds. As (2.6) is equivalent to (2.5),

extended per commit is optimal whenever *per commit* is optimal. Part 1 of Proposition 2.4 shows that *extended per commit* is also optimal whenever supply is sufficient. Hence, it is reasonable to assume that *extended per commit* leads to acceptable allocations when supply levels are high—that is, x_0 is close to x_0^r. Therefore, it should always be preferred over conventional *per commit*.

Rank Based Allocation Under a *rank based* allocation, customer groups are ordered (ranked) according to some priority measure. The available supply is then allocated in ascending order according to the rank; that is, the customer group with rank 1 receives its required allocation first, after which the customer group with rank 2 receives its required allocation, and this sequential allocation continues until the available supply is exhausted. More formally, we define the *rank based* allocation as shown in Definition 2.4:

Definition 2.4 (Rank based allocation)**.** *Assume that the set of successor nodes of n, $S_n = \{m_1, m_2 = m_1 + 1, \ldots, m_{|S_n|}\}$, is priority-ordered—that is, the priority of m_1 is higher than m_2, etc. The rank based allocations from an intermediate node $n \in \mathcal{N} \backslash \mathcal{I}_K$ to its successors $m \in \mathcal{S}_n$ are*

$$x_m^{RB} = \begin{cases} 0 & x_n \leq \sum_{j=m_1}^{m-1} x_j^r \\ x_n - \sum_{j=m_1}^{m-1} x_j^r & \sum_{j=m_1}^{m-1} x_j^r \leq x_n \leq \sum_{j=m_1}^{m-1} x_j^r \\ x_m^r & else, \end{cases}$$

where $x_m^r = \sum_{m' \in \mathcal{S}_m} x_{m'}^r$ is the total required allocation of node m.

Rank based requires more information-sharing between the nodes and their predecessors than *per commit* does: In addition to the successor nodes' total mean demand, some ordinal measure of priority must be transmitted; that is, node n requires information before it can priority-order its successor nodes $m \in \mathcal{S}_n$. At first glance, this requirement does not appear to be particularly restrictive because on level $K-1$ the priority can be directly inferred from the known service-level targets of the customer groups $l \in \mathcal{I}_K$. However, establishing a priority order for nodes on higher levels (i.e., $K-2, \ldots, 1$) is not straightforward and requires a specific rule. Consider the hierarchies illustrated in Examples (a) and (b) (cf. Figure 2.3). While in (a), node 1 should clearly be prioritized over node 2, the priority order is not so obvious in (b). If node 1 is prioritized, then customer group C3 will receive an allocation before customer group C2 does, and if node 2 is prioritized, then C2

and C4 receive their allocations before C1 and C3 do. Clearly, the performance of the *rank based* allocation depends on how the priority order is established on levels $K-2,\ldots,1$, which also suggests that, in contrast to *per commit*, the outcome of a *rank based* allocation depends on the hierarchy's structure.

Rank based allocations' dependence on rules for establishing the priority order and the structure of the sales hierarchy make it difficult to derive general results regarding the allocations' performance. To obtain formal results without having to account for the dependence on the prioritization rules and the hierarchy's structure, we introduce what we term *centralized rank based* allocation. In this variant of the *rank based* allocation, the planner at the root node directly allocates the available supply to the customer groups in descending order of their service-level targets. In formal terms, the *centralized rank based* allocation can be defined as shown in Definition 2.5:

Definition 2.5 (Centralized rank based allocation)**.** *Let $\mathcal{I}_K = \{l_1, l_2 = l_1+1, \ldots, l_{|\mathcal{I}_K|}\}$ denote the set of customer groups that are ordered according to their service levels, such that $\alpha_{l_1} > \alpha_{l_2} > \ldots > \alpha_{l_{|\mathcal{I}_K|}}$. The centralized rank based allocation from node o to customer group $l \in \mathcal{I}_K$ is*

$$x_l^{cRB} = \begin{cases} 0 & x_0 \leq \sum_{j=l_1}^{l-1} x_j^r \\ x_0 - \sum_{j=l_1}^{l-1} x_j^r & \sum_{j=l_1}^{l-1} x_j^r \leq x_0 \leq \sum_{j=l_1}^{l} x_j^r \\ x_l^r & else. \end{cases} \tag{2.7}$$

The allocations to intermediate nodes $n \in \mathcal{N}\backslash\mathcal{I}_K$ are $x_n = \sum_{m\in\mathcal{S}_n} x_m$.

This approach does not require that assumptions be made about how priorities are determined for nodes on intermediate levels, as it is independent of the structure of the sales hierarchy, so it serves as a benchmark for our analysis of the optimality of *rank based* allocation.

Proposition 2.5 (Optimality of centralized rank based allocations)**.** *For a centralized rank based allocation, the following hold:*

1. *If $x_0 = x_0^r$, then $x_l^{cRB} = x_l^*$ for all $l \in \mathcal{I}_K$.*
2. *$x_l^{cRB} = x_l^*$ for all $l \in \mathcal{I}_K$ and all $x_0 \in [0, x_0^r]$ if and only if*

$$w_l\Big(1 - \lim_{x_l \nearrow x_l^r} G_l(x_l)\Big) \geq w_{l+1}\Big(1 - G_{l+1}(0)\Big) \quad \text{for all } l \in \mathcal{I}_K\backslash\{l_{|\mathcal{I}_K|}\}. \tag{2.8}$$

Proposition 2.5 states that (for $x_0 < x_0^r$) the *centralized rank based* allocation is optimal only if the marginal weighted shortfall reduction that occurs by allocating the x_l^r-th unit to l is larger than the marginal weighted shortfall reduction that is associated with allocating the first unit to $l+1$.

Because the weights w_l are inferred from the service levels (see Lemma 2.2) the marginal weighted shortfall reduction $[1 - G_l(x_l)] \cdot w_l$ is ≤ 1 for $x_l = x_l^r$ and > 1 for $x_l < x_l^r$. Hence, (2.8) can hold only if G_l is non-continuous and has a sufficiently large step at x_l^r. In particular, (2.8) requires

$$\lim_{x_l \nearrow x_l^r} G_l(x_l) \leq 1 - \big[1 - G_{l+1}(0)\big] \cdot \frac{w_{l+1}}{w_l}, \tag{2.9}$$

while from the definition of x_l^r, it follows that $G_l(x_l^r) \geq \alpha_l$. Thus, the *centralized rank based* allocation is always optimal if demand is deterministic.

For stochastic demand we observe that increasing differences in service-level targets decrease $\frac{w_{l+1}}{w_l}$ (because w_l increases in α_l), so Condition (2.9) becomes less restrictive. Hence, we can assume that the performance of *centralized rank based* allocation improves with increasing heterogeneity among the service levels.

As *centralized rank based* allocations are optimal for deterministic demand, we can surmise that decreasing the forecasting accuracy (increasing uncertainty) lowers *centralized rank based* allocations' performance. Assume that G_l and G_{l+1} are continuous, that $G_l(x) = F(\frac{x-\mu_l}{\sigma_l})$, and for reasons of analytical tractability, that $G_{l+1}(0) = 0$. While under a *centralized rank based* allocation, customer group l receives an allocation of $x_l^r = G_l^{-1}(\alpha_l)$ before the next customer group $l+1$ receives its first allocation, the optimal allocation to l before the first allocation to $l+1$ is $G_l^{-1}\big[(\alpha_l - \alpha_{l+1})/(1-\alpha_{l+1})\big]$ (cf. Corollary 2.1). Thus, we can express the excess allocation to customer group l as $\sigma_l\big[F^{-1}((\alpha_l - \alpha_{l+1})/(1-\alpha_{l+1})) - F^{-1}(\alpha_l)\big]$, which clearly increases in the standard deviation σ_l. This supports our conjecture that the optimality gap increases if forecast accuracy decreases.

Overall, the conditions for optimality of the somewhat optimistic case of the *centralized rank based* rule are also very restrictive. In contrast to *per commit*, the *rank based* allocation appears to benefit from increasing service level heterogeneity. As the *centralized rank based* rule can be seen as a best-case reference for the (decentral) *rank based* allocation, we assume that the latter is affected by the service levels' heterogeneity and forecast accuracy in a similar way. Section 2.6 provides numerical evidence for these conjectures.

2.5.2 Advanced Allocation Rules

Our previous analyses revealed that the frequently used conventional allocation rules lead to optimal allocations only under highly restrictive conditions. Companies that use these rules should expect suboptimal results, especially when supply is scarce and service levels and forecast accuracy are heterogeneous. In this section, we develop two new approaches—the *hybrid approach* and the *service level aggregation approach*—based on the insights from Sections 2.4 and 2.5.1 with the intention to strike a balance between practical feasibility and performance.

Hybrid Approach Under optimal allocation, supply is allocated to the customer groups in descending order of their service-level targets (similar to the *rank based* allocation) and based on the probability that customer groups will actually consume the allocated supply (cf. Section 2.4.1). Hence, decentral allocation approaches should exploit more of the information about the customer groups in order to prioritize them according to their service-level targets and the probability that allocated units are actually consumed. Depending on the hierarchy's structure, prioritization is important on different levels. For example, when sub-trees on higher levels of the sales hierarchy are similar, while differences between the customer groups occur at lower levels of the hierarchy (i.e., when there is low between heterogeneity as in Example (b) in Figure 2.1), prioritization will be less important at the higher levels, while lower-level allocations should reflect the differences in service-level targets and demand distributions. Thus, under conditions of low between heterogeneity, a combination of (extended) *per commit* to determine allocations for higher levels of the hierarchy and a local optimal allocation for the lower levels is likely to lead to good overall performance. We call this the *hybrid* approach, which we formalize in Definition 2.6.

Definition 2.6 (Hybrid allocation)**.** *The hybrid allocation from intermediate node $n \in \mathcal{I}_1 \cup \ldots \cup \mathcal{I}_{K-2}$ to its successor $m \in \mathcal{S}_m$ is $x_m^H = x_n x_m^r / x_n^r$. The vector of allocations $\boldsymbol{x}_m^H = (x_l | l \in \mathcal{S}_m)$ from intermediate node $m \in \mathcal{I}_{K-1}$ to its successor (leaf) nodes $l \in \mathcal{S}_m$ is*

$$\boldsymbol{x}_m^H = \underset{\boldsymbol{y} \in \mathbb{R}_{\geq 0}^{|\mathcal{S}_m|}; \|\boldsymbol{y}\|_1 \leq x_m}{argmin} \sum_{l \in \mathcal{S}_m} W_l(y_l),$$

where $W_l(y_l) = w_l \left[L_l(y_l) - L_l(x_l^r) \right]^+$.

Because *extended per commit* is employed on levels $1, \ldots, K-2$, allocations on these levels are based on total required allocations x_n^r of the successor nodes. (See Definition 2.3.) Only on level $K-1$ are allocations determined by solving the decentralized optimal allocation problem for each node using the demand distributions and service-level targets of each successor node (customer group) on level K. Therefore, while the information used and shared throughout the hierarchy under the *extended per commit* and the *hybrid* approaches is identical, the *hybrid* approach uses the information about the individual customer groups on level K more efficiently. Implementing the *hybrid* approach is only slightly more challenging than implementing *per commit* because it requires the planners of nodes $K-1$ to employ local optimization techniques. However, even from a practical point of view, this requirement does not appear to be a major obstacle; as the term suggests, the local optimization can be carried out decentrally and independent of other planners, and we can assume that the planners of level $K-1$ have access to the required information (service-level targets and characteristics of the successor nodes' demand distribution) and the current planning logic of APS can easily be extended to determine local optimal allocations for the customer groups.

Figure 2.6 compares the resulting expected service levels of the *hybrid* approach with the service levels of an optimal allocation for Examples (a) and (b) from Figure 2.3. As the *hybrid* approach is designed for hierarchies that have low between heterogeneity, the resulting expected service levels are closer to the optimum for Example (b) than they are for Example (a). In Example (b), the customer groups with the highest service-level targets (C1 and C2) receive high service levels even when there is relatively scarce supply, while the less important customer groups (C3 and C4) receive their first allocations much later. In Example (a), when there is a medium supply rate, the low-priority customer group C3 receives an allocation much too early and achieves higher service levels than customer group C2 because in Example (a) overall heterogeneity results from differences between the sub-trees on the first level (i.e., from between heterogeneity). On this level, *hybrid* uses *extended per commit* to allocate supply to the individual nodes and does not account for the service levels' heterogeneity.

These results suggest that the performance of the *hybrid* approach depends on between and within heterogeneity. Proposition 2.6 formalizes the conditions under which the *hybrid* approach leads to optimal allocations.

Proposition 2.6 (Optimality of hybrid allocations)**.** *Assume G_l is continuous and strictly increasing for all $l \in \mathcal{I}_K$; then the following hold:*

1. *If $x_0 = x_0^r$ then $x_l^H = x_l^*$ for all $l \in \mathcal{I}_K$.*
2. *$x_l^H = x_l^*$ for all $l \in \mathcal{I}_K$ if*

$$\frac{x_{n'}^*(\lambda)}{x_{n''}^*(\lambda)} = \frac{x_{n'}^r}{x_{n''}^r} \text{ for all } n', n'' \in \mathcal{I}_{K-1}, \lambda \in \left\{ \lambda \mid 0 \leq \sum_{l \in \mathcal{I}_K} x_l^*(\lambda) \leq x_o^r \right\}. \quad (2.10)$$

Proposition 2.6 shows that the optimality of the *hybrid* approach hinges on the optimality of the allocations to the nodes $n \in \mathcal{I}_{K-1}$ on level $K-1$—that is, the level at which we "switch" from *extended per commit* to a decentral optimal allocation. Therefore, if (extended) *per commit* is optimal, the *hybrid* approach is also optimal. However, (2.10) imposes a much milder condition, as it requires only that the sub-trees on level $K-1$ have no between heterogeneity. Assume, without loss of generality, a hierarchy with two sub-trees $\mathcal{N}_{n'}$ and $\mathcal{N}_{n''}$ on level $K-1$ that have no between heterogeneity, so there exists a bijective function $f : \mathcal{L}_{n'} \to \mathcal{L}_{n''}$ that maps the leaf nodes below n' to n'' such that $l' \in \mathcal{L}_{n'}$ and $l'' = f(l') \in \mathcal{L}_{n''}$ have the same service level and the same scaled demand distribution—that is, $\alpha_{l'} = \alpha_{l''}$ and $G_{l'}(x_{l'}) = G_{l''}(a x_{l'})$ for all $x_{l'} \in [0, x_{l'}^r]$ and some $a > 0$. In that case, the required allocations are scaled by $a = x_{n'}^r / x_{n''}^r$, so the optimal allocations to l' and l'' have a fixed ratio; that is, $x_{l''}^* = x_{l'}^* \cdot x_{n''}^r / x_{n'}^r$. Therefore, it is straightforward that (2.10) holds, the *hybrid* allocation is optimal, and the optimality of *hybrid* is independent of the within heterogeneity of the sub-trees on level $K-1$.

However, for increasing levels of between heterogeneity, the deviation between the *extended per commit* allocations to the nodes of level $K-1$ and the optimal allocations increases. Hence, we can expect that the *hybrid* approach's overall performance degrades as between heterogeneity increases. Nevertheless, because a local optimization determines allocations to level K, the *hybrid* approach leads to (weakly) lower weighted shortfalls than the *extended per commit* allocation does. We formalize this result in Proposition 2.7.

Proposition 2.7 (Total weighted shortfall of hybrid and extended per commit)**.** *For any x_0, $W(\boldsymbol{x}^H) \leq W(\boldsymbol{x}^{ePC})$.*

Proposition 2.7 shows that the *hybrid* approach should always be preferred over an *extended per commit* allocation. In Section 2.6 we carry out extensive numerical analyses to demonstrate the performance of the *hybrid* approach.

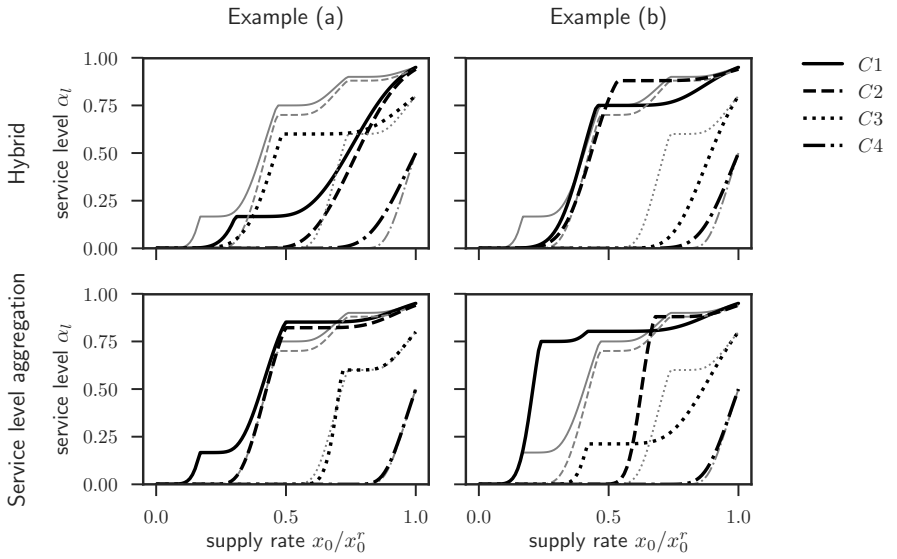


Figure 2.6: Realized service levels of the *hybrid* and the *service level aggregation* approach (black) compared to optimal allocation (gray) for two sales hierarchies.

Service Level Aggregation Approach Our previous analysis revealed that the *hybrid* approach is not appropriate under conditions of high between heterogeneity. In such a setting, the sub-trees of the hierarchy at lower levels are similar in terms of their service-level targets and demand distributions and the differences between the customer groups occur on higher levels of the hierarchy. So we propose an allocation approach termed the *service level aggregation* approach that uses aggregated information on the service levels and demand distributions of the customer groups to prioritize allocations to sub-trees on these higher levels. The rationale behind the *service level aggregation* approach is that similar customer groups can be aggregated into and represented by a single set of parameters (i.e., the aggregate mean, standard deviation, and a service-level target). This aggregated information is then passed on to the predecessor node's planner, who solves a local optimization problem to determine allocations to its successor nodes. Definition 2.7 formalizes the approach.

Definition 2.7 (Service level aggregation approach)**.** *Let $F(t)$ be the cdf of a standardized continuous distribution function with mean* 0 *and standard deviation* 1*, and let $F^{-1}(t)$*

denote its inverse. Choose λ_n *for all* $n \in \mathcal{I}_1 \cup \ldots \mathcal{I}_{K-1}$ *such that* $x_n = \sum_{m \in \mathcal{S}_n} x_m^{sl}(\lambda_n)$ *with*

$$x_m^{sl}(\lambda_n) = \begin{cases} 0 & \text{if } \lambda_n \geq [1 - F(\frac{-\mu_m}{\sigma_m})] \cdot w_m \\ \sigma_m F^{-1}(1 - \frac{\lambda}{w_m}) + \mu_m & \text{else} \end{cases} \qquad \forall m \in \mathcal{S}_n, \tag{2.11}$$

where $\mu_m = \sum_{l \in \mathcal{S}_m} \mu_l$, $\sigma_m = \sum_{l \in \mathcal{S}_m} \sigma_l$, $x_m^r = \sum_{l \in \mathcal{S}_m} x_l^r$ *are the aggregate parameters for sub-tree* m*, and* $w_m = 1/(1 - \alpha_m)$ *and* $\alpha_m = F\big(\frac{x_m^r - \mu}{\sigma_m}\big)$*. Then the service level aggregation allocation of intermediate node* n *to its successor* $m \in \mathcal{S}_n$ *is* $x_m^{sl}(\lambda_n)$*.*

As $F(t)$ is, by definition, continuous and strictly increasing in t, $x_m^{sl}(\lambda_n)$ is continuous and increasing in λ_n. Hence, there is exactly one λ_n that fulfills $x_n = \sum_{m \in \mathcal{S}_n} x_m^{sl}(\lambda_n)$, so Definition 2.7 is well-defined.

The *service level aggregation* approach builds on the results established in Corollary 2.1 but uses aggregated parameters for the demand distributions and shortfall weights for nodes $l \in \mathcal{S}_m$. While determining the aggregated mean μ_m and standard deviation σ_m is comparatively straightforward, there is no single right way to determine the aggregated service level. By inferring service levels from the required allocations of the sub-trees (i.e., $\alpha_m = F\big(\frac{x_m^r - \mu_m}{\sigma_m}\big)$), we ensure that under sufficient supply allocations are optimal.

While it was necessary in the decentralized optimal allocation to share an entire real-valued objective function, the *service level aggregation* approach requires only (total) mean demand μ_m, standard deviation σ_m, and required allocations x_m^r to be shared with planners on the next-higher level. Any other information (e.g., the aggregated service level and corresponding weights) can be inferred from these parameters. Calculating the allocation itself is then based only on local information and it can be solved efficiently, so it can be incorporated into the planning logic of APS.

In Figure 2.6 we compare the expected per-group service levels associated with the allocations that are based on the *service level aggregation* approach[6] with the optimal service levels for Examples (a) and (b) in Figure 2.3. We observe that the resulting service levels of Example (a) are close to optimal, while in Example (b) the difference is significant. Because in Example (b) the sub-trees have a high within heterogeneity, the single set of parameters does not represent the sub-trees

[6] We use the standard normal distribution (i.e., $F(z) = \Phi(z)$) to calculate the allocations, as the demand of the customer groups follows a normal distribution.

correctly. Consequently, the sub-tree with customer groups C1 and C3 is prioritized over the sub-tree with groups C2 and C4, which results in C2's receiving its allocation too late and reaching acceptable service levels only at very high supply rates. In Example (a), the within heterogeneity is low and the aggregated information represents the customer groups within each sub-tree accurately. Therefore, the sub-tree that contains the high service-level customer groups C1 and C2 is correctly prioritized.

This example shows that the performance of the *service level aggregation* approach depends on the sales hierarchy's structure. Proposition 2.8 describes the conditions under which the *service level aggregation* approach allocates optimally.

Proposition 2.8 (Optimality of service level aggregation allocations)**.** *For a service level aggregation allocation, the following hold:*

1. *If* $x_0 = x_0^r$ *then* $x_l^{sl} = x_l^*$ *for all* $l \in \mathcal{I}_K$.

2. $x_l^{sl} = x_l^*$ *for all* $l \in \mathcal{I}_K$ *and all* $x_0 \in [0, x_0^r]$ *if, for all intermediate nodes* $n \in \mathcal{I}_1$*, the following hold:*

$$G_l(x_l) = F\left(\frac{x_l/\mu_l - 1}{CV_n}\right) \qquad \text{for all } l \in \mathcal{L}_n, x_l \in [0, x_l^r]$$
$$\alpha_l = \alpha_n \qquad \text{for all } l \in \mathcal{L}_n.$$

Proposition 2.8 shows that the *service level aggregation* approach leads to optimal allocations under scarce supply if all demand distributions are based on the same standardized distribution $F(\cdot)$ and the customer groups in each sub-tree on the first level of the sales hierarchy have homogeneous service-level targets.[7] In this case, the objective functions of the intermediate nodes n are accurately represented by distributions $F\left(\frac{x_n-\mu_n}{\sigma_n}\right)$ and service levels α_n, so the resulting allocations are optimal. When within heterogeneity increases, the differences between the objective functions and their approximations increase, and we can expect a decreasing performance of the *service level aggregation* approach. Section 2.6 quantifies the impact of within heterogeneity on the *service level aggregation* approach and compares it with the other allocation rules that have been proposed in this section.

[7]The proposition demands a constant CV for the demand distributions in a sub-tree. This demand is only a technical assumption since, in practice, $G_l(0) = F(-1/CV_l) \approx 0$, so allocations are (close to) optimal for heterogeneous CVs.

Table 2.1: Performance drivers of decentral allocation rules.

	supply rate $x_0/x_0^r \to 1$	forecast accuracy	forecast heterogeneity	(overall) service-level heterogeneity	within heterogeneity	between heterogeneity
Per commit			−	−		
Extended per commit	+		−	−		
Centralized rank based	+	+		+		
Rank based	+	+		+		
Hybrid	+					−
Service level aggregation	+				−	

"+"/"−" indicates a positive/negative impact on the allocation's performance.

2.6 Numerical Analysis

Section 2.5 identified a number of drivers that impact the performance of the conventional allocation rules and our new allocation approaches. Table 2.1 provides a high-level overview of these drivers and how they affect overall performance. In this section we carry out a numerical study to compare the performance of the allocation approaches and to quantify the impact of these performance drivers. Our first objective is to shed light on the optimality gap of the individual allocation rules and how this gap depends on the performance drivers listed in Table 2.1. Our second (and more important) objective is to integrate these results into a comprehensive framework that allows decision-makers to assess when and under what conditions they can or should employ certain approaches for allocation planning.

In Section 2.6.1 we show how we operationalize the performance drivers described in Table 2.1, and in Section 2.6.2 we show how we measure the allocation approaches' performance. Section 2.6.3 outlines how we conducted our numerical experiments, in which we vary forecast accuracy, overall heterogeneity, and between and within heterogeneity. In the subsequent sections we compare the approaches' performance in each scenario and quantify the impact of the performance drivers on the performance of the allocation rule.

2.6.1 Operationalization of Performance Drivers

This section explains how we operationalized the performance drivers summarized in Table 2.1.

Supply rate. As in Section 2.5, we measure the supply rate as x_0 / x_0^r.

Forecast accuracy. From a practical perspective, we want to capture differences in forecasting performance. In our setting, the coefficient of variation (CV_l) is an appropriate proxy for the forecast performance of customer group l.

As outlined in Section 2.5, heterogeneity can occur in the form of differences in the ability to forecast the individual customer groups' demand and service-level targets. To isolate the effects of these two types of heterogeneity, we introduce measures that capture forecast heterogeneity and the service levels' heterogeneity separately.

Forecast heterogeneity. We measure the forecasts' heterogeneity using the demand-weighted standard deviation of the CV, relative to the average CV, to ensure that the measure is robust against scaling of the forecasts' accuracy:

Definition 2.8 (Forecast heterogeneity)**.** *The forecast heterogeneity for customers $l \in \mathcal{I}_K$ is*

$$H_{CV} = \frac{1}{\overline{CV}} \sqrt{\frac{\sum_{l \in \mathcal{I}_K} \mu_l (CV_l - \overline{CV})^2}{\mu_0}},$$

where $\overline{CV} = \sum_{l \in \mathcal{I}_K} \frac{\mu_l}{\mu_0} CV_l$ is the demand-weighted average of the CV.

Service level heterogeneity. To measure the service level heterogeneity, we use the service level-inferred shortfall weights instead of the service levels themselves because the weights reflect the differences between the customer groups more precisely. Hence, we use the demand-weighted standard deviation of the shortfall weights, standardized relative to the (demand-weighted) average shortfall weight, as our measure for the service level heterogeneity:

Definition 2.9 (Service level heterogeneity)**.** *The service level heterogeneity for customers $l \in \mathcal{I}_K$ is*

$$H_{SL} = \frac{1}{\bar{w}} std_{SL}^{\mathcal{I}_K},$$

where $std_{SL}^{\mathcal{I}_K} = \sqrt{\frac{\sum_{l \in \mathcal{I}_K} \mu_l (w_l - \bar{w})^2}{\mu_0}}$ *is the demand-weighted standard deviation of the service levels' inferred weights of nodes* $\mathcal{I}_K$*, and* $\bar{w} = \sum_{l \in \mathcal{I}_K} \frac{\mu_l}{\mu_0} w_l$ *denotes the demand-weighted average of the shortfall-weights.*

We illustrate this and the following heterogeneity measures using an example at the end of this section.

Within heterogeneity. The within heterogeneity is the heterogeneity of the customer groups in a sub-tree. To operationalize within heterogeneity, we modify Definition 2.9 to measure the heterogeneity of each sub-tree's service levels. As we are interested in the within heterogeneity of the entire hierarchy, rather than that of individual sub-trees, Definition 2.10 uses the demand-weighted sum of the sub-trees' service-level heterogeneities as a measure for the within heterogeneity.

Definition 2.10 (Within heterogeneity)**.** *Let* $\mathcal{L}_n$ *denote the leaf nodes to node* n*; then the within heterogeneity of the hierarchy is*

$$H_{SL}^{within} = \frac{1}{\bar{w}} \sum_{n \in \mathcal{I}_1} \frac{\mu_n}{\mu_0} std_{SL}^{\mathcal{L}_n}. \tag{2.12}$$

In (2.12) $std_{SL}^{\mathcal{L}_n}$ measures the demand-weighted standard deviation of the shortfall weights in the individual sub-trees to nodes $n \in \mathcal{I}_1$ according to Definition 2.9.

Between heterogeneity. The between heterogeneity reflects the differences between sub-trees. Recall our example from Figure 2.3, where the sub-trees in Example (a) exhibit high between heterogeneity because the individual service-level targets in the two sub-trees differ substantially, so the sub-trees' average service-level targets differ. In Example (b), the sub-trees are much more homogeneous because the service-level targets of the customer groups in sub-tree 1 are similar to the service-level targets of the customer groups in sub-tree 2. As a consequence, the average service level and the heterogeneity of the sub-trees are similar. This example reveals some basic requirements that a measure of between heterogeneity must fulfill: First, the between heterogeneity should be zero if all sub-trees are identical in terms of their weighted average service levels and the heterogeneity of their service levels. In this case, the sub-trees' within heterogeneity are identical and the overall heterogeneity can be fully explained by the sub-trees' within heterogeneities of the sub-trees. Second, a measure of between heterogeneity should increase as the individual sub-trees exhibit greater differences in terms of their average service levels and the heterogeneity of their service levels.

The standardized (squared) sum of differences between the sub-trees' standard deviation of the shortfall weights and the overall hierarchy's standard deviation of the shortfall weights is an adequate measure for between heterogeneity that meets these requirements. More formally, the between heterogeneity is defined in Definition 2.11:

Definition 2.11 (Between heterogeneity)**.** *The between heterogeneity of the hierarchy is*

$$H_{SL}^{between} = \frac{1}{\bar{w}} \sqrt{\sum_{n \in \mathcal{I}_1} \frac{\mu_n}{\mu_0} (std_{SL}^{\mathcal{L}_n} - std_{SL}^{\mathcal{I}_K})^2}.$$

The intuition behind this measure and the other heterogeneity measures becomes clearer when our two sample hierarchies from Figure 2.3 are considered: As both use the same customer groups, the overall heterogeneity of the service levels is 0.69 in both hierarchies. However, in Example (a) each sub-tree has a small within heterogeneity and we obtain $H_{SL}^{within} = 0.21$, while in Example (b) we observe $H_{SL}^{within} = 0.98$. Regarding the between heterogeneity, we calculate in Example (a) that the two sub-trees' weighted standard deviations of the shortfall weights are 1.6 and 1.5, so they are substantially lower than the overall standard deviation (7.58). Hence, Example (a) has a relatively large between heterogeneity of 0.55. In Example (b), the two sub-trees' weighted standard deviations are similar (at 7.50 and 7.33) to the entire hierarchy's standard deviation (7.58). Therefore, the between heterogeneity is low (0.02).

2.6.2 Performance Measures

We propose two measures with which to compare the allocation approaches' performance with that of the optimal allocation that serves as our benchmark. We use the "absolute gap to optimality" (*AGO*) to compare the allocation approaches at a specific supply rate. The *AGO* is defined as the absolute difference between an allocation approach's total weighted shortfall and the weighted shortfall associated with an optimal allocation averaged over all hierarchy variants.

Definition 2.12 (Absolute gap to optimality)**.** *The absolute gap to optimality (AGO) for allocation method a is*

$$AGO_a = \frac{1}{|V|} \sum_{v \in V} W_v^a(x_0) - W^*(x_0),$$

Table 2.2: Parameterization of the analyses.

Analyses	Performance driver setting		
	Service level heterogeneity	Forecast accuracy	Forecast heterogeneity
Baseline	0.56	0.20	0.00
Forecast accuracy	0.56	0.10, 0.20, . . . , 0.80	0.00
Forecast heterogeneity	0.00	0.50	0.00, 0.05,. . . , 0.6
Service level heterogeneity	0.00, 0.05,. . . , 0.65	0.20	0.00

where $W_v^a(x_0)$ and $W^*(x_0)$ denote the weighted shortfall of allocation method a and the optimal allocation for a supply of x_0, respectively, and V denotes the set of the hierarchy variants.

The *AGO* allows one to compare the performance only at a specific level of supply x_0. To compare the general performance of the various allocation approaches, we define an additional performance measure, the "relative average gap to optimality" (*RAGO*) as the average performance over all levels of supply:

Definition 2.13 (Relative average gap to optimality)**.** *The relative average gap to optimality (RAGO) of allocation method a is*

$$RAGO_a = \frac{1}{|V|} \sum_{v \in V} \frac{\sum_{x_0 \in [0, x_0^r]} W_v^a(x_0)}{\sum_{x_0 \in [0, x_0^r]} W^*(x_0)} - 1.$$

2.6.3 Experimental Design

To assess how the performance of the allocation approaches depends on the performance drivers, we first carry out four analyses. Table 2.2 provides an overview of these analyses and the corresponding parameters. Our baseline scenario is a setting with moderate heterogeneity in service levels and constant forecast accuracy across all customer groups. In Section 2.6.4, we use the results of this baseline analysis to provide first insights into the allocation approaches' performance. Thereafter, in Section 2.6.5, we carry out three analyses (Table 2.2) in which we vary one performance driver at a time in order to isolate the effect each driver has on the approaches' performance.

Here we describe the steps in our evaluation of the allocation approaches' performance for the analyses and parameters described in Table 2.2.

1. *Determine customer parametrization:* In all analyses we use six customer groups with normally distributed demand with mean 10. To determine the standard deviation, we rearrange the formula in Definition 2.8 to give us equally spaced CVs for each customer group, leading to the required forecast accuracy and heterogeneity. To determine the service-level targets we rearrange the formula in Definition 2.9 to calculate equally spaced shortfall weights (which can be converted to service levels) with a constant maximum of 50 (corresponding to a service-level target of 0.98) that lead to the desired heterogeneity of service levels.[8] This approach results in a parametrization matrix for each scenario that contains the shortfall weights, standard deviations, and mean demand for all six customer groups.

2. *Generate hierarchies:* For our numerical experiments we use a hierarchy with nine nodes, where node 0 is the root node, nodes $\{1, 2\}$ are the intermediate nodes, and nodes $\{3, \ldots, 8\}$ are the leaf nodes that represent the customer groups. As some allocation rules (i.e., *rank based*, *hybrid*, and *service level aggregation*) are influenced by the hierarchy's structure, we take special care when generating the hierarchies. Certain approaches might perform better under some structures, diluting our results. Hence, we systematically permute the six leaf nodes to obtain all possible structures for our nine-node setup, leading to ten symmetric, fifteen moderately asymmetric, and six asymmetric variants of the sales hierarchy. (See Figure 2.7 for an illustration.) We then assign the parametrized customer groups to the leaf nodes and obtain thirty-one fully specified hierarchy variants (denoted by $v \in V$).

3. *Determine the allocations:* We implemented all allocation approaches from Section 2.5[9] and the optimal central allocation (Section 2.4.1) in Python and used these approaches $a \in A$ to obtain the allocations for each hierarchy variant $v \in V$ and supply rate $s \in S = \{0, 0.01, \ldots, 1\}$.

4. *Evaluate the performance of the allocation approaches:* From the allocations determined in Step 4 we determine the performance measures as described in Section 2.6.2. In this step, we also compute the within and between heterogeneity for each hierarchy variant according to Definitions 2.10 and 2.11.

[8] The detailed formulae to obtain the parameters are shown in Appendix A.2.

[9] For the (centralized) *rank based* allocation, we used the average service level to determine ranks on the intermediate levels.

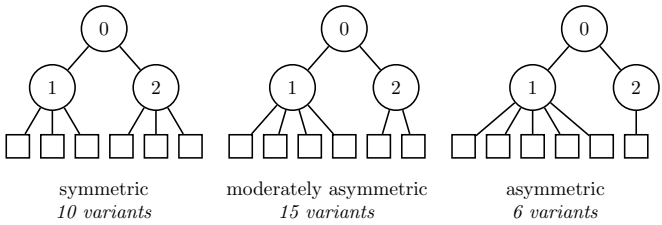


Figure 2.7: Variations of a three-stage hierarchy with six customer groups.

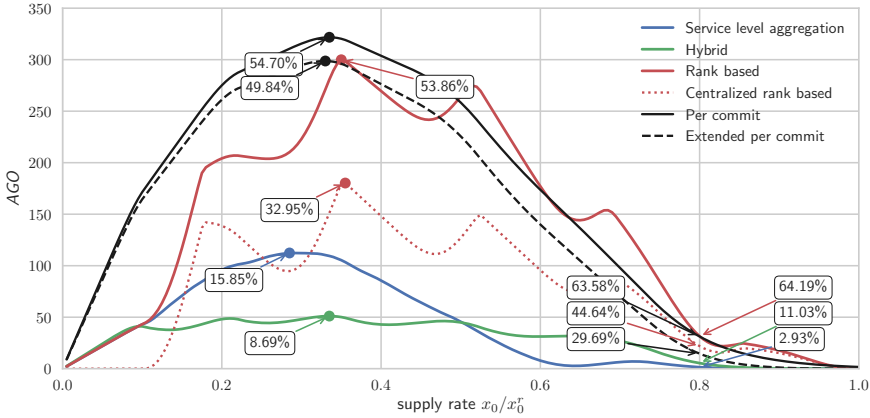


Figure 2.8: *AGO* for all allocation methods for the baseline scenario; annotations display the relative gap to optimality at a supply rate of 0.8 and at the maximum *AGO*.

2.6.4 Baseline

Figure 2.8 shows the baseline *AGO* for all allocation approaches, dependent on the supply rate. At a supply rate of 1, all allocation approaches except *per commit* allocate optimally (cf. Section 2.5) so they lead to an *AGO* of zero. *Per commit* leads to a very small and, in this instance, negligible *AGO* of 1.71. (Because the shortfall weights in the baseline are between 5 and 50, that suggests much less than one unit of additional shortfall is caused by the sub-optimal allocation.)

As supply rates decrease (i.e., shortage increases), the *AGOs* increase across all allocation approaches, although at different rates.[10] Consider, for example, the

[10]The non-monotonic behavior of the *AGOs* that are associated with *rank based* allocation can be explained by the particular sequential allocation logic: Performance increases relative to the optimal solution whenever an additional customer with the next lower priority receives an allocation.

situation of a moderate shortage (e.g., at a supply rate of 0.8). Here, *per commit* and *rank based* exhibit the highest *AGOs* of 31.2 and 31.5, respectively, while our advanced approaches (*hybrid* and *service level aggregation*) lead to *AGOs* that are still close to zero. As we suggested in Section 2.5.1, *extended per commit* has a relatively good performance at moderate shortage. Although the *AGOs* of *rank based* and *per commit* appear to still be relatively low, the corresponding relative gap to optimality is considerable (64%). For the *hybrid* and the *service level aggregation* approaches, the gaps to optimality are only 11 percent and 3 percent, respectively.

For all allocation rules, the *AGO* reaches a maximum at low supply rates (between 0.2 and 0.4). However, the relative gap for the *rank based* and the *per commit* rule is lower at low supply rates than it is for moderate shortages (e.g., for 0.4) because, at these low levels of supply, the optimal allocation causes a substantial (weighted) shortfall. The *hybrid* approach appears to be less sensitive to low supply rates and, in contrast to the other allocation approaches, does not exhibit a clear maximum *AGO*.

AGOs decrease at low supply rates; simply speaking, supply is so constrained that the performance differences between the optimal allocation and the allocation rules decline. Obviously, *AGOs* are zero when no supply is available.

The advanced allocation approaches (*hybrid* and *service level aggregation*) outperform the conventional rules (*per commit* and *rank based*) for supply rates that are larger than 0.2. In fact, the *AGOs* of the *hybrid* and the *service level aggregation* approaches are substantially lower when supply is strongly constrained, that is, at supply rates of 0.2 to 0.7. Comparing the two advanced allocation approaches shows that, for moderately constrained supply (supply rates that are larger than 0.6), the *service level aggregation* approach leads to higher gaps than the *hybrid* approach does, as the *hybrid* approach is robust toward the supply rate and produces comparably low *AGOs* across all supply rates.

The allocations (and, presumably, the performance) of the *hybrid*, the *service level aggregation*, and the *rank based* approaches are hierarchy-dependent. We now explore for the baseline scenario how the performance of these allocation approaches depends on the structure (more specifically, the symmetry) of the hierarchy (Figure 2.7). The box plot in Figure 2.9 plots the *RAGOs* of these allocation approaches for asymmetric, moderately asymmetric, and symmetric hierarchies, and contrasts them with the *RAGOs* of the *per commit*, the *extended per commit*, and the *centralized rank based* rules, which are independent of the hierarchy. Under all

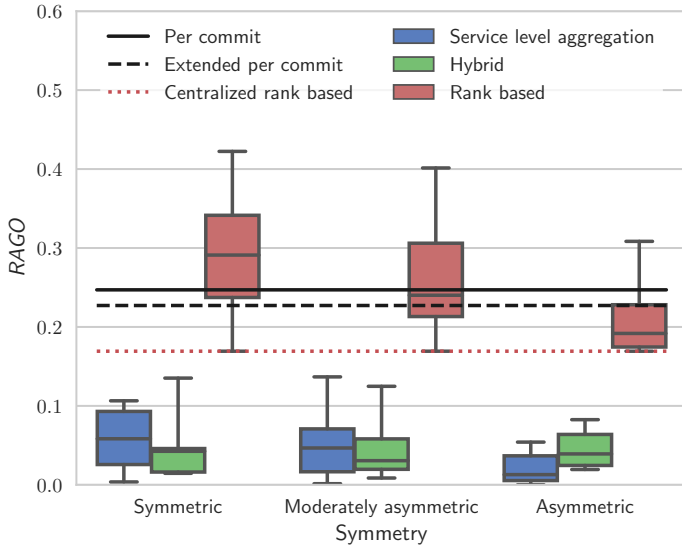


Figure 2.9: Effect of the symmetry of the hierarchy on the *RAGO* (whiskers extend to the min/max, boxes to the first and third quantile).

hierarchy structures the advanced allocation approaches outperform the conventional allocation rules.

While the performance of the *hybrid* approach is almost unaffected by the hierarchy's symmetry, the *RAGO* for *rank based* and the *service level aggregation* approaches is higher for more symmetric hierarchies. However, the overall effect—especially for the *service level aggregation* approach—is relatively small and does not explain the performance differences between the hierarchy variants well. Section 2.6.5 shows that the performance of the *service level aggregation* approach depends on the hierarchy's within heterogeneity, so we can assume that the hierarchy's symmetry impacts the performance only indirectly. For asymmetric hierarchies, the size of the smallest sub-tree decreases, so the sub-tree becomes more homogeneous, decreasing the within heterogeneity and explaining the slight positive effect on the *service level aggregation* approach. We conclude that the hierarchy's symmetry does not have a substantial direct effect on the allocation approaches' performance.

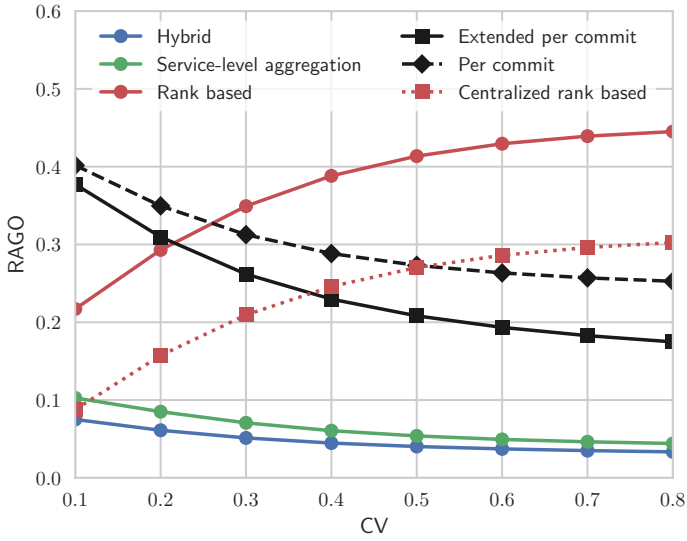


Figure 2.10: Effect of different levels of CV on the *RAGO*.

Our initial results for the baseline scenario suggest that our advanced allocation approaches may be suitable for remedying the conventional allocation rules' shortcomings while being relatively easy to implement.

2.6.5 Influence of Performance Drivers

In this section we explore whether and how the initial results of Section 2.6.4 depend on the remaining performance drivers that were summarized in Table 2.1.

Forecasting Accuracy Figure 2.10 plots the *RAGO* for all allocation approaches under varying CVs. For the *rank based* and the *centralized rank based* rule, performance decreases in the CV, which is in line with our conjecture in Section 2.5.1. The performance of all other allocation approaches increases in the CV. The *RAGOs* of the *per commit* and the *extended per commit* rules reduce from 0.4 and 0.38 (at a CV of 0.1), respectively, by about half to 0.25 and 0.17 (at a CV of 0.8), respectively. The advanced allocation approaches show similar effects, although on a lower overall *RAGO* level (from 0.08 to 0.03 for the *hybrid* approach and 0.10 to

0.04 for the *service level aggregation* approach). It may seem counterintuitive that the advanced allocation approaches' performance increases relative to the optimum as forecast accuracy declines (CVs increase). However, under high demand uncertainty, misallocated supply is more likely to be consumed than under more certain supply conditions. Hence, the effect of suboptimal allocations on the *RAGO* declines as demand uncertainty increases. *Per commit*, *extended per commit*, and our advanced approaches benefit from this effect. With the exception of the *centralized rank based* rule at a low level of demand uncertainty ($CV = 0.1$), the advanced allocation approaches outperform the conventional allocation rules.

Forecast Heterogeneity Next, we analyze the allocation approaches' sensitivity to the forecast heterogeneity. To rule out any confounding effects of heterogeneity among the service levels, all customers are assigned a service-level requirement of 0.98 (making the service levels homogeneous, cf. Table 2.2). Taking this step suggests that we must exclude *rank based* allocations from this particular analysis.

In Figure 2.11 we plot the *RAGO* for the allocation approaches for increasing levels of forecast heterogeneity. All allocation approaches' *RAGOs* increase as the forecast heterogeneity increases, with the exception of the *service level aggregation* approach, which has a *RAGO* of close to zero at all levels of forecast heterogeneity.

Extended per commit and *per commit* have the largest sensitivity to forecast heterogeneity, which is in line with our results in Section 2.5.1. The *RAGO* of the *hybrid* approach also increases at higher levels of forecast heterogeneity, albeit a considerably lower overall increase than that of *per commit*. However, with a maximum of less than 0.06, the *RAGOs* across all allocation rules and all levels of CV are small compared to our previous analyses. Nevertheless, our advanced allocation approaches clearly outperform the conventional allocation rules.

Service Level Heterogeneity Figure 2.12 shows how the service level heterogeneity impacts the *RAGOs* of the allocation approaches.

When the service level heterogeneity is zero, all service levels are equal and all allocation approaches but the *rank based* approach[11] lead to optimal allocations (cf. Section 2.5). When the service level heterogeneity is low, the *rank based* and *centralized rank based* rules have a high *RAGO* (e.g., approximately 0.28 when the service level heterogeneity is 0.1), which is again in line with the analytical results

[11] For a heterogeneity of zero, all customers are identical and no rank exists, so we cannot determine the allocations for the *rank based* rules.

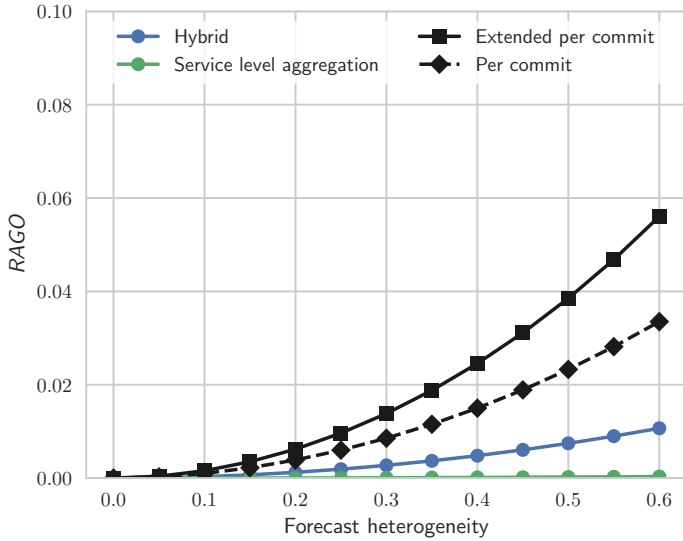


Figure 2.11: Effect of forecast heterogeneity on the *RAGO*.

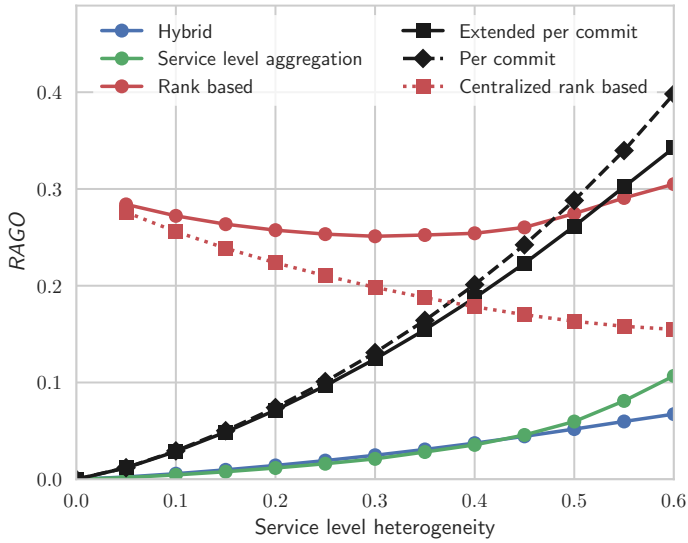


Figure 2.12: Effect of service level heterogeneity on the *RAGO*.

established in Section 2.5. As the heterogeneity of the service levels increases, the *centralized rank based* allocation's *RAGO* strictly decreases, while for the decentralized *rank based* allocation, the *RAGO* first decreases and then increases. The latter result is caused by the interplay of two effects: *rank based* works better when customers are heterogeneous in terms of their service-level requirements, which is reflected in the *centralized rank based* rule's strictly decreasing *RAGO*. However, the decentralized *rank based* allocation requires a rank-aggregation on the intermediate nodes (Section 2.5.1), which suggests that some of the information about the customers' heterogeneity is lost. With increasing heterogeneity, the negative impact of this information loss on performance becomes more pronounced and outweighs the former effect at some point. All other rules' performance decreases with increasing heterogeneity in service levels, although at varying rates. As predicted in Section 2.5.1, such heterogeneity has a strong detrimental impact on *per commit's* performance. *Per commit's RAGO* is zero when the service level heterogeneity is zero, but it grows at an increasing rate as that heterogeneity increases. In contrast, the advanced allocation approaches are less sensitive to an increase in the service level heterogeneity. Up to a service level heterogeneity of 0.45, both *hybrid* and *service level aggregation* perform equally well and exhibit *RAGOs* below 0.05. At high levels of service level heterogeneity, the *hybrid* approach appears to perform better than the *service level aggregation* approach. Therefore, both advanced allocation approaches outperform their conventional counterparts when the service level heterogeneity is larger than zero.

Between and Within Heterogeneity For the *hybrid* and the *service level aggregation* approaches, we identified between and within heterogeneity as potential performance drivers. To determine their effects on the performance of both allocation rules, we re-use the service level heterogeneity scenarios and calculate the between and within heterogeneity of the thirty-one hierarchy variants for each scenario. To illustrate how overall, within, and between heterogeneity are related in the individual instances, Figure 2.13 plots both heterogeneity measures on the x- and y-axes and color-codes the (overall) service level heterogeneity. A radial pattern begins at the origin (at an overall heterogeneity of 0). All instances on an individual "ray" exhibit the same structure, with linearly increasing levels of (overall) service level heterogeneity, as described in Section 2.6.2. The overall heterogeneity of the service levels is associated with differing levels of within and between heterogeneity. For example, for hierarchy variant "[145][236]" (where customer groups 1, 4, and 5 are

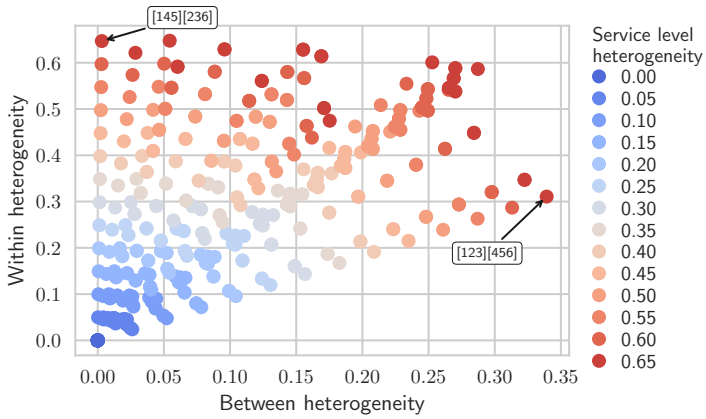
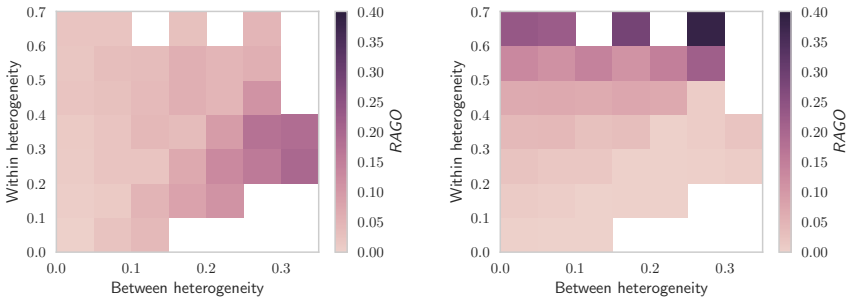


Figure 2.13: Relative between and within heterogeneity of the service level heterogeneity scenarios.

in sub-tree 1, and customer groups 2, 3, and 6 are in sub-tree 2), the increase in the overall service level heterogeneity is almost fully associated with an increase in the within heterogeneity, while in hierarchy variant "[123][456]," the increase is more strongly associated with an increase in the between heterogeneity. This variation in within heterogeneity and between heterogeneity across the hierarchy variants allows us to assess the *hybrid* and the *service level aggregation* approaches' performance at various levels of the respective heterogeneity. The corresponding results are displayed in Figure 2.14.

The *hybrid* approach (Figure 2.14a) leads to a *RAGO* that is close to zero for low between heterogeneities and only to a *RAGO* that is higher than 0.1 if the between heterogeneity is higher than 0.2. This result supports our conjecture that between heterogeneity is a major performance driver for the *hybrid* approach. However, when between heterogeneity is high (i.e., higher than 0.4), performance improves at higher levels of within heterogeneity.

This seemingly counterintuitive result can be explained by the properties of the instances in this parameter range. A hierarchy with high between heterogeneity and zero within heterogeneity is given when all customers in one sub-tree have equally high service-level requirements, and all customers in the other sub-tree have equally low requirements. This scenario would be the worst case for the *hybrid* approach. Increasing the within heterogeneity at the same high level of between heterogeneity, then, is beneficial for the *hybrid* approach, as despite the



(a) Hybrid

(b) Service level aggregation

Figure 2.14: Effect of within and between service level heterogeneity on the *RAGOs* of the *hybrid* and the *service level aggregation* approaches. (Each tile of the heat maps represents the *RAGO* averaged over all instances that fall into the corresponding range of between and within heterogeneity.)

high between heterogeneity, the *hybrid* approach would have "better" allocations to the individual customers in the sub-trees. Therefore, at constant high levels of between heterogeneity, the *hybrid* approach's performance improves for increasing levels of within heterogeneity.

We expect the *service level aggregation* approach's performance to be negatively impacted by increasing within heterogeneity, as Figure 2.14b shows. At low levels of within heterogeneity, the *RAGO* is consistently below 0.1, independent of the between heterogeneity. It increases at moderate levels of within heterogeneity (between 0.3 and 0.5) and takes on high values of up to 0.4 at high levels of within heterogeneity (beyond 0.5). Overall, the *service level aggregation* approach appears to be robust with respect to between heterogeneity.

The results for the *hybrid* and the *service level aggregation* approaches are synthesized in Figure 2.15, which superimposes the individual results shown in Figure 2.14. For each combination of within heterogeneity and between heterogeneity, we identify the superior rule (i.e., the rule that leads to the lower *RAGO*) and depict the corresponding *RAGO*.

The results in Figure 2.15 explain which of the two advanced allocation approaches should be selected based on within heterogeneity and between heterogeneity. The *hybrid* (*service level aggregation*) approach should be chosen for low to medium between (within) heterogeneity; under these conditions the "correct" advanced allocation rule performs well, that is, they both lead to *RAGOs* of close

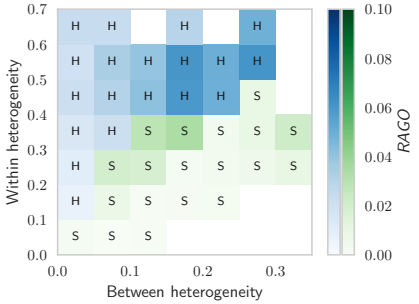


Figure 2.15: *RAGO* of the best rule for varying levels of between and within heterogeneity. (*Hybrid* approach in blue and marked with "H," *service level aggregation* approach in green and marked with "S.")

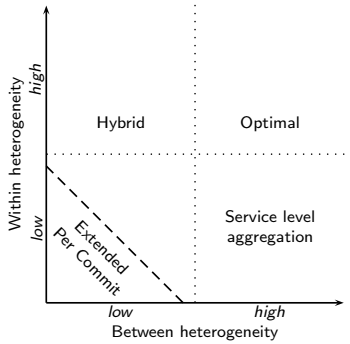


Figure 2.16: Decision matrix for choosing allocation planning approaches.

to zero. However, the *hybrid* and the *service level aggregation* approaches lead to higher *RAGOs* when moderate to high levels of within heterogeneity and between heterogeneity coincide. Even in these instances, *RAGOs* are still at fairly low levels (never exceeding 0.1) compared to the results of the conventional allocation rules (Figure 2.12).

To conclude this section, we summarize the main findings of our numerical analyses. Our results show that *rank based* leads to a detrimental performance in almost any situation and that even *centralized rank based* should only be used when demands can be forecasted very accurately ($CV \leq 0.1$). *Per commit* and *extended per commit* lead to good, or at least acceptable, results when the overall service levels' heterogeneity is low and supply rates are high. At moderate levels of heterogeneity both lead to considerable performance losses. Under moderate to high heterogeneity of service levels, the decision-maker can improve performance substantially by choosing the right advanced allocation approach: If between heterogeneity is low, the *hybrid* approach will result in good allocations, and the decision-maker can expect close to optimal performance; when within heterogeneity is low, the *service level aggregation* approach can be expected to yield good results. When the levels of both types of heterogeneity are moderate to high at the same time, the advanced allocation rules still lead to reasonable performance, although it is not close to optimal. Achieving even better performance will require more involved approaches,

such as the decentralized optimal allocation described in Section 2.4.2. Figure 2.16 summarizes our findings and recommendations.

2.7 Conclusion

This paper addresses centralized and decentralized approaches to the allocation of scarce supply in a sales hierarchy with stochastic customer demand and service-level targets. Its analytical and numerical contributions provide a number of methodical and practical insights. First, Section 2.4 provides an analytical solution for the case in which an omniscient central planner plans allocations. It covers a broad range of possible demand distributions, including discrete and other not-absolutely-continuous demand distributions. Optimal solutions to this problem are natural benchmarks for any decentralized allocation procedure. We also show that the optimal solution can be achieved in a decentralized planning regime, although at the cost of excessive information-sharing and somewhat involved computations.

Second, Section 2.5 provides a detailed discussion of the common decentralized allocation rules, and building on the insights developed from the analysis of the central-planner case, introduces two new decentralized allocation approaches: the *hybrid* approach and the *service level aggregation* approach. While these new approaches are simple enough to be implemented in a hierarchical sales organization, they result in improved supply allocations over those of the conventional approaches because they exploit more of the information about the customer groups.

Taken together, the analytical results presented in Section 2.5 and the numerical results presented in Section 2.6 show that the two new allocation approaches typically outperform the conventional approaches. They also explain when to use which approach: The *hybrid* allocation performs well under low between heterogeneity, and the *service level aggregation* approach performs well under low within heterogeneity. These insights are especially valuable for decision-makers who work in companies with hierarchical sales organizations, as they help them to identify the best allocation approach for their organizations using two simple heterogeneity measures that can be computed easily and that typically remain stable over time.

The research presented in this paper has a number of limitations. Among them, we consider a single period setting, while allocation decisions are actually interrelated across multiple subsequent periods. For example, unused supply is likely to be carried over to the next period, and unfilled demand may be backlogged

and served later. Multi-period allocation planning models will require a number of additional assumptions (e.g., regarding backorder-clearing mechanisms, the option of trans-shipments, supply/inventory-netting procedures) that are not difficult to implement but that complicate the comparison of the approaches to allocation planning and make it even more difficult to obtain analytically tractable results. We perceive our models, approaches, and insights as a starting point for carrying out multi-period analyses. Although developed for the single-period case, our advanced allocation rules can be enhanced by various features that are relevant to a multi-period setting. We leave these extensions and further analyses to future research.

Chapter 3

Single-Period Stochastic Demand Fulfillment in Customer Hierarchies[1]

3.1 Introduction

This paper addresses the problem of allocating scarce supply to hierarchically structured customer segments so as to maximize profitability. The problem connects the supply chain planning task of profit-oriented demand fulfillment (DF) to the business reality of multilevel customer hierarchies.

DF aims to optimally match customer orders with available resources. In make-to-stock production systems, DF comprises fulfilling customer orders from inventory. Since acceptable customer response times are shorter than production lead times, in this setting, supply is essentially fixed when demand materializes (Fleischmann and Meyr, 2004). Therefore, firms face the risk of short-term supply shortages, especially when demand is uncertain. Under a first-come-first-served (FCFS) fulfillment approach, any customer may suffer from such shortages. However, customers commonly differ in their importance and profitability. FCFS demand fulfillment ignores these differences and therefore performs poorly under heterogeneous demand (Ketikidis et al., 2006; Meyr, 2009; Barut and Sridharan, 2005).

[1] This chapter is co-authored by Moritz Fleischmann, Maryam Nouri and Richard Pibernik.

Revenue management (RM) approaches to demand fulfillment address this deficiency (Quante et al., 2009b). Such approaches divide the overall customer base into different segments based on profitability or strategic importance. The DF problem is then solved in a two-stage process. First, in the allocation planning stage, available-to-promise quantities (ATP) are determined and allocated as quotas to different customer segments. Second, in the order promising stage, these quotas are consumed by fulfilling realized orders from the corresponding customer segments (Ball et al., 2004; Kilger and Meyr, 2008). Orders exceeding the corresponding quota are lost or deferred to less constrained periods. This process prioritizes more profitable orders and avoids depleting scarce supply by fulfilling less profitable orders.

Available RM approaches to demand fulfillment rely on a one-dimensional ranking of the customer segments. In reality, however, customer segments commonly have a multilevel hierarchical structure that reflects the structure of the sales organization. A typical customer hierarchy includes different geographies, different distribution channels, and different customer groups, similar to that shown in Figure 3.1. Roitsch and Meyr (2015) study an example of such a hierarchy in the downstream business of the European oil industry. The industry faces long lead times, and after deciding about the crude oil supply, quantities cannot easily be changed. The available supply is then iteratively allocated to different business units in 14 different countries, producing different products for different customers and yielding different profits.

In such hierarchies, there is no direct ranking of individual customer segments. Instead, allocation planning is an iterative and decentralized process in which higher-level sales quotas are disaggregated one level at a time by multiple local planners. This hierarchical problem, although practically relevant, has barely been studied in the academic literature. Vogel and Meyr (2015) are the first to address the problem while assuming deterministic demand. In practice, simplistic rules of thumb are applied to determine sales quotas, which leads to suboptimal results (Vogel, 2014).

This paper investigates the hierarchical DF problem. Specifically, the study addresses the question of what information is required at the individual levels of the hierarchy to allow for an effective allocation. Mathematically, the optimal decision in each allocation step depends on the projected demand distributions of all individual customer segments. While technically feasible, sharing this fine-grained information across the levels of the decision-making hierarchy is undesirable from

a managerial perspective because it overloads higher-level decision makers with potentially insignificant details and makes the resulting allocation decisions difficult to communicate. Therefore, companies commonly aggregate the demand information propagated along the levels of the hierarchy. While aggregation simplifies the decision process, overly coarse information may result in ineffective allocation decisions. To strike the right balance, it is crucial to identify those pieces of information that yield the greatest benefits in terms of steering the consecutive allocation steps towards an overall optimum.

Our paper is meant to provide insight into this information-performance trade-off. In contrast to Vogel and Meyr (2015), we assume stochastic demand. Therefore, potentially relevant information about customer segments can be broadly divided into information on expected demand, demand uncertainty, and unit profits. We investigate the role of each of these dimensions in hierarchical allocation planning for demand fulfillment.

In summary, our paper makes the following contributions:

- We formalize the allocation planning problem in customer hierarchies by defining information aggregation and allocation functions;
- We characterize the optimal centralized solution to the stochastic allocation problem;
- We develop robust and near-optimal decentralized allocation methods for the hierarchical stochastic DF problem;
- We compare the numerical performance of the proposed methods with benchmarks commonly applied in APS and investigate the parameters driving the respective gaps;
- We reflect on the role of information sharing in hierarchical demand fulfillment and identify crucial information for good decentralized allocation decisions.

The paper proceeds as follows. In Section 3.2, we review the related literature and position our contribution. In Section 3.3, we formalize the hierarchical DF problem. In Section 3.4, we explain the best-case and worst-case benchmarks for the problem, including the optimal centralized solution. We present our new decentralized heuristics in Section 3.5 and evaluate their performance in extensive numerical experiments in Section 3.6. In Section 3.7, we provide our conclusions and managerial insights.

3.2 Literature Review

Demand fulfillment matches customer orders with available resources (Lin and Shaw, 1998; Stadtler and Kilger, 2008) and thereby provides an additional short-term lever to maximize performance for given supply and demand. Croxton (2003) provide an introduction to DF, including an analysis of its components, requirements, and goals.

The potential of DF to increase profitability has attracted a growing stream of research (Chen and Dong, 2014). The relevant literature can be subdivided by the type of production system considered. In this paper, we consider a make-to-stock (MTS) system; thus, inventory is the relevant resource for supply-and-demand matching. For DF in make-to-order (MTO) systems, we refer to Chiang and Wei-Di Wu (2011), and for assemble-to-order (ATO) systems, to Guhlich et al. (2015).

Quante et al. (2009b) further classify DF models based on demand management levers and the degree of supply flexibility. In the present paper, we assume exogenous prices and exogenous supply. DF then relies on segmenting the customer base and optimizing the supply quotas allocated to the individual customer segments.

Table 3.1 summarizes the relevant literature on DF in MTS systems. Single-period models consider a single replenishment cycle, analogous to traditional RM in service industries. Corresponding deterministic demand models essentially rank customer segments by unit profit (Jeong et al., 2002; Vogel, 2014). Stochastic demand models estimate opportunity costs to balance current sales revenues and future sales opportunities (Caldentey and Wein, 2006; Samii et al., 2012). Multiperiod models consider multiple exogenous replenishments simultaneously and thus are faced with a multicommodity allocation task. Inventory holding and backorder costs differentiate the profitability of different replenishments for fulfilling a given customer order. Deterministic models typically use LP to optimize these allocations (Ketikidis et al., 2006; Meyr, 2009; Jung, 2010; Alemany et al., 2013), whereas stochastic models commonly rely on stochastic dynamic programming (Quante et al., 2009a; Pibernik and Yadav, 2009; Tiemessen et al., 2013; Yang and Fleischmann, 2013; Gössinger and Kalkowski, 2015).

In this paper, we consider a single replenishment cycle and assume stochastic demand. A major distinction between our work and that discussed above is that we consider a multilevel hierarchical allocation, whereas all the aforementioned literature assumes a "flat" customer structure, i.e., a single allocation level. Vogel and

Table 3.1: Literature on demand fulfillment in MTS production systems.

	Deterministic	
	Single-period	**Multi-period**
Flat	Jeong et al. (2002) Vogel (2014)	Ketikidis et al. (2006) Meyr (2009) Jung (2010) Alemany et al. (2013)
Hierarchy	Vogel and Meyr (2015)	Cano-Belmán and Meyr (2019)
	Stochastic	
	Single-period	**Multi-period**
Flat	*traditional RM* Caldentey and Wein (2006) Samii et al. (2012)	Quante et al. (2009a) Pibernik and Yadav (2009) Tiemessen et al. (2013) Yang and Fleischmann (2013) Gössinger and Kalkowski (2015)
Hierarchy	Kloos et al. (2018) **this paper**	

Meyr (2015) are the first to investigate hierarchical DF. Assuming deterministic demand, their work devises an aggregate measure of customer heterogeneity, which enables the hierarchical problem to be decomposed into a sequence of single-stage continuous knapsack problems. Vogel and Meyr (2015) propose using Theil's index for this purpose, thereby approximating the cumulative revenue function by a Lorenz curve. Their approach results in a decentralized allocation rule with a nonlinear objective function. The authors show that their rule performs very well when demand is deterministic. Demand uncertainty, however, degrades the performance relative to the considered benchmarks. Cano-Belmán and Meyr (2019) extend Vogel and Meyr's result to a multiperiod setting.

The work presented in this paper intends to overcome the limitations of the aforementioned approaches by developing and analyzing new approaches to hierarchical DF that account for demand uncertainty *and* profit heterogeneity. In addition, we address the question of which information has to be shared to obtain effective decentralized allocation decisions.

To the best of our knowledge, the only other paper that analyzes hierarchical demand fulfillment under demand uncertainty is Kloos et al. (2018). Their setting differs from ours in that they consider customer segments that are differentiated

by different α-service-level targets and seek to determine allocations that minimize deviation from these targets.

Related hierarchical allocation processes have also been studied outside of the field of supply chain management, in particular, in the economics literature. Similar decentralized problems arise, e.g., in capital budgeting and in the regulation of public utilities. Van Zandt (1995) and Van Zandt (2003) consider information processing from an organizational theory perspective and explain the upward flow of information and the downward disaggregation of allocations in hierarchies. Van Zandt and Radner (2001) show the effects of decentralized information processing on returns to scale of organizations. Mookherjee (2006) provides a review of the costs and benefits of decentralized decision making in hierarchical organizations, focusing mainly on incentives and coordination. How information is aggregated is considered as given in the above literature, and different information aggregation alternatives are not compared. What distinguishes our research is that we explicitly consider information aggregation functions and evaluate different decentralization methods.

3.3 Problem Definition

Our research addresses the DF problem of a manufacturer operating a make-to-stock system and seeking to maximize expected profits by serving demand from hierarchically structured customer segments. We formalize this problem as follows.

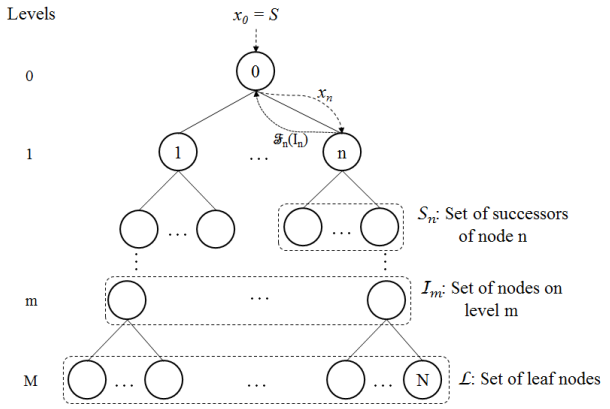


Figure 3.1: Hierarchical customer structure (adapted from Vogel and Meyr, 2015).

Let $\mathcal{N}$ denote the set of nodes in a customer hierarchy encompassing $M+1$ levels, as depicted in Figure 3.1. $\mathcal{I}_m$ denotes the set of all nodes on level $m \in 1, \ldots, M$. Specifically, $0 \in \mathcal{N}$ denotes the root node on Level 0, and $\mathcal{L} = \mathcal{I}_M$ denotes the set of leaf nodes, which represent the base customer segments, i.e., the most disaggregated type of customer segment considered. Moreover, for each node $n \in \mathcal{N}$, let $\mathcal{S}_n$ be the set of successor nodes. Given the hierarchical structure of the segments, each successor node has a unique parent node.

We consider a single replenishment cycle and assume that the available supply of inventory S is exogenous to the fulfillment problem and known. The supply quantity results from the company's production planning, which has a lead time longer than the accepted customer response time. Therefore, the supply quantity cannot be adjusted once demand materializes.

Demand is stochastic and materializes at the leaf node level, i.e., it originates from base customer segments. Let D_l denote the demand from customer segment $l \in \mathcal{L}$ with cdf F_l, pdf f_l, mean d_l, and standard deviation σ_l. Demands from different segments are mutually independent and independent of S. Unit profits are homogeneous within base customer segments but heterogeneous across segments. Let p_l denote the unit profit generated by serving the demand of customer segment l.

The available supply is allocated sequentially, level by level, top-down, from the root node to the base customer segments. That is, at each node n, a planner decides how to allocate the supply available at that node x_n to the respective successor nodes in $\mathcal{S}_n$. At the leaf node level, the amount of supply allocated to a given base customer segment is the quantity available for satisfying demand from that segment. Excess demand is lost. We do not consider nesting since it may require transshipments between different geographical regions and complicates communication in the decentralized allocation process by not providing firm commitments of quota availability.

To make the allocation decision, each planner uses demand information provided by the corresponding successor nodes. Let the information vector I_n describe the demand-related information available for the allocation decision at node n. The most detailed demand information is available on the leaf node level and concerns the demand distributions (f_l) and unit profits (p_l) of the base customer segments. This information is then transmitted in an aggregated fashion bottom-up across the hierarchy; that is, the planner at a given node aggregates the information available from all direct successors and transmits the information to the predecessor node.

The question of how to aggregate the relevant demand information and how to use the aggregated information in an effective allocation rule is at the heart of our research. To formalize this question, we introduce the concepts of an information aggregation function and an allocation function.

Definition 3.1 (Information aggregation function)**.** *The information aggregation function $\mathfrak{F}_k$ maps the information vector (I_k) of node $k \in \mathcal{S}_n$ to the information vector I_n of node n, such that $I_n = (\mathfrak{F}_k(I_k))_{k\in\mathcal{S}_n}$. Let $\mathfrak{F}$ denote the set of feasible aggregation functions.*

Definition 3.2 (Allocation function)**.** *The allocation function $\mathfrak{A}_n$ maps the supply x_n and information vector I_n available at node n to the allocations x_k of the successor nodes $k \in \mathcal{S}_n$, such that $(x_k)_{k\in\mathcal{S}_n} = \mathfrak{A}_n(x_n, I_n)$. Let $\mathfrak{A}$ denote the set of feasible allocation functions.*

The functions $\mathfrak{F}_n$ and $\mathfrak{A}_n$ describe the bottom-up aggregation of demand information and the top-down disaggregation of the available supply in the hierarchical fulfillment process. By means of these concepts, we can express the company's hierarchical demand fulfillment problem as follows.

Problem 3.1 (Decentralized hierarchical allocation problem)**.**

$$\underset{\substack{\mathfrak{F}_1,\dots,\mathfrak{F}_{|\mathcal{N}|}\in\mathfrak{F}\\ \mathfrak{A}_1,\dots,\mathfrak{A}_{|\mathcal{N}\setminus\mathcal{L}|}\in\mathfrak{A}}}{\text{maximize}} \quad \sum_{l\in\mathcal{L}} p_l \cdot \mathbb{E}[\min(x_l, D_l)] \tag{3.1}$$

s.t.

$$x_0 = S \tag{3.2}$$

$$x_n \geq 0 \qquad \forall n \in \mathcal{N} \tag{3.3}$$

$$x_n \geq \sum_{k\in\mathcal{S}_n} x_k \qquad \forall n \in \mathcal{N}\setminus\mathcal{L} \tag{3.4}$$

$$I_n = (\mathfrak{F}_k(I_k))_{k\in\mathcal{S}_n} \qquad \forall n \in \mathcal{N}\setminus\mathcal{L} \tag{3.5}$$

$$(x_k)_{k\in\mathcal{S}_n} = \mathfrak{A}_n(x_n, I_n) \qquad \forall n \in \mathcal{N}\setminus\mathcal{L} \tag{3.6}$$

The company seeks to maximize the total expected profit (3.1), which is equal to the sum of the expected profits generated at the leaf nodes. Constraint (3.4) ensures that the amount allocated to the successor nodes does not exceed the allocation to the respective parent node. Constraint (3.5) defines the information available to node n dependent on the information aggregation function, and Constraint (3.6) describes how allocations on level n are transformed into allocations on level k, given information vector I_n.

Problem 3.1 provides a formal description of the fulfillment problem outlined in Section 1. However, note that this formulation optimizes over two sets of interrelated functions; therefore, it does not easily lend itself to computational approaches. In addition, the formulation requires a specification of the feasible sets $\mathfrak{F}$ and $\mathfrak{A}$, which raises conceptual issues beyond what we deem meaningful for the purpose of our investigation. Therefore, we do not seek to solve Problem 3.1 but rather use it as a framework for a unified description of potential approaches. Specifically, we characterize several fulfillment approaches in terms of their underlying information aggregation and allocation functions and evaluate and compare their performance. We start by investigating two benchmark approaches in Section 3.4 and then present two new heuristics in Section 3.5.

3.4 Full and Minimum Information-Sharing Benchmarks

We seek to investigate the information-performance trade-off in hierarchical DF. To assess the effectiveness of our proposed methods, we introduce two benchmarks based on full and minimum information sharing. To this end, we investigate centralized allocation planning, which optimizes allocated quotas based on full demand information and per commit allocation, which is a simple heuristic requiring very limited information sharing. These methods represent upper and lower bounds for the degree of information aggregation in the customer hierarchy. Their relative performance provides insights into the dependence of effective DF on information availability. Moreover, they serve as benchmarks for heuristics that use some intermediate level of information aggregation.

3.4.1 Full Information: Centralized Allocation

Although transmitting full information on all base customer segments through the levels of the hierarchy is practically infeasible, this approach does provide an insightful benchmark. The corresponding information aggregation function $\mathfrak{F}_n^c$ is an identity function for all n. Thus, starting at the leaf nodes, the demand distributions and unit profits of all underlying customer segments are transmitted from any node to its respective parent node. In this case, the total available supply S can be directly allocated to the leaf nodes. Allocations to intermediate nodes do

not matter. If desired, they can be determined by simply summing the allocations to the respective successor nodes. Thus, full information transmission results in a single-level allocation planning problem, which we denote as centralized allocation. For this case, Problem 3.1 reduces to the following.

Problem 3.2 (Centralized allocation)**.**

$$\underset{(x_l)_{l\in\mathcal{L}}}{\text{maximize}} \quad P = \sum_{l\in\mathcal{L}} p_l \cdot \mathbb{E}[\min(x_l, D_l)] \tag{3.7}$$

s.t.

$$\sum_{l\in\mathcal{L}} x_l \leq S \tag{3.8}$$

$$x_l \geq 0, \qquad \forall l \in \mathcal{L} \tag{3.9}$$

This is a nonlinear continuous knapsack problem. We can easily characterize its solution using known results from the literature. The proofs of Lemma 3.1 and Proposition 3.1 are given in Appendix B.1.

Lemma 3.1. *The objective function (3.7) is concave and increasing in x_l.*

Proposition 3.1 (Optimal allocation)**.** *There exists a constant $\gamma \geq 0$, such that the following set of equations yields an optimal solution to Problem 3.2.*

$$x_l = \begin{cases} 0 & if \quad \overline{S_l} > S \\ {F_l}^{-1}(1 - \frac{\gamma}{p_l}) & if \quad \overline{S_l} \leq S \end{cases} \qquad \forall l \in \mathcal{L}$$

$$\sum_{l\in\mathcal{L}} x_l = S$$

where $\overline{S_l}$ is defined by:

$$\overline{S_l} = \sum_{\{i\in\mathcal{L} \mid p_l(1-F_l(0)) \leq p_i(1-F_i(0))\}} {F_i}^{-1}\left(1 - \frac{p_l(1-F_l(0))}{p_i}\right) \tag{3.10}$$

If $F_l(\cdot)$ is strictly increasing for all l, the solution is unique.

Proposition 3.1 shows that for each node l, there is a supply threshold value beyond which that node receives a nonzero quota under optimal centralized allocation and that expected profits for all nodes receiving a nonzero quota are balanced.

These properties implicitly define the allocation functions $\mathfrak{A}_n^c$ for the centralized allocation approach. Because this approach maximizes the expected profit under full information transmission, it provides an upper bound on the expected profits that can be achieved under aggregated information.

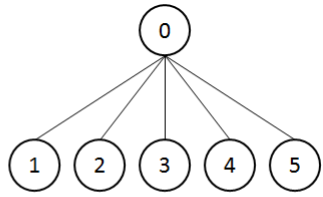
We conclude this subsection by illustrating the relationship between the maximum expected profit and available supply. This perspective is instructive because the decentralized methods introduced in Section 3.5 can be associated with different ways of approximating this profit curve. We denote by $P_k(S)$ the maximum objective value of Problem 3.2 dependent on the available supply S, with $\mathcal{L}$ restricted to the set of leaf nodes in the subtree below node k.

Consider the customer hierarchy consisting of six nodes on two levels displayed in Figure 3.2(a). The leaf nodes have identical demand distributions but differ in their unit profits. We define the supply rate as the available supply quantity, scaled by total expected demand. Figure 3.2(b) then shows the quantities allocated to the five base customer segments by the centralized approach as a function of the supply rate. The allocation curves reflect the aforementioned properties of $\mathfrak{F}_n^c$. In particular, we observe the threshold supply values at which we start supplying another node. The solid line in Figure 3.2(c) shows the corresponding expected profits, i.e., $P_0(S)$. The curve is piecewise nonlinear, concave and increasing, with breakpoints at the aforementioned supply threshold levels. The two remaining curves in Figure 3.2(c) reflect the per commit allocation method, which we introduce in the next subsection.

3.4.2 Minimum Information: Per Commit Allocation

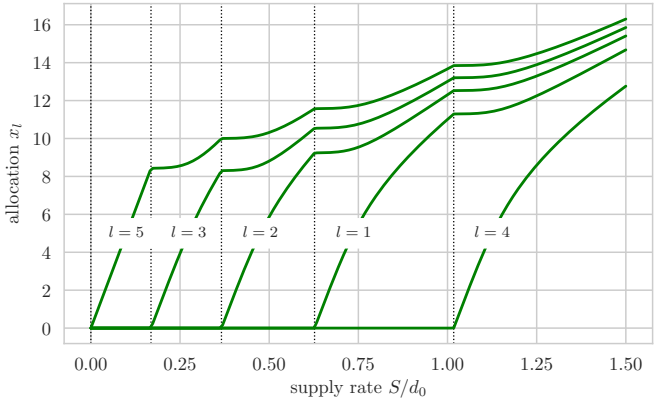
Per commit allocation is a decentralized allocation method commonly used in DF modules of APS (cf. Kilger and Meyr, 2015). This method allocates scarce supply to the successor nodes proportional to their expected demand, which is the only information transmitted across the levels of the hierarchy. Information on demand uncertainty and unit profits is disregarded. We formally define this method in terms of the previously introduced aggregation and allocation functions.

Definition 3.3 (Per Commit)**.** *Per commit allocation uses the information aggregation functions $\mathfrak{F}_n^{pc}$ and allocation functions $\mathfrak{A}_n^{pc}$, defined as*

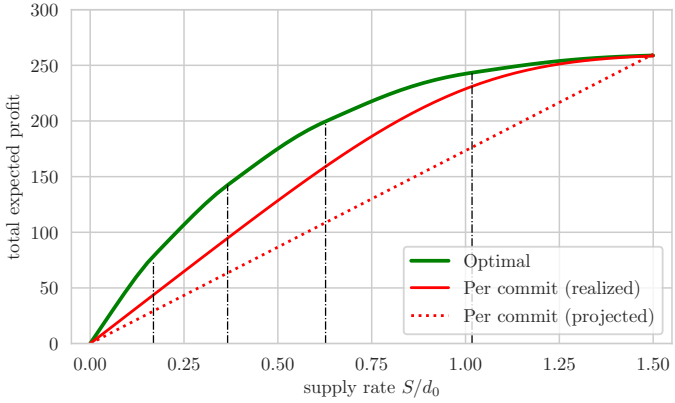


Unit profit (p_n)	p_1=3	p_2=5	p_3=7	p_4=1	p_5=10
Demand distribution (f_n)	N(10,3)	N(10,3)	N(10,3)	N(10,3)	N(10,3)

(a) Customer configuration



(b) Optimal allocations



(c) Total expected profit

Figure 3.2: Illustration of the optimal allocation and profit function approximation.

$$\mathfrak{F}_n^{pc}(I_n) = \begin{cases} \mathfrak{F}_n^{pc}((F_k, p_k)_{k\in\mathcal{S}_n}) = \sum_{k\in\mathcal{S}_n} d_j = d_n & for \quad n \in I_{M-1} \\ \mathfrak{F}_n^{pc}((d_k)_{k\in\mathcal{S}_n}) = \sum_{k\in\mathcal{S}_n} d_j = d_n & for \quad n \in I_m, \ m < M-1 \end{cases}$$

$$\mathfrak{A}_n^{pc}(x_n, I_n) = \quad \mathfrak{A}_n^{pc}(x_n, (d_k)_{k\in\mathcal{S}_n}) = \left(\frac{d_k}{\sum_{k\in\mathcal{S}_n} d_k} \cdot x_n\right)_{k\in\mathcal{S}_n}$$

Per commit ignores demand uncertainty and unit profit heterogeneity and can thus be interpreted as assuming deterministic demand from homogeneous customers. The corresponding assumed profit curve is a simple linear line, as shown in Figure 3.2(c). The figure also displays the actual expected profits of a per commit allocation under heterogeneous stochastic demand. The fact that the per commit method is based on a simplistic profit approximation results in a performance gap relative to the optimal centralized allocation. This gap defines the improvement potential of the smarter decentralized allocation heuristics presented in the next section. In our example, the maximum absolute profit gap is given at a supply rate of 36.7 percent and amounts to 33.4 percent. Not surprisingly, the absolute profit gap diminishes for high supply rates. If supply is not scarce, the allocation problem disappears. Note, however, that even for a supply rate of 100 percent, a per commit allocation still results in a profit gap of 5.3 percent.

3.5 Decentralized Allocation Heuristics

In the previous section, we have seen that the popular yet simplistic per commit allocation method may yield poor performance for relevant supply rates. In this section, we propose two novel allocation heuristics that aim to overcome this deficit while respecting the decentralized and iterative nature of the allocation process. The first method, presented in Subsection 3.5.1, uses the concept of a heterogeneity index; the second method, presented in Subsection 3.5.2, relies on clustering. Unlike per commit, both of these methods transmit and use information on profit heterogeneity and demand uncertainty, albeit in an aggregated manner. Specifically, both methods approximate the piecewise nonlinear profit curve of the centralized problem (see Figure 3.2(c)) and then use an optimal allocation given that approximation.

3.5.1 Stochastic Theil Index Method

In a deterministic setting, Vogel and Meyr (2015) introduce the idea of transmitting information on unit profit heterogeneity by means of a heterogeneity index. Specifically, they use Theil's index, established in the economics literature for approximating a single-parameter Lorenz curve by Chotikapanich (1993). In the deterministic hierarchical DF problem, the profit function at each intermediate node k is piecewise linear and concave. Theil's index approximates this function by means of a smooth nonlinear flipped Lorenz curve. Formally, Theil's index at node k is calculated recursively as

$$T_k = \sum_{j \in \mathcal{S}_k} \frac{d_j}{d_k} \cdot \frac{p_j}{p_k} \cdot T_j + \sum_{j \in \mathcal{S}_k} \frac{d_j}{d_k} \cdot \frac{p_j}{p_k} \cdot \ln\left(\frac{p_j}{p_k}\right), \quad with \;\; T_j = 0 \;\; \forall j \in \mathcal{L}. \tag{3.11}$$

The Theil index (T_k) implies a Lorenz curve parameter (θ_k) through

$$\ln\left(\frac{\theta_k}{(e^{\theta_k} - 1)}\right) + \frac{\theta_k}{(e^{\theta_k} - 1)} + \theta_k - 1 - T_k = 0. \tag{3.12}$$

The resulting Lorenz curve approximation of the profit function at node k is then

$$\pi_k(x_k, \theta_k) = \frac{e^{\theta_k \cdot \frac{x_k}{d_k}} - 1}{e^{\theta_k} - 1} \cdot d_k \cdot p_k. \tag{3.13}$$

Vogel and Meyr (2015) use these concepts to define a decentralized allocation method that transmits aggregated mean demand (d_k), weighted average unit profit (p_k) and Theil's index (T_k) along the hierarchy. Given these inputs, they determine the optimal allocation under the assumption of profit functions, as in (3.13).

Vogel and Meyr (2015) show that this method performs very well for deterministic demand but degrades for stochastic demand. To understand this observation, consider the deterministic versus stochastic profit curves in Figure 3.3 for the same example as in Figure 3.2. While the Lorenz curve approximates the piecewise linear deterministic profit curve, it systematically deviates from the piecewise nonlinear stochastic profit curve and therefore may result in an inefficient allocation in the latter case.

We build on this observation and construct a Theil index-based approximation of the stochastic profit curve. Note that all stochasticity arises at the leaf nodes. This suggests that if we capture the effects of uncertainty appropriately at the leaf

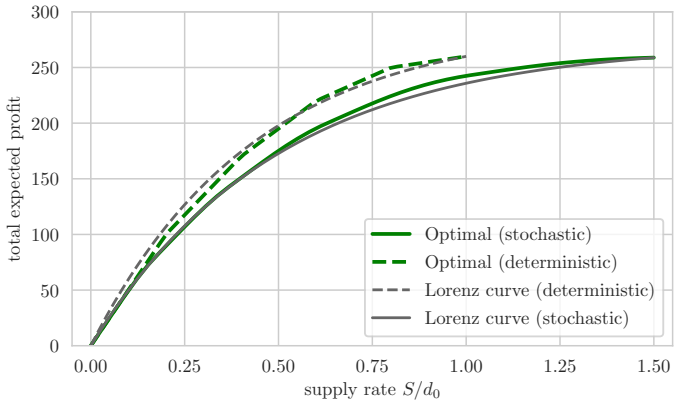


Figure 3.3: Illustration of Lorenz curve approximation.

node level, we can proceed as in the deterministic problem at the higher levels of the hierarchy. To implement this idea, we approximate the expected profit curve of any leaf node l by a piecewise linear function. We then apply the method in Vogel and Meyr (2015) to approximate these piecewise linear functions by Lorenz curves and to propagate the corresponding parameters upwards in the hierarchy. Different methods can be used to create the initial piecewise linear functions. We asses several options in our numerical study in Section 3.6. Figure 3.3 shows the resulting Lorenz curve in our example when using three equidistant points on the expected profit curves of the leaf nodes.

We conclude this section by defining the stochastic Theil method in terms of its aggregation and allocation functions.

Definition 3.4 (Stochastic Theil method)**.** *The stochastic Theil method uses the information aggregation functions $\mathfrak{F}_n^{Th}$ and allocation functions $\mathfrak{A}_n^{Th}$, defined as follows.*

Assume that for any node k the available information vector is $(d_j, p_j, T_j)_{j\in\mathcal{S}_k}$. Then, the stochastic Theil method aggregates this information as follows:

$$\mathfrak{F}_k{}^{Th}((d_j,p_j,T_j)_{j\in\mathcal{S}_k}) = \left(\sum_{j\in\mathcal{S}_k} d_j, \frac{\sum_{j\in\mathcal{S}_k} d_j\cdot p_j}{\sum_{j\in\mathcal{S}_k} d_j}, \sum_{j\in\mathcal{S}_k}\frac{d_j}{d_k}\cdot\frac{p_j}{p_k}\cdot T_j + \sum_{j\in\mathcal{S}_k}\frac{d_j}{d_k}\cdot\frac{p_j}{p_k}\cdot\ln\left(\frac{p_j}{p_k}\right)\right) =: (d_k,p_k,T_k)$$

For nodes $k \in I_{M-1}$, we initialize the aggregation procedure by setting $T_j = 0$ for all j and defining (d_j, p_j) through a piecewise linear approximation of the expected profit function $P_k(S)$. To this end, we replace $\mathcal{S}_k$ in the above definition of the aggregation function $\mathfrak{F}_n^{Th}$ with $\tilde{\mathcal{S}}_k := \{0, ..., \overline{n}\}$, where $\overline{n}$ is the number of line segments of the piecewise linear approximation. Letting $x_0, ..., x_{\overline{n}}$ denote the intersection points of the approximation function with the expected profit curve, we set $d_j = x_j - x_{j-1}$ and $p_j = (P_k(x_j) - P_k(x_{j-1}))/d_j$ for $j = 1, ..., \overline{n}$.

Given these information vectors, the allocation functions $\mathfrak{A}_n^{Th}$ are defined implicitly through the solution of the following nonlinear optimization problem, using π_k defined in (3.13):

$$
\begin{aligned}
\underset{(x_k)_{k\in\mathcal{S}_n}}{\text{maximize}} \quad & \sum_{k\in\mathcal{S}_n} \pi_k(x_k, \theta_k) \\
\textbf{s.t.} \quad & \\
& \sum_{k\in\mathcal{S}_n} x_k \leq x_n \\
& x_k \geq 0, \qquad\qquad \forall k \in \mathcal{S}_n
\end{aligned}
$$

Since the planners at level $M-1$ have detailed information about the leaf nodes, allocations to the leaf nodes are determined by solving Problem 3.2.

Note that this method considers stochasticity explicitly only in the information aggregation function on level $M-1$, namely, through the piecewise linearization of the expected profit functions on that level. For levels higher in the hierarchy, the method is identical to Vogel and Meyr's original approach. However, the resulting parameter values and allocations are different because they depend on the values propagated upwards from level $M-1$ (comp. Figure 3.3).

Furthermore, note that the definition does not specify how to choose the piecewise linear approximation of the expected profit curve on level M_1, i.e., the number of line segments $\overline{n}$ and the corresponding break points. We asses different alternatives for setting these parameters in our numerical study in Section 3.6.

3.5.2 Clustering

Clustering is a very general approach for aggregating information that is commonly applied, e.g., in market segmentation (Sarstedt and Mooi, 2019). Closer to our context, Zipkin (1980a) proposes a clustering method for solving large linear

optimization problems. Instead of the large original problem, the method solves a smaller aggregated problem based on clustered variables and then disaggregates the outcome over the original variables.

In our hierarchical DF problem, we apply clustering by grouping the successor nodes of a given node into C clusters and by transmitting aggregated information about each cluster to the next higher level in the hierarchy. In this case, the information aggregation function ${\mathfrak{F}_k}^{cl}$ is thus a clustering function that receives the unit profits (p_k) and demand distributions (F_k) of the successor nodes as input and returns the aggregated unit profits and aggregated demand distribution parameters of C clusters.

To define a clustering heuristic for our hierarchical DF problem, we need to specify the clustering attributes, the number of clusters, and the evaluation metric.

The clustering attributes define which data determine whether two customer segments will be regarded as similar, and thus potentially clustered together, or as different. In Section 3.4.1, we saw that in the full-information benchmark, customer segments enter the solution in the order of their unit profits; therefore, we use unit profits as our clustering attribute. In this way, we intend to preserve relevant information on profit heterogeneity in the aggregation process.

We treat the number of clusters C as an input parameter. Its choice is linked to a trade-off between complexity and performance. Clustering with $C = |\mathcal{L}|$ results in the full-information case. Decreasing the number of clusters reduces the complexity but conveys a less fine-grained image of customer heterogeneity, thereby potentially resulting in inferior allocation decisions. For the special case of $C = 1$, unit profits are aggregated into a single parameter. Thus, information on profit heterogeneity will be lost, while the aggregated demand distribution of the successor nodes will be transmitted. We assess the impact of different values of C in our numerical study in Section 3.6.

The general goal of clustering is to create clusters that are homogeneous within but heterogeneous between each other. Different clustering approaches use different metrics to operationalize this goal. Many criteria rely on some type of distance measure. The popular K-means clustering approach minimizes the sum of the distances between the objects in each cluster and the empirical cluster centers (Jain, 2010). We adopt the K-means approach to define the aggregation function for our clustering heuristic.

Definition 3.5 (Clustering method)**.** *The clustering method for hierarchical demand fulfillment uses the information aggregation functions $\mathfrak{F}_n^{cl}$ and allocation functions $\mathfrak{A}_n^{cl}$, which are defined as follows.*

The information vector I_n available at node n is $((d_{kj}, \sigma_{kj}, p_{kj}))_{k\in\mathcal{S}_n, j=1,\dots,C}$, where vector $(d_{kj}, \sigma_{kj}, p_{kj})$ denotes the aggregated mean and standard deviation of demand and the aggregated unit profit of customer cluster j of successor node k and C is an exogenous parameter. We use the same number of clusters on all levels of the hierarchy, except for the leaf node level, where we set $C = 1$. The information aggregation function further aggregates the available information as follows.

$$
\begin{aligned}
&\mathfrak{F}_n{}^{cl}((d_{kj}, \sigma_{kj}, p_{kj}))_{k\in\mathcal{S}_n, j=1,\dots,C} \\
&\quad = \left(\sum_{\substack{k\in\mathcal{S}_n \\ j=1,\dots,C}} v_{ckj} \cdot d_{kj}, \sum_{\substack{k\in\mathcal{S}_n \\ j=1,\dots,C}} v_{ckj} \cdot \sigma_{kj}, \sum_{\substack{k\in\mathcal{S}_n \\ j=1,\dots,C}} v_{ckj} \cdot \frac{d_{kj} \cdot p_{kj}}{d_{cn}} \right)_{c=1,\dots,C} \\
&\quad =: (d_{cn}, \sigma_{cn}, p_{cn})_{c=1,\dots,C},
\end{aligned}
$$

where $v_{ckj} = 1$ when cluster j of node k belongs to cluster c of node n and is zero otherwise.

Given the information vector $I_n = ((d_{kj}, \sigma_{kj}, p_{kj}))_{k\in\mathcal{S}_n, j=1,\dots,C}$ and available supply x_n at node n, the allocation function $\mathfrak{A}_n^{cl}$ allocates a quantity $\sum_{c=1}^{C} x_{ck}$ to node successor $k \in \mathcal{S}_n$, where x_{ck} solves

$$
\begin{aligned}
\underset{(x_{ck})_{k\in\mathcal{S}_n, c\in\{1,\dots,C\}}}{\text{maximize}} \quad & \sum_{\substack{k\in\mathcal{S}_n \\ c=1,\dots,C}} p_{ck} \cdot \mathbb{E}[\min(x_{ck}, D_{ck})] \\
\textbf{\textit{s.t.}} \quad & \\
& \sum_{\substack{k\in\mathcal{S}_n \\ c=1,\dots,C}} x_{ck} \le x_n \\
& x_{ck} \ge 0, \qquad \forall k \in \mathcal{S}_n, c \in \{1, \dots, C\}
\end{aligned}
$$

and D_{ck} is a random variable with mean d_{ck} and standard deviation σ_{ck}.

A few comments are in order. First, the allocation function $\mathfrak{A}_n^{cl}$ solves Problem 3.2 to determine the allocations to the clusters. Each successor node k then receives the sum of the amounts allocated to its underlying clusters. Second, in the definition of $\mathfrak{F}_n^{cl}$, we aggregate demand uncertainty within a cluster by summing the standard deviations of the underlying lower-level clusters because of the

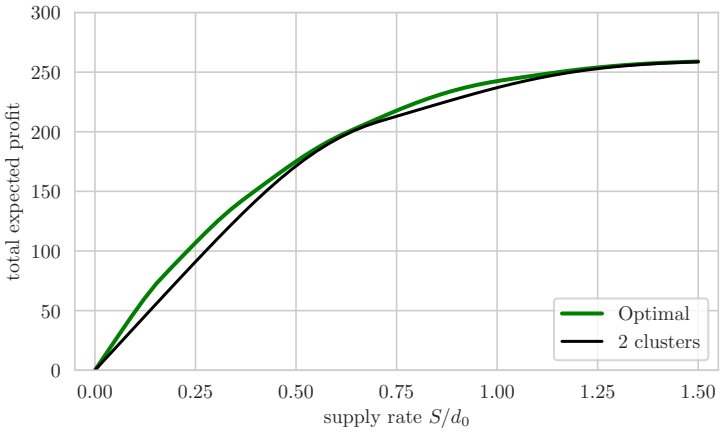


Figure 3.4: Illustration of the clustering approximation.

partitioned consumption of supply, which prevents risk pooling effects. Third, the definition of $\mathfrak{A}_n^{cl}$ specifies only the first two moments of the probability distribution of the cluster demands D_{ck}. In our numerical study in Section 6, we assume normal distributions.

By relying on Problem 3.2, the cluster allocation function is optimal if the cluster information is exact. In general, clustering provides a piecewise nonlinear approximation of the centralized profit function, and the number of clusters determines the number of pieces. Figure 3.4 displays the approximated profit function using 2 clusters for the example introduced in Section 3.4.

We conclude this section by summarizing the decentralized allocation methods introduced in Sections 3.4 and 3.5. Table 3.2 indicates the information transmitted across the hierarchy by each of these methods. As discussed, per commit allocation represents the minimum information benchmark in that it uses only expected demand information. The deterministic Theil approximation of Vogel and Meyr (2015) complements this information with information on profit heterogeneity, but it ignores demand uncertainty. Conversely, clustering with $C = 1$ ignores profit heterogeneity but captures demand uncertainty. Both our modified stochastic Theil approximation and clustering with $C \geq 2$ transmit and use information about all three attributes of the customer segments, albeit in an aggregated manner. In the following section, we assess and compare the performance of the various methods

Table 3.2: Information shared in decentralized allocation methods.

Method	Demand uncertainty		Profit		Parameters per node
	Homog.	Heterog.	Homog.	Heterog.	
Per commit, *§3.4.2*	–	–	–	–	1
Deterministic Theil, *§3.5.1*	–	–	–	✓	3
Stochastic Theil, *§3.5.1*	–	✓	–	✓	3
Clustering ($C = 1$), *§3.5.2*	✓	–	✓	–	3
Clustering ($C \geq 2$), *§3.5.2*	–	✓	–	✓	$3 \cdot C$

and relate the performance differences to the information shared and used by the various methods, as described in Table 3.2.

3.6 Numerical Analysis

In this section, we present the results of an extensive numerical study conducted to evaluate the performance of the decentralized allocation heuristics proposed in Section 3.5 in comparison to the full-information benchmark (central allocation) and the minimum-information benchmark (per commit) from Section 3.4. Beyond mere performance comparisons, we also want to shed light on the role of information sharing, as discussed in the previous section. We want to provide a conclusive answer to the question of which information depicted in 3.2 should be shared and utilized to ensure effective allocation planning.

In Section 6.1, we first describe our experimental setup and how we evaluated the performance of the different allocation methods. Subsequently, in Section 3.6.2, we explain how we implemented and parameterized both the stochastic Theil method and the clustering method in our experiments. In Section 3.6.3, we report, compare, and discuss the performance of the four different allocation methods for a baseline scenario. We provide an extensive evaluation and discussion of the performance differences across the different allocation methods and derive insights into the role of information sharing. In Section 3.6.4, we assess the robustness of our results by extending our analysis to other scenarios, including different customer hierarchies and different input parameters.

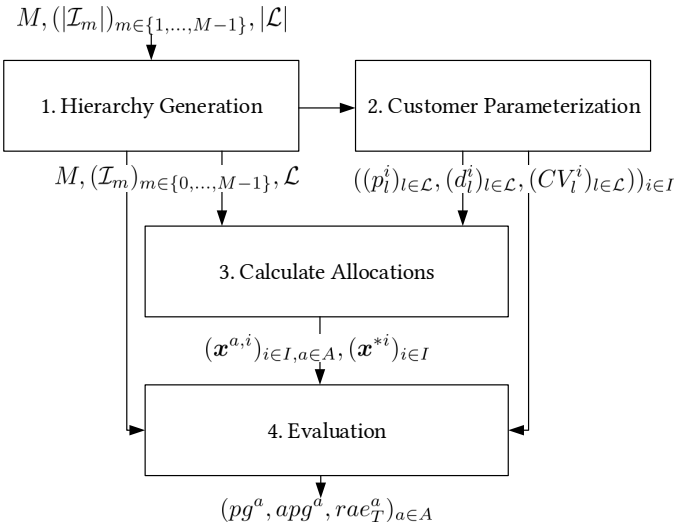


Figure 3.5: Overview of the simulation procedure.

3.6.1 Experimental Setup

In this section, we explain the simulation procedure and the data used to evaluate the decentralized allocation methods from Section 3.5 and to compare their performance with the full-information and minimum-information benchmarks from Section 3.4.

Our simulation procedure follows the four-step process depicted in Figure 3.5. In the first step, we generate the hierarchy for a specific set of experiments. We restrict our experiments to symmetric hierarchies. Therefore, a hierarchy is fully defined by the number of nodes on each level. Figure 3.6 illustrates the hierarchy of the baseline scenario with $M = 4$ levels and $|\mathcal{I}_2| = 2, |\mathcal{I}_3| = 6$ and $|\mathcal{L}| = 30$.

In the second step, we assign unit profits (p_l), mean demand (d_l) and coefficients of variation of demand (CV_l) to the leaf nodes. For the baseline scenario, we draw $|I| = 100$ realizations of p_l from a uniform distribution with support $[1, 10]$ for each leaf node $l \in \mathcal{L}$ and set the mean demand to 10 and the CV to 0.2 for all leaf nodes. This process provides 100 instances $((p_l^i)_{l\in\mathcal{L}}, (d_l^i)_{l\in\mathcal{L}}, (CV_l^i)_{l\in\mathcal{L}})_{i\in I}$. In the additional experiments of our robustness analysis, we vary the support of

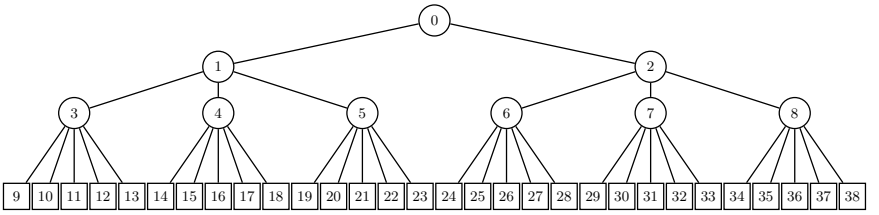


Figure 3.6: Hierarchy in the baseline scenario.

Table 3.3: Hierarchy Parametrization.

Parameter	Baseline	Variations
$\lvert\mathcal{I}_2\rvert$	2	
$\lvert\mathcal{I}_3\rvert$	6	-
$\lvert\mathcal{I}_4\rvert$	-	12
Number of Customers $\lvert\mathcal{L}\rvert$	30	18, 60
Number of Levels M	4	3 (18 customers), 5 (60 customers)
Profits p_l	$U[1,10]$	$U[1,5], U[1,20]$
Coefficient of Variation CV_l	0.2	0.1, 0.3, 0.4, 0.5, $U[0.1,0.5]$
Mean demand d_l	10	$U[5,15]$

p_l, the mean demand d_l and the CV of the different leaf nodes (see Table 3.3 for details).

The specification of the hierarchy and the instances generated in Step 2 constitute the input to the third step of our procedure. For each instance $i \in I$, we compute the optimal allocations $\boldsymbol{x}^{*i}$ of the centralized full-information benchmark and the allocations $\boldsymbol{x}^{a,i}$ of the allocation methods $a \in A = \{$per commit, deterministic Theil, stochastic Theil, clustering$\}$. We vary the supply levels x_0 in 50 equal steps from $0.5 \cdot d_0$ to $1.5 \cdot d_0$, where $d_0 = \sum_{l \in \mathcal{L}} d_l$ is the expected total demand.

In Step 4 of our procedure, we evaluate the performance of the different allocation methods. For this purpose, we use three performance measures, the relative profit gap (*rpg*), the average relative profit gap (*arpg*) and the relative allocation error (*rae*), defined as follows.

Definition 3.6 (Relative profit gap)**.** *The relative profit gap (rpg) of allocation method a with allocation $\boldsymbol{x}_i^a$ for a supply of x_0 is*

$$rpg_a(x_0) = \frac{1}{|I|} \sum_{i \in I} \left(1 - \frac{P(\boldsymbol{x}_i^a(x_0))}{P(\boldsymbol{x}^*(x_0))}\right)$$

Definition 3.7 (Average relative profit gap)**.** *The average relative profit gap (arpg) of method a evaluated for supply interval $\overline{S}$ is*

$$arpg_{\overline{S}} = \frac{1}{|I|} \sum_{i \in I} \left(1 - \frac{\sum_{x_0 \in \overline{S}} P_i(\boldsymbol{x}_i^a(x_0))}{\sum_{x_0 \in \overline{S}} P_i(\boldsymbol{x}_i^*(x_0))} \right)$$

Definition 3.8 (Relative allocation error)**.** *The relative allocation error (rae) of method a for a supply of x_0 is*

$$rae_T(x_0) = \frac{1}{|I|} \sum_{i \in I} \frac{\sum_{l \in T} (x_{i,l}^a(x_0) - x_{i,l}^*(x_0))}{x_0}$$

where $T = T_h, T_a, T_l \subset \mathcal{L}$ is the set of customers belonging to the tercile with high, average and low profits, respectively.

3.6.2 Implementation and Parametrization of the Allocation Approaches

In this section, we explain how we implemented and parametrized the stochastic Theil method and the clustering method introduced in Section 3.5.

To determine the Theil index for node $k \in \mathcal{I}_{M-1}$, we require $\overline{n}$ points on the expected profit curve (cf. Definition 3.4). In our implementation, we choose $\overline{n}$ equidistant points between 0 and $loc \cdot d_k$, where d_k is the expected demand at successor node k and loc is an exogenous input parameter. Hence, the points are given by $x_{kr} = r\frac{loc \cdot d_k}{\overline{n}}$ for $r \in \{0, \ldots, \overline{n}\}$ and $k \in \mathcal{I}_{M-1}$. Thus, we have to specify the input parameters loc and $\overline{n}$.

To this end, we performed various pretests and found that the number of points $\overline{n}$ has a negligible effect on performance for $\overline{n} > 2$ but that the optimal choice of the location parameter loc is affected by the supply rate. However, within a certain range of loc, the performance differences remain very small (cf. Figure B.1 in Appendix B.2). On the basis of these observations, we set $\overline{n} = 3$ and $loc = 1.5$ throughout our numerical experiments.

When implementing the clustering method, we use the K-means algorithm as implemented in SCIPY to determine the clusters. As discussed in Section 3.5.2, the performance of the clustering method depends on the number of clusters C. More clusters capture customer heterogeneity in greater detail and thus should enable a more effective allocation.

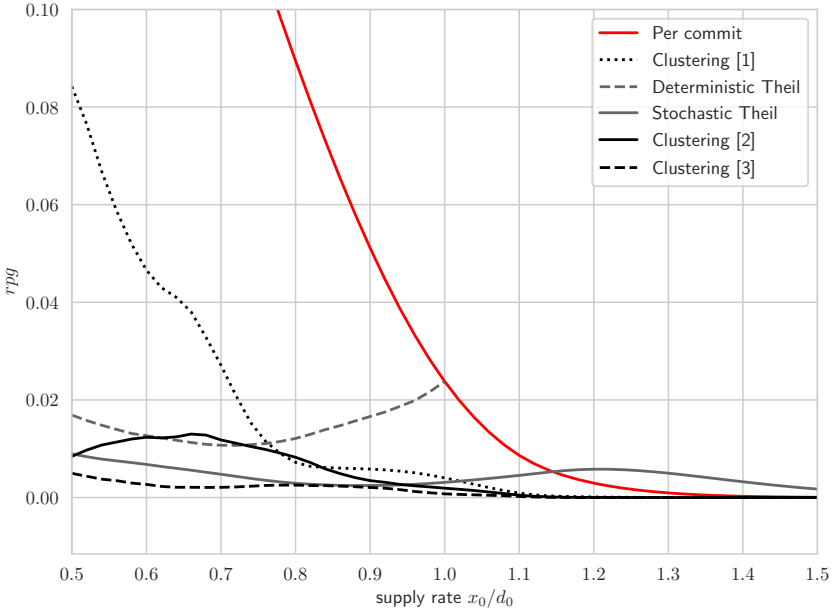


Figure 3.7: rpg of the allocation rules depending on the supply rate.

We again performed some pretests to assess the impact of C in our context. Specifically, we ran the baseline experiment for different numbers of clusters and found that the improvement from $C = 3$ to $C = 4$ is minimal (cf. Figure B.2 in Appendix B.2). Therefore, we consider the clustering method with one, two and three clusters in the remainder of our numerical analysis.

3.6.3 Results for the Baseline Scenario

In this section, we evaluate the performance of the different allocation methods for the baseline scenario. We structure our discussion according to Table 3.2. In Figure 3.7, we plot the relative performance (measured by *rpg*) of the considered allocation methods, i.e., the per commit method, the deterministic and stochastic Theil methods, and the clustering methods, at different levels of supply. To help explain the observed performance gaps, we also consider the *rae* of the different methods. Figure 3.8 depicts the *rae* of the considered allocation methods for differ-

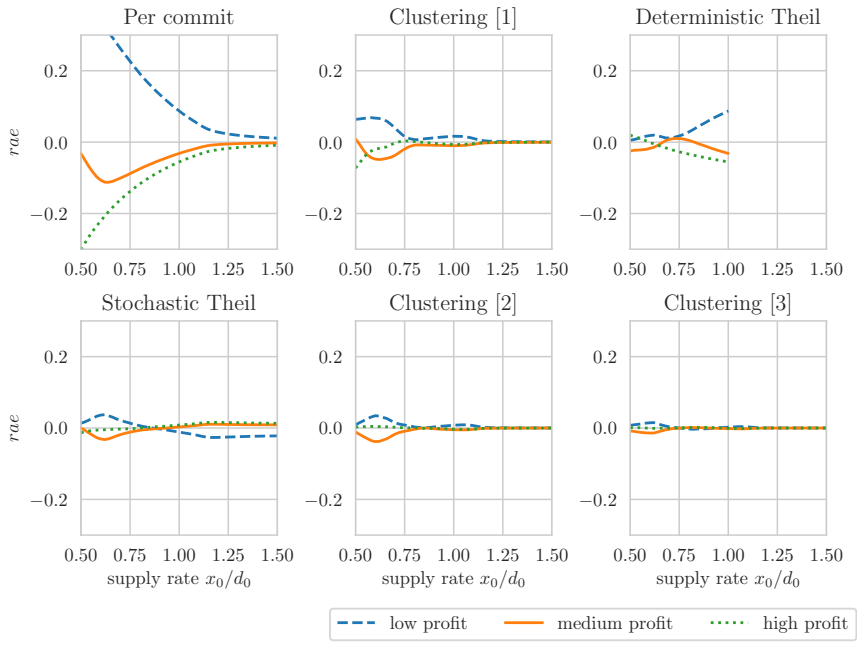


Figure 3.8: rae of the allocation methods depending on the supply rate.

ent customer groups and varying supply levels. Recall that *rae* captures deviations from the optimal quantities allocated to different customer groups. Therefore, any deviation from $rae = 0$ in Figure 3.8 can be interpreted as a "misallocation." In the following, we discuss the observed performance of each allocation method.

Per commit, our minimum-information sharing benchmark, leads to a substantial performance gap when supply is scarce (*rpg*=8.9% for a supply rate of 0.8), which decreases with increasing supply availability (*rpg*=2.4% (0.3%) for a supply rate of 1.0 (1.2)). This behavior is intuitive since per commit transmits and uses only the aggregated mean demands of the customer segments and ignores (the heterogeneity of) unit profits and demand uncertainty. By not prioritizing customers based on profitability, per commit consistently overserves low-profit customers while underserving high-profit customers (see Figure 3.8). This misallocation disappears only once supply is sufficient to essentially serve all demand.

The *deterministic Theil method* explicitly shares and uses information on profit heterogeneity but not on demand uncertainty. Because of its deterministic nature,

the method does not allocate more than the respective mean demand to each customer segment. Therefore, we report the *rpg* only for supply rates up to 1.0. At a supply rate of 1.0, the deterministic Theil method coincides with the per commit approach. For scarce supply, however, the deterministic Theil method clearly outperforms per commit (*rpg* consistently below 2%). This result reflects the benefit of sharing and using information on profit heterogeneity in the allocation process, albeit in a deterministic fashion.

The *clustering method* with a *single cluster* ("clustering [1]") shares and uses aggregated (i.e., homogeneous) information on profitability and demand uncertainty at each node of the customer hierarchy but ignores heterogeneity between customer segments within a node. From Figure 3.7, we observe that clustering [1] performs reasonably well as long as supply is not highly constrained ($rpg \leq 1\%$ for supply rates ≥ 0.78). Under high scarcity, however, the performance rapidly degrades. As highlighted in Figure 3.8, this method again insufficiently prioritizes high-profit customers under these circumstances. This result reflects the strong information aggregation within the nodes. Yet, the fact that clustering [1] substantially outperforms the per commit approach proves even this highly aggregated information to be valuable.

It is instructive to compare the performance of clustering [1] to that of the deterministic Theil method. Both approaches use complementary information in the sense that the deterministic Theil method captures profit heterogeneity within a node but ignores demand uncertainty, whereas clustering [1] acknowledges demand uncertainty but assumes homogeneous profits within a node (see Table 3.2). In our results, the former (latter) approach is superior for supply rates below (above) 0.76, which suggests that for allocations under low supply rates, information on profit heterogeneity is crucial, whereas demand uncertainty becomes more important in the allocation decision for higher supply rates. A potential explanation is that for highly scarce supply, it is optimal to strongly prioritize the most profitable customer segments. This prioritization requires information on profit differences between customer segments. At the same time, when supply is low, demand uncertainty is less of an issue since supply, rather than demand, is the constraining factor. For higher supply rates, it becomes optimal to allocate quantities larger than expected demand to high-profit customers. This requires an allocation approach that uses stochastic demand information.

The *stochastic Theil method* shares and uses information on both profit heterogeneity and (the heterogeneity of) demand uncertainty. By doing so, the stochastic

Theil method significantly outperforms all the previously discussed methods. The corresponding *rpg* is consistently below 1 percent (see Figure 3.7). In particular, comparison of the deterministic and stochastic variants of the Theil method (see Figures 3.7 and 3.8) illustrates the benefit of basing the method on expected profit curves at the leaf node level rather than on expected demand. Thus, this process shows how to make the idea of Vogel and Meyr (2015) of a decentralized allocation method available for stochastic demand.

The *clustering methods* with *more than one cluster* also use information on both profit heterogeneity and demand uncertainty, but in a different way. While under the stochastic Theil method, the Theil index captures the effects of both profit heterogeneity and demand stochasticity, clustering [2] and [3] share and use aggregated profits, aggregated mean demand, and the standard deviation of demand for 2 or 3 clusters, respectively (cf. Table 3.2).

We observe that for low supply rates, the stochastic Theil method outperforms clustering [2]. However, the performance differences are small and, as shown in Figure 3.8, the allocations have a similar structure. The performance difference is rooted in the fact that the Theil index provides a more accurate representation of the true profit heterogeneity across customers than does clustering [2], which accounts only for the aggregated profits of customers that are grouped into two clusters—that is, high- and low-profit customers. The more accurate information about customer heterogeneity enables the stochastic Theil method to better prioritize high-profit customers. As we can see in Figure 3.8, and for the reasons explained above, this benefit decreases as supply increases.

Compared to clustering [3], the stochastic Theil method no longer benefits from its particular measure of customer heterogeneity. At least for the baseline scenario, it appears to be sufficient to consider three customer clusters (with high, medium and low profitability) and to base allocations on aggregated information per cluster. Clustering [3] outperforms all other allocation approaches, and its *rpg* is consistently below 0.5 percent in our baseline scenario.

From a practical perspective, these results are remarkable because they suggest that a relatively simple clustering logic with three clusters (high-, average- and low-profit customers) is sufficient to obtain solutions that lead to virtually the same profit as a centralized full-information approach, which typically cannot be implemented in practice. The stochastic Theil method leads to similar, albeit slightly lower, performance. However, because this method aggregates information on profit heterogeneity and demand uncertainty into a single parameter (i.e.,

the Theil index), it requires less information to be shared and processed in the sales hierarchy (see Table 3.2). Following our arguments in Section 3.5, this is an advantage in terms of the complexity-performance trade-off. This advantage, however, comes at a cost. First, expected profits are slightly lower than those for clustering [3]. Second, from a practical perspective, the Theil index is more difficult to interpret for different (local) planners in the sales hierarchy than are clusters of high-, medium-, and low-profit customers. Our results suggest that both the stochastic Theil method and clustering ([2] or [3]) yield effective allocations; companies can choose among these approaches based on the respective implementation effort in their particular organizational context.

In summary, our results indicate that information on both profit heterogeneity and demand uncertainty must be shared and used in the sales hierarchy to enable good decentralized allocation decisions. The relative value of either of these pieces of information depends on the level of supply. Under severely constrained supply, it is more important to use accurate information about profit heterogeneity to correctly prioritize customer allocations. In situations of moderate scarcity, information about customer demand uncertainty gains importance. However, our results suggest that a relatively low level of granularity of this information, in conjunction with fairly straightforward allocation logic, is sufficient to obtain very good allocations and close to optimal performance.

3.6.4 Robustness Analysis

In this subsection, we assess the robustness of our results obtained for the baseline scenario. To provide a conclusive answer, we conducted extensive additional numerical analyses. Specifically, we evaluated and compared the performance of the different allocation methods in 20 additional experiments in which we varied both the setup of the hierarchy and all the relevant input parameters, as displayed in Table 3.3.

In these experiments, we explore various combinations of profit heterogeneity (achieved by varying the support of the uniform distribution from which we draw the profits) and CVs of demand; we also analyze the effect of heterogeneous CVs and demands by randomly drawing values of these parameters for each customer from a uniform distribution with the support specified in Table 3.3. In these experiments, we use the same 100 instances of random profits as in our baseline scenario and combine them with 20 randomly drawn CVs/demands, resulting in $|I| = 2000$

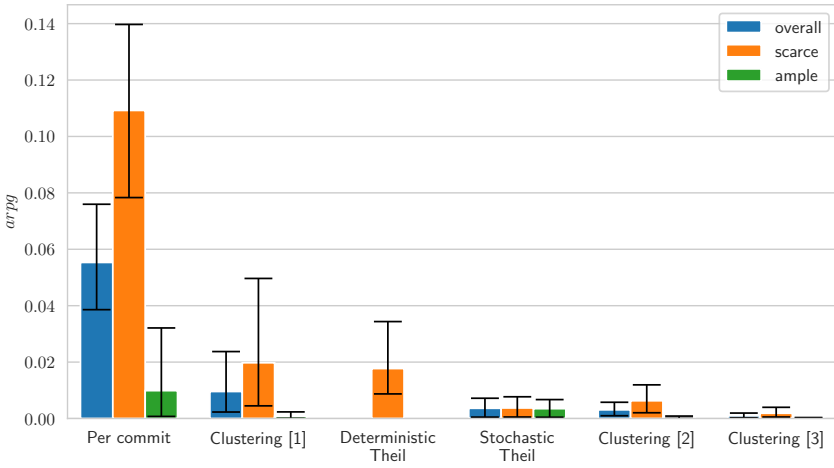


Figure 3.9: Average arpg of the allocation rules for the scenarios of the robustness analysis. (Whiskers denote the lowest/highest arpg observed.)

individual instances. Finally, we explore whether the structure of the hierarchy has an impact on the performance and modify the number of levels and the number of customers in the hierarchy.

In Figure 3.9, we report the *arpg* for the different allocation methods averaged across all 20 scenarios of our robustness analysis. The whiskers denote the best and worst results. The detailed results by scenario are listed in Table B.1 in Appendix B.3 across all supply rates and in Tables B.2 and B.3 for scarce supply (supply rate $\leq$ 1.0) and ample supply (supply rate $\geq$ 1.0), respectively.

The summarized results presented in Figure 3.9 are consistent with the results we obtained for the baseline scenario: clustering [3] leads to the lowest performance gaps and strictly outperforms its contenders in all experiments. The stochastic Theil method also produces very good results. Even though the performance slightly trails that of clustering [3], the stochastic Theil method achieves, on average, performance gaps of less than 0.5 percent for overall, scarce and ample supply.

In addition to this high-level evaluation, we also assessed and compared the performance of the different allocation methods for each individual experiment at

different levels of supply. We found the results to be perfectly in line with the structural insights we derived from our analysis of the baseline scenario.

3.7 Conclusion

This paper addresses the problem of allocating scarce supply to hierarchically structured customer segments. In such hierarchies, allocation planning is an iterative and decentralized process, in which higher-level sales quotas are disaggregated one level at a time by multiple local planners. Optimal allocations depend on the demand distributions and unit profits of all customer segments. However, sharing such detailed information across the levels of the hierarchy is undesirable from a managerial perspective. Therefore, companies commonly aggregate the demand information that is propagated through the hierarchy.

By using very coarse information, however, common information aggregation approaches, such as per commit, result in ineffective allocations. In this paper, we address the question of what information is required on the individual levels of the hierarchy to achieve effective allocations. We propose two corresponding decentralized allocation methods, namely a stochastic Theil-index approximation and a clustering approach. Both methods approximate the profit curve of the centralized problem and then solve an allocation optimization problem, given that approximation.

To evaluate the performance of our proposed methods, we consider two benchmarks: full information sharing, that is, centralized allocation, and minimum information sharing, that is, per commit allocation. These methods represent upper and lower bounds for the degree of information aggregation in the customer hierarchy. Our proposed heuristics represent an intermediate level of information aggregation.

Our results allow us to assess the importance of transmitting different types of information. We observe that to obtain good decentralized allocations, information on both profit heterogeneity and demand uncertainty must be shared and used in the hierarchy. However, a relatively coarse representation of this information turns out to be sufficient. In addition, information about profit heterogeneity is more important for correctly prioritizing customer allocations in situations of scarcity, while information on demand uncertainty is more important in situations of moderate scarcity.

Our numerical analyses suggest that both the stochastic Theil-index method and clustering with two or three customer clusters yield effective allocations; companies can choose among these approaches based on the respective implementation effort in their particular organizational context. The stochastic Theil-index method requires less information to be shared and processed in the sales hierarchy while resulting in slightly lower expected profits than clustering with three clusters. Additionally, the Theil-index method is more difficult to interpret than are clusters of high-, medium-, and low-profit customers. Even for large complicated hierarchies, a relatively simple clustering logic with three clusters is sufficient to obtain solutions that lead to virtually the same profit as a centralized full-information approach. In addition, the performance of the clustering method can always be improved by adding more clusters, which makes this method even more appealing for practical applications.

The research presented in this study opens opportunities for future research in multiple directions. First and foremost, we address a single-period setting whereas in most applications, allocation decisions have to be made repeatedly, and periods are interconnected by inventory or backlog. It would be interesting to verify our findings in such a multiperiod setting. It appears that our clustering approach lends itself to an extension in that direction. Second, in the present paper we disregard the effects of strategic behavior of individual planners. For example, planners may seek to manipulate the allocation process to their advantage by transmitting distorted information. It would be interesting to analyze how different allocation approaches encourage or discourage such strategic behavior. Vogel and Meyr (2015) point out that one advantage of their Theil-index approach in the deterministic setting is that false reporting of the Theil parameters is not beneficial, as it does not guarantee a larger allocation. Our research provides a starting point for addressing these issues in a stochastic setting and for different allocation methods.

Chapter 4

Allocation Planning under Service-Level Contracts[1]

4.1 Introduction

Allocation planning, part of the demand fulfillment process in state-of-the-art Advanced Planning Systems (APS), supports decision makers in assigning planned supply to sales organizations, customer groups, and individual customers. For instance, in SAP's APO the Global Available-to-Promise module, which is responsible for demand fulfillment (cf. Pradhan and Verma, 2012), uses a two-stage hierarchical planning process: First, planned supply, determined by master planning and/or production planning, is allocated to customer groups or individual customers based on a demand forecast. (We refer to this step as allocation planning.) The supply allocations constitute an input to order promising where, in a second step, orders are confirmed and promised a due date until the corresponding allocations are depleted. SAP's Global Available-to-Promise module offers a number of strategies to deal with orders from customers whose allocation is exhausted. (See Pradhan and Verma, 2012.)

The current systems employ relatively simple rules for both allocation planning and order promising, and the results are often "enhanced" through manual inter-

[1] This chapter was published in *European Journal of Operational Research* as Kloos and Pibernik (2020) and is co-authored by Richard Pibernik .

ventions by decision makers (planners). The per-commit approach is an example of a simple rule that is frequently used in allocation planning (Kilger and Meyr, 2015). Under per-commit, available supply is distributed evenly among the customers based on their demand forecasts, but this approach (and other simple allocation approaches) leads to suboptimal allocations when supply is scarce and customers differ in terms of their profitability and importance. (See Vogel and Meyr, 2015, and Kloos et al., 2018.) Researchers have developed and studied more sophisticated techniques for allocation planning that can remedy the problems associated with simple allocation rules (as discussed in Section 4.2). These new approaches allocate scarce supply to customers with the objective of maximizing expected profits or minimizing deviations from service-level targets. (See, e.g., Cano-Belmán and Meyr, 2019, and Kloos et al., 2018.). In doing so, they assume a direct functional relationship between the allocations in a period and the expected profit (service-level deviation) in that period. This assumption, which appears to be reasonable in numerous settings, has a problematic practical limitation as companies increasingly engage in service-level contracts with their customers. As we explain, the aforementioned assumption does not typically hold under service-level contracts. As a consequence, existing approaches to allocation planning cannot account for the particular logic and structure that underlie most service-level contracts.

Service-level contracts have become common in B2B-relationships between manufacturers and their business customers (Sieke et al., 2012). The three main elements of a service-level contract are the performance level the manufacturer must achieve, a period after which the performance is reviewed (review horizon), and the financial consequences of missing the performance level, but the design of service-level contracts varies with respect to these elements. An example of a service-level contract with a per-order fill-rate target of 100 percent and a lump-sum penalty is when the manufacturer incurs a fixed fee if it does not fulfill an order on time in full. Alternatively, a service-level contract with a fill-rate target measured over a finite horizon and a linear penalty-cost function is when the customer reviews the manufacturer's fill rate every quarter and claims a penalty for every percentage point by which the actual fill rate falls short of the target defined in the contract. As we explain in Section 4.2, the latter type of contract tends to be most plausible in manufacturing industries, so our study focuses on this type of service-level contract.

We study the allocation planning problem of a manufacturer that enters into service-level contracts with multiple customers, where the contractually speci-

fied fill-rates and penalties vary across customers (i.e., the manufacturer pursues service-level differentiation), supply is known and fixed in each period up to the review horizon, and individual customers' demand is uncertain. At the beginning of each period, the planner decides how much of her given supply she will allocate to each customer. Then, when the customer demand materializes, it is fulfilled up to the individual allocation the planner decided on, and demand that exceeds the allocation is backlogged and fulfilled from the next period's supply. The planner's objective is to allocate the available supply to minimize the total penalty payments incurred at the end of the review horizon. The structure of this problem differs significantly from the underlying structure of the allocation planning models that have been proposed in the literature: allocations and expected profits in each period of the planning horizon do not have a direct functional relationship, so instead of maximizing the sum of expected profits or minimizing expected deviations from service-level targets in individual periods, the objective is to minimize penalties that vary across customers with a piecewise-linear functional form—zero for positive deviations from the fill-rate target and increasing for negative deviations.

We formulate the allocation planning problem under service-level contracts as a stochastic dynamic program and analyze its structural properties. In particular, we derive properties of the optimal solution to this dynamic program, but because of the large state space and the ensuing "curse of dimensionality," we cannot derive an optimal allocation policy. However, our analytical results do allow us to derive the requirements that a good allocation policy must fulfill. We use these theoretical results to provide a rigorous analysis and discussion of simple allocation rules that have been proposed and are popular in practice and to develop and study new and advanced allocation policies that are more complex but that lead to superior performance under most conditions. In addition to our theoretical analysis, we carry out extensive numerical analyses to quantify the performance of various policies under various conditions and to derive recommendations for when to use which policy.

The remainder of this paper is organized as follows: Section 4.2 provides an overview of the literature on service-level contracts and allocation planning, explains our choice of a type of contract, and positions our contribution relative to previous work. In Section 4.3 we describe our setting and formulate the stochastic dynamic programming model. Section 4.4 characterizes the optimal allocation policy and derives insights on the factors that influence the optimal decision. Based on

those insights, Section 4.5 derives a myopic policy and three multi-period heuristic policies to generate "good" allocations. Section 4.6 presents the results of an extensive numerical study that we carried out to evaluate the performance of the various allocation policies and to derive theoretical insights and managerial recommendations.

4.2 Literature Review

The research presented in this paper is related to the literature on allocation planning and work on the management of service-level agreements and service-level contracts.

Most research on allocation planning focuses on determining profit-maximizing allocations for customers who differ in terms of their profitability. Ball et al. (2004), who refer to allocation planning as "push-based ATP," are among the first to propose deterministic and stochastic models for allocation planning. They point out that allocation planning can be viewed as a specific type of quantity-based revenue management. Using the term "allocated Available-to-Promise," Quante et al. (2009b) provide a comprehensive overview of the literature and of software that apply revenue management to demand fulfillment. They stress that demand fulfillment in the manufacturing industry is typically a multi-period problem, so traditional revenue management methods cannot be readily applied. Quante et al. (2009a) propose a stochastic dynamic programming model to allocate available supply to customers that differ in terms of their per-unit profits. Using a numerical experiment, they demonstrate that their approach leads to a significant increase in profits over that provided by other simple rules or by promising orders on a first-come-first-served basis. Their results also suggest that their (stochastic) approach can significantly increase profits over those offered by a deterministic method developed by Meyr (2009). Eppler (2015) extends Quante et al.'s (2009a) approach to incorporate nesting across allocations of multiple customer classes.

Based on a case study by Roitsch and Meyr (2015), Vogel and Meyr (2015) address the issue of allocation planning in hierarchical sales organizations, where local decision makers determine allocations in a decentralized fashion. For a single-period setting with deterministic demand, the authors show that a common decentralized, profit-based allocation can lead to a significantly higher loss in total profit compared to the global optimum because the decentralized allocation approach av-

erages customers' profits on each level, resulting in a loss of relevant information. As a consequence, the approach fails to prioritize customers with higher profits. Using a measure of income inequality from the economic literature, the Theil index, to capture the profit heterogeneity of customer groups, they develop a new allocation approach that overcomes this problem. Based on numerical analyses, the authors show that their approach leads to close to optimal allocations and outperforms the conventional rules, at least when demand is deterministic. Cano-Belmán and Meyr (2019) extend this approach to a multi-period setting.

While most research focuses on determining profit-maximizing customer allocations, some also considers service-related objectives. For example, Pibernik and Yadav (2008) and Pibernik and Yadav (2009) propose multi-period allocation and order promising models for two customer classes and a decision maker who wants to ensure that high-priority customers receive a minimum service level. Kloos et al. (2018) analyze a hierarchical single-period setting, similar to that of Vogel and Meyr (2015) but with stochastic demand and with the objective of reaching heterogeneous service-level targets. Similar to Vogel and Meyr (2015), Kloos et al. (2018) find that close-to-optimal allocations can be achieved by means of advanced, decentralized allocation rules.

The literature on inventory rationing (e.g., Deshpande et al., 2003; Schulte and Pibernik, 2016) also addresses the question concerning of how best to allocate supply (inventory) to various customer classes. As such, it shares similarities with allocation planning, but the key difference is that allocation planning assumes that supply is known and fixed over a given period, while inventory rationing considers supply a decision variable and seeks to optimize both replenishment and rationing decisions, which may make inventory rationing more difficult than allocation planning. However, in many settings, supply that is available to a local sales organization cannot be adjusted freely, as it is mostly fixed for a given period of time. While long lead times are the most obvious reason, manufacturers' current hierarchical planning practices are the more important reason. Manufacturers that have decentralized and hierarchical sales organizations—see Kloos et al. (2018) for examples—commonly plan for supply on a central level (e.g., at the firms' headquarters) and allocate it top-down to regional sales organizations and then to local sales organizations. This approach is also reflected in state-of-the-art APS like SAP APO (Kilger and Meyr, 2015). Under such a decentralized allocation logic, sales organizations have limited flexibility in the supply they require, especially when overall supply is scarce. Our analysis focuses on planners in local sales organi-

zations who cannot adjust their supplies, particularly when supply is scarce so allocation planning is highly consequential.

Our work on allocation planning under service-level contracts differs from previous research on allocation planning and that on inventory rationing in two primary ways. First, as highlighted in Section 4.1, when there is a service-level contract, there is no direct functional relationship between the allocations in a period and the expected penalties, as the penalties incurred can be determined only after all demands have materialized. Consequently, our model must track fulfilled demand and total demand across multiple periods, whereas previous allocation planning models track only the inventory position. Second, we assume that supply is known and fixed during the review horizon that the service-level contract specifies, whereas inventory rationing research considers supply a decision variable.

Our research is also related to previous work on service-level agreements and service-level contracts, both of which have gained attention in operations management research. Table 4.1 provides a high-level overview of studies on service-level contracts/agreements and their service-level measures, review horizons, and penalty mechanisms. The literature does not use the terms *service-level contract* and *service-level agreement* consistently, as most authors use *service-level contract* when the consequences of missing the service-level targets are specified and *service-level agreement* when the consequences are not explicit. Therefore, we refer to a *service-level agreement* as the combination of a performance measure (e.g., a fill-rate target), a review horizon (e.g., three months), and a service-level target (e.g., 95%) and use *service-level contract* to refer to an explicit penalty assigned to deviations from the service-level target.

Chen and Thomas (2018) study on the purchasing conditions of retailers in the US finds that about 64 percent of the companies they surveyed use service-level agreements in combination with some kind of penalty mechanism—that is, a service-level contract. The most common type of contract in their study was a single-period/per-order fill-rate-based contract with lump-sum penalties, although most studies on service-level contracts and/or agreements assume a multi-period review horizon (cf. Table 4.1). We attribute this to the requirements of the retail industry, which is characterized by high service-level requirements—for instance, 73 percent of the service-level agreements in the study require a fill-rate of 100 percent—frequent orders, and flexible production (cf. Quante et al., 2009b). Orders in the manufacturing industry are less frequent, lead-times are longer, and capacity

Table 4.1: Overview on the literature on service-level contracts and service-level agreements.

	Measure		**Horizon**			**Penalty**	
	Fill-rate	Ready-rate	Per order	Single-period	Multi-period	Lump sum	Linear
Abbasi et al. (2017)	✓	–	–	–	✓	–	–
Alamri et al. (2017)	–	✓	–	–	✓	✓	✓
Chen and Thomas (2018)	✓	(✓)	✓	✓	✓	✓	✓
Katok et al. (2008)	✓	–	–	–	✓	–	–
Liang and Atkins (2013)	✓	–	–	–	✓	✓	✓
Protopappa-Sieke et al. (2016)	✓	–	–	–	✓	✓	–
Sieke et al. (2012)	✓	✓	–	✓	–	✓	✓
Thomas (2005)	✓	–	–	–	✓	–	–

is less flexible, so penalizing minor deviations from the service-level targets in the short term is undesirable, as deviations may result from small fluctuations in demand or lead times. Since a longer review horizon appears to be more appropriate for manufacturing settings (cf. Liang and Atkins, 2013), we focus on service-level contracts with multi-period review horizons. Clearly, such is also the more general case, as a multi-period setting with a review horizon of one period corresponds to a single-period setting.

Liang and Atkins (2013) compare the optimal inventory policies for lump-sum and linear penalties under service-level contracts for a multi-period review horizon and observe that lump-sum penalties lead to situations in which it is optimal for the manufacturer to stop serving some customers—that is, it is cheaper for the manufacturer to pay the penalty than to try to reach the fill-rate target, which may still be missed because demand in future periods is uncertain. This issue does not occur for penalty costs that are proportional to (negative) deviation from the fill-rate target. Other authors also observe this disadvantage of lump-sum penalties under multi-period review horizons and various settings (cf. Protopappa-Sieke et al., 2016; Chen and Thomas, 2018; Sieke et al., 2012). From a theoretical perspective, it makes sense that the optimal policy may prescribe to stop serving a customer when lump-sum penalties are incurred, although, from a practical perspective, it is unlikely a manufacturer will do so for fear of additional negative consequences (e.g., contract termination). However, these consequences are difficult to quantify, so service-level contracts with lump-sum penalties are likely either to lead to policies that a planner would not pursue or to require additional assumptions that are difficult to justify. To avoid these methodological drawbacks, we focus on contracts with piece-wise linear penalty cost functions—that is, penalty cost functions that linearly increase in negative deviations from the fill-rate target and that are zero for positive deviations from the fill-rate target.

Few studies address the problem of allocating supply under service-level contracts with linear penalty mechanisms and multi-period review horizons. Abbasi et al. (2017) consider the case of homogeneous service-level contracts—that is, service-level contracts that are identical for all customers—and find that a myopic policy that minimizes fill-rate deviations leads to a substantially higher probability of reaching the customers' fill-rate targets than does a policy that uses a simple first-come-first-served approach.

The setting Abbasi et al. (2017) consider differs from ours in several respects. First, their setting assumes that the manufacturer can observe demand before mak-

ing the allocation decision, while in our setting, allocations are made before demands are realized. Second, Abbasi et al. (2017) assume a base-stock policy with zero lead-time, which implies that supply in each period is variable and independent of the demand realizations in previous periods, while in our setting, the supply that can actually be allocated to customers depends on the planned (fixed) supply and the demand realized in previous periods, so the supply that can be allocated is a random variable. Third, Abbasi et al. (2017) assume that the service-level contracts are homogeneous, where we account for different penalties and fill-rate targets. Finally, Abbasi et al. (2017) focus only on evaluating myopic policies that do not anticipate future supply and demand realizations, while we formulate and study a multi-period stochastic allocation problem and derive (heuristic) policies that anticipate future supply and demand scenarios and, as such, are not myopic but forward-looking.

Chen and Thomas (2018) analyze various service-level contracts and agreements under a base stock policy with zero lead time, among them a service-level contract with fill-rate targets and a multi-period review horizon. For this type of contract they propose a myopic policy that maximizes the number of customers over their fill-rate target and find that this policy outperforms other myopic approaches. Just as Abbasi et al. (2017), Chen and Thomas (2018) assume that demand is known prior to the allocation decision.

Our study is the first to address allocation planning in the presence of heterogeneous service-level contracts. We formalize the manufacturer's problem, establish properties of the optimal allocation policy, derive requirements for good allocation policies based on those properties, and use these requirements to evaluate established ("simple") allocation rules and other approaches that are proposed in the literature. We also develop four new advanced allocation policies and evaluate their performance by means of an extensive numerical experiment so we can generate theoretical insights and derive recommendations on when to use which policy.

4.3 Model Description

We consider a single manufacturer selling a single product to multiple customers $l \in \mathcal{L}$. The manufacturer negotiates a service-level contract with each customer that specifies a review period of R periods (which we assume to be the same

for all customers), a customer-specific fill-rate target β_l to be met at the end of the review horizon (at time $R+1$), and a customer-specific penalty cost p_l that penalizes (negative) deviations from the fill-rate target. We provide a formal characterization of the manufacturer's penalty cost structure below.

Customer l's demand in period t is uncertain, and we model it as a random variable, denoted by $D_{l,t}$. We assume $D_{l,t}$ is continuous and iid, with known mean μ_l and standard deviation σ_l. Let $f_{l,t}$ and $F_{l,t}$ denote the pdf and the cdf of the demand distribution, respectively.

We denote the manufacturer's inventory position at the beginning of period $t \in \{1, 2, \ldots, R\}$ by i_t. The inventory position can be negative (i.e., $i_t < 0$ for any $t \in \{2, \ldots, R\}$ if some demand was backordered in the previous period, $t-1$. The manufacturer receives a supply of r_t units at the beginning of each period t. Because we assume that any backordered demand in period $t-1$ has to be cleared before any new demand (in period t) can be fulfilled, the manufacturer has a net supply of $r_t + i_t$ units of the product at its disposal. When it knows the net supply, the manufacturer determines individual allocations $a_{l,t} \geq 0$ to all customers $l \in \mathcal{L}$ in period t. We denote by $\boldsymbol{a}_t$ the $|\mathcal{L}|$-dimensional allocation vector in period t. (We use bold symbols to represent vectors.) Demand from any customer $l \in \mathcal{L}$ in period t can be fulfilled only from the corresponding allocation $a_{l,t}$—we do not consider nesting of any kind. If the demand realization $d_{l,t}$ exceeds $a_{l,t}$, the customer receives $a_{l,t}$ and the manufacturer backlogs $d_{l,t} - a_{l,t}$.

In order to keep track of the realized fill rates at the beginning of period t ($t \in \{2, \ldots, R+1\}$), the manufacturer records the total demand $x_{l,t}$ of each customer $l \in \mathcal{L}$ that materialized in periods $t = \{1, \ldots, t-1\}$ and the total demand of each customer that was fulfilled on time in periods $t = \{1, \ldots, t-1\}$. We denote the latter by $y_{l,t}$. The current customer fill rate in t is $\hat{\beta}_{l,t} = y_{l,t}/x_{l,t}$. In each period t the sequence of events is as follows:

1. The manufacturer receives r_t units of supply.
2. If $i_t < 0$, the manufacturer clears all backordered demand from the previous period.
3. The manufacturer computes the net supply $r_t + i_t$ that is available for period t.
4. The manufacturer determines the allocations $\boldsymbol{a}_t$ for its customers $l \in \mathcal{L}$.
5. The manufacturer observers the demand realization $d_{l,t}$ for each customer $l \in \mathcal{L}$ in period t, fulfills $\min\{a_{l,t}, d_{l,t}\}$, and backorders $\max\{0, d_{l,t} - a_{l,t}\}$.

6. The manufacturer records the total demand $x_{l,t+1}$ of each customer $l \in \mathcal{L}$ that materialized in periods $t = \{1, \dots, t\}$ and records $y_{l,t+1}$, the total demand of each customer that was fulfilled on time in periods $t = \{1, \dots, t\}$.

The resulting state at the beginning of period t is given by the total demands $\boldsymbol{x}_t = (x_{l,t})_{l \in \mathcal{L}}$, fulfilled demands $\boldsymbol{y}_t = (y_{l,t})_{l \in \mathcal{L}}$, and the inventory position i_t. Therefore, the state space is $2|\mathcal{L}| + 1$ dimensional. In the first period, $t = 1$, no demands have yet occurred or been fulfilled, and $\boldsymbol{x}_1 = \boldsymbol{y}_1 = \mathbf{0}$. Based on this sequence of events, we formulate the state transition in Definition 4.1.

Definition 4.1 (State transition function)**.** *Let $\boldsymbol{s}_t = (\boldsymbol{x}_t, \boldsymbol{y}_t, i_t)$ denote the state at the beginning of period $t \in \{1, 2, \dots, R+1\}$, and let u denote the state transition function, such that $\boldsymbol{s}_{t+1} = u(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{d}_t)$. The state transition can be characterized by the following equations:*

$$
\begin{aligned}
x_{l,t+1} &= x_{l,t} + d_{l,t} && \forall l \in \mathcal{L} \\
y_{l,t+1} &= y_{l,t} + \min\{d_{l,t}, a_{l,t}\} && \forall l \in \mathcal{L} \\
i_{t+1} &= i_t + r_t - \sum_{l \in \mathcal{L}} d_{l,t}
\end{aligned}
$$

In any period $t \in \{1, 2, \dots, R\}$, the manufacturer cannot allocate more than the net supply $r_t + i_t$. Therefore, we can define the set of feasible allocations as in Definition 4.2.

Definition 4.2 (Set of feasible allocations)**.** *Let $A_t(i_t)$ denote the set of feasible allocations in period $t \in \{1, \dots, R\}$. Then*

$$
A_t(i_t) = \left\{ \boldsymbol{a}_t \;\middle|\; ||\boldsymbol{a}_t||_1 \leq [r_t + i_t]^+, \boldsymbol{a}_t \geq \mathbf{0} \right\},
$$

where $[z]^+ = \max\{z, 0\}$.

We assume a linear penalty cost scheme that penalizes negative deviations from the fill-rate target β_l. Accordingly, we define the penalty cost function of customer $l \in \mathcal{L}$ as:

$$
C_l(x_{l,R+1}, y_{l,R+1}) = p_l[\beta_l - y_{l,R+1}/x_{l,R+1}]^+. \tag{4.1}
$$

For the remainder of our analysis, we assume that the penalty costs are the only monetary consequences that are associated with the allocation decision. While

this assumption is convenient for our analytical and numerical analyses, it is also reasonable in our setting. Because supply is fixed and all demands are either fulfilled immediately, or backlogged and fulfilled at a later time, differences in unit profits and holding costs are not relevant to the allocation decisions. It is also reasonable to assume that no additional costs for backordering occur when service-level contracts are employed, as backorder costs are already captured by the penalty costs and the fill-rate target. Simply speaking: under a service-level contract, some backordering is permitted but will be punished if it exceeds the limit defined by the service-level target.

Next, we formalize the manufacturer's decision making problem.

Problem 4.1 (Decision problem of the manufacturer)**.**

$$\begin{aligned} \min \quad & \mathbb{E}\left[\sum_{l\in\mathcal{L}} C_l(x_{l,R+1}, y_{l,R+1})\right] & \\ \text{subject to} \quad & \boldsymbol{s}_{t+1} = u(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{D}_t) & \forall t \in \{1,2,\ldots,R\} \\ & \boldsymbol{a}_t \in A_t(i_t) & \forall t \in \{1,2,\ldots,R\}. \end{aligned}$$

In Problem 4.1 the objective function depends only on the realized state $\boldsymbol{s}_{R+1}$ at the end of period R. However, because subsequent states are interrelated (through the state transition function u), and the sets of feasible allocations $A_t(i_t)$ in periods $t \in \{1,2,\ldots,R\}$ are state-dependent, Problem 4.1 is a multi-period stochastic optimization problem. The manufacturer wants to determine an allocation policy $\alpha_t : \boldsymbol{s}_t \mapsto \boldsymbol{a}_t$ that determines an allocation $\boldsymbol{a}_t$ that depends on the current state $\boldsymbol{s}_t$ and minimizes the total expected penalty costs at the end of the review horizon. In the next section, we derive properties of the optimal allocation policy and discuss their implications.

4.4 Optimal Allocation Policy

This section first provides the dynamic programming formulation for Problem 4.1 and characterizes separately the optimal allocation policies in the terminal period R and in non-terminal periods $t = 1,\ldots,R-1$. It turns out that computing the optimal policy for periods $t = 1,\ldots,R-1$ is not feasible because of the high dimensionality of the state space, so we resort to alternative techniques (e.g., approximate dynamic programming) to solve Problem 4.1. The results we obtain

from our analysis of the optimal allocation policy will help us to develop and assess alternative heuristics for solving the manufacturer's problem.

We denote the value of state $\boldsymbol{s}_t$ in period t as $V_t(\boldsymbol{s}_t)$ and state the Bellman equations for Problem 4.1 as:

$$V_R(\boldsymbol{s}_R) = \min_{\boldsymbol{a}_R \in A_R(i_R)} \mathbb{E}\left[\sum_{l \in \mathcal{L}} C_l(X_{l,R+1}, Y_{l,R+1})\right] \tag{4.2}$$

$$V_t(\boldsymbol{s}_t) = \min_{\boldsymbol{a}_t \in A_t(i_t)} \mathbb{E}\left[V_{t+1}(u(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{D}_t))\right] \qquad \forall t \in \{R-1, R-2, \ldots, 1\}. \tag{4.3}$$

From the Bellman equations we derive the corresponding optimal policy functions $\alpha_t^*(\boldsymbol{s}_t)$.

$$\alpha_R^*(\boldsymbol{s}_R) = \operatorname*{argmin}_{\boldsymbol{a}_R \in A_R(i_R)} \mathbb{E}\left[\sum_{l \in \mathcal{L}} C_l(X_{l,R+1}, Y_{l,R+1})\right] \tag{4.4}$$

$$\alpha_t^*(\boldsymbol{s}_t) = \operatorname*{argmin}_{\boldsymbol{a}_t \in A_t(i_t)} \mathbb{E}\left[V_{t+1}(u(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{D}_t))\right] \qquad \forall t \in \{R-1, R-2, \ldots, 1\} \tag{4.5}$$

In the next section, we first determine the optimal policy for the terminal period R and then analyze the optimal policy for non-terminal periods $R-1, \ldots, 1$.

4.4.1 Optimal Allocation Policy in the Terminal Period

Given a certain state $\boldsymbol{s}_R$ at the beginning of period R, the manufacturer has to solve a stochastic knapsack problem to determine the optimal allocations $\boldsymbol{a}_R^* = \alpha_R^*(\boldsymbol{s}_R)$. Knowing the penalty cost function (4.1) and the demand distribution $f_{l,R}(d_{l,R})$ for period R, we can derive expressions for the expected penalty costs and the marginal expected penalty costs of customer $l \in \mathcal{L}$, depending on the allocation a_l. Recall that the penalty cost function of customer $l \in \mathcal{L}$ is piecewise linear with a slope of zero for $y_{l,R+1}/x_{l,R+1} \geq \beta_l$ and a slope of p_l for $y_{l,R+1}/x_{l,R+1} \leq \beta_l$. At the beginning of period R, the manufacturer knows the current fill-rate $\hat{\beta}_{l,R} = y_{l,R}/x_{l,R}$ and can calculate the minimum demand that needs to be fulfilled in period R to meet the fill-rate target β_l and avoid penalty costs. We denote this quantity as $d_{min,l}$ and compute it as $d_{min,l} = \frac{\beta_l x_{l,R} - y_{l,R}}{1-\beta_l}$. (If $d_{min,l} \leq 0$, the current fill rate is above the target; that is, $\hat{\beta}_{l,R} \geq \beta_l$.)

If the manufacturer allocates less than $d_{min,l}$ units of supply to customer l, the fill-rate target β_l will definitely not be reached, independent of customer l's demand in period R. Clearly, for any allocation $a_l \geq d_{min,l}$, whether the manufacturer can meet the fill-rate target β_l of customer l depends on the demand realization $d_{l,R}$. For a given allocation $a_{l,R}$, we can calculate the maximum demand of customer l at which the manufacturer still meets the fill-rate target and does not incur penalty costs. We denote this quantity as $d_{max,l}$ and calculate it as $d_{max,l}(a_{l,R}) = \frac{y_{l,R}+a_{l,R}}{\beta_l} - x_{l,R}$.

Proposition 4.1 (Expected and marginal expected penalty of a customer l in the terminal period R)**.**

1. $$\mathbb{E}\left[C_l(X_{l,R+1}, Y_{l,R+1})\right] = \begin{cases} \int_{d_{max,l}(a_{l,R})}^{\infty} p_l \left(\beta_l - \frac{y_{l,R}+a_{l,R}}{x_{l,R}+d_{l,R}}\right) f_{l,R}(d_{l,R})\, \mathrm{d}d_{l,R} & \text{if } \hat{\beta}_{l,R} \geq \beta_l \\ \int_0^{d_{min,l}} p_l \left(\beta_l - \frac{y_{l,R}+d_{l,R}}{x_{l,R}+d_{l,R}}\right) f_{l,R}(d_{l,R})\, \mathrm{d}d_{l,R} & \\ \quad + \int_{d_{max,l}(a_{l,R})}^{\infty} p_l \left(\beta_l - \frac{y_{l,R}+a_{l,R}}{x_{l,R}+d_{l,R}}\right) f_{l,R}(d_{l,R})\, \mathrm{d}d_{l,R} & \text{if } \hat{\beta}_{l,R} < \beta_l,\ a_{l,R} \geq d_{min,l} \\ \int_0^{a_{l,R}} p_l \left(\beta_l - \frac{y_{l,R}+d_{l,R}}{x_{l,R}+d_{l,R}}\right) f_{l,R}(d_{l,R})\, \mathrm{d}d_{l,R} & \\ \quad + \int_{a_{l,R}}^{\infty} p_l \left(\beta_l - \frac{y_{l,R}+a_{l,R}}{x_{l,R}+d_{l,R}}\right) f_{l,R}(d_{l,R})\, \mathrm{d}d_{l,R} & \text{if } \hat{\beta}_{l,R} < \beta_l,\ a_{l,R} < d_{min,l} \end{cases}$$

2. $$\frac{\mathrm{d}}{\mathrm{d}a_{l,R}} \mathbb{E}\left[C_l(X_{l,R+1}, Y_{l,R+1})\right] = -\lambda_{l,R}(a_{l,R}) = \begin{cases} -p_l \int_{d_{max,l}(a_{l,R})}^{\infty} \frac{1}{x_{l,R}+d_{l,R}} f_{l,R}(d_{l,R})\, \mathrm{d}d_{l,R} & \text{if } a_{l,R} \geq d_{min,l} \\ -p_l \int_{a_{l,R}}^{\infty} \frac{1}{x_{l,R}+d_{l,R}} f_{l,R}(d_{l,R})\, \mathrm{d}d_{l,R} & \text{else.} \end{cases}$$

Part 1 of Proposition 4.1 distinguishes three cases based on the current fill rate $\hat{\beta}_{l,R}$ and the allocation $a_{l,R}$. In the first case, the fill rate at the beginning of period R is greater than or equal to the fill-rate target ($\hat{\beta}_{l,R} \geq \beta_l$). Since a penalty is incurred only if demand is higher than the threshold level $d_{max,l}(a_{l,R})$, the first term captures the expected penalty for this case. In the second case, the current fill rate is below the fill-rate target, and the manufacturer chooses an allocation that exceeds the minimum threshold level. In this situation, the manufacturer incurs penalty costs for demand realizations that are either too low to achieve the fill-rate target (i.e., $d_{l,R} \leq d_{min,l}$) or too high compared to the manufacturer's allocation (i.e., $d_{l,R} > d_{max,l}(a_{l,R})$). The third case is straightforward: The manufacturer incurs a penalty

for customer l because the current fill rate is lower than the fill-rate target, and the allocation $a_{l,R}$ is lower than the minimum threshold $d_{min,l}$. The realized penalty costs are positive and decreasing (increasing) in demand realizations $d_{l,R} \leq d_{min,l}$ ($d_{l,R} > d_{max,l}(a_{l,R})$). The last expression in Part 1 of Proposition 4.1 captures the corresponding expected penalty costs.

Based on the results presented in Proposition 4.1, we can show that the total expected cost in period R is convex.

Lemma 4.1 (Convexity of total expected cost in the terminal period R)**.** $\mathbb{E}\left[\sum_{l\in\mathcal{L}} C_l(X_{l,R+1}, Y_{l,R+1})\right]$ *is convex in* $\boldsymbol{a}_R$.

Knowing that $\mathbb{E}\left[\sum_{l\in\mathcal{L}} C_l(X_{l,R+1}, Y_{l,R+1})\right]$ is convex in $\boldsymbol{a}_R$, we can apply standard Lagrangian optimization techniques to characterize the optimal allocation decision in period R (Equation 4.4).

Theorem 4.1 (Optimal allocation in the terminal period R)**.** $\boldsymbol{a}_R^* = (a_{l,R}^*)_{l\in\mathcal{L}}$ *is optimal if and only if there exists a value of* $\lambda > 0$*, such that, using*

$$A_\lambda = \left\{l \mid l \in \mathcal{L}, \lambda \geq \lambda_{l,R}(0)\right\} \tag{4.6}$$

the following hold

$$\lambda = \lambda_{l,R}(a_{l,R}^*) \qquad \text{for all } l \in \mathcal{L}\backslash A_\lambda \tag{4.7}$$

$$a_{l,R}^* = 0 \qquad \text{for all } l \in A_\lambda. \tag{4.8}$$

The results presented in Theorem 4.1 and the corresponding proof are established in Kloos et al. (2018) for a single-period allocation problem. We review the underlying logic and intuition because they are useful in our further analyses.

The intuition behind Theorem 4.1 is most apparent in a situation in which supply becomes available gradually, staring from zero supply. The first unit of supply is allocated to the customer with the highest initial marginal expected penalty ($\lambda_{l,R}(0)$), and no other customers receive an allocation, as they are contained in the set A_λ. As more supply becomes available and is allocated to the customer with the highest initial marginal expected penalty, the probability of reaching the fill-rate target increases while the probability that additional units of supply will be consumed decreases—and both reduce the marginal expected penalty. At some point, the marginal expected penalty (λ) will be equal to the initial marginal expected penalty of another customer—who then receives its first allocation. The sup-

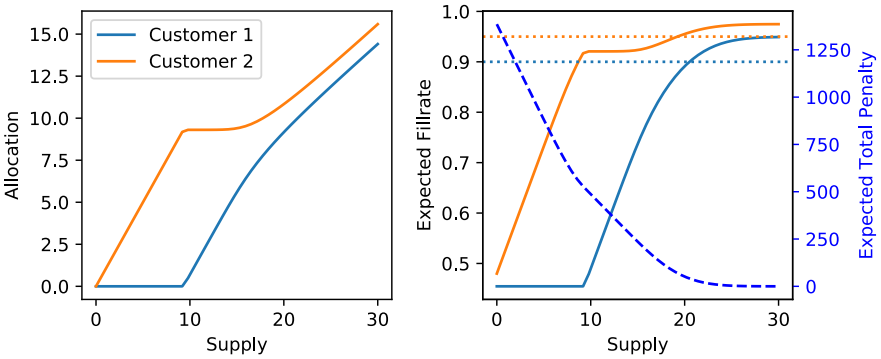


Figure 4.1: Allocation, expected fill-rates and expected penalties (dashed line) under optimal allocation for two customers with normal demand under varying levels of supply. ($\mu_1 = \mu_2 = 10$, $\sigma_1 = \sigma_2 = 3$, $p_1 = 1000$, $p_2 = 2000$, $\beta_1 = 0.9$, $\beta_2 = 0.95$, $x_{1,R} = x_{2,R} = 10$, $y_{1,R} = 9$, $y_{2,R} = 9.5$)

ply is shared among these customers ($\mathcal{L} \backslash A_\lambda$) such that they have equal marginal expected penalties (Equation 4.7). As yet more supply becomes available, λ decreases, and more customers receive an allocation—that is, the cardinality of set A_λ decreases.

Figure 4.1 plots the optimal allocations and the associated expected fill rates and penalties for an example with two customers. The figure shows the sequential nature of the optimal allocation: As Customer 2 has a higher penalty than Customer 1 and both customers' demands follow the same distribution, Customer 2's initial marginal expected penalty is higher than that of Customer 1. At about ten units of supply, the marginal penalty of Customer 2 is equal to the initial marginal penalty of Customer 1, at which point the first unit of supply is allocated to Customer 1.

Figure 4.1 plots the expected total penalty and the expected fill rates for both customers and shows that an expected penalty that is close to zero is achieved only when both customers' expected fill rates are significantly above their corresponding targets.

The expressions of the marginal expected penalties (Proposition 4.1, Part 2) allow us to determine how the optimal allocation is influenced by the state and other parameters. If we assume customer $l \in \mathcal{L} \backslash A_\lambda$, and $|\mathcal{L} \backslash A_\lambda| \geq 2$, then, ceteris paribus, an increasing marginal penalty $\lambda_{l,R}(a_{l,R})$ increases the allocation to this customer. Hence, by analyzing the effect on $\lambda_{l,R}(a_{l,R})$, we can determine how a change in parameters affects the optimal allocation.

First, as $\frac{\mathrm{d}}{\mathrm{d}p_l}\lambda_{l,R}(a_{l,R}) > 0$, increasing the penalty parameter p_l increases the corresponding customer's marginal expected penalty, so higher penalties lead to increased allocations, which is highly intuitive. Second, because $\frac{\mathrm{d}}{\mathrm{d}x_{l,R}}\lambda_{l,R}(a_{l,R}) < 0$, higher total demands $x_{l,R}$ decrease the allocation. Third, we analyze the effect of the total fulfilled demand $y_{l,R}$ and observe that, when $a_{l,R} \geq d_{min,l}$, $\frac{\mathrm{d}}{\mathrm{d}y_{l,R}}\lambda_{l,R}(a_{l,R}) < 0$, so having fulfilled more demands in preceding periods decreases the allocation to the corresponding customer. If, however, $a_{l,R} < d_{min,l}$, $\frac{\mathrm{d}}{\mathrm{d}y_{l,R}}\lambda_{l,R}(a_{l,R}) = 0$, the optimal allocation does not change in the fulfilled demand, suggesting that in certain situations the optimal allocation in period R does not depend on all elements of the state space $\boldsymbol{s}_R$. We formalize this observation in Proposition 4.2.

Proposition 4.2 (Independence of the optimal allocation in the terminal period R)**.** *Let $\boldsymbol{a}_R^* = \alpha_R(\boldsymbol{s}_R)$ be the optimal allocation for state $\boldsymbol{s}_R = (\boldsymbol{x}_R, \boldsymbol{y}_R, i_R)$, and denote with $J = \{l \mid l \in \mathcal{L}, a_{l,R}^* \leq d_{min,l}\}$ the set of customers that receive an optimal allocation $a_{l,R}^* < d_{min,l}$. Then $\boldsymbol{a}_R^* = \alpha_R(\boldsymbol{s}_R')$ is the optimal allocation for all states $\boldsymbol{s}_R' = (\boldsymbol{x}_R', \boldsymbol{y}_R', i_R')$ for which the following hold:*

$$
\begin{aligned}
\boldsymbol{x}_R' &= \boldsymbol{x}_R \\
i_R' &= i_R \\
y_{l,R}' &\leq \beta_l x_{l,R} - a_{l,R}^*(1-\beta_l) && \forall l \in J \\
y_{l,R}' &= y_{l,R} && \forall l \in \mathcal{L} \backslash J
\end{aligned}
$$

Proposition 4.2 allows us to generalize an optimal solution to a subset of the state space. Although the property described in Proposition 4.2 is comparatively straightforward, it is highly useful, as it substantially decreases the computational effort, especially when larger areas of the state space are evaluated. We exploit this property in the numerical analyses of our temporal aggregation heuristic that we introduce and analyze in Section 4.5.2.

4.4.2 Optimal Allocation Policy in Non-terminal Periods

We now address the problem of determining optimal allocations in periods $t = R-1, \ldots, 1$—that is, we solve Equation (4.3). We begin our analysis by characterizing an allocation's expected marginal penalty in an arbitrary (non-terminal) period $t \in \{1, \ldots, R-1\}$

Proposition 4.3 (Expected marginal penalty in non-terminal period t)**.** *The expected marginal penalty of allocating $a_{l,t}$ to customer $l \in \mathcal{L}$ in period $t \in \{1, \dots, R-1\}$ is*

$$\begin{aligned}\mathbb{E}\left[\frac{\mathrm{d}}{\mathrm{d}a_{l,t}} V_{t+1}(u(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{D}_t))\right] &:= -\lambda_{l,t}(\boldsymbol{a}_t) \\ &= \mathbb{E}\left[\frac{-p_l}{X_{l,R+1}} \cdot \mathbb{1}[D_{l,t} \geq a_{l,t}] \cdot \mathbb{1}\left[\frac{Y_{l,R+1}}{X_{l,R+1}} \leq \beta_l\right]\right], \quad (4.9)\end{aligned}$$

where $\mathbb{1}[exp] = \begin{cases} 1 & \text{if } exp \text{ is True} \\ 0 & \text{else.} \end{cases}$

The formula for the expected marginal penalty in Proposition 4.3 is largely intuitive: the first term captures the expected marginal change in the penalty, and the second and the third terms are the probabilities that the allocation is consumed and that the fill rate is below the target fill rate, respectively. Based on (4.9) we can show that (4.3) is convex.

Lemma 4.2. $\mathbb{E}[V_{t+1}(u(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{D}_t))]$ *is convex in* $\boldsymbol{a}_t$.

Because $\mathbb{E}[V_{t+1}(u(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{D}_t))]$ is convex, we can use the Karush-Kuhn-Tucker conditions to characterize optimal solutions to (4.3).

Theorem 4.2 (Optimal allocation in non-terminal period t)**.** $\boldsymbol{a}_t^*$ *is optimal if and only if there exists a value of $\lambda > 0$, such that, using*

$$A_\lambda = \left\{ l \mid l \in \mathcal{L}, \lambda \geq \lambda_{l,t}(\boldsymbol{a}_t^*) \right\}, \quad (4.10)$$

the following hold

$$\lambda = \lambda_{l,t}(\boldsymbol{a}_t^*) \qquad \text{for all } l \in \mathcal{L} \backslash A_\lambda \quad (4.11)$$

$$a_{l,t}^* = 0 \qquad \text{for all } l \in A_\lambda. \quad (4.12)$$

The results presented in Theorem 4.2 are structurally similar to those of Theorem 4.1: In the optimum, customers with a marginal expected penalty reduction lower than λ receive an allocation of zero (4.12); all other customers receive an allocation greater than zero, leading to equal marginal expected penalties (4.11). However, finding solutions to (4.10) and (4.11) is difficult because the value function V_{t+1} must be evaluated over the entire real-valued and $|\mathcal{L}|$-dimensional demand vector $\boldsymbol{D}_t$ to compute the expected marginal penalty $\lambda_{l,t}(\boldsymbol{a}_t)$. Even though

we can solve the problem for period R efficiently, evaluating V_R for all possible demand realizations is already a computational challenge, and doing so for periods $t \in \{1, \ldots, R-1\}$ becomes computationally infeasible. In addition, the expected marginal penalty $\lambda_{l,t}(\boldsymbol{a}_t^*)$ in (4.10) and (4.11) depends on $\boldsymbol{a}_t^*$, that is, on all (other) optimal allocations, implying that the marginal penalties of all allocations are interdependent. As a consequence, we would have to solve a knapsack problem with convex but inseparable utilities, a problem that cannot be solved efficiently (cf. Bretthauer and Shetty, 2002a).

Because we cannot obtain exact solutions to Equations (4.10) and (4.11), we use heuristic approaches to find good solutions to Problem 4.1. Our previous analysis and discussion not only revealed the mathematical problems associated with solving Problem 4.1, but also provided first valuable insights into the major requirements a suitable heuristic policy should meet. Next, we derive these requirements and use them to develop practical recommendations that will be useful when we evaluate existing policies and develop new approaches for solving Problem 4.1.

Clearly, to provide feasible allocations, any policy must consider the supply available in the period under consideration—that is, it must consider $r_t + i_t$.

From the first term in Equation (4.9) ($\frac{-p_l}{X_{l,R+1}}$), we observe that an allocation should depend on a customer's penalty (reduction) per unit of total demand observed at the end of the review period. This observation suggests that a suitable allocation policy should explicitly account for differences in different customers' penalties by prioritizing customers when determining their allocations. However, allocations should also be based on an estimate of the total (uncertain) demand at the end of the review period.

While these requirements are comparatively straightforward, the second term ($\mathbb{1}[D_{l,t} \geq a_{l,t}]$) poses an additional challenge because it accounts for the fact that the value of allocating one additional unit to a customer depends on whether the unit will actually be consumed. Therefore, an estimate of the probability that a customer's demand will be at least as high as its allocation is required for each customer in each period. Thus, the allocation policy should account for the stochasticity of periodic demand.

Finally, the last term in Equation (4.9) ($\mathbb{1}\left[\frac{Y_{l,R+1}}{X_{l,R+1}} \leq \beta_l\right]$) requires an estimate of the probability that the fill-rate target will be reached at the end of the review period. This requirement is the most challenging requirement because it means that the total fulfilled demand ($Y_{l,R}$) at the end of period R must be anticipated; obtaining an estimate for $Y_{l,R}$ is inherently difficult because $Y_{l,R}$ depends on the current

state and all allocation decisions and demand realizations in periods $t, \ldots, R$. Matters are further complicated by the fact that the allocations in individual periods are restricted by the available supply in those periods. Hence, an allocation policy should account for the demand and supply situation in the remaining periods $t+1, \ldots, R$.

We conclude that an approach to determining an allocation policy should account for the supply available in the current period, the customers' total demand $x_{l,t}$ and total fulfilled demand $y_{l,t}$ in the current period, the customers' fill-rate targets β_l, the customers' penalties p_l, the stochasticity of demand in the current period, and the demand/supply situation in remaining periods.

Of course, any heuristic approach to determining an allocation policy will neglect some of these parameters and/or make simplifications in how it takes them into account. Therefore, in evaluating alternative approaches, one must understand which of these parameters can be neglected/relaxed and which are critical to each policy's performance.

The next section presents existing approaches to determining an allocation policy and proposes several new approaches. We analyze and discuss these approaches in terms of the requirements derived from our previous analysis.

4.5 Heuristic Allocation Policies

This section addresses three basic allocation policies: the per-commit approach, which is frequently used in practice, a myopic service-level-based allocation policy (MSLAP), and a myopic penalty-based allocation policy (MPAP) that we adopt from the literature. Our analysis makes clear that these policies meet only a few of the requirements we derived in Section 4.4. Our assumption is that these shortcomings will lead to inferior allocations and excessive penalties at the end of the review period.

In the second part of this section we motivate, define, and discuss four alternative policies—a deterministic allocation policy (DAP), a randomized deterministic allocation policy (RDAP), a myopic stochastic allocation policy (MSAP), and a stochastic time-aggregated allocation policy (STAP)—that rely on fewer simplifications and meet more of the aforementioned requirements. Table 4.2 provides a high-level overview of these policies and how they meet the requirements we identified previously.

Table 4.2: Overview of heuristic allocation policies and their properties.

	Current supply	Current total (filled) demands	Fill-rate targets	Penalties	Current demand stochasticity	Subsequent supply/ demand
Per Commit	✓	–	–	–	–	–
MSLAP	✓	✓	✓	–	–	–
MPAP	✓	✓	✓	✓	–	–
DAP	✓	✓	✓	✓	–	✓
RDAP	✓	✓	✓	✓	(✓)	✓
MSAP	✓	✓	✓	✓	✓	–
STAP	✓	✓	✓	✓	✓	(✓)

"–": requirement not fulfilled, "(✓)": requirement partially fulfilled, "✓": requirement fulfilled.

4.5.1 Basic Allocation Policies

This section describes and discusses basic heuristic allocation policies. We begin with the per-commit approach, which is probably the simplest approach to determining a feasible allocation.

Per Commit

Under the per-commit approach, customers receive their allocations based on their expected share of the total demand (cf. Ball et al., 2004). Definition 4.3 formalizes the approach for our setting.

Definition 4.3 (Per commit allocation)**.** *The per commit allocation $a_{l,t}^{pc}$ to customer l in period t is*

$$a_{l,t}^{pc} = \frac{\mu_{l,t}}{\sum_{m \in \mathcal{L}} \mu_{m,t}} [r_t - i_{t-1}]^+.$$

Definition 4.3 shows that the per-commit approach allocates based on the available supply and the customers' mean demands in the current period. Hence, apart from the available supply in the current period, none of the requirements derived in Section 4.4.2 are met. (See also Table 4.2.) Therefore, we expect low performance from the per-commit approach.

Myopic Service-Level-Based Allocation Policy

Abbasi et al. (2017) analyze several allocation policies with the objective of minimizing deviations from fill-rate targets at the end of the review period, so they do not account for differences in the penalties across customers. In addition, in their setting, the allocation decision is made after the manufacturer knows the demands of all customers. Their results suggest that, for this objective and setting, a myopic policy that minimizes per-period deviations from the fill-rate targets performs well.

We explore whether such a myopic policy also leads to satisfactory results in our setting, that is, when expected total penalties are minimized and allocations are determined before demand is realized, so we adapt Abbasi et al.'s (2017) policy to our setting. A formal definition of this policy, MSLAP, is provided in Definition 4.4. (A linearized Linear Program (LP) formulation that corresponds to Definition 4.4 can be found in Appendix C.2.)

Definition 4.4 (MSLAP)**.** *The myopic service-level-based allocation in period t is*

$$\boldsymbol{a}_t^{mslap} = \underset{\boldsymbol{a}_t \in A_t(i_t)}{\operatorname{argmin}} \max \left\{ \left[\beta_1 - \frac{a_{1,t} + y_{1,t}}{\mu_{1,t} + x_{1,t}} \right]^+, \ldots, \left[\beta_l - \frac{a_{l,t} + y_{l,t}}{\mu_{l,t} + x_{l,t}} \right]^+ \right\}$$

$$\textit{subject to} \qquad a_{l,t} \leq \mu_{l,t} \qquad \forall l \in \mathcal{L} \tag{4.13}$$

MSLAP minimizes the maximum deviation from the fill-rate target across all customers $l \in \mathcal{L}$ separately for each period $t \in \{1, \ldots, R\}$. It determines allocations in a myopic fashion for each period so the deviations from the fill-rate targets are balanced across all customers. This approach is likely to resemble a strategy that a planner would pursue, where the planner observes the current fill rate at the beginning of period t, anticipates the expected demand $\mu_{l,t}$ in period t, and allocates available supply so deviations from the fill-rate targets are expected to be balanced at the end of period t.

MSLAP accounts for the customers' current total demand, total fulfilled demand, and individual fill-rate targets. While MSLAP fulfills more of the requirements than per-commit, it is myopic in that it considers only the current period and does not account for supply and demand in future periods, and it is deterministic and does not account for the stochasticity of demand. More importantly, it does not consider differences in the customers' penalties p_l. We suggest that this policy will lead to (close to) optimal solutions only when penalties are similar across

customers, demand uncertainty is low, and the relationship between demand and supply is the same across the periods.

Myopic Penalty-Based Allocation Policy

We now introduce a straightforward modification of MSLAP that is likely to remedy at least one shortfall of MSLAP. We term this policy myopic penalty-based allocation policy (MPAP). MPAP minimizes penalty-weighted deviations from service-level targets in each period. Like MSLAP, MPAP is still myopic, but it accounts for different penalties p_l. Definition 4.5 formalizes this policy. (See Appendix C.2 for an LP formulation.)

Definition 4.5 (MPAP)**.** *The myopic penalty-based allocation in period t is*

$$\boldsymbol{a}_t^{mpap} = \operatorname*{argmin}_{\boldsymbol{a}_t \in A_t(i_t)} \sum_{l \in \mathcal{L}} p_l \left[\beta_l - \frac{a_{l,t} + y_{l,t}}{\mu_{l,t} + x_{l,t}} \right]^+ .$$

By incorporating the penalties p_l, MPAP meets one more requirement (cf. Table 4.2) than MSLAP does.

The three allocation policies presented in this section are relatively simple and easy to comprehend and implement, which can be considered advantages over more sophisticated and complex allocation policies. However, we expect this advantage to come at the cost of low overall performance in the form of comparatively high overall penalties. In Section 4.5.2, we evaluate whether and under what conditions per-commit, MSLAP, and MPAP lead to satisfactory results and when a manufacturer should resort to more advanced allocation policies.

The next section introduces four advanced allocation policies that promise results that are superior to those of the simple policies described in this section.

4.5.2 Advanced Allocation Policies

Our analysis of the dynamic program in Section 4.4 revealed that computing optimal allocations is feasible only for the terminal period R (Theorem 4.1), not for periods $t \in \{R-1, \ldots, 1\}$. The simple policies described in Section 4.4 lead to feasible allocations, but because they are myopic and deterministic, they fail to meet a number of the requirements we derived in Section 4.4, so we assume they lead to sub-optimal allocations. In this section, we propose alternative policies that

rely on established approximate dynamic programming techniques and are based on "value function approximation" (Powell, 2011).

In value function approximation the value function V_t is replaced with some approximation $\bar{V}_t$ that avoids the "curse of dimensionality" and leads to a simplified optimization problem (cf. Powell, 2011):

$$\bar{\boldsymbol{a}}_t^* = \underset{\boldsymbol{a}_t \in A_t(i_t)}{\operatorname{argmin}} \ \bar{V}_t(\boldsymbol{a}_t)$$

Clearly, a policy's performance hinges on the quality of the approximation of the value function. Here we propose four variants of a value function approximation approach for our setting. First, we adopt "certainty equivalent control" (CEC) (Bertsekas, 2005) and develop a deterministic allocation policy (DAP). Under CEC, a policy is obtained by solving the deterministic equivalent of the problem. Second, we use "randomized linear programming" (RLP) to extend the resulting deterministic problem to include demand uncertainty (Talluri and van Ryzin, 1999). Third, we propose a myopic stochastic allocation policy (MSAP) that uses the terminal period problem from Section 4.4.1 to approximate the value function. This policy can also be interpreted as a specific one-period look-ahead approach. Finally, we extend MSAP to consider multiple periods. We term this the stochastic time-aggregated allocation policy (STAP). Under STAP, the value functions $V_{t+1}, \ldots, V_R$ are aggregated into a single value function $V_{[t+1,R]}$ so we can employ the results from Section 4.4 to obtain an allocation policy.

Figure 4.2 illustrates the original dynamic program (Equations (4.2) and (4.3)) and our advanced (but heuristic) allocation policies that are based on approximate dynamic programming.

Next, we provide a formal definition of these allocation policies.

Deterministic Allocation Policy

The idea of DAP is to replace the stochastic problem with a deterministic problem by assuming that all uncertain quantities realize at their "typical" values (Bertsekas, 2005). For our problem, fixing the random demands $D_{l,t}$ at their means $\mu_{l,t}$ for $l \in \mathcal{L}$ and $t \in \{1, \ldots, R\}$ is an obvious choice. Thus, Problem 4.1 is transformed into a deterministic LP, making it easy to determine allocations $\bar{\boldsymbol{a}}_t^*$ ("offline") for $t \in \{1, \ldots, R\}$ at the beginning of period 1. However, this offline optimization may lead to infeasible solutions in periods $t \in \{2, \ldots, R\}$, depending on the real-

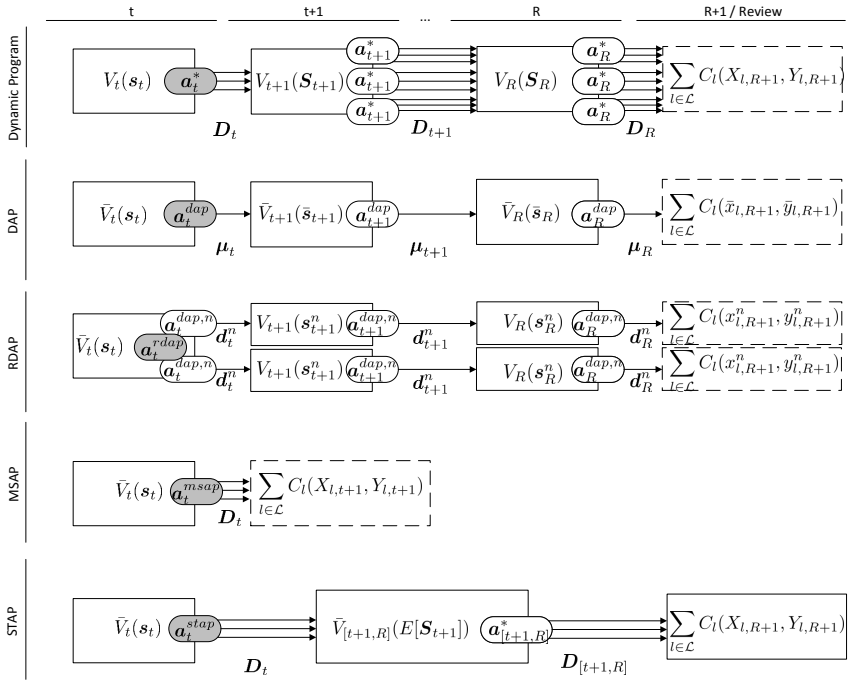


Figure 4.2: Visualization of the dynamic program and the approximated dynamic program approaches for determining an allocation in period t (highlighted in gray).

izations of $\boldsymbol{D}_t$. Therefore, DAP has to be applied "online"; that is, in each period $t \in \{1, \dots, R\}$ optimal allocations $\bar{\boldsymbol{a}}_t^*, \dots, \bar{\boldsymbol{a}}_R^*$ are calculated for the remaining periods $\{t, \dots, R\}$ based on the current state $\boldsymbol{s}_t$, but only $\bar{\boldsymbol{a}}_t^*$ is implemented. We formalize DAP in Definition 4.6.

Definition 4.6 (DAP)**.** *Denote with*

$$\bar{r}_\tau = \left[i_t + \sum_{k=t}^{\tau} r_k - \sum_{k=t}^{\tau-1} \sum_{l \in \mathcal{L}} \mu_{l,k} \right]^+$$

the nominal supply in period τ and let $\bar{\boldsymbol{a}}_t^*, \dots, \bar{\boldsymbol{a}}_R^*$ be the allocations that solve

$$\min_{\bar{a}_t, \dots, \bar{a}_R, (c_l)_{l \in \mathcal{L}}} \quad \sum_{l \in \mathcal{L}} c_l$$

subject to

$$c_l \geq p_l \left(\beta_l - \frac{y_{l,t-1} + \sum_{\tau=t}^{R} \bar{a}_{l,\tau}}{x_{l,t-1} + \sum_{\tau=t}^{R} \mu_{l,\tau}} \right) \qquad \forall l \in \mathcal{L} \tag{4.14}$$

$$c_l \geq 0 \qquad \forall l \in \mathcal{L} \tag{4.15}$$

$$\bar{a}_{l,\tau} \leq \mu_{l,\tau} \qquad \forall l \in \mathcal{L}, \tau \in \{t, ..., R\} \tag{4.16}$$

$$\sum_{l \in \mathcal{L}} \bar{a}_{l,\tau} \leq \bar{r}_\tau \qquad \forall \tau \in \{t, ..., R\} \tag{4.17}$$

$$\bar{a}_{l,\tau} \geq 0 \qquad \forall l \in \mathcal{L}, \tau \in \{t, ..., R\}. \tag{4.18}$$

Then the deterministic allocation in period t is $\boldsymbol{a}_t^{dap} = \bar{\boldsymbol{a}}_t^*$.

The LP in Definition 4.6 uses dummy variables c_l and constraints (4.14) and (4.15) to linearize the non-linear penalty for each customer. (4.16) ensures that no more than expected demand is allocated to each customer, (4.17) limits allocations to available supply, and (4.18) ensures allocations are non-negative.

DAP captures the trade-offs between the customers' penalties and anticipates the supply/demand situation, but it ignores the stochasticity of the problem, as allocations are based only on mean demands. As this policy fulfills more of the requirements of an optimal allocation than per-commit and the myopic policies (MSLAP and MPAP) do, we expect it to perform better than they do and expect relatively small deviations from the optimum when the forecast accuracy is high. We evaluate this policy's performance in Section 4.6.

Randomized Deterministic Allocation Policy

To include stochasticity in the LP of DAP, we use the RLP approach and term the resulting policy RDAP. The RLP approach was developed by Talluri and van Ryzin (1999) to determine optimal bid prices in airline networks. The idea behind the approach is to solve the deterministic LPs repeatedly and separately for different demand realizations sampled from the demand distributions and to then determine the bid prices by averaging the individual bid prices obtained from the dual problem of the LP. For their problem, Talluri and van Ryzin (1999) show that these bid prices are asymptotically optimal. Quante (2009) and Eppler (2015) apply the approach to the problem of allocation planning for customers with differing profitabilities and show numerically that the approach results in good allocations, although the authors provide no performance guarantees.

DAP, as formalized in Definition 4.6, can be extended easily to the RLP concept, as we have only to replace the mean demand $\mu_{l,\tau}$ with randomly generated demand realizations$d_{l,t}^n$ for demand scenarios $n \in \{1, \ldots, N\}$, determine the optimal allocations for each demand scenario n, and average the optimal allocations of the N scenarios to obtain the final allocation. RDAP also has to be applied online. Definition 4.7 formalizes the policy.

Definition 4.7 (RDAP)**.** *Let $a_{l,t}^{dap,n}$ be the allocation of DAP in Definition 4.6 for demand scenario $n \in \{1, ..., N\}$. Then the randomized deterministic allocation for period t to customer l is $a_{l,t}^{rdap} = \frac{1}{N} \sum_{n=1}^{N} a_{l,t}^{dap,n}$.*

As each allocation of DAP used to determine RDAP's allocation is bounded by the demand realization—that is, $a_{l,t}^{dap,n} \leq d_{l,t}^n$ and $\mathbb{E}\, d_{l,t}^n = \mu_{l,t}$—RDAP's allocations are bounded by the mean demand in expectation $\mathbb{E}\, a_{l,t}^{rdap} \leq \mu_{l,t}$. Hence, RDAP's allocations will not typically exceed the corresponding customers' mean demand.

In contrast to Talluri and van Ryzin's (1999) approach, Definition 4.7 uses primal variables to determine allocations. Therefore, we do not claim that our approach retains the original approach's property of asymptotic optimality, and we limit our analysis to the numerical evaluation we perform in Section 4.6. Nonetheless, RDAP incorporates stochasticity in the allocation policy, so it should lead to better solutions than DAP does.

Myopic Stochastic Allocation Policy

MSAP, which can be considered a stochastic version of MPAP, determines allocations for any non-terminal period $t \in \{1, 2, \ldots, R-1\}$ assuming that it is the terminal period (i.e, $t = R$). We formalize the policy in Definition 4.8.

Definition 4.8 (MSAP)**.** *The optimal allocation under MSAP in period t is*

$$\boldsymbol{a}_t^{msap} = \underset{\boldsymbol{a}_t \in A_t(i_t)}{\arg\min}\, E\left[\sum_{l \in \mathcal{L}} C_l(X_{l,t+1}, Y_{l,t+1})\right]. \tag{4.19}$$

The optimal allocation of MSAP in Equation (4.19) can be computed efficiently because Theorem 4.1 applies.

MSAP considers the stochasticity of the individual customers' demand more accurately than RDAP does because it determines allocations based on the demand distributions, rather than on samples of their realizations. In contrast to RDAP, MSAP can lead to allocations that exceed customers' expected demand, which is a clear advantage in cases of high fill-rate targets because these targets typically require allocations that are higher than the mean demand. However, MSAP is myopic, as it does not anticipate the supply and demand in subsequent periods. At this point, we cannot estimate how MSAP performs compared to RDAP, but a detailed performance comparison of the two policies is part of our numerical evaluation in Section 4.6.

Stochastic Time-aggregated Allocation Policy

STAP's underlying rationale is illustrated in Figure 4.2. To deal with Problem 4.1's "curse of dimensionality," STAP converts the $R - t + 1$-period problem into a two-period problem by aggregating periods $[t+1, R]$ into a single period and evaluating the resulting approximated value function $\bar{V}_{[t+1,R]}(\mathbb{E}\, \boldsymbol{S}_{t+1})$.

We denote by $\bar{D}_{l,[t+1,R]}$ the aggregated demand of customer $l \in \mathcal{L}$ in periods $t+1, \ldots, R$, by $\bar{f}_{l,[t+1,R]}$ its pdf, and by $\bar{r}_{[t+1,R]}$ the aggregated supply. The aggregate supply is calculated as $\bar{r}_{[t+1,R]} = \sum_{\tau=t+1}^{R} \left[r_\tau + \left[\bar{i}_{\tau-1}\right]^-\right]^+$, where $\bar{i}_\tau = \bar{i}_{\tau-1} + r_{\tau-1} - \sum_{l \in \mathcal{L}} \mu_{l,\tau-1}$ is the expected inventory position[2]. The expected state $\mathbb{E}\, \boldsymbol{S}_{t+1} = \mathbb{E}\, q_t(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{D}_t)$ is given by the following set of equations:

[2]The aggregated supply is corrected for the expected backlog (the negative part of the inventory position) because backlogged demand in our model is fulfilled before any new demand is fulfilled. Supply that is used to clear backlogged demand does not contribute to the fill rate. To avoid double-counting the backlogged demand, we use the positive part of the available supply.

$$
\begin{aligned}
\bar{x}_{l,t+1} &= x_{l,t} + \mu_{l,t} \\
\bar{y}_{l,t+1}(a_{l,t}) &= y_{l,t} + \int_0^{a_{l,t}} d_{l,t} g_{l,t}(d_{l,t}) \, \mathrm{d}d_{l,t} + a_{l,t}[1 - G_{l,t}(a_{l,t})] \\
\bar{i}_{t+1} &= i_t + r_t - \sum_{l \in \mathcal{L}} \mu_{l,t}.
\end{aligned}
$$

STAP reduces the "curse of dimensionality" in two ways: first, considering only a two-period problem instead of an $R - t + 1$-period problem and by evaluating the value function only at the expected state $\mathbb{E}\, S_{t+1}$ in $t + 1$. The value function for the aggregated period is defined as

$$
\bar{V}_{[t+1,R]}(\mathbb{E}\, S_{t+1}) = \min_{\boldsymbol{a}_{[t+1,R]}} \mathbb{E} \sum_{l \in \mathcal{L}} C_l(\bar{X}_{l,R+1}, \bar{Y}_{l,R+1}) \tag{4.20}
$$

$$
\begin{aligned}
\text{subject to} \qquad \bar{X}_{l,R+1} &= \bar{x}_{l,t+1} + \bar{D}_{l,[t+1,R]} && \forall l \in \mathcal{L} \\
\bar{Y}_{l,R+1} &= \bar{y}_{l,t+1} + \min\{\bar{D}_{l,[t+1,R]}, a_{l,[t+1,R]}\} && \forall l \in \mathcal{L} \\
\sum_{l \in \mathcal{L}} a_{l,[t+1,R]} &\leq \bar{r}_{[t+1,R]} \\
a_{l,[t+1,R]} &\geq 0 && \forall l \in \mathcal{L}.
\end{aligned}
$$

We are now able to provide a formal definition of STAP.

Definition 4.9 (STAP)**.** *The optimal allocation under STAP in period t is*

$$
\boldsymbol{a}_t^{stap} = \underset{\boldsymbol{a}_t \in A_t(i_t)}{\arg\min} \, \bar{V}_{[t+1,R]}(\mathbb{E}\, u(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{D}_t)) \tag{4.21}
$$

Next, we show how to solve (4.21). As a first step, we determine the derivative of the approximated value function in Proposition 4.4.

Proposition 4.4 (Derivative of the approximated value function). *Denote by $a^*_{l,[t+1,R]}$ the optimal solution for the aggregated period (Equation (4.20)). Then the approximated marginal penalty of allocation $a_{l,t}$ to customer l in period t is*

$$\frac{\mathrm{d}}{\mathrm{d}a_{l,t}} \bar{V}_{[t+1,R]}(\mathbb{E}\, \boldsymbol{S}_{t+1}) = -\bar{\lambda}_{l,[t+1,R]}(a_{l,t})$$

$$= \begin{cases} -p_l(1 - F_{l,t}(a_{l,t})) \int_{d_{max,l}}^{\infty} \frac{\tilde{f}_{l,[t+1,R]}(d_{l,[t+1,R]})}{\bar{x}_{l,t+1}+d_{l,[t+1,R]}} \,\mathrm{d}d_{l,[t+1,R]} & \text{if } \bar{y}_{l,t+1}/\bar{x}_{l,t+1} \geq \beta_l \\ -p_l(1 - F_{l,t}(a_{l,t})) \Bigg(\int_0^{d_{min,l}} \frac{\tilde{f}_{l,[t+1,R]}(d_{l,[t+1,R]})}{\bar{x}_{l,t+1}+d_{l,[t+1,R]}} \,\mathrm{d}d_{l,[t+1,R]} & \\ \qquad + \int_{d_{max,l}}^{\infty} \frac{\tilde{f}_{l,[t+1,R]}(d_{l,[t+1,R]})}{\bar{x}_{l,t+1}+d_{l,[t+1,R]}} \,\mathrm{d}d_{l,[t+1,R]} \Bigg) & \text{if } a^*_{l,[t+1,R]} > d_{min,l} \\ -p_l(1 - F_{l,t}(a_{l,t})) \int_0^{\infty} \frac{\tilde{f}_{l,[t+1,R]}(d_{l,[t+1,R]})}{\bar{x}_{l,t+1}+d_{l,[t+1,R]}} \,\mathrm{d}d_{l,[t+1,R]} & \text{else,} \end{cases}$$

where $d_{min,l} = \frac{\beta_l \bar{x}_{l,t+1} - \bar{y}_{l,t+1}}{1-\beta_l}$ *and* $d_{max,l} = \frac{\bar{y}_{l,t+1} + a^*_{l,[t+1,R]}}{\beta_l} - \bar{x}_{l,t+1}$.

Based on our results in Theorems 4.1 and 4.2, we can derive an optimal solution to Equation (4.21).

Proposition 4.5 (Optimal solution for STAP). $\boldsymbol{a}_t^{stap}$ *is the optimal solution of STAP if and only if there exists a value of $\lambda > 0$, such that, using*

$$A_\lambda = \left\{ l \mid l \in \mathcal{L}, \lambda \geq \bar{\lambda}_{l,[t+1,R]}(a_{l,t}^{stap}) \right\} \tag{4.22}$$

the following hold

$$\begin{aligned} \lambda &= \bar{\lambda}_{l,[t+1,R]}(a_{l,t}^{stap}) && \text{for all } l \in \mathcal{L} \backslash A_\lambda \\ a_{l,t}^{stap} &= 0 && \text{for all } l \in A_\lambda. \end{aligned}$$

Proposition 4.5 ensures that we can apply standard gradient-based non-linear solution techniques to find an optimal solution to Equation (4.21). Because Equation (4.22) depends on $\boldsymbol{a}_t$ (via the expected state), the resulting convex knapsack is not separable, so finding the optimal solution is more involved than it is for the allocation problem in period R (Theorem 4.1).

STAP considers all of the requirements we derived in Section 4.4.2, including aggregated supply and demand in subsequent periods, which MSAP does not. Hence, we expect STAP to lead to better performance than MSAP.

4.6 Numerical Evaluation

This section presents the results of several numerical experiments we carried out to evaluate the performance of the allocation policies presented in Section 4.5. The purpose of these experiments is not only to quantify and compare the policies' performance but also to assess the importance of the requirements we derived in Section 4.4. Section 4.6.1 explains our experimental setup and the individual experiments we carried out, while Section 4.6.2 describes our simulation environment and the performance measures we use to evaluate the policies. Sections 4.6.3 to 4.6.7 present the results obtained from our numerical experiments. Finally, Section 4.6.8 summarizes our results and points out which requirements a "good" allocation policy should meet.

4.6.1 Experimental Setup

To analyze the performance of the allocation policies we presented in Section 4.5 and to assess the importance of the requirements of an optimal allocation policy (Table 4.2 in Section 4.4), we carried out a number of experiments in which we systematically varied the allocation policies' input parameters. We structured our experiments according to the requirements described in Table 4.2.

First, we address the importance of incorporating into the allocation decision information about customers' current total demands and their total fulfilled demands into the allocation decision. To exclude confounding effects from other requirements we conduct a first set of analyses for customers that have identical demand distributions, fill-rate targets, and penalties. Next, we vary the customers' fill-rate targets of the customers in order to induce fill-rate heterogeneity determine whether and at what level of heterogeneity it becomes important to incorporate the customers' fill-rate targets into the allocation policy (Section 4.6.4). Then, in Section 4.6.5, we vary the customers' penalties to determine how differences in fill-rates and penalties jointly impact the allocation policies' performance and whether and when it is beneficial to consider the differences in penalties. In Section 4.6.6 we vary the coefficient of variation (CV) of the customers' demand distributions to determine at what level of uncertainty this information must be incorporated into the allocation policy. Finally, we determine the importance of anticipating the future supply and demand by adding a trend to the demand, as discussed in Section 4.6.7, while keeping the per-period supply constant. The results of this

Table 4.3: Parameterization of the numerical experiments.

Parameter	Homog. customers	Het. fill-rates, homog. penalties	Het. fill-rates, het. penalties	Demand uncertainty	Trend
Average demand	10	10	10	10	10
CV	0.3	0.3	0.3	$0.1, \ldots, 0.5$	0.3
Customers	3	3	3	3	3
Penalty parameter	0	0	$-0.5, -0.25, \ldots, 0.5$	0.5	0.5
Fill-rate targets	$\begin{pmatrix} 0.965 \\ 0.965 \\ 0.965 \end{pmatrix}$	$\begin{pmatrix} 0.965 \ldots 0.88 \\ 0.965 \\ 0.965 \ldots 0.995 \end{pmatrix}$	$\begin{pmatrix} 0.88 \\ 0.965 \\ 0.995 \end{pmatrix}$	$\begin{pmatrix} 0.88 \\ 0.965 \\ 0.995 \end{pmatrix}$	$\begin{pmatrix} 0.88 \\ 0.965 \\ 0.995 \end{pmatrix}$
Review horizon	10	10	10	10	10
Safety buffer	25%	25%	25%	$0.7\%, \ldots, 56.7\%$	25%
Trend	0	0	0	0	$-0.4, \ldots, 0.4$

analysis allows us to identify the importance of demand/supply anticipation by comparing the myopic policies with the multi-period (forward-looking) policies.

The parameter values for our analyses are shown in Table 4.3. All of our experiments assume that per-period supply is constant and equal to the average expected demand per period (which is 30 in all of our experiments). We perceive this assumption as realistic for the purpose of our analysis because it means that the manufacturer produces according to its demand forecast and observes neither high overall scarcity nor excessive supply, at least on an aggregate level. We also assume that the manufacturer has some safety buffer to deal with demand uncertainty. In our experiments we implement the safety buffer as the inventory that is available at the beginning of $t = 1$. The first three experiments fix the safety buffer to 25 percent of the mean demand (7.5 units) and set the review horizon to $R = 10$. Each experiment is repeated $M = 100$ times to ensure stable numerical results.

4.6.2 Simulation Environment

To carry out our experiments, we create a simulation environment in Python, use Gurobi to solve the LPs, and use NLopt to solve the non-linear optimization problems for MSAP and STAP. Figure 4.3 provides a high-level overview of the simulation environment.

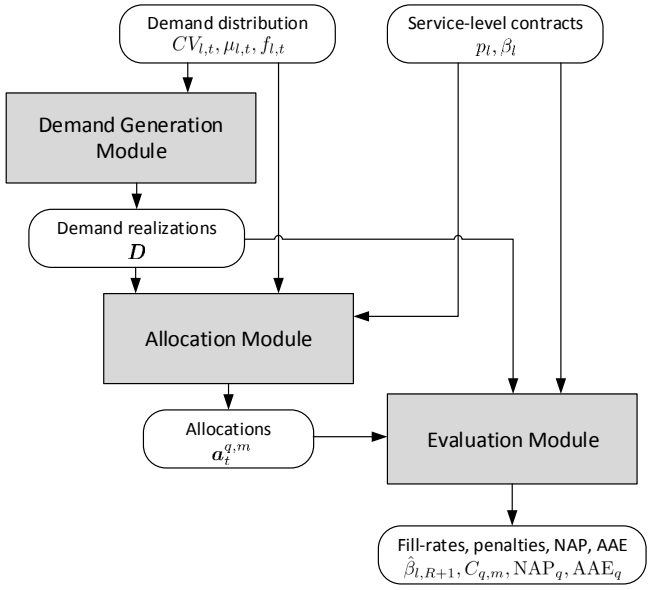


Figure 4.3: Overview of the simulation environment.

Our simulations are performed in three consecutive steps: First, we generate demand realizations in the *demand generation module*, which takes as inputs the mean $\mu_{l,t}$ and the coefficient of variation CV_l of the demand distributions $f_{l,t}$ (for all $l \in \mathcal{L}$). We assume $f_{l,t}$ to be iid and that it follows a normal distribution that is truncated at zero. The demand-generation module draws samples from $f_{l,t}$ for each period t and customer l and outputs a demand matrix $\boldsymbol{D}$ with dimension $|\mathcal{L}| \times R \times M$, where M is the number of repetitions. To reduce the variance in our simulation results, we employ the antithetic variates technique (Ross, 2006) which is commonly used in Monte Carlo simulations.

The *allocation module* takes as input the demand matrix $\boldsymbol{D}$, the mean $\mu_{l,t}$, and the coefficient of variation CV_l of the demand distributions $f_{l,t}$, and the parameters of the service-level contracts—that is, the fill-rate targets β_l and the penalties p_l for customers $l \in \mathcal{L}$, and outputs allocations $\boldsymbol{a}_1^{q,m}, \ldots, \boldsymbol{a}_R^{q,m}$ for each demand instance m and allocation policy q. The simulation module contains implementations of the heuristic allocation policies described in Section 4.5: per-commit, MSLAP, MPAP,

DAP, RDAP[3], MSAP, and STAP. Because the deterministic allocation approaches and RDAP do not allocate more than the mean demand to each customer, any supply that exceeds the customers' total mean demand remains unallocated. Clearly, it is not optimal to retain unallocated supply when demand is uncertain, so we account for this problem in our experiments by allocating any remaining supply on a per-commit basis.

In addition to the allocation policies described in Section 4.5, we implement an ex-post optimization that solves the problem defined in Definition 4.6 for realized demands $d_{l,t}$ ($l \in \mathcal{L}$, $t = 1, \ldots, R$) instead of expected demands $\mu_{l,t}$. The ex-post optimization provides a theoretical upper bound on the performance that we use for evaluation purposes.

The evaluation module takes as input the allocations $\boldsymbol{a}_1^{q,m}, \ldots, \boldsymbol{a}_R^{q,m}$ from the allocation module and the parameters β_l and p_l from the service-level contracts. The module first calculates the achieved fill-rate $\hat{\beta}_{l,R+1}^{q,m}$ for each customer group l, allocation policy q and demand realization m. Based on the values of these output parameters, the evaluation module then calculates the associated total penalty costs $C_{q,m}$. To compare the allocation policies, we use "normalized additional penalty" (NAP) as our main performance measure. The NAP is the absolute difference in penalties between the allocation policy under consideration and the ex-post optimization, normalized by the maximum penalty. Normalizing the NAP makes the measure more robust towards scaling of fill-rate targets and penalties. Note, that because we normalize by the maximum possible penalty (resulting from fulfilling no customer demands at all), we will typically observe very small values for the NAP. Definition 4.10 formalizes this measure.

Definition 4.10 (NAP)**.** *The NAP of allocation policy q is*

$$NAP_q = \frac{1}{M} \sum_{m=1}^{M} \frac{C_{q,m} - C_{expost,m}}{\sum_{l \in \mathcal{L}} \beta_l \cdot p_l}, \tag{4.23}$$

where $C_{expost,m}$ is the penalty of the ex-post optimization for demand realization m.

To explain the allocation policies' performance differences, we introduce a secondary performance measure that we term "average allocation efficiency" (AAE). Definition 4.11 formalizes this measure.

[3]For RDAP, we sample twenty realizations from the demand distributions.

Definition 4.11 (AAE). *The AAE of allocation policy q is*

$$AAE_q = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{|T_m|}\sum_{t\in T_m}\frac{\sum_{l\in\mathcal{L}}\min\{d_{l,t}^m, a_{l,t}^{q,m}\}}{r_t + i_t^m},$$

where $T_m = \{t \in \{1,\dots,R\} \mid r_t + i_t^m > 0\}$ is the set of periods for demand realization m in which the available supply is positive.

The AAE averages how much of the allocated supply was consumed relative to total supply in each period of the planning horizon. It takes a value of 1 for perfect allocation efficiency—that is, when each unit of allocated supply was consumed by the corresponding customer. Values that are less than 1 indicate the extent to which supply was misallocated—that is, allocated to a customer, but not consumed, because the realized demand was lower than the allocation.

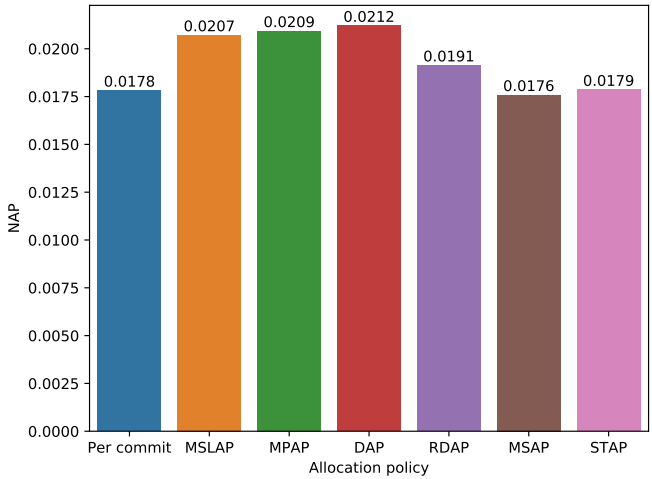
In Definition 4.11 we divide the fulfilled demand per customer and period by the available supply in the corresponding period. We introduce the set T_m to avoid division by zero in periods with no available supply.

4.6.3 Homogeneous Customers

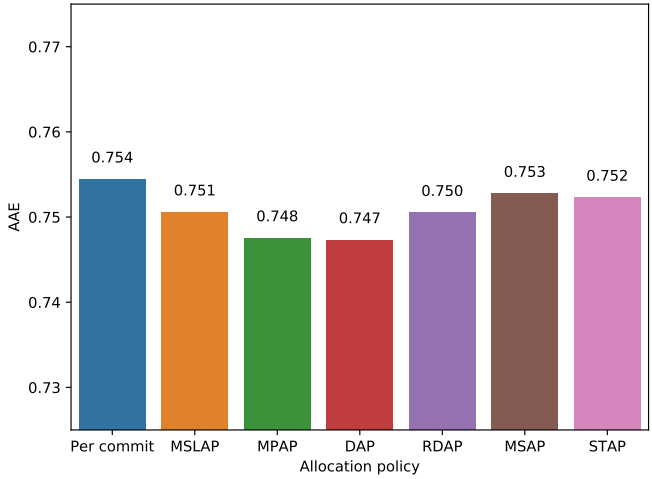
In this experiment we analyze the case of homogeneous customers, so all customers have the same fill-rate targets, penalties, and demand distributions. From a practical point of view, this setting is of little interest because the service-level contracts manufacturers offer differ. However, assuming homogeneous customers allows us to determine the importance of accounting for the system's current state when allocations are determined. Under homogeneous customers, differences in the allocations $a_{l,t}$ (for $t = 2,\dots,R$) to customers $l \in \mathcal{L}$ should be based only on the current state s_t of the system, which is defined by the realized total demands x_t, the fulfilled demands y_t, and the inventory position i_t. Thus, differences in the performance of the allocation policies should, at least in theory, be attributable to how well they account for the system's current state.

The setting we consider in this experiment is similar to that of Abbasi et al. (2017), although we determine allocations before demands realize, while inAbbasi et al. (2017) the planner observes the demand before allocating supply to the customers.

Figure 4.4a shows the NAPs of the allocation policies for homogeneous customers. MSAP has the highest performance, with a NAP of only 0.0176, but it is



(a) NAP



(b) AAE

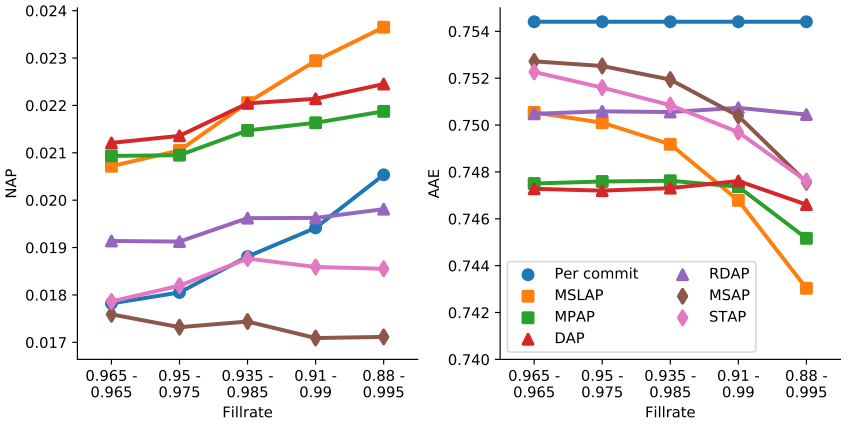
Figure 4.4: NAP and AAE of the allocation policies for homogeneous customers.

closely followed by per-commit and STAP, while DAP's performance is the lowest, with a NAP of 0.0212. The differences in the NAPs appear to be small because we normalize using the maximum penalties that would be incurred if zero demand was fulfilled. To put the values into perspective, in this experiment MSAP's (DAP's) penalties average 27 percent (33%) higher than the penalties that result from the ex-post optimization. Hence, the performance differences across the allocation policies can be considered substantial, even in this setting with homogeneous customers.

Both MSAP and STAP account for the current state and adjust their allocations accordingly, while per-commit allocates based only on the customers' mean demands. It is somewhat counterintuitive that "simple" per-commit performs almost as well as the more sophisticated MSAP and STAP, but because fill-rates are calculated over ten periods and demand fluctuates only moderately (with a CV of 0.3), the realized fill rates at the end of the review period do not vary substantially and are typically lower than the fill-rate target of 0.965 for all customers. Because penalties are the same across all customers, how negative deviations from the fill-rate targets are distributed across customers is irrelevant. Therefore, an allocation policy must avoid allocations that are not consumed, and over-fulfillment of fill-rate targets—recall that there are no bonuses for a realized fill rate that is higher than the target fill rate. In this respect, per-commit performs relatively well because it leads to the same (rather conservative) allocations to each customer in each period based on the mean demand. The almost identical performances of per-commit, MSAP, and STAP indicate that reacting to customers' current total demand and total fulfilled demand does not provide an advantage in our setting.

The detrimental performance of MPAP and DAP can be explained based on their objective functions (cf. Definition 4.5), as when all penalties and service levels are identical, the policies prioritize customers with the lowest total demand and allocate sequentially in order of increasing total demand. As Figure 4.4b shows, this prioritization decreases the allocation efficiency. The somewhat better performance of RDAP can be attributed to its accounting for the stochasticity of demand, and, thus, allocating sequentially, giving it a higher AAE.

MSLAP's low performance is more difficult to explain. Because the policy equalizes the deviations from the fill-rate targets, it does not prioritize sequentially like MPAP and DAP do. Hence, MSLAP's AAE is higher than those of MPAP and DAP. However, in situations of supply scarcity, the approach allocates more to the customer with the lowest current fill-rate, regardless of the probability that



(a) NAP **(b)** AAE

Figure 4.5: NAP and AAE of the allocation policies for various fill-rate targets.

allocations to this customer will actually be consumed. This approach to allocation is less efficient in supply situations in which efficient use of scarce supply is the most important consideration. As a direct consequence, MSLAP leads to a comparatively low AAE and a high NAP. As a consequence, our results differ structurally from those of Abbasi et al. (2017), who find that a myopic deterministic policy that is similar to MSLAP performs well. MSLAP's aforementioned issues occur only when the allocation decision is made before demand is known, so we conclude that, in our setting, the stochasticity of demand must be accounted for. We address this issue in more detail in Section 4.6.6.

The results of this initial experiment suggest that accounting for customers' current total demand and total fulfilled demand is not crucial. Per-commit, the only allocation policy that neglects this information, performs well when customers are homogeneous. More important, our results indicate that, when supply is scarce, avoiding allocations that are not consumed is an important aspect of an allocation policy's performance. Our subsequent analyses shed more light on these preliminary findings.

4.6.4 Heterogeneous Fill-rate Targets and Homogeneous Penalties

Our second experiment is conducted to determine the importance of an allocation policy's accounting for differences in fill-rate targets. To this end, we gradually increase heterogeneity in customers' fill-rate targets and evaluate the allocation policies' resulting performance. More specifically, we increase the fill-rate target of one customer and decrease the fill-rate target of another customer so the overall supply required to meet the fill-rate targets remains the same in all instances.[4]

Figure 4.5 plots the allocation policies' NAPs and AAEs for increasing differences in fill rates. We observe that MSAP's NAP and AAE decrease with increasing heterogeneity—that is, the approach's performance increases while its allocation efficiency decreases. On the other hand, per-commit shows a strongly increasing NAP, while its AAE remains constant.

The underlying effects of the policies change in performance cannot be explained easily based on the policies' AAEs (Figure 4.5b): Per-commit leads to the same (high) AAE in all instances because its allocations are independent of the fill-rate targets, but MSAP's AAE decreases as the fill-rate heterogeneity increases because, as it prioritizes customers with higher fill-rate targets, its allocation efficiency decreases. In short, the higher allocations to customers with higher fill-rates lead to more instances in which the allocated supply is not consumed. This does not, however, translate into a lower overall performance. To explain the underlying dynamics, Figure 4.6 shows the average fill-rates and the corresponding fill-rate targets for each customer. Per-commit achieves a fill rate for Customer 1 that is higher than the target fill-rate but does not meet the fill-rate targets of Customers 2 and 3. In contrast, MSAP avoids over-allocating supply to Customer 1 and achieves higher fill rates for Customers 2 and 3, resulting in lower overall penalties than per-commit sees, even though MSAP's AAE is lower.

Although MSAP is a completely myopic policy, it outperforms STAP, which has a similar underlying logic but incorporates (aggregated) information about future supply and demand. The results, presented in Figure 4.6, shed light on the performance differences between MSAP and STAP. Both policies lead to almost identical AAEs (Figure 4.5b) and to the same average fill rates for Customer 3 (Figure 4.6), but STAP over-achieves Customer 1's fill-rate target, thereby wasting some of the supply, not because it is not consumed but because it does not reduce

[4]Based on a single-period model, we choose the fill-rate targets such that 37.5 units of available supply (30 units + 25% safety buffer) are sufficient to fulfill the fill-rate targets in expectation.

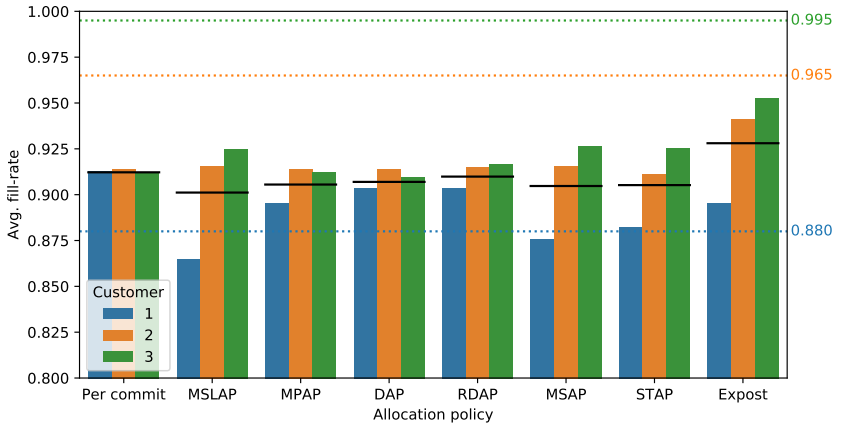


Figure 4.6: Average fill-rates and corresponding fill-rate targets (dotted lines) for each customer and allocation policy for fill-rate targets between 0.88 and 0.995. (Black lines indicate the overall fill-rate.)

the penalty. In contrast, MSAP allocates, on average, less to Customer 1 and more to Customer 2, so it incurs substantially lower overall penalties than STAP does.

It appears that STAP's ability to account for future supply and demand does not translate into higher performance compared to a myopic stochastic policy (i.e., MSAP). On the contrary, we observe—at least in this particular setting—a negative impact on STAP's performance relative to MSAP. The key difference between MSAP and STAP is that, because of its myopic nature, MSAP will never allocate more supply to a customer than what is required to achieve the customer's fill-rate target in the current period. However, STAP may allocate more than this quantity in anticipation of future demand and supply—that is, it may assign a positive marginal profit to overachieving a customer's fill rate target in the current period, which is why, in our experiments, STAP allocates more to Customer 1 than MSAP does. However, STAP's approximation of future supply and demand is sufficiently inaccurate to lead to excessive allocations to Customer 1 and too-low allocations to Customer 2, so MSAP performs better than MSAP does.

MSLAP minimizes the maximum fill-rate deviation, so it tries to equalize the deviations from the fill-rate targets, a logic that seems particularly suitable when fill-rate targets differ. However, MSLAP's AAE decreases significantly as fill-rate targets become more heterogeneous, leading to the lowest overall average fill rate (Figure 4.6). Because of their high fill-rate targets, Customers 2 and 3 receive high

allocations that are frequently not consumed, while Customer 1, who has a higher probability of consuming its (lower) allocations, receives too little supply and experiences fill-rates that average substantially below its (already low) fill-rate target. By ignoring whether an allocated unit of supply is likely to be consumed and because it prioritizes customers with high fill-rate targets, MSLAP over-allocates and under-allocates at the same time, leading to the highest overall penalties when fill-rate targets differ across customers. MSLAP's poor performance is somewhat surprising, as the policy's presumed strength—its ability to prioritize according to service-level targets—turns out to be a disadvantage that leads to poor performance when fill-rate heterogeneity is high.

For MPAP, DAP, and RDAP, the fill-rate targets affect only the maximum allocation to each customer, so these policies suffer from the same problems we identified for the case of homogeneous customers. The results presented in Figure 4.6 show that average fill rates are almost identical across all customers for these policies; only Customer 1's average fill rates are slightly lower, but they are still above the customer's fill-rate target. Unsurprisingly, the three policies cannot prioritize adequately based on differences in the fill-rate targets, so they allocate too much to customers with low fill-rate targets and too little to customers with higher fill-rate targets.

Again, we observe that RDAP has a higher AAE than its deterministic counterpart, DAP, because RDAP accounts for the stochasticity of demand, so RDAP performs better than DAP.

The results of this experiment demonstrate that heterogeneous fill-rate targets should be accounted for. MSAP and STAP both fulfill this requirement, leading to the best performance among the allocation policies. In line with the results from our first experiment, we find that it is important to account for the probability that allocated supply will be consumed when determining customer allocations. Deterministic approaches lead to particularly unfavorable results when fill-rate targets are heterogeneous and customer demand is uncertain. In light of the performance of MSAP and STAP, we find no evidence that the anticipation of future supply and demand is an important prerequisite for an allocation policy. We explore this open issue further in Section 4.6.7.

4.6.5 Heterogeneous Fill-rate Targets and Heterogeneous Penalties

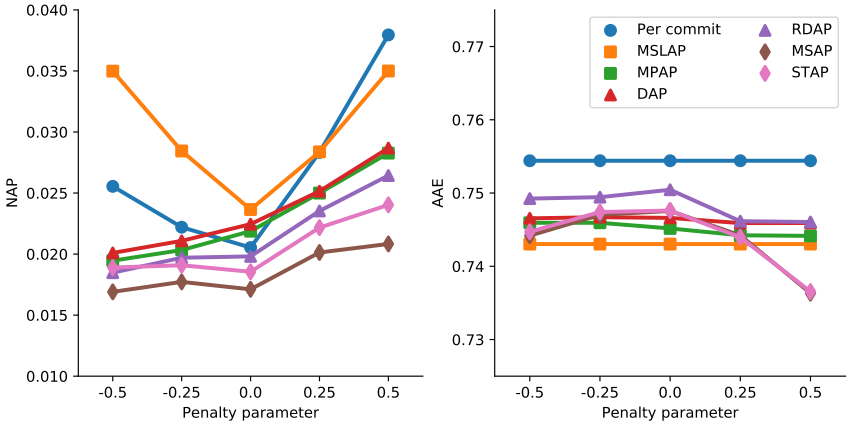
Our third experiment examines the effect of heterogeneity in fill-rate targets *and* penalties, an examination that is particularly useful from a practical point of view because manufacturers are likely to negotiate service-level contracts in which both fill-rate targets and penalties vary across customers.

To determine the individual effects of each type of heterogeneity, we fix a high fill-rate heterogeneity (0.88–0.995 in the experiment described in Section 4.6.4) and increase the heterogeneity in terms of penalties. We do not report the results of an experiment in which we set the fill-rate heterogeneity to zero and varied only the penalty's heterogeneity because it yielded structural results similar to those presented in Section 4.6.4.

A priori, we do not know how fill-rates and penalties for individual customers will be reflected in service-level contracts. It is conceivable that high fill-rates will be associated with high penalties, but we cannot rule out the opposite relationship, so we explore both positive and negative relationships between fill-rates and penalties for individual customers. We introduce a penalty parameter ρ that measure the additional penalty associated with the customer that has the highest fill-rate target (Customer 3) relative to the average penalty (assigned to Customer 2) (i.e., $p_3 = (1+\rho)p_2$, $p_1 = (1-\rho)p_2$). A penalty parameter $\rho > 0$ ($\rho < 0$) indicates that the customer with the highest (lowest) fill-rate target has the highest (lowest) penalty, while the customer with the lowest (highest) fill-rate target has the lowest (highest) penalty. As Table 4.3 shows, we vary ρ from -0.5 to +0.5 in increments of 0.25.

Figure 4.7 plots the NAPs and AAEs of the allocation policies at different values of ρ. The NAPs of per-commit and MSLAP increase with the penalties' heterogeneity—that is, the more ρ differs from 0. The NAPs of MPAP, DAP, and RDAP increase as ρ increases, while the NAPs of MSAP and STAP increase only for $\rho > 0$.

The results presented in Figure 4.7 highlight that MSAP outperforms all other allocation methods and also appears to be less sensitive to an increase in penalties' heterogeneity than the other allocation policies are. In contrast, per-commit and MSLAP are highly sensitive to differences in customers' penalties, leading to steep increases in the NAP at higher levels of heterogeneity. The underlying effects are similar to what we observed in the experiment in which we studied the effect of



(a) NAP **(b)** AAE

Figure 4.7: NAP and AAE of the allocation policies for various penalty parameters.

fill-rate heterogeneity, and they can again be explained based on the AAE (Figure 4.7b): While per-commit and MSLAP retain their constant AAEs, as allocations are independent of the penalties, MSAP (and STAP) have lower allocation efficiency at higher levels of penalty heterogeneity; because they prioritize customers with higher penalties, supply is allocated but not consumed more often. Clearly, this effect is more pronounced at positive values of ρ, where high fill-rate targets coincide with high penalties. The effects are asymmetric for positive and negative values of ρ: while high penalties and high fill-rate targets ($\rho > 0$) both warrant prioritization—that is, larger differences in the allocations—high penalties and low fill-rate targets ($\rho < 0$) require less prioritization and more balanced allocations. As a result, the NAP increases only for ρ values that are greater than zero.

The NAPs of MPAP, DAP, and RDAP indicate that their performance improves as the penalties of the customers with the lowest fill-rate targets increase. This result is intuitive, as the approaches' allocating only up to the mean demand is less harmful when the customer with the highest penalty has a low fill-rate target.

Structurally, the results of this experiment support our conjecture that an allocation policy should account for differences in customers' penalties (Section 4.4.2). While this result is intuitive, penalty heterogeneity has a significant impact on allocation policies' performance. When penalties and fill-rate targets differ sub-

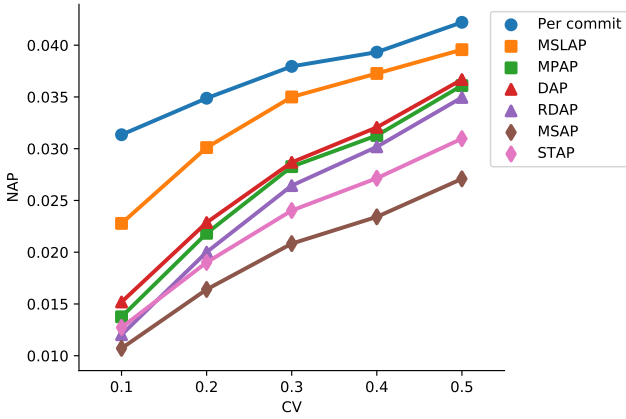


Figure 4.8: NAP of the allocation policies under varying CV.

stantially across customers, choosing the wrong allocation policy (e.g., per-commit or MSLAP) can have a substantially negative impact on performance.

4.6.6 Demand Uncertainty

The results of our first and second experiments (Sections 4.6.3 and 4.6.4) suggest that an allocation policy should consider whether the supply allocated to a particular customer is likely to be consumed. These results should favor MSAP, STAP, and RDAP over their deterministic competitors. The experiment described in this section explores the performance effect of incorporating the consumption probability into the logic of the allocation policy. To do so, we vary the CVs of the customers' demand distributions (which was fixed at 0.3 in our other experiments) and assess how doing so impacts the individual allocation policies' performance.

Our analysis is based on the scenario with a penalty parameter of $\rho = 0.5$. To ensure a fair comparison of the scenarios, we adjust the safety buffer accordingly (0.7% for a CV of 0.1 and up to 56.7% for a CV of 0.5).

The results, shown in Figure 4.8, indicate that the performance of all allocation policies decreases with increasing CV. This result is not unexpected: as the demand's stochasticity increases, prioritizing the correct customers becomes more difficult and NAPs increase.

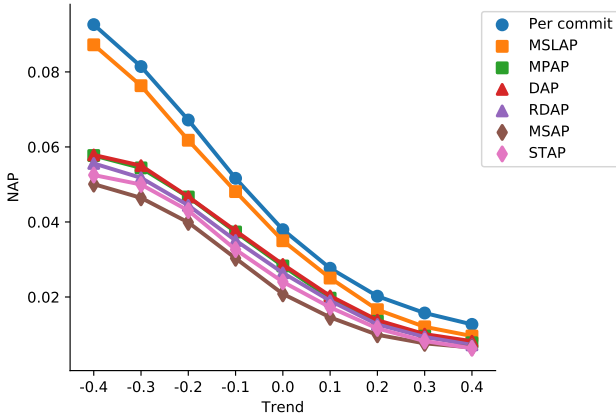


Figure 4.9: NAP of the allocation policies for various trend parameters.

As expected, the performances of the deterministic allocation policies, MDAP and DAP, decrease more in the CV than do the performances of the stochastic allocation policies, MSAP and STAP. Consequently, MSAP consistently outperforms its competitors.

In contrast to our conjecture, RDAP has the strongest sensitivity to an increase in the CV. While the policy outperforms STAP at a CV of 0.1, its performance is significantly lower at a CV of 0.5 because of RDAP's limitation to allocate only up to the customers' mean demand. When the CV is low, the allocations required to achieve a certain fill-rate target decrease; hence, this disadvantage becomes less severe for small values of the CV, and RDAP's relative performance improves. An unreported experiment shows that, for the same reasons, decreasing the fill-rate targets also improves RDAP's relative performance, although its penalty is always higher than that of MSAP.

This experiment supports our conjecture that the stochasticity of customer demand should be accounted for. While the importance of considering this stochasticity increases as stochasticity increases, our results suggest that it is an important performance driver even at very low levels of uncertainty.

4.6.7 Demand Trend

To determine the importance of anticipating the future demand and supply situations, we perform an experiment in which we keep per-period supply constant and add a linear trend to customer demands. We introduce the trend parameter $\theta = \frac{\mu_{l,R}}{\frac{1}{R}\sum_{t=1}^{R}\mu_{l,t}} - 1$, which measures the additional demand in the last period compared to the average demand. Hence, $\theta > 0$ corresponds to increasing demand and $\theta < 0$ to decreasing demand. Because supply remains constant, $\theta > 0$ implies ample supply in the first periods and scarce supply in later periods, while $\theta < 0$ implies the opposite.

Figure 4.9 plots the AAEs of the allocation policies for various values of θ. The AAEs are decreasing in the trend parameter θ, as, for a negative trend, supply is scarce in the beginning and this scarcity is carried over to future periods by the backlog, so supply is scarce in many periods, and the differences among the policies are most pronounced. Under a positive trend, there is ample supply in the first periods, and because unused supply is carried over to future periods, most periods have sufficient supply, so the policies differ only slightly in terms of performance.

The results in Section 4.6.5 suggest that STAP's ability to anticipate future demand and supply situations do not lead to performance benefits because of inaccurate forecasts resulting from the aggregation of demand and supply across all future periods. We now observe that this result not only holds for demand with a constant mean but also for negative trends and for positive trends up to $\theta = 0.3$. Only at the high value of $\theta \geq 0.4$ does STAP benefit from its ability to anticipate future demand and supply situations.

4.6.8 Summary of Numerical Results

In Section 4.4 we developed a set of requirements that an allocation policy under service-level contracts should meet. Our conjecture was that, among the allocation policies we introduced in Section 4.5, STAP is the only one to meet all of the requirements, so it will outperform its competitors, particularly because of its ability to anticipate future demand and supply. This feature is particularly important from a practical point of view: A decision-maker (planner) who is responsible for meeting multiple heterogeneous customers' service-level targets faces a complex stochastic dynamic problem, so using a (myopic) heuristic to determine allocations

period by period without anticipating what may happen in the future does not seem to be an appropriate strategy. In contrast, our numerical results suggest that a myopic stochastic policy (i.e., MSAP) will have satisfactory results under most conditions, or at least more satisfactory than those of the six other policies considered in our study. However, we must be careful not to draw hasty conclusions about the importance of considering future supply and demand. STAP's ability to look ahead rests on a coarse approximation of the system's future state, and we can expect that a more refined approximation leads to better results. However, as we show in Section 4.4, the large state space of the original dynamic program renders a more accurate approximation difficult, if not impossible.

Two of the requirements we derived in Section 4.4 emerged from our numerical analysis to be of particular importance: accounting for heterogeneous penalties and incorporating demand stochasticity. Correctly prioritizing customer allocations based on penalties while avoiding unused allocations and over-achievement of fill-rate targets appear to be most important. MSAP and STAP meet these requirements, and all of the policies that ignore one or both of these requirements—per-commit, MSLAP, MPAP, and DAP—have substantially lower performance in any realistic setting. RDAP, the approach based on randomized linear programming, also accounts for heterogeneity in penalties and, at least to an extent, for the stochasticity of demand, but because of its inherent logic, it does not (in expectation) allocate more than the mean demand to each customer. Therefore, RDAP's ability to allocate "correctly" based on heterogeneous penalties and demand uncertainty is restricted, and the policy typically performs considerably less well than MSAP and STAP do.

How the policies account for differences in customers' fill-rate targets turns out to be less important than prioritizing customer allocations based on penalties and avoiding unused allocations and over-achievement of fill-rate targets. The results presented in Section 4.6.4 suggest that increasing heterogeneity in fill-rate targets affects the policies' performance. Contrary to our intuition, however, this effect is not strong, and it is MSAP that deals best with fill-rate heterogeneity. Heterogeneous fill-rate targets and heterogeneous penalties (Section 4.6.5) have interaction effects. Performance differences between the allocation methods are largest when fill-rate targets differ most and when high penalties are incurred for customers with high fill-rate targets. In these instances, the gaps between MSAP and its competitors are largest.

Overall, our numerical results are in line with the findings of Abbasi et al. (2017) and Chen and Thomas (2018), who assumed that allocation decisions can be made after demand is known. Of course, considering stochasticity of demand is not important when demand uncertainty is resolved before customer allocations are determined, which is why a deterministic myopic approach works well in their studies but not in ours. The results of our numerical analyses suggest that a myopic stochastic approach also performs well when allocations have to be made in the presence of demand uncertainty and differentiated service-level contracts (instead of the homogeneous service-level contracts Abbasi et al. (2017) studied).

4.7 Conclusion

Motivated by some of the practical limitations of current demand fulfillment systems, this study addresses the problem of allocation planning under service-level contracts. We provide a formal definition of the allocation-planning problem under a specific type of service-level contract—that is, a contract with fill-rate targets and linear penalties—and formulate the decision-maker's problem as a stochastic dynamic program. While we derive optimality conditions for the dynamic program, our analyses reveal that the large state space of the system makes it impossible to derive an optimal policy for non-terminal periods. However, our formal analysis did allow us to derive the requirements a good heuristic allocation policy should satisfy. We use these results in two ways: to provide a rigorous discussion of the limitations of "simple" allocation rules that the literature proposes and that are popular in practice, and to develop and study a number of advanced allocation policies that have the potential to improve the performance of allocation planning. We propose several new allocation policies, all of which are based on approximated dynamic programming techniques. After a detailed characterization and discussion of these policies with respect to the aforementioned requirements, we carry out an extensive numerical analysis to determine whether and under what conditions the policies lead to satisfying results.

The results of our numerical analyses provide useful insights into the requirements a "good" allocation policy must fulfill. The allocation policies' performance is predominantly driven by how well they can prioritize customer allocations based on differences in penalties while avoiding misallocations, that is, allocations that are not consumed because demand is too low or that lead to over-achievement of

the service-level targets specified in the contract. Although this driver seems intuitive, other requirements that emerged as important from our theoretical analysis, such as the ability to account for the current state of the system, to prioritize according to customers' specific fill-rate, and to anticipate future supply and demand, turned out to have less influence or insignificant influence on the performance of the allocation policies we studied.

Two of our advanced allocation policies outperform their contenders across all relevant settings: a myopic stochastic policy, MSAP, and a forward-looking stochastic policy, STAP, which is based on the solution of a two-period stochastic program that approximates the $R + 1$-period problem. MSAP outperforms STAP in all of our experiments except in the presence of a strong trend in demand that leads to severe shortages in later periods. This result can be attributed to how STAP approximates the system's future states. Future research may identify better approximations of the underlying stochastic dynamic program's value functions, despite the large state space of the original problem, but our results suggest that a stochastic myopic policy—which is, of course, computationally significantly less expensive—can lead to satisfactory results.

The research we present in this paper has several limitations, at least two of which should be addressed by future research. First, in most of our analyses, the performance of the policy with the ability to look ahead, STAP, was inferior to that of a myopic policy, MSAP. This result raises questions concerning whether more appropriate techniques for approximating the value function of the underlying stochastic dynamic program exist, and whether these techniques can be translated into policies that lead to better performance while being computationally feasible. Finding such techniques is a useful avenue for future research.

Second, our analyses assumed that allocated supply is dedicated exclusively to a particular customer, so we did not account for nesting effects. However, it is reasonable that some form of nesting will occur in practice, as a planner who anticipates that some customer will not make use of its allocation during an individual planning period, while another customer exhausts its allocation, is likely to re-allocate volumes at some point during this period. Such a re-allocation is not supported by today's standard demand fulfillment systems, nor do our models account for this particular form of nesting and its impact on allocation policies' performance. Nested policies are common in revenue management, so future research on nesting in the context of allocation planning under service-level constraints would be useful.

Chapter 5

Managing Service-Level Contracts in Sales Hierarchies[1]

5.1 Introduction

Manufacturers often follow a three-step approach to demand and supply planning that is also implemented in state-of-the-art Advanced Planning Systems. First, during master production planning, the supply availability is forecast for the medium term based on aggregated demand forecasts. Then, during allocation planning (Kilger and Meyr, 2015), the forecasted supply is allocated to local sales organizations (LSOs) and their customers so the manufacturer can prioritize the demands of more important/profitable customers. Last, during order promising, customer orders materialize and are filled until the allocated supply is exhausted (cf. Ball et al., 2004).

As Kilger and Meyr (2015) describe, allocation planning is typically performed throughout companies' sales hierarchies. For example, a company may structure its sales organization with regional managers on the first level (i.e., Europe and North America), country managers on the second level (e.g., Germany, France), and with multiple LSOs for branches (e.g., automotive, aviation) or areas (e.g., North, South) on the third level. In this setting, there is no central planner who can determine all allocations simultaneously, so allocation planning is a decentralized

[1] This chapter is single-authored. The authors use of "we" is for consistency.

process that is performed level-by-level (cf. Vogel and Meyr, 2015). In our example, planners in the company's headquarters would allocate the forecasted supply to regional managers, who would distribute it among the countries for which they are responsible. Then country managers would share the supply among the LSOs, which would, finally allocate the supply to individual customers.

In practice, allocation planning in sales hierarchies is typically determined by simple rules and then manually adjusted by the planners/managers (cf. Kilger and Meyr, 2015). Per commit is an example of such a simple rule that is popular in practice. Under per commit, supply is distributed evenly based on the expected demand from customers. As the approach is based on the expected demands, it requires little information to be shared in the hierarchy and is easy to understand (cf. Fleischmann et al., 2019). However, the resulting allocations are often suboptimal (cf. Kloos et al., 2018). Several recent studies develop more advanced, decentralized approaches to improve allocations over those that result from the simple rules currently applied. These new approaches allocate supply with the objective of maximizing profits or minimizing deviations from service-level targets (e.g., Cano-Belmán and Meyr, 2019; Kloos et al., 2018).

Liang and Atkins (2013) describe the increasing popularity of service-level contracts in the B2B relationships of manufacturers and their customers. Service-level contracts specify a performance target the manufacturer must achieve, the service-level target, over a given period of time, the review horizon, and a penalty for missing it. Under these service-level contracts, allocation planning is particularly difficult because the service-level targets allow a certain part of demand to remain unfilled without incurring penalties. Thus, the value of an allocation becomes clear only when all demands have realized, the service-level is evaluated, and the penalties are determined.

Kloos and Pibernik (2020) analyze this setting under a central planner who can decide on all allocations simultaneously, and developed a myopic policy that significantly decreases expected penalties compared to the simple rules applied in practice and other policies from the literature.

The focus of the present study is on the decentralized allocation planning found in companies with hierarchical sales organizations that seek to minimize the penalties outlined in service-level contracts. Our objective is to develop new approaches based on previous work on hierarchical allocation planning and to quantify their gaps compared to those of central approaches like the myopic policy developed in Kloos and Pibernik (2020). To this end, we decompose the hierarchical-allocation

problem into two subproblems: the customer-allocation problem (CAP) and the hierarchy-allocation problem (HAP). The CAP is the base-level problem and concerns the LSO planners who decide how to allocate supply to their customers with the objective of minimizing penalties. Planners on this level have complete control over their allocations and full information about all of their customers' service-level contracts. Central approaches like the myopic policy can be readily applied to this problem.

The HAP is the top-level problem. It is that of the managers/planners on higher levels of the hierarchy, who cannot directly decide on the allocations to individual customers. Planners on these levels receive allocations from planners at the next higher level and decide how to distribute them among the planners at the next lower level. Typically, planners on this level do not have full information on all the customers' service-level contracts assigned to the individual LSOs under their responsibility (Fleischmann et al., 2019).

Addressing the HAP can be done based on simple rules, but these do not typically lead to optimal allocations to the LSOs (cf. Kloos et al., 2018). One could also solve the HAP centrally and then infer the optimal allocations to the planners individually using a bottom-up-aggregation. However, this approach is not only opposed to the idea of decentralized allocation planning but is also likely meet resistance from the planners, as the resulting allocations are usually not flexible. Other decentralized allocation approaches from the literature are not directly applicable to our setting, as there is no one-to-one correspondence between fulfilled demand and penalty, so profit-based approaches cannot be employed directly. Kloos and Pibernik (2020) find that policies that ignore the differences in customers' penalties lead to poor performance, so we can expect similar results from the service-level-based methods in Kloos et al. (2018). Hence, we develop three new allocation approaches that improve performance and significantly decrease the gap between centralized and decentralized planning.

Our first approach is the penalty-based allocation, which infers approximated per-unit penalties from the service-level contracts and determines allocations based on Fleischmann et al.'s (2019) clustering method. Our second new approach, the dynamic penalties approach, is also based on the clustering method, but it extracts dynamic penalties from the CAP's optimal myopic solution. Our third approach, the smoothed dynamic penalty approach is a variation of the dynamic penalty approach, in which we smooth the dynamic penalties to obtain more stable allocations that benefit the approach's performance.

We analyze the performance of these three allocation approaches in two settings: one in which only allocation planning is decentralized, but inventory and backlog are cleared centrally, and one in which the LSOs are responsible for clearing remaining inventory and backlog, and allocation planning is performed only for newly available supply.

Our numerical results show that the approaches' performance differs significantly between the two settings. Under central inventory/backlog and symmetric hierarchies, applying per commit for the HAP leads to a performance close to that of central planning, while under asymmetric hierarchies, the penalty-based approach leads to good allocations. In this setting, we observe little benefit from the more involved dynamic penalty approach, but that situation changes in the setting with decentralized inventory and backlog-clearing, when the dynamic penalty approach clearly outperforms the other allocation approaches.

The reminder of the paper is structured as follows: Section 5.2 provides an overview of the literature, and Section 5.3 formalizes our setting. Section 5.4 introduces our allocation approaches for the CAP and the HAP. In Section 5.5, we test the resulting allocation systems for their performance with a numerical experiment and derive suggestions for when to apply which approach. In Section 5.6 we analyze a case of decentralized inventory and backlog-clearing, adopt the allocation approaches for this setting, and evaluate their performance numerically. Section 5.7 summarizes and concludes the study.

5.2 Literature Review

Two streams of literature are closely related to our research: studies on the management of service-level contracts and the literature on allocation planning in sales hierarchies.

The problem of allocation planning in sales hierarchies goes back to Kilger and Schneeweiss (2000), who are the first to describe allocation planning as a decentralized process that must be aligned with the company's multi-level organizational structure. Vogel and Meyr (2015) analyze this problem for a single-period setting with deterministic demand and formalize the sales hierarchy as a mathematical tree. They use a numerical study to show that one can obtain close-to-optimal allocations using decentralized allocation approaches. Cano-Belmán and Meyr (2019) extend the analysis to a multi-period setting.

Fleischmann et al. (2019) analyze profit-based allocation planning under stochastic demand, show how to obtain optimal allocation centrally, and test several decentralized allocation approaches. They propose a clustering method that uses limited decentralized information on demand distributions and profit heterogeneity and show that the approach results in close-to-optimal allocations. In a setting in which planners seek to achieve customer-specific (alpha) service-level targets, Kloos et al. (2018) develop and analyze several decentralized allocation approaches and show that a combination of a simple per commit allocation on the upper levels, paired with an optimal allocation at the customer level, can lead to near-optimal allocations when hierarchies are symmetric. While their research is similar to ours, there are two major differences: while Kloos et al. (2018)) analyze a single period setting, we consider multiple periods, and while Kloos et al. (2018) infer penalties from the alpha-service-level targets, we consider service-level contracts with explicit penalties and fill-rate targets.

Also relevant to our study is research on the management of service-level contracts. While most of these studies assume an inventory-management setting in which there is a central planner whose supply is unconstrained, most of their results are relevant to our setting. Thomas (2005) finds that, in practice, service levels are not measured in real time but over a finite review horizon and shows that the length of the review horizon impacts the likelihood of reaching fill-rate targets. Sieke et al. (2012) and Liang and Atkins (2013) analyze the effect of linear and lump-sum penalty schemes. Under lump-sum penalty schemes, a fixed penalty is incurred for any deviation from the set service-level target, while under a linear penalty scheme, increases in the deviation from the service-level target incur increased penalties. Sieke et al. (2012) show that, in a setting with a single customer, a lump-sum penalty the optimal policy for the supplier is to stop serving its customer when reaching the service-level target becomes too costly. Liang and Atkins (2013) observe the same effect for a setting with multiple customers and suggest using linear penalties to incent suppliers always to serve their customers demand.

Protopappa-Sieke et al. (2016) analyze optimal allocations to customers over a two-period setting with homogeneous service-level contracts and lump-sum penalties. Because customers are assumed to be homogeneous, the first period's allocation is trivial, as all customers receive the same share of supply. Allocations in the second period are, as in Fleischmann et al. (2019), given by equal marginal expected penalties. In this setting, customers whose first-period service levels are below a certain threshold receive no allocation in the second period.

Kloos and Pibernik (2020) analyze allocation planning in a setting with constrained supply, heterogeneous customers, linear penalties, and fill-rate targets on a multi-period review horizon. To compute optimal allocations in each period, the planner must solve a stochastic dynamic program, which turns out to be infeasible. Instead, they propose a myopic allocation policy that substantially lowers expected penalties compared to the conventional approaches in their numerical study.

Our research is closely related to Kloos and Pibernik (2020) from which we adopt our general setting. However, Kloos and Pibernik assume a central planner with complete information, where we explicitly model the multiple planners in a multilevel sales hierarchy and the corresponding decentralized decision process. In our setting, planners at the higher levels of the hierarchy plan with only aggregated information and can decide only on the allocations to their direct successors. Hence, our research can also be viewed as a hierarchical extension of Kloos and Pibernik's study.

5.3 Setting

We consider a manufacturer that supplies a single product over R periods to a set of diverse customers who have service-level contracts. The manufacturer's sales organization has a hierarchical structure, where LSOs are responsible for fulfilling the customers' demand, and supply is provided/produced centrally at the top level of the hierarchy. After formalizing the sales hierarchy, the customers' service-level contracts, and the decentralized planning process (Section 5.3.1), we introduce the dynamics of our model and describe the sequence of events in Section 5.3.2.

5.3.1 Organizational Structure and the Planning Process

A sales hierarchy can be described as a balanced mathematical tree with nodes $n \in \mathcal{N}$ on K levels (cf. Vogel and Meyr, 2015). Each node on levels 1 to $K-2$ represents an individual planner in the hierarchy, the nodes on level $K-1$ represent the planners in the LSOs, and the nodes on level K represent the heterogeneous customers. At level 1 there is only the root node $0 \in \mathcal{N}$; all levels $k \in \{1, \ldots, K\}$ contain at least one node $n \in \mathcal{I}_k \subset \mathcal{N}$. Figure 5.1 depicts an example of such a sales hierarchy.

Each customer $l \in \mathcal{I}_K$ on level K has a service-level contract specifying a fill-rate target β_l and a penalty parameter p_l for deviations. The fill-rate is calculated

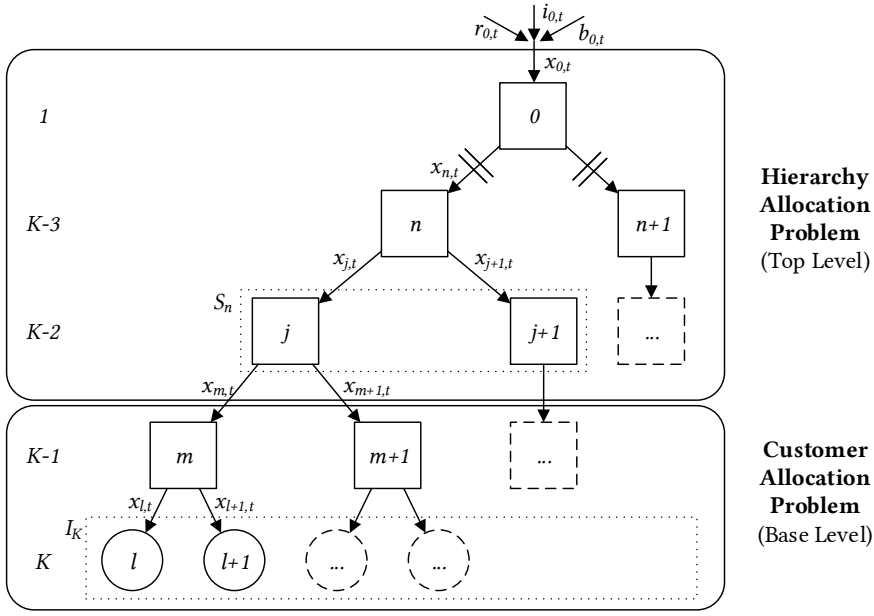


Figure 5.1: A general sales hierarchy.

at the end of the planning horizon R from the total on-time fulfilled demand $y_{R+1,l}$ and total demand $z_{R+1,l}$. Definition 5.1 specifies the linear penalty mechanism.

Definition 5.1 (Penalty of a customer)**.** *Denote the fill-rate target of customer l with β_l, the penalty parameter with p_l, the total demand after period R with $z_{R+1,l}$ and the total fulfilled demand with $y_{R+1,l}$. Then the penalty of customer l is:*

$$P_l(z_{R+1,l}, y_{R+1,l}) = p_l \cdot \left[\beta_l - \frac{y_{R+1,l}}{z_{R+1,l}}\right]^+,$$

where $[x]^+ = \max\{x, 0\}$.

The demand of each customer l in period $t \in \{1, \ldots, R\}$, $D_{l,t}$, is stochastic and follows a continuous distribution with mean $\mu_{l,t}$, standard deviation $\sigma_{l,t}$, pdf $f_{l,t}$, and cdf $F_{l,t} : \mathbb{R}^+ \Rightarrow [0, 1]$.

Level $K-1$ represents the LSOs $m \in \mathcal{I}_{K-1}$ responsible for fulfilling their customers' demands $D_{l,t}$. Customer demand is fulfilled from dedicated allocations $x_{l,t}$—we consider no nesting. The LSOs determine their allocations based on complete information about their customers' service-level contracts and their corresponding demand distributions. The allocations are limited by the allocation $x_{m,t}$ received from the higher level, that is, $\sum_{l \in \mathcal{L}} x_{l,t} = x_{m,t}$. We call this the customer allocation problem (CAP):

Problem 5.1 (Customer allocation problem)**.** The customer allocation problem for LSO $m \in \mathcal{I}_{K-1}$ is

$$\bar{P}_m(x_{m,t}) = \min_{(x_{l,t})_{l \in \mathcal{S}_m}} \mathbb{E}\left[\sum_{l \in \mathcal{S}_m} P_l(Z_{R+1,l}, Y_{R+1,l})\right]$$

subject to

$$\sum_{l \in \mathcal{S}_m} x_{l,t} \leq x_{m,t}$$

$$x_{l,t} \geq 0 \qquad \forall l \in \mathcal{S}_m$$

The hierarchy levels 1 to $K-2$ determine the allocations to the LSOs level-by-level: Starting from the first level, planners on nodes $n \in \mathcal{I}_h$ for $h \in \{1, \ldots, K-2\}$ determine the allocations $x_{j,t}$ to their planners on the second level $j \in \mathcal{S}_n$. All planners have no additional sources of supply, and no supply is retained, so the sum of allocations to the successor nodes equals the supply received from the parent node; i.e., $\sum_{j \in \mathcal{S}_n} x_{j,t} = x_{n,t}$ for all $n \in \mathcal{I}_h, h \in \{1, \ldots, K-2\}$ and $t \in$

$\{1, \ldots, K\}$. We call this the hierarchy allocation problem (HAP) and formalize it as Problem 5.2.

Problem 5.2 (Hierarchy allocation problem)**.** The hierarchy allocation problem for planner $n \in \mathcal{I}_h$ and $h \in \{1, \ldots, K-2\}$ is

$$\bar{P}_n(x_{n,t}) = \min_{(x_{j,t})_{l \in \mathcal{S}_n}} \sum_{j \in \mathcal{S}_n} P_j(x_{j,t})$$

$$\textbf{subject to} \quad \sum_{j \in \mathcal{S}_n} x_{j,t} \leq x_{n,t}$$

$$x_{j,t} \geq 0 \qquad \forall j \in \mathcal{S}_n$$

Problem 5.1 and Problem 5.2 are connected by the expected penalty functions $\bar{P}_m$ of the LSOs $m \in \mathcal{I}_{K-1}$: The hierarchy levels must anticipate the penalty they can expect from an allocation to the LSOs, and the LSOs' decisions are limited by the allocation received from the next higher level. Kloos et al. (2018) show in a similar setting that solving the decentralized allocation problem optimally yields the same results as solving the central problem. However, solving the decentralized allocation problem requires that the planners communicate their real-valued penalty functions $\bar{P}_m$ with planners on the next higher level and that each planner at the hierarchy levels solves a non-linear convex knapsack problem. While feasible from a theoretical standpoint, we consider this approach to be impractical.

5.3.2 Model Dynamics

Supply r_t is given and becomes available at root node 0 at the beginning of each period t. The customers' demand is fulfilled from their respective allocation $x_{l,t}$, and any demand that exceeds $x_{l,t}$ is backlogged and fulfilled before any new demands are fulfilled. Excess allocations and/or backlog are collected centrally. Therefore, the following sequence of events takes place:

1. The root node receives deterministic supply r_t and the remaining inventory $i_{0,t}$.
2. Any remaining backlogged demand $b_{0,t}$ is cleared.
3. The supply available for allocation $x_{0,t} = [r_t + i_{0,t} - b_{0,t}]^+$ is calculated.
4. The planners $n \in \mathcal{I}_h$ on hierarchy levels $h \in \{1, ..., K-2\}$ determine the allocations $x_{n,t}$ to their successor nodes $m \in \mathcal{S}_n$. (They solve the HAP.)

5. The LSOs $m \in \mathcal{I}_{K-1}$ determine their allocations $x_{l,t}$ to their customers $l \in \mathcal{S}_m$. (They solve the CAP.)

6. The demands $D_{l,t}$ of the customers realize and the LSOs fulfill $\min\{D_{l,t}, x_{l,t}\}$ and backlog $[D_{l,t} - x_{l,t}]^+$.

7. The LSOs $m \in \mathcal{I}_{K-1}$ update the variables for fulfilled demand $y_{t+1,l} = \min\{d_{l,t}, x_{l,t}\} + y_{t,l}$ and total demand $z_{t+1,l} = d_{l,t} + z_{t,l}$.

8. The LSOs $m \in \mathcal{I}_{K-1}$ clear any backlogged demand and calculate backlog $b_{m,t+1} = \left[-x_{m,t} + \sum_{l \in \mathcal{S}_m} d_{l,t}\right]^+$ and inventory $i_{m,t+1} = \left[x_{m,t} - \sum_{l \in \mathcal{S}_m} d_{l,t}\right]^+$.

9. The remaining backlog $b_{0,t+1} = \sum_{m \in \mathcal{I}_{K-1}} b_{m,t+1}$ and the leftover inventory $i_{0,t+1} = \sum_{m \in \mathcal{I}_{K-1}} i_{m,t+1}$ are reported to the root node.

In this model, we assume that no physical allocations are made and that unconsumed allocations are available at the root node in the next period. In practice, however, the allocations to the LSOs often result in a physical reallocation of the supply, that is, the supply is shipped to the LSOs. This reallocation does not necessarily contradict our model: When backlog and inventory levels are transparent, they can be cleared decentrally by adjusting the physical allocations to $x^p_{n,t} = x_{n,t} + b_{n,t} - i_{n,t}$, where $b_{n,t} = \sum_{j \in \mathcal{S}_n} b_{j,t}$ and $i_{n,t} = \sum_{j \in \mathcal{S}_n} i_{j,t}$. However, we can also imagine a case in which LSOs are not willing to share their inventory levels or backlog levels with higher levels of the hierarchy. We analyze this alternative setting in Section 5.6.

5.4 Allocation Approaches

The problem of allocating supply in sales hierarchies that we described in Section 5.3 consists of two sub-problems: The CAP is that of the LSOs that determine the allocations to their customers and have complete information on all their customers' demand distributions and service-level contracts, and the HAP is that of planners who determine their allocations level-by-level based on local information. The two problems are interrelated, as allocations to the LSOs limit the supply that can be allocated to fulfill customers' demands, and the hierarchy level relies on the information it receives from the customer level to determine the allocations. Given these characteristics, the allocation problem forms a hierarchical distributed

decision-making system (Schneeweiss, 2003), which is characterized by two decision levels: a top-level and a base-level. The top-level in such a system does not directly determine the final decision, so it must maintain some ability to anticipate the base-level's decisions. As Schneeweiss (2003) explains, the performance of these decision making systems depends not only on both levels' isolated performance but also on how well they interact, that is, how well the top-level anticipates the base-level's decision and objective. In our setting, the HAP is the top-level problem, as its allocations only indirectly influence the performance. The CAP can be seen as the base-level problem, as its decision space is limited by the top-level's decisions, but the allocations of the LSOs to the customers directly influence the penalties.

Section 5.4.1 addresses how to determine allocations for the CAP. Then Section 5.4.2 proposes and discusses four allocation approaches for the HAP. From the allocation approaches on the CAP and the HAP we construct four allocation systems, whose performance we analyze numerically in Section 5.5.

5.4.1 Customer Allocation Problem

Each LSO $m \in \mathcal{I}_{K-1}$ fulfills the demands of its customers $l \in \mathcal{S}_m$ by determining the allocations $x_{l,t}$ with the aim of minimizing the expected penalty. (See Problem 5.1.) The LSOs have complete information about all service-level contracts, including their fill-rate targets and penalty parameters, and about the customers' demand distributions, the total demand $z_{l,t}$, and each customer's total fulfilled demand $y_{l,t}$. The supply available for allocation in the current period is known at the time of the decision and has been determined by the hierarchy level.

Kloos and Pibernik (2020) analyze a similar setting, show that the planners' problem can be described as a stochastic dynamic program, and identify the factors that are relevant to the decision: The planner must simultaneously balance the probability that an allocation in the current period is consumed, the probability that the fill-rate target is reached, and the expected penalty for the deviation. The probability of reaching the fill-rate target depends on all allocations from the current period t to the terminal period R, so it is difficult to obtain, as it requires the planner to consider the available supply and the demand distributions of all remaining periods. In our setting, reaching the fill-rate target is even more challenging, as the allocations to the LSO in future periods are not known at the time the decision is made. Kloos and Pibernik (2020) find that the resulting stochastic

dynamic program has a large state-space that make it infeasible to solve to optimality. Instead, they propose a myopic policy and show that this policy outperforms previous approaches.

Under the myopic policy, allocations are determined as if fill rates are evaluated and penalties are incurred directly after the demand is known. Hence, this policy does not require information on the supply that will be available in subsequent periods, making it suitable for our hierarchical setting. In the following, we describe the concept of the myopic allocation policy and show how to adopt the approach to our setting.

Under the assumption that penalties are incurred directly after demand is known, that is, in the subsequent period $t+1$, Kloos and Pibernik (2020) determine explicit expressions of the expected penalties and, based thereon, characterize the optimal solution. Definition 5.2 adopts the myopic policy to our setting.

Definition 5.2 (Myopic allocation)**.** *The myopic allocation* $\boldsymbol{x}^m_{m,t} = (x^m_{l,t})_{l\in\mathcal{S}_m}$ *of subsidiary* m *in period* t *is:*

$$\boldsymbol{x}^m_{m,t} = \underset{(x_{l,t})_{l\in\mathcal{S}_m}}{\operatorname{argmin}} \sum_{l\in\mathcal{S}_m} \mathbb{E}[C_l(x_{l,t})] \tag{5.1}$$

$$\textit{subject to} \quad \sum_{l\in\mathcal{S}_m} x_{l,t} \leq x_{m,t} \tag{5.2}$$

$$x_{l,t} \geq 0 \qquad \forall l \in \mathcal{S}_m \tag{5.3}$$

where

$$\mathbb{E}[C_l(x_{l,t})] = \begin{cases} \int_{d_{max,l}(x_{l,t})}^{\infty} p\left(\beta_l - \frac{y_{l,t}+x_{l,t}}{z_{l,t}+d_{l,t}}\right) f(d_{l,t})\,\mathrm{d}d_{l,t} & \text{if } \hat{\beta}_{l,t} \geq \beta_l \\ \int_0^{d_{min,l}} p\left(\beta_l - \frac{y_{l,t}+d_{l,t}}{z_{l,t}+d_{l,t}}\right) f(d_{l,t})\,\mathrm{d}d_{l,t} + \dots & \\ \quad \dots \int_{d_{max,l}(x_{l,t})}^{\infty} p\left(\beta_l - \frac{y_{l,t}+x_{l,t}}{z_{l,t}+d_{l,t}}\right) f(d_{l,t})\,\mathrm{d}d_{l,t} & \text{if } \hat{\beta}_{l,t} < \beta_l \text{ and } x_{l,t} > d_{min,l} \\ \int_0^{x_{l,t}} p\left(\beta_l - \frac{y_{l,t}+d_{l,t}}{x_{l,t}+d_{l,t}}\right) f(d_{l,t})\,\mathrm{d}d_{l,t} + \dots & \\ \quad \dots \int_{x_{l,t}}^{\infty} p\left(\beta_l - \frac{y_{l,t}+x_{l,t}}{x_{l,t}+d_{l,t}}\right) f(d_{l,t})\,\mathrm{d}d_{l,t} & \text{else.} \end{cases}$$

with $d_{min,l} = \frac{\beta_l z_{l,t} - y_{l,t}}{1-\beta_l}$ *and* $d_{max,l}(x_{l,t}) = \frac{y_{l,t}+x_{l,t}}{\beta_l} - z_{l,t}$.

The problem that underlies Definition 5.2 is a stochastic knapsack. As Kloos and Pibernik (2020) show, the optimal solution is characterized by the marginal

change in the expected penalty $\frac{\mathrm{d}}{\mathrm{d}x_{l,t}} \mathbb{E}[C_l(x_{l,t})] = \lambda_{l,t}(x_{l,t})$ of each customer $l \in \mathcal{S}_n$. The marginal change in the expected penalty decreases in the allocation because, first, as the allocation increases, the probability that the customer will consume the whole allocation decreases; and, second, the customers' expected fulfilled demand increases with the allocation and, with it, the customers' fill rate. As the fill rate increases, the probability that the target will be achieved increases along with the probability that no penalty will be incurred. Hence, the maximum marginal change in expected penalty is achieved for the first allocated unit. Consequently, in the optimum the customers receive an allocation such that the marginal change in expected penalty is equal. Customers whose maximum marginal change in expected penalty is too low receive no allocation.

Thus, solutions to Definition 5.2 can be computed efficiently. (See Kloos and Pibernik, 2020.) We use the approach in Definition 5.2 with the approaches for the HAP we develop in Section 5.4.2.

5.4.2 Hierarchy Allocation Problem

The HAP can be viewed as the top-level problem of an hierarchical distributed decision-making system. As Schneeweiss (2003) explains, the top level must anticipate the outcome of the planning conducted on the base level. Section 5.3.2 discussed that, for an optimal allocation, the planners must know the expected penalty for a specific allocation to the next level and that that this is not feasible in our setting. Schneeweiss (2003) suggests using the level of anticipation to classify decision-making systems and distinguishes approaches with non-reactive, implicit-reactive, and explicit-reactive anticipation. Approaches with non-reactive anticipation do not anticipate the decisions made at the base level but use only some general features to make decisions. Approaches with implicit-reactive anticipation consider only parts of the of the base level's decisions, whereas explicit-reactive approaches explicitly model the base level's decision, although it may be approximated. Clearly, the higher the level of anticipation, the closer the approach will be to a central model, and we expect performance to improve. On the other hand, a more detailed anticipation typically increases the approach's complexity in terms of the top level's decision-making and the information required from the base level. We propose four allocation approaches that correspond to various levels of anticipation.

Non-reactive Anticipation Per commit, discussed in the introduction, is frequently used in practice (cf. Kilger and Meyr, 2015) and is shown to perform well for allocations in symmetric hierarchies (cf. Kloos et al., 2018). As the approach determines allocations based on expected demands, it can be viewed as a non-anticipating model. We formalize the approach in Definition 5.3.

Definition 5.3 (Per Commit). *Denote with $\mu_{n,t} = \sum_{j \in \mathcal{S}_n} \mu_{j,t}$ the total expected demand of node n in period t. Then the allocation of node n in period t to its successors $j \in \mathcal{S}_n$ is*

$$x_{j,t}^{PC} = x_{n,t} \mu_{j,t} / \mu_{n,t}.$$

As Definition 5.3 shows, per commit requires that the LSOs $m \in \mathcal{I}_{K-1}$ communicate to the next higher level only its customers' total expected demand, so allocations can be determined in a two-step process: The total expected demands are calculated by bottom-up aggregation, and then the available supply is allocated top-down and level-by-level. All calculations are simple and easy to understand, which may be why the approach is popular in practice.

Kloos et al. (2018) analyze per commit for allocating in a sales hierarchy and combine it with a local optimal allocation on the lowest level, terming it the "hybrid approach." They show analytically that the resulting allocations are optimal for symmetric hierarchies, but numerical results suggest that the approach's performance declines as the hierarchy becomes more asymmetric. Although Kloos et al. (2018) consider only a single-period setting with no penalties on deviations from service-level targets, we expect a comparable performance of the per commit approach when the hierarchy is symmetric. In our numerical evaluation in Section 5.5, we compare the performance of applying per commit in symmetric and asymmetric settings.

Implicit-Reactive Anticipation The goal of an approach with implicit-reactive anticipation is to achieve a sufficiently good performance by considering only the most important aspects of the base model. Kloos and Pibernik (2020) compare several allocation approaches featuring various aspects of the problem and find that differences in penalties and in customers' demand distributions are most decisive in how allocation approaches perform. Hence, we expect approaches that consider only differences in penalties to perform comparatively well. The penalty-based allocation approach that we discuss below ignores customers' fill-rate targets and infers per-unit penalties from the penalties in the service-level contracts.

Ignoring the fill-rate targets—or, rather, assuming that fill-rate targets are set to 1—allows us to linearize the penalty function and derive a constant per-unit penalty, thereby simplifying the problem into a penalty-minimization problem. Because of the equivalence of profit maximization and penalty minimization, we can then use the clustering method Fleischmann et al. (2019) propose to compute allocations decentrally. Next we discuss how to derive the per-unit penalties and to use the clustering method in our setting.

Assuming a customer's service-level target is 1, any unfilled demand leads to a penalty: Set $\beta_l = 1$, and the total penalty for customer l simplifies to $p_l(1 - \frac{Y_{R+1,l}}{Z_{R+1,l}}) = \frac{p_l}{Z_{R+1,l}}(Z_{R+1,l} - Y_{R+1,l})$ (cf. Definition 5.1), where $p_l / Z_{R+1,l}$ is the per-unit penalty for each unit of unfilled demand. While $Z_{R+1,l}$ is a random variable, we can approximate it to the total expected demand in the period $\bar{z}_{R+1,l} = z_{t,l} + \sum_{\tau=t}^{R} \mu_{l,\tau}$ and obtain the approximate per-unit penalty[2] $p^u_{l,t} = \frac{p_l}{\bar{z}_{R+1,l}}$.

Based on these per-unit penalties, we can formulate a surrogate allocation problem that minimizes the total per-unit penalties in each period t. For simplicity, we formulate it as a central problem and then discuss how Fleischmann et al.'s (2019) clustering method can be used to solve the problem decentrally:

$$\begin{aligned} & \min \sum_{l \in \mathcal{L}} p^u_{l,t} \int_{x_{l,t}}^{\infty} (d_{l,t} - x_{l,t}) \cdot f_{l,t}(d_{l,t}) \, \mathrm{d}d_{l,t} && (5.4) \\ \textbf{subject to} \quad & \sum_{l \in \mathcal{L}} x_{l,t} \leq x_{0,t} \\ & x_{l,t} \geq 0 && \forall l \in \mathcal{L}. \end{aligned}$$

The surrogate allocation problem in Equation (5.4) minimizes the expected penalties. We can use Fleischmann et al.'s (2019) clustering method to calculate allocations decentrally. The clustering methods starts from the lowest level, clusters the customers according to their per-unit penalties, and calculates the aggregated demand distribution and the average per-unit penalty for each cluster. Then the clusters are shared with the next higher level, which also receives clusters from all of its successors. This level clusters the clusters again, such that the number of clusters made at each level remains constant. This process is repeated up to the highest level. Then the allocation process starts from the top, where each planner

[2] Clearly $p^u_{l,t} \neq \mathbb{E}[\frac{p_l}{Z_{R+1,l}}]$, as $1/\mathbb{E}[Z_{R+1,l}] \neq \mathbb{E}[1/Z_{R+1,l}]$. However, because we calculate the expectation over R periods, the differences are negligible in practical applications, so our simplified formulation is more suitable for practical applications.

determines the optimal allocation to the clusters and then calculates the allocations to nodes on the next lower level. This process continues until the LSOs obtain their allocations. Thus, the planners have to solve only relatively small problems based on the average per-unit penalties and the clusters' aggregated demand distributions. While the method is clearly more complex than determining allocations by per commit, these small problems are tractable for the planners, especially compared to solving the central problem.

Appendix D provides a formal algorithm for the clustering method. To focus on the principle idea behind the penalty-based allocation approach and avoid potentially confusing notation, we define the clustering allocation function as $\boldsymbol{y} = \boldsymbol{C}[\boldsymbol{p}, x]$, which maps the per-unit penalties $\boldsymbol{p} = (p_l)_{l \in \mathcal{I}_K}$ and the available supply x to the LSOs' allocations $\boldsymbol{y} = (y_m)_{m \in \mathcal{I}_{K-1}}$.

We formalize the penalty-based allocation approach in Definition 5.4.

Definition 5.4 (Penalty-based allocation)**.** *The penalty-based allocation to the LSOs in period t is*

$$(x_{n,t}^{pen})_{n \in \mathcal{I}_{K-1}} = \boldsymbol{C}\big[(p_{l,t}^{u})_{l \in \mathcal{I}_K}, x_{0,t}\big]$$

where

$$p_{l,t}^{u} = \frac{p_l}{z_{l,t-1} + \sum_{\tau=t}^{R} \mu_{l,\tau}} \qquad \forall l \in \mathcal{I}_K.$$

The penalty-based approach in Definition 5.4 considers differences in customers' penalties and demand distributions but is based on the assumption that fill-rate targets are set to 1. Therefore, we expect the approach's performance to decline for lower fill-rate targets. We evaluate the performance of the penalty-based approach in our numerical experiment in Section 5.5.

Explicit-Reactive Anticipation An explicit-reactive anticipation approximates the base levels' characteristics explicitly. At the base level, optimal allocations are characterized by an equal marginal change in all customers' expected penalties (cf. Section 5.4.1). Consequently, the allocation on each hierarchy level should aim to achieve an equal marginal change in expected penalties for all customers in all LSOs. More formally, the optimal allocations to the LSOs are $x_{n,t}^{*} = \sum_{l \in \mathcal{S}_n} x_{l,t}^{*}$ for all $n \in \mathcal{I}_{K-1}$ for which, with $A_\lambda = \{l \mid \lambda_{l,t}(0) < \lambda\}$, the following hold: (See Kloos and Pibernik, 2020 for a formal proof.)

$$\lambda_{l,t}(x^*_{l,t}) = \lambda \qquad \forall\, l \in \mathcal{I}_K \backslash A_\lambda \tag{5.5}$$

$$x^*_{l,t} = 0 \qquad \forall\, A_\lambda. \tag{5.6}$$

This characterization of optimal solutions is typically found in convex knapsack problems, so it is structurally similar to the single-period profit-maximizing case Fleischmann et al. (2019) analyzes. However, in their case, $\lambda_{l,t}(x^*_{l,t})$ can be expressed as $\rho_l[1 - F_{l,t}(x^*_{l,t})]$, that is, as the per-unit profit ρ_l times the probability that the allocation is consumed. We show in the following that we can approximate $\lambda_{l,t}(x_{l,t})$ as $p^d_{l,t}[1 - F_{l,t}(x_{l,t})]$, where $p^d_{l,t}$ is a kind of dynamic penalty. With this approximation, the problem's structural form is identical to that of Fleischmann et al. (2019), and we can again use their clustering method to obtain allocations decentrally.

As a first step in developing the dynamic penalty approach, we approximate $\lambda_{l,t}(x_{l,t})$ around a given allocation $x^{base}_{l,t}$. $\lambda_{l,t}$ decreases in $x_{l,t}$ because the probability that $x_{l,t}$ is consumed decreases, and the probability of not paying a penalty increases (cf. Section 5.4.1). Because the fill rate is measured over multiple periods, the probability of not paying a penalty is typically less sensitive to $x_{l,t}$, and the change in the probability that the allocation is consumed has the largest impact, so changes in $\lambda_{l,t}$ result mainly from the change in the probability of consumption. Hence, we use the consumption probability to approximate the marginal change in the expected penalty. Definition 5.5 formalizes this approach.

Definition 5.5 (Approximated marginal change in expected penalty)**.** *The approximated marginal change in expected penalty is*

$$\tilde{\lambda}_{l,t}(x_{l,t}) = \lambda_{m,t}(x^{base}_{l,t}) \cdot \frac{1 - F_{l,t}(x_{l,t})}{1 - F_{l,t}(x^{base}_{l,t})}. \tag{5.7}$$

Definition 5.5 adjusts the marginal change in expected penalty at $x^{base}_{l,t}$ by $\frac{1-F_{l,t}(x_{l,t})}{1-F_{l,t}(x^{base}_{l,t})}$, the change in consumption probability. With $\frac{\lambda_{m,t}(x^{base}_{l,t})}{1-F_{l,t}(x^{base}_{l,t})} = p^d_{l,t}$, we obtain $\tilde{\lambda}_{l,t}(x_{l,t}) = p^d_{l,t}[1 - F_{l,t}(x_{l,t})]$. With this approximation, Equation 5.5 and 5.6 have the same structure as the optimal solution in Fleischmann et al. (2019), and we can apply the clustering method to calculate allocations. Definition 5.6 formalizes the approach.

Definition 5.6 (Dynamic penalty approach)**.** *Denote with* $p^d_{l,t} = \frac{\lambda_{m,t}(x^{base}_{l,t})}{1-F_{l,t}(x^{base}_{l,t})}$ *the dynamic penalty in period* $t \in \{1, \dots, R\}$*, where*

$$x^{base}_{l,1} = \begin{cases} \mu_{l,1} & \text{for } t = 1 \\ x_{l,t-1} & \text{for } t > 1. \end{cases}$$

Then the dynamic penalty approach's allocation to LSOs $m \in \mathcal{I}_{K-1}$ *in period* t *is*

$$(x^{dp}_{m,t})_{m \in \mathcal{I}_{K-1}} = \boldsymbol{C}\big[(p^d_{l,t})_{l \in \mathcal{I}_K}, x_{0,t}\big].$$

The dynamic penalty approach in Definition 5.6 uses the allocation from the previous period $x_{l,t-1}$ to calculate the dynamic penalties for periods $t > 1$. For the first period, the dynamic penalty is calculated for the mean demand $\mu_{l,t}$ of customer l.

The dynamic penalty approach is a two-level approximation of the optimal allocation to the LSOs. First, we approximate the marginal change in expected penalties as shown in Definition 5.5. Then we employ the clustering method to obtain allocations, instead of directly choosing the allocation that leads to equal approximated marginal changes in penalties for all customers. Because of the two levels of approximation, it is difficult to provide any formal bounds on the approach's performance, but as it is directly based on the optimality condition, we expect it to perform well, independent of the level of heterogeneity and the hierarchy's structure. However, the approach is more complex and more difficult to understand than the penalty-based approach, especially for the LSOs, who now have to derive the dynamic penalties from the myopic allocation approach. At the hierarchy level, both the penalty-based and the dynamic penalty approach use the clustering method, so they have the same complexity. However, the per-unit penalty the penalty-based approach uses is easier to understand than the dynamic penalty of the dynamic penalty approach.

When we evaluate the dynamic penalty approach numerically, we observe that dynamic penalties and, with them, the allocations to the LSOs fluctuate, which is detrimental to the approach's performance. To counter this effect, we develop in Definition 5.7 an alternative formulation of the dynamic penalty approach that uses single exponential smoothing to "dampen" the dynamic penalties' effect.

Definition 5.7 (Smoothed dynamic penalty approach)**.** *Denote the smoothed dynamic penalty in period $t \in \{1, \dots, R\}$ with*

$$p^d_{l,t} = \begin{cases} \frac{\lambda_{m,1}(\mu_{l,1})}{1-F_{l,1}(\mu_{l,1})} & \text{for } t = 1 \\ \alpha \frac{\lambda_{m,t}(x_{l,t-1})}{1-F_{l,t}(x_{l,t-1})} + (1-\alpha) p^d_{l,t-1} & \text{for } t > 1. \end{cases}$$

Then the smoothed dynamic penalty approach's allocation to LSOs $m \in \mathcal{I}_{K-1}$ in period t is

$$(x^{sdp}_{m,t})_{m \in \mathcal{I}_{K-1}} = \boldsymbol{C}\big[(p^d_{l,t})_{l \in \mathcal{I}_K}, x_{0,t}\big].$$

If we set $\alpha = 1$, Definition 5.7 is equivalent to Definition 5.6, so both approaches are similarly complex. However, we expect the smoothed dynamic penalty approach to remedy the problem observed for the dynamic penalty approach, but it comes at a cost, as the more the penalty is smoothed, the less reactive the approach is to the newly calculated dynamic penalty that considers updated information. For instance, if a customer's demand is much higher than predicted, its fill rate will decrease, as will the probability of reaching the fill-rate target, and the dynamic penalty in the next period will increase. Because the smoothed dynamic penalty approach adjusts the reported penalty toward the previous dynamic penalty, the allocation under the smoothed dynamic penalty approach will increase less than it will under the dynamic penalty approach. Our numerical evaluation in the next section shows whether this effect on the overall performance is positive or negative.

5.5 Numerical Evaluation

This section presents the results of several numerical experiments conducted to evaluate the performance of the allocation approaches to the HAP, which we developed in Section 5.4.2. Our objective is to identify when each approach leads to a good overall performance and to quantify the losses that are due to penalties compared to those that occur with a central allocation. These results can help decision-makers decide when simple approaches like per commit suffice and when a more complex approach would be beneficial.

In Section 5.5.1, we discuss which parameters and settings are relevant to our analysis. Then Section 5.5.2 outlines how we obtained our results. Finally, Sec-

tion 5.5.3 compares the decentralized allocations systems' performance with a central benchmark and derives suggestions for when to apply each approach.

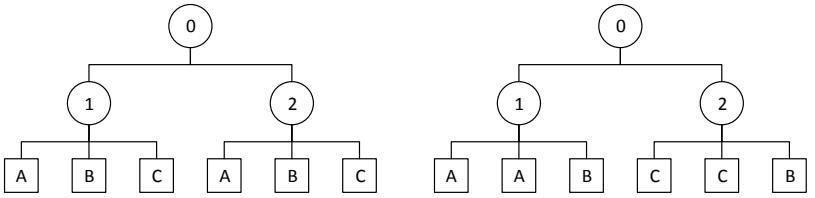
5.5.1 Design

Section 5.4 presented three conjectures on the performance of our allocation approaches:

1. Per commit will perform well when heterogeneity is low and/or the hierarchy is symmetric.
2. The profit-based approach's performance will be similar to that of the central case for high fill-rate targets.
3. The smoothed dynamic penalty approach leads to close-to-optimal allocations, independent of the setting.

From these conjectures we derive three parameters that are most relevant to the allocation approaches' performance: the setup of the hierarchy, the customers' heterogeneity in penalties, and their heterogeneity in fill-rate targets. We want our numerical evaluation to determine the isolated and combined effects of these parameters. In the following we show how we implemented our numerical evaluation.

- **Set-up of the Hierarchy:** To test how the hierarchical set-up impacts the allocation approaches, we use two three-level hierarchies with six customers and three service-level contracts (A, B, C). We analyze a symmetric set-up in which the two sub-trees of the hierarchy are identical, and an asymmetric set-up in which the two sub-trees have different customers. Both set-ups are shown in Figure 5.2. We do not vary the number of customers or the levels of the hierarchy, as the clustering method we use for our allocation approaches is robust against these parameters (cf. Fleischmann et al., 2019).
- **Fill-rate Target Heterogeneity**: We analyze the fill-rate target heterogeneity by means of two scenarios, as the relationship between fill-rate targets and the corresponding required allocations is highly non-linear. Assume, for instance, a single-period setting in which demand is normally distributed with mean 10 and a standard deviation of 2. In this setting, customers with fill-rate targets of 0.8 and 0.81 require allocations of 8.2 units and 8.3 units,



(a) Symmetric **(b)** Asymmetric

Figure 5.2: Setup of the hierarchies analyzed in the paper.

respectively, to achieve their fill-rate targets, so the allocations are similar. For two customers with fill-rate targets of 0.98 and 0.99, the corresponding allocations are 11.8 units and 12.5 units, respectively, so the allocations differ substantially. Consequently, linear scaling for the fill-rate targets would introduce additional effects that could dilute our results and make the individual instances difficult to compare. Therefore, we use only two scenarios: one with homogeneous fill-rate targets, where we set the fill-rate targets of service-level contracts A, B and C to 0.965, and one with heterogeneous fill-rate targets to which we assign a fill-rate of 0.995 to service-level contract A, 0.965 to contract B, and 0.88 to contract C. These settings mirror those in Kloos and Pibernik's (2020) experiments.

- **Penalty Heterogeneity:** We measure the penalty heterogeneity with an approach similar to that of Kloos and Pibernik (2020): We use penalty parameter ρ to compare the additional penalty of the service-level contract that has the highest fill-rate target with the average penalty. Hence, for $\rho = 0$, all service-level contracts have the same penalties, but for $\rho > 0$ ($\rho < 0$) service-level contract A has the highest (lowest) penalties. In the scenario with homogeneous fill-rate targets, there is no difference between positive or negative values of ρ.

All of our experiments assume that customer demand follows a zero-truncated normal distribution with mean $\mu_{l,t} = 10$ and standard deviation $\sigma_{l,t} = 2$ for all customers $l \in \mathcal{I}_K$ and period $t \in \{1, \ldots, R\}$. We set the average penalty to 1000 and, where not stated otherwise, use a review horizon of $R = 10$ periods.

5.5.2 Simulation Environment

We implemented all four decentralized allocation approaches for the HAP and used the myopic allocation policy (Section 5.4.1) for the CAP, so we had four allocation systems: The myopic approach with per commit on the hierarchy levels (M-PC), the myopic approach with the penalty-based approach (M-P), the myopic approach with the dynamic penalty approach (M-DP) and, the myopic approach with the smoothed dynamic penalty approach (M-SDP). To identify the loss in performance that results from the decentralized planning processes, we compare the performance of these allocation systems with a central myopic allocation policy (MC). As the myopic allocation policy is a heuristic, it does not result in optimal allocations. However, the performance of the hierarchical allocation systems with MC lets us estimate how much "optimality" we lose by using a decentralized allocation system. As an additional performance reference, we implement a "pure" per commit allocation (PC), which uses per commit to determine all allocations. Because per commit does not prioritize, its performance can be interpreted as a lower bound.

The allocation systems and the simulation environment are implemented with Python and use NLopt to solve the non-linear optimization problems. We perform our calculations in three steps—initialization, simulation, and evaluation.

Initialization We first draw the customers' demands from a normal distribution truncated at zero. For each scenario we use $I = 200$ individual demand realizations. As is common in Monte-Carlo experiments, we employ the antithetic variates technique (Ross, 2006) to reduce the experiment's variance.

We generate the customer hierarchies from the two hierarchy set-ups, parameterize the customers according to the scenarios' fill-rate targets and penalties, and then assign the demand realizations to the leaf-nodes/customers. The result is a hierarchy that contains all information on the customers and their demand realizations, fill-rate contracts, and demand distributions.

Simulation For each hierarchy and all allocation methods our simulation iterates through the periods of the review horizon in several steps. First, starting from the customer level and iterating up through the hierarchy, we compute the necessary information for each allocation approach. For instance, for per commit we calculate the total mean demand for each node in the hierarchy, and for the penalty-based

approach we compute the clusters' penalties and aggregated demand distributions. Then we clear open backlog and obtain the supply that is available for allocation at the root node. Next, we calculate the allocations top-down until we get to the LSOs. Finally, we solve the CAP to obtain the allocations to the customers. Based on the allocations obtained and the demand realized, we update total demand and total fulfilled demand and calculate the backlog and inventory levels. We store allocations, the total demand, and the total fulfilled demand and proceed to the next period until we reach the end of the review horizon.

Evaluation Having obtained the total fulfilled demand and the total demand for each customer in the review period, we calculate the penalties for each demand realization and allocation system, from which we can calculate the total penalty. To evaluate the performance of the various decentralized allocation systems, we need to compare the average penalties with those from central planning. Therefore, we measure the decentralization error (DE) as the systems' average gap relative to a central myopic allocation. We formalize the DE in Definition 5.8.

Definition 5.8 (Decentralization Error)**.** *The decentralization error of allocation system a is*

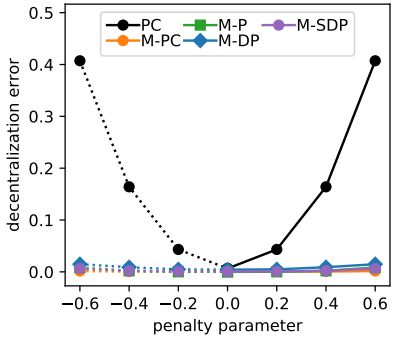
$$DE^a = \frac{\frac{1}{I}\sum_{i=1}^{I}\sum_{l\in\mathcal{I}_K} C_{l,i}^a}{\frac{1}{I}\sum_{i=1}^{I}\sum_{l\in\mathcal{I}_K} C_{l,i}^{MC}},$$

where $C_{l,i}^a$ is customer l's penalty in demand realization $i \in \{1,\dots,I\}$ and allocation system a, and $C_{l,i}^{MC}$ is the penalty under a central myopic allocation.

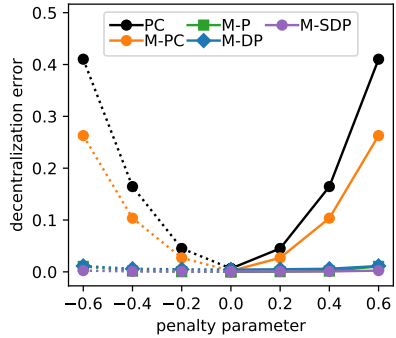
The performance of M-SDP is affected by the choice of the smoothing parameter α. We analyze the optimal settings of α for our scenarios and find that $\alpha = 0.9$ consistently offers the highest performance. Hence, we refrain from optimizing α individually for each scenario and use $\alpha = 0.9$ for all our experiments.

5.5.3 Analysis

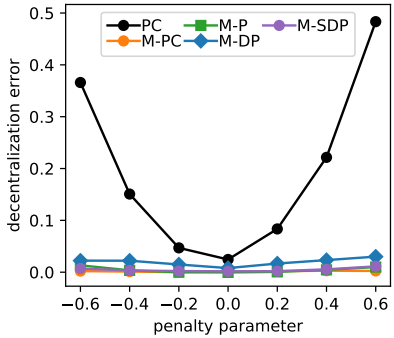
Figure 5.3 plots for various penalty parameters the allocation systems' DEs that result from a symmetric and an asymmetric hierarchy and the homogeneous and heterogeneous fill-rate scenarios. As there is no difference in the positive and negative values for the penalty parameter ρ when fill-rate targets are homogeneous, we depict the DE for negative values of ρ as dotted lines, as they just mirror the results for $\rho > 0$.



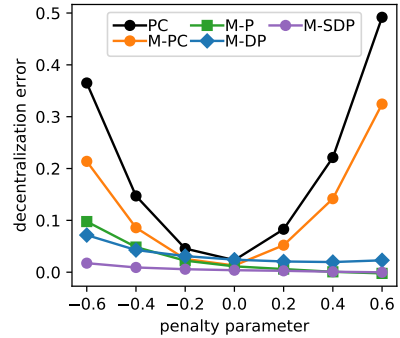
(a) Homogeneous fill-rate targets in symmetric hierarchy



(b) Homogeneous fill-rate targets in asymmetric hierarchy



(c) Heterogeneous fill-rate targets in symmetric hierarchy



(d) Heterogeneous fill-rate targets in asymmetric hierarchy

Figure 5.3: The allocation systems' performance for various penalty parameters.

Before we focus on the allocation systems' performance, we discuss PC's performance. We observe that the DE increases significantly with heterogeneous penalties in all scenarios, reaches 0.49 in the scenario with heterogeneous fill-rate targets, asymmetric hierarchy and $\rho = 0.6$, indicating, that expected penalties are almost 50 percent higher than they are under central planning. This result is intuitive, as the more the customers' penalties differ, the more important it is to prioritize and the relative performance of PC, which does not prioritize, decreases. The effect is not as clear for heterogeneous fill-rate targets: for $\rho \geq 0$, the DE of PC for scenarios with heterogeneous fill-rate targets is higher than it is for scenarios in which the fill-rate targets are homogeneous. However, when $\rho < 0$, the DE of PC for scenarios with heterogeneous fill-rate targets is lower than it is for scenarios in which the fill-rate targets are homogeneous, because PC typically achieves the same fill rates for all customers, so deviations from the fill-rate targets are smaller for customers who have low fill-rate targets. Consequently, when the customers who have low fill-rate targets have the highest penalties, PC's relative performance improves.

As Figures 5.3a and 5.3c show, the DE of M-PC for symmetric hierarchies is close to zero (DE < 0.005), and planners can apply the per commit approach for the HAP without losing performance. This result confirms our initial conjecture regarding per commit's performance: For asymmetric hierarchies, M-PC mirrors PC's performance, although with about 30 percent lower DEs, because M-PC's customer allocations are determined with the myopic allocation approach, so the allocations are at least locally optimal, and the performance is much higher than it would be under PC. Still, the DE is significant, so planners should refrain from applying per commit for the HAP when the sales hierarchy is asymmetric and the customers' penalties are heterogeneous.

M-P's DE is close to zero (< 0.019) in all scenarios but only under negative ρ, heterogeneous fill-rate targets, and an asymmetric hierarchy (cf. Figure 5.3d). In this setting, customers who have high penalties have low fill-rate targets. The penalty-based M-P that is used to solve the HAP ignores the customers' individual fill-rate targets, so it over-allocates to the LSO that is responsible for customers with low fill-rate targets and high penalties. This result suggests that, contrary to our initial conjecture, it is sufficient for M-P that customers who have high penalties also have high fill-rate targets.

In all scenarios M-SDP has the lowest DEs and, as suggested in Section 5.4.2, it outperforms M-DP. We observe that M-SDP's performance decreases under heterogeneous fill-rate targets and a negative ρ, which partially contradicts our initial

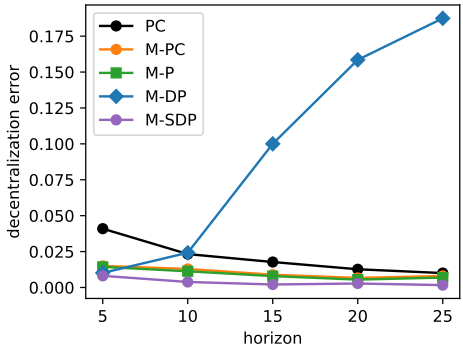


Figure 5.4: Performance of the allocation systems for different review horizons for a scenario with heterogeneous fill-rate targets, asymmetric hierarchy and $\rho = 0$.

conjecture that it performs well independent of the specific setting. In this situation, customers who have low fill-rate targets have high penalties. Recall that the dynamic-penalty approach that underlies M-SDP approximates the marginal change in expected penalties with the consumption probability and ignores the change in the probability of reaching the fill-rate target. When fill-rate targets are low, they are easier to achieve, so the probability of reaching them is more relevant to the allocation decision. Hence, the approximation of the marginal change in expected penalties is less accurate for low fill-rate targets, and M-SDP's performance decreases. Nonetheless, the M-SDP's DE is lower than 0.016, which is probably sufficient for any practical application.

To see how the review horizon affects the allocation systems, we plot the DE of the allocation systems for various lengths of the review horizon $R = \{5, 10, \ldots, 25\}$ for the scenario with heterogeneous fill-rate targets, an asymmetric hierarchy, and $\rho = 0$ (Figure 5.4). The figure shows that the DEs of all allocations systems decrease in the review horizon, while M-DP's DE increases significantly. The general increase in performance of most allocation systems can be traced back to the distribution of the backlog and inventory. Because supply is fixed in our setting, the probability of an extreme backlog or large inventory increases with the review horizon. In these situations, either no supply can be allocated or supply is ample and all demands can be fulfilled, and there are no performance differences between the allocation systems.

To explain why M-DP's DE increases, we plot the average relative allocations (i.e., $x_{n,t}/x_{0,t}$) to the two LSOs using a review horizon of 25 periods (Figure 5.5).

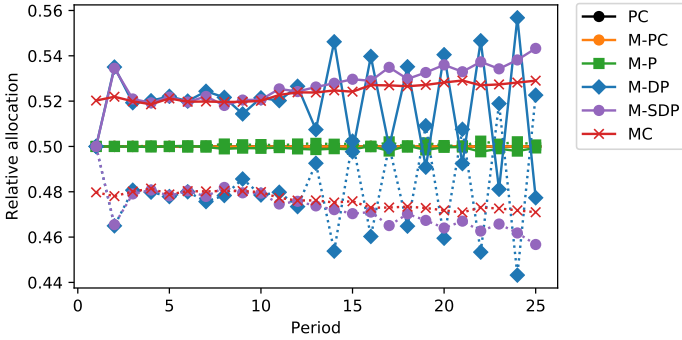


Figure 5.5: Relative allocations to LSO 1 (solid line) and LSO 2 (dotted line) for the AS for a review horizon of 25 periods, heterogeneous fill-rate targets, a asymmetric hierarchy and $\rho = 0$.

We observe that M-DP's allocations fluctuate drastically, while those of M-SDP are more consistent. In particular, M-SDP's allocations are similar to those of MC after a few periods because the dynamic penalties in the first period are calculated for an allocation of the mean demand, after which the approaches' allocations converge relatively quickly toward the of central planner's allocations (MC).

Our experiments show that the performance obtained from a decentralized allocation planning system is close to that of a central planning system when the right allocation approach is selected for the HAP. Our results suggest that two parameters—hierarchy symmetry and customer heterogeneity—affect the optimal choice of an allocation approach. If the hierarchy is symmetric and/or customers are homogeneous with respect to their penalties, using the per commit approach to allocate to the LSOs (i.e., solve the HAP) results in a performance that is close to that of central planning. When M-PC does not lead to a good performance, M-P using the penalty-based approach on the hierarchy levels can close the gap to a central allocation. Only in the situation in which low fill-rate targets coincide with high penalties does M-P's performance deteriorate. Then decision makers should resort to M-SDP, as the approach shows a good performance, independent of the setting. While M-SDP is clearly more complex than and not as easy to understand as per commit or the penalty-based method, the approach is still suitable for practice, as the performance improvements compared to, for instance, a per commit approach, are significant.

5.6 Model Extension: Decentralized Inventory

Section 5.3.2 discussed our assumption that inventory and backlog are managed centrally. Now we want to analyze the effect of this assumption on the performance of decentralized allocation planning. To this end, we formulate an alternative setting in which each LSO is responsible for clearing its own inventory and backlog locally and does not share the information about current inventory and backlog levels with planners in the hierarchy. We chose this extreme case so we could determine the maximum impact of decentralize planning on our allocation system.

Section 5.6.1 introduces the modified sequence of events, while Section 5.6.2 explains how to adopt the allocation approaches for the CAP and HAP to the modified setting. Finally, Section 5.6.3 repeats our previous numerical experiments with the modified setting and discusses the results.

5.6.1 Model Dynamics

In a setting in which the LSOs keep local inventory, the supply at the root node is not affected by the realized demand, so $x_{0,t} = r_t$. However, when the LSOs $m \in \mathcal{I}_{K-1}$ track their own inventory $i_{m,t}$ and backlog $b_{m,t}$, there is a difference between the allocation $x_{m,t}$ the LSOs $m \in \mathcal{I}_{K-1}$ receive and the supply available for allocation $x^a_{m,t}$, leading to an adjusted sequence of events:

1. The root node receives deterministic supply $x_{0,t} = r_t$.
2. The planners $n \in \mathcal{I}_h$ on hierarchy levels $h \in \{1, ..., K-2\}$ determine the allocations $x_{n,t}$ to their successor nodes $m \in \mathcal{S}_n$.
3. The LSOs $m \in \mathcal{I}_{K-1}$ clear remaining backlog $b_{m,t}$ and calculate the supply that is available for allocation $x^a_{m,t} = [x_{m,t} + i_{m,t} - b_{m,t}]^+$.
4. The LSOs $m \in \mathcal{I}_{K-1}$ determine their allocations $x_{l,t}$ to their customer $l \in \mathcal{S}_m$.
5. The customers' demands $D_{l,t}$ are realized, and the LSOs fulfill $\min\{D_{l,t}, x_{l,t}\}$ and backlog $[D_{l,t} - x_{l,t}]^+$.
6. The LSOs update the variables for fulfilled demand $y_{t+1,l} = \min\{d_{l,t}, x_{l,t}\} + y_{t,l}$ and total demand $z_{t+1,l} = d_{l,t} + z_{t,l}$.

7. The LSOs clear potential backlogged demand and calculate the remaining backlog $b_{m,t+1} = \left[b_{m,t} - x_{m,t} - i_{m,t} + \sum_{l \in \mathcal{S}_m} d_{l,t}\right]^+$ and remaining inventory $i_{m,t+1} = \left[x^a_{m,t} - \sum_{l \in \mathcal{S}_m} d_{l,t}\right]^+$.

5.6.2 Adaption of Allocation Approaches

This section discusses how we adapt the allocation approaches from Section 5.4 to the new sequence of events.

At the LSO level, we have to distinguish between the allocation received from the hierarchy $x_{m,t}$ and the supply available for allocation $x^a_{m,t}$. It is straightforward to modify the myopic approach we use for the CAP to consider $x^a_{m,t}$ instead of $x_{m,t}$.

For the CAP, the allocation approach sees almost no impact from the modified setting, but, as we show in the following, such is not the case for the allocation approaches for the HAP.

Per commit allocates based only on the expected demands of the customers, which do not change with the new setting. In addition, the supply that is available for allocation at the root node is comprised of only the deterministic replenishment, so per commit's allocations are now deterministic and can be calculated for all periods in advance. Consequently, the allocation system separates into $|\mathcal{I}_{K-1}|$ problems, where each LSO $m \in \mathcal{I}_{K-1}$ has a fixed supply of $x_{m,t} = \frac{\mu_m}{\mu_0} r_{0,t}$. Thus, there is no more pooling effect among the LSOs, and it is likely that some LSOs have backlogs while others accumulate inventory. Therefore, we expect a significantly lower performance from per commit, as well as from M-PC under a setting with decentralized inventory and backlog clearing.

To determine allocations to the LSOs, the penalty-based allocation uses per-unit penalties $p^u_{l,t}$ derived from the service-level contracts of the customers $l \in \mathcal{I}_K$ and their total expected demand. Our definition of the penalty-based allocation updated the total expected demand in each period based on the corresponding customer's actual demand realizations (cf. Definition 5.4). Hence, $p^u_{l,t}$ changes slightly in each period, so we cannot calculate the allocations up front. However, to allow all allocations to the LSOs to be calculated for the review horizon at once, we can simply fix the per-unit penalties to a single value based on, for instance, only the mean demand (i.e., $p^u_{l,t} = p^u_l = \frac{p_l}{\sum_{\tau=1}^{R} \mu_{l,\tau}}$). Although we do not adjust the penalty-based approach to allow for a fair comparison of the two settings, the approach's allocations do not change with the LSOs' inventory or backlog levels,

so we expect the approach to suffer from decentralized inventories, as it no longer benefits from the pooling effect.

The dynamic penalty approach approximates the marginal change in the expected penalty with dynamic penalty $p^d_{l,t}$ at $x^{base}_{l,t} = x_{l,t-1}$ (cf. Definition 5.6). This approach works well under central inventory, as the allocation to the LSO is likely to be similar in the next period. Under decentralized inventory, the remaining inventory or backlog from the previous period affects the supply that is available for allocation, so it directly impacts the marginal change in the expected penalty. Therefore, instead of calculating the marginal change in the expected penalty for the previous allocation, we calculate it for the allocation we are likely to get. To this end, we add an intermediate step to the dynamic penalty approach by fixing the allocation received from the hierarchy in the last period and correcting this allocation using the remaining backlog/inventory. We then calculate the optimal myopic allocation of this assumed supply $\tilde{x}^a_{m,t} = x_{m,t-1} + i_{m,t} - b_{m,t}$ to the customers and use this allocation to obtain the dynamic penalty $p^d_{l,t}$. Definition 5.9 formalizes our approach.

Definition 5.9 (Dynamic penalty approach for decentralized inventory)**.** *Denote with*

$$\tilde{x}^a_{m,t} = \begin{cases} \sum_{l \in \mathcal{S}_m} \mu_{l,0} & \text{for } t = 1 \\ x_{m,t-1} + i_{m,t} - b_{m,t} & \text{for } t > 1 \end{cases}$$

the assumed supply available for allocation in period t and with $\tilde{x}^m_{l,t}$ the allocation of the myopic approach to customers $l \in \mathcal{L}$. Then $p^d_{l,t} = \frac{\lambda_{m,t}(\tilde{x}^m_{l,t})}{1-G_{l,t}(\tilde{x}^m_{l,t})}$ is the dynamic penalty in period $t \in \{1, \dots, R\}$, and the dynamic penalty approach's allocation to LSOs $m \in \mathcal{I}_{K-1}$ is

$$(x^{dp}_{m,t})_{m \in \mathcal{I}_{K-1}} = \boldsymbol{C}\left[(p^d_{l,t})_{l \in \mathcal{I}_K}, x_{0,t}\right].$$

With the modified dynamic penalty approach for decentralized inventory in Definition 5.9, the LSOs have to calculate their optimal allocations twice: first to obtain the dynamic penalty $p^d_{l,t}$ and then to calculate the actual allocation $x^m_{l,t}$. This double calculation only adds to the complexity of the approach at the LSO level, while the allocation process at the hierarchy levels is unaffected.

As the dynamic penalty approach suffers the same problem of unsteady allocation under decentralized inventory, we also formulate a smoothed version of the dynamic penalty approach in Definition 5.10.

Definition 5.10 (Smoothed dynamic penalty approach for decentralizes inventory)**.** *Denote with*

$$\tilde{x}^a_{m,t} = \begin{cases} \sum_{l \in \mathcal{S}_m} \mu_{l,0} & \text{for } t = 1 \\ x_{m,t-1} + i_{m,t} - b_{m,t} & \text{for } t > 1 \end{cases}$$

the assumed supply available for allocation and with $\tilde{x}^m_{l,t}$ the allocation of the myopic approach to customers $l \in \mathcal{L}$. Then the smoothed dynamic penalty in period $t \in \{1, \dots, R\}$ is

$$p^d_{l,t} = \begin{cases} \frac{\lambda_{m,t}(\tilde{x}^m_{l,t})}{1-G_{l,t}(\tilde{x}^m_{l,t})} & \text{for } t = 1 \\ \alpha \cdot \frac{\lambda_{m,t}(\tilde{x}^m_{l,t})}{1-G_{l,t}(\tilde{x}^m_{l,t})} + (1-\alpha) p^d_{l,t-1} & \text{for } t > 1, \end{cases}$$

and the smoothed dynamic penalty approach's allocation to LSOs $m \in \mathcal{I}_{K-1}$ is

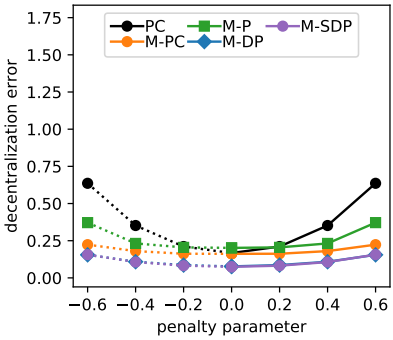
$$(x^{sdp}_{m,t})_{m \in \mathcal{I}_{K-1}} = \boldsymbol{C}\big[(p^d_{l,t})_{l \in \mathcal{I}_K}, x_{0,t}\big].$$

Adopting the allocation approaches to a setting with decentralized inventory affects their complexity. With per commit and, depending on the implementation, with the penalty-based approach, planners can calculate the allocations for all periods of the review horizon at once, which allows planning on the hierarchy levels to be decoupled from planning at the LSO level and reduces the effort required on the hierarchy levels. However, the smoothed dynamic penalty approach becomes more complex: Under decentralized inventory clearing, the LSOs have to calculate the optimal allocation to their customers twice: once to obtaining the virtual penalty and once to perform the actual allocation. However, this problem affects the complexity only at the LSO level; on the hierarchy level, the steps to determine an allocation with the dynamic penalty approach are unaffected.

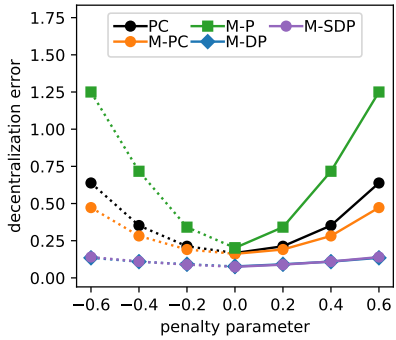
5.6.3 Numerical Evaluation

From the four modified approaches to the HAP and the myopic approach to the CAP we obtain four allocation systems. To evaluate these systems' performance in the setting with decentralized inventory and backlog-clearing, we adopt our simulation environment for the new sequence of events. Again, we use the central myopic allocation approach (MC) as our benchmark, so MC's performance is identical to that in the previous setting.

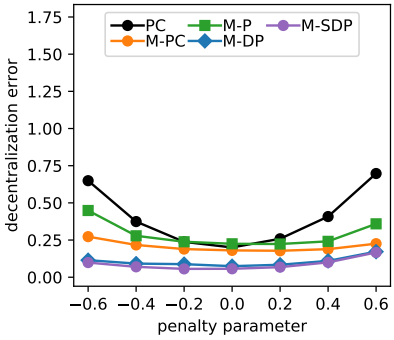
Figure 5.6 plots the DEs of the allocation systems for the four scenarios we analyzed in Section 5.5. Observe that the DEs of all allocation systems are much



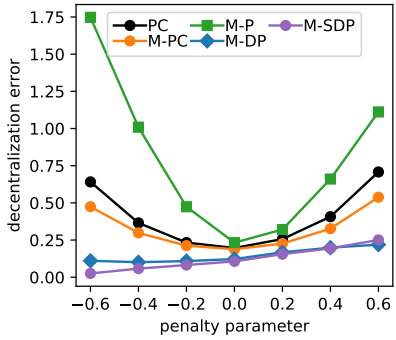
(a) Homogeneous fill-rate targets in symmetric hierarchy



(b) Homogeneous fill-rate targets in asymmetric hierarchy



(c) Heterogeneous fill-rate targets in symmetric hierarchy



(d) Heterogeneous fill-rate targets in asymmetric hierarchy

Figure 5.6: The allocation systems' performance for different values of the penalty parameter and decentralized inventory.

higher than they are in Figure 5.3. For instance, in the scenario with homogeneous fill rates and homogeneous penalties, per commit had a DE of less than 0.01, while in the new setting its DE is 0.17. We can attribute this loss in performance to the LSOs' keeping local inventory, as under per commit, allocations do not consider the LSOs' inventory or backlog, so some LSOs experience backlog-induced shortages while others have surplus supply. This reduces the amount of demand that is fulfilled on time, decreases fill rates, and increases penalties. The same arguments hold for M-PC: Its performance is slightly higher because the LSOs apply the myopic allocation approach, but overall performance still suffers.

M-P's performance suffers even more. Under central inventory and backlog-clearing, the highest DE we observed was below 0.1, and M-P outperformed M-PC in all scenarios. Under decentralized inventory and backlog-clearing, M-P exhibits almost the worst performance in all scenarios, with a DE as high as 1.75 in the scenario with asymmetric hierarchies. Like per commit, the profit-based allocation on hierarchy levels does not change with the inventory or backlog levels in the LSOs. However, because the approach prioritizes customers based on their penalties, repeated prioritization of customers leads to increasing inventory levels for the LSOs that are responsible for high-penalty customers, increasing the backlog for the other LSOs. This effect intensifies with increasing penalty heterogeneity and explains the high DEs we observe for these scenarios.

M-DP and M-SDP have the lowest DEs in every setting, as both approaches adjust their allocations based on the backlog and inventory levels in the LSOs. Still, M-SDP's performance is much lower than it is in Figure 5.3. While the highest DE under central inventory is 0.017, under decentralized inventory the DE reaches 0.25. In contrast to our previous setting, M-DP's performance is not much lower than M-SDP's, perhaps because the LSOs keep local inventories, so the supply that is available for LSOs' allocations is already smoothed.

In the scenario with heterogeneous fill-rate targets and an asymmetric hierarchy, the approaches' performances are highly sensitive to the penalty parameter ρ: M-SDP's performance strictly decreases in the penalty parameter, while DEs are as low as 0.03 for $\rho = -0.6$, they reach up to 0.25 for $\rho = 0.6$. To determine why M-SDP's performance decreases in the penalty parameter, we plot in Figure 5.7 the average total inventory levels in the review period R for the scenario with heterogeneous fill-rate targets and asymmetric hierarchy (Figure 5.6d) for various values of the penalty parameter ρ. Under a large ρ, using the M-SDP leads to increased inventory levels. In these settings, one LSO is responsible for customers who have

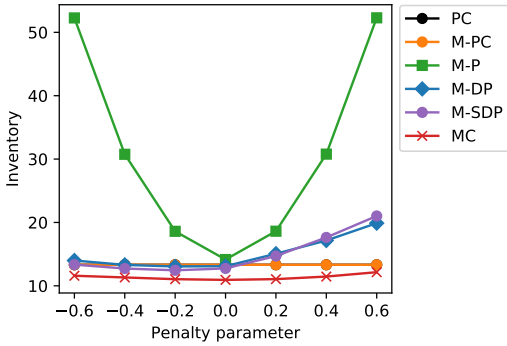


Figure 5.7: Inventory of the allocation systems in period R under decentralized inventory and a scenario with heterogeneous fill-rate targets and asymmetric hierarchy for various values of the penalty parameter ρ.

high fill-rate targets and high penalties, so that LSO requires large allocations to achieve the high-fill rate targets, which frequently are not consumed. Even under MC, these allocations increase the inventory levels, although at a much lower level. Under MC or scenarios with central inventory, any remaining inventory at the end of each period is used to clear the backlog in other LSOs. However, under decentralized inventory, allocations to the LSOs remain untouched and are not used to clear backlog in other LSOs, so inventory levels are much higher, and performance decreases. For $\rho < 0$, this effect is much less pronounced, as high penalties coincide with low fill-rate targets, and required allocations are much lower.

Our numerical results for decentralized inventory differ substantially from our results for the setting with central inventory. While the DEs of all allocation approaches increase significantly under decentralized inventory, they do so at differing rates. M-P shows the strongest decrease in performance and now almost always shows the highest DEs. M-PC's performance also decreases and now leads to significant DEs in all analyzed scenarios. M-SDP's performance also decreases, although the decrease is much lower than it is for the other approaches, and its DEs are the lowest in almost all scenarios.

Decision-makers in companies that have decentrally managed inventories can expect low levels of performance compared to planning centrally, even when they apply the relatively complex M-SDP approach. Therefore, they should consider decentralizing their inventory-clearing to improve performance and allow the use of simpler allocation approaches.

5.7 Conclusion

This study analyzes allocation planning for manufacturers with hierarchical sales organizations that have entered heterogeneous service-level contracts with their customers. In this setting, there is no central planner with complete information on all of the customers' service-level contracts, but planning is decentralized along the company's sales hierarchy. Starting from the top, planned supply is gradually disaggregated along the hierarchy and finally allocated to the LSOs, which are responsible for fulfilling actual customer demand. The LSOs then perform yet another allocation to their individual customers to prioritize their more important customers' demands. Because detailed information on the customers' service-level contracts are available only at the LSOs, we separate the problem into two subproblems: The CAP (base-level problem), which is the allocation problem of the LSOs' having full information, and the HAP (top-level problem), which is the problem of the planners in the hierarchy who decide on the allocations to the planners on the next lower hierarchy level based on some typically aggregated information that they receive from the lower level.

Our study compared the performance loss that companies can expect from this decentralized planning system to that of a central planning approach and develops allocation systems that reduce or minimize this gap. As the CAP is structurally identical to the central model, we adapt Kloos and Pibernik's (2020) myopic allocation approach under complete information to our setting. Hence, the focus of our study is on the HAP, for which we propose and analyze four allocation approaches with differing complexity. Our first approach is per commit, a simple allocation rule that determines allocations based on the customers' expected demands and that is popular in practice. Our second approach is the penalty-based allocation approach which uses per-unit penalties inferred from the service-level contracts to determine the allocations to the LSOs. Our third approach, the dynamic penalty approach, is based on the characterization of optimal solutions and allocates based on dynamic penalties we infer from the optimization performed at the LSOs. Our fourth approach, the smoothed dynamic penalty approach, a variation of the dynamic penalty approach, uses single exponential smoothing to reduce fluctuations in the allocations that we observe in our numerical evaluation of the dynamic penalty approach and significantly improves performance.

From the four allocation approaches for the HAP and the myopic approach for the CAP come four allocation systems: M-PC, M-P, M-DP, and M-SDP. We

Table 5.1: Decision matrix for selecting a suitable decentralized allocation system.

Inventory	**central**		**decentral**	
Hierarchy	**symmetric**	**asymmetric**	**symmetric**	**asymmetric**
$\rho > 0$	M-PC	M-P	M-SDP	centralize inv.
$\rho = 0$	M-PC	M-PC	M-SDP	M-SDP
$\rho < 0$	M-PC	M-SDP	M-SDP	M-SDP

compare these allocation systems to the central planning approach in a numerical experiment and evaluate two distinct settings: one in which we assume that inventory and backlog are cleared centrally, and one in which we assume that the LSOs are responsible for clearing their local inventory and backlog levels and that they do not communicate those levels to other planners.

Our results suggest that, in a setting in which inventory and backlog are cleared centrally, the performance of the decentralized allocation systems are comparable to that of a central planning approach if the planner selects a suitable allocation approach for the HAP. While M-SDP always performs well, in many scenarios much simpler approaches achieve the same performance. For instance, for symmetric hierarchies and/or homogeneous service-level contracts, the per commit rule for the HAP (M-PC) produces high performance. Under asymmetric hierarchies and heterogeneous penalties, M-PC's performance decreases, but M-P, using the penalty-based allocation, still performs well. Only in the case in which low fill-rate targets coincide with high penalties does M-P's performance suffer, so decision-makers should apply the M-SDP approach in this scenario.

When inventory and backlog are cleared locally, decentralized allocation systems perform significantly worse than central planning does. While M-SDP shows the best performance, decision-makers who use it must accept performance losses of about 10 percent compared to a central planning approach. In settings in which hierarchies are asymmetric and high fill-rate targets coincide with high penalties, even M-SDP's performance losses are significant. In such settings, decision-makers should try to centralize the inventory-clearing process. Table 5.1 summarizes our suggestions.

Our analysis provides three important insights for hierarchical allocation planning under service-level contracts: First, decentralized decision-making can, if it is used with appropriate allocation approaches, achieve a level of performance that is similar to that of centralized planning. Second, we provide a useful guideline

for which approach a planner should apply in which situation. Third, our results suggest that having LSOs manage inventories and clear backlog individually affects the allocation systems' performance and requires more complex planning approaches for the hierarchy levels. Therefore, decision-makers should use planning processes in which LSOs share at least information about inventory and backlog levels with the hierarchy.

Chapter 6

Conclusion

This dissertation addresses decentralized allocation planning in sales hierarchies with the objective of reaching customer-specific service-level targets. It is motivated by the gap we observe between the theoretical models suggested in relevant academic literature and the allocation planning as it is performed in companies and implemented in state-of-the-art APS. While most models assume a single planner with the ability to decide on all allocations simultaneously, in practice allocation planning is a decentralized process aligned with the company's hierarchical structure and only supported by simple allocation rules. Thus, our objective is to develop new allocation approaches tailored toward decentralized decision making, identify the gap of conventional allocation rules and our new approaches compared to central planning and provide insight on the relevant information required to obtain good allocations. To this end we perform our analysis in two settings. Chapters 2 and 3 analyze the problem for a single-period setting that allows us to gain structural insights, Chapters 4 and 5 address a more realistic setting with service-level contracts and a multi-period planning horizon. In the following we discuss the individual contributions of the four chapters in this dissertation.

Chapter 2 provides an approach to infer the relative importance of customers from the service-level targets, and characterizes the central and decentralized optimal allocation. Based on these analytical results we also show when conventional allocation approaches result in optimal allocations and develop two new allocation approaches for which we show analytically and numerically that their performance depends on the structure of the hierarchy.

Chapter 3 analyses a profit-maximizing setting and focuses on the information-sharing aspect of decentralized allocation planning. Our numerical analyzes show that two types of information are most relevant for good allocations: the customers' profit heterogeneity and demand stochasticity. With the clustering and the stochastic Theil approach we developed two new approaches that both use this information and, thus, lead to close-to-optimal performance. For practitioners the results for the clustering approach may be most beneficial. Our analyses suggest, that communicating the profit heterogeneity by the means of two or three clusters (high, medium and low profits) leads to a performance that is comparable to that of a central approach.

In Chapter 4 we provide a formal definition of the central allocation planning problem under a service-level-contract with fill-rate targets and a linear penalty and formulate the corresponding stochastic dynamic program. Based on our analytical analysis for the dynamic program, we identify six requirements a "good" allocation policy should fulfill. Based on these requirements we analyze heuristics from literature and practice and propose several new allocation policies based on approximated dynamic programming. An extensive numerical study allows us to quantify the importance of the requirements and compare the performance of the allocation policies.

Chapter 5 extends the research from Chapter 4 to decentralized planning. We decompose the resulting problem into two hierarchical sub-problems which allow us to combine the central approaches developed for allocation planning under service-level contracts with the hierarchical allocation methods developed in Chapter 3. In a numerical study we evaluate the performance of the resulting allocation systems comprised of the approaches applied for both subproblems and find that, by choosing the correct allocation approaches, performance is similar to central planning.

Chapters 2, 3 and 5 provide important insights for the hierarchical allocation problem in different settings. Our results show that decentralized planning does not have to be accompanied by performance losses when applying "suitable" allocation approaches. We also provide insight on how to select a "suitable" allocation approach: In settings where the hierarchy is symmetric with respect to the customers' (service-level) requirements, simple rules such as per commit typically lead to close-to-optimal performance. Consider, for instance, a company that structured its sales hierarchy by countries, i.e., Germany and Austria. When the customer base in both countries is similar (with respect to their service-level

requirements), we can advise decision makers to resort to a simple per commit approach and expect no or only little performance losses as compared to central planning approaches.

In other settings, however, the performance that can be expected from simple rules as per commit is detrimental. Consider, for instance, a company having sales organizations in countries on different continents, e.g., Germany and China. Then the customers base in the two countries may be very different and decision makers should put more effort in distributing their supply among the different countries. In these settings, our advanced approaches yield superior performance while being only slightly more complicated. For instance, the clustering method, which we tested both in a setting under profit maximization and planning under service-level contracts, requires the planners to cluster their customers into only two or three clusters according to their profits/penalties.

Finally, we want to mention further avenues for research. Throughout this dissertation we assume a partitioned allocation and ignore the possibility of nested consumption, that is, we assume that allocations to a customer can only be consumed by this very customer. While nesting has been shown to be very beneficial as it allows to realize inventory pooling-effects between customers, models considering nesting are typically associated with strong assumptions on the customers' ordering behavior (i.e., Poisson process). Data-driven modeling approaches may allow to generate nested allocations directly from a company's order data and avoid problematic assumptions on the customers' ordering behavior. This appears as a promising direction for further investigation.

This dissertation only addresses settings where decision makers either face the same type of service-level contracts for all customers or only regard the customers' different profitabilities. In practice, however, it is likely that a company faces customers both with and without service-level contracts. Such mixed allocation systems raise numerous questions that offer interesting research opportunities.

Appendix A

Appendix to Chapter 2

A.1 Proofs of the Mathematical Results

Proposition 2.1. Part 1 is straightforward from $\hat{x_l}(x_l) = \mathbb{E}[\min(D_l, x_l)]$.

Clearly, $\hat{x_l}(x_l) + L_l(x_l) = \mathbb{E}[D_l]$. Hence, combining this with part 1 yields part 2. Part 3 holds by definition. □

Lemma 2.1. As sums of convex functions are convex and L_l is independent from x_k for all $k \neq l$, we only show that $L_l(x_l)$ is convex in x_l, which is a straightforward consequence of the more general statement that $F(x) := \int_x^\infty f(t)\ \mathrm{d}t$ is convex if f is monotonously decreasing and uniformly bounded above and below, i.e., $|f| \leq C$

for some $C < \infty$. In order to show the latter, note that F is continuous and fix some $-\infty < x < y < \infty$. Now

$$\begin{aligned}
F\left(\frac{x+y}{2}\right) &= \frac{1}{2}\left(\int_{\frac{x+y}{2}}^{\infty} f(t)\,\mathrm{d}t + \int_{\frac{x+y}{2}}^{\infty} f(t)\,\mathrm{d}t\right) \\
&= \frac{1}{2}\left(\int_{x}^{\infty} f(t)\,\mathrm{d}t - \int_{x}^{\frac{x+y}{2}} f(t)\,\mathrm{d}t + \int_{\frac{x+y}{2}}^{y} f(t)\,\mathrm{d}t + \int_{y}^{\infty} f(t)\,\mathrm{d}t\right) \\
&= \frac{1}{2}\left(\int_{x}^{\infty} f(t)\,\mathrm{d}t + \int_{x}^{\frac{x+y}{2}} \underbrace{\left[f(t) - f\left(t + \frac{y-x}{2}\right)\right]}_{\leq 0}\,\mathrm{d}t + \int_{y}^{\infty} f(t)\,\mathrm{d}t\right) \\
&\leq \frac{1}{2}\left(\int_{x}^{\infty} f(t)\,\mathrm{d}t + \int_{y}^{\infty} f(t)\,\mathrm{d}t\right) \\
&= \frac{1}{2}F(x) + \frac{1}{2}F(y),
\end{aligned}$$

which concludes the proof. □

Theorem 2.1. (2.4) follows immediately from $x_0 < x_0^r$, which can easily be seen by contraposition: If $\epsilon := x_0 - \sum_{l \in \mathcal{I}_K} x_l^*$ were strictly greater than 0, then for at least some $l' \in \mathcal{I}_K$ we would have $x_{l'}^* < x_{l'}^r$ and, thus, replacing $x_{l'}^*$ by $x_{l'}^* + \epsilon$ would further reduce L, which contradicts optimality.

Regarding the remaining proof, note that Problem 2.1c is a non-linear optimization problem with linear constraints, namely, $\sum_{l \in sI_K} x_l \leq x_0$ and $x_l \geq 0$ for all l. Since all constraints are linear and $\boldsymbol{x} \equiv 0$ is a feasible allocation, the refined version of Slater's condition (cf. Boyd and Vandenberghe, 2004) holds. In addition, by Lemma 2.1, $W(x)$ is convex; hence, by standard non-linear optimization theory (cf. Ruszczyński, 2006), for any $\boldsymbol{x^*}$ there exist $\lambda, \mu_l \in \mathbb{R}_0^+$ such that for all $l \in \mathcal{I}_K$

$$(\mu_l - \lambda)/w_l \in \partial L_l(x_l^*) \tag{A.1}$$

$$\lambda\Big(\sum_{l \in \mathcal{I}_K} x_l^* - x_0\Big) = 0 \tag{A.2}$$

$$\mu_l x_l^* = 0 \tag{A.3}$$

and any feasible point for which $\lambda, \mu_l \in \mathbb{R}_0^+$ exist such that equations (A.1) - (A.3) hold is a solution of Problem 2.1c.

Hence, (2.2) holds for all l with $\mu_l = 0$ and (2.3) holds for all l with $\mu_l > 0$. By definition of Problem 2.1c, $x_0 < x_0^r$, that is, the constraint $\sum_{l \in sI_k} x_l \leq x_0$ is binding. Accordingly, $\lambda > 0$. It remains to show that $\{k \mid k \in \mathcal{I}_K, \mu_l > 0\} = A_\lambda$.

If $\mu_l > 0$, then by (A.3) $x_l^* = 0$. Furthermore, (A.1) together with the fact that G_l is càdlàg and, therefore, $G_l(0) - 1$ is an upper bound for $\partial L_l(0)$ yields $-\lambda/w_l < \mu_l/w_l - \lambda/w_l \leq G_l(0) - 1$. Accordingly, $\{k \mid k \in \mathcal{I}_K, \mu_l > 0\} \subset A_\lambda$.

Conversely, if $\lambda \geq (1 - G_l(0))w_l$, then for all $t \geq 0$ we have $-\lambda/w_l \leq G_l(0) - 1 \leq G_l(t) - 1$, where we used the monotonicity of G_l and which immediately yields $\mu_l > 0$ as (A.1) entails $\mu_l/w_l - \lambda/w_l \geq G_l(x_l^*) - 1$ (where we again used G_l's being càdlàg). Hence, also $\{k \mid k \in \mathcal{I}_K, \mu_l > 0\} \supset A_\lambda$, which concludes the proof. □

Corollary 2.1. If G_l is continuous, then by the first fundamental theorem of calculus L is differentiable with respect to x_l and $\frac{\mathrm{d}}{\mathrm{d}x_l} L(x_l) = G_l(x_l) - 1$. Hence, (2.2) reduces to $\lambda/w_l = -L_l'(x_l^*) = (1 - G_l(x_l^*))$, which—as G_l is strictly increasing on $\{G_l < 1\}$ and, by Theorem 2.1, $\lambda > 0$—implies the assertion. □

Lemma 2.2. To avoid the use of obfuscating notation, we provide the proof for the case where G_l is continuous and strictly increasing.

1. Fix l, k with $\alpha_l \geq \alpha_k$ and note that, in this case, $w_l \geq w_k$. Now, let $\boldsymbol{x}^*$ be the optimal allocation and let λ^* be the corresponding supply parameter. We distinguish three cases depending on the relation between w_l, w_k and λ^*:

   - $\lambda^* > w_l \geq w_k$: In this case $x_k^* = x_l^* = 0$ and the assertion holds.
   - $w_l \geq \lambda^* \geq w_k$: In this case $x_l^* \geq 0 = x_k^*$ and the assertion holds.
   - $w_l \geq w_k \geq \lambda^*$: In this case, $\hat{\alpha}_l = G_l(x_l^*) = 1 - \lambda/w_l = 1 - \lambda(1 - \alpha_l)$ $\geq 1 - \lambda(1 - \alpha_k) = G_k(x_k^*) = \hat{\alpha}_k$.

2. Without loss of generality, set $\lambda = 1$, note that

$$\lambda = 1 \leq 1/(\underbrace{1 - \alpha}_{\geq 1}) \cdot (\underbrace{1 - G_l(0)}_{=1})$$

   and $x_l^r = G_l^{-1}(\alpha_l) = G_l^{-1}(1 - 1/w_l) = G_l^{-1}(1 - \lambda/w_l)$. Hence, by Corollary 2.1 $x_l^* = x_l^r$ is the unique solution of Problem 2.1c and, thus, the assertion holds.

□

Lemma 2.3. Lemma 2.3 is a reformulation of Bellman's (1957) principle of optimality.

□

Proposition 2.2. Proposition 2.2 is a straightforward consequence of Lemma 2.3.

□

Proposition 2.3. We first show that if (2.5) holds $x^r_{l'}/\mu_{l'} = x^r_0/\mu_0$ for all $l \in \mathcal{I}_K$. Set $\chi = x^r_{l'}/\mu_{l'}$; then from (2.5) and the fact that $w_l = 1/(1-\alpha_l)$ and $G_l(x^r_l) = \alpha_l$ follows

$$\frac{1-\alpha_{l''}}{1-\alpha_{l'}} = \frac{1-G_{l''}(\mu_{l''}x_{l'}/\mu_{l'})}{1-\alpha_{l'}}$$

and thus $\alpha_{l''} = G_{l''}(\mu_{l''}x_{l'}/\mu_{l'})$. From this, by definition of $x^r_{l''}$ and the fact that $G_{l''}$ is strictly increasing, follows $x^r_{l'}\mu_{l''}/\mu_{l'} = x^r_{l''}$. As $x^r_0 = \sum_{l\in\mathcal{L}} x^r_l$, we derive that $x^r_0 = x^r_l\mu_0/\mu_l$. Hence $x^r_{l'}/\mu^r_{l'} = x^r_0/\mu^r_0$.

With this $\chi \in \left(0, x^r_{l'}/\mu_{l'}\right]$ is equivalent to $\chi \in \left(0, x^r_0/\mu^r_0\right]$ and we can rearrange (2.5) to:

$$w'_l(1-G_{l'}(x_0\mu_{l'}/\mu_0)) = w_{l''}(1-G_{l''}(x_0\mu_{l''}/\mu_0)) \qquad \text{for all } l', l'' \in \mathcal{I}_K, x_0 \in (0, x^r_0].$$

As $x^{PC}_{l'} = x_0\mu_{l'}/\mu_0$ it follows that $w'_l(1-G_{l'}(x^{PC}_{l'})) = \lambda$ for all $l' \in \mathcal{I}_K$ and $x_0 \in \left(0, x^r_0\right]$. Replace $(1-G_l(x^*_l)) = -L'_l(x^*_l)$ to see that Theorem 2.1 applies and thus $x^{PC}_{l'}$ is optimal.

Next we proof the inverse implication: If (2.5) does not hold, then there exits at least one $\chi \in \left(0, x^r_l/\mu_l\right]$ and a pair of customers l, l' such that:

$$w_{l'}(1-G_{l'}(\mu_{l'}\chi)) \neq w_{l''}(1-G_{l''}(\mu_{l''}\chi)) \tag{A.4}$$

For $x_0 = \chi\mu_0$ the *per commit* allocations are $x^{PC}_l = \chi\mu_l$ for all $l \in \mathcal{I}_K$. As $\chi > 0$, $x^{PC}_l > 0$ and hence, all customer groups receive the allocation x^{PC}_l. By Theorem 2.1 this allocation is optimal if and only if

$$w_l(1-G_l(x^{PC}_l)) = \lambda \qquad \text{for all } l \in \mathcal{I}_K \backslash A_\lambda$$

This equivalent to

$$w_{l'}(1-G_{l'}(\mu_{l'}\chi)) = w_{l''}(1-G_{l''}(\mu_{l''}\chi))$$

which contradicts (A.4) and, thus, completes the proof. □

Proposition 2.4. The proof of part 1 is straightforward from the definition of *extended per commit*.

For part 2, assume the requirements of Proposition 2.4 part 2 hold. Then in analogy to the proof of Proposition 2.3, the identity $x_0^r/\mu_0 = x_l^r/\mu_l$ holds. Hence $x_0^r/x_l^r = \mu_0/\mu_l$ with which it is easy to show that $x_l^{ePC} = x_l^{PC} = x_l^*$. □

Proposition 2.5. Part 1 follows directly from the definition of *rank based* allocations.

For part 2 we first prove that, if (2.8) holds, the *rank based* allocation is optimal. Assume, without loss of generality, there are only two customers, l' and l'', with $\alpha_{l'} > \alpha_{l''}$. Then the allocation vector $\boldsymbol{x}^{RB} = (x_{l'}, x_{l''})$ has two cases (cf. Definition 2.5):

$$\boldsymbol{x}^{RB} = \begin{cases} (x_0, 0) & \text{if } x_0 < x_{l'}^r \\ (x_{l'}^r, x_0 - x_{l'}^r) & \text{else.} \end{cases}$$

First, we regard the case that $x_0 < x_{l'}^r$. Set $\lambda = [1 - \lim_{x_l \nearrow x_0} G_{l'}(x_l)] \cdot w_{l'}$; then (2.2) holds for l' and because $G_{l'}$ is increasing, $\lambda \geq [1 - \lim_{x_l \nearrow x_l^r} G_{l'}(x_l)] \cdot w_{l'}$. By (2.8) it is straightforward that $l'' \in A_\lambda$ and the allocation is optimal.

Now we are left to show that the allocation $\boldsymbol{x}^{RB} = (x_{l'}^r, x_0 - x_{l'}^r)$ is optimal if $x_0 \geq x_{l'}^r$. Set $\lambda = [1 - G_{l''}(x_0 - x_{l'}^r)]$; then (2.2) holds for l''. With (2.8) and as, by definition, $G_{l''}(x) \leq \alpha_{l''}$ for all $x < x_{l''}^r$, we can bound λ to

$$[1 - \lim_{x_l \nearrow x_0} G_{l'}(x_l)] \cdot w_{l'} \geq \lambda > 1.$$

By definition, $G_{l'}(x_{l'}^r) \geq \alpha_{l'}$ and $w_{l'} = 1/(1 - \alpha_{l'})$, therefore $w_{l'}(1 - G_{l'}(x_{l'}^r)) \leq 1$. With this it is straightforward that (2.2) holds for l' and the allocation is optimal.

Finally, we show that if (2.8) is violated there is always at least one x_0 for which the *rank based* allocation is not optimal. Assume that (2.8) does not hold for customers l_k and l_{k+1}, and set $x_0 = \sum_{l \in \{l_1, \ldots, l_k\}} x_l^r$. Then the corresponding allocations are $x_{l_k}^{RB} = x_{l_k}^r$ and $x_{l_{k+1}}^{RB} = 0$ and the following holds by assumption

$$w_{l_k}(1 - \lim_{x_{l_k} \nearrow x_{l_k}^r} G_{l_k}(x_{l_k})) < w_{l_{k+1}}(1 - G_{l_{k+1}}(0)).$$

The conditions for an optimal allocation are (cf. Theorem 2.1):

$$\text{for } l_k: \qquad w_{l_k}\left(1 - G_{l_k}(x^r_{l_k})\right) \leq \lambda \leq w_{l_k}(1 - \lim_{x_{l_k} \nearrow x^r_{l_k}} G_{l_k}(x_{l_k}))$$

$$\text{for } l_{k+1}: \qquad w_{l_{k+1}}\left(1 - G_{l_{k+1}}(0)\right) \leq \lambda$$

It follows that $\lambda \leq w_{l_k}(1 - \lim_{x_{l_k} \nearrow x^r_{l_k}} G_{l_k}(x_{l_k})) < w_{l_{k+1}}(1 - G_{l_{k+1}}(0)) \geq \lambda$ which leads to a contradiction. Hence, the allocation is not optimal which concludes the proof. □

Proposition 2.6. As $x^H_m = x^{ePC}_m$, part 1 follows directly from Proposition 2.4.

Part 2 follows directly from Corollary 2.1 and Lemma 2.3. □

Proposition 2.7. Straightforward from Proposition 2.6 part 1 and the property that $x^H_n = x^{ePC}_n$ for all $n \in \mathcal{I}_{K-1}$. □

Proposition 2.8. Note that (2.11) mirrors the results of Corollary 2.1. Therefore, straightforward computations suffice to check optimality of the *service level aggregation* approach. □

A.2 Formulae to Determine Customer Parametrization from Performance Drivers

Forecast Heterogeneity Denote with $|\mathcal{I}_K|$ the number of customer classes, with $\overline{CV}$ the average CV (forecast accuracy) and with H_{CV} the targeted forecast heterogeneity. Then with

$$r = \frac{2\sqrt{3}H_{CV}\sqrt{|\mathcal{I}_K| - 1}\overline{CV}}{\sqrt{|\mathcal{I}_K| + 1}}$$

the n'th customers CV is:

$$CV_n = \overline{CV} - \frac{r}{2} + (n-1) \cdot \frac{r}{|\mathcal{I}_K| - 1}.$$

Service-level Heterogeneity Let $|\mathcal{I}_K|$ be the number of customer groups, w_{max} the maximum shortfall-weight and H_{SL} the targeted service level heterogeneity. Then with

$$w_{min} = \frac{2\sqrt{3}\sqrt{(|\mathcal{I}_K|^2 - 1)H^2_{SL}} + 3H^2_{SL} - |\mathcal{I}_K| - 3H^2_{SL}|\mathcal{I}_K| - 1}{3H^2_{SL}|\mathcal{I}_K| - 3H^2_{SL} - 1} w_{max}$$

the n'th customer groups shortfall weight is

$$w_n = w_{min} + (n-1)\frac{w_{max} - w_{min}}{|\mathcal{I}_K| - 1}.$$

Appendix B

Appendix to Chapter 3

B.1 Proofs of Analytical Results

Proof of Lemma 3.1. $\frac{dP}{dx_l} = p_l(1 - F_l(x_l)) \geq 0$ therefore the objective function is increasing in x_l. $E[\min(x_l, D_l)]$ is concave, and thus (3.7) is concave as a weighted sum of concave functions. □

Proof of Proposition 3.1. According to Lemma 3.1, Problem 3.2 is a convex continuous knapsack problem. Letting γ denote the Lagrange multiplier for $\sum_{l \in \mathcal{L}} x_l \leq S$, Bretthauer and Shetty (2002b) use the KKT conditions to show that the optimal solution satisfies:

$$x_l = \begin{cases} 0 & if \quad {F_l}^{-1}(1 - \frac{\gamma}{p_l}) \leq 0 \\ {F_l}^{-1}(1 - \frac{\gamma}{p_l}) & if \quad 0 < {F_l}^{-1}(1 - \frac{\gamma}{p_l}) \end{cases} \tag{B.1}$$

Thus, marginal expected profits are balanced for all nodes that receive a non-zero allocation. Moreover, Zipkin (1980b) proves that for increasing capacity, the non-zero variables appear in the optimal solution consecutively, in decreasing order of $p_l(1 - F_l(0))$, which is the marginal expected profit of starting to allocate supply to node l. Thus, for each node l, we can define a supply threshold $\overline{S_l}$ which implies a non-zero allocation to that node. For $S = \overline{S_l}$, it follows from (B.1) that $\gamma = p_l(1 - F_l(0))$. Again (B.1) and Zipkin's result then imply $x_i = {F_i}^{-1}(1 - \frac{p_l(1-F_l(0))}{p_l})$ for all i with $p_i(1 - F_i(0)) \geq p_l(1 - F_l(0))$, as in (3.10).

Since the objective function is increasing, it is always optimal to allocate all available supply to the leaf nodes. Thus, we can assume constraint (3.8) to be binding. Hence, γ and consequently the optimal allocations can be determined by solving

$$\sum_{\{l \in \mathcal{L} | \overline{S_l} \leq S\}} {F_l}^{-1}(1 - \frac{\gamma}{p_l}) = S. \tag{B.2}$$

The uniqueness result follows from the monotonicity of F_l. □

B.2 Additional Figures

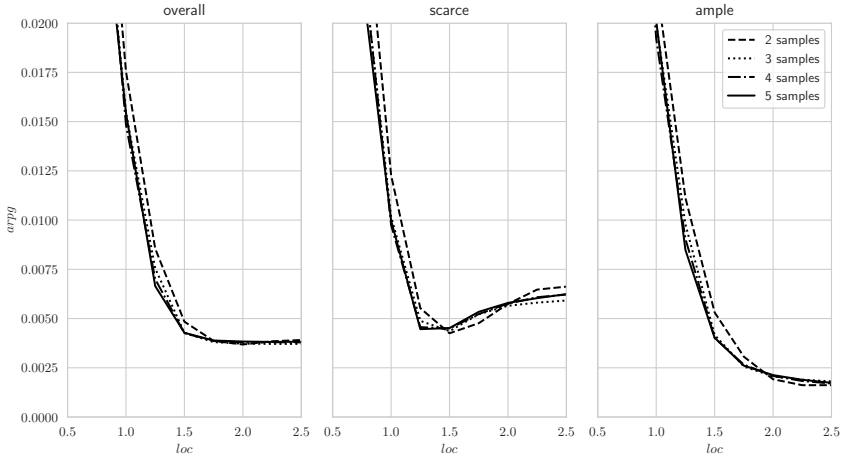


Figure B.1: arpg of the stochastic theil method for different number of sample R and location parameter loc under overall ($x_0 \in [0.5d_0, 1.5d_0]$), scarce ($x_0 \in [0.5d_0, 1.0d_0]$) and ample supply ($x_0 \in [1.0d_0, 1.5d_0]$) for the baseline scenario.

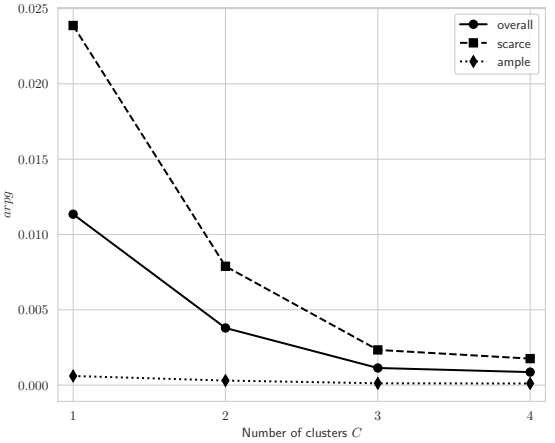


Figure B.2: arpg of the clustering method for different number of clusters C under overall ($x_0 \in [0.5d_0, 1.5d_0]$), scarce ($x_0 \in [0.5d_0, 1.0d_0]$) and ample supply ($x_0 \in [1.0d_0, 1.5d_0]$) for the baseline scenario.

Table B.1: arpg of different allocation methods in individual experiments across all supply levels.

Scen.	Cust.	Levels	Demand	CV	Profit het.	Per commit	Clust. [1]	Det. Theil	Stoch. Theil	Clust. [2]	Clust. [3]
1	30	4	10	0.10	high	6,71%	2,37%	–	0,52%	0,58%	0,18%
2					medium	5,92%	2,14%	–	0,51%	0,54%	0,17%
3					low	4,72%	1,71%	–	0,46%	0,40%	0,12%
4				0.20	high	6,23%	1,33%	–	0,49%	0,44%	0,13%
Bl.					medium	5,35%	1,13%	–	0,43%	0,38%	0,11%
5					low	4,04%	0,82%	–	0,37%	0,28%	0,08%
6				0.30	high	6,32%	0,88%	–	0,41%	0,33%	0,09%
7					medium	5,32%	0,71%	–	0,36%	0,27%	0,08%
8					low	3,86%	0,44%	–	0,30%	0,18%	0,05%
9				0.40	high	6,83%	0,68%	–	0,33%	0,26%	0,07%
10					medium	5,69%	0,53%	–	0,29%	0,20%	0,06%
11					low	4,03%	0,29%	–	0,24%	0,13%	0,03%
12				0.50	high	7,60%	0,61%	–	0,27%	0,23%	0,06%
13					medium	6,31%	0,45%	–	0,23%	0,17%	0,05%
14					low	4,43%	0,23%	–	0,20%	0,10%	0,03%
15	30	4	het.	0.20	medium	5,33%	1,15%	–	0,66%	0,37%	0,11%
16			10	het.		5,78%	0,95%	–	0,43%	0,34%	0,10%
17	18	4	10	0.20	medium	5,25%	1,25%	–	0,72%	0,26%	0,06%
18	60	3		0.20		5,51%	0,24%	–	0,05%	0,19%	0,06%
19		4				5,51%	1,01%	–	0,23%	0,40%	0,14%
20		5				5,51%	1,37%	–	0,49%	0,50%	0,19%

Table B.2: arpg of different allocation methods in individual experiments under scarce supply.

Scen.	Cust.	Levels	Demand	CV	Profit het.	Per commit	Clust. [1]	Det. Theil	Stoch. Theil	Clust. [2]	Clust. [3]
1	30	4	10	0.10	high	13,97%	4,97%	1,29%	0,74%	1,19%	0,37%
2					medium	12,46%	4,52%	1,06%	0,71%	1,14%	0,36%
3					low	10,11%	3,68%	0,87%	0,72%	0,87%	0,27%
4				0.20	high	12,65%	2,74%	1,79%	0,49%	0,88%	0,25%
Bl.					medium	11,04%	2,39%	1,46%	0,44%	0,79%	0,23%
5					low	8,55%	1,78%	1,12%	0,44%	0,60%	0,16%
6				0.30	high	12,21%	1,73%	2,30%	0,32%	0,64%	0,17%
7					medium	10,51%	1,43%	1,87%	0,28%	0,54%	0,15%
8					low	7,88%	0,95%	1,37%	0,27%	0,40%	0,10%
9				0.40	high	12,41%	1,26%	2,84%	0,24%	0,48%	0,12%
10					medium	10,61%	1,01%	2,30%	0,20%	0,39%	0,11%
11					low	7,83%	0,59%	1,64%	0,18%	0,27%	0,07%
12				0.50	high	13,02%	1,07%	3,44%	0,19%	0,40%	0,09%
13					medium	11,12%	0,82%	2,78%	0,16%	0,32%	0,08%
14					low	8,17%	0,45%	1,93%	0,13%	0,20%	0,05%
15	30	4	het.	0.20	medium	11,01%	2,41%	1,78%	0,78%	0,76%	0,22%
16			10	het.		11,25%	1,97%	1,68%	0,41%	0,69%	0,20%
17	18	4	10	0.20	medium	10,87%	2,63%	1,45%	0,77%	0,55%	0,11%
18	60	3		0.20		11,37%	0,50%	1,20%	0,05%	0,41%	0,14%
19		4				11,37%	2,13%	1,38%	0,23%	0,84%	0,29%
20		5				11,37%	2,90%	1,55%	0,51%	1,04%	0,40%

Table B.3: arpg of different allocation methods in individual experiments under ample supply.

Scen.	Cust.	Levels	Demand	CV	Profit het.	Per commit	Clust. [1]	Det. Theil	Stoch. Theil	Clust. [2]	Clust. [3]
1	30	4	10	0.10	high	0,19%	0,04%	–	0,34%	0,03%	0,01%
2					medium	0,14%	0,02%	–	0,33%	0,01%	0,01%
3					low	0,07%	0,00%	–	0,24%	0,00%	0,00%
4				0.20	high	0,65%	0,10%	–	0,48%	0,05%	0,02%
Bl.					medium	0,48%	0,06%	–	0,42%	0,03%	0,01%
5					low	0,27%	0,01%	–	0,30%	0,01%	0,00%
6				0.30	high	1,35%	0,15%	–	0,47%	0,07%	0,03%
7					medium	1,01%	0,10%	–	0,42%	0,04%	0,02%
8					low	0,59%	0,03%	–	0,32%	0,01%	0,00%
9				0.40	high	2,22%	0,20%	–	0,41%	0,08%	0,03%
10					medium	1,68%	0,13%	–	0,35%	0,05%	0,02%
11					low	1,00%	0,04%	–	0,29%	0,01%	0,00%
12				0.50	high	3,21%	0,23%	–	0,34%	0,09%	0,03%
13					medium	2,48%	0,16%	–	0,28%	0,05%	0,02%
14					low	1,51%	0,06%	–	0,24%	0,02%	0,00%
15	30	4	het.	0.20	medium	0,48%	0,06%	–	0,56%	0,03%	0,01%
16			10	het.		1,15%	0,09%	–	0,44%	0,04%	0,02%
17	18	4	10	0.20	medium	0,45%	0,06%	–	0,67%	0,02%	0,01%
18	60	3				0,49%	0,01%	–	0,05%	0,01%	0,00%
19		4				0,49%	0,04%	–	0,22%	0,02%	0,01%
20		5				0,49%	0,07%	–	0,47%	0,03%	0,02%

Appendix C

Appendix to Chapter 4

C.1 Proof of Analytical Results

Proof of Proposition 4.1. From Equation (4.1) and the state transitions (Definition 4.1) we obtain the following expression of the penalty for a customer l:

$$C_l(X_{l,R}, Y_{l,R}) = p_l \left[\beta_l - \frac{y_{l,R} + \min\{a_{l,R}, D_{l,R}\}}{x_{l,R} + D_{l,R}} \right]^+ .$$

From this and the demand distribution of the customer, we can obtain the following expression for the expected penalty of a customer:

$$\begin{aligned} \mathbb{E}\left[C_l(X_{l,R}, Y_{l,R})\right] = p_l \int_0^{a_{l,R}} \left[\beta_l - \frac{y_{l,R} + d_{l,R}}{x_{l,R} + d_{l,R}} \right]^+ \cdot f_{l,R}(d_{l,R}) \, \mathrm{d}d_{l,R} + \\ p_l \int_{a_{l,R}}^{\infty} \left[\beta_l - \frac{y_{l,R} + a_{l,R}}{x_{l,R} + d_{l,R}} \right]^+ \cdot f_{l,R}(d_{l,R}) \, \mathrm{d}d_{l,R} \end{aligned} \quad \text{(C.1)}$$

The first part of Equation C.1 is only different from zero, when $d_{l,R} \geq d_{min,l} = \frac{\beta_l x_{l,R} - y_{l,R}}{1 - \beta_l}$; the second part of is only different from zero for $d_{l,R} \geq d_{max,l}(a_{l,R}) = \frac{y_{l,R} + a_{l,R}}{\beta_l} - x_{l,R}$. Straightforward case differentiation then leads to Proposition 4.1 Part 1.

For Proposition 4.1 Part 2 we differentiate the elements of Part 1 separately. The differential of a parameter-depended integral $G(x) = \int_{u(x)}^{v(x)} g(x,y)\ \mathrm{d}y$ is

$$G'(x) = -g(x,u)u' + g(x,v)v' + \int_{u(x)}^{v(x)} \frac{\mathrm{d}}{\mathrm{d}x} g(x,y)\ \mathrm{d}y.$$

Hence, for

$$\int_{d_{max,l}(a_{l,R})}^{\infty} p_l \left(\beta_l - \frac{y_{l,R} + a_{l,R}}{x_{l,R} + d_{l,R}}\right) f_{l,R}(d_{l,R})\ \mathrm{d}d_{l,R}$$

we have:

$$\begin{aligned} u' &= \frac{1}{\beta_l} \\ v' &= 0 \\ g(x = a_{l,R}, y = d_{max,l}) &= p_l \left(\beta_l - \frac{y_{l,R} + a_{l,R}}{x_{l,R} + d_{max,l}(a_{l,R})}\right) f_{l,R}(d_{l,R}(a_{l,R})) = 0 \\ \frac{\mathrm{d}}{\mathrm{d}x} g(x = a_{l,R}, u = d_{l,R}) &= -p_l \frac{f_{l,R}(t)}{x_{l,R} + t} \end{aligned}$$

Consequently, we obtain

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}a_{l,r}} \int_{d_{max,l}(a_{l,R})}^{\infty} p \left(\beta_l - \frac{y_{l,R} + a_{l,R}}{x_{l,R} + d_{l,R}}\right) f_{l,R}(d_{l,R})\ \mathrm{d}d_{l,R} \\ = -p_l \int_{d_{max,l}(a_{l,R})}^{\infty} \frac{1}{x_{l,R} + d_{l,R}} f_{l,R}(d_{l,R})\ \mathrm{d}d_{l,R} \end{aligned} \tag{C.2}$$

Obviously,

$$\frac{\mathrm{d}}{\mathrm{d}a_{l,r}} \int_0^{d_{min,l}} p \left(\beta_l - \frac{y_{l,R} + d_{l,R}}{x_{l,R} + d_{l,R}}\right) f_{l,R}(d_{l,R})\ \mathrm{d}d_{l,R} = 0. \tag{C.3}$$

For

$$\int_0^{a_{l,R}} p \left(\beta_l - \frac{y_{l,R} + d_{l,R}}{x_{l,R} + d_{l,R}}\right) f_{l,R}(d_{l,R})\ \mathrm{d}d_{l,R}$$

we have

$$\begin{aligned} u' &= 0 \\ v' &= 1 \end{aligned}$$

$$g(x = a_{l,R}, v = a_{l,R}) = p\left(\beta_l - \frac{y_{l,R} + a_{l,R}}{x_{l,R} + a_{l,R}}\right) f_{l,R}(a_{l,R})$$
$$\frac{\mathrm{d}}{\mathrm{d}x} g(x = a_{l,R}, y = d_{l,R}) = 0$$

and we obtain

$$\frac{\mathrm{d}}{\mathrm{d}a_{l,R}} \int_0^{a_{l,R}} p\left(\beta_l - \frac{y_{l,R} + d_{l,R}}{x_{l,R} + d_{l,R}}\right) f_{l,R}(d_{l,R})\, \mathrm{d}d_{l,R} = p\left(\beta_l - \frac{y_{l,R} + a_{l,R}}{x_{l,R} + a_{l,R}}\right) f_{l,R}(a_{l,R}). \tag{C.4}$$

Similarly, for

$$\int_{a_{l,R}}^{\infty} p\left(\beta_l - \frac{y_{l,R} + a_{l,R}}{x_{l,R} + d_{l,R}}\right) f_{l,R}(d_{l,R})\, \mathrm{d}d_{l,R},$$

we obtain

$$u' = 1$$
$$v' = 0$$
$$g(x = a_{l,R}, u = a_{l,R}) = p\left(\beta_l - \frac{y_{l,R} + a_{l,R}}{x_{l,R} + a_{l,R}}\right) f_{l,R}(a_{l,R})$$
$$\frac{\mathrm{d}}{\mathrm{d}x} g(x = a_{l,R}, y = d_{l,R}) = -p_l \frac{f_{l,R}(t)}{x_{l,R} + t}.$$

This results in

$$\frac{\mathrm{d}}{\mathrm{d}a_{l,R}} \int_{a_{l,R}}^{\infty} p\left(\beta_l - \frac{y_{l,R} + a_{l,R}}{x_{l,R} + d_{l,R}}\right) f_{l,R}(d_{l,R})\, \mathrm{d}d_{l,R} =$$
$$- p\left(\beta_l - \frac{y_{l,R} + a_{l,R}}{x_{l,R} + a_{l,R}}\right) f_{l,R}(a_{l,R}) - p_l \int_{a_{l,R}}^{\infty} \frac{1}{x_{l,R} + d_{l,R}} f_{l,R}(d_{l,R})\, \mathrm{d}d_{l,R}. \tag{C.5}$$

Case 1 of Part 1 results in (C.2); Case 2 results also in (C.2) as (C.3) is zero. This leads to Case 1 of Part 2. Case 3 of Part 1 results in (C.4) and (C.5), which together reveal Case 2 of Part 2. □

Proof of Lemma 4.1. Sums of convex functions are convex; a continuous and a twice-differentiable function is convex, if its second derivative is non-negative (Boyd and Vandenberghe, 2004).

Note that, although $\mathbb{E}\, C_l(X_{l,R+1}, Y_{l,R+1})$ is a piecewise function, it is continuous and differentiable, because $d_{max,l}(d_{min,l}) = d_{min,l}$. Straightforward computations from Proposition 4.1 reveal that

$$\frac{\mathrm{d}}{\mathrm{d}^2 a_{l,R}} \mathbb{E}\, C_l(X_{l,R+1}, Y_{l,R+1}) = \begin{cases} p_l \frac{1}{a_{l,R}+y_{l,R}} f_{l,R}(d_{max}(a_{l,R})) & \text{if } a_{l,R} \geq d_{min,l} \\ p_l \frac{1}{a_{l,R}+x_{l,R}} f_{l,R}(a_{l,R}) & \text{else.} \end{cases}$$

By definition, $f_{l,R}(x)$ and all other parameters are non-negative. Therefore

$$\frac{\mathrm{d}}{\mathrm{d}^2 a_{l,R}} \mathbb{E}\, C_l(X_{l,R+1}, Y_{l,R+1}) \geq 0,$$

and $\mathbb{E}\, C_l(X_{l,R+1}, Y_{l,R+1})$ is twice-differentiable with respect to $a_{l,R}$, which concludes the proof. □

Proof of Proposition 4.2. Straightforward from the fact that $\frac{\mathrm{d}}{\mathrm{d}y_{l,R}} \lambda_{l,R}(a_{l,t}) = 0$ for all $a_{l,R} \leq d_{min,l}$. □

Proof of Proposition 4.3. For now, assume $t = R - 1$ and, for ease of notation, set $V_{R+1} = C_l(x_{l,R+1}, y_{l,R+1})$. The allocation in $a_{l,R-1}$ only affects the fulfilled demand in R (cf. Definition 4.1). Thus, $\frac{\mathrm{d}u}{\mathrm{d}a_{l,R-1}}$ is zero in all dimensions but $y_{l,R}$. Consequently, using the chain differentiation rule, we can show that:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}a_{l,R-1}} \mathbb{E}\, V_R(u(\boldsymbol{s}_{R-1}, \boldsymbol{a}_{R-1}, \boldsymbol{D}_{R-1})) &= \mathbb{E}\, \frac{\mathrm{d}}{\mathrm{d}a_{l,R-1}} V_R(u(\boldsymbol{s}_{R-1}, \boldsymbol{a}_{R-1}, \boldsymbol{D}_{R-1})) \\ &= \mathbb{E}\, \frac{\mathrm{d}u}{\mathrm{d}a_{l,R-1}} \frac{\mathrm{d}}{\mathrm{d}u} V_R(u(\boldsymbol{s}_{R-1}, \boldsymbol{a}_{R-1}, \boldsymbol{D}_{R-1})) \\ &= \mathbb{E}\, \mathbb{1}[D_{l,R-1} \geq a_{l,R-1}] \frac{\mathrm{d}}{\mathrm{d}y_{l,R}} V_R(\boldsymbol{S}_R) \end{aligned} \tag{C.6}$$

Equation C.6 shows that the derivative of the penalty function in period $R-1$ directly corresponds with the derivative of the penalty function in the subsequent period R. Here, $\boldsymbol{S}_R$ denotes the stochastic state in period R.

By including the state transition to period $R+1$ and again applying the chain rule together with the fact that $\frac{\mathrm{d}u}{\mathrm{d}y_{l,R}}$ is zero in all dimensions but $y_{l,R+1}$ (cf. Definition 4.1), we can obtain a detailed formulation for $\frac{\mathrm{d}}{\mathrm{d}y_{l,R}} V_R(\boldsymbol{S}_R)$.

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}y_{l,R}} V_R(\boldsymbol{S}_R) &= \frac{\mathrm{d}}{\mathrm{d}y_{l,R}} \mathbb{E}\, V_{R+1}(u(\boldsymbol{S}_R, \boldsymbol{a}_R^*, \boldsymbol{D}_R)) \\ &= \mathbb{E}\, \frac{\mathrm{d}}{\mathrm{d}y_{l,R+1}} V_{R+1}(u) + \frac{\mathrm{d}\boldsymbol{a}_R^*}{\mathrm{d}y_{l,R}} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{a}_R^*} V_{R+1}(u). \end{aligned} \tag{C.7}$$

(C.7) has two parts: the first part covers how fulfilled demand directly changes the state in R and the second part covers how the fulfilled demand in R affects the optimal allocation in R and with it the penalty. We can show that under an optimal allocation in period R the second part of (C.7) resolves to zero:

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{a}_R^*}{\mathrm{d}y_{l,R}} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{a}_R^*} V_{R+1}(u) &= \sum_{m \in \mathcal{L}} \frac{\mathrm{d}a_{m,R}^*}{\mathrm{d}y_{l,R}} \frac{\mathrm{d}}{\mathrm{d}a_{m,R}^*} V_{t+2}(u) \\ &= \sum_{m \in A_\lambda} \underbrace{\frac{\mathrm{d}a_{m,R}^*}{\mathrm{d}y_{l,R}}}_{=0} \frac{\mathrm{d}}{\mathrm{d}a_{m,R}^*} V_{R+1}(u) + \sum_{m \in \mathcal{L} \setminus A_\lambda} \frac{\mathrm{d}a_{m,R}^*}{\mathrm{d}y_{l,R}} \underbrace{\frac{\mathrm{d}}{\mathrm{d}a_{m,R}^*} V_{R+1}(u)}_{=\lambda} \end{aligned} \tag{C.8}$$

$$= \lambda \underbrace{\sum_{m \in \mathcal{L} \setminus A_\lambda} \frac{\mathrm{d}a_{m,R}^*}{\mathrm{d}y_{l,R}}}_{=0} = 0 \tag{C.9}$$

(C.8) separates the customers receiving an allocation from those who's allocation is bound to zero (set A_λ). Theorem 4.1 we know that the allocation to customers in A_λ does not change with a marginal change in their marginal penalty, so $\frac{\mathrm{d}a_{m,R}^*}{\mathrm{d}y_{l,R}} = 0$. Customer receiving an allocation have the same marginal penalty, hence, $\frac{\mathrm{d}}{\mathrm{d}a_{m,R}^*} V_{R+1}(u) = \lambda$. As supply is constrained, the sum of all allocations is constant, and thus the sum of marginal changes in the allocations to the customer is equal to zero.

Then, we can simplify (C.7) with Equation (4.1) to:

$$\frac{\mathrm{d}}{\mathrm{d}y_{l,R}} V_R(\boldsymbol{S}_R) = \mathbb{E}\, \frac{\mathrm{d}}{\mathrm{d}y_{l,R+1}} V_{R+1}(u) = \mathbb{E}\, \frac{-p_l}{X_{l,R+1}} \cdot \mathbb{1}\left[\frac{Y_{l,R+1}}{X_{l,R+1}} \leq \beta_l\right].$$

Combining this with (C.6) gives Proposition 4.3 for period $R-1$. Which proves that Theorem 4.2 holds for $R-1$. Backward induction shows that Proposition 4.3 holds for any period $t < R-1$. □

Proof of Lemma 4.2. A continuous, twice differentiable function of several variables is convex if and only if its Hessian matrix of second partial derivatives is positive semidefinite (Boyd and Vandenberghe, 2004).

With Proposition 4.3 we know that

$$\mathbb{E}\,\frac{\mathrm{d}}{\mathrm{d}a_{l,t}}V_{t+1}(u(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{D}_t)) = \int_{a_{l,t}}^{\infty}\mathbb{E}\,\frac{-p_l}{X_{l,R+1}}\mathbb{1}\left[\frac{Y_{l,R+1}}{X_{l,R+1}}\leq\beta_l\right]\cdot f_l(d_{l,t})\,\mathrm{d}d_{l,t}.$$

Hence the second derivative can be evaluated as:

$$\begin{aligned}\mathbb{E}\,\frac{\mathrm{d}}{\mathrm{d}^2a_{l,t}}V_{t+1}(u(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{D}_t)) &= \frac{\mathrm{d}}{\mathrm{d}a_{l,t}}\int_{a_{l,t}}^{\infty}\mathbb{E}\,\frac{-p_l}{X_{l,R+1}}\mathbb{1}\left[\frac{Y_{l,R+1}}{X_{l,R+1}}\leq\beta_l\right]\cdot f_l(d_{l,t})\,\mathrm{d}d_{l,t}\\ &= -\,\mathbb{E}\,\frac{-p_l}{X_{l,R+1}}\mathbb{1}\left[\frac{Y_{l,R+1}}{X_{l,R+1}}\leq\beta_l\right]\cdot f_l(d_{l,t})\geq 0. \qquad \text{(C.10)}\end{aligned}$$

As all derivatives $\mathbb{E}\,\frac{\mathrm{d}}{\mathrm{d}a_{l,t}}\frac{\mathrm{d}}{\mathrm{d}a_{m,t}}V_{t+1}(u(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{D}_t))$ for $l,m\in\mathcal{L}$, $l\neq m$ are zero, the Hessian is diagonal matrix with non-negative entries. Consequently, the matrix is positive semidefinite which concludes the proof. □

C.2 LP-Formulations

MSLAP

$$\begin{aligned}&\min Z\\ \textbf{subject to: } &Z\geq\beta_l-\frac{a_{l,t}+y_{l,t}}{\mu_{l,t}+x_{l,t}} && \forall l\in\mathcal{L}\\ &[s_t+i_t]^+\geq\sum_{l\in\mathcal{L}}a_{l,t}\end{aligned}$$

MPAP

$$\begin{aligned}&\min\sum_{l\in\mathcal{L}}Z_l\\ \textbf{subject to: } &Z_l\geq p_l\left[\beta_l-\frac{a_{l,t}+y_{l,t}}{\mu_{l,t}+x_{l,t}}\right] && \forall l\in\mathcal{L}\end{aligned}$$

$$Z_l \geq 0 \qquad \forall l \in \mathcal{L}$$

$$[s_t + i_t]^+ \geq \sum_{l \in \mathcal{L}} a_{l,t}$$

Appendix D

Appendix to Chapter 5

D.1 Algorithm for the Clustering Allocation

for $m \in \mathcal{I}_K$ **do**

Let $C_{c,m} = \{l \mid l \in \mathcal{S}_m, l \text{ is in cluster } c\} \subset \mathcal{S}_m$ denote the set of nodes that belong to cluster $c \in \{1, \ldots, D\}$, where D is the number of clusters.

Set

$$\mu^c_{m,t} := \sum_{l \in C_{c,m}} \mu_{l,t} \tag{D.1}$$

$$\sigma^c_{m,t} := \sum_{l \in C_{c,m}} \sigma_{l,t} \tag{D.2}$$

$$p^c_{m,t} := \frac{1}{\mu^c_{m,t}} \sum_{l \in C_{c,m}} \mu_{l,t} p_{l,t} \tag{D.3}$$

end for

for $n \in \mathcal{I}_h, h \in \{K-2, \ldots, 0\}$ **do**

Let $C_{c,n} = \{(m, c') \mid m \in \mathcal{S}_n, c' \in \{1, \ldots, D\}, (m, c') \text{ is in cluster } c\}$ denote the set of clusters of nodes $m \in \mathcal{S}_n$ belonging node n's cluster c.

Set

$$\mu^c_{n,t} := \sum_{(m,c') \in C_{c,n}} \mu^c_{m,t}$$

$$\sigma^c_{n,t} := \sum_{(m,c')\in C_{c,n}} \sigma^c_{m,t}$$
$$p^c_{n,t} := \frac{1}{\mu^c_{n,t}} \sum_{(m,c')\in C_{c,n}} \mu^c_{m,t} p^c_{m,t}.$$

end for

for $n \in \mathcal{I}_h, h \in \{0, \dots, K-2\}$ **do**

(Determine the local optimal allocation)

Solve

$$\min \sum_{m\in\mathcal{S}_n} \sum_{c\in\{1,\dots,D\}} \int_{x^c_{m,t}}^{\infty} p^c_{m,t}(d^c_{m,t} - x^c_{m,t}) g^c_{m,t} \, \mathrm{d}d^c_{m,t}$$

s.t.

$$\sum_{m\in\mathcal{S}_n} \sum_{c\in\{1,\dots,D\}} x^c_{m,t} \le x_n$$
$$x^c_{m,t} \ge 0 \qquad \forall m \in \mathcal{S}_n, c \in \{1, \dots, D\}$$

where $g^c_{m,t}$ is the cluster's aggregated demand distribution with mean $\mu^c_{n,t}$ and standard deviation $\sigma^c_{n,t}$.

Set $x_{m,t} = \sum_{c\in\{1,\dots,D\}} x^c_{m,t}$ for all $m \in \mathcal{S}_n$.

end for

return $(x_{m,t})_{m\in\mathcal{I}_K}$

Bibliography

Abbasi, B., Hosseinifard, Z., Alamri, O., Thomas, D., and Minas, J. P. (2017). Finite time horizon fill rate analysis for multiple customer cases. *Omega*, 76:1–17.

Alamri, O., Abbasi, B., Minas, J. P., and Zeephongsekul, P. (2017). Service-level agreements: Ready-rate analysis with lump-sum and linear penalty structures. *Journal of the Operational Research Society*, 65(12):1–15.

Alemany, M., Lario, F.-C., Ortiz, A., and Gómez, F. (2013). Available-to-promise modeling for multi-plant manufacturing characterized by lack of homogeneity in the product: An illustration of a ceramic case. *Applied Mathematical Modelling*, 37(5):3380–3398.

Allen, S. C. (1985). Redistribution of total stock over several user locations. *Naval Research Logistics*, 5(4):337–345.

Arslan, H., Graves, S. C., and Roemer, T. A. (2007). A single-product inventory model for multiple demand classes. *Management Science*, 53(9):1486–1500.

Avrahami, A., Herer, Y. T., and Levi, R. (2014). Matching supply and demand: Delayed two-phase distribution at yedioth group—models, algorithms, and information technology. *Interfaces*, 44(5):445–460.

Ball, M. O., Chen, C.-Y., and Zhao, Z.-Y. (2004). Available to promise. In Simchi-Levi, D., Wu, S. D., and Shen, Z.-J., editors, *Handbook of quantitative supply chain analysis*, International series in operations research & management science, pages 447–484. Kluwer, Boston.

Barut, M. and Sridharan, V. (2005). Revenue management in order-driven production systems. *Decision sciences*, 36(2):287–316.

Bellman, R. (1957). *Dynamic programming*. Princeton university press, Princeton.

Bertsekas, D. P. (2005). *Dynamic programming and optimal control*, volume 1 of *Athena Scientific optimization and computation series*. Athena Scientific, Belmont, MA, 3 edition.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press, Cambridge.

Bretthauer, K. M. and Shetty, B. (2002a). The nonlinear knapsack problem – algorithms and applications. *European Journal of Operational Research*, 138(3):459–472.

Bretthauer, K. M. and Shetty, B. (2002b). The nonlinear knapsack problem–algorithms and applications. *European Journal of Operational Research*, 138(3):459–472.

Caldentey, R. and Wein, L. M. (2006). Revenue management of a make-to-stock queue. *Operations Research*, 54(5):859–875.

Cano-Belmán, J. and Meyr, H. (2019). Deterministic allocation models for multi-period demand fulfillment in multi-stage customer hierarchies. *Computers & Operations Research*, 101:76–92.

Chen, C.-M. and Thomas, D. J. (2018). Inventory allocation in the presence of service–level agreements. *Production and Operations Management*, 27(3):553–577.

Chen, J. and Dong, M. (2014). Available-to-promise-based flexible order allocation in ato supply chains. *International Journal of Production Research*, 52(22):6717–6738.

Chiang, D. M.-H. and Wei-Di Wu, A. (2011). Discrete-order admission atp model with joint effect of margin and order size in a mto environment. *International Journal of Production Economics*, 133(2):761–775.

Chopra, S. and Meindl, P. (2010). *Supply chain management: Strategy, planning, and operation*. Prentice Hall, Boston.

Chotikapanich, D. (1993). A comparison of alternative functional forms for the lorenz curve. *Economics Letters*, 41(2):129–138.

Croxton, K. L. (2003). The order fulfillment process. *The International Journal of Logistics Management*, 14(1):19–32.

Deshpande, V., Cohen, M. A., and Donohue, K. (2003). A threshold inventory rationing policy for service-differentiated demand classes. *Management Science*, 49(6):683–703.

Diks, E. B. and de Kok, A. G. (1998). Optimal control of a divergent multi-echelon inventory system. *European Journal of Operational Research*, 111(1):75–97.

Eppler, S. (2015). *Allocation Planning for Demand Fulfillment in Make-to-Stock Industries: A Stochastic Linear Programming Approach*. Dissertation, TU Darmstadt, Darmstadt.

Fleischmann, B. and Meyr, H. (2004). Customer orientation in advanced planning systems. In Dyckhoff, H., Lackes, R., and Reese, J., editors, *Supply chain management and reverse logistics*, pages 297–321. Springer, Berlin and New York.

Fleischmann, M., Kloos, K., Nouri, M., and Pibernik, R. (2019). Single period stochastic hierarchical demand fulfillment. Working Paper.

Framinan, J. M. and Leisten, R. (2010). Available-to-promise (ATP) systems: A classification and framework for analysis. *International Journal of Production Research*, 48(11):3079–3103.

Gössinger, R. and Kalkowski, S. (2015). Robust order promising with anticipated customer response. *International Journal of Production Economics*, 170:529–542.

Guhlich, H., Fleischmann, M., and Stolletz, R. (2015). Revenue management approach to due date quoting and scheduling in an assemble-to-order production system. *OR Spectrum*, 37(4):951–982.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666.

Jeong, B., Sim, S.-B., Jeong, H.-S., and Kim, S.-W. (2002). An available-to-promise system for tft lcd manufacturing in supply chain. *Computers & Industrial Engineering*, 43(1):191–212.

Jung, H. (2010). An available-to-promise model considering customer priority and variance of penalty costs. *The International Journal of Advanced Manufacturing Technology*, 49(1):369–377.

Katok, E., Thomas, D., and Davis, A. (2008). Inventory service-level agreements as coordination mechanisms: The effect of review periods. *Manufacturing & Service Operations Management*, 10(4):609–624.

Ketikidis, P. H., Lenny Koh, S., Gunasekaran, A., and Pibernik, R. (2006). Managing stock-outs effectively with order fulfilment systems. *Journal of manufacturing technology management*, 17(6):721–736.

Kilger, C. and Meyr, H. (2008). Demand fulfilment and atp. In Stadtler, H. and Kilger, C., editors, *Supply chain management and advanced planning*, pages 181–198. Springer, Berlin.

Kilger, C. and Meyr, H. (2015). Demand fulfilment and ATP. In Stadtler, H., Kilger, C., and Meyr, H., editors, *Supply Chain Management and Advanced Planning*, pages 181–198. Springer, Berlin, Heidelberg.

Kilger, C. and Schneeweiss, L. (2000). Demand fulfilment and ATP. In Stadtler, H. and Kilger, C., editors, *Supply chain management and advanced planning*, pages 135–148. Springer, Berlin and New York.

Kloos, K. (2019). Managing service-level contracts in sales hierarchies. Working Paper.

Kloos, K. and Pibernik, R. (2020). Allocation planning under service-level contracts. *European Journal of Operational Research*, 280(1):208–218.

Kloos, K., Pibernik, R., and Schulte, B. (2018). Allocation planning in sales hierarchies with stochastic demand and service-level targets. *OR Spectrum*.

Liang, L. and Atkins, D. (2013). Designing service level agreements for inventory management. *Production and Operations Management*, 26(1):1102–1117.

Lin, F.-R. and Shaw, M. J. (1998). Reengineering the order fulfillment process in supply chain networks. *International Journal of Flexible Manufacturing Systems*, 10(3):197–229.

Meyr, H. (2009). Customer segmentation, allocation planning and order promising in make-to-stock production. *OR Spectrum*, 31(1):229–256.

Mookherjee, D. (2006). Decentralization, hierarchies, and incentives: A mechanism design perspective. *Journal of Economic Literature*, 44(2):367–390.

Pibernik, R. (2005). Advanced available-to-promise: Classification, selected methods and requirements for operations and inventory management. *International Journal of Production Economics*, 93-94:239–252.

Pibernik, R. and Yadav, P. (2008). Dynamic capacity reservation and due date quoting in a make-to-order system. *Naval Research Logistics*, 55(7):593–611.

Pibernik, R. and Yadav, P. (2009). Inventory reservation and real-time order promising in a make-to-stock system. *OR Spectrum*, 31(1):281–307.

Powell, W. B. (2011). *Approximate dynamic programming: Solving the curses of dimensionality*. Wiley series in probability and statistics. Wiley, Hoboken, NJ, 2. ed. edition.

Pradhan, S. and Verma, P. (2012). *Global available to promise with SAP: Functionality and configuration*. Galileo Press, Bonn and Boston, 1st ed. edition.

Protopappa-Sieke, M., Sieke, M. A., and Thonemann, U. W. (2016). Optimal two-period inventory allocation under multiple service level contracts. *European Journal of Operational Research*, 252(1):145–155.

Quante, R. (2009). *Management of stochastic demand in make-to-stock manufacturing*, volume 37 of *Forschungsergebnisse der Wirtschaftsuniversität Wien*. Lang, Frankfurt am Main.

Quante, R., Fleischmann, M., and Meyr, H. (2009a). A stochastic dynamic programming approach to revenue management in a make-to-stock production system. *ERIM Report Series Reference No. ERS-2009-015-LIS*.

Quante, R., Meyr, H., and Fleischmann, M. (2009b). Revenue management and demand fulfillment: matching applications, models, and software. *OR Spectrum*, 31(1):31–62.

Roitsch, M. and Meyr, H. (2015). Oil industry. In Stadtler, H., Kilger, C., and Meyr, H., editors, *Supply Chain Management and Advanced Planning*, pages 443–458. Springer, Berlin, Heidelberg.

Ross, S. M. (2006). *Simulation*. Elsevier, Amsterdam.

Ruszczyński, A. P. (2006). *Nonlinear optimization*, volume 13. Princeton university press, Princeton.

Samii, A.-B., Pibernik, R., Yadav, P., and Vereecke, A. (2012). Reservation and allocation policies for influenza vaccines. *European Journal of Operational Research*, 222(3):495–507.

Sarstedt, M. and Mooi, E. (2019). Cluster analysis. In *A concise guide to market research*, pages 301–354. Springer, Berlin, Heidelberg.

Schneeweiss, C. A. (2003). *Distributed Decision Making*. Springer, Berlin and New York.

Schulte, B. and Pibernik, R. (2016). Service differentiation in a single-period inventory model with numerous customer classes. *OR Spectrum*, 38(4):921–948.

Sieke, M. A., Seifert, R. W., and Thonemann, U. W. (2012). Designing service level contracts for supply chain coordination. *Production and Operations Management*, 21(4):698–714.

Stadtler, H. and Kilger, C., editors (2008). *Supply chain management and advanced planning: Concepts, models, software, and case studies*. Springer, Berlin.

Stadtler, H., Kilger, C., and Meyr, H., editors (2015). *Supply Chain Management and Advanced Planning*. Springer, Berlin, Heidelberg.

Talluri, K. and van Ryzin, G. (1999). A randomized linear programming method for computing network bid prices. *Transportation Science*, 33(2):207–216.

Thomas, D. J. (2005). Measuring item fill-rate performance in a finite horizon. *Manufacturing & Service Operations Management*, 7(1):74–80.

Tiemessen, H., Fleischmann, M., van Houtum, G. J., van Nunen, J., and Pratsini, E. (2013). Dynamic demand fulfillment in spare parts networks with multiple customer classes. *European Journal of Operational Research*, 228(2):367–380.

Van Zandt, T. (1995). Hierarchical computation of the resource allocation problem. *European Economic Review*, 39(3):700–708.

Van Zandt, T. (2003). Real-time hierarchical resource allocation with quadratic costs. *CEPR Discussion Paper*, (4022).

Van Zandt, T. and Radner, R. (2001). Real-time decentralized information processing and returns to scale. *Economic Theory*, 17(3):545–575.

Vogel, S. (2014). *Demand fulfillment in multi-stage customer hierarchies*. Produktion und Logistik. Springer Gabler, Wiesbaden.

Vogel, S. and Meyr, H. (2015). Decentral allocation planning in multi-stage customer hierarchies. *European Journal of Operational Research*, 246(2):462–470.

Yang, Y. and Fleischmann, M. (2013). A safety margin model for revenue management in a make-to-stock production system. Working Paper.

Zipkin, P. H. (1980a). Bounds on the effect of aggregating variables in linear programs. *Operations Research*, 28(2):403–418.

Zipkin, P. H. (1980b). Simple ranking methods for allocation of one resource. *Management Science*, 26(1):34–43.