About the Author

Nicolas Vandeput is a supply chain data scientist specialized in demand forecasting and inventory optimization. He founded his consultancy company SupChains in 2016 and co-founded SKU Science—a smart online platform for supply chain management—in 2018. He enjoys discussing new quantitative models and how to apply them to business reality. Passionate about education, Nicolas is both an avid learner and enjoys teaching at universities: he has taught forecasting and inventory optimization to master students since 2014 in Brussels, Belgium. He published *Data Science for Supply Chain Forecasting* in 2018 and *Inventory Optimization: Models and Simulations* in 2020.

Foreword

What makes a supply chain great? As supply chains are made of people, a frequent answer is *great leadership* is what it takes. Indeed…but *great* is vague. More specifically, what sort of qualities and competencies should a company seek to foster among its supply chain management? The 20th century answer to this question has primarily been reliable, diligent, energetic if not charismatic leaders, capable of organizing the work of thousands of workers, and literally creating the *mass production* supply chains as we know them today.

Yet, with the advent of the barcode reader, supply chains had, by the end of the 20th century, already outgrown the *direct* capabilities of the human mind, even the most talented ones. There are too many SKUs, too many suppliers, too many clients, too many channels to expect management to sort it all out through sheer willpower. Instead of *directly* controlling fine-grained supply chain decisions, companies transitioned toward *indirect* management through software.

In this respect, supply chain management has been lagging behind. Factories have been heavily automated for over two decades—a few verticals such as textiles aside. Warehouses are getting there and will be almost exclusively automated by the end of the 2020s. Transportation still faces the *last mile* problem that resists automation, but within one decade—two at most—autonomous vehicles will be commonplace and deliver the last major productivity gains to be ever observed in supply chains. Beyond this point, blue collar jobs will have almost completely disappeared from supply chains.

Yet, software-driven supply chain management takes a fairly distinctive skill set compared to people-driven supply chain management. Being great at software takes a *hacker* mindset that emphasizes tinkering with programs or machines, experimenting for fun and profit, and treating reverse engineering as much as a learning opportunity as a way to hack your way into a better system. This hacker mindset wasn't part of the recipe for 20th century leadership, and yet, I firmly believe it will be the cornerstone of 21st century leadership.

In this book *Inventory Optimization: Models and Simulations*, Nicolas Vandeput hacks his way through the maze of quantitative supply chain optimizations. This book illustrates how the quantitative optimization of 21st century supply chains should be crafted and executed. This book is based on many years of experience, earned the hard way from the supply chain trenches of large companies.

With usually no more than 10 lines of Python, Nicolas Vandeput revisits classic models and turns mathematical models into actionable pieces of software. Doing so, Nicolas demystifies the discourse of (most?) large software vendors in supply chain who—under the guise of Big Data/AI/Demand Sensing (pick your buzzword)—end up delivering *less* than what Nicolas achieves with highly accessible tools, be it Excel or Python.

https://doi.org/10.1515/9783110673944-202

Also, after revising the classics, the last five chapters, starting from “Beyond Normality,” venture into hard problems that do not have nice, closed-form, analytical solutions. Yet, once again, the book delivers the hands-on demonstration that even tough problems can be brute-forced to some extent with histograms, kernels or Monte Carlo algorithms.

The optimization of the 21st century supply chain will be driven by hackers. Hackers share insights, but more importantly, they share *code* as well. Nicolas Vandeput is at the forefront of a new and better way of doing supply chains, and thanks to a richly illustrated book, where every single situation gets its own illustrating code snippet, *so could you*.

Joannes Vermorel
March 2020

Contents

Introduction

Supply chains are complex.

In the current business landscape of our global economy, supply chains and businesses are made of international networks of suppliers and clients. As the competition became fiercer on quality and price, the pressure increased on supply chain execution. And its management only grew more complex. The catalog of products each company is offering is ever expanding—despite their best efforts to keep it limited. This is mainly due to an increased (global) competition pressure; and a wave of customization and product innovation that started during the second half of the 20th century. As businesses are dealing with more products, the average life-cycle is becoming shorter and the demand variability is increasing. On top of this, international supply chains result in longer lead times, imposing more constraints on operational planning.

The complexity of supply chains is also due to all the humans—us!—interacting with each other over long distances, with different tools, capabilities and information. More importantly, even if business leaders strive to align their teams, each actor is pursuing its own objectives often resulting in divergent actions.

For better or for worse, inventories lie at the bottom of these complex supply chains. The central question of **how much inventory is needed**, and **where it is needed**, is often an endless debate among colleagues. Especially when the game of politics drives decisions.

Inventory Done Right

Stocking products is helping companies around the globe to supply their clients on time and provides a buffer against any unforeseen event (we'll discuss this in the second part of the book). Since holding inventory disconnects the production process from the sales process, it allows planners to produce longer production batches decreasing the production costs (we'll discuss this in Chapter 2). In other words, inventory optimization done right reduces overall costs, while optimizing the service level.

Inventory Done Wrong

Nevertheless, holding inventory comes with two drawbacks. The first one is of course its cost: inventory is nothing more than sleeping cash that is depreciating over time. It costs money to store products, and it comes with risks. The more you keep, the higher the cost, and the riskier it gets. Will you really be able to sell all these products? There is always the risk of ending up with dead stock. Keeping less inventory might partially prevent the risk of dead stock, but won't help to provide adequate service levels to your clients. Actually, in some cases, inventory management is so flawed that supply chains face *both* low service levels *and* dead stock.

https://doi.org/10.1515/9783110673944-203

Too much inventory will also prevent companies from improving their processes because they don't see any of the process fluctuation anymore. Too much protection against unforeseen events won't incentivize the different process-owners to improve.

Impact of Inventory

Based on the 30th Annual Council of Supply Chain Management Professionals State of Logistics Report,[1] inventory **yearly costs** accounted for around $500 billion in the US in 2018 (around 2% of US GDP). These are more or less equally spread between financial costs, storage costs and other costs (like obsolescence, handling, insurance). We will discuss holding costs in detail in Chapter 2.

As shown in Figure 1, the U.S. Census Bureau (2019) reports that the amount of inventory for manufacturers, wholesalers and retailers has been rather flat for the last four years at around 1.4 months of sales (their all-time low was in 2010, with inventory levels 10 to 15% lower).

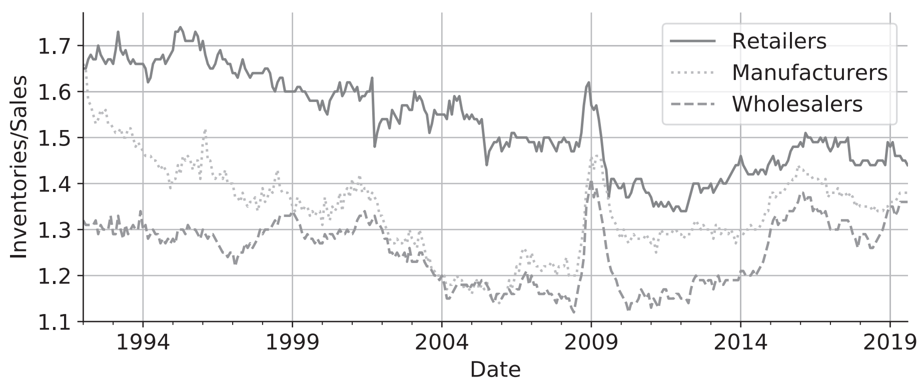

Figure 1: Seasonally Adjusted Inventories/Sales. Source: U.S. Census Bureau (2019).

Inventory is here to stay. We should therefore be serious about optimization—avoiding the trap of bad service levels and dead stock, and reaping the benefits of a properly orchestrated inventory policy. In this book you will see how to optimize your own inventory policies based on various tools and models.

1 Ward et al. (2019).

Inventory Modeling

All models are wrong, but some are helpful.

George Box (1919–2013)

As the amount of inventory grows, so does the need for optimization. Unfortunately, the science of optimizing inventories—keeping the *right* amount of inventory—can become as complex as the different modern supply chains. Even if an inventory policy seems simple—"Buy 4 when you have 5 left"—the models behind these can easily become *massively* complex. In order to solve these policies we have two methods at our disposal: mathematical models (i. e., equations) and simulations.

Models

Since the first inventory optimization model was published in 1913 by Harris, academics have published an ever-growing list of mathematical models to optimize **how much inventory should a supply chain hold**, and **where it should be stored**. Each new set of equations would tackle a supply chain under another set of assumptions. A model would discuss supply chains with perishable products; another one with multiple warehouses; or with a specific seasonality and production constraints; and so on. Unfortunately, even the cutting-edge models won't be enough to *perfectly* describe a supply chain.[2] As explained by Ton de Kok—professor at the Eindhoven University of Technology—*"Inventory models are abstractions that cannot capture all possible actions to balance supply and demand."* We are only at the beginning of our journey of inventory optimization and yet we already know that perfection is out of reach. And most likely will be forever.

Simulations

An equation can't handle the complexity of a modern supply chain, but we can nevertheless *try* to simulate it. Simulations can either be used to test the outcome of a mathematical model; or they can be used to find a *good-enough* policy when no equations are up to the challenge. In the final part of the book, **you will learn to use simulations to optimize inventory policies**. Simulations will allow you to cope with much more complexity than equations, but this will come at a cost: longer computation times and variable results. We will discuss in Chapter 13 how to overcome these limitations.

Limitations and Assumptions

Throughout the book we will pay careful attention to the assumptions (read *limitations*) behind the models (and simulations) we use. We won't be able to create perfect

2 Immanuel Kant (1724–1804), a German philosopher during the Age of Enlightenment, prognosticated that, *"From such crooked wood as that which man is made of, nothing straight can be fashioned."* Can we then really expect our models to be perfect?

simulations or models, but as we understand their weaknesses, we will be able to assess when it is right to use them (and when we should go for another model).

Model Accuracy and Sustainability

Does it make sense then to continue on this path of optimizing inventory policies despite the impossibility to be absolutely correct? Yes. Being correct 95% of the time should be enough to bring substantial savings to any supply chain.[3]

In this book, we will discuss models that deal with enough complexity to properly understand what drives supply chain inventory levels, costs and profits. We will also discuss the assumptions we make, and how we can refine them with some hacks and fit them to reality, while keeping them practical. As supply chain scientists, we have to make trade-offs for our models between complexity and accuracy on one side, and simplicity and practicality on the other. As Jordan Ellenberg explained in his book *How Not to Be Wrong: The Power of Mathematical Thinking*: *"If the universe hands you a hard problem, try to solve an easier one instead, and hope the simple version is close enough to the original problem that the universe doesn't object."*[4] This is exactly what we will do. A complex model will most likely be more accurate resulting in fewer costs and more profits; but less understandable and usable for different users. A simple model, on the other hand, will be less accurate, resulting in higher costs and waste; but might be more usable and sustainable for a business process. At some point, it is better for a model to be 80% accurate but trusted and always used; rather than be 95% accurate and never used nor well understood.

There is a Latin saying, "Vires acquirit eundo," which means, "We gather strength as we go." This represents perfectly the path we will follow in this book: as shown in Figure 2, we will start with a simple model and, as we apply it, we will learn new things. These will allow us to build a stronger model. And so on. We will gather insights and intuitions about supply chains as we go.

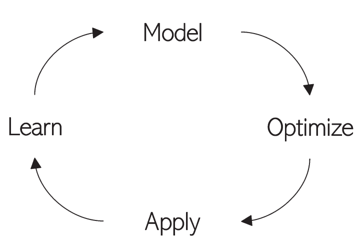

Figure 2: Inventory optimization journey.

3 My friend François Grisay and I love to say at conferences that "95% of the time, safety stock works every time."
4 Ellenberg (2014).

Book Organization

The book is divided into four parts:

I **Deterministic Supply Chains** where our journey starts by discussing common inventory policies (Chapter 1) and then creating the first simple deterministic model (Chapters 2 and 3). This part will be mostly theoretical, but will lay the necessary ground for the next parts.

II **Stochastic Supply Chains** where you will create your first *stochastic* model:[5] first based on stochastic demand only (Chapters 4 and 5), then all together with stochastic lead times (Chapter 6). **You** will code in Chapter 5 your first inventory simulation. Going forward, it will help you to assess the accuracy of each of your models.

III **Advanced Stochastic Models** where you will refine the stochastic model from part II. We will first discuss the expected backorders and fill rate (Chapter 7). Then you will learn to minimize the cost of your inventory policies in Chapter 8, and create policies for non-normal demand (Chapter 9). These three chapters are the most technical of the book, nevertheless they should help you to optimize and refine your inventory policies **beyond** what is usually done in the industry. Finally, you will answer the question "where should we locate inventory" as you will create a model to optimize a *multi-echelon inventory policy*[6] (Chapter 10).

IV **Discrete Inventory Optimization** where you will learn to use simulations to optimize inventory policies. We discuss first a simple newsvendor model (Chapter 11), then we model custom demand probability distributions (Chapter 12) to finally optimize inventory policies with regard to simulations (Chapter 13).

Of course, inventory modeling is not an end. The real goal is to analyze your supply chain, optimize your inventory policy, minimize your costs and maximize your profits. We will often discuss the business implications of the various models, shifting from a scientific mindset to a business one.

Inventory Confusion

Throughout the book we will also discuss major sources of confusion and sub-optimalities in inventory policies.

Vocabulary and definitions First and foremost, inventory management suffers from major confusion due to its terminology: "safety stock," "reorder point," "stock target," "expected inventory"... These might all refer to different concepts depending

5 Stochastic is a fancy word for random. In a stochastic model the outcome is not know in advance but can be estimated by statistical distributions; whereas you always know what will happen in a deterministic model.

6 Multi-echelon inventory policies refer to inventory policies that deal with complex warehouse networks with multiple layers (echelons) of stocking points.

on the software vendor, consultant or colleague you are talking to. We will take the time to define each of these in Parts I and II.

Long lead times As the lead times get longer, the spread between the order up-to level of an inventory policy and the expected average inventory gets bigger. We will discuss these in Chapters 3 and 5.

Review period Often practitioners forget to include the review period of their policies into the computation of safety stock targets (and only look at the lead time). You will see how to include the review period in Chapter 5.

Fill rate vs. Cycle service level A major confusion for practitioners is how to measure the service level of an inventory policy. Most of the safety stock models use the cycle service level whereas most companies record the fill rate. We will define both in Chapter 4 and learn to fit our inventory policies to a fill rate target in Chapter 7.

Optimal service level Finally, the service level targets are often set as arbitrary numbers. You will see in Chapter 8 how to set the optimal service level that will minimize your costs.

Other Resources

In order to help you with the various concepts presented in this book, there is a glossary and a list of symbols at the end of the book. Please note that throughout the book you will see that boldface is used to highlight important text for emphasis.

We will discuss many models, and refer whenever possible to their initial authors as well as current research and helpful resources. You can also check the following general resources about inventory and supply chains:

- *Inventory and Production Management in Supply Chains* by Edward Silver, David Pyke and Douglas Thomas.[7] The authors are well-known professors in the field of inventory management and operation management; their book is most likely one of the main reference for PhD students as it includes a thorough literature review and covers many topics in-depth.
- *Matching Supply with Demand: An Introduction to Operations Management* by Gérard Cachon and Christian Terwiesch.[8] The book offers an introduction to many supply chain subjects: it is great as a textbook for students.
- *Inventory Control* by Sven Axsäter.[9] This (mathematical) reference book covers in detail various inventory models (mainly continuous ones).

7 Silver et al. (2016).
8 Cachon (2018).
9 Axsäter (2015).

– Next to these reference books, you can also check the shorter *Inventory Management in Supply Chain Networks* from Horst Tempelmeier.[10] It also focuses on mathematical models.
– You can also register to the excellent online class *Supply Chain Fundamentals* proposed by MIT on the online platform edX.[11] It covers various inventory optimization models as well as forecast models.
– MIT also offers the online class “Supply Chains for Manufacturing I” on edX.[12] This class covers inventory and forecast models and is given by Stephen Graves and Sean Willems—known for their research work in the field of inventory optimization (we will discuss their models in Chapter 10).

Tools

As a painter needs the proper brush, the inventory analyst needs the proper tools to create her or his model. We will use two tools to build our models, experiment and share our results.

Excel

Excel is the data analyst’s Swiss knife. It will allow you to easily perform simple calculations and plot data. The big advantage of Excel, compared to any programming language, is that you can **see the data**. It is much easier to debug a model or to test a new one if you see how the data is transformed by each step of computation. Therefore, Excel can be a first go-to to experiment with new models or data.

Excel also has many limitations. It won’t perform well on big datasets and will hardly allow us to automate difficult tasks. It is therefore a good tool to test ideas and models on small datasets, or to visualize data, but it is not our preferred tool to create complex or massive models.

Python

Python is a programming language initially published in 1991 by Guido van Rossum, a Dutch computer scientist. If Excel is a Swiss knife, Python is a full army of construction machines awaiting instructions from an analyst. Python will allow you to perform computations on huge datasets in an automated, fast and simple way. It also comes

10 Tempelmeier (2011).
11 www.edx.org/course/supply-chain-fundamentals
12 www.edx.orgns-and-manufacturing-systems-planning-1

with many libraries dedicated to data analysis (pandas), data visualization (Seaborn and Matplotlib), scientific computations (NumPy and SciPy) and machine learning (scikit-learn). These will soon be your best friends, if they aren't already.

Why Python?

We chose Python over other programming languages because it is both user-friendly (it is easy to read and understand) and one of the most used programming languages in the world. In 2019, it was the programming language that was the most googled and it is the most commonly used for machine learning and deep learning.

Should You Start Learning Python?

Yes, you should.

Excel will be perfect to visualize results, perform some simple computations and simple data cleaning tasks. But it won't allow you to scale up your models to bigger datasets nor to easily automate any data cleaning. Excel will also make your life difficult if you want to define customized statistical functions, such as those we will need in the second half of the book.

Many practitioners are afraid to learn a coding language. Everyone knows a colleague who uses some macros/VBA in Excel—maybe you are this colleague—and the complexity of these macros might be frightening to the uninitiated. **Python is much simpler than Excel macros and much more powerful**. As you will see yourself in the following chapters, even the most advanced models won't require so many lines of code or complex functions. It means that you do not have to be an IT genius to use Python. **You can start to use it yourself, today, on your own computer, for free**. Python will give you a definitive edge over anyone using Excel.

Today is a great day to start learning Python. Many resources are available: videos, blogs, articles, books… You can, for example, look for Python courses on the following online platforms:

www.edx.org
www.coursera.org
www.udemy.com
www.datacamp.com

I personally recommend the MIT class: *"Introduction to Computer Science and Programming Using Python,"* available on edX.[13] This will teach you everything you need to know about Python to start using the models presented in this book.

13 www.edx.org/course/introduction-to-computer-science-and-programming-7

You can also look at the Appendix A, where I briefly introduce the most useful Python concepts (and how to install it) in order to help you out with the first code extracts.

Python Libraries

We will use throughout the book some of Python's very well-known libraries. As you can see below, we will use their usual import conventions. And, for the sake of clarity, we won't show the `import` lines over and over in each code extract.

```
import numpy as np
import pandas as pd
import scipy.stats as stats
from scipy.stats import norm, gamma
import matplotlib.pyplot as plt
```

Simplicity vs. Efficiency

The various Python code extracts throughout the book are made with the objective of simplicity and clarity. Simple code is much easier to understand, maintain, share and improve than complex one. This simplification was sometimes done at the expense of efficiency or speed. This means that the codes are not as fast as an experienced Python user could produce, but the implementations are easy to understand—which is the primary goal here.

> *Perfection is finally attained not when there is no longer anything to add, but when there is no longer anything to take away.*
>
> Antoine de Saint-Exupéry (1900–1944)

Other Resources

You can download the Python codes shown in this book on supchains.com/resources-invopt. There is also a glossary (and an index) at the end of the book where you can find a short description of all the specific terms we will use. Do not hesitate to consult it if you are unsure about a term or acronym.

1 Inventory Policies

An inventory policy determines **how much** and **when** a product should be ordered (or produced).[1] Inventory policies determine how the products flow through a supply chain. We can categorize them into two types, based on when the inventory review is done:
- Continuous review policies: an order can be made at any time.
- Periodic review policies: the orders can only be made at specific times.

Let's discuss in detail three of the main inventory policies used in practice.

1.1 Policy #1 – Continuous Review and Reorder Point

In this first inventory policy, we order our products based on a fixed threshold: **as soon as the net inventory reaches the threshold (or goes below), we order a predetermined number of units from our supplier** (or launch a production batch). This threshold is called the **reorder point** or ROP.

On-hand inventory

Inventory physically available for a client to buy.

Backorders

Backlog of open orders that are not yet fulfilled. This happens when you do not have enough on-hand inventory to fulfill orders directly and the orders are not lost.

Net inventory

Inventory level including: available on-hand inventory and *in-transit* inventory, minus backorders, orders not yet shipped, etc.

1 We will discuss **where** to keep inventory in Chapter 10.

https://doi.org/10.1515/9783110673944-001

In-transit inventory

Goods ordered from a supplier but not yet available in our warehouse for our clients to buy. These goods are considered to be in-transit between two warehouses (or in pipeline). See Section 3.1.

With a continuous review policy, the elapsed time between two consecutive orders will vary (as the demand fluctuates), but the order quantity will always be the same (as you can see in Figure 1.1).

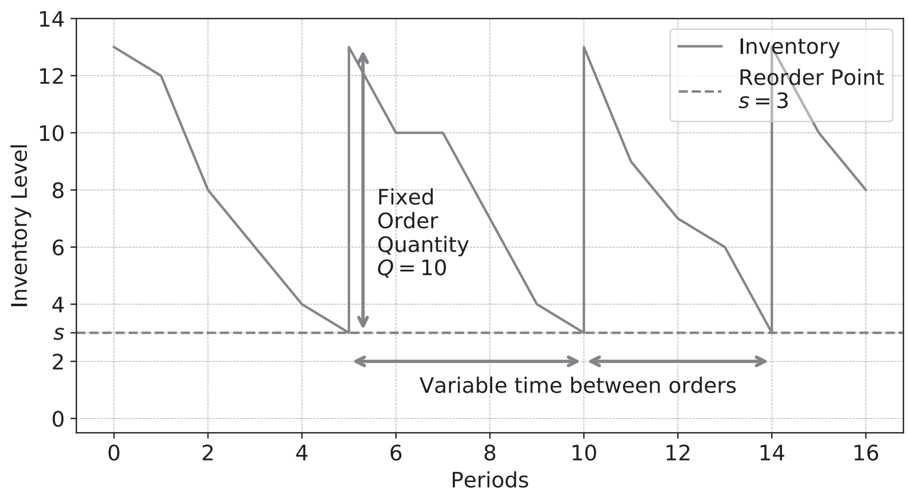

Figure 1.1: Fixed reorder point policy (with immediate replenishment): "Order 10 when less than 3 left."

Typically, with such a policy, you could say that:

– *When the stock level reaches 3 pieces, I order 10.*
  Here the fixed reorder point is 3, and the order quantity is 10 (see Figure 1.1). Note that, even if you only have 2 pieces in stock, you will still only order 10 pieces. Remember: with the fixed reorder point, the order quantity (10 in this case) is always fixed.
– *As soon as I am left with 2 bottles of milk in my refrigerator, I'll go to the supermarket and buy 6.*
– *When my printer says that I am left with only 10% of ink, I'll order a new set of cartridges.*

Advantages

This policy is safe (i. e., the risk of being out-of-stock is low) as it assumes you can make an order whenever you need to. It is therefore a good policy for expensive and/or important items that need to be monitored closely.

Another important advantage of this policy is that you can optimize the order quantity based on some (often obvious) constraints or costs. For example, you might get a rebate if you order a full pallet or a truckload. With such a policy you are sure to get the reduction each time you make an order.

Limitations

First, **it won't allow you to group into a single order different items with a single supplier**. Imagine that you supply a hundred different products from a single-favorite supplier: you might want to simplify your operations and buy all these products at once (in one single order) instead of doing different orders multiple times a day, when each one of your items needs to be replenished.

Then, we assume that **a client can make an order with its supplier at any time**. In reality, this might not be the case. For example, a supplier might only accept orders once a month (or only send one shipment a month—which is the same). In such a case, it is foolish to think that you would follow a fixed reorder point policy, as the supplier is actually following its own calendar.

Finally, the pure theoretical mathematical model of this policy (that we will discuss in the following chapters) assumes that **each client can only buy one product at a time**, so that the reorder point will always be perfectly reached.

In practice, these assumptions are not often respected. A pure continuous policy is therefore exceptional (some fully automated production processes, or internal processes, could follow these assumptions). It is therefore used as a (useful) theoretical simplification.

Notation

The fixed reorder point policy is noted (s, Q), with s the reorder point and Q the fixed order quantity.

1.2 Policy #2 – Periodic Review and Order Up-to Level

For this second inventory policy, we will order products periodically, based on a **fixed schedule** and on an **up-to level**. At the beginning of any review period, we will order enough products to bring our net inventory to the up-to level. As you can see in Figure 1.2, with this fixed periodic schedule, we will order a different quantity each time. The order quantity depends on how much inventory we have at the time we make the order, and is therefore variable. On the other hand, the orders are made following a fixed schedule: the elapsed time between two consecutive orders will always be the same (but the order quantity will always change).

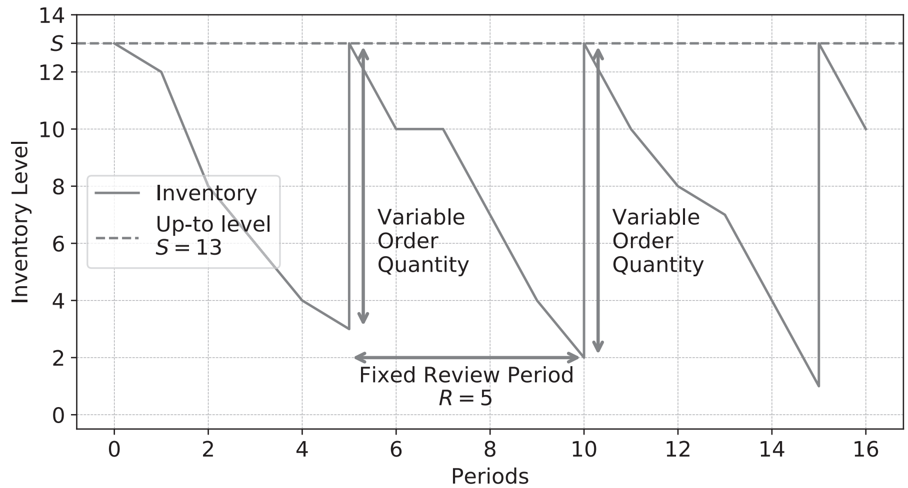

Figure 1.2: Fixed review period policy (with immediate replenishment): "Order up to 13 every 5 days."

Here are some typical examples:

– *We make an order every Friday evening to our supplier, so that they can prepare our order on Monday morning and deliver it on Tuesday.*
– *We make an order every third business day of the month with our supplier in China.*
– *I go to the supermarket every Saturday morning. I buy enough bottles of milk in order to have a total stock of 6 liters.*

Advantages

This periodic replenishment policy (with an order up-to level) is actually the most common inventory policy because it allows businesses to group their orders with each of their suppliers. This will help both clients and suppliers to streamline their operations as they can plan orders and workload in advance, and define a periodic review process.

A periodic replenishment policy is also often forced onto supply chains by the use of a MRP/DRP.[2] These tools follow a predefined schedule—often daily or weekly—resulting in the implicit use of a periodic review policy.

Limitations

As we will discuss later in Chapter 4, "Safety Stocks," this policy is riskier, due to the blind spot it creates: you cannot order in-between two review periods. If you make an order every Friday with your supplier, but are out-of-stock on Monday evening,

2 MRP stands for Materials Requirements Planning, it is a software/methodology that is used to plan the sourcing and production of goods. DRP stands for Deployment Requirement Planning, it is also a software/methodology used to plan the delivery of goods. Due to running time motives, these tools often run daily or weekly, imposing therefore a periodic review policy.

you will have to wait four more days before making a new order. You will possibly suffer lost sales due to being out-of-stock in the meantime. That is riskier than the fixed reorder point policy, where you would have made a new order directly on Monday evening.

Another issue is that the order quantity will vary at each order. This might disrupt a smooth operational flow. If you have a palletized product, you might not want to remove the packaging around an entire pallet in order to get a single unit.

Notation

The fixed review period policy is often noted (R, S), with R being the fixed review period and S the up-to level. Don't get confused by the notation: academics usually use s to denote the reorder point and S to denote the order up-to level. As a mnemonic device, **lowercase *s* denotes a minimum amount of stock**, and **uppercase *S* a maximum**.

1.3 Policy #3 – Periodic Review, Reorder Point and Fixed Order Quantity

We just discussed a first inventory policy that offered us the convenience of a fixed order quantity, and a second that gave us the convenience of a fixed review period. We can then create an inventory policy combining the best of both worlds in terms of operational convenience. This policy will consist of making orders of fixed quantity Q, based on a fixed schedule, if the inventory level reaches a threshold s (see Figure 1.3).

Here are some examples:

– *Every Saturday, if I have less than 2 bottles of milk in my refrigerator, I go to the supermarket to buy 6 bottles of milk.*

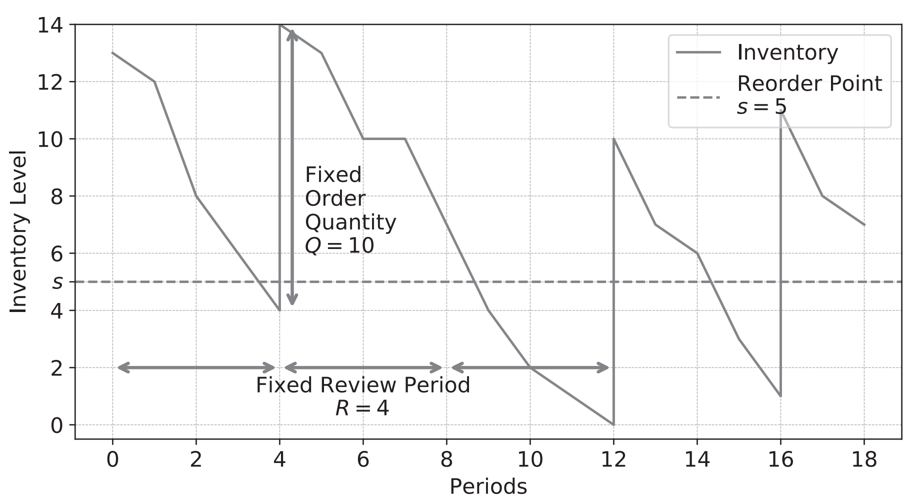

Figure 1.3: Period review with reorder point and fixed order quantity: "Order 10 pieces every 4 days only if inventory level reaches 5 pieces."

– *Every Friday evening, if our net inventory reaches 10 units (or below), we make an order of 10 pieces to our supplier (we order 10 pieces due to the packaging). We place the order on Friday, so that the supplier can prepare the order on Monday morning and deliver it on Tuesday.*

Advantages
This policy has two strong advantages:
– The order quantity is always constant, allowing transportation and packaging to be optimized (e. g., full pallet, full truckload).
– The order is always made at a predefined time slot (e. g., at the end of the day/week), allowing smooth operations and the ability to group orders with a supplier.

Limitations
The risk is high with this kind of policy: we have a review period possibly blocking us from making an order when we need it; on top of that, we will only make an order if the stock reaches a certain threshold. This extra risk will need to be compensated for by an extra amount of safety stock, resulting in higher inventory levels and costs. Besides, this policy is also much more difficult to optimize mathematically, and therefore less discussed in the academic literature.[3] As the models become too complex (and based on too many assumptions), we will optimize this policy with simulations in Chapter 13.

Notation
We will note this policy (R, s, Q), where R denotes the fixed review period, s the fixed reorder point, and Q the fixed order quantity. This policy is also sometimes noted $(R, s, n \cdot Q)$ if we can order a multiple of Q units at once (typically in the case of truckload or packaging constraints).

1.4 Other Policies

Of course, a supply chain can be piloted with other policies.

(R, s, S) Policy
We can imagine an (R, s, S) policy, where the inventory is reviewed periodically and, if the inventory level reaches a threshold s, an order is placed up-to a certain level S.

3 See Tempelmeier (2011), Tijms (1994) for more information.

Again, such policies are more difficult to optimize mathematically—yet analyzed since at least the 1960s—and are, therefore, out of scope for this book.[4]

Multi-Sourcing

We can also create even more complex policies. For example, in order to reduce the supply risk, it is rather common for a supply chain to have an emergency supply lane in addition to a regular supply lane with a preferred supplier. As the behavior of these policies become more complex, it will become *impossible* to solve them *perfectly* based on *tractable*[5] models.

That, of course, is not a reason not to use them. The question should be: how can we approximate these policies in order to get a useful and understandable model? Even if we cannot create such a model, we have another tool to optimize these: running simulations. We will discuss this technique in Chapter 13, "Simulation Optimization."

1.5 Confusion Curse: Inventory Target

Pay attention to the fact that supply chain practitioners—and software vendors—often use different terms to refer to the same policies. For example the term "reorder point" (or ROP) could define an order up-to level or a replenishment threshold. This means that a periodic review policy could be called an ROP policy as you set a "reorder point" (which is actually nothing more than the up-to level).

The term "stock target" is actually the most confusing. When a practitioner refers to a "stock target of 10," it is never clear what she refers to. Is it an order up-to target in a periodic review policy? Or the reorder point in a fixed reorder point policy? Or the average on-hand inventory they expect to have? We will discuss this confusion curse further in Chapter 3, "When Should I Order?" and Chapter 4, "Safety Stocks."

Important Point

When discussing with colleagues, software vendors or consultants, always define clearly what a "reorder point" and a "stock target" is.

4 See Veinott and Wagner (1965), Ehrhardt (1979), Tijms and Groenevelt (1984), Strijbosch and Moors (1999) for detailed models.

5 Tractable is a fancy word that mathematicians use to describe models or equations that are simple enough to be used or solved.

1.6 Recap

An inventory policy defines **when** an order needs to be made and **how many** units should be ordered. We can categorize them between continuous and periodic review inventory policies. A policy can be defined by different parameters:
– the order up-to level S
– the review period R
– the order quantity Q
– the reorder point s

We analyzed three policies:

(s, Q) A continuous review policy with a reorder point s and a fixed order quantity Q.
+ Pros: low amount of safety stock needed; optimized order quantity.
- Cons: continuous review needed; multiple items cannot be grouped in one order with one supplier.
  See Section 1.1.

(R, S) A periodic review policy following a review period R and an order up-to level S.
+ Pros: multiple items can be grouped in one order with one supplier.
- Cons: more safety stock needed; order quantity varies and can't be optimized.
  See Section 1.2.

(R, s, Q) A periodic review policy following a review period R, with a reorder point s and a fixed order quantity Q.
+ Pros: can group multiple items in one order with one supplier; optimized order quantity.
- Cons: even more safety stock needed; optimization is difficult.
  See Section 1.3.