WAREHOUSE & DISTRIBUTION SCIENCE
Release 0.97

Check for the latest version at www.warehouse-science.com

John J. BARTHOLDI, III [1]
Steven T. HACKMAN

The Supply Chain & Logistics Institute
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205 USA

Revised August 21, 2016

 Corresponding author: john.bartholdi@gatech.edu

Contents

List of Figures

List of Tables

Preface

0.1 Why this book

The topic of this book is the science of warehouse layout and operations. We say "science" because we develop mathematical and computer models. Too much of current warehousing practice is based on rules-of-thumb and simplistic ratios. This is fine as far as it goes; but there is much more that can be done. Most warehouses have elaborate records of exactly what every customer ordered and when. Typically this information is generated by the IT department as part of financial reporting; but this information can be used by operations to "tune" the warehouse layout and operations to the patterns of customer orders. This book shows you how to do this.

There are other books on warehousing but they mostly confine themselves to listing types of equipment, types of order-picking, and so on. This is necessary and good; but the emphasis here is not on taxonomy but on developing methodology to optimize warehouse operations.

This book is distinctive in another way as well: We try to avoid monolithic optimization models, which are expensive and inflexible. Instead we adopt an approach that emphasizes decentralizing decisions. For example, in allocating space in a warehouse, we will require each stockkeeping unit to make a "business plan" that details the economic benefits it will generate in return for space; then we will rank the skus according to which offers the best return. The business plan may well be a mathematical model; but the process of allocation is done simply from a sorted list. Such an approach offers great simplicity of use; indeed, one can make decisions dynamically, re-allocating space to the current best bidder of the moment. The trick is in pricing out the bids appropriately. Ultimately, this leads to a view of the pallets, cartons, pieces of product moving through the distribution center as commuters might, with each one aware of its dimensions and requirements for special handling. They might enter into negotiations with their environment to learn, for example, the available space and labor economics of each type of storage, and then make their decisions about how to move through the required processes and on to the customer. This degree of decentralization is imaginable as RFID continues to spread, placing memory and, increasingly, computing power on each item.

Another example of decentralization is our treatment of "bucket brigades" as a new style of order-picking in which the work is reallocated by the independent movements of the workers. If the bucket brigade is configured properly, the order-pickers will balance the work amongst themselves and so eliminate bottlenecks. Moreover, this happens spontaneously, without intention or awareness of the workers. This means that the order-picking can be more effective than if planned by a careful engineer or manager.

To some extent the book reflects the challenges of distribution in North America, where there are relatively high labor costs, relatively low capital costs, and high volumes. However, all the tools and models we build can be used in other environments as well, even when their emphasis is on minimizing labor costs. The model will determine the "exchange rate" by which space can be exchanged for time (labor); and so anyone, even in a low labor cost region, can perform the optimization, clarify the exchange of space for time, and then make informed engineering decisions that are appropriate to his or her context.

0.2 Organization

Part I, Issues, equipment, and processes

- We begin with a brief discussion of material flow and provide an simple, aggregate way of viewing it. This "fluid model" provides useful insights in the large.
- Next we give an overview of warehouse operations: Typical kinds of warehouses; how they contribute to the operations of a business; what types of problems a warehouse faces; what resources a warehouse can mobilize to solve those problems; and some simple tools for analysis.
- We survey typical equipment used in a warehouse and discuss the particular advantages and disadvantages of each.

Part II, Warehouse layout

Warehouse layout sets the stage for order-picking, the most important activity in the warehouse. If product is staged for quick retrieval, customers will receive good service at low cost.

Everyone knows how to lay out a warehouse: Conventional wisdom says to put the fastest-moving skus in the most convenient locations. The problem is that all this depends on what is meant by "fast-moving" and what is meant by "convenient". We elucidate this by building models of space and time (labor) to make this truism mean something precise. Frequently the answer is surprisingly at odds with standard practice and with intuition.

- We start with a particularly simple type of warehouse: A "unit-load" warehouse in which the skus arrive on pallets and leave on pallets. In such warehouses, the storage area is the same as the picking area and models of space and time (labor) are simple linear models. Accordingly, we can estimate the work inherent in using each storage location and the work inherent in moving each sku through the warehouse. This enables us to say exactly where each pallet should be stored to minimize labor.

- We move to more complicated warehouses in which most skus arrive packaged as pallets and leave as cartons. It is harder to make distinctions amongst all the storage locations, as we could do for unit-load, but we can identify those skus that deserve to be stored as pallets in a forward storage area to minimize labor.
- Next we examine high-volume, labor-intensive warehouses, such as those supporting retail stores: Skus may be stored as cartons and leave as pieces. Orders typically consist of many skus and may be assembled as on an assembly line. We are able to identify those skus that deserve storage as cartons in the forwardmost locations to minimize labor.

Part III, Order-picking

Order-picking is the most labor-intensive activity in the warehouse. It also determines the service seen by your customers. It must be flawless and fast.

- When there is a common pick-path, order-pickers can operate as a sort of assembly line, "building" each customer order and passing it downstream. We show a new way of coördinating the order-pickers in which each follows a simple, decentralized rule but global coördination emerges spontaneously.
- When there is not a pick-path that is common to all order-pickers, then it must be decided by what path an order-picker should travel through the warehouse to retrieve the items of an order.

Part IV, Automation

When is automation appropriate? How can one estimate the value conferred by automation? With what logic does one imbue automation so that it is effective? We answer these questions for some of the most common types of automation.

Part V, Special topics

Here we discuss some special types of warehouses and the issues that are unusual to them.

- A crossdock is a kind of high-speed warehouse in which product flows with such velocity that there is not point in bothering to put it on a shelf. Occupancy times are measured in hours rather than days or weeks and efficient operation becomes a matter of finely-tuned material handling.

Part VI, Measuring warehouse performance

- What are the essential statistics by which the operations of a warehouse are to be understood? Where is the information found?
- Which of several warehouses is most efficient? It would be nice to know because then others could copy its best practices. It can be tricky to compare the performance of two different warehouses, especially if they are of different sizes, different configuration, or serve different industries.

Part VII, Miscellaneous

- Local conditions dictate different designs and operations in warehouses around the world. Sometimes what appears to be obvious inefficiency is a rational response to local economics.

In teaching the graduate class in warehousing and distribution at Georgia Tech, we generally follow the book from start to finish, with a single exception. When the class is engaged in a project for a company, we insert a lecture on warehouse activity profiling (Chapter 14) early in the course so the students know how to mine the data typical to a warehouse.

0.3 Resources

We support this book through our web site `www.warehouse-science.com`, from which you may retrieve the latest copy and find supporting materials, such as photographs, data sets, programming tools, and links to other resources. We are constantly adding to this collection.

The photographs and data sets are from our consulting and from generous companies, whom we thank. At their request, some company identities have been disguised and/or some aspect of the data cloaked.

0.4 But first…

A few friends and colleagues have contributed so much to the work on which this book is based that they deserve special mention. We particularly thank Don Eisenstein of the University of Chicago (Chapter 11 and more), Kevin Gue of Auburn University (Chapter 13), and Pete Viehweg for critical reading throughout and comments based on years of experience in industry. Their contributions have been so important that each has some claim to co-authorship. In addition, we have relied on the work of Ury Passy of The Technion (Chapter 8) and Loren "Dr. Dunk" Platzman (Chapter 8). We also thank many friends and colleagues who caught errors or suggested improvements. These include Yossi Bukchin of Tel Aviv University, Ileana Castillo of Tecnológico de Monterrey (Toluca), Chen Zhou of Georgia Tech, Russ Meller of the University of Arkansas, Bryan Norman of the University of Pittsburgh, Morrakot Raweewan of Sirindhorn International Institute of Technology and Thammasat University, Jacques Renaud of Université Laval, Neale Smith Cornejo of Tecnológico de Monterrey, and Leonardo Rivera of Universidad Icesi. Thanks also to many students who suffered under earlier versions and found errors: Ayhan Aydin, Karin Boonlertvanich, Asgeir Brenne, Dominik Boehler, Christian Buchmann, Luis Felipe Cardona Olarte, Wilson Clemons, Anwesh Dayal, Mike Depace, Ozan Gozbasi, Huang Chien-Chung, Li Jiexin, Ron Chung Lee, Amanda Mejia, Geoffrey Miller, Sabahattin Gokhan Ozden, and Philipp Schuster.

Part I

Issues, equipment, processes

Warehouses are the points in the supply chain where product pauses, however briefly, and is touched. This consumes both space and time (person-hours), both of which are an expense. The goal of this book is to develop mathematical and computer models to allow you to reduce space and time requirements or to exchange one for the other. But first it is necessary to understand the role a warehouse serves in the supply chain and the means by which it does this.

This part starts with the big picture and then looks inside a warehouse.

Chapter 1

Warehouse rationale

1.1 Why have a warehouse?

Why have a warehouse at all? A warehouse requires labor, capital (land and storage-and-handling equipment) and information systems, all of which are expensive. Is there some way to avoid the expense? For most operations the answer is no. Warehouses, or their various cousins, provide useful services that are unlikely to vanish under the current economic scene. Here are some of their uses:

To better match supply with customer demand: One of the major challenges in managing a supply chain is that demand can change quickly, but supply takes longer to change. Surges in demand, such as seasonalities strain the capacity of a supply chain. Retail stores in particular face seasonalities that are so severe that it would be impossible to respond without having stockpiled product. For example, Toys R Us does, by far, most of its business in November and December. During this time, their warehouses ship product at a prodigious rate (some conveyors within their warehouses move at up to 35 miles per hour). After the selling season their warehouses spend most of their time building inventory again for the following year. Similarly, warehouses can buffer the supply chain against collapsing demand by providing space in which to slow or hold inventory back from the market.

In both cases, warehouses allow us to respond quickly when demand changes. Response-time may also be a problem when transportation is unreliable. In many parts of the world, the transportation infrastructure is relatively undeveloped or congested. Imagine, for example, sourcing product from a factory in Wuhan, China for retail sale within the US. After manufacture, the product may travel by truck, then by rail, by truck again, and then be loaded at a busy port; and it may repeat the sequence of steps (in reverse order) within the US. At each stage the schedule may be delayed by congestion, bureaucracy, weather, road conditions, and so on. The result is that lead time is long and variable. If product could be warehoused in Los Angeles, closer to the customer, it could be shipped more quickly, with less variance in lead time, and so provide better customer service.

Warehouses can also buffer against sudden changes in supply. Vendors may give a price break to bulk purchases and the savings may offset the expense of storing the product. Similarly, the economics of manufacturing may dictate large batch sizes to amortize large setup costs, so that excess product must be stored. Similarly, warehouses provide a place to store a buffer against unreliable demand or price increases.

To consolidate product to reduce transportation costs and to provide customer service.

There is a fixed cost any time product is transported. This is especially high when the carrier is ship or plane or train; and to amortize this fixed cost it is necessary to fill the carrier to capacity. Consequently, a distributor may consolidate shipments from vendors into large shipments for downstream customers. Similarly, when shipments are consolidated, then it is easier to receive downstream. Trucks can be scheduled into a limited number of dock doors and so drivers do not have to wait. The results are savings for everyone.

Consider, for example, Home Depot, where more than a thousand stores are supplied by several thousands of vendors. Because shipments are frequent, no one vendor ships very much volume to any one store. If shipments were sent direct, each vendor would have to send hundreds of trailers, each one mostly empty; or else the freight would have to travel by less-than-truckload (LTL) carrier, which is relatively expensive (Figure 1.1). But there is enough volume leaving each vendor to fill trailers to an intermediate crossdock. And each crossdock receives product from many vendors, sorts it, and prepares loads for each store, so that the the total freight bound for each store is typically sufficient to fill a trailer. The result is that vendors send fewer shipments and stores receive fewer shipments. Moreover, the freight is more likely to travel by full truck-load (TL) and so pay significantly less transportation costs (Figure 1.2).

A warehouse also provides opportunities to postpone product differentiation by enabling generic product to be configured close to the customer. Manufacturers of consumer electronics are especially adept at this. Country-specific parts, such as keyboards, plugs, and documentation, are held at a warehouse and assembled quickly in response to customer order. This enables the manufacturer to satisfy many types of customer demand from a limited set of generic items, which therefore experience a greater aggregate demand, which can be forecast more accurately. Consequently safety stocks can be lower. In addition, overall inventory levels are lower because each item moves faster.

Another example is in pricing and labeling. The state of New York requires that all drug stores label each individual item with a price. It is more economical to do this in a few warehouses, where the product must be handled anyway, than in a thousand retail stores, where this could distract the workers from serving the customer.

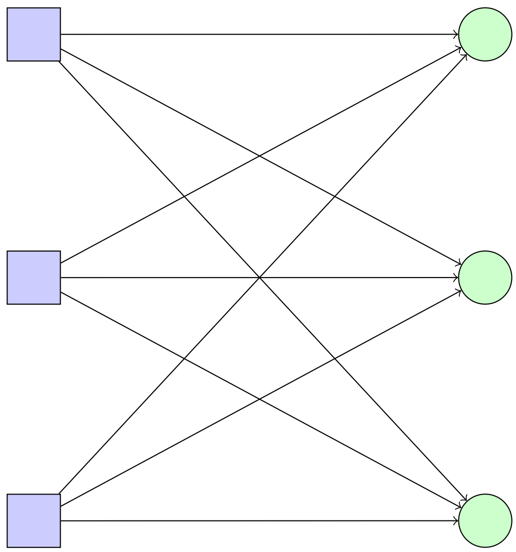


Figure 1.1: With m vendors and n stores the transportation plan consists of mn direct shipments, each relatively small and likely subject to the higher, less-than-truckload rates.

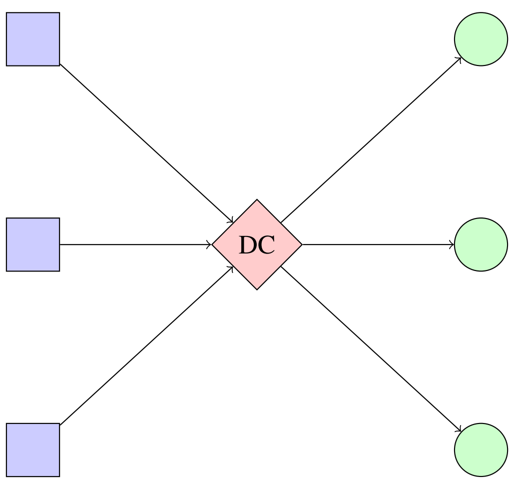


Figure 1.2: There are only $m + n$ shipments through an intermediate aggregator, such as a distribution center or crossdock. Furthermore, each shipment is larger and more likely to qualify for the lower, full-truckload rates.

1.2 Types of warehouses

Warehouses may be categorized by type, which is primarily defined by the customers they serve. Here are some of the more important distinctions:

A *retail distribution center* typically supplies product to retail stores, such as Wal-Mart or Target. The immediate customer of the distribution center is a retail store, which is likely to be a regular or even captive customer, receiving shipments on regularly scheduled days. A typical order might comprise hundreds or thousands of items; and because the distribution center might serve hundreds of stores, the flow of product is huge. The suite of products changes with customer tastes and marketing plans; but because the orders are typically known a day or more in advance, it is possible to plan ahead. Some product may be pushed from the distribution center to the stores, especially in support of marketing campaigns.

A *service parts distribution center* is among the most challenging of facilities to manage. They hold spare parts for expensive capital equipment, such as automobiles, airplanes, computer systems, or medical equipment. Consequently, one facility may represent a huge investment in inventory: tens or even hundreds of thousands of parts, some very expensive. (A typical automobile contains almost 10,000 parts.) Because of the large number of parts, total activity in the DC may be statistically predictable, but the demand for any particular part is relatively small and therefore hard to predict. This means that the variance of demand can be large and so relatively large quantities of safety stock must be held, especially since there can be usually long lead times to replenish parts to the warehouse. Indeed, sometimes there is as much safety stock as cycle stock, and so, in aggregate, these skus require much space. This in turn increases travel distances and makes order-picking less efficient.

A typical service parts warehouse manages two distinct order streams: stock orders, by which dealers replenish their shelves; and emergency orders, in which an equipment owner or independent repair shop urgently requires a few special parts to repair a broken piece of capital equipment. Stock orders tend to be large and fairly predictable replenishments of popular consumables, while emergency orders are typically small (two to three pick-lines), unpredictable, and urgent, because expensive capital equipment is likely waiting for repair. Emergency orders are typically for items that are ordered infrequently (otherwise they could have been provided by the dealer from stock inventory). Such orders—a few, slow-moving items that must be picked immediately—are relatively expensive to handle. Worse, customers ordering for repair might order before they are absolutely sure which parts need replacement; and so there can be a significant percentage of returns to be handled at the warehouse.

For most product in a service parts warehouse there are not sufficiently reliable patterns of movement to justify special processes, but one can layout stock to be more space efficient by storing similar sizes together, thereby reducing travel. Furthermore, one can hedge chances of having to travel long distances. For ex-

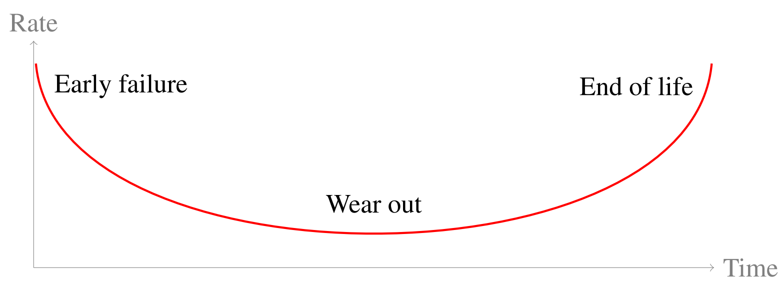


Figure 1.3: The failure rate of many mechanical components follows a "bathtub" distribution, which means demand for these parts is higher at the beginning and at the end of life.

ample, it can be advantageous, especially for emergency orders, to store products together that are likely to be ordered together.

Another complication is that the life cycle of a service part is unusual, with three stages of product life, as shown in Figure 1.3. Early failures are generally due to manufacturing imperfections; mid-life failures are generally due to random events that stress the part beyond its tolerance; and end-of-life failures are due to expected wearing out of the product. Demand for product generally reflects this pattern, and creates challenges in the warehouse. For example, there is little time to ramp up availability of new product at the start of its life cycle. Also, parts are more frequently requested at the end of the product life cycle, and so it is easy for the warehouse to be stuck with obsolete merchandise. Finally, it may be necessary to relocate product as its popularity changes.

A *catalog fulfillment* or *e-commerce* distribution center typically receives small orders from individuals by phone, fax, or the Internet. Orders are typically small, for only 1–3 items, but there may be many such orders, and they are to be filled and shipped immediately after receipt. Because customer orders require instant response, such distributors typically try to shape demand by offering special prices for ordering at certain times or in certain quantities or for accepting more variable delivery dates.

A *3PL warehouse* is one to which a company might outsource its warehousing operations. The 3PL provider might service multiple customers from one facility, thereby gaining economies of scale or complementary seasons that the customers would be unable to achieve on their own. 3PL facilities may also be contracted as overflow facilities to handle surges in product flow.

A *perishables warehouse* may handle food, fresh flowers, vaccines, or other product requiring refrigeration to protect its very short shelf life. They are typically one link in an extended *cold chain*, along which perishable product is rushed to the consumer. Such DCs are distinctive in that product dwells within for very short times, frequently only hours. Also, there is a great emphasis on using space effectively because, with refrigeration, it is so expensive. They face many challenges in inventory management, including requirements to ship product according to FIFO (First-In-First-Out) or FEFO (First-Expired-First-Out). Also, there are typically many restrictions on how product is handled. For example, chicken cannot be stacked on top of anything else, to protect against juices dripping onto product below and contaminating it. Finally, appropriate temperatures must be maintained and this can be different for different kind of products. A typical food DC operates separate areas for ambient temperatures, chilled (around 2 degrees C, 35 degrees F), and frozen product (-18 degrees C, around 0 degrees F). To protect stored product, it is important to avoid bringing in anything warmer.

This type of warehouse will become more common as China, India, Brazil, and other rapidly industrializing countries build a middle class, which will increasingly want fresh fruit, vegetables, meat, and dairy.

While there are many types of warehouses in the supply chain, one of the main themes of this book is that there is a systematic way to think about a warehouse system regardless of the industry in which it operates. As we shall show, the selection of equipment and the organization of material flow are largely determined by

- Inventory characteristics, such as the number of products, their sizes, and turn rates;
- Throughput and service requirements, including the number of lines and orders shipped per day;
- The footprint of the building and capital cost of equipment;
- The cost of labor.

Chapter 2

Material flow

Here we briefly discuss a few issues that help lay the foundations for warehouse analysis. The most fundamental idea is of the management of two resources: space and time (that is, labor or person-hours).

2.1 The fluid model of product flow

The "supply chain" is the sequence of processes through which product moves from its origin toward the customer. In our metaphor of fluid flow we may say that warehouses represent storage tanks along the pipeline.

The analogy with fluid flows can also convey more substantial insight. For example, consider a set of pipe segments of different diameters that have been joined in one long run. We know from elementary fluid dynamics that an incompressible fluid will flow faster in the narrower segments of pipe than in the wider segments. This has meaning for the flow of product: The wider segments of pipe may be imagined to be parts of the supply chain with large amounts of inventory. On average then, an item will move more slowly through the region with large inventory than it will through a region with little inventory.

The fluid model immediately suggests other general guidelines to warehouse design and operation, such as:

- Keep the product moving; avoid starts and stops, which mean extra handling and additional space requirements.
- Avoid layouts that impede smooth flow.
- Identify and resolve bottlenecks to flow.

Later we shall rely on the fluid model to reveal more profound insights.

It is worth remarking that the movement to "just-in-time" logistics is roughly equivalent to reducing the diameter of the pipe, which means product flows more quickly and so flow time and in-transit inventory are reduced (Figure 2.1).

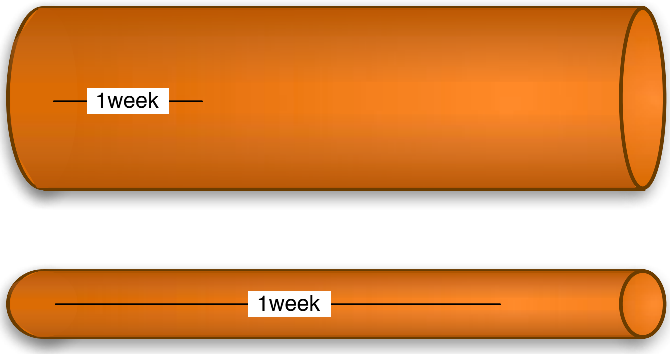


Figure 2.1: If two pipes have the same rates of flow, the narrower pipe holds less fluid. In the same way, faster flow of inventory means less inventory in the pipeline and so reduced inventory costs.

2.2 Units of handling

Even though it is a frequently useful metaphor, most products do not, of course, flow like incompressible fluids. Instead, they flow more like a slurry of sand and gravel, rocks and boulders. In other words, the product is not infinitely divisible but rather is granular at different scales.

A *stock keeping unit*, or *sku*, is the smallest physical unit of a product that is tracked by an organization. For example, this might be a box of 100 Gem Clip brand paper clips. In this case the final customer will use a still smaller unit (individual paper clips), but the supply chain never handles the product at that tiny scale.

Upstream in the supply chain, product generally flows in larger units, such as pallets; and is successively broken down into smaller units as it moves downstream, as suggested in Figure 2.2. Thus a product might move out of the factory and to regional distribution centers in pallet-loads; and then to local warehouses in cases; and finally to retail stores in inner-packs or even individual pieces, which are the smallest units offered to the consumer. This means that the fluid model will be most accurate downstream, where smaller units are moved.

2.3 Two fundamental resources

Warehouse management is all about careful use of space and time (that is, labor or person-hours). Both space and time are expensive and so one would like to use as little of each as possible in delivering product to customers.

Figure 2.3 shows a plot of the popularity (number of times requested, or *picks*) of each sku of a warehouse together with the physical volume (*flow*) of the sku moved through the warehouse during one month. There is little correlation between popularity and flow, and this is one of the challenges of warehouse management, because it is

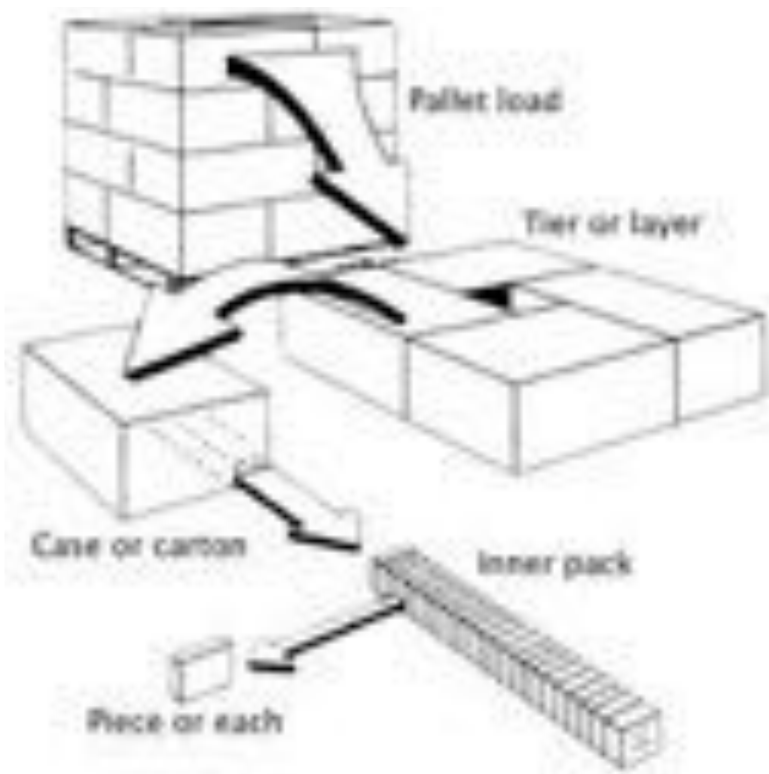


Figure 2.2: A product is generally handled in smaller units as it moves down the supply chain. (Adapted from "Warehouse Modernization and Layout Planning Guide", Department of the Navy, Naval Supply Systems Command, NAVSUP Publication 529, March 1985, p 8–17).

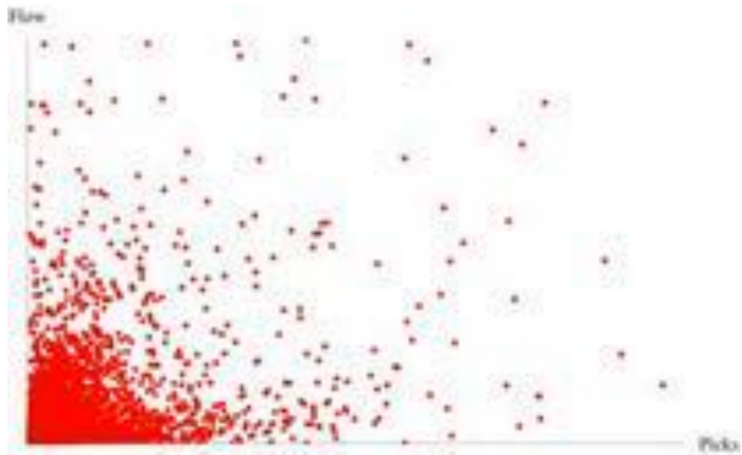


Figure 2.3: Among these 25,000 skus there is little correlation between popularity and physical volume of product sold.

hard to design processes that work well with skus that may be any combination of popular/unpopular and low-volume/high-volume.

Figure 2.4 plots just the popularity of the same set of about 25,000 skus, here ranked from most popular to least. This is typical of such plots in that a small fraction of the skus account for most of activity. It is easy to design processes for these skus because they are fairly predictable. If popular yesterday, such a sku is likely to be popular again tomorrow.

On the other hand, consider all the skus in the so-called long-tail, in this case the 20,000 skus that are requested infrequently. It is impossible to know whether any particular sku will be requested tomorrow. Such skus, by their sheer number, occupy most of the space in a warehouse. This effect is further magnified by safety stock, which is held to protect against stockout in the face of customer demand that is highly variable in comparison to the amounts held.

Each warehouse then is, in a sense, two warehouses. The first is organized around a small set of predictably popular skus that are easy to plan for and for which the challenge is to manage flow. The other warehouse is much larger, and for which the work is predictably only in aggregate. This makes it much harder to plan and one is forced to hedge decisions. The first warehouse is where labor is concentrated; and the second consumes space.

2.4 Storage: "Dedicated" versus "Shared"

Each storage location in a warehouse is assigned a unique address. This includes both fixed storage locations, such as a portion of a shelf and mobile locations such as the forks of a lift truck. Storage locations are expensive because they represent space, with consequent costs of rent, heating and/or air-conditioning, security, and so on. In

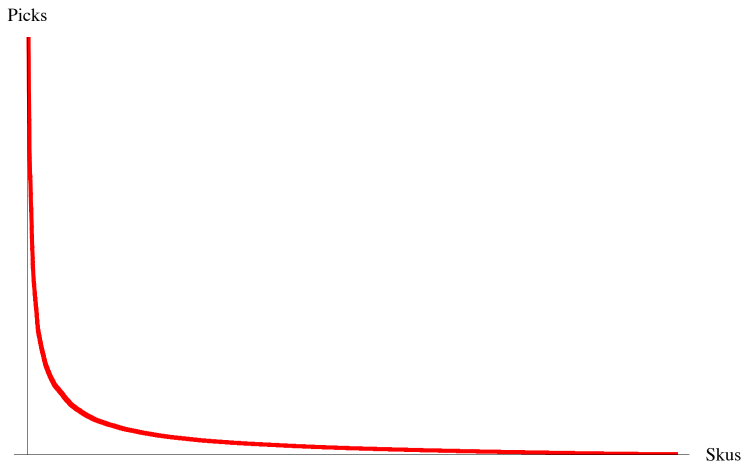


Figure 2.4: Popularity among these 25,000 skus varies enormously, which presents special challenges to effective management.

addition, storage locations are typically within specialized equipment, such as shelving or flow rack, which are a capital cost. These costs impel us to use storage space as efficiently as possible.

There are two main strategies used in storing product. The simplest is *dedicated* storage, in which each location is reserved for an assigned product and only that product may be stored there. Because the locations of products do not change, more popular items can be stored in more convenient locations and workers can learn the layout, all of which makes order-picking more efficient.

The problem with dedicated storage is that it does not use space efficiently. This can be seen by tracking the amount of inventory in a given location. If we plot the inventory level, measured for example by volume, we would see a sawtooth shape such as in Figure 2.5 (which represents an idealization of the inventory process.) In one cycle the storage location is initially filled but empties as product is withdrawn to send to customers. As a result, on average this storage location is half empty.

A warehouse may have thousands or tens-of-thousands of storage locations. If using dedicated storage, each will have an assigned product. Each product may have a different replenishment cycle and so, upon entering such a warehouse, one expects to see many storage locations that are nearly empty, many that are half-full, and many that are nearly full. On average the storage capacity is only about 50% utilized.

To improve on this, one can use a strategy of *shared* storage. The idea here is to assign a product to more than one storage location. When one location becomes empty, it is available for reassignment, perhaps to a different product. This space then can be filled again, rather than waiting until the original product is replenished (presumably when the last of the warehouse supply has been exhausted). The more storage locations over which a product is distributed, the less product in each location, and so the sooner one of those locations is emptied and the sooner that space is recycled. Therefore we

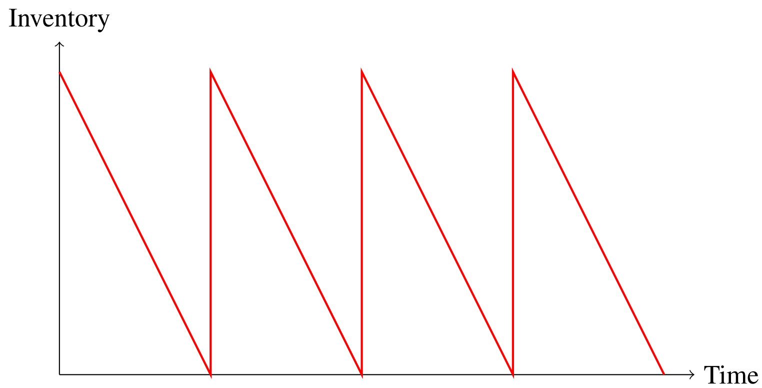


Figure 2.5: An idealization of how the inventory level at a location changes over time

expect better utilization of space when shared storage is used.

Unfortunately, shared storage also has some disadvantages. Most immediately, the locations of products will change over time as locations are emptied and restocked with other products. This means that workers cannot learn locations and so must be directed to locations by a warehouse management (software) system. Another disadvantage is that it becomes more time-consuming to put away newly received product because it has to be taken to more locations. There can be other, social complications as well. For example, imagine an order picker who has been directed to the other side of the warehouse to pull a product for a customer. That order picker may be tempted to pick the product from a more convenient location, thus creating discrepancies between book and physical inventory at two locations. For these reasons, shared storage requires greater software support and also more disciplined warehouse processes.

Shared storage is generally more complicated to manage because it introduces many possible trade-offs. In particular, one can manage the trade-off between space and time (labor) on an activity-by-activity basis. For example, one can retrieve product from the least-filled location (to empty and recycle that location as soon as possible) or from the most convenient location (to save labor). Similarly, one can replenish product to the most nearly empty location to fill that empty space or to the most convenient location to save labor time.

How much improvement in space utilization is possible with shared storage? Consider a product that is requested at a constant rate, as in our idealization of Figure 2.5. Suppose we hold two weeks supply of this product. If we store it in two locations of equal size and direct all order-picking to only one location then after one week, the first location will be emptied and available for reassignment. After the second week the second location will be emptied and available for reassignment. During the first week, the first location was half full on average and the second location was completely full, so that average space utilization was 75%. During the second week the second location was half-full, for an average space utilization of 50%, and the space utilization over the two weeks was 62.5%. More improvement is possible if a product is stored in more locations, though the improvement diminishes and, moreover, the practical problems

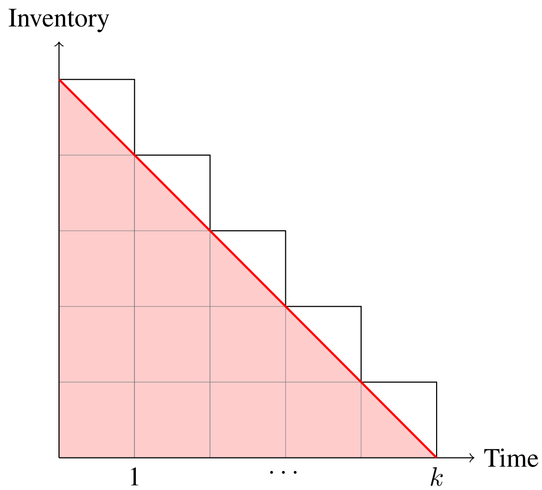


Figure 2.6: Use of k locations to hold product under a policy of shared storage. The step function represents the space devoted to holding inventory.

of management increase.

More generally we can argue as follows. Assume for convenience that demand is constant, the sku has been stored in k locations of identical size, and all picks are from a single active pick location until that location is exhausted. Then the inventory cycle may be imagined as consisting of k periods, each devoted to a particular pick location, as in Figure 2.6. The average space utilization in period i is

$$\frac{k-i+1/2}{k-i+1}$$

and the average utilization over all k periods is

$$\left(\frac{1}{k}\right)\sum_{i=1}^{k}\frac{k-i+1/2}{k-i+1}=1-\left(\frac{1}{2k}\right)\sum_{i=1}^{k}\frac{1}{i} \tag{2.1}$$

Figure 2.7 shows how the space utilization of storing a sku in more locations increases but with diminishing returns.

Interestingly, there is a slight sampling bias if one were to examine space utilization directly by taking a census within the warehouse. This is the subject of Exercise 2.7 and it suggests that you will tend to measure slightly smaller values than predicted by Expression 2.1.

In practice, a strategy of shared storage is typically used in the bulk storage areas, where most of the volume of product is held on pallets. Dedicated storage may be used in the most active picking areas, which are much smaller than the bulk storage. Thus one gets efficient use of most of the space (bulk storage) with labor benefits where it matters most (active picking areas).

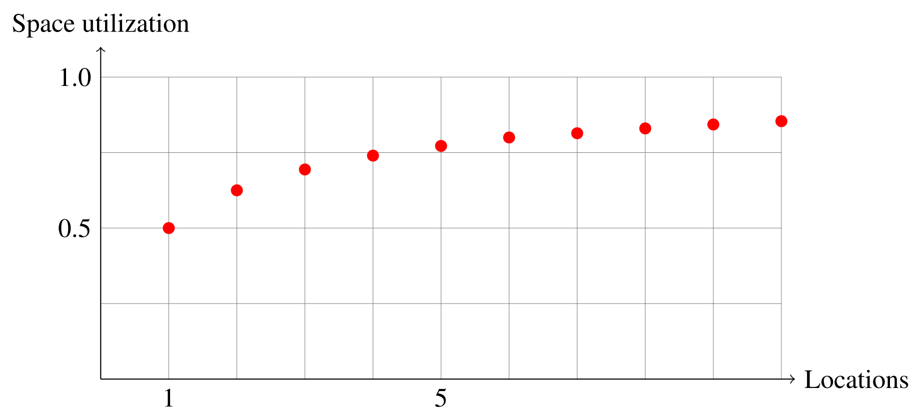


Figure 2.7: Under shared storage, space utilization increases with additional storage locations, but at a diminishing rate.

There are also hybrid schemes in which regions are reserved for groups of skus, but locations are not reserved. For example, an aisle might be reserved for skus of one type or from one vendor, but within that aisle, space would be shared amongst the skus.

2.5 The warehouse as a queuing system

A *queuing system* is a model of the following structure: Customers arrive and join a queue to await service by any of several servers. After receiving service the customers depart the system.

A fundamental result of queuing theory is known as Little's Law, after the man who provided the first formal proof of a well-known piece of folk-wisdom [36].

Theorem 2.1 (Little's Law)**.** *For a queuing system in steady state the average length L of the queue equals the average arrival rate λ times the average waiting time W. More succinctly:*

$$L = \lambda W.$$

A warehouse may be roughly modeled as a queuing system in which skus are the customers that arrive at the receiving dock, where they join a queue (that is, are stored in the warehouse) to wait for service (shipping). If the warehouse is at steady state the product will be shipped at the same average rate at which it arrives. Then Little's Law applies and the average amount of product in the warehouse equals the flow rate of product multiplied by the average time product is resident in our warehouse.

Generally the rate λ of flow is determined by the customers, and so Little's Law tells us that a warehouse holding lots of inventory (large L) will see most of it sitting (large W).

Here is an example of how we can use Little's Law to tease out information that might not be immediately apparent. Consider a warehouse with about 10,000 pallets in residence and that turn an average of about 4 times a year. What labor force is necessary to support this? By Little's Law:

$$10{,}000 \text{ pallets} = \lambda(1/4 \text{ year}).$$

so that

$$\lambda \approx 40{,}000 \text{ pallets/year}.$$

Assuming one 8-hour shift per day and about 250 working days per year, there are about 2,000 working hours per year, which means that

$$\lambda \approx 20 \text{ pallets/hour}.$$

Notice what we have just done: From a simple count of pallets together with an estimate of the number of inventory turns per year we estimated the labor requirements.

Little's Law can be very useful like this. Another typical use would be to compute an estimate of inventory turns after simply walking through the distribution center: One can estimate inventory (queue length) by counting storage positions, and rate of shipping (throughput) by observing the shipping dock, and then apply Little's Law.

What makes Little's Law so useful is that it continues to hold even when there are many types of customers, with each type characterized by its own arrival rate λ_i, waiting time W_i, and queue length L_i. Therefore the law may be applied to a single sku, to a family of skus, to an area within a warehouse, or to an entire warehouse.

2.6 Questions

Question 2.1. *What are the five typical physical units-of-measure in which product is handled in a warehouse? For each unit-of-measure, state whether there are any standardized dimensions and, if so, identify them.*

Question 2.2. *In what ways has the inventory process depicted in Figure 2.5 been idealized?*

Question 2.3. *What are the advantages and disadvantages of dedicated (reserved) storage? Of shared (random) storage?*

Question 2.4. *Why is random storage (shared storage) likely to generate more space efficiency in a pallet storage area than where cartons are stored directly on shelves?*

Question 2.5. *Consider a sku that has been allocated more than one location within a warehouse that is organized by "shared storage".*

- *What are the advantages/disadvantages of allowing picking from any or all of the locations?*
- *What are the advantages/disadvantages of restricting picking to only one of those locations (until empty, when another location would be designated the one from which to pick)?*

Question 2.6. *In real life a certain amount of safety stock may be held in a storage location to guard against stockout while awaiting replenishment to that location (which would interrupt order-picking). How would this affect the average utilization of storage space?*

Question 2.7 (Harder)**.** *What value of space utilization would you expect to observe if you showed up at random to examine storage at a warehouse using shared storage? For simplicity, consider a single sku that is stored in k locations of identical size. Assume an idealized inventory process such as depicted in Figure 2.5 and assume that all picks are directed to the location that is most nearly empty.*

Prove that the expected space utilization under the model above (random sampling) is never greater than the mean space utilization.

Where is the difference in the mean and expected mean values greatest?

Explain how this difference arises.

Question 2.8. *Why does the model of Question 2.7 break down (that is, lose practical meaning) for very large numbers of storage locations?*

Question 2.9. *Your third-party warehouse has space available for 10,000 pallets and you have 20 forklift operators per 8-hour day for 250 working days a year. If the average trip from receiving to storage to shipping is 10 minutes, how many inventory turns a year could you support for a full warehouse?*

Question 2.10. *Your third-party warehouse is bidding for a contract to store widgets as they are manufactured. However, widgets are perishable and should be turned an* *average of six times per year. The manufacturer produces at an average rate of 32 pallets per day. How many pallet positions should you devote to widgets to ensure that widgets turn as required.*

Question 2.11. *A pallet storage facility holds about 10,000 pallets in storage. Arrivals and departures are handled by 7 forklift operators and the average forklift travel time from receiving to a storage location and then to shipping is about 6 minutes. Estimate the inventory turns per year. Assume each driver works 8 hours per day for 250 days of the year.*

Question 2.12. *Suppose a dairy distributor, experiencing approximately constant demand for each sku, ships an average of 200 pallets per day. If there are approximately 1,000 occupied pallet positions, how long must the shelf life of its product be to avoid spoilage?*

Chapter 3

Warehouse operations

A warehouse reorganizes and repackages product. Product typically arrives packaged on a larger scale and leaves packaged on a smaller scale. In other words, an important function of this warehouse is to break down large chunks of product and redistribute it in smaller quantities. For example, some skus may arrive from the vendor or manufacturer in pallet quantities but be shipped out to customers in case quantities; other skus may arrive as cases but be shipped out as eaches; and some very fast-moving skus may arrive as pallets and be shipped out as eaches.

In such an environment the downstream warehouse operations are generally more labor-intensive.

This is still more true when product is handled as eaches. In general, *the smaller the handling unit, the greater the handling cost*. It can require much labor to move 10,000 boxes of paper clips if each box must be handled separately, as they may when, for example, stocking retail stores. Much less labor is required to handle those 10,000 boxes if they are packaged into cases of 48 boxes; and still less labor if those cases are stacked 24 to a pallet.

Even though warehouses can serve quite different ends, most share the same general pattern of material flow. Essentially, they receive bulk shipments, stage them for quick retrieval; then, in response to customer requests, retrieve and sort skus, and ship them out to customers.

The reorganization of product takes place through the following physical processes (Figure 3.1).

- Inbound processes
  - Receiving
  - Put-away
- Outbound processes
  - Order-picking
  - Checking, packing, shipping

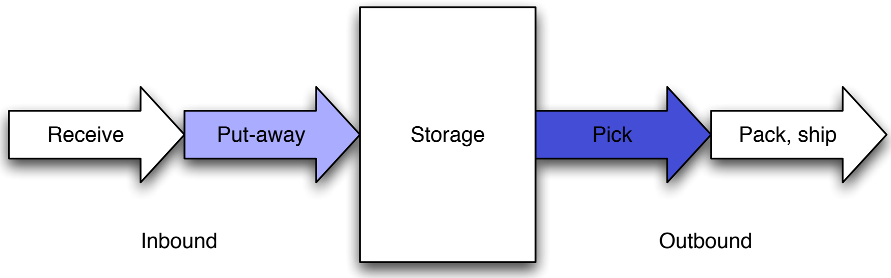


Figure 3.1: Order-picking is the most labor-intensive activity in most warehouses. Travel can be reduced by careful putaway.

A general rule is that product should, as much as possible, flow continuously through this sequence of processes. Each time it is put down means that it must be picked up again sometime later, which is double-handling. When such double-handling is summed over all the tens-of-thousands of skus and hundreds-of-thousands of pieces and/or cases in a warehouse, the cost can be considerable.

Another rule is that product should be scanned at all key decision points to give "total visibility of assets", which enables quick and accurate response to customer demand.

3.1 Receiving

Receiving may begin with advance notification of the arrival of goods. This allows the warehouse to schedule receipt and unloading to coördinate efficiently with other activities within the warehouse. It is not unusual for warehouses to schedule trucks to within 30-minute time windows.

Once the product has arrived, it is unloaded and possibly staged for put away. It is likely to be scanned to register its arrival so that ownership is assumed, payments dispatched, and so that it is known to be available to fulfill customer demand. Product will be inspected and any exceptions noted, such as damage, incorrect counts, wrong descriptions, and so on.

Product typically arrives in larger units, such as pallets, from upstream and so labor requirements are not usually great. (However, mixed pallets may need to be broken out into separate cartons; and loose cartons may need to be palletized for storage.) All-in-all, receiving accounts for only about 10% of operating costs in a typical distribution center [21, 22]—and RFID is expected to further reduce this.

3.2 Put-away

Before product can be put away, an appropriate storage location must be determined. This is very important because where you store the product determines to a large extent how quickly and at what cost you later retrieve it for a customer. This requires managing a second inventory, not of product, but of storage locations. You must know at all times what storage locations are available, how large they are, how much weight they can bear, and so on.

When product is put away, the storage location should also be scanned to record where the product has been placed. This information will subsequently be used to construct efficient pick-lists to guide the order-pickers in retrieving the product for customers.

Put-away can require a fair amount of labor because product may need to be moved considerable distance to its storage location. Put-away typically accounts for about 15% of warehouse operating expenses [21].

3.3 Order-picking

On receipt of a customer order the warehouse must perform checks such as verifying that inventory is available to ship. Then the warehouse must produce pick lists to guide the order-picking. Finally, it must produce any necessary shipping documentation and schedule the order-picking and shipping. These activities are typically accomplished by a *warehouse management system*, a large software system that coördinates the activities of the warehouse. This is all part of the support to expedite the sending of the product to the customer.

Order-picking typically accounts for about 55% of warehouse operating costs; and order-picking itself may be further broken like this [21]:

Activity	% Order-picking time
Traveling	55%
Searching	15%
Extracting	10%
Paperwork and other activities	20%

Notice that traveling comprises the greatest part of the expense of order-picking, which is itself the most expensive part of warehouse operating expenses. Much of the design of the order-picking process is directed to reducing this unproductive time.

The outbound processes of the warehouse are initiated by receipt of a customer order, which may be thought of as a shopping list. Each entry on the list is referred to as an *order-line* and typically consists of the item and quantity requested. The warehouse management system (WMS) then checks the order against available inventory and identifies any shortages. In addition, the WMS may reörganize the list to match the layout and operations of the warehouse for greater efficiency. For example, if a customer has ordered 15 of a particular item, the warehouse management system (WMS) may check to see how the item is packaged. If 12 of the item comprise a carton, the WMS may convert the order-line for 15 eaches to two *pick-lines*, one for 1 carton and the other for 3 eaches. In many warehouses, each-picking and carton-picking are separate processes, and the pick-lines are diverted appropriately.

Pick-lines are instructions to the order-pickers, telling them where and what to pick and in what quantity and units of measure. Each pick-line (or, more briefly, *pick* or *line*) represents a location to be visited, and since travel is the largest labor cost in a typical warehouse, the number of pick-lines is an indication of the labor required.

Note that a pick (line) may require more than one *grab* if, for example, several items of a sku are to be retrieved for an order. Generally, this represents a much smaller proportion of the labor, because it is controllable by appropriate packaging (for example, pick one carton rather than 12 eaches).

The WMS organizes pick-lines into *pick-lists* to achieve still more efficiencies, so that an order-picker may be able to concentrate on one area of the warehouse and so reduce travel. In addition, the WMS may sequence the pick-lines so that the locations to be visited appear in the sequence in which they will normally be encountered as the picker moves through the warehouse. (This will be explored in more detail in Chapter 10).

The pick-list may be a physical sheet of paper, or merely a sequence of requests communicated by a stream of printed shipping labels, or by light, RF, or voice transmission.

The most labor-intensive order-picking is the picking of less-than-carton quantities, referred to typically as *broken-case* or *split-case* picking. Broken-case picking is labor-intensive because it requires handling the smallest units of measure in the warehouse and this is generally resistant to automation because of the size and variety of skus to be handled. In contrast, *carton-picking* (picking full cartons) can sometimes be automated because of the relative uniformity of cartons, which are almost always rectangular and packed to resist damage.

The *pick face* is that 2-dimensional surface, the front of storage, from which skus are extracted. This is how the skus are presented to the order picker. In general, the more different skus presented per area of the pick face, the less travel required per pick. An informal measure of this is *sku density*, which counts the number of skus available per unit of area on the pick-face. If a warehouse has a suitably high sku density then it will likely achieve a high value of *pick density*, or number of picks achieved per unit of area on the pick face, and so require less travel per pick.

Sometimes it is useful to interpret the informal measures sku density and pick density as measuring skus or picks per unit of distance along the aisle traveled by an order-picker. One can then talk about, for example, the pick density of an order. An order that is of high pick density does not require much travel per pick and so is expected to be relatively economical to retrieve: we are paying only for the actual cost of retrieval and not for travel. On the other hand, small orders that require visits to widely dispersed locations may be expensive to retrieve because there is more travel per pick.

Pick density depends on the orders and so we cannot know it precisely in advance of order receipt. However, it is generally true that pick density can be improved by ensuring high *sku density*, which is number of skus per foot of travel.

Pick density can be increased, at least locally, by storing the most popular skus together. Then order-pickers can make more picks in a small area, which means less walking.

Another way to increase the pick density is to *batch* orders; that is, have each worker retrieve many orders in one trip. However, this requires that the items be sorted into orders either while picking or else downstream. In the first case, the pickers are slowed down because they must carry a container for each order and they must sort the items as they pick, which is time-consuming and can lead to errors. If the items are sorted downstream, space and labor must be devoted to this additional process. In both cases even more work and space may be required if, in addition, the orders themselves must be sorted to arrive at the trailer in reverse sequence of delivery.

It is generally economic to batch single-line orders. These orders are easy to manage since there is no need to sort while picking and they can frequently be picked directly into a shipping container.

Very large orders can offer similar economies, at least if the skus are small enough so that a single picker can accumulate everything requested. A single worker can pick that order with little walking per pick and with no sortation.

The challenge is to economically pick the orders of intermediate size; that is, more than two pick-lines but too few to sufficiently amortize the cost of walking. Roughly speaking, it is better to batch orders when the costs of work to separate the orders and the costs of additional space are less than the extra walking incurred if orders are not batched. It is almost always better to batch single-line orders because no sortation is required. Very large orders do not need to be batched because they will have sufficient pick density on their own. The problem then is with orders of medium-size.

To sustain order-picking product must also be replenished. Restockers move skus in larger units of measure (cartons, pallets) and so a few restockers can keep many pickers supplied. A rule of thumb is one restocker to every five pickers; but this will depend on the particular patterns of flow.

A restock is more expensive than a pick because the restocker must generally retrieve product from bulk storage and then prepare each pallet or case for picking. For example, he may remove shrink-wrap from a pallet so individual cases can be retrieved; or he may cut individual cases open so individual pieces can be retrieved.

3.3.1 Sharing the work of order-picking

A customer order may be picked entirely by one worker; or by many workers but only one at a time; or by many at once. The appropriate strategy depends on many things, but one of the most important is how quickly must orders flow through the process. For example, if all the orders are known before beginning to pick, then we can plan efficient picking strategies in advance. If, on the other hand, orders arrive in real time and must be picked in time to meet shipping schedules then we have little or no time in which to seek efficiencies.

A general decision to be made is whether a typical order should be picked in serial (by a single worker at a time) or in parallel (by multiple workers at a time). The general trade-off is that picking serially can take longer to complete an order but avoids the complications of coördinating multiple pickers and consolidating their work.

A key statistic is *flow time*: how much time elapses from the arrival of an order into our system until it is loaded onto a truck for shipping? In general, it is good to reduce flow time because that means that orders move quickly through our hands to the customer, which means increased service and responsiveness.

A rough estimate of the total work in an order is the following. Most warehouses track picker productivity and so can report the average picks per person-hour. The inverse of this is the average person-hours per pick and the average work per order is then the average number of pick lines per order times the average person-hours per pick. A rough estimate of the total work to pick the skus for a truck is the sum of the work-contents of all the orders to go on the truck. This estimate now helps determine our design: How should this work be shared?

- If the total work to pick and load a truck is small enough, then one picker may be devoted to an entire truck. This would be a rather low level of activity for a commercial warehouse.

- If the total work to pick and load an order is small enough, then we might repeatedly assign the next available picker to the next waiting order.

- If the orders are large or span distant regions of the warehouse or must flow through the system very quickly we may have to share the work of each order with several, perhaps many, pickers. This ensures that each order is picked quickly; but there is a cost to this: Customers typically insist on *shipment integrity*, which means that they want everything they ordered in as few packages as possible, to reduce their shipping costs and the handling costs they incur on receipt of the product. Consequently, we have to assemble the various pieces of the order that have been picked by different people in different areas of the warehouse; and this additional process is labor-intensive and slow or else automated.

- For warehouses that move a lot of small product for each of many customers, such as those supporting retail stores, order-picking may be organized as an assembly-line: The warehouse is partitioned into zones corresponding to workstations, pickers are assigned to zones, and workers progressively assemble each order, passing it along from zone to zone.

  Advantages include that the orders emerge in the same sequence they were released, which means you make truck-loading easier by releasing orders in reverse order of delivery. Also, order-pickers tend to concentrate in one part of the warehouse and so are able to take advantage of the learning curve.

  The problem with zone-picking is that it requires all the work of balancing an assembly line: A work-content model and a partition of that work. Typically this is done by an industrial engineer.

Warehouses tend to use combinations of several of these approaches.

3.4 Checking and packing

Packing can be labor-intensive because each piece of a customer order must be handled; but there is little walking. And because each piece will be handled, this is a convenient time to check that the customer order is complete and accurate. Order accuracy is a key measure of service to the customer, which is, in turn, that on which most businesses compete.

Inaccurate orders not only annoy customers by disrupting their operations, they also generate returns; and returns are expensive to handle (up to ten times the cost of shipping the product out).

One complication of packing is that customers generally prefer to receive all the parts of their order in as few containers as possible because this reduces shipping and handling charges. This means that care must be taken to try to get all the parts of an order to arrive at packing together. Otherwise partial shipments must be staged, waiting completion before packing, or else partial orders must be packaged and sent.

Amazon, the web-based merchant, will likely ship separate packages if you order two books fifteen minutes apart. For them rapid response is essential and so product is never staged. They can ship separate packages because their customers do not mind and Amazon is willing to pay the additional shipping as part of customer service.

Packed product may be scanned to register the availability of a customer order for shipping. This also begins the tracking of the individual containers that are about to leave the warehouse and enter the system of a shipper.

3.5 Shipping

Shipping generally handles larger units than picking, because packing has consolidated the items into fewer containers (cases, pallets). Consequently, there is still less labor here. There may be some walking if product is staged before being loaded into freight carriers.

Product is likely to be staged if it must be loaded in reverse order of delivery or if shipping long distances, when one must work hard to completely fill each trailer. Staging freight creates more work because staged freight must be double-handled.

The trailer is likely to be scanned here to register its departure from the warehouse. In addition, an inventory update may be sent to the customer.

3.6 Summary

Most of the expense in a typical warehouse is in labor; most of that is in order-picking; and most of that is in travel.

3.7 More

Many warehouses also must handle returns, which run about 5% in retail. This will become a major function within any warehouse supporting e-commerce, where returns run 25–30%, comparable to those supporting catalog sales.

Another trend is for warehouses to assume more value-added processing (VAP), which is additional work beyond that of building and shipping customer orders. Typical value-added processing includes the following:

- Ticketing or labeling (For example, New York state requires all items in a pharmacy to be price-labeled and many distributors do this while picking the items in the warehouse.)

- Monogramming or alterations (For example, these services are offered by Lands End, a catalog and e-mail merchant of clothing)
- Repackaging
- Kitting (repackaging items to form a new item)
- Postponement of final assembly, OEM labeling (For example, many manufacturers of computer equipment complete assembly and packaging in the warehouse, as the product is being packaged and shipped.)
- Invoicing

Such work may be pushed on warehouses by manufacturers upstream who want to postpone product differentiation. By postponing product differentiation, upstream distributors, in effect, see more aggregate demand for their (undifferentiated) product. For example, a manufacturer can concentrate on laptop computers rather than on multiple smaller markets, such as laptop computers configured for an English-speaking market and running Windows 2000, those for a German-speaking market and running Linux, and so on. This aggregate demand is easier to forecast because it has less variance (recall the Law of Large Numbers!), which means that less safety stock is required to guarantee service levels.

At the same time value-added processing is pushed back onto the warehouse from retail stores, where it is just too expensive to do. Both land and labor are typically more expensive at the retail outlet and it is preferable to have staff there concentrate on dealing with the customer.

3.8 Questions

Question 3.1. *What are the basic inbound operations of a warehouse? What are the outbound operations? Which is likely to be most labor intensive and why?*

Question 3.2. *At what points in the path of product through a warehouse is scanning likely to be used and why? What is scanned and why?*

Question 3.3. *What is "batch-picking" and what are its costs and benefits?*

Question 3.4. *What are the issues involved in determining an appropriate batch-size?*

Question 3.5. *What are the costs and benefits of ensuring that orders arrive at the trailers in reverse sequence of delivery?*

Question 3.6. *Explain the economic forces that are pushing more value-added processing onto the warehouses.*

Question 3.7. *Why is the receiving staff in a typical warehouse much smaller than the staff of order-pickers?*

Question 3.8. *Explain the difference between:*

- *A customer order and a pick list*
- *A pick-line and a sku*
- *An order and a batch*
- *Popularity and demand*

Question 3.9. *Define the following:*

- *Pick face*
- *Sku density*
- *Pick density*

Chapter 4

Warehouse management systems

A large modern distribution center might extend over 600,000 square feet (60,000 square meters), contain a hundred thousand SKUs, and have hundreds of people working to gather and consolidate thousands of customer orders in time to meet daily shipping schedules.

How can such coordination be achieved?

A *Warehouse Management System* (WMS) is a complex software package that helps manage inventory, storage locations, and the workforce, to ensure that customer orders are picked quickly, packed, and shipped. A typical WMS knows about every item in the warehouse, its physical dimensions, how it is packed by the vendor, all the storage locations in the warehouse, and their addresses and physical dimensions. With this knowledge, the WMS orchestrates the flow of people, machines, and product.

The WMS receives customer orders and transforms them to *pick lists* organized for easy retrieval: In customer orders items appear in arbitrary sequence, just like the grocery shopping list one might casually prepare during the week. When it is time to shop, it may be worthwhile to reorganize entries for convenience (all the dairy items together, all the fresh fruits and vegetables, and so on).

Finally, the WMS tracks the assembly of each customer order.

The scope of WMS is growing, as it acquires new responsibilities, such inducting newly arrived product and allocating available locations, coordinating the assembly of customer orders to meet shipping schedules, tracking productivity of workers, and so on. It may even talk to other specialized software such as *Yard Management Systems* (YMS), which coordinates the movement of full and empty trailers in the yard (a sort of warehouse of trailers). Finally, the WMS may provide summary data to an even larger *Supply Chain Management System* (SCMS) that plans and coordinates inventory levels and transportation from manufacturer to customer.

It is thanks to the control afforded by software systems such as WMS that the pace of the supply chain has accelerated so much during the last 20 years. Not so very long ago any customer order was accompanied by the warning "Please allow six to eight weeks for delivery". No one would put up with such service today. Precisely controlled product moves faster, which means that customers get better service, and with less inventory in the system.

4.1 Receiving and shipping

The most fundamental capability of a WMS is to record receipt of inventory into the warehouse and to register its shipment out. This is fundamental because it drives essential financial transactions: receipt drives the paying of bills to suppliers upstream; and shipping drives the sending of invoices downstream to the consignee. This is the base from which modern, complex WMS's have grown.

4.2 Stock locator system

The next significant increase in functionality is to add a stock locator system. This is essentially the ability to manage an inventory of storage locations in addition to an inventory of product. With this capability a software system can do more than trigger financial transactions but can finally support warehouse operations by directing warehouse activities to/from storage locations.

In addition, a WMS should also track the inventory of storage locations in the warehouse. A good WMS will track every place that product can be, down to and including the forks of individual forklift trucks. The ability to manage the inventory of storage locations makes possible the most fundamental capability of a WMS, which is the *stock locator* system, which supports directed put-away and directed picking.

To track warehouse activities in real-time, the database must support transaction processing, which means that the database can maintain its integrity even though being updated simultaneously from multiple sources (purchasing, receiving, picking, shipping, etc.).

4.3 Menu of features

- Basic features of most WMS's include tools to support
  - Appointment scheduling
  - Receiving
  - Quality assurance
  - Put-away
  - Location tracking
  - Work-order management
  - Picking
  - Packing and consolidation
  - Shipping

- High-end features include support for
  - RF-directed operation
  - Cycle counting
  - Carton manifesting
  - Replenishment
  - Value-added services
  - Vendor/carrier compliance
  - Trailer manifesting
  - Configurability
  - Returns
  - Pick/put to light
  - Yard management
  - Wave management
  - Labor management
  - Task interleaving
  - Flow-through processing
- Advanced features include support for
  - Multi-DC view
  - Sku slotting
  - Broken-case flow
  - EDI capability
  - Parcel shipping
  - Impact analysis
  - Traffic management
  - Import/export management
  - ASP capability

4.4 The market

At the time of this writing, there are over 300 WMS vendors in the US alone. The largest companies hold less than 20% of the market.

4.5 Supply Chain Execution Systems

Warehouse Management Systems are extending their functionality out along the supply chain, both upstream and downstream, to include features that support collaboration. In this they are increasingly competing with enterprise systems, which are trying to build specialization in warehouse management. WMS's have an advantage in that they are already connected to financial systems and already hold information that is important to supply chain visibility and execution systems. The enterprise systems have advantages where they can grow out of a manufacturing enterprise, especially if the manufactured product is an important, high-value item.

As WMS's grow out along the supply chain it is natural that the WMS providers become global, pulled by the supply chains they hope to manage. The global WMS providers have a big advantage when selling to multinational companies, who can then standardize their WMS operations around the world.

At the time of this writing, there are hundreds if not thousands of WMS vendors in the world but only a few companies with significant global presence.

4.6 Summary

The core of a Warehouse Management System (WMS) is a database of skus and a stock locator system so that one can manage both the inventory of skus and the inventory of storage locations.

There are significant opportunities to save labor when the system is expanded to include systems to direct receiving/put-away and/or order-picking. Additional systems are available to control ever finer detail within the warehouse as well as events ever farther up/downstream in the supply chain.

4.6.1 More

The working life of a warehouse management system is generally greater than that of the computer language in which it was written. Consequently, most WMS's in current use are an accretion of many different computer languages, including COBOL, PL1, Fortran, C, C++, SQL, and others. This can make them hard to maintain or customize.

Most WMS's manage transactions but do not currently optimize anything. Instead, they are extremely configurable and let the user choose from among various rules to guide decision-making. But the client—or more likely a consultant—must choose the logic from among the modules provided, or else have the software customized.

Most vendors will customize their WMS for the right price; and in fact, some derive the bulk of their revenue from customization. But it is typical that the vendor owns the intellectual property inherent in the customization.

See `www.mywms.de` for an interesting open source project to write a warehouse management system in Java.

Chapter 5

Storage and handling equipment

There are many types of special equipment that have been designed to reduce labor costs and/or increase space utilization.

Storage and retrieval equipment can reduce labor costs by

- Allowing many skus to be on the pick face, which increases pick density and so reduces travel per pick, which means more picks per person-hour
- Facilitating efficient picking and/or restocking by making the product easier to handle (for example, by presenting it at a convenient height and orientation).
- Moving product from receiving to storage; or from storage to shipping.

Storage equipment can increase space utilization by:

- Partitioning space into subregions (bays, shelves) that can be loaded with similarly-sized skus. This enables denser packing and helps make material-handling processes uniform.
- Making it possible to store product high, where, up to about 30 feet (10 meters), space is relatively inexpensive. (Above this height, the building requires additional structural elements.)

5.1 Storage equipment

By *storage mode* we mean a region of storage or a piece of equipment for which the costs to pick from any location are all approximately equal and the costs to restock any location are all approximately equal.

Common storage modes include pallet rack for bulk storage, carton flow rack for high-volume picking, and (static) shelving for slower, lower-volume picking.

Stringer length $\times$ deckboard length	**Most prevalent in**
1219×1016 mm $(48.00 \times 40.00$ in)	North America
1000×1200 mm $(39.37 \times 47.24$ in)	Europe, Asia
1165×1165 mm $(44.88 \times 44.88$ in)	Australia
1067×1067 mm $(42.00 \times 42.00$ in)	North America, Europe, Asia
1100×1100 mm $(43.30 \times 43.30$ in)	Asia
800×1200 mm $(31.50 \times 47.24$ in)	Europe

Table 5.1: The six standard sizes of pallet, from *ISO Standard 6780: Flat pallets for intercontinental materials handling – Principal dimensions and tolerances*. (The *stringers* are the supports underneath that are spanned by the *deckboards*.)

5.1.1 Pallet storage

Within the warehouse the largest standardized material-handling unit is generally the *pallet*, which is just a rigid base on which cartons can be stacked. Most are made of wood, but some are made of durable plastic.

Pallets are available in a range of qualities and prices. In general order of quality and price, they include string pallets, block pallets, and perimeter base pallets. Any pallet expected to be handled by automation will generally have to be of high quality. As supply chains get longer, there is an incentive to use higher quality pallets.

There are several standards, the most important of which appear in Table 5.1. Of these, the most common pallet in North America is the 1219×1016 mm (48×40 inch) pallet, also known as the Grocery Manufacturer's Association or GMA pallet. (The 1000×1200 mm pallet is generally interchangeable with the GMA pallet.)

Some pallets are designed with special uses in mind. For example, the Australian pallet was designed to be space efficient in Australian railroad cars; and the *EURO pallet* was designed to fit through doorways. Neither is very space efficient in ISO standard shipping containers, while the GMA pallet fits well.

A *2-way* pallet allows forks from a standard forklift or pallet jack to be inserted on either of the 40 inch sides. A *4-way* pallet also has slots on the 48 inch sides by which it can be lifted by fork lift. The 4-way pallets are slightly more expensive, but the extra flexibility in handling can save both time and space. Less maneuvering is required to pick up such a pallet, and it can be oriented in either direction even in confined space.

There is no standard height to which a pallet may be loaded.

The simplest way of storing palletized product is floor storage, which is typically arranged in *lanes*. The *depth* of a lane is the number of pallets stored back-to-back away from the pick aisle. The height of a lane is normally measured as the maximum number of pallets that can be stacked one on top of each other, which is determined by pallet weight, fragility, number of cartons per pallet, and so on. Note that the entire footprint of a lane is reserved for a sku if any part of the lane is currently storing a pallet. This rule is almost always applied, since if more than one sku was stored in a lane, some pallets may be double-handled during retrieval, which could offset any space savings. Also, it becomes harder to keep track of where product is stored. For similar reasons, each column is devoted to a single sku.

This loss of space is called *honey-combing*.

Pallet rack is used for bulk storage and to support full-case picking (Figure 5.1). Pallet length and width are reasonably uniform and pallet rack provides appropriately-sized slots. The height of slots can be adjusted, however, as pallet loads can vary in height.

The advantage of rack storage is that each level of the rack is independently supported, thus providing much greater access to the loads, and possibly permitting greater stack height that might be possible in floor storage.

The most common types of rack storage are:

Selective rack or single-deep rack stores pallets one deep, as in Figure 5.1. Due to rack supports each pallet is independently accessible, and so any sku can be retrieved from any pallet location at any level of the rack. This gives complete freedom to retrieve any individual pallet but requires relatively more aisle space to access the pallets.

Double-deep rack essentially consists of two single-deep racks placed one behind the other, and so pallets are stored two deep. Due to rack supports each 2-deep lane is independently accessible, and so any sku can be stored in any lane at any level of the rack. To avoid double-handling, it is usual that each lane be filled with a single sku, which means that some pallet locations will be unoccupied whenever some sku is present in an odd number of pallets. Another disadvantage of deep lanes is that slightly more work is required to store and retrieve product. However, deep lanes have the advantage of requiring fewer aisles to access the pallets, which means that the warehouse can hold more product. A special truck is required to reach past the first pallet position.

Push-back rack. This may be imagined to be an extension of double deep rack to 3–5 pallet positions, but to make the interior positions accessible, the rack in each lane pulls out like a drawer. This means that each lane (at any level) is independently accessible.

Drive-In or drive-through rack allows a lift truck to drive within the rack frame to access the interior loads; but, again to avoid double-handling, all the levels of each lane must be devoted to a single sku. With drive-in rack the put-away and retrieval functions are performed from the same aisle. With drive-through rack the pallets enter from one end of the lane and leave from the other, so that product can be moved according to a policy of First-In-First-Out (FIFO). Drive-in/through rack may be thought of as floor-storage for product that is not otherwise stackable. It does not enable the flexibility of access that other types of pallet rack achieve. In addition, there are some concerns; for example, in this rack each pallet is supported only by the edges, which requires that the pallets be strong. In addition, it requires a more skilled forklift driver to navigate within the lane, and such a person will be more expensive.

Pallet flow rack is deep lane rack in which the shelving is slanted and lines with rollers, so that when a pallet is removed, gravity pulls the remainder to the front. This enables pallets to be put-away at one side and retrieved from the other,

Figure 5.1: Simple pallet rack. (Adapted from "Warehouse Modernization and Layout Planning Guide", Department of the Navy, Naval Supply Systems Command, NAVSUP Publication 529, March 1985, p 8–17).

which prevents storage and retrieval operations from interfering with each other. Because of weight considerations, storage depth is usually limited to about eight pallets. This type of rack is appropriate for high-throughput facilities.

Except for automated storage-and-retrieval systems (AS/RS), some type of *lift truck* is required to access the loads in pallet rack; and specialized racks may require specialized trucks. The most common type of lift trucks are:

Counterbalance lift truck is the most versatile type of lift truck. The sit-down version requires an aisle width of 12–15 feet (3.7–4.6 meters), its lift height is limited to 20–22 feet (6.1–6.7 meters), and it travels at about 70 feet/minute (21.3 meters/minute). The stand-up version requires an aisle width of 10–12 feet (3.1–3.7 meters), its lift height is limited to 20 feet (6.1 meters), and it travels at about 65 feet/minute (19.8 meters/minute).

Reach and double-reach lift truck is equipped with a reach mechanism that allows its forks to extend to store and retrieve a pallet. The double-reach truck is required to access the rear positions in double deep rack storage. Each truck requires an aisle width of 7–9 feet (2.1–2.7 meters), their lift height is limited to 30 feet (9.1 meters), and they travel at about 50 feet/minute (15.2 meters/minute). A reach lift truck is generally supported by "outriggers" that extend forward under the forks. To accommodate these outriggers, the bottom level of deep pallet rack is generally raised a few inches (approximately 10 centimeters) off the ground so that the outriggers can pass under.

Turret Truck uses a turret that turns 90 degrees, left or right, to put-away and retrieve loads. Since the truck itself does not turn within the aisle, an aisle width of only 5–7 feet (1.5–2.1 meters) is required, its lift height is limited to 40–45 feet (12.2–13.7 meters), and it travels at about 75 feet/minute (22.9 meters/minute). Because this truck allows such narrow aisle, some kind of guidance device, such as rails, wire, or tape, is usually required. It only operates within single deep rack and super flat floors are required, which adds to the expense of the facility. This type of truck is not easily maneuverable outside the rack.

Stacker crane within an AS/RS is the handling component of a unit-load AS/RS, and so it is designed to handle loads up to 100 feet high (30.5 meters). Roof or floor-mounted tracks are used to guide the crane. The aisle width is about 6–8 inches (0.15–0.20 meters) wider than the unit load. Often, each crane is restricted to a single lane, though there are, at extra expense, mechanisms to move the crane from one aisle to another.

5.1.2 Bin-shelving or static rack

Simple shelving is the most basic storage mode and the least expensive (Figure 5.2). The shelves are shallow: 18 or 24 inches (0.46 or 0.61 meters) are typical, for example, but 36 inch (0.91 meter) deep shelf is sometimes used for larger cartons. Because the

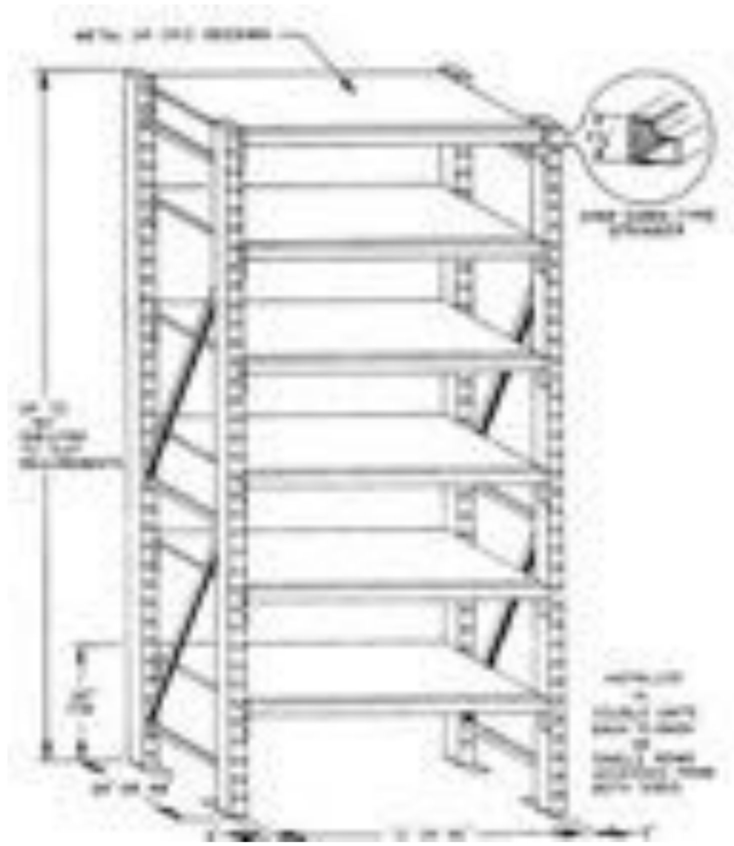


Figure 5.2: Shelving, or static rack. (Adapted from “Warehouse Modernization and Layout Planning Guide”, Department of the Navy, Naval Supply Systems Command, NAVSUP Publication 529, March 1985, p 8–17.)

shelves are shallow, any significant quantity of a sku must spread out along the pick-face. This reduces sku-density and therefore tends to reduce pick density, increase travel time, and reduce picks/person-hour.

Skus which occupy more than one shelf of bin-shelving are candidates for storage in another mode that will enable greater sku-density.

A typical pick rate from bin-shelving is 50–100 picks/person-hour. (Of course this and the pick rates for any equipment depends on which skus are stored there.)

With shelving, both picking and restocking must be done from the pick-face and so, to avoid interference, must be scheduled at different times. This can mean working an additional shift.

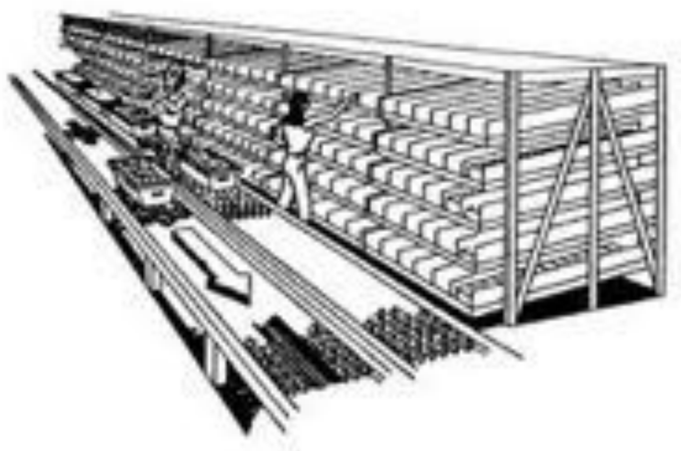


Figure 5.3: Gravity flow rack for cartons. (Adapted from "Warehouse Modernization and Layout Planning Guide", Department of the Navy, Naval Supply Systems Command, NAVSUP Publication 529, March 1985, p 8-17.)

5.1.3 Gravity flow rack

Flow rack is a special type of shelving with shelves that are tilted, with rollers, to bring cases forward for picking (Figure 5.3). The shelves may be 3–10 feet deep (0.91–3.0 meters). This means that only one case of a product need be on the pick face, which means that many skus can be available in a small area of pick-face. This means high sku-density, which tends to increase the pick-density, decrease travel, and increase picks/person-hour.

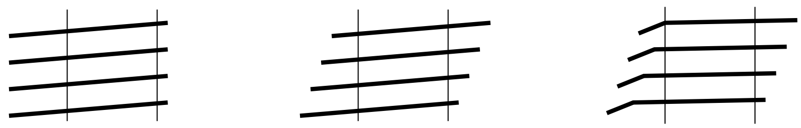
Frequently the picking from flow rack is accelerated by supporting technology such as a pick-to-light system, by which a centralized computer lights up signals at every location within a bay from which product is to be picked. After the worker picks the appropriate quantity, who pushes a button to signal the computer. There are several benefits of this: The order-picker is freed from handling a paper pick-list; he does not have to search for the next storage location; and because picking is guided, it is more accurate.

A typical pick rate from flow rack is about 150–500 picks/person-hour, but this varies widely.

Flow rack is restocked from the back, independently of picking. This means that restocking never interferes with picking, as is the case for static shelving, for which picking and restocking must alternate. However, additional aisle space is required to access the back of flow rack for restocking.

There are several subtypes of carton flow rack, as shown in Figure 5.4.

- Square front, vertical frame flow rack is suited to picking full cases, such as canned goods. (This is a specialized use, however, because it requires full-cases in and full-cases out, which suggests excessive handling.)



(a) Vertical frame (b) Layback shelves (c) Front-tilted shelves

Figure 5.4: Side views of the shelves of three styles of carton flow rack. Each successive configuration presents the product more conveniently but at the cost of consuming more space. Configurations (b) and (c) occupy more horizontal space than (a) and configuration (c) also occupies more vertical space.

- "Layback frame" presents the forwardmost carton more fully to the order picker by reducing the overhang of upper shelves, which might otherwise interfere with picking. This style of rack is suited to picking from open cases when those cases vary in size, such as health and beauty aids; but this rack occupies more space than vertical frame rack of equal capacity.
- Layback frame with front-tilted shelves tips the forwardmost cartons forward at a greater angle so that the entire carton is accessible. This is best suited to picking from open cases when those cases are similar in size, such as liquor or books.

5.2 Conveyors

Main points:

- Conveyors change the economics of travel: Storage locations close to the conveyor are, in terms of labor, close to shipping.
- Conveyors partition the warehouse into zones. The track restricts movement of workers and product because it is hard to cross; and so create problems of balancing work among zones. To alleviate this, conveyors are run up high whenever possible.
- Issues: How many products are conveyable? What is capacity, especially surge capacity, of conveyor?
- Guidelines for layout: Store conveyable product far from shipping because it can travel "for free". Reserve locations that are physically close to shipping for non-conveyables because they will have to be carried (for example, fork lift).

5.2.1 Sortation equipment

Sortation equipment is an expensive form of automation that is typically integrated with a conveyor system and installed downstream from picking. It is used mostly when picking cartons, because they tend to be fairly uniform in dimension and weight.

A sortation system enables pick lists to be constructed purely for efficiency. For example, if twenty customers all want sku A, it might make sense to send one order picker to pick all the requested sku A in one trip and rely on the sortation system to separate it all out for the customers.

Naturally this requires scanning technologies (bar codes or RFID) and significant IT support for real-time control.

There are many different types of sorters, depending on required speeds and types of material to be handled. One common sorter is a *push sorter*, which simply pushes a passing carton off the main conveyor and onto an alternative path, such as onto a spur at which an order collects. A *tilt-tray sorter* is used for material that cannot be easily pushed but can slide, such as apparel. Also, a tilt-tray sorter does not need to know the orientation of the item it carries, while a push sorter typically must know the size and orientation.

Tilt trays serve as both conveyor and sorter; but they must circulate and so must be built as loop. In contrast, a belt conveyor can be run from one point to another and so can be cheaper.

It is a challenge to design an effective sortation system because it must handle a range of sizes, shapes, and textures; and it must have sufficient capacity as well as the ability to handle surges. Another important design issue is to decide how many spurs are required, because this limits the number of orders that can be picked at a time. There are also challenges in deciding how orders should be assigned to spurs and how recirculation should be managed (exit the system? recirculate? divert into a separate recirculation lane?).

As with all automation, there is an element of risk: A broken sorter can idle the entire warehouse.

5.3 Summary

The most common equipment is pallet rack, with corresponding trucks, for pallets; simple shelving for slower-moving and/or small items; and gravity flow rack for cartons of faster-moving items.

5.4 On the lighter side

When space is expensive economics says to store product high, which is cheaper than expanding the warehouse out. This has led to some pallet storage at dizzying heights. For example, three stories is not unusual. Automated storage and retrieval devices are generally used for dangerous heights but not always. We heard of an IBM site that relies on intrepid workers on person-aboard trucks that lift them high into the air. They are safely linked to their truck and so cannot fall; but the problem is how to get them down in case of accident. IBM requires that all order-pickers in this part of the warehouse be trained in rappelling!

5.5 Questions

Question 5.1. *What are the advantages of a "4-way" pallet, in terms of the resources of space and time, compared to a "2-way" pallet?*

Question 5.2. *Which types of pallet storage generally provide the most efficient use of floor space: floor storage or pallet rack? Explain.*

Question 5.3. *What are the relative advantages and disadvantages of push-back rack compared to pallet rack? How do both compare to gravity flow rack?*

Question 5.4. *Which type of storage generally makes pallets more accessible: Pallet flow rack or drive-in rack? Explain.*

Question 5.5. *Consider two skus: One a small, slow-moving sku of which the warehouse has only a small amount; and the other a small, fast-moving sku of which the warehouse has considerable quantity. Which is a candidate for shelving and which for flow rack? Explain.*

Question 5.6. *In a warehouse with conveyors, conveyable skus should be stored* far *from shipping; why?*

Question 5.7. *Suppose you have many slow-moving skus in less-than-pallet quantities and these are shipped as cases. What arguments are there for storing them in carton flow rack?*

Question 5.8. *What are the dimensions of a standard pallet? Go to the web and see how many other "standards" you can find.*

Question 5.9. *Under what conditions might you prefer to store pallets with the* shorter *side on the pick face (that is, along the aisle)? When might you want to store pallets with the* longer *side facing the aisle?*

Question 5.10. *In each case explain why the warehouse action described below is probably unwise.*

- *Storing cases in a gravity flow rack that is 0.5 meter deep*
- *Picking fast-moving product from static shelving*
- *Storing pallets in an aisle of single-deep rack that is free-standing (there is an aisle on either side of it)*
- *Storing product as free-standing eaches, with every piece lined up neatly*

Part II

Layout

When product is stored in convenient locations, then it is easy to retrieve when requested by a customer. But what is meant by "convenience" depends on models of labor and of space. These models are simplest for pallets, because they have some standard dimensions and are, in many cases, handled one at a time. The concept of "convenience" becomes progressively harder to pin down as we discuss smaller units of storage and handling, such as cartons and pieces.

Chapter 6

Layout of a unit-load area

The simplest type of warehouse is a *unit-load* warehouse, which means that only a single, common "unit" of material is handled at a time. The typical unit-load is a pallet. Because pallets are (mostly) standardized and are (mostly) handled one-at-a-time, both space and labor requirements scale: It takes about n times the space to store n pallets as for one; and it takes about n times the labor to handle n pallets as for one.

An example of unit-load is a *3rd party transshipment warehouse* that receives, stores, and forwards pallets. The 3rd party warehouse is a subcontractor to others for warehouse services. A 3rd party warehouse typically charges its customers for each pallet handled (received and later shipped); and rent for space occupied. In this chapter we study unit-load issues in the context of such a warehouse; but many warehouses have some portion of their activity devoted to moving unit-loads, as in Figure 6.1.

6.1 Space

One revenue source of the 3rd party warehouse is to charge rent by the pallet-month. But because the warehouse typically tallies its own expenses by the square-foot or square-meter (for example, rent of the building, climate control, cleaning, and so on), the warehouse naturally wants many pallet-positions per unit area. It can achieve this in two ways: by taking advantage of vertical space and by using deep lanes.

6.1.1 Rack or stack?

Pallets that can be stacked high allow many pallet positions per unit of floor space. Conversely, pallets that are unusually heavy or fragile or that have uneven top surfaces cannot be stacked very high and so render unusable all the space above. This waste may be avoided by installing pallet rack, so that pallets may be stored independently of each other.

How much pallet rack should be purchased and what should be stored therein? The value of the rack to the user depends on the sizes and movement patterns of the

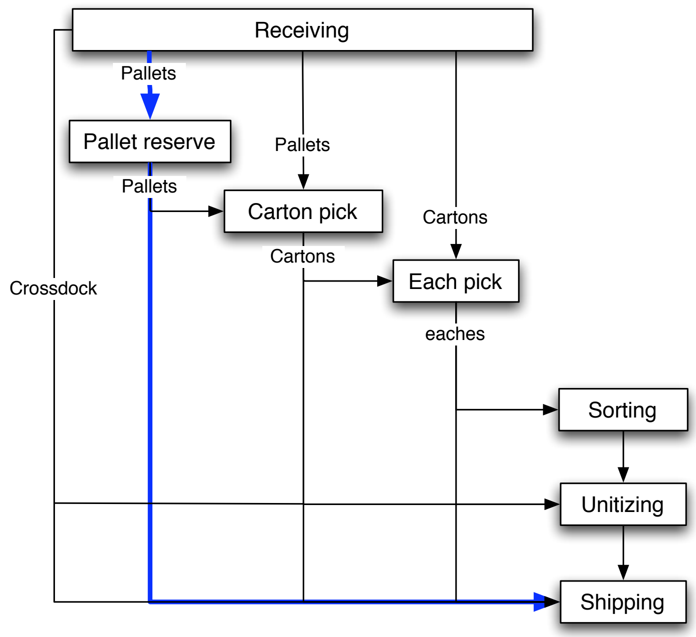


Figure 6.1: Flow of unit-loads through a typical warehouse

Figure 6.2: Pallets that require extra labor to retrieve

particular skus that visit the warehouse. We require that each sku present an economic argument for why it should be stored in pallet rack rather than floor storage.

For the moment, imagine that we are considering purchase of a particular type of pallet rack. (All dimensions are fixed; we are just trying to decide how much of it to purchase.) Suppose that we have n_i pallets of sku i and, for simplicity, these populations are fairly stable. The idea is that sku i should go into rack only if it is beneficial to do so. We just have to be careful to say exactly what we mean by "beneficial". Intuitively, we want the benefits we get from putting a sku in rack to justify the cost of the rack. What sorts of benefits might we get from putting skus in rack?

Here are some possible benefits of moving a sku from floor storage into pallet rack.

- It might reduce labor by making product easier to store and retrieve. For example, consider the pallet stack in Figure 6.2, where the uneven top pallet cannot be easily picked up by forklift. In this warehouse, a second forklift was needed to straighten the stack so that the first forklift could insert its forks under the pallet. This means that at least twice the labor was required to retrieve each pallet above the ground floor. This savings can be estimated easily and could be realized as increased throughput or reduced labor requirements.

- It might create additional pallet positions. For example, storing the pallets of Figure 6.2 in 4-high pallet rack would create an additional pallet position, which is potentially revenue-generating. The value of this additional pallet position depends on how easily it can be rented and so the extent of savings depends on the market for space.

  Some skus might not generate any new pallet positions if put in rack, and, in fact, might even lose pallet positions. For example, consider 3 pallets, each 4 feet (1.2 meters) high, of a sku that stack 3-high to come within a two feet (0.6 meters) of the ceiling. Moving these pallets to rack will not create additional pallet locations. In fact, it might not be possible to install 3-high pallet rack to hold these pallets because of the additional height required to accommodate the cross-beams of the rack and the space above each pallet in rack.

- It will help protect product from damage, for example, by forklifts. This savings is hard to quantify except by comparison to past experience.

- It might help provide a safer work environment by avoiding unstable pallet stacks. This savings is also hard to quantify.

Each sku builds its economic argument by estimating the savings for each of these categories and then sums the total savings. The result is a number of dollars, which represents the value of storing that sku in pallet rack. Now one can compare that the cost of pallet rack and decide, on a sku-by-sku basis, whether the pallet rack is economically justified.

Finally, to compare different rack configurations, repeat the process on each alternative and choose the one that is of greatest value.

6.1.2 Lane depth

Aisles provide accessibility, not storage, and so this space for aisles is not directly revenue-generating. Consequently we prefer to reduce aisle space to the minimum necessary to provide adequate accessibility. For this the aisles must be at least wide enough for a forklift to insert or extract a pallet.

By storing product in *lanes*, additional pallet positions can share the same aisle space and so amortize that cost. Should lanes be four pallets deep? Six? Ten? There are many issues to consider, but the most important one is effective utilization of space. For example, double-deep layout (lanes that are two pallet-positions deep) fits about 41% more pallet positions in the same floor area than does single-deep layout in the example of Figure 6.3; but is it a better layout? Are enough of the additional pallet positions usefully engaged? There is a trade-off: The single-deep layout has eight aisles and provides 196 pallet storage locations, all of which are directly accessible, which means that they are available for reassignment as soon as the current pallet is shipped out. In contrast, the double-deep layout has only six aisles and provides 280 pallet storage locations—but only 140 of them are directly accessible. Moreover, the 140 that are directly accessible are not available for reuse until the interior pallet location in the same lane becomes available. So: Deeper lanes produce more pallet storage locations but they are of diminishing value.

By *pallet position* we mean the floor space required to hold a pallet. This includes not only the footprint of the pallet but also any required gap between one pallet and an adjacent one (Figure 6.4). Let the lanes be k pallet positions deep. Each lane requires aisle space at its head so that pallets can be inserted and removed. Let the aisle space in front of the pallet be of area a, measured in pallet positions. Then the total area charged to one lane is $k + a/2$ pallet positions. This area is the sum of space devoted to storage (k) and space that provides accessibility ($a/2$).

In most warehouses a lane is dedicated entirely to a single sku to avoid double-handling pallets. This saves time but incurs a cost in space: When the first pallet is retrieved from a lane, that position is unoccupied but unavailable to other skus. The deeper the lane, the greater this cost. The first pallet position in a k-deep lane that holds uniformly moving product will be occupied only $1/k$ of the time, the second $2/k$ of

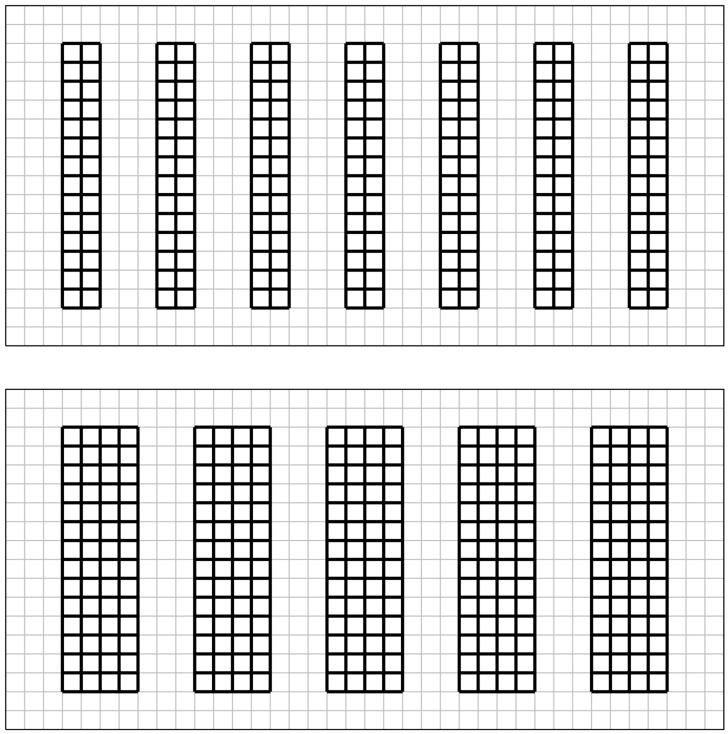


Figure 6.3: 1-deep versus 2-deep storage

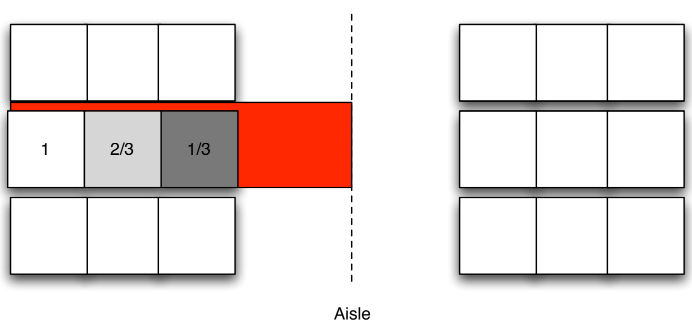


Figure 6.4: The floor space charged to a lane includes storage space, any gap between lanes, and one-half the aisle width in front of the lane.

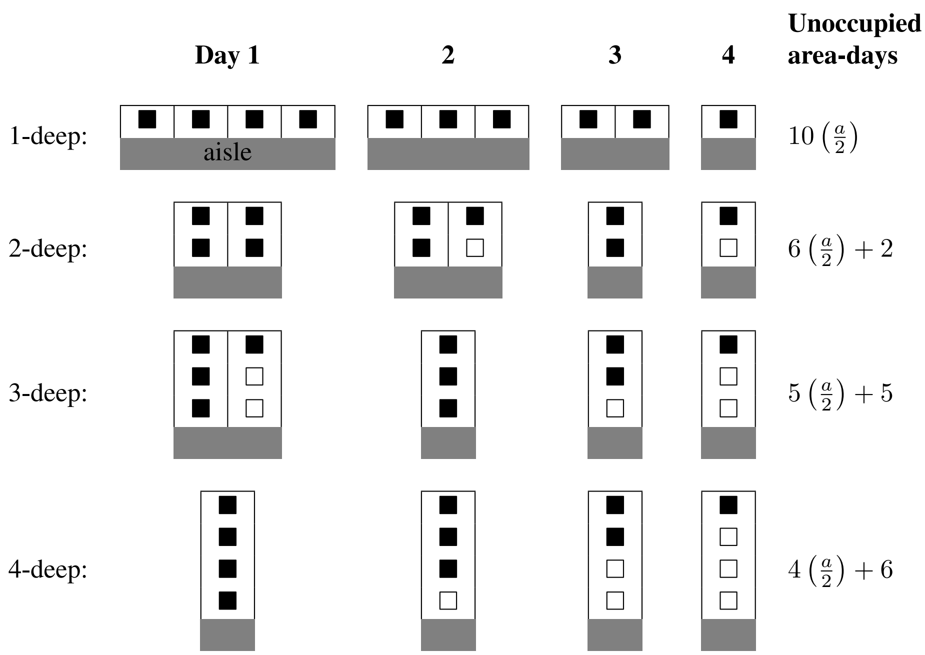


Table 6.1: Four pallet positions of a sku with constant demand are arranged in various lane depths. The area that is unoccupied but unusable by other skus is waste. Area is measured in pallet positions; and a is the width of the aisle, measured as a fraction of the depth of a pallet position.

the time, and so on (Figure 6.4). This waste is called *honeycombing*. Deeper lanes are more susceptible to honeycombing; but shallow lanes use more space for accessibility.

Table 6.1 displays daily snapshots of a sku with an order size that occupies four pallet positions and which is extracted at a rate of one pallet-position per day. In single-deep rack, all the unused space is the aisle space required for accessibility. Deeper aisles require less aisle space per pallet position — but the additional pallet positions are less fully occupied.

To maximize space efficiency sku i should be stored in a lane of depth that minimizes floor space-time that is unoccupied but unavailable to other skus. The optimum lane depth can be determined by simply evaluating all possibilities, as has been done in Table 6.1. In Figure 6.5 waste has been plotted for each lane depth, and it can be seen that 2-deep lanes are most space efficient when the aisle is less than 4 pallet positions wide; otherwise 4-deep is best.

If an approximate answer is acceptable then there is an easier way to compute a lane depth that is space efficient. Suppose that sku i experiences constant demand of D_i pallets annually and that the order quantity is q_i pallets, so that a pallet departs every $1/D_i$ years and the order cycle is of duration q_i/D_i years.

Assume the pallets of sku i are stackable in columns z_i high so that a full allotment of sku i will require $\lceil q_i/z_i \rceil$ floor positions. Another column of pallets will be gone

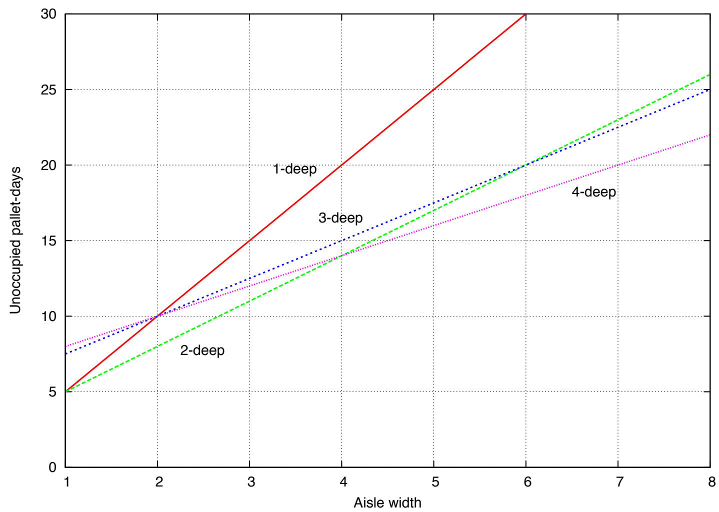


Figure 6.5: Waste, measured as pallet position-days that are unoccupied but unavailable, depends on both the lane depth and the aisle width. The four pallets of this sku should be stored either 2-deep or else 4-deep, depending on the width of the aisle.

and a floor position will become unoccupied after every z_i/D_i time periods.

If lanes are k positions deep then sku i will require $\lceil q_i/(z_i k)\rceil$ lanes. Each lane will have an additional pallet position on the floor that is unoccupied but unavailable after time intervals of $1z_i/D_i$, $2z_i/D_i$, $3z_i/D_i$, $\ldots$, $(k-1)\,z_i/D_i$, for a waste of $z_i k\,(k-1)\,/\,(2D_i)$ pallet position-years per lane. If we approximate the number of lanes by $q_i/(z_i k)$ and multiply this times the waste per lane, we get an approximation of the total waste due to (horizontal) honeycombing during one inventory cycle:

$$\left(\frac{k-1}{2}\right)\left(\frac{q_i}{D_i}\right) \text{ pallet position-years.}$$

The other type of unoccupied space is that devoted to accessibility and we may charge each lane $a/2$ pallet positions per time period for this. The first lane will be depleted and available for reassignment after kz_i time periods, the second after $2kz_i$ time periods, and so on. If there are a total of $q_i/(z_i k)$ lanes, then there are

$$\left(\frac{kz_i\,(q_i/(z_i k))\,(q_i/(z_i k)+1)}{2}\right)\left(\frac{1}{D_i}\right)$$

lane-years, each to be charged $a/2$ pallet positions per lane, for a total accessibility cost per cycle of

$$\left(\frac{a}{2}\right)\left(\frac{q_i/(z_i k)+1}{2}\right)\left(\frac{q_i}{D_i}\right) \text{ pallet position-years.}$$

The total space lost from storage during this inventory cycle is the sum of the honeycombing and accessibility costs; and the average cost per unit time is the total cost per cycle divided by the duration of an inventory cycle, which is q_i/D_i:

$$\left(\frac{k-1}{2}\right)+\left(\frac{a}{2}\right)\frac{\left(q_i/\left(z_i k\right)+1\right)}{2}. \tag{6.1}$$

Theorem 6.1 (Optimal Lane-Depth)**.** *The most space-efficient lane depth for sku i with q_i pallets stackable z_i high is*

$$\sqrt{\left(\frac{a}{2}\right)\left(\frac{q_i}{z_i}\right)}. \tag{6.2}$$

Proof. Differentiating the total cost (Expression 6.1) with respect to k, setting equal to zero, and solving for k gives the result. □

This theorem gives an *ideal* lane depth for sku i. It is an ideal because it ignores the space constraints imposed by the physical layout of the warehouse, so it should be taken as advisory. Also, this is an approximate answer because we avoided combinatorial complications by treating the number of lanes as $q_i/\left(z_i k\right)$, which is possibly fractional.

It would be impractical to allow each sku in a warehouse to choose its own lane depth, because the resulting assortment of lanes might not fit well together within the warehouse. Instead, one typically forces a community of skus to choose jointly a single lane depth that all will share. Let there be n skus; then:

Theorem 6.2 (Floor storage)**.** *To minimize the average floor space consumed per pallet, floor storage should be configured with lane depth of approximately*

$$\sqrt{\left(\frac{a}{2}\right)\left(\frac{1}{n}\right)\left(\sum_{i=1}^{n}\frac{q_i}{z_i}\right)}. \tag{6.3}$$

In other words, the most space efficient lane depth is of the form:

$$\sqrt{\text{(accessibility cost per lane) (Average \#-floor positions per sku)}}. \tag{6.4}$$

Proof. If all skus adopt the same lane depth k, the total average cost per unit time, from summing Expressions 6.1, is

$$\sum_{i=1}^{n}\left[\left(\frac{1}{2z_i}\right)(k-1)+\left(\frac{a}{2}\right)\left(\frac{1}{z_i}\right)\frac{\left(q_i/\left(z_i k\right)+1\right)}{2}\right].$$

Taking the derivative with respect to k, setting it to zero, and solving for k yields the result. □

Example 6.1. *Consider the following population of skus:*

Sku	q_i	z_i
A	*50*	*3*
B	*40*	*4*
C	*36*	*2*

What is the optimal lane depth in floor storage if aisles are 15 feet across (about 4.6 meters) and the pallets are 48 inches deep and 42 inches wide (1.22 meters by 1.07 meters)?

First, convert the unit of measure for aisle width to "pallet depths": 15 feet represents $15/(48/12) = 3.75$ *pallet positions. Substituting into Expression 6.3 gives*

$$\sqrt{\left(\frac{3.75}{2}\right)\left(\frac{1}{3}\right)\left(\frac{50}{3}+\frac{40}{4}+\frac{36}{2}\right)} \approx 5.28 \tag{6.5}$$

so lanes should be about five pallet positions deep.

Let us pause to assess the model. First note that it is based on the assumption that each sku moves at a relatively constant rate. If this does not hold, such as for highly seasonal skus, then we must estimate, based on forecast movement velocities, a weighted average of the wasted space within a lane.

Another important observation is that this model applies however the pallets leave, whether intact or picked carton-by-carton.

It should also be noted that the approximation that enables the simple, continuous solution of Theorem 6.2 will be most accurate when there are relatively large numbers of pallets of each sku. On the other hand, when there are few pallets of each sku then lane depth is not an issue.

It is also worth remarking that this model is designed to reduce space that is unoccupied but unavailable for storage. It is a slightly different problem to determine how many pallet positions the warehouse should have. To answer this requires more detailed information, such as time series of expected quantities of each sku. The answer depends heavily on correlations amongst the inventory levels and the aversion to risk of under- or over-filling the warehouse.

Finally, this analysis says nothing about how the aisles should be arranged within the warehouse. In effect it assumes that the warehouse is large so that edge effects are negligible.

We can perform similar analysis for rack storage. Consider, for example, pallet flow rack configured to a common height of z pallet openings. If each opening holds at most one pallet of any sku, then we find that again the optimum lane depth follows the form of Expression 6.4:

Theorem 6.3 (Pallet flow rack)**.** *To minimize the average floor space consumed per pallet, pallet flow rack should be configured with lane depth of approximately*

$$\sqrt{a\left(\frac{1}{n}\right)\left(\sum_{i=1}^{n}\frac{q_i}{z}\right)}. \tag{6.6}$$

Proof. Left as an exercise: See Question 6.14. □

6.2 Labor

When a pallet arrives at the receiving dock, it is driven by a forklift driver to a storage location, where it resides until requested by the customer. Then a forklift driver moves it to a trailer on the shipping dock.

The warehouse pays its fork lift drivers for person-hours but it bills its customers for two handles for each pallet (in/out); therefore the warehouse wants many handles/person-hour.

Because a forklift handles one pallet at a time, the variable labor cost it incurs can be estimated fairly accurately as the time it takes to drive a forklift from receiving to the storage location to shipping. (Differences in insertion/extraction times are generally small in comparison to differences in travel costs; therefore we treat insertion/extraction times as fixed costs, which may be ignored in deciding where to place skus.)

6.2.1 Reducing labor by dual-cycle operations

The movement of forklifts or other unit-load equipment does not add value if the forklift is traveling with empty forks (*dead-heading*). For example, a forklift will deadhead to a storage location to get a pallet, but then travel productively to carry that pallet to shipping.

Movement in a unit-load warehouse is typically either to stow or retrieve pallets. In the simplest and most common protocol the forklift is devoted to unloading a trailer and stowing the pallets one-by-one or else retrieving pallets and loading them into a trailer. This protocol is known as *single cycle*. In a unit-load warehouse where all pallet movements are either stows or retrievals, a forklift operating under single-cycle protocol dead-heads at least half the time, because it must constantly return empty to receiving to unload another pallet or else travel empty to retrieve another pallet for shipping, as in Figure 6.6.

One way to reduce labor is to store product in convenient locations so that travel with the pallet is reduced — this is the subject of the next section. Another way is to reduce deadheading by careful interleaving of put-aways and retrievals, so that after a put-away, the forklift travels directly to pick up another pallet.

But how can the forklift driver know what retrievals are nearby? The identification of nearby tasks relies on some global view of the warehouse, such as that provided by a supervisor or the warehouse management system. In either case, the "controller" maintains a queue of waiting tasks, from which he or it assigns trips that are pairs of stows and retrievals. The pairs of stows and retrievals will be chosen to reduce deadheading as in Figure 6.7.

A human controller can build efficient trips in an *ad hoc* manner, but to be most effective, dual cycle requires additional IT support to coördinate tasks. With more IT support it is possible to construct the pairings that minimize dead-heading. Assume there is a task list consisting of stows $i = 1, \ldots, m$ and retrievals $j = 1, \ldots, n$. Let the shortest distance between the location of stow i and the location of retrieval j be d_{ij}. (The shortest distances from each stowage location to all the retrieval locations must be computed, for example, by repeated use of the Shortest Path Algorithm described in Appendix C. While this is tedious to contemplate, it is trivial for a computational device that knows the layout of the warehouse and the addresses of the storage locations.) Then the problem of finding the pairings of stows and retrievals to minimize total dead-

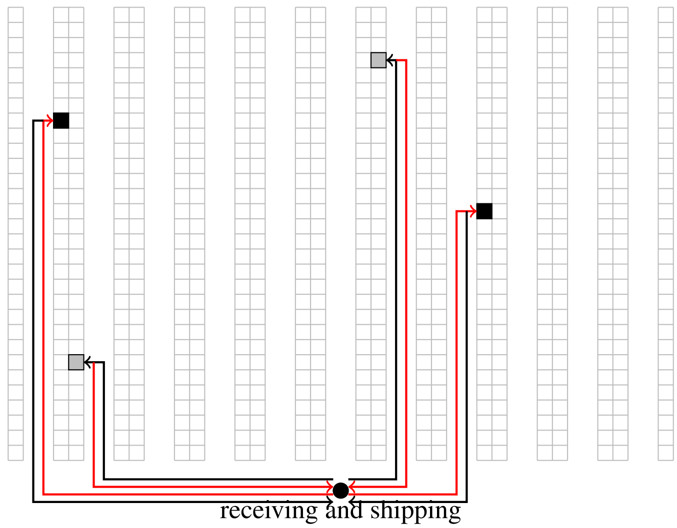


Figure 6.6: Under single-cycle protocol, half of all forklift travel is unproductive (red lines). In this example, both receiving and shipping are at the bottom (black disk). Trips to stow a pallet in an available location (gray squares) require that the forklift return with empty forks. Similarly, any trip to retrieve a pallet from a location (black squares) requires that the journey begin with empty forks.

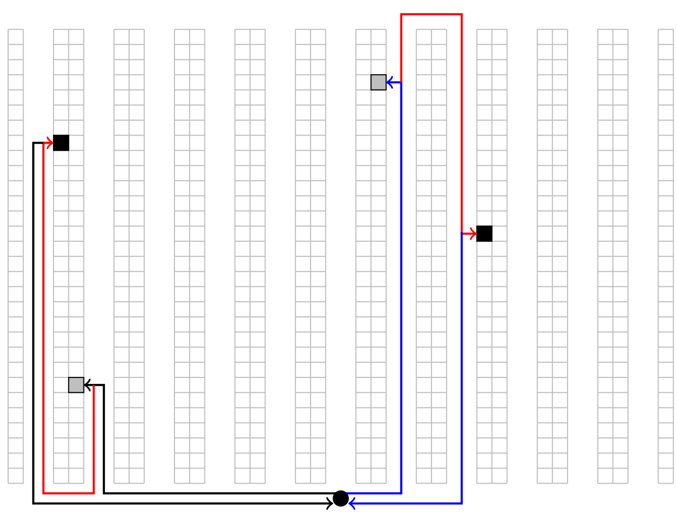


Figure 6.7: Dual-cycles can reduce dead-heading by enabling travel directly from a stow to a retrieval. In this example, each of two stows (gray locations) have been paired with a retrieval (black locations). Dead-heading is shown in red.

heading may be expressed mathematically as

$$\min \sum_{i,j} d_{ij} x_{ij}$$
$$\sum_i x_{ij} = 1$$
$$\sum_j x_{ij} = 1$$
$$x_{ij} \in \{0, 1\}$$

where $x_{ij} = 1$ indicates that the forklift making stow i should then proceed in the most direct way possible to retrieval j.

The first constraint requires that every stow be paired with some retrieval, and the second constraint that every retrieval be paired with some stow. If there are fewer stows than retrievals, then we simply add enough "dummy" stows to make them equal, where each dummy stow represents travel from the shipping dock to a retrieval location. Similarly, if there are more stows than retrievals, we add dummy retrievals, each of which represents dead-heading to the receiving dock.

This is a special type of linear program called an *Assignment Problem* and can be solved quickly (for example, via the linear programming solver found in most spreadsheet programs).

To be most effective, dual-cycle operations should account for dynamically arriving tasks (stows or retrievals): When a forklift becomes available, re-solve the assignment problem to build a set of trips, and assign the trip with least deadheading to the waiting forklift. (We postpone any trip with lots of deadheading in hopes that a later-arriving task might be a better match and so further reduce deadheading.)

For a more general discussion of dynamic pickup and delivery problems, see [14].

If the warehouse requires more general moves than stows or retrievals, such as during a reorganization, then the problem becomes harder, a version of the Rural Postman Problem, which is to find the shortest route that visits each of a set of directed edges [20, 23].

When is dual cycle worth it?

Whether this is worth the effort depends on whether there is contemporaneous receiving and shipping. There might not be much opportunity to take advantage of dual cycles if, for example, product was shipped in the morning, and received in the afternoon.

The layout of the warehouse may make it harder to reduce dead-heading. In the layout of Figure 6.8, receiving, at the bottom, is separated from shipping, at the top, and so any dual cycle incurs unavoidable deadheading between them. In such a case, the dual-cycle protocol may yield no savings.

So: It is certain that the forklift must travel from receiving to a storage location and, later, from that location to shipping. There may be additional travel in the form of dead-heading, but this may be reduced if there are many opportunities for dual-cycle trips.

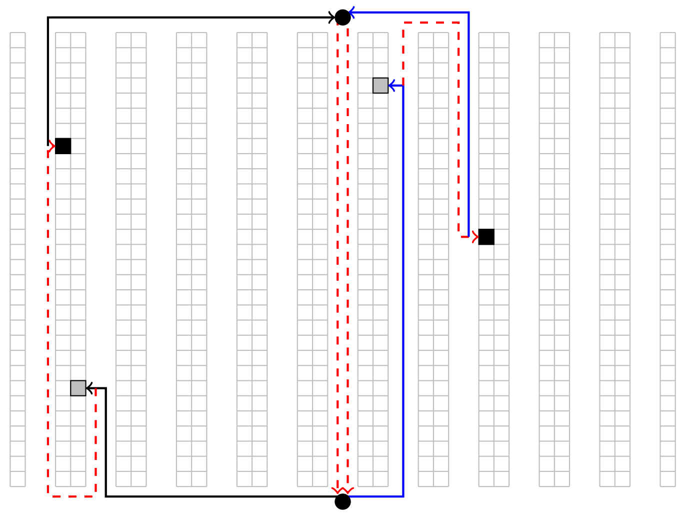


Figure 6.8: When receiving and shipping are separated, a dual-cycle trip must dead-head between them (dashed red line in the center).

6.2.2 Reducing labor by careful product placement

There is unanimous agreement in industry that the “fastest-moving” skus should be stored in the most convenient locations. Unfortunately, both engineering literature and practice disagree on what these terms mean. To be precise each must be based on an explicitly model. First: What is a “convenient” location?

Convenient storage locations

Consider the warehouse of Figure 6.9. Assume that the receiving doors are distributed across the lower edge of the warehouse and the shipping doors are along the upper edge. In this simplest model, a trailer may be parked at any door and so we assume the average location of receiving or shipping is the center location on the dock.

Now each time a pallet is stored at a particular location, the following variable labor costs will be incurred: Travel from receiving dock to location; and, later, travel from location to shipping dock. Therefore with each location i is associated a labor cost incurred by its use. (This labor cost is proportional to the distance d_i from receiving to location to shipping and so we may, without loss of generality, discuss convenience in terms of distance.) In a unit-load warehouse this cost is independent of what is stored in other locations and so, if location i is visited n_i times during the year, the annual labor cost will be proportional to

$$\sum_i d_i n_i. \tag{6.7}$$

Distances d_i are determined by the layout of the warehouse and are not easily changed. But frequencies of visit n_i are determined by customer orders and by our choices of what to store where. By storing the pallets carefully, we can ensure that the

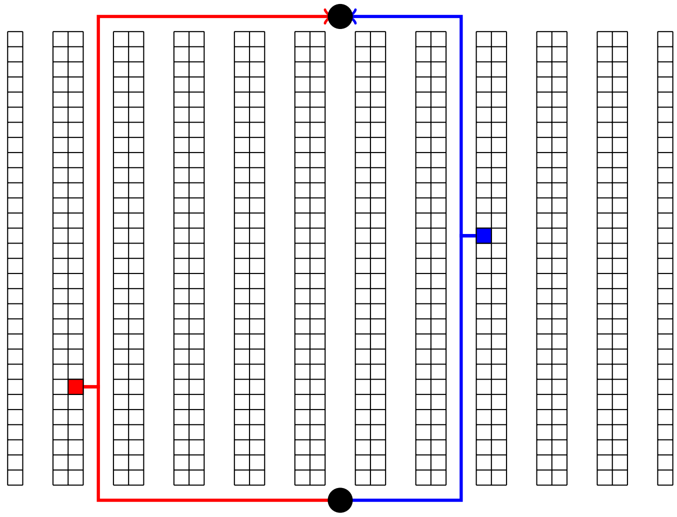


Figure 6.9: The blue location (right) is more convenient than the red location (left) because the total distance from receiving to the location, and from there to shipping is smaller.

most frequently visited locations are those of greatest convenience (smallest total travel d_i), thereby minimizing Expression 6.7.

Fast-moving skus versus fast-moving pallets

From the expression 6.7 for the labor cost of a location we prefer that skus with a lot of movement per storage location be stored in the best locations. In other words, we want to identify those skus that generate the most frequent visits per storage location.

In steady state, during a fixed interval of time

$$\text{average #-visits per storage location} = \frac{\text{number of units shipped}}{\text{number of units in storage}}$$

Thus, for example, a sku that is received in order quantity 5 pallets and that sold 20 pallets last year would have generated about $20/5 = 4$ visits per pallet location, which is more than a sku that moved 100 pallets but was received in quantity 50.

So to minimize labor costs:

- Rank all the available pallet positions of the warehouse from least cost c_i to greatest cost.
- Rank all skus from most to least turns.

- Move down the list, assigning the pallets of the next fastest-turning skus to the next best locations available.

This analysis is based on a warehouse that is operating at approximately steady state. What if the system is far from steady state? We can still approximate our intuition that the busiest skus belong in the most convenient locations; only now we must adopt a more detailed view that considers the rate at which individual pallets turn (not just skus). Now we choose that particular pallet that will be leaving soonest to be stored in the best location. So to minimize labor costs:

- Rank all the pallet positions of the warehouse from least cost d_i to greatest cost.
- Rank all pallets from soonest departure to latest departure.
- Move down the list, assigning the next pallet to the next location.

6.2.3 Location of receiving and shipping

The layout of the warehouse determines the cost associated with each storage location. In the layout of Figure 6.10 the average positions of receiving and shipping are in the middle of opposite sides of the warehouse. This is sometimes referred to as a *flow-through* configuration because all product flows from one side of the facility to the other. As a result all the storage locations along one side of an aisle are equally convenient.

Now imagine how the convenience of the storage locations changes if the shipping and receiving doors were both moved to the right. Then storage locations to the left would become less convenient and the locations on the right more convenient; but the quality of the very best locations would *not* improve, while the quality of the very worst locations would becomes strictly worse. The result is that the layout would be absolutely less efficient.

If both receiving and shipping on the *same* side of the warehouse this induces a different economic terrain within, as shown in Figure 6.11. Because product flows in and out the same side of the warehouse, this is sometimes referred to as a *U-flow* configuration.

Now the best storage locations, which are at the middle of the common receiving and shipping dock, are very convenient indeed because a location that is close to receiving would also be close to shipping. However, there are relatively few such prime locations and there are more inconvenient locations than before; and the least convenient locations, at the far top corners are even less convenient than the least convenient of Figure 6.10.

Which layout is better? In this case, as with so many of the design decisions, it depends on the populations of skus passing through the warehouse. If there are few very fast-moving skus, as is typical in cosmetics, apparel, or other fashion businesses, it may be more efficient to put receiving and shipping on the same side of the facility, because the savings from the few very convenient locations may offset any loss due to the greater number of less convenient locations.

Here are some characteristics of each type of layout:

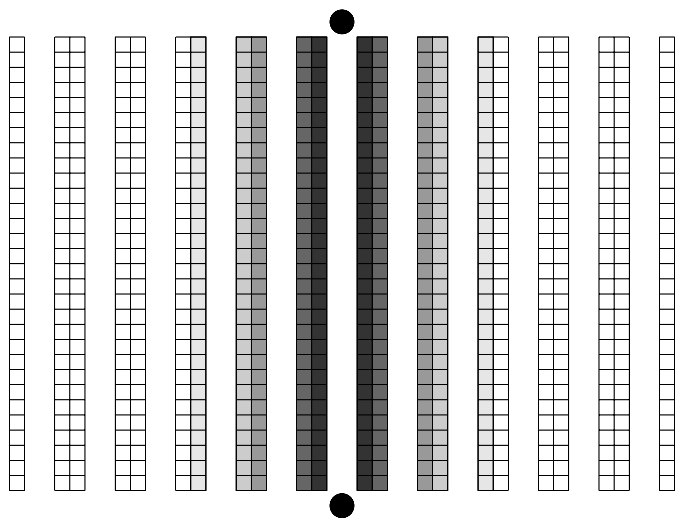


Figure 6.10: When receiving and shipping are located at opposite sides of the warehouse there are many locations of equal convenience, and with the most convenient on a line between shipping and receiving. (The darker locations are the more convenient.)

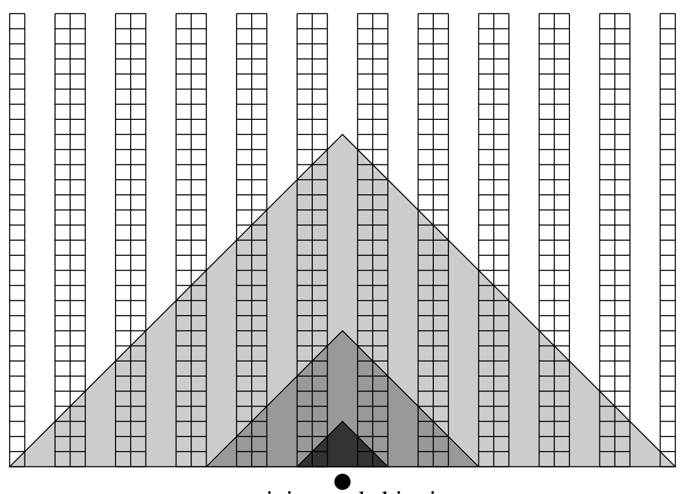


Figure 6.11: When receiving and shipping share the same dock (at the bottom in this example) then there are a very few, very convenient locations as well as some very inconvenient locations. (Darker shading indicates more convenient locations.)

- U-flow configuration
  - Receiving and shipping located on same side of the warehouse
  - Makes the most convenient locations still more convenient, less convenient locations even worse.
  - Appropriate when product movement has strong ABC skew (that is, when very few skus account for most of the activity)
  - Provides dock flexibility for both shipping, receiving: If one experiences a surge of activity, can make use of additional doors from other function.
  - Permits more efficient use of fork lifts: When a forklift reports for an assignment, he may be given a put away and a retrieval, matched to reduce dead-heading.
  - Minimizes truck apron and roadway
  - Allows expansion along other three sides of warehouse.
- Flow-through configuration
  - Receiving and shipping on opposite sides of the warehouse
  - All product flows in the same direction so there is less opportunity for interference.
  - Makes many storage locations of equal convenience.
  - Conservative design: More reasonably convenient storage locations but fewer that are very convenient.
  - More appropriate for extremely high volume.
  - Preferable when building is long and narrow
  - Reduces any efficiencies that might be gained from dual cycle transactions

6.2.4 Aisle configuration

Cross-aisles

To reduce travel between storage and receiving/shipping, it is generally preferable to orient aisles so that they run parallel with the direction of material flow. However, it is sometimes advantageous to support movement between storage locations, such as if a forklift, operating under dual-cycle, travels directly from putting away one pallet to retrieving another as in Figure 6.12.

There is also a cost to having a cross-aisle in that more floorspace is required for the same number of pallet locations, and so additional travel is introduced. If receiving and shipping are located on opposite sides of the cross-aisle, then every location is made slightly less convenient because the cross-aisle each pallet must cross the aisle once. And if receiving and shipping are on the same side of the cross-aisle, the near locations are unaffected, but the far locations are made even less convenient, because each pallet stored there must cross the aisle twice: once to store and again to retrieve the pallet.

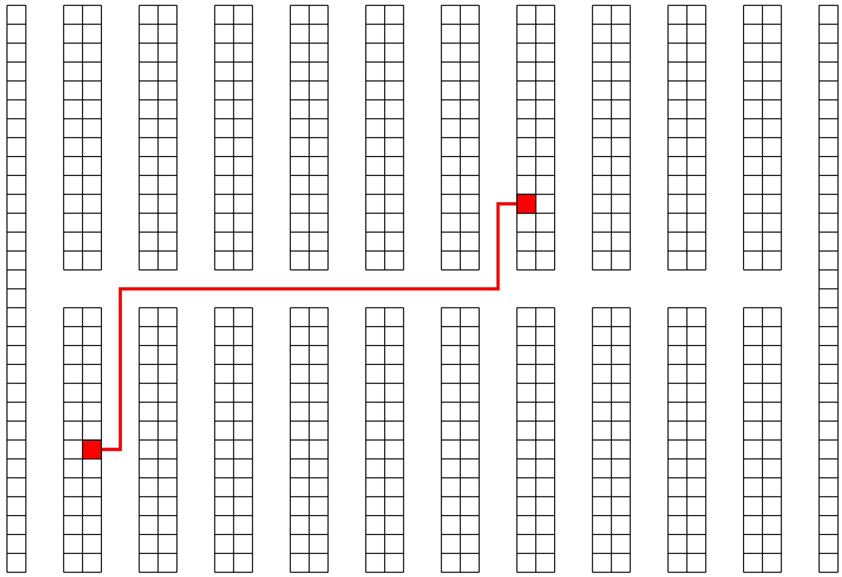


Figure 6.12: A cross-aisle allows more direct and therefore shorter travel between storage locations.

Angled aisles

Most warehouses have parallel aisles aligned with the receiving and shipping docks, perhaps with orthogonal cross-aisles; but this need not always be the case. Kevin Gue of Auburn University and Russ Meller of the University of Arkansas [26] have argued that travel times can be reduced by up to 20% by reorienting some aisles and including some angled cross-aisles, as in Figure 6.13, which they call a *fishbone* layout. The overall warehouse must be slightly larger to compensate for the space lost to the additional aisles; but this is more than made up for by the efficiency of more direct travel to or from a centralized point of receiving and shipping.

It is possible to take advantage of this more direct travel if most pallet movement is to or from the central dispatch point. But if a forklift finishes putting away a pallet and then must retrieve another, the orientation of the aisles of the fishbone arrangement may not help at all, and indeed may be an impediment. Nevertheless, this possibly inefficiency seems to be more than made up for by the direct travel to and from the central dispatch point.

6.3 Summary

- Location of shipping and receiving and layout/orientation of storage determines which pallet positions are convenient and which are not.
- Arrange your warehouse so that the convenience of storage positions matches the

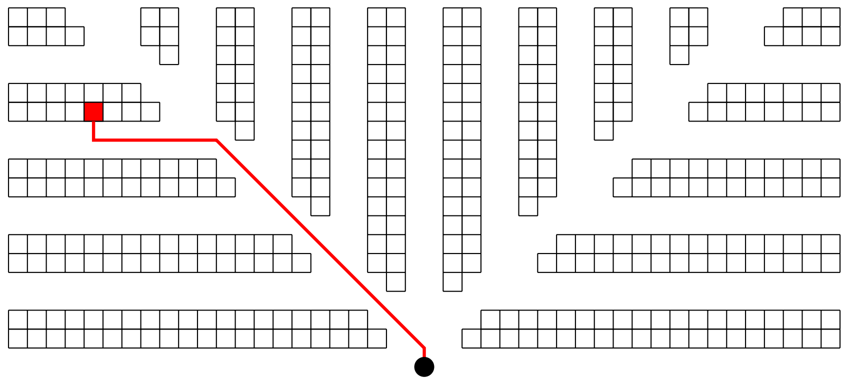


Figure 6.13: Angled aisles, suggested by Gue and Meller, allow more direct travel between storage and a central location of receiving/shipping (at the bottom).

velocity of the products. For example, when activity is concentrated within a few skus, it is better to put receiving and storage near each other, which concentrates convenience in a few storage positions.

- Once the layout is determined, store the fastest-moving pallets in the most convenient positions.
- Deep-lane storage reduces aisle space but loses use of pallet positions due to honey-combing, the creation of empty but unusable storage positions. (The positions cannot be used until the remaining product in the same column and/or aisle are removed.) The optimal lane depth balances these two losses to get the greatest (time-averaged) occupancy of floor space. Optimal lane depth are given by Theorems 6.2 and 6.3.
- It is possible to use detailed histories or forecasts of product movement to exactly optimize equipment layout and configuration. Consider doing this when space is very expensive.

6.4 More

In laying out pallet storage there are other issues to consider besides space utilization. For example, in some warehouses, such as those of food distributors, it is important to move product in accordance with the rule of "First In First Out" (FIFO). Some special types of rack such as pallet flow rack support FIFO; but otherwise FIFO can be guaranteed only to within the lane depth. Making lanes deeper might give better space utilization; but it reduces compliance with FIFO. In addition, deeper lanes can increase insert/extract times.

Computer distributors handle a challenging mix: Their pallets tend to be either high and light, such as a pallet of printers, which can be 7 feet high (2.1 meters); or low and heavy, such as a pallet of software or printed documentation.

The management of pallets is a constant problem: They flow downstream in the supply chain and so must be either replaced or recirculated; they become damaged; they eventually have to be replaced. Some companies address these problems by maintaining a communal pool of high-quality pallets that they recirculate among their clients, each of whom pays rent for the pallets they use.

The ability to floorstack can be seasonal. Corrugated cardboard, the paper product of which most cartons are constructed, tends to absorb moisture. This means that cartons can weaken under prolonged exposure to high humidity. For example, in Atlanta, Georgia, USA, many skus that can be floorstacked during the winters, which are relatively dry, cannot by floorstacked during the summers, which are hot and humid.

6.5 Questions

Question 6.1. *What is a unit-load warehouse and why is it easier to layout than others?*

Question 6.2. *For each of the following, explain whether the description makes the sku more or less appropriate to be stored in pallet rack rather than stacked on the floor, and why.*

1. *A pallet of this sku is fragile.*
2. *A pallet of this sku is heavy.*
3. *Pallets of this sku can be stacked safely almost to the ceiling.*
4. *A pallet of this sku is strong and square.*
5. *There are never more than two pallets of this sku in the warehouse.*
6. *Each case of this sku is dense and so, to keep the pallet from being too heavy, it is loaded only one meter high.*

Question 6.3. *Why is it generally better to centrally locate receiving and shipping doors.*

Question 6.4. *When is a cross aisle valuable? When are angled aisles valuable?*

Question 6.5. *Explain why it is almost always preferable to orient aisles parallel to a line connecting receiving to shipping.*

Question 6.6. *Compare the efficiencies of aisle orientations in a unit-load warehouse shown in Figure 6.14. Assume in each case that receiving is located at the bottom of the figure and shipping at the top. How would your answer change if both receiving and shipping were located at the bottom?*

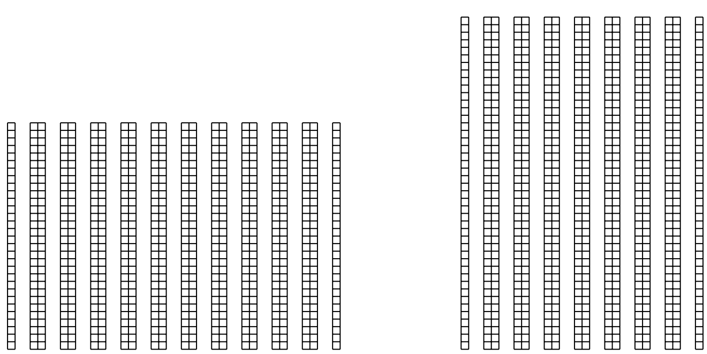


Figure 6.14: Question 6.6: Which is the better layout?

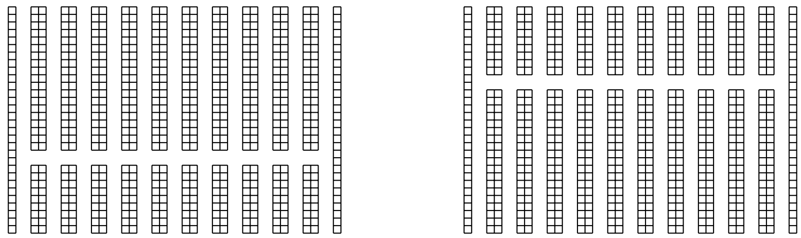


Figure 6.15: Question 6.7: Should a cross-aisle be located close to receiving/shipping or far?

Question 6.7 (K. R. Gue)**.** *Where is the best place for a single cross-aisle in a unit-load warehouse such as shown in Figure 6.15? Assume that all operations are dual-cycle, all locations are equally likely to be used, and that both receiving and shipping are at the bottom of the figure. What if receiving and shipping are on opposite sides? How would your answers change if the challenge were to locate several cross-aisles?*

Question 6.8. *Figure 6.16 shows the layout of a distributor of spare parts. The left side of the warehouse is pallet rack and cantilever rack. Receiving is on the bottom left and shipping is on the bottom center of the figure. Critique this layout.*

Question 6.9. *Under what circumstances are deep lanes appropriate? What advantages and disadvantages accrue for deeper lanes?*

Question 6.10. *Suppose you have 100 pallets of SKU A and 100 pallets of SKU B. Pallets of SKU A may be stacked 4 high while, because of fragility, pallets of SKU B may be stacked only two high. Which one should have deeper lanes if they are floor-stacked? How much deeper?*

Question 6.11. *Suppose you are laying out a floor stack area with an aisle 14 feet (4.3 meters) wide to accommodate forklift trucks. All pallets are 48 by 40 inches (1200 by 1000 mm) and are stored with the 40-inch side facing the aisle. Assume that each sku experiences demand that is constant and is reordered once each inventory cycle as below.*

SKU	*Order quantity (pallets)*	*Stack height (pallets)*	*Order cycle (weeks)*
A	*20*	*2*	*4*
B	*24*	*3*	*3*
C	*12*	*1*	*6*
D	*4*	*3*	*2*

A. For each sku, compute a space-efficient lane depth.

B. Compute a single, common lane depth that will be space-efficient for the set of skus.

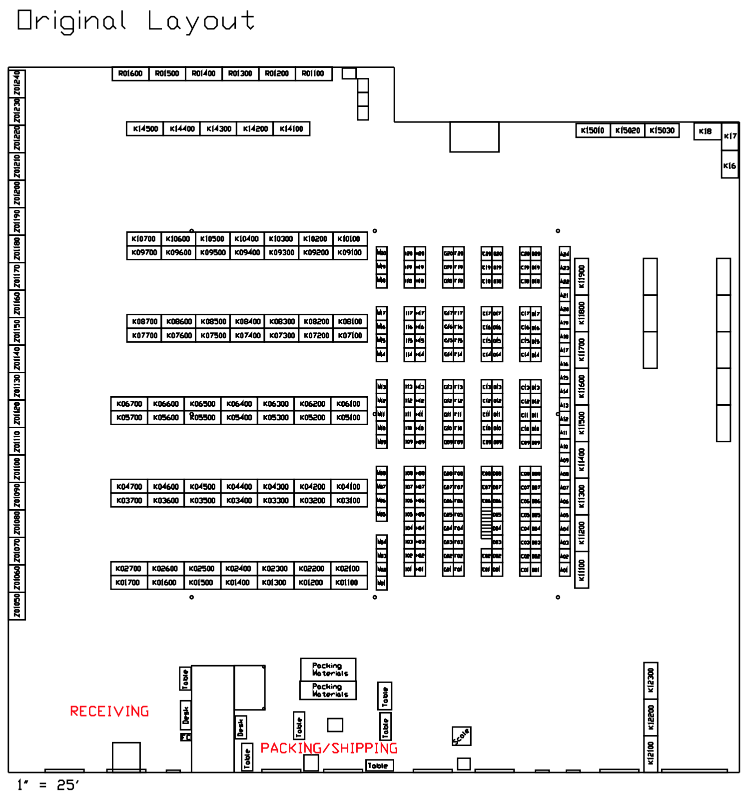


Figure 6.16: Question 6.8: Critique this layout of a service parts distribution center.

Question 6.12. *Suppose you are laying out a floor stack area that will be a single pallet position deep. The aisle will be 2 pallets wide. Assuming that demands are constant, which skus will give the best space utilization in this area?*

Question 6.13. *Consider a collection of pallets of paint of various kinds. Each sku will be floor-stacked 4 pallets high. Which skus are best stored in the area of the warehouse with lane depth of 3 pallet positions and an aisle 3-pallets wide?*

Question 6.14. *Prove Theorem 6.3, which establishes the optimal lane depth for pallet flow rack. Hint: You must also charge for aisle space* behind *the flow rack, for restocking.*

Question 6.15. *True or false: Faster-moving product should be stored in deeper lanes. Explain your answer.*

Question 6.16. *Why cannot high space utilization be guaranteed simply by storing product in sufficiently deep lanes?*

Question 6.17. *What is "honeycombing" and why is it a problem for warehouses? What techniques or tools can be used during the phase of warehouse design to reduce horizontal honeycombing? Vertical honeycombing?*

Question 6.18. *For a given set of skus and fixed aisle width, which type of storage would likely have the greater lane depth when optimized for space efficiency: floor storage or pallet flow rack? Why?*

Question 6.19. *Suppose that you have planned a layout for floor storage with one global, optimal lane depth for your skus. Subsequently you learn that inventory levels will be half of what you had expected; how does this change the optimal lane depth?*

Question 6.20. *Consider a sku with pallets so short that a stack of two can be stored within a single pallet opening in rack. How would you estimate the average number of visits per storage location generated by such a sku? How can you continue to use the formula for optimal lane depth for such a sku?*

Question 6.21. *Why do you suppose single-deep rack also called "selective rack"?*

Question 6.22. *Selective (single-deep) rack can never provide utilization of floorspace in a real warehouse exceeding which value: 25%? 33%? 50%? 66%? 75%? 100%? Explain why your answer is the best choice.*

Question 6.23. *Which method of storage makes empty pallet positions more quickly available, thereby increasing storage capacity: floor storage or rack storage? Explain.*

Question 6.24. *Explain why a warehouse with only one or two pallets of each of many skus is likely to prefer rack to floor storage.*

Question 6.25. *Consider a unit-load warehouse that is rectangular, of dimension $m \times n$ (with $m >> n$). If it is to be laid out with receiving and shipping on opposite sides, is it more efficient to run aisles parallel to the long or the short side of the warehouse? Why? What if receiving and shipping were on the* same *side?*

Question 6.26. *In Section 6.2.2 we estimated the "convenience" of a storage location by the forklift travel from receiving to the location plus that from the location to shipping. We did not explicitly account the unloaded travel ("dead-heading") that the forklift must make after storing a pallet in the location or delivering the pallet to shipping. How might this additional travel affect our analysis? Assume that we load and ship full trailers in the mornings and that we receive and unload full trailers in the afternoons.*

Question 6.27. *Suppose that we want to store all the skus of each customer together in a pallet-in, pallet-out warehouse. The skus of which customer are candidates for the most convenient locations? Explain.*

- *That customer who has the most pallets in the warehouse*
- *That customer whose pallets we receive and ship in the greatest quantities*
- *That customer whose product represents the greatest dollar-volume when we account for the value of the product*
- *That customer whose shipments into the warehouse are the largest*

- *That customer whose shipments out of the warehouse are the largest*
- *That customer whose skus achieve the greatest annual turns*
- *It is impossible to tell*
- *None of the above*

Question 6.28. *Consider two skus moving through a unit-load warehouse. Every Monday morning 100 pallets of sku A are shipped out and in the afternoon 100 new pallets of sku A arrive. Every* other *Monday, 10 pallets of sku B are shipped out and in the afternoon 10 pallets arrive. On average there are 500 pallets of sku A on-hand and 20 pallets of sku B.*

- *Which sku has priority for storage in the most convenient storage locations under FIFO?*
- *How would your answer change if the FIFO requirement were dropped?*

Question 6.29. *Consider three pallet locations A, B, C, for which the total time to travel from receiving to the location to shipping is 1, 2 and 2 minutes, respectively.*

Through these locations flow two skus, x and y, which you must manage in a strict FIFO manner. Every three days one pallet of sku x is shipped out in the morning and one pallet arrives in the afternoon and is put away. Every two days one pallet of sku y is shipped out in the morning and one pallet arrives and is put away. You maintain a constant inventory of one pallet of sku x and two pallets of sku y on-hand at all times.

A. If you are using a policy of dedicated storage, what is the assignment of skus to storage locations that minimizes average labor? (Justify your answer.) What is the corresponding average minutes per day spent moving these products?

B. What is the smallest sustainable value of average minutes per day spent moving these skus if you adopt a policy of random storage? Does it make any difference where the skus are stored initially?

C. What is the best way of assigning pallets to storage locations if you are permitted to drop the FIFO requirement? Would this save any labor? If so, how much; if not, why not?

D. How would your previous answers change if the pallets of sku y arrive in batches of two every four days? Assume the pattern of shipping is unchanged.

Question 6.30. *Label each of the following as true or false, and explain your answer. For a unit-load transshipment warehouse*

- *Space is better utilized if the pallet expected to depart soonest is stored in the most convenient location.*
- *Labor is reduced if the pallet expected to depart soonest is stored in the most convenient location.*
- *Space is better utilized if the sku expected to depart soonest is stored in the most convenient location.*

- *Labor is reduced if the sku expected to depart soonest is stored in the most convenient location.*
- *Space is better utilized if the pallet expected to depart soonest is stored in the location closest to shipping.*
- *Labor is reduced if the pallet expected to depart soonest is stored in the location closest to shipping..*

Question 6.31 (Exploration)**.** *Consider a unit-load warehouse with regular, patterned input stream of arriving pallets and output stream of departing pallets, as in the previous question. Explore how, under a policy of random storage, the disposition of skus in the warehouse organizes itself.*

Question 6.32 (Harder)**.** *Suppose you are free to choose* two *lane depths for floor storage: One region of the warehouse will be devoted to deep lane storage and one to shallower storage.*

- *Prove the following: When the number of columns of each sku greatly exceeds any candidate lane depth then, if a particular sku belongs in the deeper lane storage, any sku with more columns also belongs in the deeper lane storage.*
- *How can you use this fact to decide where to store the skus?*
- *How can you use this fact to choose the best two values for lane depth?*
- *Generalize to three or more lane depths.*

Question 6.33 (Exploration)**.** *Build a simulation model to study the effect of lane depth on warehouse performance. Measure especially the effect on space utilization, labor, and observance of FIFO.*

Chapter 7

Layout of a carton-pick-from-pallet area

The handling unit *carton* or *case* is not standardized, but generally it refers to a rectangular box that:

- weighs between about 5 and 50 pounds (2.3–22.7 kilograms);
- can be handled by one person;
- is conveyable (which depends on the type of conveyor);
- can be stored on a pallet.

When handled in volume, cartons are typically stored on pallets and so restocking is a unit-load process, but picking is not, which creates additional complexities in our models of space and labor. One consequence is that it becomes much more difficult to measure the convenience of an individual location; but it is still possible to make broad distinctions in convenience. A particularly convenient region that is replenished from within the warehouse is called a *forward area* or *fast-pick area*.

The material flow on which we concentrate here is that of Figure 7.1.

7.1 Layout for a forward area

The distinguishing feature of a forward-pick area is that it is especially convenient from which to pick, but, because space is limited, it must be restocked from elsewhere in the warehouse. The most common forward pick area is the ground floor of pallet rack. Popular skus may be picked from ground level and replenished by "dropping" overstock pallets from above, as in Figure reffig:carton-pick-from-bottom-level.

For very high volume distribution of product that is conveyable, cartons may be picked from pallet flow rack to conveyor, as in Figure 7.3. In this case, an order-picker can walk up and down the aisle picking cartons, labeling them with destination, and placing them on the conveyor, which can take them to shipping.

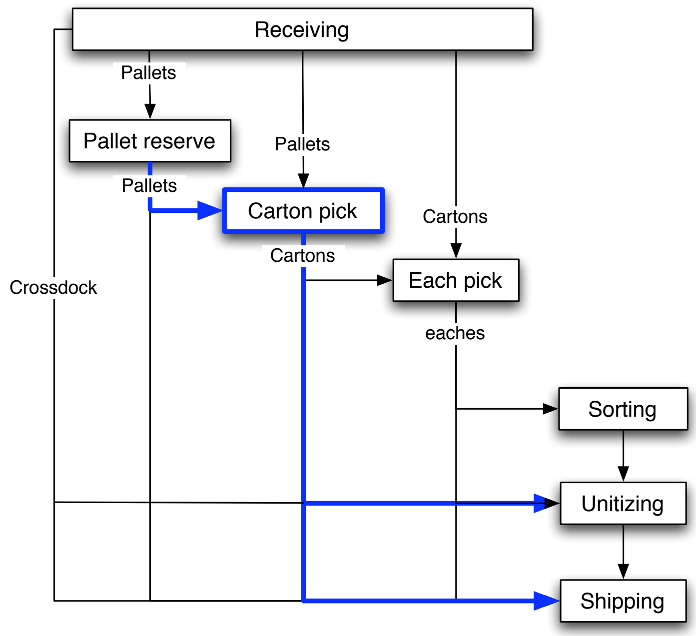


Figure 7.1: Typical flow of cartons through a warehouse

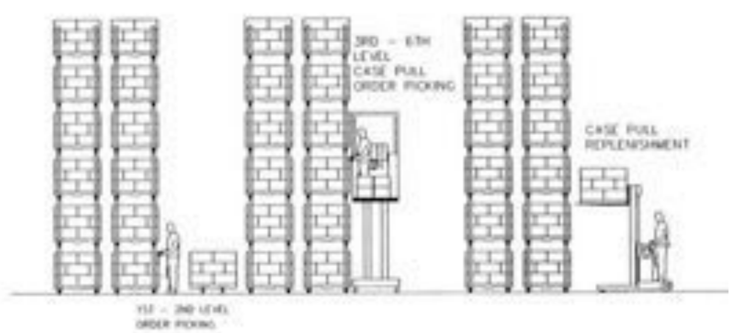


Figure 7.2: Cartons are picked from the bottom, most convenient, level; when the bottom location has been emptied, a restocker refills it by dropping a pallet from above. Newly arriving pallets are inserted into the (high) overstock. (Adapted from “Warehouse Modernization and Layout Planning Guide”, Department of the Navy, Naval Supply Systems Command, NAVSUP Publication 529, March 1985, p 8–17).

Figure 7.3: Cartons are picked from pallet flow rack onto a conveyor. The flow rack is replenished from behind. (Adapted from "Warehouse Modernization and Layout Planning Guide", Department of the Navy, Naval Supply Systems Command, NAVSUP Publication 529, March 1985, p 8–17).

Both of these configurations share the same organization: a small number of locations from which it is convenient to pick from are restocked from a *bulk area* (also known as *reserve* or *overstock*). In the first case, the forward area is the ground floor pallet locations and the reserve includes all higher locations. In the second case, the forward pick area includes all pallet locations in the pallet flow rack and the reserve is may be a separate area of the warehouse, perhaps with very high pallet rack (not shown in the figure).

Because it is the more typical, we shall discuss the first case, picking from ground level, although everything we develop applies equally to picking from pallet flow rack.

7.1.1 Operating protocols

The simplest, most typical operating protocols are expressed in the following assumptions:

Assumption 7.1 (Storage policies)**.** *All locations in the forward area are reserved locations to which specific skus will be assigned, while all overstock locations will be shared storage.*

Reserved (dedicated) storage is employed in the forward pick area to support rapid order-picking. When storage is reserved, it can be assigned based on activity rather than mere availability, and order-pickers can more easily learn where product is. Even though reserved storage is not the most space-efficient, this does not cost much because the forward pick area is typically a relatively small part of the warehouse. The reserve area is generally devoted to bulk storage and is much larger. Using shared storage in the reserve area ensures high space utilization in most of the warehouse.

Assumption 7.2 (Active pick locations)**.** *At any time each sku has a single location from which to pick less-than-pallet quantities; furthermore, that location will remain within a single area, either forward or reserve.*

If a sku has pallets in both the forward and reserve areas, it makes economic sense to pick full-pallet quantities from reserve: If we were to take a full pallet from the forward area, it must be replenished from reserve, and this entails three trips, one to retrieve the pallet from within the forward area, one to retrieve a replacement pallet from reserve, and one to put the replacement pallet in the newly-available location forward. It is clearly more efficient to fill such an order by retrieving a full-pallet from reserve.

When a ground level pallet location is emptied then a high-reach forklift is dispatched to find a pallet of the same sku from above and restock the now empty ground location. Because pallet moves are unit-load moves, the work to restock a sku assigned to the ground floor is proportional to the total number of pallets moved through the fast-pick area. Thus, for a sku that has some pallets on the ground floor and some in reserve, the number of restocks is estimated by the total demand *due to less-than-full-pallet picks*. Full-pallet picks generate no restocks because either they are picked from reserve (which is not restocked internally) or else, if they are picked from a forward location, this must be because all of the sku is forward and again there are no restocks.

7.1.2 Quantities to store forward

Suppose a sku has been assigned to the forward area; how many pallets locations should be devoted to it?

The "minimum practical number of locations", which we shall write as l_i, is determined by several factors, including the need to avoid stockout while awaiting restock. For example, if the time to restock a sku is large or unpredictable, it might be preferable to store two pallets of that sku in the forward-pick area. The extra pallet gives some flexibility to the timing of restock, so that when one pallet has been picked empty, the order-pickers can continue to pull product from the remaining pallet. Sometimes is preferable to defer all restocks until the end of the *pick wave* (scheduled chunk of work), after which the forward area could be restocked without interfering with picking. In this case, each forward sku would be stored in quantity sufficient to satisfy customer requests during the wave. In effect, the extra pallets allow scheduling of restocks rather than simply reacting to need. The cost is forfeiture of some of the value of the forward area, because the extra pallets take up space that could be used by other skus.

Another reason to store more than one pallet of a sku forward is to avoid congestion among the order-pickers, especially if they are picking to pallet jack or truck. By storing popular products in multiple forward locations there is less chance that a picker has to wait for a colleague to move out of the way.

Example 7.1. *Suppose that, to avoid stockout, sku i must be have at least three pallets forward if any and to avoid congestion it must occupy at least two locations. Then if* *the forward pick area is the ground floor of 1-deep pallet rack, holding one pallet of sku i per location, sku i would require at least*

$$l_i = \max\left\{\frac{3\,\textit{pallets}}{1\,\textit{pallet per location}}, 2\,\textit{locations}\right\} = 3\,\textit{locations},$$

which, in this case, would hold three pallets. On the other hand, if the forward pick area is 2-deep pallet flow rack, sku i would require at least

$$l_i = \max\left\{\frac{3\,\textit{pallets}}{2\,\textit{pallets per location}}, 2\,\textit{locations}\right\} = 2\,\textit{locations},$$

which would hold a total of four pallets.

Once a sku has been allocated its minimum required number of pallet locations, there is no benefit to be gained from giving it additional space — unless one has decided to store every pallet of this sku forward, in which case both less-than-pallet and full-pallet picks are faster, and there are no internal replenishments.

It is a bit trickier to estimate the maximum number u_i of forward locations that might be used by a sku.

7.1.3 Choosing skus for the forward-pick area

What skus should be stored in the forward-pick area? The main insight here is that once one has decided to store a product in the forward pick area, giving it additional storage locations, beyond the minimum required, conveys no benefit: It does not increase the number of picks from the forward area, nor does it reduce the number of restocks. (As long as restocking is a unit-load process, the number of restocks is always equal to the number of pallets sold.) There is additional savings only when one puts every pallet of the sku in the forward area so that no restocking is required (that is, no internal moves from bulk storage to the forward pick area). Therefore the only amounts to consider storing are: no pallets, the minimum practical number of locations, or else all the pallets. We formalize this as:

Theorem 7.1 ("Law of None, Min, or All"). *Any sku that is picked from pallets should either not be in the fast-pick area at all; or it should have the minimum practical number of locations; or it should have all of its on-hand pallets in the forward-pick area.*

Let p_i be the number of picks for less-than-pallet quantities, d_i the number of pallets moved by such picks, and D_i the number of pallets moved by full-pallet picks. Let l_i be the minimum number of locations required by sku i in the fast-pick area and and u_i be the maximum number of forward locations. (This value might be guessed from historical records; but unless there is some confidence in the upper bound, it is best taken as $u_i = \infty$.) Suppose that, on average, it saves s minutes when a pick is made from the forward area rather than from bulk-storage; and that each restock of the forward area (that is, each move of a pallet from reserve to the forward area) requires

c_r minutes. Then the net-benefit of allocating x forward locations to sku i is

$$\text{net benefit} = \begin{cases} 0 & \text{if } x = 0; \\ sp_i - c_r d_i & \text{if } l_i < x < u_i; \\ s\left(p_i + D_i\right) & \text{if } x = u_i \end{cases} \tag{7.1}$$

Notice that some skus could positively hurt efficiency if they were stored in the forward pick area in less than their maximum amounts. For such a sku i the net-benefit $sp_i - c_r d_i$ is negative if

$$\left(\frac{p_i}{d_i}\right) s < c_r. \tag{7.2}$$

In other words, for such a sku the average picks per pallet do not generate sufficient savings to pay for restocking that pallet. The average picks per pallet must exceed c_r/s to be worth storing forward. Equivalently, it is uneconomical to store a sku forward if the average size of a pick exceeds fraction s/c_r of a pallet.

On the other hand, for any sku, the net benefit of storing all of its pallets in the forward pick area is always positive because no restocking from bulk storage is required. The difficulty here is knowing how many constitute "all".

Assume for now that "all" is quite large in relation to the size of the forward pick area and so we take $u_i = \infty$. We can write the problem of selecting skus for forward storage by means of choice variable $x_i \in \{0, 1\}$. The object is to minimize total labor costs (picking plus restocking) subject to the space constraint that only N storage locations are available in the forward-pick area. Let c_1 be the average cost per pick from the forward area and let c_2 be the average cost per pick from bulk storage. We may assume $c_1 < c_2$.

$$\min \sum_{i=1}^{n} \left(c_1 p_i + c_r d_i\right) x_i + c_2 p_i \left(1 - x_i\right)$$

$$\sum_{i=1}^{n} l_i x_i \leq N$$

$$x_i \in \{0, 1\}$$

Rearranging terms and changing the sense of the optimization allows us to write the equivalent form (where the pick savings $s = c_2 - c_1$).

$$\max \sum_{i=1}^{n} \left(sp_i - c_r d_i\right) x_i$$

$$\sum_{i=1}^{n} l_i x_i \leq N$$

$$x_i \in \{0, 1\}$$

This is an instance of the *knapsack* problem, about which see Appendix B and references therein. The Greedy Heuristic is an appropriate solution technique in this case and calls for ranking the skus by "bang-for-buck". In this context bang-for-buck is the labor-savings per forward location, which we term *labor efficiency*:

$$\frac{(sp_i - c_r d_i)}{l_i}. \tag{7.3}$$

Example 7.2. *If picks from the forward area average one minute apiece and picks from the bulk storage area average 2 minutes apiece, then the savings per pick for storage in the forward-pick area is $s = 2 - 1 = 1$ minute/pick. Assume that restocks to the forward area average three minutes each. Then the labor efficiencies of the following three skus can be computed as follows:*

Sku	*Picks*	*Demand*	l_i	*Labor Efficiency*
A	*600*	*20*	*3*	$((1)(600) - (3)(20))/3 = 180$
B	*1000*	*100*	*5*	$((1)(1000) - (3)(100))/5 = 140$
C	*200*	*2*	*2*	$((1)(200) - (3)(2))/2 = 97$

Sku A has the strongest claim to storage in the forward area because it is estimated to generate the greatest labor savings per forward location. Note in particular that sku A has a stronger claim than sku B despite the fact that sku B is requested almost twice as frequently and is sold in five times the volume. This is the problem: A typical order for sku B is for 0.1 pallets and so generates more restocks than sku A, which detracts from any pick savings. Moreover, to ensure timely replenishment and to avoid congestion, sku B requires five pallet locations, and so the net labor savings per pallet location is further reduced.

Figure 7.4 shows the distribution of labor efficiency for a collection of skus competing for space in the forward pallet area of a beverage distributor. Only about the top 100 skus would save labor if picked from a forward location. The next 400 may or may not be picked forward, without much affect on labor costs either way.

It would clearly be uneconomical to include any of the bottom 100 skus in the forward area. Interestingly, the latter group can include the most important skus, those that are very popular and that move in great volume. In Example 7.2, imagine an additional sku, call it sku D competing for space against the others of. If sku D is in such demand that an average pick is for half a pallet, then there are only about 2 picks per pallet before restocking. Each forward pallet of sku D would save 2 minutes of picking, but cost 3 minutes to restock, for a net loss of 1 minute. Such a sku should not be stored forward, even if it is very popular.

Such skus are inappropriate for the forward pick area, but their popularity and volume of movement is an invitation to improve their flow through the warehouse. For example, they may be made more attractive to the forward area if their restocking were made more efficient, perhaps by keeping reserve inventory in a convenient dedicated area from which restocks can be made more quickly. Or perhaps they can be picked from some other forward area such as from floor stack near the shipping doors. Or perhaps their entire on-hand inventory should be in the forward pick area or on the loading dock so that restocks are unnecessary, as discussed next.

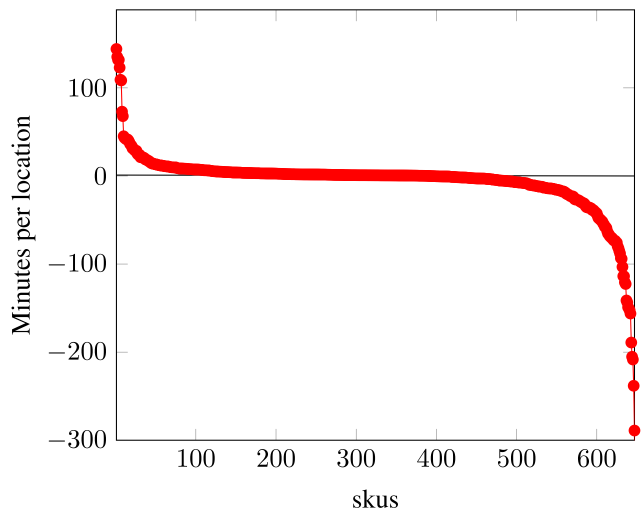


Figure 7.4: Skus ranked by labor efficiency

Storing all of a sku forward

If a small upper bound on the future inventory of sku i is known then we should consider allocating enough forward locations to hold all of sku i, in which case restocking is unnecessary and all picks, whether for less-than-pallet quantities or for full-pallets will be from the forward area. In this case the net-benefit is

$$s\left(p_i + D_i\right).$$

(Note that, because pallets are unit-loads, the number of trips to pick full pallets must equal the number of pallets requested, not the number of requests. This is different from carton-picks, where we estimate the number of trips by the number of requests.) The labor-savings per pallet location (that is, the labor efficiency) is then

$$s\left(p_i + D_i\right)/u_i. \tag{7.4}$$

Theorem 7.2. *If, by storing all of its pallets, a sku offers greater labor savings per pallet location than if stored in the minimum amount, then—if it will fit—it should be stored completely in the forward pick area.*

Let us consider such skus for a moment. Considering Expressions 7.3 and 7.4, we see that they are those skus that are represented by relatively few pallets and that are requested frequently and often in full-pallet quantities. It makes sense that they would be attractive candidates for a forward-pick location because then they could be picked quickly and without having to replenish from reserve.

Let us call such skus (those for which Expression 7.4 is greater than or equal to Expression 7.3) *immediate candidates for complete inclusion in the forward pick area*.

Notice that these skus include those for which it would be disadvantageous to store minimum quantities forward. As we have argued, any sku that is an immediate candidate for complete inclusion should be stored so that either all of its pallets are within the forward pick area or else none are. For the remaining skus Expression 7.3 is greater and so each such sku has a stronger argument, at least initially, to win its minimum practical number of pallet locations.

We can now model the decision of which skus to store forward and in what quantities. For each sku j that is an immediate candidate for complete inclusion define a decision variable z_i that equals 1 if all of the pallets of that sku are stored in the forward pick area, requiring u_i locations, and 0 if none are stored. If $z_i = 1$ then the net benefit accrued is $s\,(p_i + D_i)$.

For each remaining sku i define a decision variable x_i that equals 1 if the sku is stored forward in the minimum practical quantity and 0 otherwise. If $x_i = 1$ then l_i pallet locations forward are allocated to sku i and the total net benefit is $(sp_i - c_r d_i)$. Similarly, define y_i to be 1 if the remaining pallets of sku i are to be stored forward, which would be allocated an additional $u_i - l_i$ pallet locations and return total net benefit of $sD_i + c_r d_i$. But we can allocate the additional pallet locations to sku i only *after* the minimum allowable has been allocated. We enforce this by adding the constraint $y_i \leq x_i$.

We can now write a formalization of the problem of allocating N forward pallet locations.

$$\max \sum_i \left((sp_i - c_r d_i)\, x_i + (sD_i + c_r d_i)\, y_i \right)$$

subject to

$$\sum_i \left(l_i x_i + (u_i - l_i)\, y_i \right) \leq N \tag{7.5}$$

$$x_i, y_i \in \{0, 1\} \tag{7.6}$$

$$y_i \leq x_i \tag{7.7}$$

This is similar to a Knapsack Problem (Appendix B) but with additional constraints 7.7 for those skus that are not immediate candidates for complete inclusion. The Greedy Heuristic for the Knapsack Problem ranks items by net-benefit per unit of resource consumed and then successively chooses items until the next candidate fails to fit (would cause the resource constraint to be violated). This heuristic produces a feasible solution that is guaranteed to be no farther from an optimal solution than the value of a single item. In the context of a warehouse such a bound will be expected to be very small and the solution very near optimal because there may be hundreds or thousands of pallet locations to be allocated and no sku will request very many.

Theorem 7.3. *The Greedy Heuristic for the Knapsack Problem generates a feasible and near-optimal solution to the problem of allocating pallet locations.*

Proof. It need only be argued that the Greedy Heuristic produces a solution that also satisfies the additional constraints 7.7, which only to those skus that are not immediate candidates for complete inclusion. This may be done by showing that the marginal rate-of-return of any x_i is always at least as large as that of y_i and so the Greedy Heuristic is never forced to set $y_i = 1$ before setting $x_i = 1$. Consequently the heuristic solution must satisfy the additional constraints 7.7.

The skus that are not immediate candidates for complete inclusion satisfy

$$\frac{sp_i - c_r d_i}{l_i} > \frac{sp_i + sD_i}{u_i}.$$

Appealing to the simple algebraic fact that, for any numbers a, b, c, d, if $a/b > c/d$ then $a/b > (c-a)\,/\,(d-b)$, it follows that

$$\frac{sp_i - c_r d_i}{l_i} > \frac{sD_i + c_r d_i}{u_i - l_i}.$$

Therefore the Greedy Heuristic for the Knapsack Problem will set x_i to 1 before setting y_i to 1 and so must generate a solution that satisfies constraints 7.7. □

7.1.4 Allocating space by auction

Our solution procedure is another realization of a general theme of this book: When possible, avoid monolithic solution techniques and decentralize the decision-making (in this case, the allocation of space). In effect, we invite each sku to make a business case with which to compete for an allocation of space. That business case is simply the net-benefit per pallet location.

We can think of the Greedy Heuristic as a sort of auction in which each sku bids for pallet locations. First, the skus are divided into two types: Those that are immediate candidates for complete inclusion, whose sole bid is $s\,(p_i + D_i)\,/u_i$; and the remaining skus. The remaining skus have two bids: an initial bid to win their minimum allotment l_i of space; after which they use a second bid to compete for $u_i - l_i$ locations to hold any remaining pallets.

One might image each of these skus to hold two bidding paddles with the allowable values of its bids imprinted thereon. Bid #1 is

$$(sp_i - c_r d_i)\,/l_i, \tag{7.8}$$

which is the value per pallet location sku i can offer for its minimum required pallet locations. If sku i wins with this bid, it is awarded l_i pallet locations and its paddle #1 is discarded. Thereafter, sku i can bid only with its second paddle, for which the bid is

$$(sD_i + c_r d_i)\,/\,(u_i - l_i)\,, \tag{7.9}$$

which is the value per pallet location sku i can offer for additional space beyond l_i. If sku i wins with its second bid, it receives sufficiently many locations to hold all its pallets and then retires from subsequent bidding.

During the auction, each pallet location will have been auctioned off for the best price, which means that it will generate the greatest possible labor savings. However, this might not be true of the last few pallet locations: Eventually a sku j will win the auction and there will not be enough remaining pallet locations to hold all the required pallets, so these last few pallet locations may not be allocated optimally. This error is inherent in the Greedy Heuristic, but for problems of realistic size will be negligible.

7.2 Redirecting uneconomical picks

Some facilities relax Assumption 7.2 to allow more than one active pick location for a sku:

Assumption 7.3 (Multiple active pick locations)**.** *Each sku may have up to two location from which to pick less-than-pallet quantities, one in the forward-pick area and one in the reserve area.*

Under this protocol we can evaluate each pick to see whether it makes sense to fill it from a forward location. If the pick exceeds the threshold fraction s/c_r of a pallet then it is cheaper to pull it from bulk storage.

Directing big picks to the bulk area can reduce labor costs, but there are disadvantages too. In particular, there are more "broken pallets" in the warehouse, and each is an opportunity for inventory loss, especially in the bulk area, which is harder to manage carefully.

Example 7.3. *Suppose that the savings per forward pick averages 1 minute and it takes 3 minutes on average to restock a forward pallet. How should a customer request for 72 cartons of sku A be filled if sku A is stored with 48 cartons per pallet?*

The request for 72 cartons represents $72/48 = 1.5$ pallets. The one full pallet should be picked from bulk; and the remaining half pallet exceeds the threshold $s/c_r = 1/3$ pallet and so should also be filled from bulk.

This can be incorporated in the procedure for assigning space in the forward-pick area by forecasting the numbers of small picks, large picks, and full-pallet requests for each sku and determining bids based on these numbers. (See Question 7.20.)

7.3 Summary

- When skus are stored as full pallets in a fast-pick area then each sku is either:
  - not in the fast-pick area at all; or
  - is in at its minimum practical amount; or else
  - every pallet of that sku is in the fast-pick area.
- Those skus of largest labor-efficiency (labor savings per pallet location) have priority for storage in the forward area.
- Any request for a fraction of a pallet exceeding

$$\frac{\text{savings per pick from forward area}}{\text{cost of replenishing a pallet to the forward area}}$$

  is cheaper to fill from the pallet reserve area.
- As for most optimization models, the data is of varying types and qualities:

- Pick and restock costs, and so s and c_r, are *historical averages*, which in turn depend on which skus have been stored forward in the past.
- Numbers of picks p_i and restocks d_i during the planning period are *forecasts*.
- Minimum number l_i of forward locations required by each sku i is a *policy* decision regarding how much protection is wanted against stockout or congestion.
- The maximum number of pallet locations u_i that could be filled by sku i depends on the maximum number of pallets, which is a *forecast*.
- The number N of forward locations is direct measurement of an area management has deemed forward.

See [49] for another approach to slotting pallets in a forward area from which to pick cartons.

7.4 More

7.4.1 Pallet presentation

When picking cartons from pallet rack it is important to configure the rack to leave ample headroom so that the order-picker does not hit his head on a crossbeam. Suggested height is around 7 feet, which means diminished space utilization. However, it is not necessary to leave headroom for pallet openings above the first level because these will be accessed by person-aboard truck and the driver can adjust his height.

Generally it is preferable to store pallets with the 40-inch side on the pick face because this means that more skus can be presented within a given length of aisle. But when storing 4-way pallets to support case-picking, sometimes it is preferable to orient them with the 48-inch side on pick face: For example, if the pallet has many small cases, the pallet is more shallow and so the order-picker does not have to reach as far.

7.4.2 Congestion

There are two types of congestion to which order-picking is susceptible.

- Interference at a location, when both pickers want to pick from the same small area of the warehouse.
- Interference in an aisle: when one worker wants to pass another but is unable to because of the narrowness of the aisle.

Locations of popular SKUs are susceptible to both types of congestion because many pickers will stop there. And the congestion will be even worse if the product is requested in large quantities because order-pickers will take longer at that location.

These issues could be seen at an auto parts distributor in Indianapolis. Their philosophy was to let orders accumulate to get pick-density, and then use many pickers on a single shift. To reduce congestion in picking, they distributed the most popular product throughout the warehouse on the ground level.

One problem with this approach was that it created congestion on the dock, as freight had to be staged and sorted before loading. An alternative approach would be to reduce congestion by using fewer order pickers over two shifts, which would allow concentration of popular product into fewer aisles, for less travel.

The simplest model of congestion is to estimate the probability of multiple order-pickers being in the same aisle at the same time. One might do that by assuming that, for any one picker, the probability of being in an aisle is proportional to the historical number of requests from that aisle. This captures the obvious intuition that order-pickers are more likely to be in an aisle containing more popular skus.

7.4.3 Pallet-building

Everything is simple if cartons are picked and then shipped by a parcel service, because the parcel service does all the handling and sortation.

It is rather more complicated if the cartons are to be assembled into "mixed" pallets; that is, assortments of cartons sharing the same pallet. Picking cartons to pallet can be challenging in the same way as 3-dimensional Tetris. Cartons of many shapes, sizes, weights, and fragility must be packed tightly together and quickly: Tightly so that the pallet remains intact and the cartons are protected during subsequent transport; and quickly because pallets are typically assembled on the floor, and so on valuable real estate.

There are several goals in building pallets but the most important is achieving a high, tight, and stable load. As in Tetris, it is a good strategy to try to build full layers of even height. Ideally, a sku going to one customer should not be split among pallets, but if it is unavoidable, put the sku on as few pallets as possible.

Large, heavy items should be on the bottom of each pallet and light, small items on top. Order-picking must support this strategy, for example, by storing product from heaviest to lightest along the pick path, so that the order-puller can build a stable pallet without undue travel. Alternatively, if using automation, cartons can be dispensed in the correct sequence.

It is also useful to think in terms of building pairs of pallets, with one to be loaded on top of the other in the trailer. This means that the bottom pallet should have a level top and be capable of supporting weight.

It can be difficult to compute a good pallet architecture if there is much variety in the cartons: Much more than in Tetris, the working surface on which to pack can become quite complex and so it can be challenging to evaluate the result of placing the next carton. The problem is much easier in a finished goods warehouse, where the challenge is to stack identical cartons. (However, there are some additional goals in this case, such as that of making every carton visible from the outside, to enable visual confirmation of quantities.)

Robotic arms may be adapted to build pallets, but they tend to be used mostly by distributors of beverages, such as beer, that have relatively high margins and generally homogeneous carton sizes. Even then, the top levels of pallets are generally finished by a human, who is better — for now — at placing small or fragile cartons.

7.5 Questions

Question 7.1. *List the six data elements necessary to allocate forward pallet slots and classify each element as one of the following: a measured datum, a policy, an average, or a forecast.*

Question 7.2. *Which of the following costs are properly included in the cost per restock of a sku in the forward pick area supporting picking cartons from pallets? Explain in each case.*

- *The time to travel between the forward pick area and reserve storage*
- *The time to travel up to the pallet (which may be at an upper level)*
- *The time to insert forks to retrieve the pallet*
- *The time to remove shrink-wrap from the pallet to make the cartons available for order-picking*

Question 7.3. *Can it every be* dis*advantageous to pick a sku from a forward-pick area? Explain.*

Question 7.4. *Consider a fast-pick area where cases are picked from pallets (and all full-pallet picks are from reserve storage and may be ignored). Storing a sku in the fast-pick area realizes a savings of 1 minute per pick; but each restock requires about 3 minutes. Because of volatility of purchasing, management is unable to specify a reasonable upper bound on how many pallets of each sku may be expected in the warehouse. Which of the following skus has greatest claim to storage in the fast-pick area?*

Sku	***Case picks***	***Demand (pallets)***
A	*600*	*20*
B	*1000*	*100*
C	*200*	*2*

Question 7.5. *Answer the previous question with the following, additional information that tells when to trigger replenishment of the fast-pick area.*

Sku	***Case picks***	***Demand (pallets)***	***Reorder point (pallets)***
A	*600*	*20*	*2*
B	*1000*	*100*	*4*
C	*200*	*2*	*1*

How many pallet locations could these skus usefully occupy in the forward pick area?

Question 7.6. *Consider four skus that are competing for space in a fast-pick area devoted to picking cartons from pallets:*

Sku	*Carton picks*	*Carton demand (pallets)*	*Full pallet demand*	*Min (pallets)*	*Max (pallets)*
A	100	8	20	3	40
B	100	40	1	1	Unknown
C	100	20	20	2	10
D	10	1	1	2	2

A. Compute and rank the bids of the skus for space on the ground floor of selective rack. Assume that on average it requires 1 minute per pick from the ground level, 2 minutes from a higher level, and 3 minutes per restock.

B. Compute and rank the bids of the skus for space in 2-deep pallet flow rack. Use the same average labor times.

C. Compute and rank the bids of the skus for storage in a fast-pick area that is 2-deep floor storage right at the shipping dock, with pallets stacked 2-high. Use the same average labor times.

D. Reconsider part A but assume that the overstocks are segregated by sku into specific reserved regions of the warehouse, with the space within each region shared. How would your answer change if it takes 3 minutes to restock a pallet of sku A, 2.1 minutes for sku B, 2.25 minutes for sku C, and 3.5 minutes for sku D?

Question 7.7. *Consider four skus that are competing for space in a forward pick area. Each forward location holds a single pallet and will be restocked from bulk storage. Assume you forecast the following activity during the next planning period (where demands and minimum/maximum to be stored are measured in pallets).*

sku	*Carton Picks*	*Carton Demand*	*Pallet Demand*	l_i	u_i
A	100	10	50	2	3
B	200	20	10	2	5
C	200	70	5	1	2
D	400	80	10	1	10

A. Compute and rank the bids for space, assuming the forward pick area is the ground floor of selective rack that is to be used only for case-picking (all full pallets must be picked from bulk storage). Assume that on average it requires 1 minute per pick from the ground level, 2 minutes from a higher level, and 3 minutes per restock.

B. Continuing the problem of part A: How would the space be allocated if it is allowed to pick full pallets from the forward pick area? Compute and rank the bids for space.

C. To protect against recent variable deliveries from the vendor, the on-hand inventory of sku A within the warehouse will be increased slightly. Will the claim of sku A to space in the forward area be weakened, strengthened, or neither? Explain.

D. Sku B, which was packed 64 cartons to a pallet, will now be packed 72 cartons per pallet. Has the claim of sku B to space in the forward pick area been weakened, strengthened, or neither? Explain your answer.

Question 7.8. *Compute and rank all positive bids for forward pallet locations offered by the following skus. Use this ranking to auction off 50 pallet locations on the first level of single-deep pallet rack. Which skus appear only in the forward area? Which in both? Which only in reserve? How many picks come from the first level? How many from higher levels? How many restocks are incurred? How many pallets are shipped from each area? How much are pallet locations #48, 49, and 50 worth to you?*

Assume that on average it requires 1 minute to pick a carton from the first level, 2 minutes to pick a carton from higher levels, and 3 minutes to drop a pallet to the first level. In the table below, all demand is given in pallets, as are the minimum and maximum amounts to be stored.

Sku	***Carton Picks***	***Carton demand (pallets)***	***Full pallet demand***	***Min (pallets)***	***Max (pallets)***
A	*18.4*	*4.4*	*1.4*	*2*	*7*
B	*17.0*	*4.0*	*1.2*	*2*	*6*
C	*18.3*	*4.8*	*10.3*	*2*	*16*
D	*1.0*	*0.1*	*0.0*	*2*	*2*
E	*4.2*	*0.6*	*3.2*	*2*	*5*
F	*2.0*	*0.1*	*1.0*	*2*	*2*
G	*12.7*	*1.3*	*10.7*	*2*	*13*
H	*17.3*	*3.6*	*4.2*	*2*	*9*
I	*4.2*	*1.0*	*5.2*	*2*	*7*
J	*1.8*	*0.2*	*0.0*	*2*	*2*

Question 7.9. *A. Suppose that, to better protect against stockouts in the fast-pick area, the minimum allowable storage amount of a sku has recently increased. Is this likely to increase or decrease its claim to space in the fast-pick area? Explain.*

B. Suppose you never have more than 1 pallet of a particular sku in the warehouse. Does knowing this increase or decrease its suitability for storage in the fast-pick area? Explain.

C. Consider two skus that are picked as cartons from pallets. Assume they experience identical patterns of customer orders but that one is never held in quantities exceeding 10 pallets and the other may be held in quantities up to 20 pallets. How does this affect their bids for space in the fast-pick area and why?

D. Consider two skus that are picked as cartons from pallets. Assume they experience identical patterns of customer orders for carton-quantities but one is ordered more frequently than the other as full pallets. How does this affect their bids for space in the fast-pick area and why?

E. Consider a sku that is picked as cartons from pallets. Up to now it was packaged by the vendor as 36 cartons per pallet but has recently been repackaged as 48 cartons per pallet. How does this affect its suitability for space in the fast-pick area and why?

Question 7.10. *Suppose that during the last week a sku was requested in full pallet quantities 23 times, which accounted for a total demand of 40 pallets. How many times did a worker have to travel to a location holding this sku? Explain.*

Question 7.11. *For an operation picking cartons from pallets we estimated the number of trips to a location as the number of times the sku in this location was requested by customers. Explain why this is* not *a good estimate for orders in which the customer requests quantities exceeding a full pallet.*

Question 7.12. *Consider a warehouse in which every sku is ordered in such large quantities that the average number of picks per pallet is less than the ratio of cost-per-restock to savings-per-forward-pick. How much restocking would be required to support a fast-pick area?*

Question 7.13. *Consider a forward area from which cartons are picked from pallets. Suppose that the customers for sku A continue to order with the same frequency as previously but the quantity that they order increases from averaging 0.1 pallet per request to 0.5 pallets per request. Has the strength of the claim of sku A for space in the forward pick area increased or decreased? Why?*

Question 7.14. *Suppose we expect never to hold more than 20 pallets of the sku A mentioned immediately above. If the customers for sku A continue to order with the same frequency as previously but the average quantity per order increases above 2 pallets per request, has the strength of the claim of sku A for space in the forward pick area increased or decreased? Why?*

Question 7.15. *Sku A is stored as pallets, with 18 cases per pallet. Here are the requests received recently for sku A.*

Order #:	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
#-Cases:	*12*	*30*	*22*	*15*	*8*	*32*	*9*	*20*	*4*	*10*

If sku A is allocated a single pallet location in the forward pick, how many times must it have been restocked from bulk storage?

Question 7.16. *Consider two skus, A and B, both of which are picked as cartons, and almost always as a single carton per customer order. Everything about the two skus are identical except that sku A is ordered twice as frequently as sku B.*

A. Which has the stronger claim to storage in the forward pick area?

B. How would your answer change if the two skus are identical except that sku A is ordered in twice the amounts as sku B?

C. How would your answer change if the two skus are identical except that reserve pallets of sku A are stored twice as far away as the reserve of sku B?

Question 7.17. *From which location would it be most economical to pick a sku for which the average order quantity is a half-pallet?*

- *Bulk storage, where the average time per pick is 2 minutes*

sku	Cartons/pallet	Forecast carton picks	Forecast carton demand
A	32	315	3,292 cartons
B	24	1,017	4,944 cartons

Table 7.1: Candidates for storage in a forward area from which cartons are picked from pallets

- *A ground floor location in 1-deep pallet rack where the average time for a pick is 1 minute and for a restock is 3 minutes*
- *2-deep pallet flow rack alongside which a conveyor runs, where the average time for a pick is 0.5 minutes and for a restock is 2 minutes*

Question 7.18. *Consider a distribution center in which customers have been allowed to order arbitrary numbers of cartons of each sku. To achieve efficiencies in handling, management is considering requiring that all purchases be rounded up to* tier *quantities (that is, to integer numbers of tiers).*

What would be the effect on the net benefits of the skus currently in the forward pick area? What would be the likely effect on the value of the forward pick area (that is, on the total net benefit)?

Question 7.19. *Consider a warehouse with a forward case-pick area. Some overstock is held in higher levels of nearby pallet rack and some is held in an outlying warehouse and restocked from there. Assume the forward pick area has been stocked and may not be changed; for which skus should the overstock be held nearby (rather than in the outlying warehouse)?*

- *Those skus forecast to be most frequently requested?*
- *Those skus forecast to sell in greatest physical volume?*
- *Those skus forecast to sell the most cases from the forward case-pick area?*
- *Those skus forecast to sell the most pallets from the forward case-pick area?*
- *Those skus forecast to sell the most pallets as full-pallet picks?*
- *Those skus forecast to sell the most total pallets (from either case-pick or full-pallet pick)?*

Question 7.20. *Consider the skus of Table 7.1 as candidates for stocking a forward area in which cartons are picked from pallets. Less-than-pallet picks can be from either the forward area or else from bulk storage; but each sku can have at most one broken pallet at a time from which to pick.*

On average it takes about 1.5 minutes per pick from the forward area and about 2.5 minutes per pick from reserve. Restocks average 4 minutes each.

Assume that you have determined that, to avoid stock outs while waiting for replenishment, any sku in the forward area should have at least 2 pallets forward. And, *because sku B is so popular, if it is stored forward, it should occupy at least 3 different locations to avoid congestion, while 1 location is sufficient for sku A.*

A. Rank the skus of Table 7.1 according to their claim to space in the forward pick area of Figure 7.5. Assume that pallets may not be stacked in the forward area because the top one would be too high for picking.

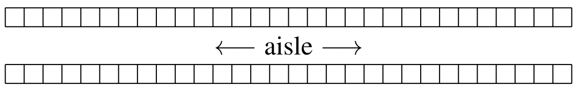


Figure 7.5: The view from above of a forward area consisting of a single aisle flanked by two rows of 1-deep pallet locations.

B. How would you answer to part A change if the forward pick area was that shown in Figure 7.6?

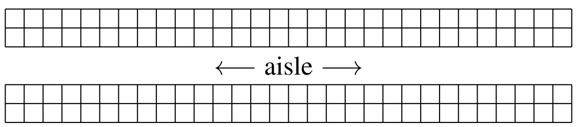


Figure 7.6: The view from above of a forward area consisting of a single aisle flanked by two rows of 2-deep pallet flow rack.

C. Suppose overstock for sku A is moved closer to the forward area than are the overstock for other skus, so that it averages only 3 minutes per restock. How would this change the ranking of part A?

D. Suppose you are considering purchase of another high-reach forklift to increase warehouse capacity to move pallets in/out of the higher levels of pallet rack. If purchased, the new forklift will make restocks faster, and they are projected to average only 3 minutes each for any sku. How would this change the net benefit of the forward pick area of part A? In other words, compute the value of this additional forklift.

E. Revisit part A assuming you are willing to have up to two active pick locations, one forward and one in bulk, for each sku. Assume you have made a detailed forecast of how this sku will be ordered, as which is given in Table 7.2.

F. How would your answer to Part A change if sku A could be double-stacked?

G. How would your answer to Part B change if sku A could be double-stacked?

Quantity (#-cartons)	**Forecast picks of sku A**	**Forecast picks of sku B**
1	4	116
2	14	163
3	15	207
4	25	185
5	17	57
6	28	60
7	19	44
8	22	20
9	16	43
10	12	35
11	13	13
12	19	15
13	18	12
14	15	6
15	14	8
16	12	4
17	10	10
18	9	6
19	3	3
20	7	5
21	5	2
22	3	2
23	2	1
24	8	-
25	2	-
26	1	-
27	0	-
28	1	-
29	0	-
30	1	-
31	0	-

Table 7.2: Forecast number of requests, by number of cartons, for each of sku A, which is packed 32 cartons per pallet, and sku B, which is packed 24 cartons per pallet.

Chapter 8

Layout of a piece-pick-from-carton area

Piece-picking is the most labor intensive activity in the warehouse because the product is handled at the smallest units-of-measure. Furthermore, the importance of piece-picking has greatly increased because of pressures to reduce inventory while expanding product lines. Warehouses that, 20 years ago, might have shipped cartons to customers now ship pieces and much more frequently.

In a typical operation pieces are picked from cartons and neither picking nor restocking is unit-load, as shown in Figure 8.1.

One of the first efficiencies a warehouse should consider is to separate the storage and the picking activities. A separate picking area, sometimes called a *fast-pick* or *forward pick* or *primary pick* area, is a sub-region of the warehouse in which one concentrates picks and orders within a small physical space. This can have many benefits, including reduced pick costs and increased responsiveness to customer demand. However, there is a science to configuring the fast-pick area.

8.1 What is a fast-pick area?

The fast-pick area of a warehouse functions as a "warehouse within the warehouse": Many of the most popular stock keeping units (skus) are stored there in relatively small amounts, so that most picking can be accomplished within a relatively small area. This means that pickers do less unproductive travel and may be more easily supervised. The trade-off is that the fast-pick area may require replenishment from bulk storage, or *reserve*.

The basic issues in the design of an fast-pick area are

- Which skus to store in the fast-pick area? And
- How much of each sku to store.

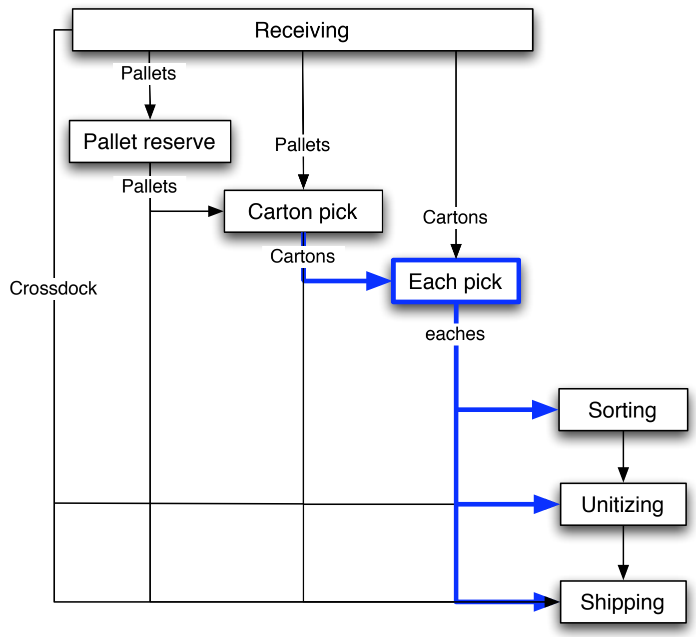


Figure 8.1: Typical flow of product through piece-picking

The answers to these questions determine the value of the fast-pick area, for if skus are stored there in insufficient amounts, the cost of restocking them can outweigh any savings in pick costs.

Initially, we will answer these questions by a *fluid model* that treats each sku as an incompressible, continuously divisible fluid; that is, we will ignore the fact that a sku actually comes in discrete "chunks" of space such as pallets or cases or individual units. In this simplest model, we can imagine the warehouse as a bucket holding various fluids (skus); and we will simply measure the volume of storage space to be devoted to each sku. During the discussion we will point out times when this point of view can lead to inaccuracies.

A more detailed product layout, usually called a "slotting", will explicitly account for the geometry of storage and tell *exactly* where each sku should be located (for example, on the third shelf of the second section of aisle 2, oriented with case width to the front). Such a plan can be constructed but it is beyond the scope of this discussion. Nevertheless, the fluid model has the advantage that it can be realized easily, for example, on a spreadsheet; and its answers are benchmarks or goals because they represent the ideal.

8.2 Estimating restocks

Since a fast-pick area is maintained by restocking, we must first estimate the cost of restocking. The cost of restocking a sku depends on the particulars of the warehouse but may include any of the following.

- The number of times the sku requires replenishment.
- The number of storage units to be replenished.
- When the restock occurs (during picking or on another shift, when timing might be less critical)

To be both usefully general and simple, we shall develop a theory in which the cost of restocking is based mostly on the number of restocks required. The first observation is that the number of restocks depends on the type of storage unit: In particular, if the skus are stored as pallets then each pallet will require separate handling. On the other hand, if the sku is stored in smaller containers, such as cases, one can estimate the number of restocks by a fluid model. Consider sku i of which volume v_i is stored in the fast-pick area. How often must we restock sku i? That depends on its rate of flow f_i through the warehouse. Flow is measured in volume per year (for example, cubic meters per year) and may be determined from warehouse data as follows:

$$\text{flow, in volume/year} = \left(\frac{\text{\# items/year}}{\text{\# items/case}}\right)\text{volume/case.}$$

Estimate 8.1 (Fluid model (for small parts))**.** *If volume v_i of sku i is stored in the forward pick area, from which an annual volume of f_i of sku i is picked, then sku i will require about*

$$\frac{f_i}{v_i} \textit{ restocks per year.} \tag{8.1}$$

These restocks are internal to the distribution center and are determined by rates of flow of the skus and the quantities stored, not by the company purchasing department.

There are some assumptions implicit in this model of restocks. We assume that a pick quantity never exceeds the full allocation of a sku in the forward pick area. (In practice unusually large order quantities are typically filled from bulk storage.) And if a pick quantity exceeds the amount available in the forward pick area at that time, a restock is triggered.

In addition we assume that the entire restock quantity for a sku can be carried in one trip. In this case the work to restock a sku consists of the following components:

1. Travel between the forward pick and bulk areas: This magnitude of this cost is typically determined by the warehouse layout and not by the locations of individual skus.

2. Travel within the bulk area to locate stock: This is variable but unpredictable because of “random storage” in the bulk area. It is reasonable to assume an average value here. (We will relax this assumption later.)

3. Travel within the forward pick area to the location(s) to be restocked: This is a small component of cost because a forward pick area is a relatively small part of the warehouse.

4. Handling storage units: This cost is is determined by the total volume of product sold and is fixed with respect to the decision of how much to store forward. For example, if a sku sells 100 cartons annually from the forward pick area, then all 100 cartons must be handled, independently of the quantity stored forward.

Because these cost components are either small or fixed with respect to the decision of quantity to store, we take the number of restocks as a measure of the cost of maintaining the forward pick area.

8.3 How much of each sku to store in the fast-pick area?

Assume that skus have already been chosen for storage in the forward pick area, and so the only variable cost is the labor to restock as necessary. For the remainder of this chapter, let the physical volume of available storage be normalized to 1 and let f_i be scaled accordingly. Let v_i represent the fraction of space allocated to sku i.

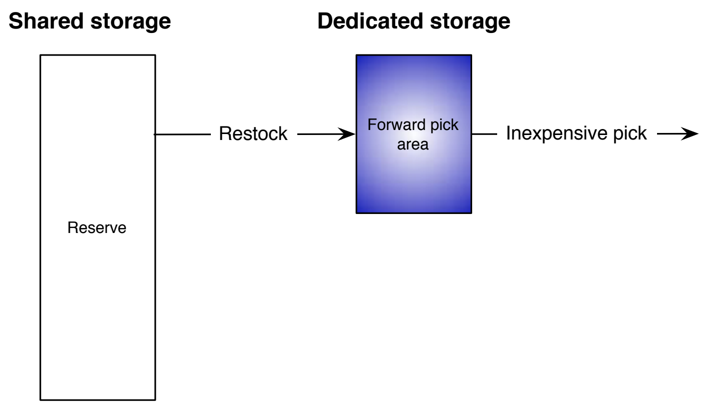


Figure 8.2: In the simplest case, all the skus in the fast-pick area have been already chosen.

8.3.1 Minimizing labor to maintain a forward pick area

Following [28], the problem of allocating space among n skus to minimize annual restocks may be expressed as:

$$\min \sum_{i=1}^{n} f_i/v_i \tag{8.2}$$

$$\sum_{i=1}^{n} v_i \leq 1 \tag{8.3}$$

$$v_i > 0, \quad i = 1, \ldots, n \tag{8.4}$$

One can anticipate the nature of the result by considering just two skus; call them A and B. Then the problem becomes

$$\begin{aligned} \min f_A/v_A + f_B/v_B \\ v_A + v_B \leq 1 \\ v_A, v_B > 0 \end{aligned}$$

We can replace the first constraint with an equality constraint $v_A + v_B = 1$ because no solution could be optimal that left unused space. Now substitute $v_A = 1 - v_B$ into the objective function so that the problem becomes

$$\min \frac{f_A}{1 - v_B} + \frac{f_B}{v_B}.$$

This minimization may be solved by setting the derivative with respect to v_B to zero, solving for v_B, and then substituting back into $v_A + v_B = 1$. The result is that

$$v_A^* = \left(\frac{\sqrt{f_A}}{\sqrt{f_A} + \sqrt{f_B}} \right), \text{ and } v_B^* = \left(\frac{\sqrt{f_B}}{\sqrt{f_A} + \sqrt{f_B}} \right).$$

More generally,

Theorem 8.1. *To minimize total restocks over all skus $j = 1, \ldots, n$ in the forward pick area, the fraction of available storage space devoted to sku i should be*

$$v_i^* = \left(\frac{\sqrt{f_i}}{\sum_{j=1}^n \sqrt{f_j}} \right). \tag{8.5}$$

Proof. (By induction.) The claim is trivially true if there is but one sku to go into the forward-pick area, for it is alloted the entire space. Assume the results holds for k skus and consider the problem of choosing v_i to solve the following.

$$\begin{aligned} \min \left(\sum_i^{k+1} f_i / v_i \right) & \\ \sum^{k+1} v_i &\leq 1 \\ v_i &> 0 \end{aligned}$$

Let v_i^* be the optimal volumes and let $v = \sum_{i=1}^k v_i$. By the inductive hypothesis, $v_i^* = v\sqrt{f_i} / \left(\sum_{j=1}^k \sqrt{f_j} \right)$ and so the number of times each sku i among the first k skus will be restocked is

$$\frac{f_i}{v_i^*} = \frac{f_i}{v\sqrt{f_i} / \left(\sum_{j=1}^k \sqrt{f_j} \right)} = \frac{\sqrt{f_i} \left(\sum_{j=1}^k \sqrt{f_j} \right)}{v}$$

and the total number of restocks among the first k skus is

$$\frac{\left(\sum_{i=1}^k \sqrt{f_i} \right)^2}{v}.$$

If the first k skus occupy volume v, then the $(k+1)$-st sku must occupy the remaining space $1 - v$, otherwise the solution could be improved. Now we can rewrite the problem as one of partitioning the space between the first k skus, which, by the inductive hypothesis, know how to share space amongst themselves, and the $(k+1)$-st sku. In other words we must find v to solve the following.

$$\begin{aligned} \min \frac{\left(\sum_{i=1}^k \sqrt{f_i} \right)^2}{v} + \frac{f_{k+1}}{1 - v} & \\ v + v_{k+1} &= 1 \\ v_i &> 0 \end{aligned}$$

Differentiating the objective function with respect to v, setting the derivative to 0, and solving for v results in

$$
\begin{aligned}
-\frac{\left(\sum_{i=1}^{k}\sqrt{f_i}\right)^2}{v^2}+\frac{f_{k+1}}{(1-v)^2} &= 0\\
\frac{f_{k+1}}{(1-v)^2} &= \frac{\left(\sum_{i=1}^{k}\sqrt{f_i}\right)^2}{v^2}\\
\frac{\sqrt{f_{k+1}}}{1-v} &= \frac{\sum_{i=1}^{k}\sqrt{f_i}}{v}\\
v\sqrt{f_{k+1}} &= \left(\sum_{i=1}^{k}\sqrt{f_i}\right)-v\left(\sum_{i=1}^{k}\sqrt{f_i}\right)\\
v &= \left(\frac{\sum_{i=1}^{k}\sqrt{f_i}}{\sum_{i=1}^{k+1}\sqrt{f_i}}\right)
\end{aligned}
$$

and the result follows by substituting the last expression into $v + v_{k+1} = 1$. □

Note that this result gives the ideal amount in which each sku should be stored and this might not be precisely realizable in practice due to the geometry of the skus and the storage medium. The fluid model treats the skus and storage as infinitely divisible rather than as boxes on shelves. Nevertheless, this computation can help identify skus that are stored in amounts that are far from optimum. And when skus are relatively small, such as pharmaceuticals, office supplies, or cosmetics, then the values can be rounded off to match the sizes of available storage containers (cartons, bins, etc.).

We refer to the volumes of Expression 8.5 as the Optimum allocations (OPT). The Optimal allocations enjoy a few special properties that are worth noting:

Theorem 8.2 (“Law of Uniform Restocking”)**.** *Under Optimal allocations each unit of storage space is restocked at the same frequency.*

Proof. Under Optimal allocations, the restocks per unit of space are

$$\frac{f_i/v_i^*}{v_i^*}=\left(\sum^{n}\sqrt{f_j}\right)^2,$$

which is constant and identical for all skus i. □

This means that restocks should be distributed *uniformly* throughout the volume of the forward pick area. Roughly speaking, each aisle should be restocked at the same frequency and each section of shelving at the same frequency. This provides a useful way to benchmark a forward pick area without any measurements whatsoever: Simply ask restockers whether they tend to visit some parts of the forward pick area more often than others; if so, then the storage policy is out of balance and there is excessive restocking.

We have just seen that Optimal allocations call for sku i to receive fraction of the available space equal to $\sqrt{f_i}/\left(\sum_j \sqrt{f_j}\right)$. A similar result holds for allocation of restocking effort:

Lemma 8.1. *Under Optimal allocations, sku i incurs a fraction of the total restocks equal to $\sqrt{f_i}/\left(\sum_j \sqrt{f_j}\right)$.*

8.3.2 Two commonly-used storage strategies

The optimum storage policy of Expression 8.5 is generally unknown to industry. We have asked hundreds of people in the warehousing industry about how they stock a forward pick area. All answers have been one of the two following (which were also observed by [49] to be typical).

- *Allocate the same amount of space to each sku.* We call this the Equal Space (EQS) strategy and model it by $v_i = 1/n$, from which it follows that sku i is restocked nf_i times a year.
- *Store an equal time supply of each sku.* We call this the Equal Time strategy (EQT). Let K be the common number of restocks per planning period. Under Equal Time allocations, each sku i would be restocked this common number of times, so that $f_i/v_i = K$. We have no incentive to leave unused space in the forward pick area and so we may assume that $1 = \sum_j v_j = \sum_j f_j/K$, from which we conclude that $K = \sum_j f_j$ and $v_i = f_i/K = f_i/\sum_j f_j$.

Our idealizations of these two stocking strategies are also fluid models. This simplicity enables us to compare the three strategies in some detail.

Equal Space allocations ignore all differences in sku popularity and size and so it is generally felt to be less effective than Equal Time allocations. In our surveys, people in the warehousing industry unanimously expressed the belief that Equal Time allocations reduce restocks because a more popular sku will be allocated more space. This observation is folk wisdom in the industry; but it is wrong:

Theorem 8.3. *For a given set of skus, Equal Time allocations require the same total restocks, $n\sum_j f_j$, as Equal Space allocations.*

Proof. By simple algebra:

	Equal Space	**Equal Time**
Allocation v_i for sku i	$1/n$	$f_i/\sum_j f_j$
Restocks for sku $i = f_i/v_i$	nf_i	$\sum_j f_j$
Total restocks over all skus	$n\sum_i f_i$	$n\sum_j f_j$

□

There is also an interesting duality between Equal Space and Equal Time allocations.

Corollary 8.1. *The space and the labor consumed by sku i in the forward pick area under each of the two storage strategies are as follows.*

	Equal Space	***Equal Time***
Fraction of space to sku i	$1/n$	$f_i / \sum_j f_j$
Fraction of restocks to sku i	$f_i / \sum_j f_j$	$1/n$

For comparison, under the Optimal strategy sku i gets fraction $\sqrt{f_i} / \left(\sum_j \sqrt{f_j} \right)$ of the space and incurs the same fraction of restocks.

8.3.3 Comparison with optimal

Numbers of restocks

The Equal Space and Equal Time allocations incur more restocks than necessary; but how severe is the waste? We answer this by studying the ratio of the number of restocks under the Equal Space or Time allocations (EQT) to the number incurred under the Optimum allocations (OPT).

We have computed the value of EQT/OPT for several warehouses and found that EQT/OPT ≈ 1.45 for 6,000 fast-moving skus of a major drug store chain. This suggests that this particular warehouse, which had 20 restockers, may have needed only 14. Similarly, for 4,000 skus of a telecommunications company we computed a ratio of 2.44, which means that storing skus in Equal Time allocations incurred more than twice as many restocks as necessary.

Example 8.1. *Consider two skus with flows of 16 and 1 units/year respectively, which are to share 1 unit of storage. The different allocation strategies would result in the following:*

	sku A	*sku B*	*Totals*
flow	*16*	*1*	
Equal Space allocations	*1/2*	*1/2*	*1*
Restocks	*32*	*2*	*34*
Equal Time allocations	*16/17*	*1/17*	*1*
Restocks	*17*	*17*	*34*
Optimum allocations	***4/5***	***1/5***	***1***
Restocks	***20***	***5***	***25***

and the optimal allocation results in almost 30% fewer restocks.

It is difficult to make a general statement about the magnitude of EQT/OPT except in some special cases, such as the following.

Theorem 8.4. *If the values of the $\sqrt{f_i}$ are drawn independently from a distribution with well-defined mean and variance (and coefficient of variation CV), then*

$$\frac{EQT}{OPT} \approx 1 + CV^2,$$

This suggests that, the more diverse the rates of flow of the skus, the more important it is to allocate space optimally, rather than by Equal Space or Equal Time strategies.

Proof. From Theorem 8.1 and 8.3

$$\frac{\text{EQT}}{\text{OPT}} = \frac{n \sum f_i}{\left(\sum \sqrt{f_i}\right)^2} = \frac{\sum f_i / n}{\left(\sum \sqrt{f_i}/n\right)^2}.$$

Estimate the sample mean μ, sample second moment, and sample variance σ^2 of the values $\sqrt{f_i}$ as $\left(\sum \sqrt{f_i}\right)/n$, $\left(\sum f_i\right)/n$, and $\sum \left(\sqrt{f_i} - \mu\right)^2 / (n-1)$, respectively; then because n is large (in the thousands for a typical large North American distribution center) the following is a good approximation.

$$\frac{\text{EQT}}{\text{OPT}} \approx \frac{\mu^2 + \sigma^2}{\mu^2}.$$

$\square$

Manageability

Equal Space allocations and Equal Time allocations each offer a type of uniformity that can simplify warehouse management. For example, under Equal Space allocations the uniformity of storage can simplify space management, especially when old skus are being phased out and new skus introduced. Because all storage slots are the same size, a newly-arrived sku always fits into a space in the fast-pick area. However, there is a cost to this; namely that the Equal Space allocations will require highly non-uniform frequencies of restocking, which might make the restocking process more difficult to manage.

On the other hand, it is not immediately obvious how much work will be required to maintain the fast-pick area by restocking. Similarly, under Equal Time allocations, each sku is restocked at the same frequency and so it is easier to estimate the restocking labor required to maintain the fast-pick area. For example, if each sku in the fast-pick area is stocked with 3-weeks supply then about one-third of the skus must be restocked each week. In some situations this can enable savings because it can allow restocks to be batched. However, Equal Time allocations vary greatly in the amount of space allocated to the skus and so the shelving of a fast pick area stocked under this policy can have many different sizes of slot (space allocated to a sku). This can make it hard to maintain the slotting when old products are discontinued and new ones introduced as it is unlikely that the newly-arrived sku would have a value of flow identical to that of the departing sku. If it is larger then the new sku requires more space than left by the old one; and if stored in this smaller space the new sku will have to be restocked more often than the intended frequency. In this way the uniformity of restocking frequency, which presumably is one of the attractions of the Equal Time Allocation, degrades over time.

Here we show that

Theorem 8.5. *Optimal allocations vary less than those of Equal Time allocations and so are easier to maintain in the warehouse. Similarly, under Optimal allocations the frequencies of restocking skus vary less than under Equal Space allocations.*

This result holds for many natural ways of measuring variability, including the size of the difference between minimum and maximum values, sample variance, and so on.

Proof sketch This idea behind this result is that, for each sku i, the size of the Equal Time allocation and the frequency of restocking an Equal Space allocation both depend directly on the value f_i's. But the size and the frequency of restocking the Optimal allocation depends on the value $\sqrt{f_i}$; and if f_i is very large then $\sqrt{f_i}$ is smaller and if f_i is very small then $\sqrt{f_i}$ is larger. Thus the Optimal allocations tend to avoid the extreme values that might be assumed by the Equal Time or Equal Space allocations. Details of the argument, which is somewhat technical, may be found in [10].

■

Figure 8.3 shows the variability in space under Equal Time allocations compared to those of Optimal allocations; and the variability in number of restocks per sku under Equal Space allocations compared to those of Optimal allocations. These comparisons were generated for 45 randomly-selected skus of a retail chain. We see in each case that Optimal allocations have significantly less variability.

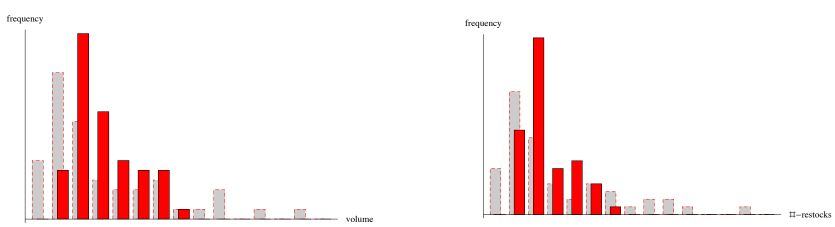


Figure 8.3: Optimal allocations (dark red) have less variability in volume than Equal Time allocations (left) and less variability in number of restocks than Equal Space allocations (right), as this example shows.

8.3.4 Differing costs per restock

Sometimes the cost of restocking a sku depends significantly on the identity of the sku, as would be the case if some overstock were held in an outlying warehouse or if bulk storage was zoned. This can be modeled by charging cost c_i per restock of sku i. Following Hackman and Rosenblatt [28], simply replace any appearance of f_i with the weighted flow $\hat{f}_i = c_i f_i$ and results describing the Optimal allocations still follow. In contrast, Equal Space/Time allocations ignore differences in the costs of restocking.

Simple algebra yields the total cost of restocking under each of the three strategies to be:

EQS	EQT	OPT
$n \sum_i c_i f_i$	$\left(\sum_i c_i\right)\left(\sum_i f_i\right)$	$\left(\sum_i \sqrt{c_i f_i}\right)^2$

The total cost of Optimal allocations is, of course, still the smallest, as may be confirmed by the Cauchy-Schwarz Inequality

$$\left(\sum_i a_i b_i\right)^2 \leq \left(\sum_i a_i^2\right)\left(\sum_i b_i^2\right).$$

Letting $a_i = 1$ and $b_i = \sqrt{c_i f_i}$ shows that OPT $\leq$ EQS; and letting $a_i = \sqrt{c_i}$, $b_i = \sqrt{f_i}$ shows that OPT $\leq$ EQT.

If the cost c_i to restock sku i depends on where in bulk storage or where in the forward area sku i is stored, then both Optimal and Equal Space allocations can take advantage of that by storing skus with large flows f_i in convenient locations (small c_i). No such savings are possible under Equal Time allocations because every sku is restocked at the same frequency and so there are no savings possible by careful placement of skus. A restocker must visit the least convenient locations in bulk storage about as often as the most convenient.

8.3.5 Minimum and maximum allocations

8.3.6 Reorder points and safety stock

Optimal allocations can easily be adapted to account for reorder points and safety stock levels. To guard against stockout in the forward pick area each sku must be stored in sufficient quantity to cover mean lead time demand l_i. We can enforce this by revising the statement of Problem 8.2 to include constraints

$$v_i \geq l_i. \tag{8.6}$$

This extended model can be solved efficiently by an algorithm of [38] that repeatedly allocates space according to Expression 8.5, identifies skus that received less than their minimum required space l_i and increases their allocation to l_i, then reallocating the remaining space among the remaining skus.

In addition, if the allocation of each sku i is to include safety stock s_i (which we assume has been exogenously determined), then a total volume of $S = \sum_i s_i$ within the forward pick area must be devoted to safety stock, leaving the remaining $1 - S$ to hold cycle stock and this is the space that is allocated to minimize total restock costs.

In contrast, it can be problematical to adapt Equal Space or Equal Time allocations to account for lead time demand and safety stock. For example, Equal Space allocations must include the safety stock as part of the allocation (otherwise allocations are unlikely to be equal). If the mean lead-time demand l_i plus safety stock s_i of sku i were to exceed its allocation $v_i = 1/n$ under Equal Space, then presumably sku i would have to be excluded from the forward pick area. Under Optimal allocations, space is reallocated from other skus and so the decision of which skus to include remains more clearly separated from the allocation decision.

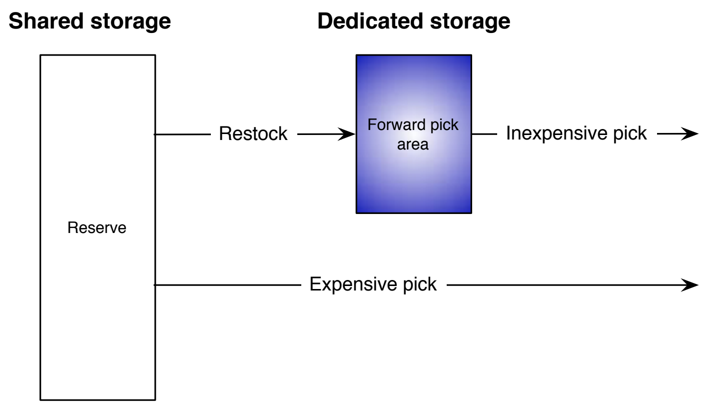


Figure 8.4: Especially large or slow-moving skus are better picked from the reserve area. This allows the space they would otherwise occupy to be devoted to more popular skus.

8.4 Which skus go into the fast-pick area?

Any warehouse has some skus that are quite slow-moving by comparison to the others. It does not make sense to store such a sku in the prime real estate of an fast-pick area: Far better to store more of a popular sku so that we can defer restocking it. This will reduce our restocks but at a cost of occasionally having to pick the slow-moving sku from *reserve*, deep in the warehouse, which is more expensive than picking from the fast-pick area. Therefore the economics become those of Figure 8.4.

To better concentrate on the fast-pick area, let us assume for the moment that the rest of the warehouse, the reserve, is "sufficiently large" that space is not an issue there.

Let the cost-per-pick from the forward pick area be c_1 and from reserve (or some alternative storage area) be c_2. Let our decision variables be

$$x_i = \begin{cases} 1 & \text{if sku } i \text{ is stored in and picked from the fast-pick area;} \\ 0 & \text{otherwise.} \end{cases}$$

The total cost to manage sku i is then the cost of picking it plus, if it is stored in the fast-pick area, the cost of restocking it to the fast-pick area. Then we can formalize the problem minimizing total labor cost by choice of which skus to store in and pick from the forward pick area. Let p_i be the number of picks forecast for sku i during the planning horizon:

$$\min \sum_{i}^{n} (c_1 p_i + c_r f_i / v_i) x_i + c_2 p_i (1 - x_i)$$
$$\sum_{i}^{n} v_i x_i \leq 1$$
$$v_i > 0$$
$$x_i \in \{0, 1\}$$

The first term in the objective function represents the labor cost resulting from maintaining sku i in the forward pick area; and the second term represents the cost of picking sku i from some alternate area.

Frequently it will be convenient to rewrite this as an equivalent problem of *maximizing* the net benefit of the fast-pick area; that is, how much total labor does it save compared to the default of storing all skus in the reserve area and picking from there. Let s be the savings realized when a pick is from the forward area rather than reserve.

$$\begin{aligned} &\text{argmin} \sum_{i}^{n} (c_1 p_i + c_r f_i / v_i) x_i + c_2 p_i (1 - x_i) \\ &= \text{argmin} \sum_{i}^{n} \left((c_1 - c_2) p_i + c_r f_i / v_i \right) x_i + c_2 p_i \\ &= \text{argmax} \sum_{i}^{n} \left((c_2 - c_1) p_i - c_r f_i / v_i \right) x_i \\ &= \text{argmax} \sum_{i}^{n} \left(s p_i - c_r f_i / v_i \right) x_i \end{aligned}$$

Therefore we can express the problem of stocking the forward pick area to maximize the net benefit (pick savings less restock costs) as

$$\max \sum_{i}^{n} \left(s p_i - c_r f_i / v_i \right) x_i \text{ such that} \tag{8.7}$$
$$\sum_{i=1}^{n} v_i x_i \leq 1, \tag{8.8}$$
$$v_i > 0, \tag{8.9}$$
$$x_i \in \{0, 1\}. \tag{8.10}$$

It is worth pausing to note the special nature of the objective function. The net benefit of storing sku i forward in amount v is a discontinuous function of v (Figure 8.5), and once forward space has been allocated to sku i, there are diminishing returns for further space:

$$\begin{cases} 0 & \text{if } v = 0 \\ s p_i - c_r f_i / v & \text{if } v > 0 \end{cases} \tag{8.11}$$

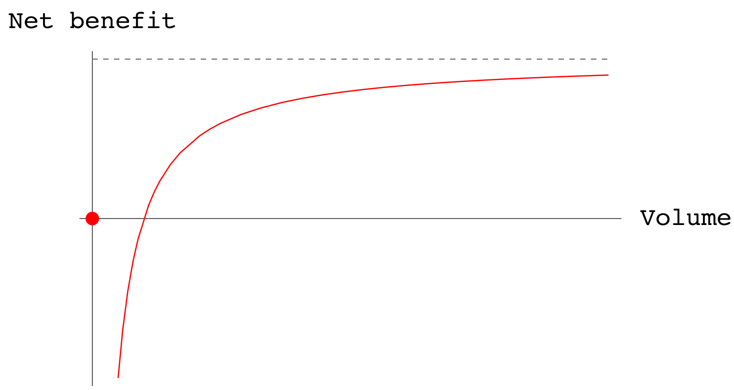


Figure 8.5: The net benefit realized by storing a sku as a function of the quantity stored. The net benefit is zero if sku i is not stored in the forward area; but if too little is stored, restock costs consume any pick savings.

Note that from the shape of the objective function there is a minimum sensible amount of each sku to store in the fast-pick area.

Theorem 8.6 (Minimum sensible storage)**.** *If sku i goes into the fast-pick area at all, put at least volume*

$$\frac{c_r f_i}{s p_i}.$$

Proof. This follows by solving $sp_i - c_r f_i / v_i = 0$ to see what value of v_i results in a net-benefit of 0. □

8.4.1 Selecting skus to minimize labor

Before selecting skus for the forward pick area we make the following observation:

Lemma 8.2. *Any labor-minimizing stocking strategy must store the selected skus in amounts given by Expression 8.5.*

This follows because otherwise a better solution could be constructed by keeping the same set of skus in the forward area but reducing the costs of restocks by allocating space in accordance with Expression 8.5.

Let S be any fixed subset of the skus and suppose we choose exactly and only the skus of this subset for inclusion. Then substituting Expression 8.5 into Model 8.7

allows us to rewrite the selection problem as

$$\max \sum_{i}^{n} \left(sp_i - c_r \sqrt{f_i} \left(\sum_{j \in S} \sqrt{f_j} \right) \right) x_i \tag{8.12}$$

$$\sum_{i}^{n} \left(\frac{\sqrt{f_i}}{\sum_{j \in S} \sqrt{f_j}} \right) x_i \leq 1 \tag{8.13}$$

$$x_i \in \{0, 1\} \tag{8.14}$$

This is an instance of the knapsack problem (Appendix B), which is known to be solvable to a good approximation by a simple greedy heuristic: Select the candidates with greatest "bang-for-buck" (revenue per unit resource consumed, which, in this case, is labor-hours per unit of forward space). There are several reasons to expect a greedy solution to be negligibly different from optimal. First, it will differ from the optimal by no more than the net benefit of a single sku, the first one that does not fit in the remaining space, and this sku will typically be one of the least profitable.

The bang-for-buck for sku i is

$$\left(\frac{p_i}{\sqrt{f_i}} \right) \left(s \sum_{j \in S} \sqrt{f_j} \right) - c_r \left(\sum_{j \in S} \sqrt{f_j} \right)^2 .$$

The main insight, which is a simple observation, is

Lemma 8.3. *The ranking of the skus according to bang-for-buck is independent of S.*

Because we do not know S, we do not know the actual values of bang-for-buck, but we do know that sorting skus by bang-for-buck is equivalent to sorting by

$$\frac{p_i}{\sqrt{f_i}}$$

We shall meet this term again and so we pause to give it a name: It is the *labor efficiency* of sku i because it gives a ranking of which skus contribute most benefit if placed forward.

Accepting the greedy solution to the knapsack problem,

Theorem 8.7. *The skus that have strongest claim to the fast-pick area are those offering the greatest labor efficiency.*

The problem of deciding exactly which skus belong in the fast-pick area is now solvable. Instead of searching over all $O(2^n)$ subsets of the n skus, we need consider only the $O(n)$ ways of partitioning the ranked list of n skus into two pieces, those that go in the fast-pick area and those that do not. The difference in effort is enormous for a typical warehouse, for which n may be on the order of 10^4 or 10^5.

Here, then, is the procedure, first presented in [28], to decide what goes into the fast-pick area and in what amounts.

- Sort all skus from most labor efficient to least.
- Successively evaluate the total net cost of putting no skus in the fast-pick area; putting only the first sku in the fast-pick area; only the first two skus; only the first three; and so on. Choose the strategy that minimizes net cost.[1]

To evaluate the net cost: Charge each sku for each of its p_i picks and for each of its f_i/v_i restocks.

Theorem 8.8. *Choosing skus based on labor efficiency will result in a fast-pick area of total net-benefit that is no farther from optimum than the net-benefit of a single sku.*

Since there are typically thousands or tens-of-thousands of skus considered for forward storage, the worst-case error of this heuristic is negligible. In other words, for all practical purposes, this procedure solves the problem of stocking the fast-pick area so as to realize the greatest possible net benefit.

8.4.2 Stocking to equalize space or restocking frequencies

Sometimes business conditions require that the forward pick area be stocked by Equal Space or Equal Time allocations. How can one minimize the total labor under such a restriction?

What ought to be done

Under EQS allocations, each sku i that goes in the forward pick area is awarded space $1/k$ if there are k skus selected. Substituting $v_i = 1/k$ allows us to rewrite the selection problem as

$$\max \sum_{i}^{n} (sp_i - kc_r f_i)\, x_i \tag{8.15}$$

$$\sum_{i}^{n} \left(\frac{1}{k}\right) x_i \leq 1 \tag{8.16}$$

$$x_i \in \{0, 1\} \tag{8.17}$$

For any fixed k, Problem 8.15 is a knapsack problem and the bang-for-buck of sku i is

$$k\,(sp_i - kc_r f_i)\,.$$

Thus, to minimize labor while selecting skus that will be assigned the same space, simply solve a series of knapsack problems, one for each value of k: For $k = 0, 1, \ldots n$, rank the skus by bang-for-buck $k\,(sp_i - kc_r f_i)$ and choose the top k for inclusion in the forward pick area. Using Model 8.15, compute the total cost of picking and restocking and choose the value of $k = k^*$ for which total costs are minimized.

[1] Technical Note: This process can be sped up significantly by specialized search methods such as Fibonacci search because the cost is unimodal [28].

Note that this is different from the case of minimizing total labor, for which a single ranking by labor efficiency is sufficient (Theorem 8.7. For Equal Space allocations, the ranking and the set of skus selected may will generally depend on the value of k.

For a fixed set of skus stored in the forward pick area, the total number of restocks required under EQT stocking strategy is identical to that required under EQS. Therefore to choose the labor-minimizing set of skus to stock in EQT amounts, we need only take the same ones that are labor-minimizing under EQS and change the amounts stored from EQS to EQT.

Special cases

Sometimes the stocking strategy is subject to further *a priori* constraints. For example, it might be that each sku allocated space in the forward pick area is allocated a predetermined amount of space, such as when there is one standard storage container (bin or tote). In this case, n is fixed, and one need only solve a straightforward knapsack problem.

If the frequency of restocking is fixed in advance, say as 3-weeks supply, then one need only compute the volume of each sku sufficient for a 3-week supply, and then choose the skus with the most expected picks in those 3 weeks, until the forward pick area is filled.

8.4.3 Further comments on the model

Our model has omitted the possibility that one might schedule EQT allocations to even out restocking, and so level the labor requirement, or else concentrate restocking to create opportunities for batching restocks (that is, retrieving in one trip multiple products to be restocked). Might one of these strategies enable EQT allocations to reduce the cost per restock so that total cost of restocking is comparable with OPT allocations? Perhaps, but it is difficult to say.

To be preferred, the total savings from batching restocks under EQT allocations must compensate for the additional restocks incurred, which, from Theorem 8.4 and discussion following, might easily be 2 to 3 times as many as under OPT allocations. Thus, for EQT allocations to be competitive, 2 or 3 skus must be carried together from bulk storage during each trip to restock. But this reduces only one component of the work to restock: travel between the forward pick area and bulk storage. Because multiple locations must be visited within bulk storage and again within the forward pick area, the travel in these areas per trip must increase, and batching k restocks delivers less than k times the efficiency. This means that EQT allocations must enable batching of *more* than 2 or 3 skus per restock to be competitive with OPT allocations. And if batching of restocks is allowed for OPT allocations, EQT allocations would have to batch even more skus, perhaps 5 or 6 or more, to be competitive. This begins to seem impractical if these must all fit on a pallet that is not shrink-wrapped, and in quantities each sufficient to fill a lane of carton flow rack that might be 8–12 feet deep (2.44–3.65 meters). Can such a load be conveyed without toppling?

There is also the question of whether, under EQT allocations, there are sufficiently many skus requiring restocking at the same time to allow significant batching. The timing of restocks is driven by customer orders, over which warehouse management has little control. The flow of a sku can be predicted with much greater accuracy over a year than over a day, and so our model predicts the total number of restocks much more accurately than their timing. Consequently one should not expect all skus with EQT allocations to require restocking simultaneously. It seems reasonable to expect more opportunities to batch restocks under EQT allocations, but unless there is little variability in daily order quantities, the actual opportunities for batching restocks might not be much greater than under OPT allocations.

Of course there are occasions in which EQT allocations make sense, such as for product with very short, common life cycle, in which case the forward pick area might be stocked with just enough of each sku to carry through the selling season. However, we believe that EQT is used most often in the simple, mistaken belief that it reduces restocks compared to EQS.

8.5 Additional issues

Here we extend the basic model in several directions; but the main idea remains unchanged: There is a single number that summarizes the relative claim of each sku or family of skus to prime real estate. Populate that real estate by starting at the top of the list of most labor efficient skus and add them until the net benefit is maximum.

8.5.1 Storage by family

Sometimes it is advantageous, or even required, to store related skus together. Reasons include the following.

- To get good space utilization, product of similar sizes or shapes might be stored together.
- To simplify put-away, product might be stored by vendor so that all product arriving on a truck goes to the same region of the warehouse.
- To simplify put-away at the downstream customer, product might be stored to reflect the layout of the customer's facility. For example, a DC supporting a chain of retail drugstores might store all hair-care products together in the warehouse. Then the hair-care products will be picked and packed together and so will be unpacked together at the retail store, which means they can be put away with less walking.
- To reduce the need for specialized equipment, such as freezers, grocery distributors store product by temperature zone. The standard zones are frozen, refrigerated, and ambient.
- Chemicals typically must be stored by requirement for special handling or storage. Categories include hazardous, flammable, or aerosol.
- Product may be grouped by security requirements. For example, small, high-value items might be stored behind a fence with controlled access.

Call a group of related skus that must be stored together a *product family*. Now we must decide which families to store in the fast-pick area: Either all the skus of a family will be stored in the forward area or else none of them. Let p_{ij} be the number of picks per year of the i-th sku of family j and so the total picks per year of family j is $\sum_i p_{ij}$. By reasoning similar to that of Theorem 8.1 the total space for family j should be $v_j^* = \sum_i v_{ij}^* = \left(\sum_i \sqrt{f_{ij}} / \sum_{ij} \sqrt{f_{ij}}\right)$. But then, in a similar derivation, we should give priority to those families with greatest labor efficiency, now generalized to be

$$\frac{\sum_i p_{ij}}{\sum_i \sqrt{f_{ij}}}.$$

Now we can search over all partitions of families, as before. For example, to decide between storage in the fast-pick area or the reserve areas:

- Sort families from most labor efficient to least.
- Successively evaluate the net cost of putting no families in the fast-pick area; putting only the first family in the fast-pick area; only the first two families; and so on. Choose the strategy that minimizes net cost.

8.5.2 Accounting for safety stock

We have assumed that f_i/v_i is an adequate estimate of the number of restocks, which implicitly assumes that we replenish only after a stock out. In practice we want to be careful to avoid stockouts, which can disrupt order-picking, and so may prefer to carry some safety stock for each sku. If ss_i is the volume of safety stock carried for sku i if it goes in the fast-pick area then we can estimate the number of restocks of sku i as $f_i/(v_i - \mathrm{ss}_i)$. Our previous results (ideal amounts of storage, pick density, labor efficiency) are then a little more complicated but essentially unchanged. For example, the optimal amount of sku i to store, formerly given by Expression 8.5, becomes:

$$v_i^* = \mathrm{ss}_i + \left(\frac{\sqrt{f_i}}{\sum_j \sqrt{f_j}}\right)\left(V - \sum_j \mathrm{ss}_j\right), \tag{8.18}$$

In other words, each sku i in the fast-pick area is guaranteed space ss_i; then any remaining space $V - \sum_j \mathrm{ss}_j$ is partitioned among the skus in the same proportions as we have already seen, that is, by the square roots of their flows.

Storing product in the optimal amounts reduces the number of restocks; and this tends to reduce lead time to restock, which reduces the required levels of safety stock—which *further* reduces the number of restocks or allows more skus into the fast-pick area or both.

8.5.3 Limits on capacity

Consider the problem of choosing skus to go into an fast-pick area: As previously described, we sort the skus by labor efficiency and then successively evaluate the total operating costs as more skus are added to the forward pick area. However, as we add more skus to the fast-pick area, the total picks from and total restocks to the fast-pick area increase. This can be a problem if there are *a priori* limits on total picks or total restocks, such as when the work force cannot be increased (by labor policy; or when picks or restocks are done robotically; or when access to the pick area is limited, such as with carousel conveyors).

Our solution procedure proceeds as before, but if either the pick rate or restock rate exceeds capacity when the k most labor efficient skus are chosen for the fast-pick area, then we know that we can restrict our search to the $(k-1)$ most labor efficient skus.

8.5.4 Accounting for on-hand inventory levels

Imagine some tiny sku, the total supply of which occupies only a small portion of one shelf. It would likely be wasteful to store it in two separate locations, fast-pick and reserve, because it would take so little space to put it all in the fast-pick area and so avoid restock costs. Thus, from among our total population of skus, there will be some that will be stored only in reserve, some that will be stored in both fast-pick and in reserve, and some that will be stored only in the fast-pick area.

How can we tell which skus should not be restocked to the fast-pick area? The following theorem tells us.

Theorem 8.9. *If sku i goes into the fast pick area at all, put all of it if the maximum on-hand volume of sku i is no greater than*

$$2\left(\frac{c_r f_i}{s p_i}\right).$$

The term above will be familiar from Theorem 8.6, which gives us the following interpretation: There should be no separate reserve storage for any sku in the fast-pick area for which maximum on-hand inventory takes no more space than twice its minimum sensible storage amount.

Another use of this result is to quantify the intuition that one should avoid restocking a product that moves swiftly enough through the warehouse. The following is a restatement of Theorem 8.9 in slightly different terms.

Corollary 8.2 (High-turnover skus should not be internally restocked)**.** *Any sku that requires fewer than $2c_r/s$ customer orders on average to turn the warehouse supply should not be restocked; that is, it should be stored either entirely or else not at all in the fast-pick area.*

Proof. Let v be the maximum volume of sku i typically held by the warehouse. If the annual flow of this sku is f_i then the inventory of this sku will turn over f_i/v times a year and will required $p_i v/f_i$ customer orders per inventory turn. Combining this with Theorem 8.9 gives the result. □

Thus, if every sku turns fast-enough, such as in a high-turnover, "Just-in-Time" distribution center then there should be no internal restocking at all.

8.5.5 Setup costs

The model can be enlarged to include setup costs if they are deemed significant. If sku i is already in the forward area, it costs some amount m_i to move it back to reserve; and if it is not in the forward area it costs M_i to move it forward. If sku i is currently in the forward pick area, the net benefit it offers during the next planning period is:

$$\begin{cases} -m_i & \text{if } v = 0 \\ sp_i - c_r f_i/v & \text{if } v > 0 \end{cases} \tag{8.19}$$

And if sku i is not currently in the forward pick area, the net benefit it offers is:

$$\begin{cases} 0 & \text{if } v = 0 \\ sp_i - c_r f_i/v - M_i & \text{if } v > 0 \end{cases} \tag{8.20}$$

Labor efficiency can now be redefined in terms of this enlarged concept of net benefit.

8.5.6 Redirecting uneconomical picks

As for pallets (Subsection 7.2), it is possible to identify individual picks that are uneconomical to fill from the forward area. If each sku is allowed multiple pick locations, both in the forward area and in reserve (Assumption 7.3), then the uneconomical picks are those for which

$$s - c_r f_i/v_i < 0,$$

or where

$$\frac{f_i}{v_i} > \frac{s}{c_r}.$$

In other words, a pick is uneconomical if it is for a volume of sku i that exceeds fraction si/c_r of the volume v_i allocated to sku i. This can be incorporated into the procedure for choosing which skus to assign to the forward pick area.

8.5.7 Multiple fast-pick areas

Suppose there are m fast-pick areas, each with its own economics: savings per pick (compared to from reserve) and restock costs, as suggested by Figure 8.6. Now the question becomes which skus go to which fast-pick areas and which to reserve. Each of the n skus can be picked from at most one area.

Brute-force search can find the best stocking strategy within $O(m^n)$ steps by enumerating all possible assignments of skus to storage modes and then evaluating the result. However, this is completely impractical for realistic values ($m = 2, 3$ and n on the order of tens of thousands).

It can be proven, but is beyond the scope of this book, that

Theorem 8.10. *Skus of greatest labor efficiency should be stored in the fast-pick areas with greatest savings per pick.*

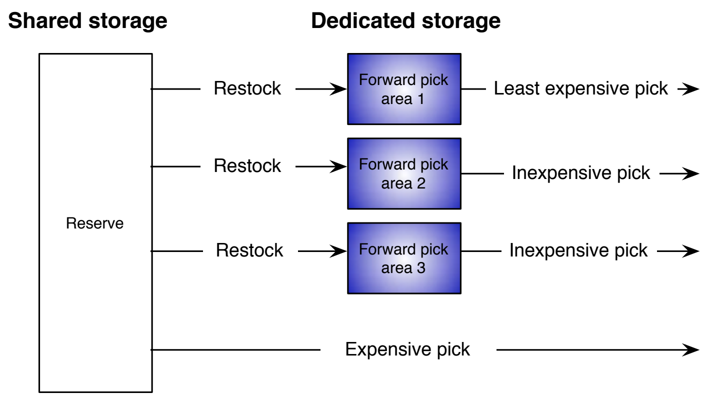


Figure 8.6: Multiple fast-pick areas, such as flow rack and carousel, each with different economics

In other words, some optimal stocking strategy has the following structure: If all skus are ranked by labor efficiency then the top k_1 skus would be assigned to the storage mode with largest savings per pick, the next k_2 would be assigned to the next best storage mode, and so on.

Note that "best storage" depends only on savings per pick and is *independent* of restock costs.

It is further possible to speed up the search for the best partition of skus so that it can be accomplished within $O((\log n)^m)$ steps.

8.6 Limitations of the fluid model

We have analyzed a continuous model that ignores geometries of skus and storage medium. That the continuous model is relevant is guaranteed by the fact that it is the ideal toward which all warehouse managers strive, by sizing shelves and packaging skus to reduce the space wasted by imperfect fit. In any event all three stocking strategies must eventually be converted to discrete allocations—specifications of exactly how many cartons of each sku and how they are arranged. Currently, warehouses approximate Equal Time and Equal Space allocations, and the same can be easily done for Optimal allocations. These approximations will be accurate to the extent that the storage containers are small with respect to the shelves, as might be expected in warehouses distributing service parts, cosmetics, pharmaceuticals, or office supplies.

The predictions of the continuous model were confirmed when we slotted the skus of a pharmaceutical distributor using a discrete model that reflected the geometry of shelving and cartons (a topic for a future paper). The discrete allocations specified exactly how many cartons were to be placed on each shelf, in what orientation, stack height, and how many lanes. The discrete approximations to the OPT allocations required less than half the restocking labor of the discrete approximations to the EQS

Figure 8.7: These plastic liners for the beds of pickup trucks nest like a set of spoons and so can be stored in little more space than that required by a single piece.

allocations.

Finally, it should be noted that there are some special situations in which the fluid model may not be as accurate as desired.

Sub-additivity of space

Some skus resist approximation by the fluid model. For example, auto glass for windshields is curved and therefore is stored in nested fashion. As a result, two windshields occupy only a little more space than a single windshield, not twice the space as implicit in the fluid model. Another example may be seen in Figure 8.7. This sort of complication is more likely to occur with *pieces* rather than with the more regular containers in which the pieces may be shipped or stored.

Granularity of space

The fluid model becomes less accurate when the units of storage are large with respect to the size of the shelves, such as when storing pallets in pallet rack. Another example may be found in flow rack, where each sku must occupy an entire lane, which can be ten feet deep (3.05 meters). In such instances the results of the fluid model may not be directly realizable, but will have to be rounded to the closest allowable amount.

Where space is critical, one must explicitly account for the geometry of storage by considering every reasonable way of storing each sku: which orientation of the case, how high to stack them, and how many lanes to devote. Now, instead of checking fit by simply summing volume, as in the fluid model, we must check fit by checking whether the spatial arrangements of the cases fit on the shelves, for every case, every shelf, and each dimension. This requires a much more detailed model and requires vastly more computation; but it can nevertheless be done with great precision. For example, we slotted over 6,000 skus for a major chain retailer, with the result that required storage in their national distribution center was reduced from 325 to 285 bays of flow rack, and required restocking *still* decreased.

Figure 8.8 shows typical results of the more powerful model, applied in this example to slot skus in a bay of flow rack. Notice that picks are concentrated in the "golden zone" (waist-high shelves) and that the program has determined the exact orientation of each sku, the number of lanes, and how high to stack the cases.

8.7 Size of the fast-pick area

8.7.1 How large should the fast-pick area be?

As the fast-pick area becomes larger, we can fit more skus in, which means more pick savings, or larger amounts, which means less restocking; but we get less savings per pick because of the additional walking.

It is possible to build explicit models of exactly how the pick savings diminishes with increased size of the fast-pick area. For example, suppose we are configuring an aisle of flow rack as our forward-pick area and are undecided about how many bays it should extend. With each additional bay, the pick savings decreases approximately linearly. The rate at which it decreases depends on the economics of each particular warehouse and must be estimated, for example, by time-motion studies. If the forward pick area is to be a single aisle of flow rack, then it seems reasonable to assume a linear model in which adding more bays reduces pick savings at a constant rate S.

Theorem 8.11. *For a linear model of storage the optimum size of the fast pick area is given by*

$$V^* = \left(\sqrt{\frac{c_r}{S \sum_{i=1}^{k} p_i}} \right) \sum_{i=1}^{k} \sqrt{f_i},$$

for some number k of the most labor efficient skus.

Proof. Let the current, default size of the forward-pick area be V_o, for which the pick-savings is estimated to be s_o. As V increases from V_o, the pick savings is assumed to decrease linearly, so that

$$s = s_o - SV.$$

(Note that if V is large enough then all pick savings disappears.)

The net benefit of storing amount v_i^* of each sku i is

$$\sum_i \left(s p_i - c_r f_i / v_i^* \right).$$

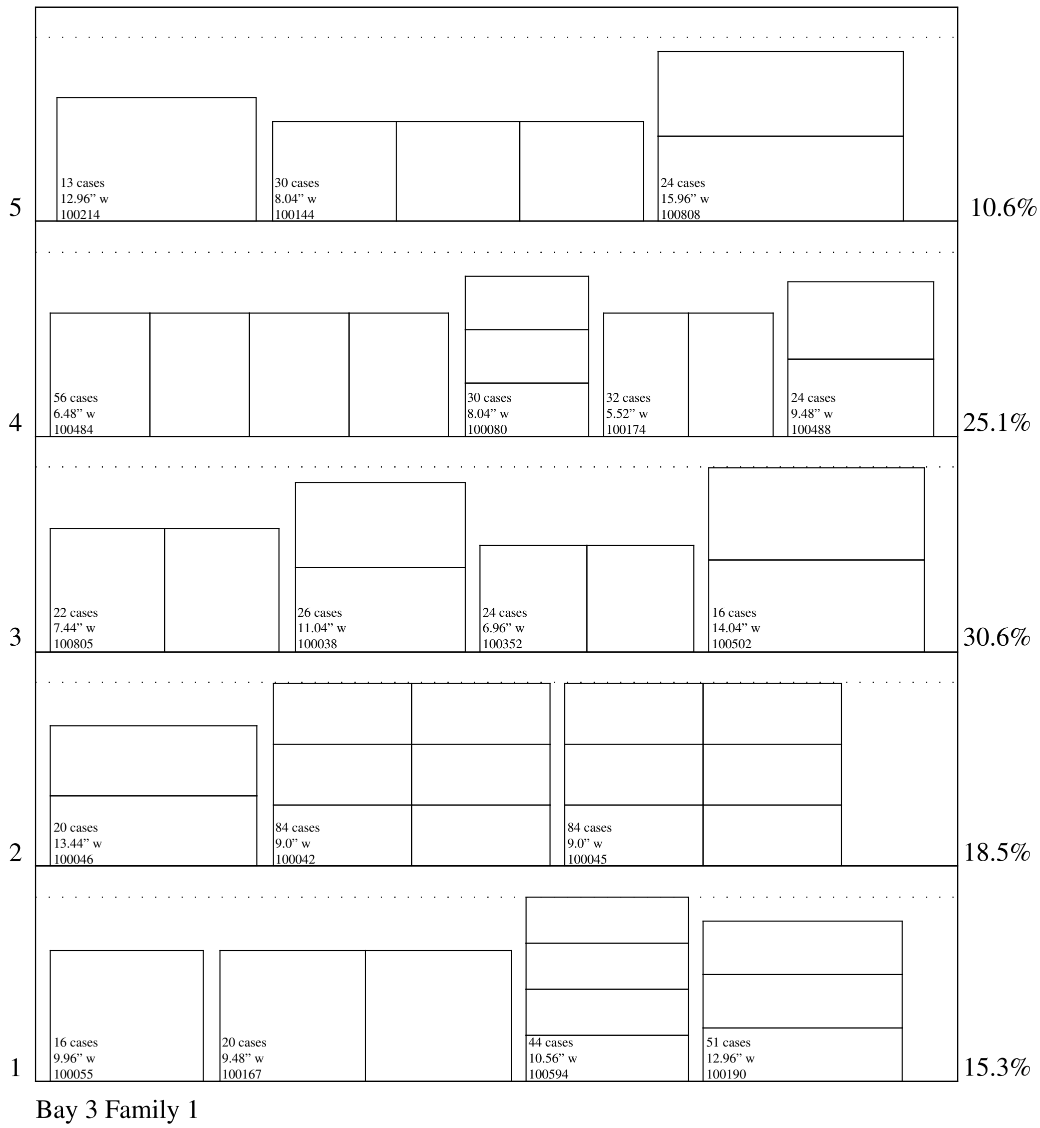


Figure 8.8: Example of slotting that accounts for the geometry of the skus and the storage mode. This pattern of storage minimizes total pick and restock costs for this set of skus and this arrangement of shelves. Numbers on the right report the percentage of picks from this bay on each shelf. (This was produced by software written by the authors.)

Substituting the optimal volumes $v_i^* = \left(\sqrt{f_i} / \sum_j \sqrt{f_j}\right) V$ gives

$$s\left(\sum_i^n p_i\right) - \left(\frac{c_r}{V}\right)\left(\sum_i \sqrt{f_i}\right)^2,$$

which may be rewritten as

$$(s_0 - SV)\left(\sum_i^n p_i\right) - \left(\frac{c_r}{V}\right)\left(\sum_i \sqrt{f_i}\right)^2.$$

Taking the derivative with respect to V, setting to zero and solving for V^* yields the result. □

From this it is straightforward to verify that

$$v_i^* = \sqrt{\frac{c_r f_i}{S\left(\sum_j p_j\right)}}, \tag{8.21}$$

and we use this to allocate the space and compute the net benefit for storing the most labor-efficient skus.

Again we recognize a familiar theme: Sort skus from most to least labor efficient and repeatedly compute allocations for the k most labor efficient skus (using Equation 8.21) together with the resultant net benefit (pick savings minus restock costs). Choose that value of k for which net benefit is maximized. The result is to simultaneously determine the optimal size of the forward-pick area together with the skus to be stored therein and in what volumes.

8.7.2 How can the fast-pick area be made larger?

Almost every warehouse manager would like to increase the size of his or her fast-pick area. Space constraints and cost may make this impractical. Fortunately, one can realize all the benefits of a larger fast-pick area simply by reducing restock costs, as we now argue.

Consider the total net benefit of having stored skus $1 \ldots n$ in a fast-pick area of volume V:

$$\sum_{i=1}^n s p_i - c_r f_i / v_i,$$

which, substituting $v_i = \left(\sqrt{f_i} / \sum_j \sqrt{f_j}\right) V$, may be written as

$$\sum_{i=1}^n s p_i - (c_r / V)\left(\sqrt{f_i} \sum_j \sqrt{f_j}\right).$$

But notice that changing restock costs by a factor of α changes the term (c_r/V) to $(\alpha c_r / V) = (c_r / (V/\alpha))$. In other words, the total net benefit of the fast-pick area is exactly the same, whether the cost-per-restock is halved or the space is doubled. Therefore we observe the following.

Theorem 8.12 ("Law of virtual space"). *Changing the cost-per-restock c_r by a factor of α is economically equivalent to changing the volume of the fast-pick area by a factor of α.*

As a practical matter it is *better* to reduce restock costs because increasing the size of the fast-pick area will in general increase the average cost per pick there and so reduce the savings per pick.

8.8 On the lighter side

More than one logistics manager told us that he too had been concerned that Equal Space Allocation, which was enforced by his warehouse management system, required too much labor to restock the fast-pick area. Each manager had paid a large sum of money to have his warehouse management system revised to support Equal Time allocations — which, as we now know from this chapter, made no difference at all!

8.9 Summary

Almost everyone in industry stocks their forward pick areas with what we believe to be insufficient regard to the labor to maintain them. This may be due to an understandable focus on reducing the work at the front-end (order-picking), which typically consumes more labor than any other warehouse process. Yet, simply by storing product in the right quantities, one can reduce the work to maintain the forward pick area without affecting any other operation. Furthermore, this can be done incrementally by adjusting quantities whenever a sku is restocked.

Each of the strategies for stocking a forward pick area has advantages and disadvantages. Equal Space and Equal Time allocations require the same work to maintain a forward pick area, while Optimal allocations may be expected to require significantly less. Equal Space allocations have the advantage of requiring no knowledge of the size or popularity of the skus; and the uniformity of storage may make it easier to manage skus with short life cycles, such as apparel or cosmetics. To use either Optimal or Equal Time allocations one must forecast the flow of each sku over the planning period. This requires knowing the physical dimensions of the product, which is common in industrialized countries but far from universal. In addition, one must forecast demand and these forecasts are more likely to be most reliable for mature, commodity products.

- Concentrate activity in a small footprint to reduce picking costs, increase responsiveness, and free up space to deal with growth, seasonalities, and other fluctuations.

- The configuration of a warehouse can be optimized based on physical size of the skus and a history of customer orders. To do this you must know the physical dimensions of the storage units and the number of selling units per storage unit.

- The *labor efficiency* of a sku i that is stored in less-than-pallet quantities is $p_i/\sqrt{f_i}$, which measures the work required to pull a given amount of physical volume through your warehouse. The most labor efficient skus are the most suitable for forward storage locations because they generate the largest net benefit (pick savings minus restock costs) for the space they consume.

- For skus restocked in less-than-pallet quantities in the fast-pick area, the restock-minimizing amounts to store, as a fraction of available storage, are

$$v_i^* = \left(\sqrt{f_i} / \sum_j \sqrt{f_j} \right).$$

  Storing skus in the fast-pick area in the optimal quantities will reduce restocks, which can reduce lead time to restock, which allows reorder points to be reduced—which further reduces restocks! In other words, there are secondary savings that accrue automatically to correcting the amounts stored.

  A test of the storage policy at a warehouse is this: At optimality each bay (section, cabinet) of shelving in the forward pick area should be restocked at the same rate. (One can ask the restockers whether they are visiting any part of the fast-pick area especially often or especially rarely.)

8.10 Questions

Question 8.1. *The estimate of the number of restocks as f_i/v_i in the fluid model is only an estimate. Explain how it can be wrong.*

Question 8.2. *Which of the following should be included in the estimated time required to restock a sku in a forward area where pieces are picked from cartons? Explain your answers.*

- *The time required to open each carton.*
- *The time required to load each carton into flow rack*

Question 8.3. *Sku A is kept in a distant region of bulk storage and so it takes 4 times longer to restock it to carton flow rack than it does to restock sku B. They are expected to sell the same volume (cubic meters) of product during the next planning period. Assuming there is one unit of space to be shared by skus A and B in the forward pick area, use the fluid model to tell how much space each should get so that the cost of restocking these two skus is minimized.*

Question 8.4. *Suppose it costs c_i to restock sku i and this cost is particular to sku i and independent of all the other skus. What are the amounts in which the skus should be stored to minimize the total costs of restocking?*

Question 8.5. *Why is shallow storage generally preferable for smaller, slower-moving skus? For example, why might you prefer to put such skus in static rack (bin-shelving), which might be only 2.5 feet deep (0.76 meters), rather than in flow rack that is 10 feet deep (3.05 meters)?*

Question 8.6. *Show that under Equal Time allocations the fraction of forward volume assigned to sku i is*

$$v_i = \left(\frac{f_i}{\sum_j f_j} \right).$$

Question 8.7. *Let $\{v_1^*, v_2^*, \ldots, v_n^*\}$ be the Optimal allocations for skus $1, 2, \ldots, n$ and let $\{v_1, v_2, \ldots, v_n\}$ be the Equal Time allocations for the same set of skus. Prove that the Equal Time allocations use a greater range of storage sizes; that is, $\max_i v_i^* - \min_i v_i^* \leq \max_i v_i - \min_i v_i$.*

Question 8.8. *Consider a distribution center that alternates 2-hour intervals of order-picking with 1/2-hour periods of restocking. Suppose that a given set of skus has been chosen for storage in the forward pick area and each sku i must be stored in volume* at least l_i *to avoid stockouts during an order-picking cycle. Is the following method for allocating space correct? (Explain your answer.) Method: Reserve volume l_i in the forward pick area for sku i and then allocate any remaining space among all the skus according to Expression 8.5.*

Question 8.9. *The reorder point for sku i in the forward area triggers restocks. It is the sum of its mean lead-time demand $\overline{ltd}_i$ plus safety stock ss_i.*

Consider the following three skus that are to be stored within 80 units of volume in the forward pick area. (All data is given in common units of volume.)

Sku	***Flow***	ss_i	$\overline{ltd}_i$
A	*90*	*2*	*10*
B	*250*	*10*	*10*
C	*490*	*28*	*15*

How much space should be allocated to each sku so that the total number of restocks is minimized? (Hint: Refer to Section 8.3.5.) How many restocks result?

Question 8.10. *Sku A is requested ten times as often as sku B but has one-half the flow. Assuming both go into the fast-pick area, what relative amounts of space should be allocated to each?*

Question 8.11. *A. Suppose you load a forward pick area with n skus and store them in Equal Space allocations. Subsequently you re-slot, adding another n new skus to the original population and re-adjusting quantities, again to Equal Space allocations. Which of the following claims are most nearly correct?*

- *The total number of restocks of each sku i will double.*
- *The total number of restocks of each sku i will increase but may not double.*
- *The total number of restocks to all skus will* more than *double.*
- *The total number of restocks to all skus will increase only if the new skus each have as much flow as the original skus.*
- *The total number of restocks to all skus might actually decrease, depending on the flows of the new skus.*
- *It is impossible to say how the total number of restocks will change.*

B. How would your answer to the above change if, when adding the additional skus, the storage strategy is changed to Equal Time allocations?

Question 8.12. *Assume that you have set up a fast-pick area in flow rack and stocked it optimally. Recently the bulk storage area has been redesigned and it now takes* longer *to carry replenishment stock from bulk storage to the fast-pick area. If you were to re-compute the optimal slotting you would find which of the following? Explain.*

- *Some skus would leave the fast-pick area and the remaining skus would get more space.*
- *The skus and amounts stored in the fast-pick area would remain unchanged.*
- *New skus would join those already present and each of the skus in the fast-pick area would get less space.*
- *A completely new mixture of skus may be selected.*
- *The flow rack will overflow.*

Question 8.13. *Consider sku A, which was picked 10,000 times last year, and sku B, which was picked 100 times. Which has greater claim to storage in the fast-pick area or is it impossible to tell? Explain.*

Question 8.14. *Red widgets were requested 10 times as often as blue widgets last year and in 10 times the volume. Which has greater claim to storage in the fast-pick area or is it impossible to tell? Explain.*

Question 8.15. *A warehouse distributes two types of pen that are identical except for color. Customers prefer the black pen: It was picked 1,500 times last year and averaged 4 eaches per pick. In comparison, the green pen was picked 800 times per year and each pick was for a single each. Which pen has stronger claim to storage in a fast-pick area and why?*

Question 8.16. *Sku i had annual picks and flow of p_i and f_i respectively. Sku j has been in the distribution system for only 3 months, during which it was picked p_j times, with flow of f_j. Do $p_i/\sqrt{f_i}$ and $p_j/\sqrt{f_j}$ accurately reflect the relative claims of sku i and j to storage in the fast-pick area? Explain.*

Question 8.17. *Suppose that a sku is repackaged into smaller cases that hold 100 units, rather than the 250 units that the previous, larger cases held. Has the suitability of this sku for storage in the forward area increased or decreased or remained the same or is it impossible to tell? Explain.*

Question 8.18. *Each drugstore of a retail chain is assigned one day in the week at which to submit its order to the regional distribution center. Each store typically includes a request for at least several bottles of popular brand of shampoo. Suppose that this shampoo, currently in the forward pick area, is replaced by a concentrated version of the same product, packaged in exactly the same way but now each bottle lasts twice as long as before. Has this sku become more or less suitable for storage in the fast-pick area or remained the same or is it impossible to tell? Explain.*

Question 8.19. *Suppose that a marketing campaign has just begun for one of the more popular skus. This campaign does not attract any new customers but it is successful in getting regular customers to buy in greater quantities than previously. Has this sku become more or less suitable for storage in the fast-pick area or remained the same or is it impossible to tell? Explain.*

Question 8.20. *True or false, and why?*

- *A high-volume sku (that is, one with a large value of flow) would be a strong candidate for storage in a forward pick area that will be stocked to minimize restocks.*
- *A high-volume sku would be a strong candidate for storage in a forward pick area that will be stocked by Equal Space allocations.*
- *A high-volume sku (large value of flow) would be allocated more space than a low-volume sku if in the forward pick area that was stocked to minimize restocks.*

- *A high-volume sku will generally require more restocks under Equal Space allocations than under Equal Time or Restock-Minimizing allocations.*

- *The optimal set of skus to be stored under Equal Space allocations is generally larger than the optimal set of skus if stocking to minimize restocks.*

Question 8.21. *Assume that you have set up a fast-pick area in flow rack and stocked it optimally. Later you add pick-to-light capability, which increases the pick rates. (Everything else, including restock costs, remain unchanged.) If you were to re-compute the optimal slotting you would find which of the following? Explain.*

- *Some skus would leave the fast-pick area and the remaining skus would get more space.*

- *The skus and amounts stored in the fast-pick area would remain unchanged.*

- *New skus would join those already present and each of the skus in the fast-pick area would get less space.*

- *A completely new mixture of skus may be selected.*

Question 8.22. *Suppose you enlarge your fast-pick area slightly. Assuming the average cost-per-pick has not changed significantly, which of the following will you find after recomputing the optimal population of skus for the fast-pick area?*

1. *Some of the skus currently in the forward area may be moved out and replaced by an assortment of other skus.*

2. *All the skus currently in the forward area will remain but they will be joined by additional skus.*

3. *No new skus will be moved into the forward area; instead, the most labor efficient will remain and each will get more space.*

4. *It is impossible to tell.*

Question 8.23. *If business increases for a chain of grocery stores, it will likely be manifest in increased volumes sold of each sku; but for an e-grocer that delivers to individual customers, an increase in business will likely be manifest in more customers. What is the effect of increased business on a forward-pick area in the grocery distribution center? On the forward-pick area in the distribution center of the e-grocer? Explain.*

Question 8.24. *Which of the following costs are properly included in the cost per restock of a sku in the forward pick area? Explain in each case.*

- *The time to travel between the forward pick area and reserve storage*

- *The time to pick up the cartons from reserve storage and place them on a pallet for transport to the forward pick area*

- *The time to pick up the cartons from the pallet and place them in forward storage*

- *The time to cut open the cartons so that eaches can be extracted by the order-pickers*

Question 8.25. *We have implicitly assumed that each restock costs the same. Is this correct? Discuss whether and how restock costs might depend on*

- *Location of sku in fast-pick area*
- *Location of sku in reserve area*
- *Amount to be restocked*

Question 8.26. *Suppose that the skus have been chosen and amounts decided for storage in a forward pick area. Even within a forward pick area, some locations may be more convenient than others; for example, locations in the "golden-zone" are more convenient than those for which a person must bend or stoop. What skus should be placed in the most convenient forward locations?*

Question 8.27. *Show how to choose skus and allocate space in a forward pick area even when each sku has its own pick costs and restock costs that assume values particular to it.*

Question 8.28. *The following is a miniature of a real problem: In what amounts should skus be stored in a forward pick area to minimize labor?*

Suppose you have 10 cubic feet available in flow rack, which is restocked from a distant reserve area, and you have just added three skus, with projected activity as follows.

sku	picks/month	pieces/month	pieces/case	ft^3/case
A	*1000*	*2000*	*200*	*2*
B	*300*	*1200*	*6*	*7*
C	*250*	*4000*	*10*	*1*

A. Suppose you have decided to put all three skus in flow rack. How much space should be allocated to each sku to minimize the labor devoted to them?

B. How often must each sku be restocked?

C. How many restocks would each sku incur if allocated an equal share of the space?

D. How many restocks would each sku incur if allocated equal time supplies?

Question 8.29. *A. Reconsider Question 8.28, part A, supposing that no more than 5 cubic feet may be allocated to any single sku. How would your answer change?*

B. Generalize your answer to part A: Devise a procedure to handle arbitrary upper bounds on allocations. Do the same for lower bounds.

Question 8.30. *A. Reconsider Question 8.28 supposing you may choose any subset l of the three skus to put in the flow rack. Which should they be? What will be the net cost of including the ones you have chosen?*

Assume that it costs an average of \$0.15 per pick from flow rack but costs about \$1/restock. The alternative is to pick from reserve, where each pick costs about \$0.25.

B. Suppose that you must rent *the space in flow rack from another department within your organization. How much rent would you be willing to pay for the use of this area?*

C. How much would it be worth to you if sku B was reformulated as a "concentrate" so that exactly the same amount of product could be fit into a much smaller case, occupying only 1 cubic foot?

Question 8.31. *In the basic fluid model, if a sku is stored in the fast pick area, economics suggests that it ought to be stored in at least what amount (volume)? Choose the best answer and explain why it is so.*

- *0*
- $c_r f_i / (sp_i)$
- $sp_i^2 / (4c_r f_i)$
- *All of the sku that is stored in the warehouse*

Question 8.32. *In Section 8.5.1 it was claimed that the correct value for the labor efficiency of family j is computed from the skus i within that family as follows:*

$$\frac{\sum_i p_{ij}}{\sum_i \sqrt{f_{ij}}}.$$

Prove that this is correct. In particular, explain why it would be incorrect to use

$$\frac{\sum_i p_{ij}}{\sqrt{\sum_i f_{ij}}}.$$

Question 8.33. *Follow the suggestion of Section 8.5.2 and derive the formulae for optimum allocations when each sku i has a non-zero reorder point ss_i. Do the same for labor efficiency.*

Question 8.34. *Is the following true, false, or impossible to determine? (Explain your answer.) Suppose the reorder point of a single sku in the fast-pick area is increased. Then, in the optimal allocation of space to minimize restocks in the fast-pick area,* every *sku must be restocked more frequently. (Assume that the population of skus remains unchanged.)*

Question 8.35. *Suppose that the reorder point of a sku (the level at which a request for restocking is issued) has been increased due to variability in customer demand. Has that sku a greater or lesser claim to storage in the fast-pick area, or neither? Explain.*

Question 8.36. *We have argued that the "best" storage modes are those with the smallest pick costs. What if the "best" mode has been placed far from reserve and so each restock is quite expensive? How can it make sense that the "best" skus should be stored here? How might the high cost of restocking affect the assignment of skus to modes?*

Question 8.37. *Consider a warehouse with multiple fast-pick areas. Which fast-pick area will get the most labor efficient skus?*

1. *The fast-pick area with the greatest savings-per-pick.*
2. *The fast-pick area with the least cost-per-restock.*
3. *The fast-pick area with the largest ratio of savings-per-pick to cost-per-restock.*
4. *The fast-pick area with the greatest volume.*
5. *The fast-pick area with the least volume.*

Question 8.38. *The savings in restocks realized by storing skus in their optimal amounts (rather than Equal Space or Equal Time allocations) is likely to be most significant when space is tight. Why is this?*

Question 8.39. *Make a numerical example that shows that sequencing skus by labor efficiency is not the same as sequencing by picks divided by flow. In other words, find values for which* $p_i/\sqrt{f_i} < p_j/\sqrt{f_j}$ *but* $p_i/f_i > p_j/f_j$.

Question 8.40. *Consider three ways of stocking the fast-pick area: Equal Space allocations, Equal Time allocations, and Optimal allocations.*

- *Rank the methods according to variability among the amounts of space allocated.*
- *Rank the methods according to variability among the numbers of restocks required for each sku.*

Question 8.41. *Consider a collection of skus that are candidates for storage in a fast-pick area composed of carton flow rack. Anything not stored in the fast-pick area will be picked from pallets in reserve.*

Assume that for a particular sku i *we know the number* p_i *of piece picks, the piece flow* f_i*, the number* p'_i *of full-carton picks, and the carton flow* f'_i*. (Piece flow is the total volume of product that was picked as individual pieces; carton flow is the total volume of product that was picked as full-cartons.)*

A. Explain how to use the methods of this chapter to decide whether it is better to pick full-cartons of sku i *from flow rack or from reserve. (Ignore consolidation costs.)*

B. Explain how to use the methods of this chapter to decide on a sku-by-sku basis *whether sku* i *should be stored in the fast-pick area and if so, how much space it should be allocated and whether carton picks of sku* i *should be from its forward location or from its reserve location. (Thus each sku may be stored in one of three configurations: No storage in the fast-pick area and all picks are from reserve; or: storage in the fast-pick area but only piece picks are from fast-pick and carton picks are from reserve; or: storage in the fast-pick area and both piece picks and carton picks are from fast-pick.)*

Question 8.42. *A. Suppose that the fast-pick area is supported by two bulk reserves. One of the reserves is off-site and both picks and restocks from this site cost more than restocks from the other. Each sku has been assigned to a single reserve and you must* *choose the skus and quantities to be stored in the forward area. How can you adapt the model and solution?*

B. How can you determine which skus ought to be assigned to which bulk reserve?

Question 8.43. *Suppose that the bulk storage area is connected to the fast-pick area by a conveyor. How can you adapt the model to choose the best set of skus and quantities for the fast-pick area while accounting for the maximum throughput of the conveyor?*

Question 8.44. *The following set of small parts are stored and picked as eaches.*

Sku	***Picks***	***Qty***	***Length***	***Width***	***Height***
A	*2,500*	*4,500*	*0.50*	*1.0*	*0.5*
B	*1,000*	*1,234*	*0.25*	*1.5*	*0.5*
C	*500*	*750*	*0.60*	*0.8*	*1.0*
D	*50*	*90*	*1.00*	*0.5*	*0.5*
E	*250*	*1,000*	*0.60*	*0.6*	*0.6*
F	*100*	*250*	*2.00*	*1.0*	*1.0*
G	*800*	*850*	*1.00*	*1.0*	*1.5*
H	*400*	*800*	*0.40*	*0.4*	*0.8*
I	*200*	*400*	*0.10*	*0.3*	*0.1*

Using the fluid model, rank the skus according to the strength of their claim for storage in a forward pick area. Give the value of the labor efficiency of each.

Question 8.45. *Suppose you are considering buying flow rack to hold the skus of Question 8.44. The labor economics are estimated to be as follows.*

Storage mode	***Physical volume***	***Cost/pick***	***Cost/restock***
Flow rack	*1.0*	*1.0*	*5.0*
Reserve	*Effectively unbounded*	*8.0*	*0.0*

Use the fluid model to answer the following. What is the total operating cost if the best k skus are stored in flow rack ($k = 0, \ldots, 9$)? Which value of k gives the minimum? Which skus should be picked from flow rack and which from reserve? In what quantities should skus be stored in flow rack? How many picks are expected from each storage mode? How many restocks to the flow rack? What is the total expected operating cost (picking costs plus restocking costs)? What is the net labor savings expected of the flow rack when stocked to minimize labor?

Question 8.46. *Continuing the analysis of Question 8.45, use the fluid model to evaluate shelving as an alternative to flow rack. Shelving is cheaper than flow rack and so you can afford twice the storage volume; but the labor economics are not as attractive.*

Storage mode	*Physical volume*	*Cost/pick*	*Cost/restock*
Shelving	*2.0*	*2.0*	*8.0*
Reserve	*Effectively unbounded*	*8.0*	*0.0*

What is the total operating cost if the best k skus are stored in shelving ($k = 0, \ldots, 9$)? Which value of k gives the minimum? Which skus should be picked from shelving and which from reserve? In what quantities should skus be stored in shelving? How many picks are expected from each storage mode? How many restocks to shelving? What is the total expected operating cost (picking costs plus restocking costs)? What would be the net labor savings expected of the shelving? Is shelving more or less attractive than flow rack in this instance?

Question 8.47. *Continuing the analysis of Question 8.46, use the fluid model to evaluate the option to purchase* both *shelving and flow rack. From which storage mode should each sku be picked? In what quantities should it be stored? What would be the net benefit of having both types of storage mode? If you had already decided to buy the flow rack, how much additional value would you expect to get from the shelving?*

sku	**Picks**	**Flow**
A	100.0	10.0
B	75.0	7.5
C	73.0	5.0

Table 8.1: Questions 8.48–8.50 refer to these skus, which are candidates for storage in a forward piece-pick area for small parts. (Picks are given as daily averages and flows have been scaled as a fraction of the volume available in the forward pick area.) Assume that the savings per pick in the forward area is 1 person-minute and that the cost per restock averages 3 person-minutes.

Question 8.48. *Consider the skus of Table 8.1. Assume that you have decided to stock the forward pick area with the primary objective of minimizing total picking plus restocking costs. Answer the following questions and explain your reasoning in each case.*

A. If A is stored forward then which other skus should be as well?

B. If B is stored forward then which other skus should be as well?

C. If C is stored forward then which other skus should be as well?

D. If all three skus were stored in the forward pick area, which would be allocated the most space?

Question 8.49. *Consider again the set of skus described in Table 8.1. Assume that you have decided to stock the forward pick area with Equal Space allocations and now undertake to choose the best skus so that total picking plus restocking costs are minimized. Answer the following questions and explain your reasoning in each case.*

A. What is the best single sku to store forward?

B. If you are going to store two skus, which should they be?

C. What is the labor-minimizing subset to store in the forward pick area managed by the EQS strategy?

D. How much space should be allocated to each under the labor-minimizing EQS strategy?

E. What is the total net benefit under the labor-minimizing EQS strategy?

Question 8.50. *Consider again the set of skus described in Table 8.1. Assume that you have decided to stock the forward pick area with Equal Time allocations and now undertake to choose the best skus so that total picking plus restocking costs are minimized. Answer the following questions and explain your reasoning in each case.*

A. If all three skus were to be stored in the forward pick area, which would be allocated the most space?

B. What is the labor-minimizing subset to store in the forward pick area managed by the EQT strategy?

C. How much space should be allocated to each under the labor-minimizing EQT strategy?

D. What is the total net benefit under the labor-minimizing EQT strategy?

Question 8.51. *The following skus are candidates for storage in a forward piece-pick area for small parts. Picks are given in person-minutes and flows have been scaled as a fraction of the volume available in the forward pick area. Assume that savings per pick from the forward pick area is 1 person-minute and a restock averages 3 person-minutes.*

sku	*Picks*	*Flow*
A	*80*	*12.0*
B	*90*	*7.9*
C	*75*	*6.0*
D	*85*	*7.0*

A. Under allocations that minimize restocks, what is the labor minimizing subset of skus to be stored in the forward pick area? How much space should be allocated to each sku? How many picks and restocks would there be expected from/to the forward pick area? What is the expected total net benefit?

B. Under Equal Space allocations what is the labor-minimizing subset of skus to be stored in the forward pick area? How much space should be allocated to each sku? How many picks and restocks would there be expected from/to the forward pick area? What is the expected total net benefit?

C. Under Equal Time allocations what is the labor-minimizing subset of skus to be stored in the forward pick area? How much space should be allocated to each sku? How many picks and restocks would there be expected from/to the forward pick area? What is the expected total net benefit?

Question 8.52. *Figure 8.9 shows that, as expected, OPT allocations achieve greater net benefit than do EQS or EQT allocations. (The maximum is marked by a •.) But*

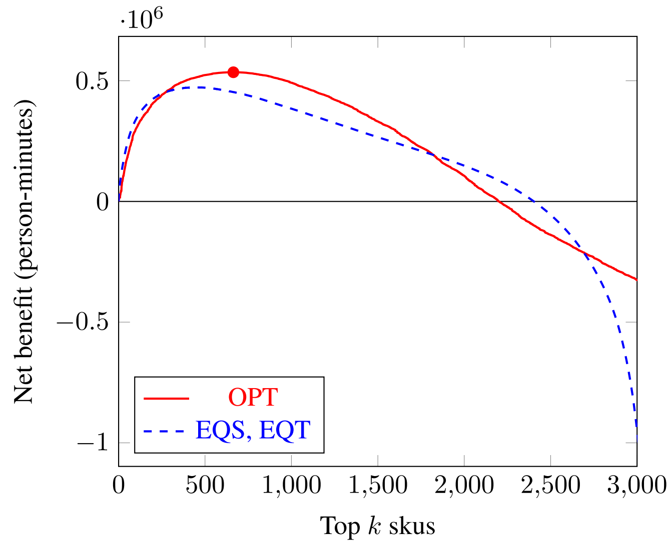


Figure 8.9: Net benefit from storing the best k skus in a forward area by either the OPT allocations or else EQS (equivalently, EQT) allocations

how is it that EQS allocations ever beat OPT allocations, as in this case, when there are only a few skus or else many skus in the forward area? Does this contradict the model?

Question 8.53. *The* cube-per-order index *of sku i is volume of space v_i allocated to storing that sku divided by the number of picks p_i, or the inverse of pick density [30, 34]. What is the cube-per-order index of skus that will be stored in quantities that allocate space equally? That allocate equal time supplies? That minimize restocks?*

Question 8.54. *Some forward pick areas have a natural upper limit on the pick rates they can support. For example, a carousel serviced by one person should not be loaded with skus likely to require more than a few hundred picks per hour, because a person will have trouble sustaining that rate. How can the choice of skus to be stored forward account for limits on the total number of forward picks? For limits on the total number of restocks?*

Question 8.55. *The Walgreens DC in Anderson, SC picks includes a collection of aisles in which pieces are picked from flow rack. Each sku in the flow rack is stored in a "tub" (a plastic tray of uniform size). All restocking is done by an automated device that retrieves one tub at a time from overstock, carries it to the assigned location, and inserts it into the back of the flow rack. Which of the storage policies—Equal Space, Equal Time, or Restock-Minimizing—would be most appropriate in this case?*

Chapter 9

Detailed slotting

Slotting refers to the careful placement of individual cases within the warehouse. It may be thought of as layout in the small.

The most immediate goals in slotting a warehouse are the following.

- Squeeze more product into available space; and
- Achieve ergonomic efficiency by putting popular and/or heavy items at waist level (the "golden zone", from which it is easiest to pick).

At the same time, one wants to avoid creating congestion by concentrating popular items too much.

There are additional concerns that vary from warehouse to warehouse. For example, some distribution centers supporting retail drug stores prefer to store similar-looking medicines apart to reduce the chance of a picking error; but they store non-drug items in the same product family so that, for example, all the hair care products will tend to be picked together. This means that, when the retail store opens the shipping container, they will find it packed with items that are to be stored on the same aisle, so that put-away requires less labor. Storing products in the warehouse by product family may cost the warehouse more in space and labor but it may save put-away labor in thousands of retail stores.

9.1 Case orientation and stack level

Space issues are most significant for skus that are stored in less-than-pallet quantities. The storage unit of many such skus is a carton or cardboard case, which is a rectangular solid. Such a shape may be placed on a shelf in any of up to six orientations. Furthermore, once an orientation has been selected, the shelf space above and behind a case of that item are rendered unusable by other items, because to store another sku there would create unnecessary work to retrieve the sku behind, as shown in Figure 9.1. Consequently one might as well fill up that space as much as possible with additional cases of the same sku, as in Figure 9.2. We refer to such arrangements as "maximal"

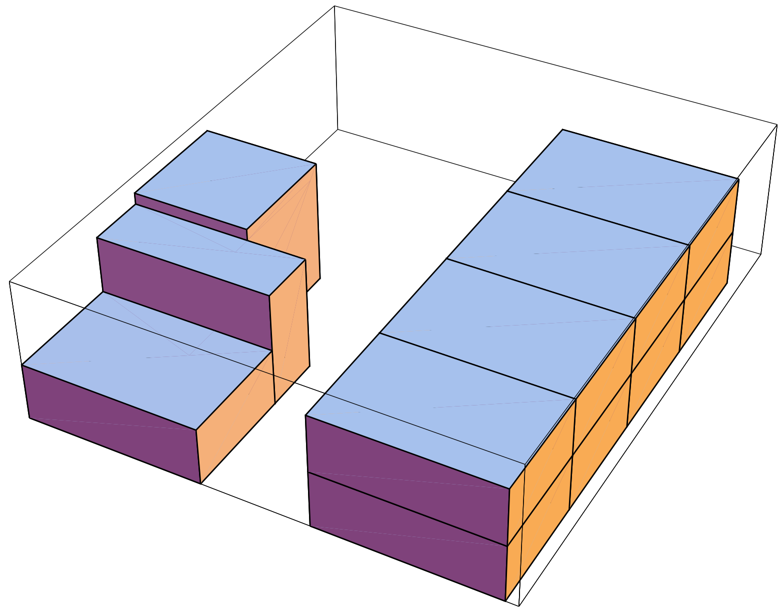


Figure 9.1: Storing an item so that it blocks access to another creates useless work and increases the chance of losing product.

{H, D, W}:	{1,2,3}	{2,1,3}	{3,1,2}	{1,3,2}	{3,2,1}	{2,3,1}
Efficiency:	0.76	0.57	0.86	0.86	0.76	0.57

Table 9.1: Local space efficiencies of each orientation of a $1 \times 2 \times 3$ carton stored in a shelf opening of height 3.5 and depth 9. (H = Height, D = Depth, W = Width)

because they attempt to fill all the space above and behind that is otherwise rendered unusable by other skus.

Note that alternative arrangements of the same sku may differ in the amount of space wasted above or behind. We can evaluate the local space efficiency of an orientation by the ratio of space actually occupied to the space rendered unusable by other skus. For example, consider cartons of dimensions $1 \times 2 \times 3$, which are to be stored on a shelf with opening of height 3.5 and depth 9. Then the (local) space efficiencies of the orientations are as given in Table 9.1.

These numbers are computed as volume of product stored divided by the volume of storage space that is directly rendered unusable by other skus. For example, the orientation {1,2,3} results in local space efficiency of $(12 \times 6)/(3.5 \times 9 \times 3) \approx 0.76$.

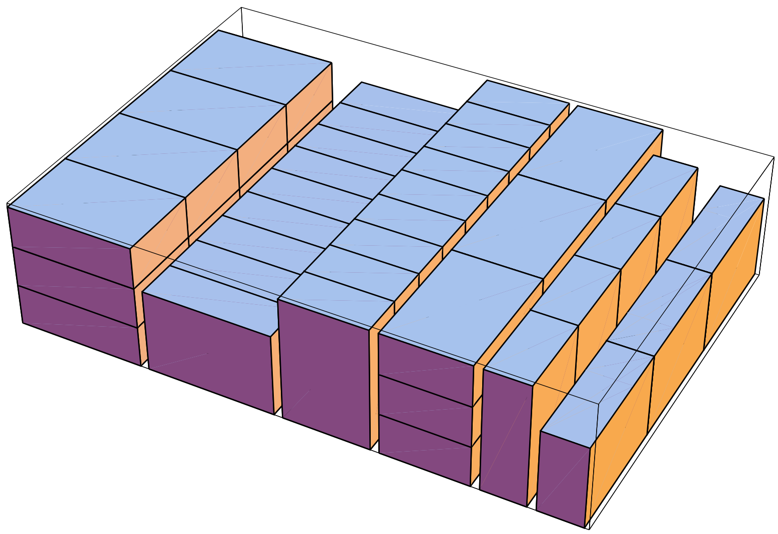


Figure 9.2: Each storage unit may be placed in any of up to six orientations; and each lane might as well be extended to the fullest depth possible and to the fullest height.

9.2 Packing algorithms

Once the skus to be slotted are known, together with the quantity of each, then there remains the problems of packing the cartons onto the shelves and then shelves into bays. First we pack cartons onto shelves to minimize the unused horizontal space; then we pack shelves into bays to minimize the unused vertical space. Both of these problems have the same general structure and so we will talk mostly about the first.

To focus on the essential ideas of packing, we make a few simplifying assumptions. First, assume that we are slotting n skus and that the quantity and orientation of each has been decided. These skus are to be slotted into shelf openings that are all of identical and fixed dimension and no sku is to be allocated more than a single shelf. Our goal is to use the fewest shelf openings possible to hold all the skus.

Because orientation has already been decided and the shelf dimensions are fixed, our main concern is with the horizontal space along the shelf. For us, a shelf is filled if the width of cartons assigned to that shelf fill the width of the shelf opening, as shown in Figure 9.3. Note that the width of a sku allocation is the width of all its cartons plus whatever gap might required between adjacent lanes of cartons.

This problem is generally abstracted as a *1-dimensional bin-packing problem*: Given n numbers $w_1, \ldots, w_n$, with each $0 < w_i \leq 1$, arrange them into as few groups ("bins") as possible such that each group sums to no more than 1. In this case, we may take the width of the shelf opening to be normalized to 1, and the w_i are the normalized widths of the oriented allocations assigned to each sku in the forward pick area.

An obvious lower bound to the solution is $\sum_{i=1}^{n} w_i$.

The bin-packing problem is NP-complete but there are many heuristic solutions

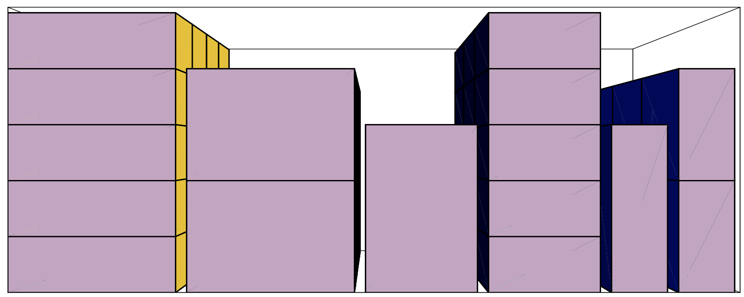


Figure 9.3: Once quantities and orientations of skus have been established there remains the challenge of fitting them all in as few shelves as possible.

that perform quite well in practice. The most commonly-used packing algorithms all have the same general logic: Build a sorted list of the skus, and iteratively remove the next sku from the list and pack it onto the shelf most suitable for it. Once a sku has been placed on a shelf by this type of algorithm, that placement is never reconsidered.

As we will see, the simplicity of this family of packing algorithms make them easy to adapt to special circumstances of each warehouse. For example, one can affect the behavior of the heuristic by choosing how to sort the skus of the list.

9.2.1 Next Fit

The algorithm `Next Fit` directs its attention to one shelf at a time. It removes the next sku from the sorted list and tries to fit it onto the current shelf. If the sku fits, it is added and the process continues; otherwise the current shelf is deemed full and then closed and never reconsidered. A new, initially empty shelf becomes the current shelf and the process continues.

This heuristic obviously wastes some space because it never reconsiders a shelf after a sku has failed to fit. Fortunately it can not waste "too much" space.

Theorem 9.1. *Slotting skus by* `Next Fit` *never require mores than twice as many shelves as the minimum possible.*

Proof. Consider the list of shelves after the heuristic has packed the skus: Any two shelves that are adjacent in the list must contain skus that fill or overflow a single shelf. □

To see how `Next Fit` can be fooled, consider a list of $2n$ skus with allocations of width 0.5 and together with another $2n$ of width $\epsilon < 1/(2n)$, to be packed onto shelves of width 1. Packing via `Next Fit` from the list $\{0.5, \epsilon, 0.5, \epsilon, \ldots, 0.5, \epsilon\}$ requires $2n$ shelves; but the packing that pairs skus with allocations of width 0.5 and places all skus with allocations of width ϵ on one shelf requires only $n + 1$ shelves.

The `Next Fit` heuristic produces a packing in which the skus appear in the shelves in exactly the same sequence as in the sorted list, which is sometimes useful. But the main use of this result is as a sort of worst-case: All the remaining packing algorithms are more careful about wasting space and so pack at least as well.

9.2.2 First Fit

The heuristic `First Fit` achieves more space-efficient packing by keeping every partially-loaded shelf open and available to receive more skus. `Next Fit` removes the next sku from the sorted list and tries to fit it on a shelf, but here is the enhancement: `First Fit` tries the sku on *each* partially-loaded shelf, in order, and puts it on the first shelf on which it fits. If it does not fit on any open shelf, `First Fit` opens a new, initially empty shelf, puts the sku there and continues.

Under `First Fit`, the first shelves to be opened are considered for every subsequent sku; consequently the first shelves tend to be packed very tightly and the last shelves to be opened are packed less well.

Theorem 9.2. *Let s^* be the minimum number of shelves required to hold a given set of allocations. Then slotting skus by `First Fit` never requires more than*

$$\left(\frac{17}{10}\right) s^* + 2 \textit{ shelves}.$$

The proof of this is too long to include here but may be found in [32].

In practice `First Fit` performs significantly better than `Next Fit`. Furthermore, it may be made to pack with still greater space-efficiency by sorting the skus from widest to narrowest allocation. This follows the general rule about achieving space-efficiency: Pack the larger items first and fill in the remaining space with small things. With this addition the heuristic is known as `First Fit Decreasing` and it is guaranteed to be within about 22% of the maximum space efficiency.

Theorem 9.3. *Let s^* be the minimum number of shelves required to hold a given set of allocations. Slotting skus by `First Fit Decreasing` never requires more than*

$$\left(\frac{11}{9}\right) s^* + 6/9 \textit{ shelves}.$$

The proof of this is too long to include here but may be found in [19].

9.2.3 More on packing algorithms

It is possible to strengthen `First Fit`. For example the heuristic `Best Fit` slightly extends `First Fit` by storing each sku on the fullest shelf on which it fits. `Best Fit` seems better in principle than `First Fit` but has the same worst-case bound and, moreover, is not observed to perform any better on average. A theoretically interesting variation is `Worst Fit`, which stores the next sku on the emptiest shelf on which it fits.

More sophisticated extensions have been made by tailoring the algorithm to the distribution of sku widths. Generally, such extensions are expected to provide tighter packing but are more complex and special purpose. No one knows an algorithm that is both fast (runs in polynomial time) and guaranteed to use the fewest shelves possible. Of course one can try every possible way of storing the skus but this would require hugely impractical amounts of time.

The bounds given by the theorems are *worst-case* bounds. In practice the heuristics perform much better. These heuristics may also be expected to pack skus efficiently because the typically large number of skus ensures that heuristics such as First Fit have many opportunities to top off any partially-packed shelves.

These heuristics are simple and fast and produce efficient packings; but sometimes they can behave in ways that are counterintuitive. For example, one would expect these heuristics to slot with greater space-efficiency when working from a list of skus sorted from allocation of greatest to least width. This is generally true, but not always! Furthermore it can happen that removing a sku forces the heuristic to use *more* shelves.

Much of the early work on packing was done by R. Graham in the context of machine scheduling; a good summary of his work can be found in [25]. More technical details, extensions, and pointers to related work can be found in [33]

The simplicity of these list-processing heuristics make it easy to adapt them to handle other complications. For example:

- If you want to use as few shelves as possible: Pack from a list in which the skus have been sorted from greatest width of allocation to least.

- If you wish to concentrate picking: Pack from a list in which skus have been sorted from most picks per width of allocation to least.

- If there are shelves of different heights: Put each sku on the least-high shelf on which it will fit.

- If skus must be stored in predefined groups (such as all hair care products together): When a new shelf is opened, designate it to hold only those skus in the same group as that sku for which the shelf was opened. Thereafter place the next sku in the appropriate shelf from among those designated for its group.

- If the orientation of the skus has not been decided in advance: One can adapt a variant such as `Best Fit` to, for example, choose the orientation and the open shelf that leave the least shelf space remaining.

9.3 Other issues

It is frequently important to group skus according to additional criteria, one of the most important of which is ergonomic efficiency. To make it easy on human order-pickers, store the skus so that the most picking takes place between waist and chest levels. (This is sometimes referred to as *golden-zoning*.) Similarly, heavy or hard-to-handle items should be at waist level.

Also, storing the heavier items at the start of the pick-path means that they will be placed at the bottom of the tote or box, which will be more stable and will avoid damaging fragile items. Similarly, cartons or totes will tend to be packed more efficiently if the larger items are placed first, so it is desirable to slot the large item early in the pick-path. These ideas are summarized in Figure 9.4, which represents a general pattern in which items may be slotted along a pick aisle.

Large but light	Less popular	Least popular
Heavy items	**Most popular items**	Less popular
Large but light	Less popular	Least popular

Figure 9.4: Arrangement of skus on the pick face of an aisle to ensure that heavy or large items are packed first, heavy items can be handled safely, and popular items can be reached easily. (Direction of material flow is left to right.)

Sometimes it is advantageous to store product by product family. For example, a retail drugstore typically displays all the hair-care products together. If these products are also stored together in the warehouse, then they will be more likely to be picked and packed together, and this can reduce work to shelve product arriving in the retail store. This creates some possible inefficiencies in the warehouse, but presumably these are offset by labor savings throughout the retail chain.

Another issue to consider is the power of certain skus to complete orders. For example, Figure 9.5 shows the most popular skus during one week in a warehouse together with the number of times each sku constituted the entire order by itself. One sku appears to be ordered frequently and alone, which suggests it should be handled separately. For example, appropriate shipping containers might be stored right beside it so that it can be picked and packed directly.

Finally, conventional wisdom suggests that there may be economies to be realized if products that are frequently requested together, such as flashlight and batteries or hat and gloves, are stored together. Our studies suggest that such opportunities are more likely to arise in service parts or catalog fulfillment or e-commerce.

Products, such as those of Figure 9.6, that tend to be ordered together are said to display *affinity*. In general affinity is easier to recognize and exploit if a typical order is small. If the customer is a retail store, the orders it places reflect the aggregate demands of many individual customers and so affinity is harder to recognize.

When there is affinity among skus it may be possible to reduce cycle times of customer orders within the warehouse by storing such skus nearby each other. This reduction in cycle time may come from:

- Travel reduction: Storing two skus near each other may reduce the travel of the order pickers.

  Whether this is so depends on many particulars within the warehouse, including the order-picking process and the geography of the warehouse. It may be that the order-picker is picking so many lines that they have to travel most of the warehouse anyway and so storing by affinity does not significantly reduce travel time. It may also depend on what is meant by "near each other": For example, it may be sufficient to store them in the same pick module.

- Order completion: If two items that are frequently requested together also fre-

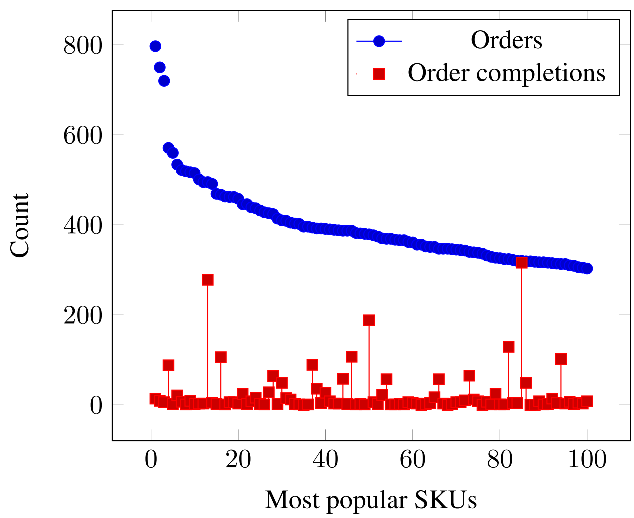


Figure 9.5: In this warehouse, one of the most popular skus (#84) was almost always ordered by itself.

Figure 9.6: Pasta and tomato sauce are two skus that have “affinity”: Any customer ordering pasta is likely to order tomato sauce. Storing them together may reduce the work to pick the order.

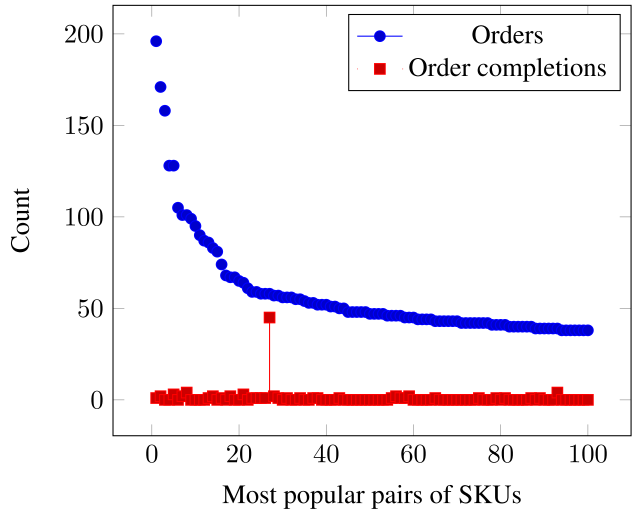


Figure 9.7: For this warehouse, one of the most frequently-ordered pair of skus almost always constituted a complete order.

> quently comprise the entire order, then one can, in effect, convert the 2-line order to a single line order by storing those two skus together. Then it may be that the order picker can travel to their common locale, pick both skus directly into a shipping container, and be done with the order. The benefits of completing orders quickly include reduced work to consolidate orders before shipping.
>
> This is more likely to be the case for catalog fulfillment centers, such as Nordstrom, or to e-fulfillment centers, for whom the average lines per order is small. It may also pertain to wholesalers, such as S. P. Richards.

The work to identify the most frequent k-tuples increases exponentially in k (there are $O(n^k)$ k-tuples appearing in orders from a population of n skus). This means that it is impractical to tabulate beyond small values of k. Fortunately, it seems sufficient in most cases to consider only $k = 2$. The greatest number of common picks must occur among the 2-tuples (pairs) and so if there is not significant affinity to be found there, there is no use in searching among larger tuples.

This is illustrated in Figure 9.7, which counts the number of times pairs of skus appeared on the same customer order during a week. The pairs of skus ordered together most frequently are candidates for storing near each other. More notably, one of the most popular pairs of skus almost always comprised a complete order by themselves. Storing them together would convert those two-line orders to one-line orders, with consequent savings in travel and handling.

Finally, it is possible to explore affinities among *groups* of skus. For example, it may be that customers frequently order from both vendor A and vendor B, suggesting that the product of these two vendors be stored near each other. But the benefits from this are even less clearly defined than from affinities among individual skus.

9.4 Questions

Question 9.1. *Can it happen that increasing the height of a shelf renders a given orientation of a sku* less *space efficient? Explain.*

Question 9.2. *Can it happen that increasing the height of a shelf renders* every *orientation of a sku less space efficient? Explain.*

Question 9.3. *Suppose the carton of dimensions* $1 \times 2 \times 3$ *from Table 9.1 was to be stored in a shelf of height 3 and depth 9; how would the values of space efficiency change? What if the shelves were configured to be of height 2? Of height 1?*

Question 9.4. *Consider a carton of dimensions* $1.25 \times 2.4 \times 2.75$ *that is to be stored in a shelf of depth 9; Compute the local space efficiencies on shelves of height 1.5, 2, 2.5.*

Question 9.5. *How many shelves of width 10 are required to hold sku allocations of widths 5, 6, 7, 4, 8, 2, 1? Use the* `Next Fit` *heuristic to pack from the list as given. Do the same with* `First Fit`*. Use each heuristic to pack the skus from the list sorted in decreasing order of width.*

Question 9.6. *Repeat the previous exercise using the* `Best Fit` *and* `Worst Fit` *heuristics.*

Question 9.7. *Use the* `First Fit` *heuristic to slot the following skus onto shelves of width 10 so as to produce a packing with high pick-density (many picks per unit width of shelf). Justify your choice of list from which to pack.*

Width:	*1*	*2*	*4*	*5*	*6*	*7*	*8*
Picks:	*50*	*200*	*100*	*350*	*120*	*50*	*100*

Part III

Order-picking

Order-picking is the most important process in most warehouses because it consumes the most labor and it determines the level of service experienced by the downstream customers.

Order-picking consists of roughly three phases:

- Travel to the neighborhood of the storage location. This is potentially the most work and it is non-value-adding.
- Local search, say within a bay, to find the exact location wanted. This can be significant when picking small parts, such as screws or other fasteners, or contact lenses. It is non-value-adding, but it is essential to avoid picking the wrong sku.
- Reach, grab, and put. This is a value-added activity; and it is the warehouse activity most resistant to automation.

In low-volume distribution, such as of service parts, customer orders will be small, urgent, and different from each other. Consequently, order-pickers may travel long distances for each pick; and their paths through the warehouse may be quite different, even from order-to-order. In such an environment, the challenge is to reduce travel by finding an efficient route visiting the required locations (Chapter 10).

In high-volume distribution, such as the supply of retail stores, customer orders will typically be large and may be similar. Each order-picker is likely to make many picks per unit of distance traveled; and all order-pickers are likely to follow a common path, such as along an aisle of flow rack. The challenge in such order-picking is to keep the work flowing smoothly by eliminating bottlenecks (Chapter 11).

Chapter 10

Routing to reduce travel

Travel time to retrieve an order is a direct expense. In fact, it is the largest component of labor in a typical distribution center. Furthermore, travel time is waste: It costs labor hours but does not add value. Travel time matters also because it affects customer service. The faster an order can be retrieved, the sooner it is available for shipping to the customer.

Consider McMaster-Carr, a distributor of hardware and tools. They distribute over 450,000 skus from four North American distribution centers. Fast service is very important to the customer, who might have large capital equipment or a construction project waiting for the part or tool. Therefore McMaster-Carr begins picking the order almost immediately on receipt, even though it might be for only 2–4 skus, representing only a few locations from among many. In such a case the problem changes from one of balancing flow, as in Chapter 11 to one of sequencing the locations to be visited so that total travel is small. Such a problem must be solved for each trip an order picker must make into the warehouse, because, unlike the fast-pick area, where the general path of travel is common and known in advance, in this case each trip into the warehouse may follow a different path.

10.1 The problem of pick-path optimization

The problem of visiting a given set of locations as quickly as possible has been nicknamed the "Traveling Salesman Problem" (TSP) and has been much studied [35]. In general, the TSP is difficult in several senses:

- There is no known fast solution technique that works in general.
- Randomly-generated instances, even small ones, can be surprisingly time-consuming to solve.
- Optimum, or even good solutions can be complex and hard to describe.

Order-retrieval in a warehouse presents a special case of the TSP in which travel is constrained by aisles and this special structure makes it possible to find optimal solutions quickly by computer [41, 29, 43]. However, most warehouse management systems do not support pick-path optimization beyond simple sorting of locations. There are several reasons for this. The most important is that any optimum-finding algorithm must know the geometric layout of the warehouse, including distances between all pairs of storage locations; and most WMSs do not maintain this level of information. Such detailed information would not only be time-consuming to gather but would have to be specialized to every site and updated after any change in physical layout.

Finally, even if the WMS does support some kind of pick-path awareness, there remains the problem of communicating the path to the picker. A high-quality path is not useful if the order picker does not follow it. Typically the WMS tells the picker only the sequence of locations, not the actual path to follow. The picker must figure out the shortest path from location to location; and this can be hard to do because order pickers work under pressure and with only local information. Figure 10.1 shows the difficulty.

Incidentally, in this regard it can be more effective to pick from a paper pick list than from an RF device. With paper, the order picker can see at a glance the next few locations to be visited and can use his knowledge of the warehouse to plan his path. On the other hand, a typical RF device displays only the very next location to be visited, which makes it impossible for the order picker to improve the picking sequence. This situation may change soon as advanced telecommunications enables the WMS to pass the order pickers actual maps of pick paths to be followed.

10.2 Heuristic methods of generating short pick paths

How can we generate short travel paths that are realizable by an order picker who has no detailed map of the warehouse?

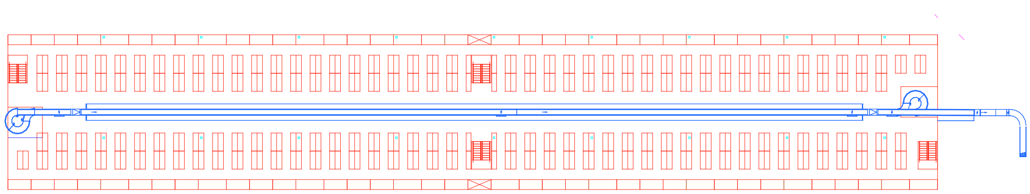
Imagine that a picker must visit *all* the storage locations of a warehouse; and suppose further that we can find an efficient global path to visit all these locations. We have to compute this efficient path only once and then we can use it many times: When a picker travels to retrieve the items of an order, we require that he simply visit the required locations in the same sequence as does the efficient global path. Thus the global path imposes a sequence that will be respected by all travel. When we receive a customer order the WMS simply sorts the pick lines by storage location so that they appear in the same sequence as the efficient global path. The idea is that if the global path is efficient the sub-path induced on each customer order is likely to be efficient as well.

The problem of finding a good global path through the storage locations is known as the “Probabilistic Traveling Salesman Problem” or PTSP.

10.2.1 Path outlines

A good global path should not only induce short sub-paths on the customer orders, it should also help the picker visualize the next location and how to travel there most directly. We want a path outline that will induce a short pick path for most orders and yet is simple in structure so that order-pickers can understand it. An effective path

(a) Local information



(b) Global information

Figure 10.1: An order picker has only local information, such as a sequence of locations and the view from an aisle (a), from which to determine a pick path that is globally efficient. It is hard for the order picker to know or account for the global layout of the pick area (b).

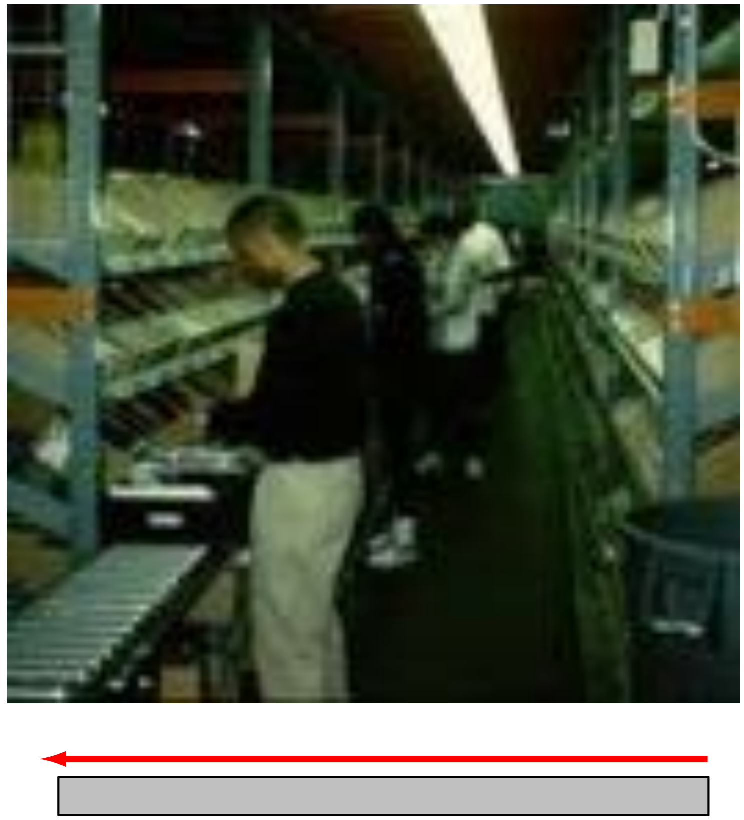


Figure 10.2: This picking area contains only popular skus and so every order is likely to require walking down the aisle. Thus an efficient route of travel is known in advance.

outline will account for the physical layout of rack, where the most popular items are stored, and what a typical order looks like. In addition, management may devise simple rules by which the path outline can be adapted for the particular customer orders. By providing the order-pickers with a set of rules to adapt the path, they leverage the intelligence of the work force, rather than embedding the decision-making in the WMS software.

The simplest path outline is that along a single aisle, as shown in Figure 10.2. Such a path outline has the desirable property that any optimal path is consistent with this ordering. This configuration is typically found in a fast-pick module of a distribution center.

A commonly found path outline through static shelving is the so-called *serpentine* pick path illustrated in Figure 10.3. The path induces 1-way direction of travel in each aisle, which means that it may be possible to conserve floor space by using narrow aisles. However, location sequence might not give optimal travel paths, as in this instance where the picker would have to travel needlessly along the lengths of aisles 3 and 6. Unless a typical customer order visits every aisle, such a path can result in

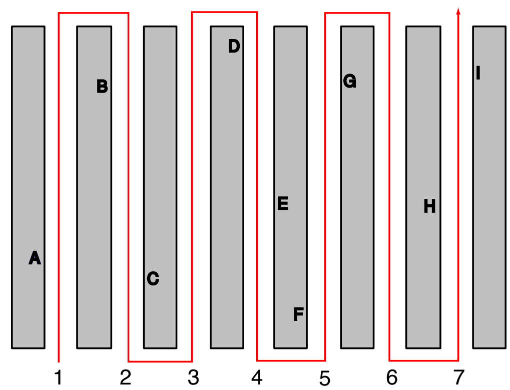


Figure 10.3: A serpentine pick path can result in unnecessary travel (in this case along aisles 3 and 6).

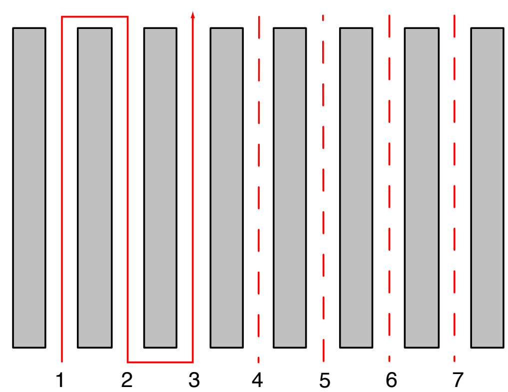


Figure 10.4: A modified path outline that sorts aisles 4–6 but not their locations.

wasted travel.

One way of ameliorating this problem is to specify only a partial ordering among the storage locations. For example, Figure 10.4 shows an incompletely-specified serpentine outline that sequences the locations of the first few aisles from the left but thereafter sequences only the aisles themselves and not locations within the aisles. This imposes less *a priori* structure on the eventual pick path and relies on the intelligence of the order picker to adapt it appropriately. This example could be an effective path outline if the early (leftmost) aisles contain the more popular products. In this case the picker can construct a more efficient pick path by skipping aisles 3 and 6.

Because this path outline cannot guarantee in advance in which direction the picker may travel the later (rightmost) aisles, the right-side of the warehouse must allow 2-way travel for passing and so may need wide aisles.

Another common type of pick path is the *branch-and-pick*, which sequences only the aisles and not the locations within an aisle. The pick path passes a reduced set of lo-

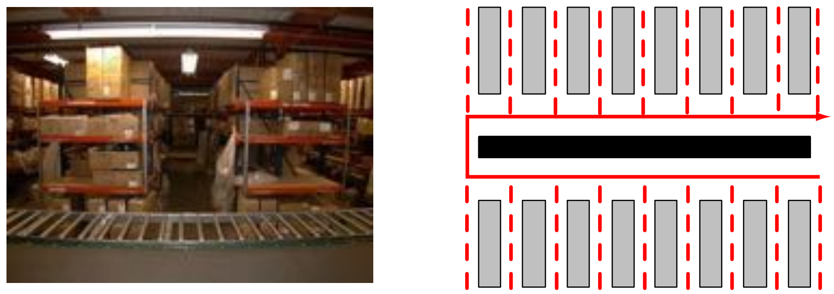


Figure 10.5: Example of branch-and-pick

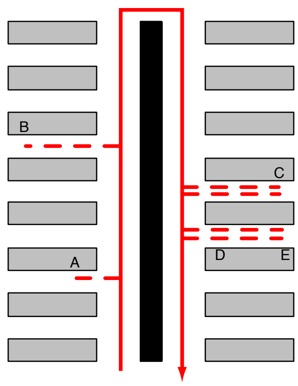


Figure 10.6: Left: Travel path generated by detouring into and back out of aisles. Right: Travel can be reduced by allowing picker to determine entry and exit from each aisle.

cations (endcaps), which are where fastest-moving items should be stored (Figure 10.5) and the picker detours as necessary into the aisles. This is typically used when there are shallow aisles with some slow-moving product.

It can be useful to further rules with which a picker can generate a travel path from a pick list. For example, the simplest algorithm might be "Visit aisles in sequence; detour into and back out of each aisle as necessary to pick product".

10.2.2 Product placement

To help pickers guess the best way to travel, one must make the geometry and addresses work together. For example, if locations that are close also have similar addresses, and *vice versa*, then the pick list also tells picker something about *where* the next address is in the warehouse. The global (layout of the warehouse) is embedded in the local (list of addresses).

In addition, it is generally better to store the most popular skus close to the path outline, so that they can be reached with little or no detour, as shown in Figure 10.7. Fewer and shorter detours also means that the eventual travel path will be simpler,

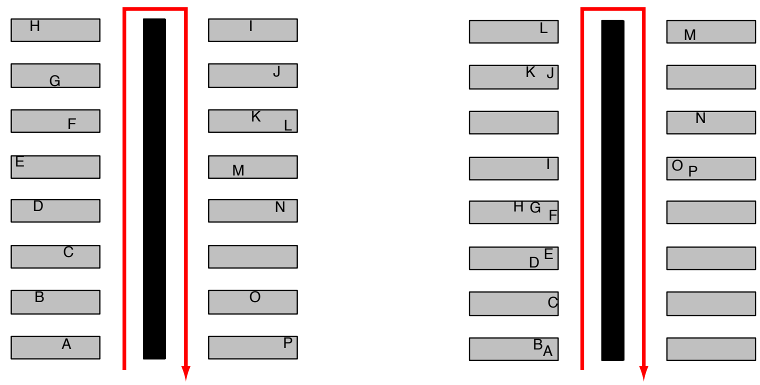


Figure 10.7: A much shorter travel path is possible if popular items are stored close to path outline, as on the right.

which means that the savings is more likely to be realized because the order-picker can follow it.

10.3 Pick-path optimization

To compute optimal pick paths requires that the computer know the shortest distance between any two locations in the warehouse (and the corresponding route of travel). As of this writing, no warehouse management systems that we know of manage an explicit geometric model of the layout of the warehouse. Therefore true pick-path optimization is not currently done.

Eventually warehouse management systems will have geometric models of their warehouses; and advances in telecommunications will make it easy and economical to give picker precise travel instructions. Then there will be no reason not to take advantage of pick-path optimization.

The fundamental result in pick-path optimization is due to Ratliff and Rosenthal [41], who gave an algorithm for quickly finding the shortest tour of a set of locations in a warehouse. We will illustrate their ideas by giving a simplified version of their algorithm, which will generate *near*-optimal pick paths. The simplification is to restrict slightly the allowable patterns of travel so that the picker is forbidden to revisit a previously-visited aisle. In other words, we will find the shortest pick path subject to the constraint that the aisles cannot be visited out of their natural sequence.

Because of this restriction the suggested path may be slightly longer than the unconstrained optimum; but

- This algorithm is in the same spirit as the optimum-finding algorithm, but is much simpler and so is easier to program and to explain.
- Any unnecessary travel required because of the no-backtracking restriction is generally small.

- The generated path is simpler in structure than an optimum path and so easier for an order picker to understand and follow.

Following [41] we generate a pick-path by dynamic programming. This takes advantage of the fact that an optimum path can assume only a limited number of patterns at each end of an aisle. Our restriction that the picker can never revisit an aisle reduces the number of possible patterns to only two so that each order-picker can be imagined to follow this rule.

> Pick all the required items from the current aisle and travel forward to the next aisle.

Note that there are two ways of traveling to the next aisle: An order-picker can either

- Travel all the way through the current aisle, picking as necessary; or else

- Enter the aisle only as far as necessary to pick all required items therein and then return to the same end of the aisle from which it was entered.

The decision points are at the ends of each aisle, as suggested in Figure 10.8. Be-

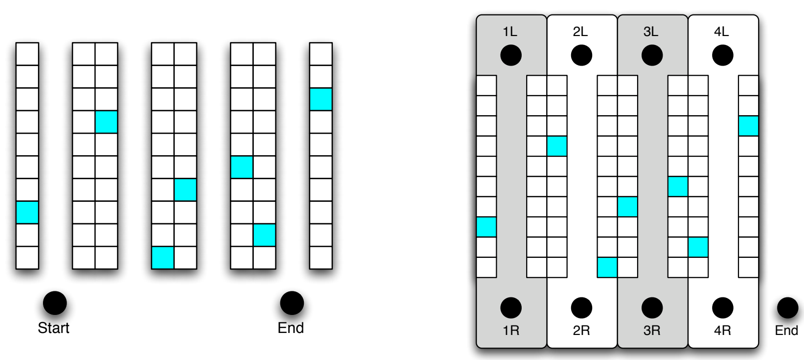


Figure 10.8: To visit the locations on the left, imagine a decision point at the end of each aisle. Because backtracking is forbidden, the order-picker must visit aisle i before visiting aisle $i + 1$.

cause there is no backtracking the picker must travel left to right across the warehouse and so we can consider sequentially the decisions to be made at each aisle. We can graphically summarize the sequence of decisions to be made as in the series of figures beginning with Figure 10.9.

Figure 10.13 shows the final graph summarizing the sequence of decisions to be made by the order picker. The shortest path in this graph corresponds to an efficient pick path through the warehouse. (The shortest path can be found by elementary algorithms from graph theory or discrete mathematics, as explained in Appendix C.)

Notice that the connections of the network need be built only once; and for subsequent use only the lengths of the edges need be updated, as shown in Figure 10.14.

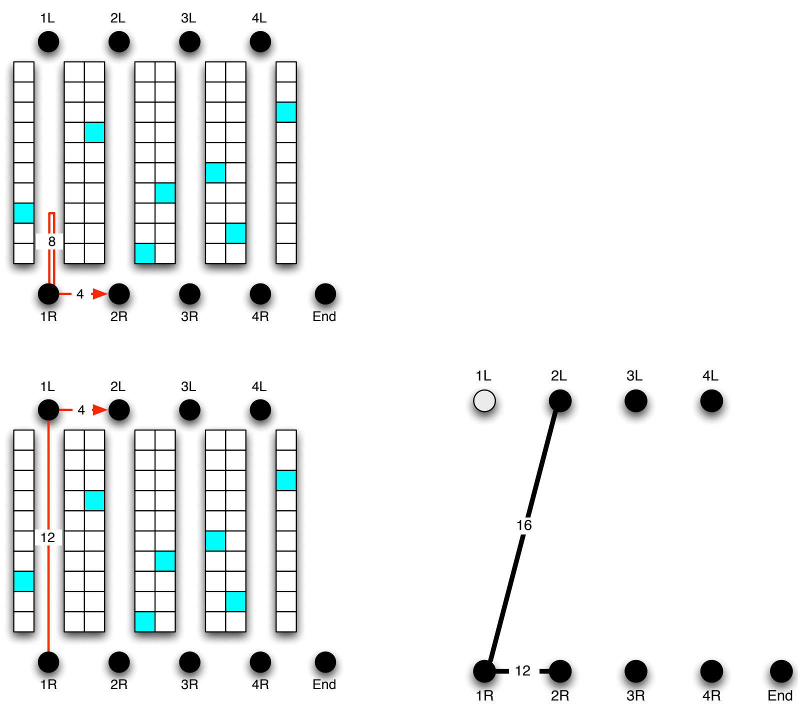


Figure 10.9: Enumeration and summary of paths from Aisle 1 to Aisle 2. Each candidate path is represented by an edge of the same length in the graph.

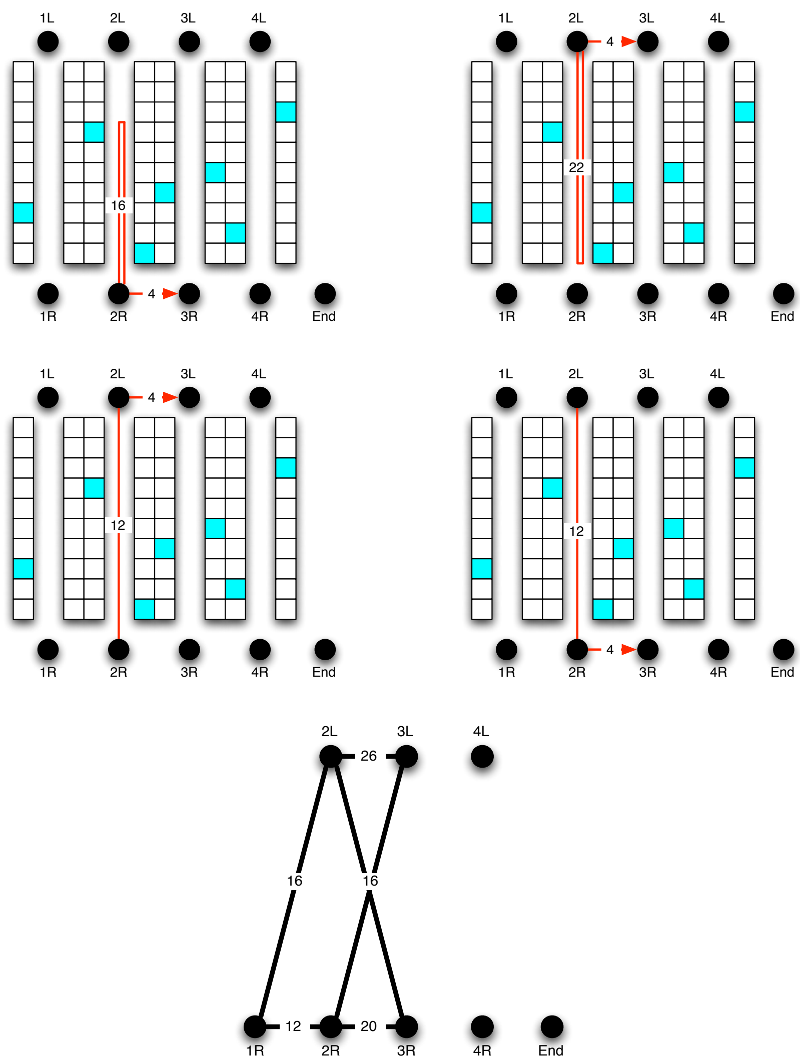


Figure 10.10: Enumeration and summary of paths from Aisle 2 to Aisle 3

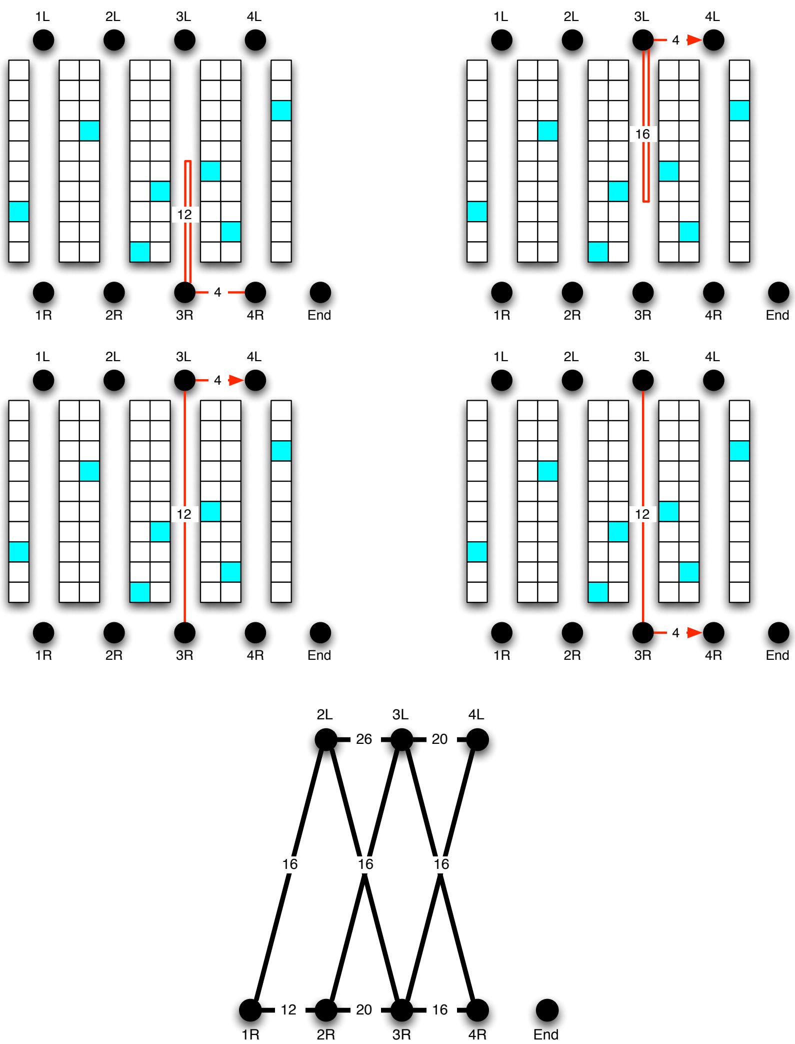


Figure 10.11: Enumeration and summary of paths from Aisle 3 to Aisle 4

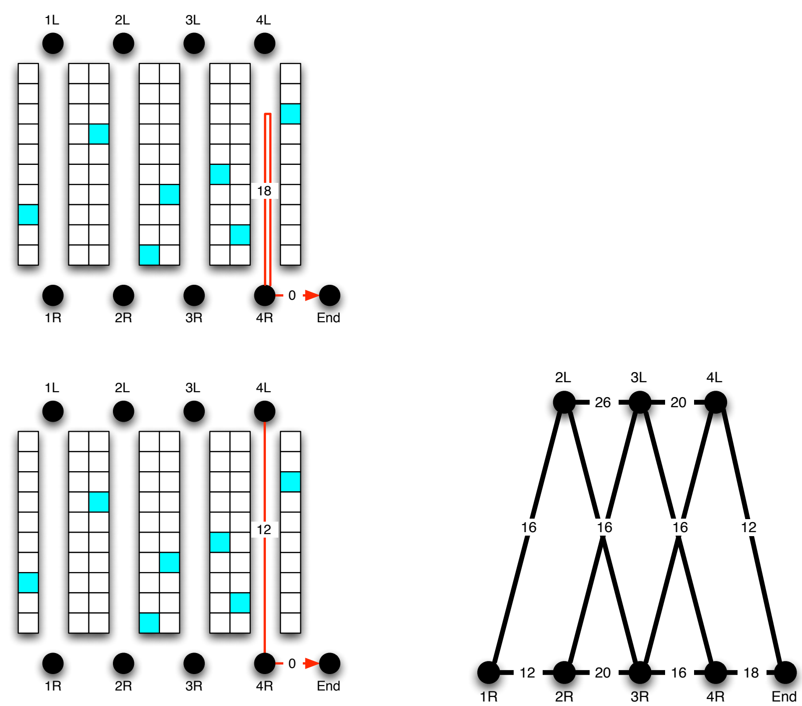


Figure 10.12: Enumeration and summary of paths from Aisle 4 to completion.

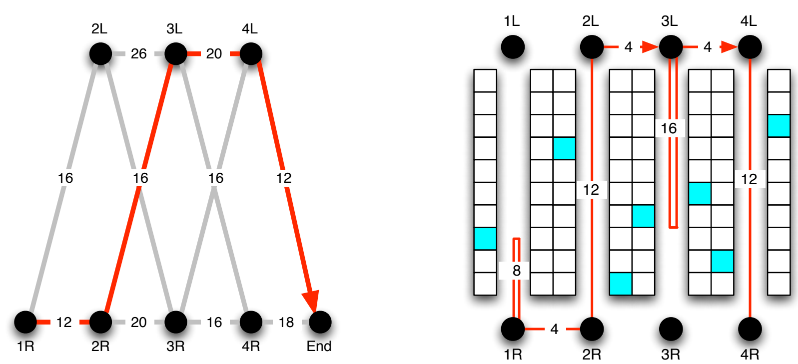


Figure 10.13: The shortest path on the associated graph gives an efficient pick path in warehouse

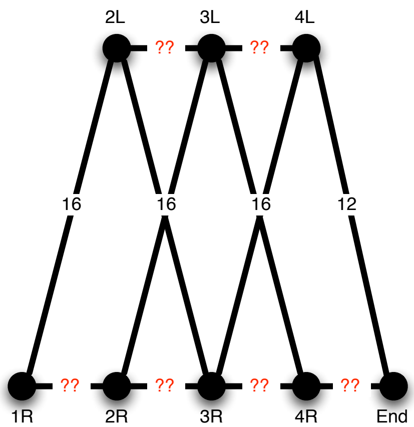


Figure 10.14: Only the lengths of the edges along each side of the graph need be updated to reflect new customer orders.

This approach can be extended naturally to handle the addition of cross aisles, although the work to solve increases quickly [41].

10.3.1 How to take advantage of optimization

For the effort of optimization to be worthwhile, it must be easy to implement a computed solution. But it can be hard for an order picker to know how to travel in the shortest way if they are told only the sequence of locations and not the paths from location to location. In fact, the shorter the pick path, the less likely it is to make sense to the order picker, who has only local information.

Nevertheless there are some simple ways to help the picker follow optimized pick paths. For example, the WMS could include instructions giving the direction the order-picker should travel after making each pick. Figure 10.15 shows an example pick list that would give sufficient information to specify routes, not just sequences of picks, for a pick path that followed a branch-and-pick outline (no backtracking to a previously passed aisle). Of course the effectiveness of this depends on how disciplined the pickers are. There is a trade off here: The more complex paths allowed, the more information that must be passed to the order pickers.

10.3.2 How much is optimization worth?

Finally, it must be asked how much pick-path optimization matters. It depends. In some cases in which pick path optimization does not matter much at all: For example, if all orders are for 1–3 items, such as at a catalog fulfillment center, then technology is not needed to optimize pick paths because the order-pickers can probably find short paths themselves. At the other extreme, if each order requires visits to very many locations then the optimal pick path must traverse most of the warehouse and could not be much

Aisle	Bay	Shelf	Sku	Qty	Continue in direction...
2A	5	1	ABC324	1	- away from conveyor
2B	8	2	CBD286	1	- toward conveyor
2B	1	2	DAB332	3	- toward conveyor
3A	4	1	ACF119	2	- away from conveyor
⋮	⋮	⋮	⋮	⋮	⋮

Figure 10.15: A pick list with travel directions suitable for use on a branch-and-pick pick-path outline

shorter than any reasonable pick path. (This suggests that one can reduce the need for pick-path optimization by batching orders, because the larger the batch, the less there is to be gained by optimization and the fewer batches there are over which to accumulate savings. Similarly, one can reduce the need to optimize by placing product so that the most frequently requested items are stored near each other.)

Warehouses that are most likely to benefit from pick-path optimization are those that have many items, most of which are slow-moving, and customer orders of moderate size. Examples include warehouses distributing hardware, building supplies or aftermarket auto parts to dealerships.

10.4 Summary

Despite advertising claims, most WMS's do not support pick-path optimization. Instead they simply sort each pick list according to storage address (path outline). To make this method work best:

- Define path outlines that will generate short, understandable routes.
- Give pickers local rules to help them adapt the path outline.
- Place product to work with the path outline.

Pick-path optimization is in principle doable now but is rarely implemented for several reasons.

- The WMS must have a geometric model of the warehouse layout to compute shortest paths between pairs of locations.
- Limited communications bandwidth makes possible to tell picker where to go next but not the route by which to travel there.
- It does not always generate significant savings.

10.5 Questions

Question 10.1. *Consider the problem of finding the shortest pick path through a warehouse with parallel aisles but no cross aisle. Give an example of when a shorter path is possible if backtracking is allowed.*

Question 10.2. *Devise a worst-case example to show how much longer a picker might be required to travel if backtracking is forbidden in a warehouse with parallel aisles (no cross aisle).*

Question 10.3. *The word "detour" generally connotes the requirement to go out of your way. In what sense is it a detour to enter a side aisle to pick an item as part of a larger order in a branch-and-pick system?*

Question 10.4. *In which of the following scenarios might pick-path optimization be economically justified?*

- *A very busy unit-load warehouse*
- *A warehouse in which orders arrive intermittently and each is picked immediately (no batching). A typical order is for 1–2 skus.*
- *A distributor of recorded music, with only a very few, very popular skus.*
- *A warehouse that does most of its picking from a few single-aisle pick modules onto conveyor.*
- *All of the above*
- *None of the above*

Question 10.5. *Which of the following best describes the effect of adding crossover aisles to a warehouse? (That is, aisles that run orthogonally across the main direction of aisles.)*

- *It reduces travel because it creates shortcuts.*
- *It complicates travel-planning because it creates more possible paths.*
- *It reduces storage capacity by taking aisle space.*
- *None of the above*
- *All of the above*

Question 10.6. *To pick a customer order requires that the order-picker visit the set of locations shown in Figure 10.16. Label the corresponding network with the appropriate distances so that solving the shortest path problem on the network generates a shortest tour visiting all the locations. Use Dijkstra's algorithm, described in Appendix C, to compute the shortest path and then interpret this in terms of travel in the warehouse.*

.

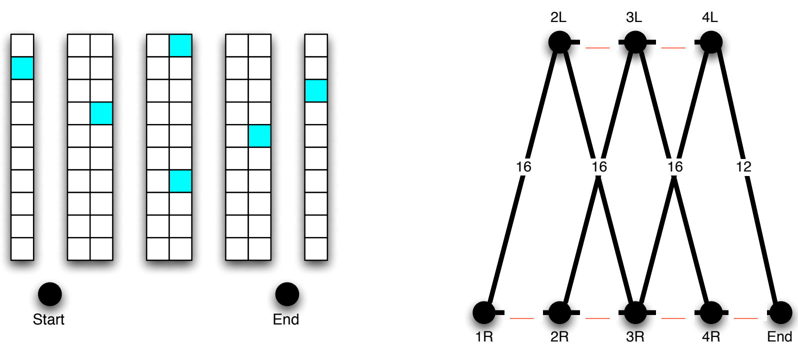


Figure 10.16: What is the shortest route by which an order-picker can travel from the starting location to visit all the shaded locations and finish at the right? (Question 10.6)

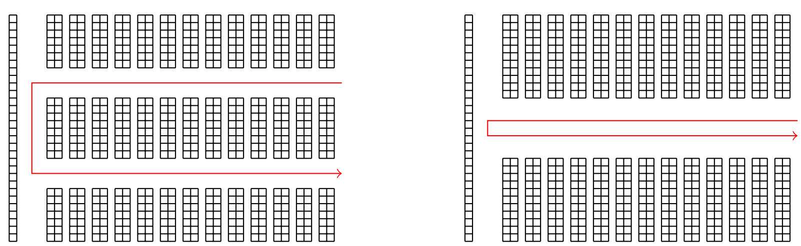


Figure 10.17: Which is the better pick-path? (Question 10.7)

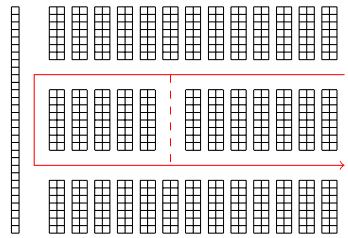


Figure 10.18: Where should a cross-aisle (dashed line) be located? (Question 10.8)

Question 10.7. *Figure 10.17 shows two possible pick-paths for a warehouse in which pickers follow the default path, detouring as necessary to pull the items on their pick-lists. Each order-picker pushes a cart to hold work-in-process; and the carts will not fit into the narrow aisles.*

Compare the two designs, explaining the strengths and weaknesses of each. Under what conditions would each be preferred?

Question 10.8. *Consider the layout of Figure 10.18, in which each order-picker pushes a cart to hold work-in-process and where the carts will not fit into the narrow aisles . How can you improve the efficiency of this design by clever slotting and batching of customer orders? How can you decide where to locate a cross-aisle?*

Question 10.9. *Which of the following significantly increases the computational difficulty of solving the traveling salesman problem in a warehouse?*

- *Lengthening the aisles in the warehouse.*
- *Extending the warehouse by adding more aisles.*
- *Widening the aisles.*
- *Adding cross-aisles to the warehouse.*

Question 10.10 (Harder)**.** *Generalize the algorithm to find shortest pick-paths to the more general case in which backtracking is allowed [41].*

Question 10.11 (Exploration)**.** *Generalize the algorithm to find shortest pick-paths to account for a single "cross-aisle" running horizontally through the middle of the warehouse [43]. Do the same for multiple cross-aisles [42].*

Chapter 11

Work flow and balance

A forward-pick area for small parts is typically the most compact and labor-intensive part of a warehouse. This activity within the warehouse has grown enormously in the past 20 years along with e-commerce, and many large retailers now ship directly to their customers. In addition, the drive to reduce inventories has led to more frequent shipments of smaller quantities.

There may be very little travel required in high-volume piece-picking, because the density of picks can be quite high. Specialized storage equipment, such as carton flow rack can present 30–50 different cartons for piece-picking within a single bay. Consequently most of the non-value-adding work is due to local search.

In North America a common way of reducing the time for local search is *pick-to-light*, wherein a computer turns on lights to indicate from where the worker should pick product. Here is how it works.

Pick-to-light is most frequently installed as part of a module that consists of an aisle of flow rack, passive roller conveyor along which totes can be pushed without having to lift, and a belt conveyor running along the aisle as in Figure 5.3. Lights are installed at each storage location or each sku. The aisle is then divided conceptually into zones of contiguous bays, so as to equalize the total work among zones. (A zone can be as small as a single bay.)

Each zone may be considered to have a queue of totes that must be picked in sequence. The order-picker will push the queue of totes along the roller conveyor, picking as required.The head of the queue is the currently active tote and the pick-to-light system signals the appropriate action for the order-picker to take regarding this tote: Either

- Pick from a location indicated by a glowing light (after which the tote will remain at the head of the queue); or
- Pass the tote to the next zone (in which case the tote exits the queue for this zone, joins the end of the queue for the next zone, and the next tote becomes the head of the queue for this zone); or
- Push the tote onto the conveyor (in which case the tote will exit the queue and

leave the aisle; the next tote becomes the head of the queue for this zone).

Totes join a queue either by being passed from the previous queue or by being brought via conveyor to an *induction point* (where the live conveyor deposits totes to the passive roller conveyor).

As they move through this process, each tote has a *license plate* (identity), which is associated with a set of picks to be made. The warehouse management system coördinates with the pick-to-light system to identify the currently active tote in each zone and to signal which picks to make next into this tote.

Workers cannot change the sequence of totes, but they can choose which zones queue to work next. Typically a worker will escort a physical collection of totes from one zone to the next, then, at some point, walk back to get another collection of totes. Exactly when this happens is a question of coördination that we discuss next.

11.1 Organizing a team of order-pickers

In high-volume order-picking the challenge is to get workers to where they are most needed, so that everyone remains busy and working on what is due next. This is a challenge of organization, and there are some standard methods in common use, which we will discuss later. But first we want to introduce an idea that may seem surprising: That it is possible for a team of order-pickers to form a *self-organizing system*.

A self-organizing system is one in which global organization spontaneously evolves from myriad local interactions of the pieces. Self-organizing systems do not require a centralized authority to manage them. Instead, they achieve global coördination spontaneously through the interaction of many simple components.

Here is an example: Consider a hive of honeybees. Each day they face a logistics problem of how to coördinate their efforts to harvest nectar. The measure of success is a social one: the good of the colony. But bees have no blueprint, no mechanism of central planning. Instead, each bee follows a simple "algorithm" that determines what she does next; and when many bees follow the same algorithm, an allocation of foragers evolves that is close to the best imaginable. In effect the colony acts as a computer that finds the (nearly) optimal allocation of effort [12].

Among the advantages of this self-organization are that it requires no central planning or higher organizational entity. There is no management function because each entity simply follows a local rule. Furthermore, it is adaptive: It will spontaneously reallocate effort in response to changes in the environment.

Exploring these simple ideas has led to some practical applications within management science/industrial engineering. Here is one in warehousing: When workers are organized into "bucket brigades" they can function as a self-organizing system that spontaneously achieves its own optimum configuration without conscious intention of the workers, without guidance from management, without any model of work content, indeed without any data at all. The system in effect acts as its own computer.

Bucket brigades are a way of coördinating workers who are progressively assembling product along a flow line in which there are fewer workers than stations (work stations in the context of manufacturing; storage locations in the context of order-picking). Each worker follows this simple rule: "Carry work forward, from station

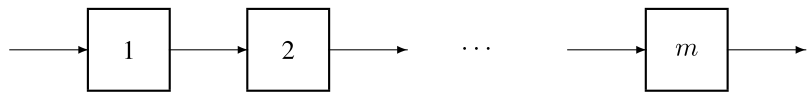


Figure 11.1: A simple flow line in which each item requires processing on the same sequence of work stations.

to station, until someone takes over your work; then go back for more". When the last worker completes a product (or customer order), he walks back upstream and takes over the work of his predecessor, who then walks back and takes over the work of his predecessor, and so on, until the first worker begins a new product (customer order) at the start of the line. No unattended work-in-process is allowed in the system.

Note that workers are not restricted to any subset of stations; rather they are to carry each product as far toward completion as possible. Note also that a worker might catch up to his successor and be blocked from proceeding; the bucket brigade rule requires that the blocked worker remain idle until the station is available.

The final requirement of bucket brigades is that the workers be sequenced from slowest to fastest along the direction of material flow. These protocols, taken together, make the bucket brigade line a perfect *pull system*.

11.1.1 A model of work and workers

Consider a flow line in which each of a set of items (customer orders) requires processing on the same sequence of m work stations (storage locations), as in Figure 11.1. A station can process at most one item at a time, and exactly one worker is required to accomplish the processing.

The Normative Model, suggested in [7], is given in the following assumptions. We call this model "normative" because it represents the ideal conditions for bucket brigades to work well. However, it is not necessary that these assumptions hold exactly: The behavior of a bucket brigade will resemble that predicted by the Normative Model to the degree that the assumptions of the Normative Model hold. Accordingly implementations should try to make these conditions hold as much as possible—but it is not necessary that they hold exactly, or even to any great extent.

The assumptions are:

Assumption 11.1 (Insignificant Walkback Time)**.** *The total time to assemble a product is significantly greater than the total time for the workers to hand off their work and walk back to get more work.*

Assumption 11.2 (Total Ordering Of Workers By Velocity)**.** *Each worker i can be characterized by a work velocity v_i.*

Assumption 11.3 (Smoothness, Predictability Of Work)**.** *The work-content of the product is spread continuously and uniformly along the flow line (the length of which we normalize to 1).*

The assumption of Insignificant Walk-back Time is uncontroversial; it claims simply that it takes longer to assemble a product than it does to walk the line; and, furthermore, it is easy to hand off work.

The assumption of Total Ordering Of Workers By Velocity is likely to hold in an mass-production environment, where work has been de-skilled so that velocity is based on a single dimension, such as motivation or eye-hand coördination.

There is clearly some license in the assumption of Smoothness And Predictability Of Work; nevertheless, this assumption is reasonable in many instances, detailed elsewhere [7]. Suffice it to remind the reader that management and engineering strive to remove variance from work and eliminate bottlenecks, a result of which is to move practice closer to the Normative Model. Still, this assumption is at least less clear than the others and accounting for this is part of the art of implementing bucket brigades.

To what extent do the conclusions of the Normative Model hold when there is variation in the work-content? In short, the behavior of a bucket brigade remains qualitatively similar to behavior predicted by the Normative Model, with this caveat: the faithfulness of the replication depends on the degree of randomness. This means that, except in degenerate cases, it remains preferable to sequence the workers from slowest to fastest and one can expect a high production rate from bucket brigades.

Bartholdi and Eisenstein (1996a) have described the behavior of bucket brigade production lines under the Normative Model [7, 5]. Their main results, slightly simplified, are as follows.

Theorem 11.1. *No matter where a given set of workers start,*

- *There is a unique balanced partition of the effort wherein worker i performs the interval of work:*

$$\textit{from } \frac{\sum_{j=1}^{i-1} v_j}{\sum_{j=1}^{n} v_j} \textit{ to } \frac{\sum_{j=1}^{i} v_j}{\sum_{j=1}^{n} v_j}, \tag{11.1}$$

  so that each worker invests the same clock time in each item produced.

- *If the workers are sequenced from slowest to fastest then, during the normal operation of the line, work is spontaneously and constantly reallocated to reach this balance; and the production rate converges to*

$$\sum_{i=1}^{n} v_i \textit{ items per unit time,}$$

  which is the maximum possible for the given set of workers.

- *If the workers are* not *sequenced from slowest to fastest, then the line will "sputter": that is, it will produce erratically and at suboptimal rate. Furthermore, the line can behave in counter-intuitive ways, such as production rate* decreasing *when a worker increases his velocity.*

Before proving the result in general, we first argue that it is true for the case with two workers. Imagine that we are taking a series of photographs of the line at those

Figure 11.2: Positions of the worker 2 immediately after having completed the k-th order and walked back to take over the order of worker 1 (who has walked back to the start of the line to begin a new customer order).

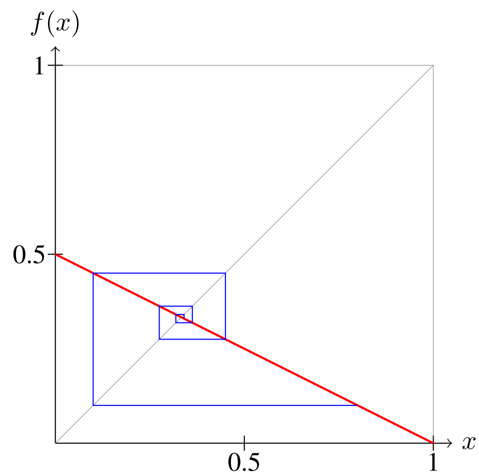


Figure 11.3: Because the dynamics function (red line) is continuous and has slope of absolute value less than 1, successive iterations (blue) converge to a globally attracting fixed point, where $f(x) = x$. In other words, the assembly line balances itself.

times when the workers have just made their hand-offs and the first, slowest worker is beginning a new product. We will study how these photographs change.

Let x be the percent completion of the product held by worker 2 in the k-th photograph (that is, after a total of k items have been completed), as in Figure 11.2. Then the next order will be completed after an elapsed time of $t = (1 - x)/v_2$. During that time, worker 1 will have traveled forward distance $v_1 t$ and so the next hand-off will occur at position $(v_1/v_2)(1 - x)$. We can therefore summarize the changes in the locations of the hand-offs as the following dynamics function:

$$f(x) = (v_1/v_2)(1 - x) .$$

This function is linear with negative slope, as illustrated in Figure 11.3. Furthermore—and importantly—because the workers have been sequenced from slower to faster, $v_1 < v_2$ and so $v_1/v_2 < 1$. Thus the slope of the dynamics function is of absolute value less than one and so is a *contraction map*, which means roughly that subsequent hand-offs get closer to each other [2]. Figure 11.3 traces a sequence of hand-offs at positions $x^{(0)}, x^{(1)} = f\left(x^0\right), x^{(2)} = f\left(x^{(1)}\right) = f\left(f\left(x^{(0)}\right)\right), \ldots$, which converges to the fixed point, the intersection of the dynamics function with the identity, where $f(x) = x$.

Here is a proof of convergence for two workers. Let $x^{(k)}$ denote the fraction of work completed on the k-th item as it is handed from worker 1 to worker 2.

Proof. Let $r = v_1/v_2$ and note that

$$\begin{aligned} x^{(1)} &= r\left(1 - x^{(0)}\right); \\ x^{(2)} &= r\left(1 - x^{(1)}\right) \\ &= r\left(1 - r\left(1 - x^{(0)}\right)\right) \\ &= r - r^2\left(1 - x^{(0)}\right); \text{ and} \\ x^{(3)} &= r - r^2 + r^3\left(1 - x^{(0)}\right). \end{aligned}$$

By induction

$$x^{(k)} = r - r^2 + r^3 - \cdots + (-r)^k\left(1 - x^{(0)}\right).$$

Because $r < 1$,

$$\lim_{k\to\infty} x^{(k)} = \frac{r}{1+r} = \frac{v_1}{v_1 + v_2}.$$

□

The more general proof, for n workers, is based on the same idea.

Proof. As before, let $x_i^{(k)}$ be the percent completion of the order held by worker i immediately after completion of order k and having walked back to start the next order. (See Figure 11.4).

Figure 11.4: Positions of the workers after having completed k products.

Then the clock time separating workers i and $i+1$ is

$$t_i^{(k)} = \frac{x_{i+1}^{(k)} - x_i^{(k)}}{v_i};$$

and the next item will be completed after time

$$t_n^{(k)} = \frac{1 - x_n^{(k)}}{v_n}.$$

In the next, $k+1$-st photograph, the clock-time separating workers i and $i+1$

becomes

$$
\begin{aligned}
t_i^{(k+1)} &= \frac{x_{i+1}^{(k+1)} - x_i^{(k+1)}}{v_i} \\
&= \frac{\left(x_i^{(k)} + v_i t_n^{(k)}\right) - \left(x_{i-1}^{(k)} + v_{i-1} t_n^{(k)}\right)}{v_i} \\
&= \left(\frac{v_{i-1}}{v_i}\right) t_{i-1}^{(k)} + \left(1 - \frac{v_{i-1}}{v_i}\right) t_n^{(k)}.
\end{aligned}
$$

Because the workers are sequenced from slowest-to-fastest $(v_{i-1}/v_i) < 1$, and so we may interpret these equations as describing a finite state Markov Chain that is irreducible and aperiodic. By the Markov Chain Theorem the $t_i^{(k)}$ and therefore the $x_i^{(k)}$ converge, and exponentially fast. The specific claims follow by simple algebra. □

Figure 11.5 shows an example of how the movement of the workers stabilizes, with the faster workers eventually allocated more work. This figure was generated by a simulation of three workers of velocities $\mathbf{v} = (1, 2, 3)$.

11.1.2 Improvements that are not

It is tempting to try to improve the performance of bucket brigade lines by modifying the protocol; however, the variants that come first to mind actually perform *worse*. For example, an appealing but flawed variation of the bucket brigade protocol is to allow any worker, when blocked, to leave his partially-completed item in a buffer before the busy station and walk back to take over the work of his predecessor. This variant protocol will increase work-in-process inventory and can even *reduce* the production rate! This can be seen in simulations, where workers tend to collect in the region of the line preceding any station that is frequently busy. This increases the production rate of the preceding segment of the line, which only accelerates the accumulation of in-process inventory immediately preceding the highly-utilized station. This, in turn, decreases overall production rate of the line for two reasons:

- Fewer workers remain to staff the final segment of the line so each tends to assume a larger share of work and the time between product completions increases.
- Because no one waits in front of the frequently busy station, it is idle every time a worker leaves it, which is contrary to the principal of keeping bottleneck stations constantly busy.

Eschewing buffers seems to contradict conventional wisdom that it is important to have buffers near a bottleneck—until one realizes that in bucket brigade production one must buffer both work-in-process *and* a worker, which is done by requiring the blocked worker to remain waiting at the bottleneck station.

One might also think that the bucket brigade protocol could be improved by requiring the workers to *circle* through the work stations. This avoids any delay in handing

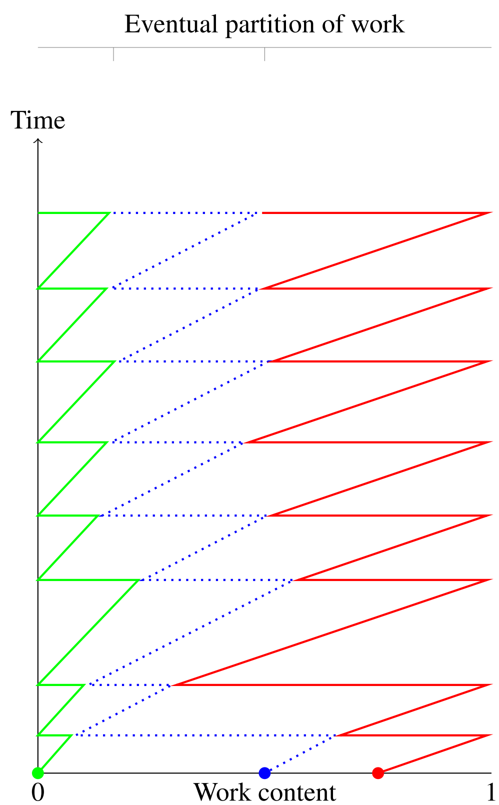


Figure 11.5: A time-expanded view of a bucket brigade production line with three workers sequenced from slowest (green) to fastest (red). The solid horizontal line represents the total work content of the product, normalized to 1. The colored disks represent the initial positions of the workers and the zigzag vertical lines show how these positions change over time. The system quickly stabilized so that each worker repeatedly executes the same portion of work content of the product.

off work but it requires that every worker perform every task. There are several objections to be made to this. First, when real workers are finally assigned to the line they will not be of identical skill levels and so the production rate will eventually be determined by that of the slowest worker, behind whom all the others will accumulate. The production rate will remain suboptimal even if faster workers are allowed to preëmpt a slower worker and pass him: The slower worker would have to remain idle until his work station became free again and so the line could not keep all workers busy. Moreover, when workers are asked to perform every task on the line then the learning effect and so realized production rate will be reduced.

11.1.3 Some advantages of bucket brigades

As a way of coördinating workers on an assembly line, bucket brigades have many attractive properties, including:

- It is a pure pull system, so work-in-process inventory is strictly controlled.
- It requires no special material handling system because the workers themselves carry the items from station to station.
- Because the line can be made self-balancing, it does not require accurate measurement of task times and so can avoid some of the expense of time-motion studies.
- It is consistent with other trends in manufacturing: For example, it exploits the advantages of work teams and the grouping of technology into cells.
- The protocol is simple and identical for each worker: Workers are not confused as to what task to perform next and management need not intervene to keep work flow balanced and production rate high.

Bucket brigades seem most appropriate when:

- *All the work is based on a single skill.* This ensures that workers can move among the stations to where the work is, without worrying about whether they can do the work there. It also allows workers to be ranked by a single score, their velocity along the production line, so that the line can be made self-balancing.

  Economic forces ensure tend to move production lines in this direction, in which the primary worker skills are simple dexterity and enthusiasm.
- *A worker can move easily among stations and can easily take over work in process.* This ensures that the bucket brigade protocol does not introduce additional wasted time to pass work.
- *Demand for the products varies significantly.* Bucket brigade manufacturing can more easily track changeable demand because cross-training of workers and low work-in-process inventory mean flexibility of configuration, and short production lead times. In addition, a bucket brigade line can be configured quickly: The assignment of tasks to stations need not be carefully balanced because the

movement of the workers balances the line; this reduces the time required to lay out a new line and so shortens changeovers. Finally, because the line is self-balancing, production rates are easily adjustable by simply adding or removing workers from a team.

11.2 Bucket brigades in the warehouse

In many high-volume distribution warehouses, fast moving items are picked from cases stored in flow rack, with the bays of flow rack arranged in aisles and a conveyor system runs down each aisle. The *start of an aisle* is the end that is upstream with respect to the movement of the conveyor. For clarity we will describe a single-aisle of flow rack. (Even when there are multiple aisles of flow rack, each aisle is generally operated as an independent module within the warehouse.)

It is typical that orders are released periodically to the picking operation as a batch. Then each order is picked by "progressive assembly": The order is picked by no more than one person at a time and the items are accumulated as the order is picked (rather than picking all orders simultaneously and sorting the items afterward).

Paperwork describing orders to be picked waits at the start of the aisle. Each order sheet lists the items and quantities to be picked in the sequence in which items will be encountered along the aisle. The first picker takes the next order sheet, opens a cardboard carton, and slides it along the passive lane of the conveyor as he moves down the aisle picking the items for that order. At some point the second picker takes over and continues picking that order while the first picker returns to the start to begin the next order. When the order is complete the carton(s) are pushed onto the powered portion of the conveyor, which takes them to the packing and shipping department.

There are several ways of coördinating the pickers. Under *zone-picking*, the bays are divided into regions and each picker works within an assigned region: Worker 1 is responsible for picking all items lying within bays $1, \ldots, b_1$; worker 2 is responsible for picking all items lying within bays $b_1 + 1, \ldots, b_2$; and so on.

In designing such order-picking systems managers try to balance the expected work among the pickers during the each picking period. The trouble with this is that it balances the work only *on the average over the picking period*, which means only that everyone will have performed the same total number of picks—yet the line can have been significantly out of balance from order to order!

The order-picking system will constantly seek balance if configured as a bucket-brigade with pickers sequenced from slowest to fastest. However, there is an important difference here: Unlike manufacturing the "items" produced on this line (that is, orders picked) are *not identical* and in fact are best modeled as "random". For example, one might think of each sku i in the warehouse as being in the next order with probability p_i independently of all other skus. Because of this, the system converges to a state of balance in a stochastic sense. This is still an improvement over a static balance because:

- It constantly seeks balance from order to order and so will be out of balance much less often and therefore it will be more productive.

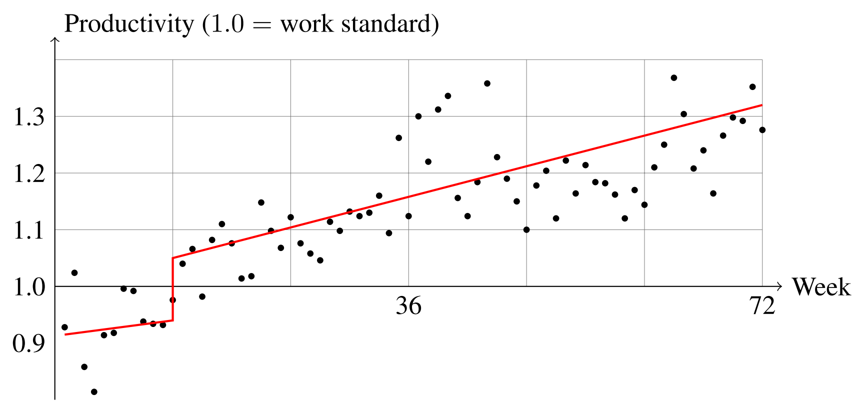


Figure 11.6: The average pick rate, reported here as a fraction of the work-standard, increased by over 30% after replacing zone-picking with bucket brigades in week 12.

- It spontaneously adapts to disruptions and seasonalities.
- It does not require anyone to compute a balance.

These advantages have been dramatically illustrated in the national distribution center of a major chain retailer that implemented a bucket brigade style of order-picking. After changing to the bucket brigade protocol, their productivity, measured in average number of picks per person-hour, increased over 30% [6], while reducing need for management intervention (Figure 11.6). This was achieved at essentially no cost, and in particular, with no change to the product layout, equipment, or control system (except to render parts of the latter unnecessary).

Previously, work on this line had been assigned by a computer-based model of work content that was run each night preceding picking. Such a model cannot be accurate because

- It cannot economically account for all the relevant detail that determines work content, such as:
  - Location, which might be at waist level or on an inconveniently high shelf.
  - Shape and weight, which might make an item easy to grab or hard to handle.
  - Velocities of the workers, who can range from 50–300% of standard.
  - Distribution of locations: One worker might have her picks distributed over three bays while another has as many picks distributed over five bays.
  - Additional work such as disposing of empty containers, sealing a full tote and opening another, prepping an sku, reaching to pull additional stock to the front of the flow rack, and so on.
  - Economies of scale: picking two units is often less than twice the work of picking one unit.

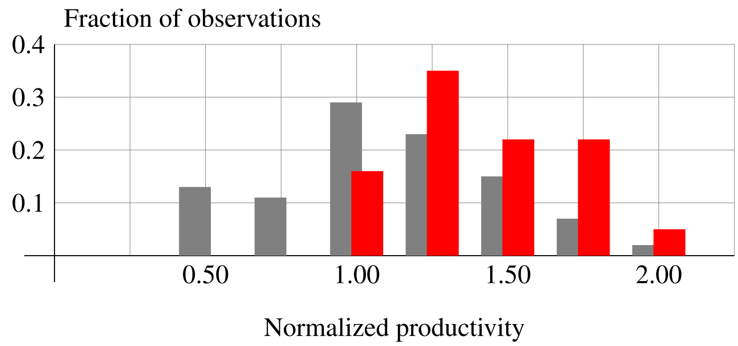


Figure 11.7: Distribution of average pick rate, measured in 2-hour intervals before (gray bars) and after bucket brigades (red bars). Under bucket brigades, average productivity increased 20% and variance in productivity was reduced by 90%.

- Even though it might appear balanced on average, the allocation of work can nevertheless be quite unbalanced for every order.
- A static balance cannot adjust to unforeseen events such as equipment malfunction, employee absences, and so on.

Because the model of work content was inaccurate, as all such must be, considerable management time was devoted to adjusting the allocation of work during the day. (In fact, the retailer dedicated a manager to this.) The bucket brigade protocol has made this centralized managerial effort unnecessary—yet still results in better performance.

Figure 11.7 shows average pick rates at 2-hour intervals at a major US distributor of recorded music. After conversion to bucket brigades the average pick rate increased by 20% and the variance of pick rate decreased by 90%; thus bucket brigades were both more productive and more predictable, which made it easier to staff.

11.3 Summary

One of the advantages of bucket brigades is that they are so flexible. As long as you are careful not to destroy the self-balancing mechanism they can be adapted to many different situations [6].

The ideas behind bucket brigades are simple:

- Abolish rigid assignments of work, which prevent people from going to where the work is.
- Sequence the workers from slowest to fastest to make a "pull system" that is self-organizing.
- Amplify the technical improvements of bucket brigades by emphasizing teamwork.

The result is to make the assembly line into an analog computer. We “program” this computer by sequencing the workers from slowest-to-fastest. There is no need to input data to this computer because the task times are “read” by doing them. The output is the optimal allocation of work.

11.4 Questions

Question 11.1. *Which of the following three ways of dividing work among order pickers is likely to be most accurate? What information is required in each case? What problems might there be during order-picking?*

- *Dividing the storage locations equally among the workers*
- *Dividing the number of picks equally among the workers*
- *Dividing the number of picks according to historical pick-rates of the workers*

Question 11.2. *Explain why the benefits to be gained from bucket brigades might be greater when there are more pickers in the aisle.*

Question 11.3. *What are the two fundamental forms of waste that are associated with bottlenecks in a flow line?*

Question 11.4. *Why is it impossible to choose and fix zones that will balance work, even if orders and worker speeds are known in advance?*

Question 11.5. *Consider the following opinion: "To keep all order-pickers occupied, they must have work to do; therefore it is advantageous to put the fastest workers at the start of the pick aisle so that they can quickly move work-in-process to downstream workers." How would you expect such an arrangement to work in a fast-pick aisle? What sorts of problems would you expect? (Assume workers may not pass one another.)*

Question 11.6. *Consider a bucket brigade line that assembles orders along an aisle of flow rack. Each order is distributed fairly evenly along the aisle. The picking is to be done by three workers that have the following work rates: worker A can pick a typical order entirely by himself in twelve minutes; worker B in ten minutes; and worker C in 5 minutes.*

- *What is the production rate of the line, measured in orders per hour, if the workers are sequenced from slowest to fastest?*
- *What fraction of each order will be produced by each of the workers?*
- *What is the production rate if the workers are sequenced from fastest to slowest?*

Question 11.7. *A. Suppose worker A averages 100 picks per hour, worker B averages 60 picks per hour, and worker C averages 40 picks per hour. If the average order requires 100 picks would be the average rate of order completion of the team under bucket brigades?*

B. Assume the workers above are assigned zones with equal work in each. For each of the following arrangements of workers along the direction of material flow, give the average rate of order completion of the team and identify where work-in-process would accumulate and where there would be idleness (starvation).

- $A \rightarrow B \rightarrow C$

- $A \to C \to B$
- $B \to C \to A$
- $B \to A \to C$
- $C \to A \to B$
- $C \to B \to A$

Question 11.8. *Consider a fast-pick aisle in which orders must maintain the sequence in which they are released to picking. Worker A can pick an order in about ten minutes on average, while worker B picks an order in about six minutes on average. Use the simple model of continuous, deterministic work to answer the following.*

A. If the workers coördinate as a bucket brigade, sequenced from slower to faster, what will be the average rate of production?

B. For the bucket brigade of part A, what is the point of perfect balance? That is, at what position, indicated by a number between 0 and 1, should hand-offs occur so that each worker invests the same clock time in each order, on average?

C. What is the dynamics function of this bucket brigade? That is, give the function that maps the position of worker 2 after one hand off to his position after the next hand off.

D. For the bucket brigade of part A, would there exist a point of perfect balance if the workers were sequenced from faster to slower? Explain.

Question 11.9. *Assuming the same model of continuous, deterministic work, Can the point of perfect balance for a 2-worker bucket brigade lie in the interval* $[1/2, 1]$*? Explain your answer.*

Question 11.10. *How does low pick density affect the effectiveness of bucket brigades in order-picking?*

Question 11.11. *How are the operation and throughput of a bucket brigade affected if the production line requires a mix of skills, rather than a single skill?*

Question 11.12 (Exploration)**.** *Suppose the fastest-moving skus in an aisle of flow rack are concentrated at the beginning of the aisle (with respect to material flow). How might this affect the operation and throughput of order-picking by bucket brigade? What if the fastest-moving skus are at the end of the aisle?*

Question 11.13 (Exploration)**.** *How does variability of work content from order-to-order affect the performance of a bucket brigade?*

Question 11.14 (Exploration)**.** *What is the throughput of a bucket brigade if the workers are sequenced other than slowest-to-fastest?*

Question 11.15. *Are bucket brigades likely to be appropriate for a service parts warehouse such as illustrated in Figure 11.8? Assume any one sku has a quite small probability of being requested and workers visit fewer than five locations on each trip.*

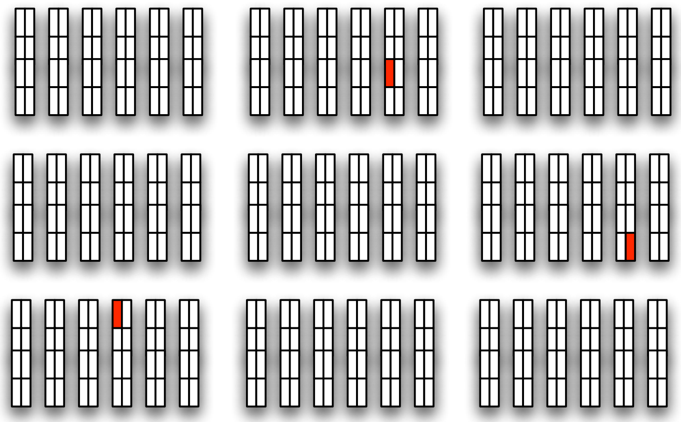


Figure 11.8: Would bucket brigades be a good way of coördinating order-pickers in this warehouse? The shaded areas represent locations to be visited to pick an order.

Question 11.16. *Explain how pick-to-light facilitates bucket brigades and also how it can interfere with bucket brigades.*

Question 11.17. *Zone picking can be improved by assigning workers to zones containing work in amounts proportional to their work speed. What are the disadvantages of this?*

Question 11.18. *Do bucket brigades reduce travel compared to walk-and-pick or zone-picking?*

Part IV

Automation

Automation is the substitution of mechanical for human labor and there is no reason to consider it unless labor is relatively expensive (or there are special considerations, such as safety or uniformity of handling). Automation makes most economic sense where labor costs are relatively high, such as in Germany, where hiring is a long-term commitment of the company to the employee. (Sometimes it can be justified in low labor cost areas such as China, when expensive product is being exported and must move quickly.)

There are disadvantages to automation, the most significant of which is that it is inflexible. It tends to perform very well those specific tasks for which it was designed, but it can be very expensive to adapt if the business changes. If automation can be justified, then it ought to be run constantly to amortize the investment.

Automation has naturally concentrated on reducing the human travel required to pick customer orders. Up to now, automation has generally taken the form of modules designed to serve as automated storage-and-retrieval devices. The automation lies entirely within the module and there are well-defined ways in which the module connects with the remainder of the warehouse. Because their logic and operation is isolated and self-contained, these modules of automation are quite susceptible to detailed mathematical analysis.

More recently there have been interesting advances in automation, especially with the introduction of a work-force of free-roaming robots that retrieve product and bring it to order-assemblers.

This part concentrates on three issues of automation. The first, and lowest level, is control; in particular, by what algorithms should product be retrieved? Answers to this provide the fundamental operating protocols for the automation.

The next higher question is where to store product so that it can, in aggregate, be retrieved most quickly.

The final and larger issue is to estimate the throughput of automation: At what rate can a stream of customer orders be retrieved? It is necessary to answer this question to choose appropriate automation for a warehouse. It is safest to answer this question both by analytical methods, such as presented here, and by simulation.

Chapter 12

Carousels, A-frames, and AS/RS

12.1 Carousels

A carousel is a rotatable circuit of shelving, as shown in Figure 12.1. Instead of the order-picker traveling to the storage location, the storage location travels to the order-picker.

This conveys no advantage unless an order-picker pulls from multiple carousels, in which case he is able, in effect, to walk through multiple aisles simultaneously. Figure 12.2 shows a typical layout, in which carousels are arranged in groups, referred to as *pods*, in this case of three carousels each, and each pod is attended by a single worker. Conveyors take completed work from the carousel area to shipping.

Because the product rotates to the person, there is no need for an aisle by which to access product. This means that carousels can be installed side-by-side, which increases space utilization and also provides security for the product. However this has disadvantages as well: Most immediately the single point of access limits the rate at which product can be extracted because it is not possible to speed extraction by assigning additional workers to a carousel. Furthermore, as more product is picked from a carousel, it becomes necessary to restock it more frequently and both of these tasks must be done by the same worker. This interleaving of picks and restocks can retard the rate of picking and reduces the ability of the warehouse to respond to surges in demand.

To understand the use and behavior of carousels, we will explore a simple model that represents a carousel as m storage locations spaced at equal intervals around a loop, as in Figure 12.3.

12.1.1 Control

Retrieval of a single customer order

When the stream of customer orders is relatively sparse, one customer order can be picked before the next arrives. We have seen this in service parts distribution centers, with many slow-moving parts stored in carousels, so that each single carousel

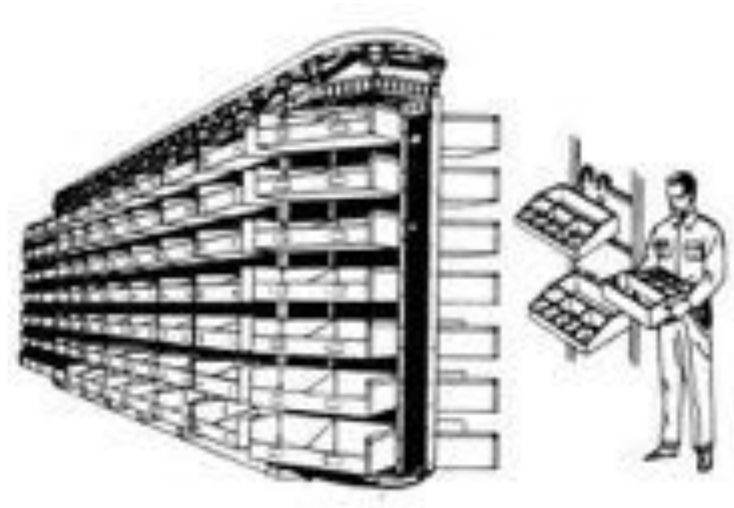


Figure 12.1: A carousel is a rotatable circuit of shelving. (Adapted from "Warehouse Modernization and Layout Planning Guide", Department of the Navy, Naval Supply Systems Command, NAVSUP Publication 529, March 1985, p 8–17.)

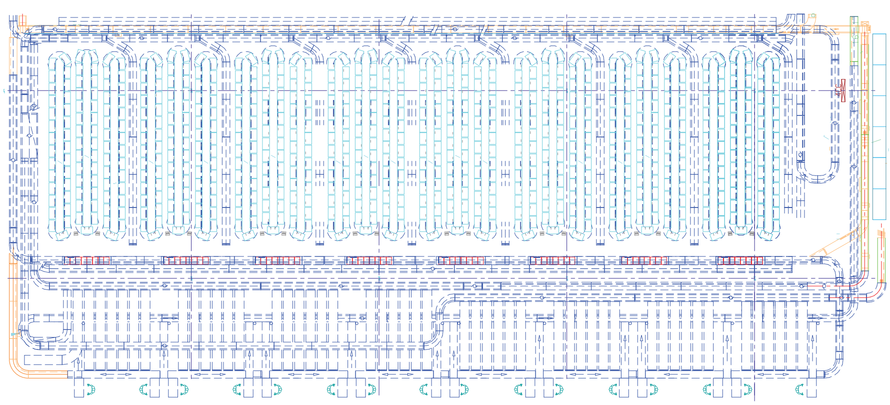


Figure 12.2: Carousels are typically arranged in pods; in this example, there are eight pods, each of three carousels. The carousels are supported by an intricate conveyor and sortation system: The vertical spurs to the right of each pod bring product to be restocked; and the horizontal conveyor at the bottom takes away completed picks to sortation and shipping.

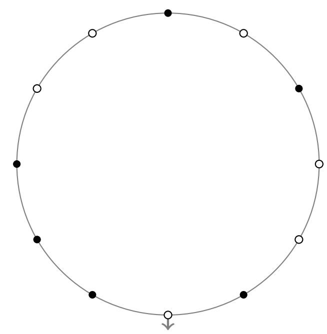


Figure 12.3: In our model of a carousel, there are m evenly-spaced storage locations. In what sequence should the locations for a customer order (represented by filled disks) be visited?

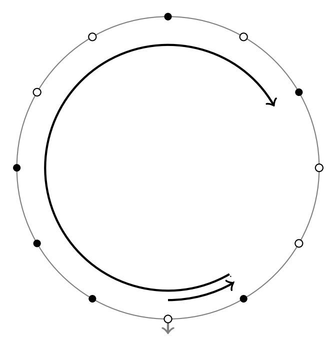


Figure 12.4: A shortest sequence to visit the required locations

experiences a sparse order stream. In this case the fundamental problem is how to retrieve a single customer order quickly; that is, given n locations on a carousel, in what sequence should they be visited to minimize travel time? This is a version of the "Traveling Salesman Problem", in this case posed on a simple loop, as in Figure 12.3.

Two insights enable us to quickly find a shortest possible rotation pattern for any order [11]. The first observation to be made is that, in the retrieval of a single order, it is never advantageous to rotate the carousel through a complete revolution or to rotate the carousel past the same location more than twice. Any such travel is excessive and so cannot be optimal. Therefore some location will be rightmost in an optimal retrieval sequence. There are only $n + 1$ candidates: The n locations to be visited and the currently indexed location, accessible to the picker. Specifying one of these locations as rightmost also determines some other location as leftmost (with an untraveled gap between them). One of these locations—rightmost or leftmost—must be the ending location. Therefore we can restrict our hunt for shortest sequence of locations to the $2n + 1$ possibilities in which each candidate location is the rightmost location in the sequence and either it or the leftmost location is the ending location. Simply enumerate these few possibilities and choose the shortest, as in Figure 12.4.

Simple as it is to compute the shortest route to all locations, it may in some cases be unnecessary. When an order is relatively large, so that it requires visiting many of the locations on the carousel, then it is unlikely that an optimal solution will reverse the direction of travel of the carousel. (Litvak [37] proved that if all locations are equally likely to be visited, then the probability of the optimal retrieval sequence reversing the carousel after j retrievals decreases exponentially in j.) Accordingly, under a heavy load, very simple strategies, such as the following, may produce sequences of retrieval that are very nearly optimal. The *Nearest Item* heuristic rotates the carousel to the nearest unvisited location required by the customer order. An even simpler heuristic is to rotate the carousel always in the same, fixed direction.

Retrieval of a sequence of orders

When a sequence of orders awaits picking, then we must account for the fact that the location at which we finish one order is where we start in picking the next, and this could be inconvenient. What we would like is a procedure that tells us how to connect the sequence of orders so that there was as little travel as necessary, both within individual orders and to connect the orders. Fortunately, this is easily done.

First we must observe that it is simple to determine the quickest way of traveling from the last location of one order to the first location of the next order: Simply evaluate the distance if rotating clockwise, compare it to the distance if rotating counterclockwise, and choose the shorter.

Similarly, we can easily determine the fastest way to pick a given order so that the picker ends at a specified location. Again, there are only two possibilities to consider: We arrive at the final destination while rotating either clockwise or else counterclockwise. If we end after rotating clockwise, we must have begun by rotating counterclockwise and then reversed direction. This is enough to specify a shortest travel path. Do the same for the case in which we end after rotating counterclockwise and then choose the shorter of the two possibilities.

Now we can use both these observations to formulate a dynamic program that determines the quickest way to pick a sequence of orders. A graphical representation of this dynamic program appears in Figure 12.5, where the current location of the order-picker is given as the left-most node in the graph and all the locations of the first order appear in the first column of nodes. The edges from the start to the locations of order 1 enumerate all the ways we would consider picking order 1, with each way leaving the order-picker at one of the locations of order 1. So: To each edge there corresponds a (shortest) sequence of picks that begins at the location represented by the node to the left and ends at the location represented by the node to the right. We specify the dynamic program by listing all the locations of each customer order and then computing the travel distance required to pick an order *and end at a specified location*. Once we have specified the details of the graph, we can compute the shortest path through it starting from the current position (the leftmost node). Then each edge of the shortest path corresponds to a pick sequence of an order or else to movement from the end of one order to the start of the next.

This process is efficient because each order visits no more than m locations and so the graph has no more than $O(mk)$ nodes and $O(m^2k)$ edges, and each edge can be

Figure 12.5: Outline of a dynamic program to compute the shortest route to retrieve a given sequence of orders from a carousel

assigned a distance in constant time. Using a shortest path algorithm, such as that of Dijkstra, gives an efficient solution.

Retrieval of a set of orders

In this version of the problem, we must decide the sequence of customer orders in addition to the sequences of retrieval within each order. Let there be k customer orders waiting to be picked.

A simple solution that is nevertheless nearly optimal can be constructed as follows. For each customer order, construct a minimum spanning tree [1], which in this case is an interval along the carousel. This can be found in a number of steps proportional to the number of locations to be visited for the order. We will refer to the endpoints of the minimum spanning interval of an order as the endpoints of that order.

Any way of picking the orders must visit the two endpoints of each order and so must travel at least the lengths of the minimum spanning intervals. Consequently, the total length of the shortest spanning intervals is a lower bound on the travel to retrieve all the items of the orders.

To reduce travel between orders, we sequence them so that the endpoint of one order is close to the start point of the next order. There are no more than $2k + 1$ points to be matched: Two endpoints for each of the k orders plus the current location of the order-picker. Choose one location as a start and then match it with the next location encountered clockwise, then match the third and fourth, the fifth and sixth, and so on. This is a candidate matching that connects endpoints of orders. Record the total length of travel required to match these endpoints; then repeat with each of the remaining $2K$ locations as the start. Choose the matching of smallest total length (which must be less than half the length of the carousel).

The matching may have failed to produce an uninterrupted sequence in which to pick orders, because the endpoints may be paired in such a way as to produce disjoint circuits (subsets of orders) rather than a single sequence. We can resolve this by constructing a minimum length tree that spans all disjoint circuits. Again, this can be done quickly, and adds no more than one revolution of the carousel beyond the current travel. Therefore the total travel cannot exceed optimal by more than 1.5 revolutions of the carousel.

Furthermore, this bound is independent of the number of orders to be picked and the number of locations visited by each order. Therefore, when the carousel is very busy—has many waiting orders—then the solution is asymptotically optimal. For more details and alternative heuristics, see [11, 48].

Even this problem is not the fullest representation of what must be done in practice, where there is typically a queue of customer orders waiting to be released to the carousel system. Additional orders may arrive while other orders are being picked. The most suitable practical strategy is to build an optimal sequence in which to retrieve orders, and then re-optimize occasionally, while ensuring that no orders wait too long in the queue.

12.1.2 Storage

The efficiency of a system of carousels is greatly determined by how it is stocked. The most important point is to balance work among carousels so that none is a bottleneck. In addition, it is important to store popular product on multiple carousels. This protects operations in case of carousel failure; and it also avoids one carousel from becoming a bottleneck to the flow of a popular item.

How one should store product within a carousel depends on the retrieval strategy. If an optimal strategy is used, then within each carousel one should store product in a so-called *organ pipe arrangement*, as in Figure 12.6. An organ-pipe arrangement is built by placing the most popular item on the carousel, the next most popular immediately adjacent (to the right in this example), the next most popular to the other side, and so on. This has been proven optimal when all orders are of size 1 [13, 50], and is probably an effective strategy whenever typical orders visit few locations on any carousel.

Finally, it should be noted that carousels are generally used as forward picking areas and so should be stocked to take into account the costs of restocking, as discussed in Chapter 8.

12.1.3 Throughput

Because a carousel offers a narrow interface to the human-side of order-picking and restocking, there is an additional problem of balance: We want to keep both the order-picker and the carousel busy. Sometimes the carousel will present an item to be picked but the picker is busy at another carousel in the pod. Similarly, the picker may be available to pick, but all the carousels in the pod are rotating. In both of these situations, the long-run average throughput of the carousels are reduced because of the imperfect coordination between worker and carousel.

There will always be some such loss of productivity because of the difficulty of keeping both picker and carousels busy. Park *et al.* showed this for a model of one picker (unrealistically) alternating picks from the left and right carousels of a 2-carousel pod [3]. They further assumed that picks were independently and identically distributed

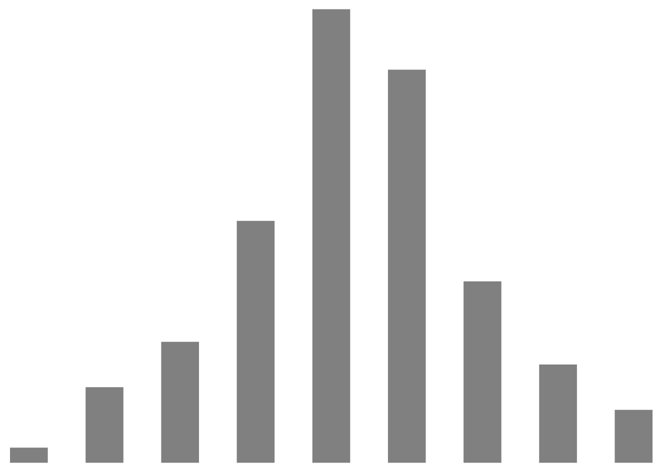


Figure 12.6: The optimal stocking strategy for a single carousel conveyor is to concentrate the most popular items together in a so-called“organ-pipe” arrangement.

around the carousel and that an infinite queue of requests awaited. Under these assumptions, the throughput of two carousels served by a single picker is only 85% of the theoretical maximum.

12.2 A-frames

An *A-frame* is an automated dispensing machine that drops items onto a conveyor, as shown in Figure 12.7.

As a conveyor moves through a tent of cartridge dispensers, the control system reserves an interval of the conveyor for one customer order. As that interval passes under dispensers, they eject the appropriate skus. As the conveyor emerges from the A-frame, it deposits the skus of one order into a box that is carried away on another conveyor.

A-frames are used when labor is expensive and product is picked in very high volumes. In addition, the skus must be suitable for dispensing: They must be small, able to withstand a fall onto a conveyor, and they must not bounce. Typical uses are for cosmetics or pharmaceuticals.

One weakness of A-frames is that all skus must be kept loaded, for if any sku stocks out, the A-frame must be halted and all order-picking stops.

With an A-frame, picking is labor-free; but there can be significant labor required to restock the device. Dispensing cartridges do not hold very much product and so one must plan to keep the A-frame refilled. Therefore an A-frame is a forward pick area such as is discussed in Chapter 8. However, it is somewhat more complicated than that, because product typically flows through several stages of replenishment such as in Figure 12.8.

Figure 12.7: An A-frame automated item dispenser, as seen from the top (start) of the conveyor. The flow rack to either side hold product to restock the A-frame, and are in turn restocked from bulk storage.

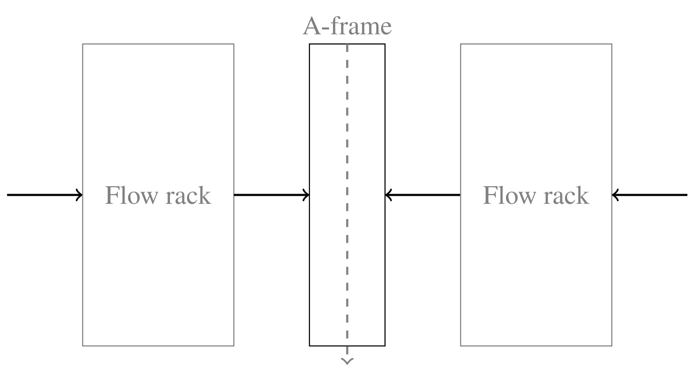


Figure 12.8: An A-frame is restocked from carton flow rack, which is itself restocked from bulk storage.

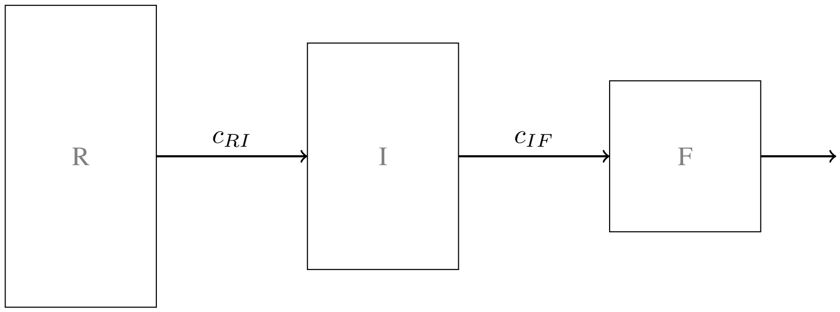


Figure 12.9: An A-frame is an example of a multi-tier forward pick area, with multiple levels of restocking. The flow rack holds an intermediate cache I of product close to the A-frame, which is the forward pick area F. The expression c_{RI} is the average cost per restock of the intermediate cache I from reserve R; and c_{IF} is the average cost per restock of the forward area F from intermediate storage I.

Which skus should be put in the forward pick area—that is, the A-frame—and in what quantities? Restocking is more complicated than in Chapter 8 and we must adjust the cost model to reflect multiple levels of restocking, from bulk storage to carton flow rack to A-frame, as indicated in Figure 12.9.

Jernigan [31] observed that a multi-tier system can be replaced with an equivalent 2-tier system by amortizing the costs of restocking intermediate storage caches, as in Figure 12.10. Here is the reasoning. Let V_I be the volume available for storage in the intermediate cache (for example, the carton flow rack) and let V_F be the volume available for storage in the forward pick area (for example, the A-frame). Restocks to each area are minimized by allocating space $\left(\frac{\sqrt{f_i}}{\sum_{j=1}^{n}\sqrt{f_j}}\right)V_I$ to sku i in the intermediate area and $\left(\frac{\sqrt{f_i}}{\sum_{j=1}^{n}\sqrt{f_j}}\right)V_F$ in the forward pick area, as derived in Theorem 8.1. Under these allocations, sku i will be restocked to the intermediate area $\sqrt{f_i}\left(\sum_{j=1}^{n}\sqrt{f_j}\right)/V_I$ times and to the forward pick area $\sqrt{f_i}\left(\sum_{j=1}^{n}\sqrt{f_j}\right)/V_F$ times. Therefore there are V_F/V_I restocks of the intermediate cache for every restock to the forward pick area.

It is now possible to optimally load a forward pick area such as an A-frame by the following steps:

1. Convert the problem to an equivalent 2-tier problem

2. Solve the 2-tier problem by the techniques of Chapter 8 (rank skus by labor efficiency and search for the most beneficial top k skus, allocating space according to Theorem 8.1.

3. Allocate the same set of skus to the intermediate cache in quantities given by Theorem 8.1.

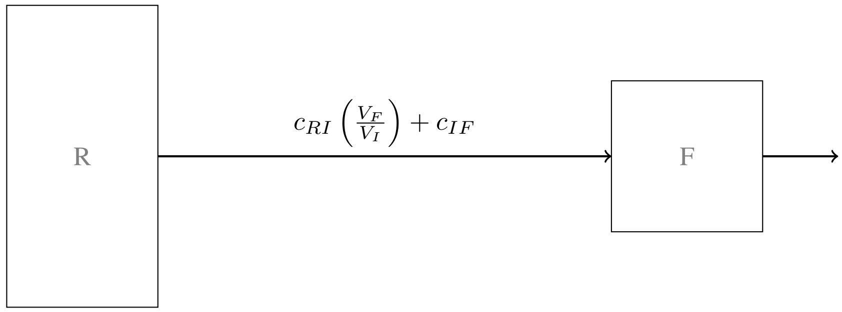


Figure 12.10: An equivalent single-tier system in which the cost of restocking the intermediate cache has been amortized over the cost of restocking the forward pick area.

We have examined one commercial A-frame installation in detail and found, unsurprisingly, that it was stocked less-than-optimally. It is very natural, after having purchased an expensive piece of automation, to try to get as many picks as possible from it; but in this case so many skus were picked from the A-frame that most were allocated too little space and consequently had to be restocked too often. We estimated that the total labor cost could have been reduced by a factor of three by moving one-fourth of the skus from the A-frame to less attractive storage, and stocking the remaining skus in optimal (restock-minimizing) quantities [31]. The extra cost to pick the skus removed from the A-frame would have been more than compensated for by the savings in labor to restock the A-frame.

12.3 In-aisle cranes, AS/RS, and their relatives

Automated storage-and-retrieval (AS/RS) devices remove humans by placing a simple robotic device within each aisle. The S/R device moves at once horizontally and vertically to convey product to or from storage. This allows aisles that are extremely narrow, indeed they are scarcely wider than the product itself; and it allows unusually high storage. Consequently, it can be useful not just where labor is expensive but also where space is expensive, such as in Singapore.

These devices are typically used to store and retrieve unit-loads and so system control and coördination is not really an issue. But, because each one represents significant capital investment and is fairly inflexible, it is important to understand their capabilities, especially throughput.

12.3.1 Throughput

This section is based on the work of Bozer and White [15, 16].

Consider an automated storage and retrieval system (AS/RS), such as shown in Figure 12.11, that consists of 10 aisles with an S/R machine within each aisle. The storage rack on each aisle-side is 240 feet long (73.1 meters) and 60 feet high (18.3

Figure 12.11: An automated storage and retrieval system (adapted from “Warehouse Modernization and Layout Planning Guide”, Department of the Navy, Naval Supply Systems Command, NAVSUP Publication 529, March 1985, p 8–17.)

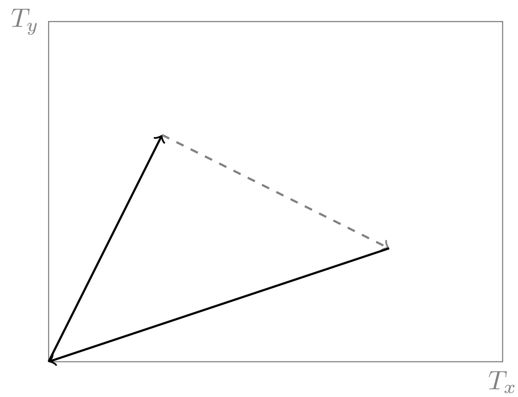


Figure 12.12: In a dual command cycle the storage-and-retrieval device puts away an item and then retrieves another before returning to the input-output point.

meters). Each unit load occupies a 5 foot by 5 foot slot (1.5 meter by 1.5 meter), and so the capacity of the AS/RS is 11,520 unit loads. The input/output (I/O) point is located at the lower left-hand corner of the rack. The S/R machine travels at 400 feet per minute (FPM, or 122 meters per minute) in the horizontal direction and 60 FPM (18.3 meters per minute) in the vertical direction and it travels both vertically and horizontally simultaneously. The time required to pick-up or deposit a unit load is 0.40 minutes.

In a *single command* (SC) cycle the S/R machine executes either a putaway or retrieval of a unit load. To execute a putaway operation the S/R machine first picks up the load at the I/O point, travels to the rack location, which we shall label by coordinates (X_1, Y_1), deposits the load into the rack, and then travels empty back to the I/O point. To execute a retrieval operation the S/R machine first travels to the rack desired rack location (X_2, Y_2), picks up the load, travels back to the I/O point, and then deposits the load (typically onto a conveyor that will transport the load to its destination). The time to execute an SC cycle is the sum of the travel time of the S/R machine, which obviously depends on the location visited, and the time it takes to execute 2 shuttle (pick-up or deposit) operations.

In a *dual command* (DC) cycle the S/R machine first stores a unit-load and then directly retrieves another, as in Figure 12.12. That is, it begins a DC cycle by first picking up a unit load, traveling to its location (X_1, Y_1), depositing the unit load in the rack, traveling empty to the location (X_2, Y_2), retrieving the unit load, and traveling back to the I/O point to deposit the second unit load. The time to execute a DC cycle is the sum of the travel time of the S/R machine on each of the three legs of the trip, which depends on the 2 locations visited, plus the time it takes to execute 4 shuttle operations, 2 for each load.

It is more productive to execute DC cycles because less time is required per operation. However, retrieval operations support downstream operations and so may be viewed as more critical. When retrieval is urgent, then it may be advisable to run only

SC cycles because DC cycles lower the rate of retrieval. Putaways are then deferred to times when retrieval is not urgent.

Average Time To Execute a SC Cycle: Random Storage Case

We begin our throughput calculations by first assuming that the unit loads are equally likely to occupy any position of the rack. This is a reasonable assumption if, as is typical in bulk storage, a policy of shared storage is employed.

Since we are only interested in estimating the time to complete a cycle, the first step is to convert the rack dimensions from distance to time: In this case we may consider our rack to be $240/400 = 0.6$ minutes long and $60/80 = 0.75$ minutes high. Each location is specified by coordinates (X, Y), where $0 \leq X \leq 0.6$ and $0 \leq Y \leq 0.8$. For example, the time location $(0.2, 0.5)$ corresponds to the physical location $(80, 30)$ (where distance is given in feet). Let H and V denote the maximum horizontal and vertical travel times, respectively. In our example $H = 0.6$ and $V = 0.8$.

We now compute the average S/R traveling time component of an SC cycle, which we denote as T_{SC}^U. (Here, U denotes the uniform distribution.) It is twice the time it takes to complete one leg of the cycle. Since T_{SC}^U is a non-negative random variable,

$$E[T_{SC}^U] = 2\int_0^V (1 - F(t))\ dt,$$

where $F(t) \equiv \Pr\{T_{SC}^U \leq t\}$, $0 \leq t \leq V$. To determine the function $F(t)$ we note that

$$\Pr\{T_{SC}^U \leq t\} = \Pr\{max(X, Y) \leq t\} = \Pr(X \leq t) \cdot \Pr(Y \leq t),$$

since by assumption the locations of X and Y are independent random variables. The marginal distributions of X and Y are each uniform; that is, $X = U(0, H)$ and $Y = U(0, V)$. Since $H < V$ note that $\Pr(X \leq t) = t/H$ when $0 \leq t \leq H$ and $\Pr(X \leq t) = 1$ when $H \leq t \leq V$. Thus,

$$E[T_{SC}^U] = 2\{\int_0^H \left(1 - t^2/HV\right)\ dt + \int_H^V (1 - t/V)\ dt\} \equiv T\left(1 + Q^2/3\right),$$

where $T = max(H, V)$ and $Q = min(H, V)/T$. (The formula is valid if $H > V$, and this is why it is stated in this normalized form.) The parameter Q is called the *shape* parameter of the rack. If $Q = 1$, then the rack is referred to as *square-in-time*. In our example, $T = 0.8$, $Q = 0.75$ and $E[T_{SC}^U] = 0.95$ minutes. It takes 0.80 minutes to complete 2 shuttle operations, and so the total average time to complete an SC cycle is 1.75 minutes, which equals 34.3 operations per hour, a measure of throughput. Note that this throughput measure represents a maximum in that it does not account for idle time or down time due to maintenance. System (maximum) throughput would be 343 operations per hour.

Average Time To Execute a DC Cycle: Random Storage Case

Once again we shall assume that the unit loads are equally likely to occupy any position of the rack.

Stochastically, the average S/R traveling time to complete a DC cycle, which we denote as $E[T_{DC}^U]$, is equivalent to $E[T_{SC}^U]$ plus the time it takes the S/R machine to travel from the putaway location (X_1, Y_1) to the retrieval location (X_2, Y_2). Let Z denote this second (random) time component. Note that $Z = \max\{|X_1 - X_2|, |Y_1 - Y_2|\}$. Due to the assumption of independence,

$$\Pr\{Z \leq t\} = \Pr(|X_1 - X_2| \leq t) \cdot \Pr(|Y_1 - Y_2| \leq t).$$

To compute $\Pr(|X_1 - X_2| \leq t)$ think of (X_1, X_2) as being randomly distributed in the square $[0, H] \times [0, H]$. The proportion of the area that defines the set $\{|X_1 - X_2| \leq t\}$ can be verified graphically as $1-\left((H-t)^2/H^2\right)$ for $0 \leq t \leq H$, which represents the probability of landing a point in this set under the uniform distribution. By replacing H with V we obtain the expression for $\Pr(|Y_1 - Y_2| \leq t)$, $0 \leq t \leq Y$. Substituting these expressions into the above integral and carrying out the integration results in

$$E[Z] = T\left(1/3 + Q^2/6 - Q^3/30\right).$$

Adding this expression to $E[T_{SC}^U] = T\left(1 + Q^2/3\right)$, we obtain the expression

$$E[T_{DC}^U] = T/30\left(40 + 15Q^2 - Q^3\right).$$

In our example, $E[T_{SC}^U] = (0.80/30)\left(40 + 15\,(0.75)^2 - (0.75)^3\right) = 1.28$ minutes. It takes 1.60 minutes to execute the 4 shuttle operations, and so the total average time to complete a DC cycle = 2.88 minutes, or 20.8 DC cycles per hour or 41.6 operations per hour. Note that $2.88 < 2\,(1.75)$, as expected, and that the throughput, measured in operations per hour, increased by 21% above the SC throughput.

System Throughput for Mixed Cycles: Random Storage Case

We illustrate this calculation with an example. Suppose that during the pick cycle 75% of the operations will be retrievals and 40% of the putaways are executed on a DC cycle. To compute the average throughput we compute the average time it will take to complete 100 operations, as follows. On average, there will be 75 retrievals and 25 putaways, of which 10 putaways will be executed on a DC cycle. Thus, on average, there will be 10 DC cycles (executing 10 retrievals and 10 putaways) and 80 SC cycles (executing the remaining 65 retrievals and 15 putaways) to complete the 100 operations. This translates into an expected time of $10\,(2.88) + 80\,(1.75) =$ 168.8 minutes or 1.69 minutes per operation, which equates to 35.5 operations per hour for a maximum system throughput during the pick cycle of 355 operations per hour. Again we have not accounted for maintenance and downtime in this calculation.

An alternative view of this calculation is to note that 20% of the time an operation will be performed on a DC cycle, which takes $2.88/2 = 1.44$ minutes on average to complete, and 80% of the time an operation will be performed on an SC cycle, which takes 1.75 minutes on average to complete. Thus, the expected time to complete an operation will be $0.20\,(1.44) + 0.80\,(1.75) = 1.69$ minutes, as before.

Class-based Storage

The idea of class-based storage is simple: increase throughput by assigning the most-frequently requested skus to the best locations on the rack face. There must be adequate capacity to ensure that a sku assigned to the best locations can actually be placed there. The simplest method for accomplishing this task is to use dedicated storage. That is, each sku is assigned a specific slot location(s). However, dedicated storage requires more space than random storage, since it must accommodate the maximum inventory. If the maximum inventory of a sku exceeds one slot, then there will be significant periods of time when there will be empty slots not in use.

We consider a hybrid dedicated/random storage policy. Here is an example of how it may be implemented for our rack system. Consider a rectangular sub-region A whose four vertices are $(0, 0)$, $(0.3, 0)$, $(0.3, 0.4)$, and $(0, 0.4)$. (Region A is a 50% scaled down version in each dimension of the original rack.) Region A has parameters $T = 0.4$ and $Q = 0.75$, and thus occupies 25% of the slots. The most frequently requested skus are assigned to region A, and the other skus are assigned to the remaining slots, which are designated as region B. When a unit load for a sku is received it is put away in the closest open location within the appropriate region. If we use a policy of shared (random) storage within each region then we can view the locations of stock as following a distribution within each region. For sake of discussion let us assume that enough skus can be assigned to region A so that 70% of all transactions involve a sku within this region. How does class-based storage average throughput?

We shall compute the average throughput assuming only SC cycles are used, which bounds system throughput and will be the case during peak picking periods. The distribution of location within each region is reasonably modeled by the uniform distribution, but clearly the overall distribution is no longer uniform. Let t_i denote the time to execute an SC cycle for the i^{th} transaction, $i = 1, 2, \ldots, N$. For N sufficiently large the average time to execute an SC cycle will be $\sum_{i=1}^{N} t_i/N$. The average can also be represented as

$$\sum_{i=1}^{N} t_i/N = \left(\frac{\sum_{i\in A} t_i}{N_A}\right)\left(\frac{N_A}{N}\right) + \left(\frac{\sum_{i\in B} t_i}{N_B}\right)\left(\frac{N_B}{N}\right),$$

where we have used the notation $i \in A$ or $i \in B$ to represent the region in which the transaction will be completed, and N_A and N_B denote the number of transaction to the respective regions. Let CB indicate "class-based"; then when N is sufficiently large,

$$\begin{aligned} E[T_{SC}^{\mathbf{CB}}] &= E[T_{SC}^{\mathbf{CB}}|A]\Pr(A) + E[T_{SC}^{\mathbf{CB}}|B]\Pr(B) \\ &= E[T_{SC}^{U}|A]\Pr(A) + E[T_{SC}^{U}|B]\Pr(B), \end{aligned} \tag{12.1}$$

where $E[T_{SC}^{CB}|C]$ is the expected time to complete an SC cycle given that the transaction occurs in regions A or B. Of the four expressions on the right-hand side of Equation 12.1, two are pre-specified (namely, $\Pr(A)$ and $\Pr(B)$) and 1 is easily obtained from using the formula $T(1 + Q^2/3)$ for appropriate choices for T and Q. Thus, the only potential complication is computing $E[T_{SC}^{U}|B]$. Note that region B is not rectangular so we cannot use any previously developed formula.

While the desired expectation can be formulated and computed via a double-integral, a much easier approach is based on the fact that the conditional expectation formula expressed in Equation 12.1 still applies under any distribution, in particular the uniform distribution. We know the value of $E[T^U_{SC}]$ for the whole rack, which is the left-hand side of Equation 12.1, and we know $E[T^U_{SC}|A]$. Under the uniform distribution $\Pr(A)$ and $\Pr(B)$ are equal to their respective proportion of the total rack face area. Thus, $E[T^U_{SC}|B]$ is determined via Equation 12.1, too. To illustrate the calculation, first compute $E[T^U_{SC}|A] = 0.40[1 + (0.75)^2/3] + 0.8 = 1.275$ minutes. From our previous calculation we know that $E[T^U_{SC}] = 1.75$ minutes. Since $\Pr(A) = 0.25$ and $\Pr(B) = 0.75$ we have that

$$1.75 = (1.275)(0.25) + E[T^U_{SC}|B](0.75),$$

which implies that $E[T^U_{SC}|B] = 1.908$ minutes. The desired average travel time is now computed as

$$E[T^{CB}_{SC}] = (1.275)(0.70) + (1.908)(0.30) = 1.465, \tag{12.2}$$

which translates into a throughput of 41 operations per hour for a 20% increase in productivity. Note that once the key conditional expectation numbers (1.275 and 1.908) have been calculated, Equation 12.2 is used to determine throughput under any probability distribution for regions A and B. The method of analysis presented here will also apply when there are more than two regions.

12.4 On the lighter side

We have heard some amusing stories about setting up carousels. The problem has usually been carelessness. The most dramatic one is of a site that was loading some thirty carousels with auto parts, which are high value, slow moving, and heavy. Especially heavy. As we heard it, each carousel was loaded one location at a time, top to bottom, then rotated one position, and so on. Of course this meant that, at some point, one long side of the current carousel was fully loaded and the other was completely empty. Inevitably, carousel number twenty-nine tipped, crashing into fully-loaded number 28, and so on in majestic, slow-motion disaster.

More recently, we visited a warehouse that distributes service parts for large equipment that is in constant use. This equipment cannot be allowed to sit idle while waiting for spare parts and so rapid, reliable service is essential. The warehouse had installed carousels to get space utilization by installing them close by each other. But they quickly realized that with any loss of electricity, for example during a storm, product would become inaccessible. To protect against this, they had to leave an aisle between each pair of carousels. The result was that they paid for specialized equipment but realized no space efficiency.

A final carousel story is from the designer of a software control system for a very fine operation. He was concerned that the hardware move quickly enough to keep order-pickers fully occupied and so rotated each carousel at the high end of the recommended velocity range. During trial runs he noticed some empty storage slots, which was not unusual before restocking; but he became alarmed when the empty slots began to increase quickly. It seems the boxes in which product was stored were just slick enough that, as the carousel rotated a shelf around the end, it (the box) might shoot off the carousel and go skidding across the warehouse floor!

12.5 Questions

Question 12.1. *What is the main reason for considering automation? What is the main disadvantage of automation?*

Question 12.2. *True or false? Explain your reasoning.*

- *Carousels are well-suited to product that sells in large quantities.*
- *Carousels are well-suited to product that is frequently requested.*
- *An AS/RS will increase efficiency if it performs all put-aways separately from all retrievals.*
- *An A-frame is* not *a forward pick area because the cost per retrieval is zero.*

Question 12.3. *For each of the following forms of automation, explain their main advantages and disadvantages: carousels, AS/RS, A-frame, distributed robots (Kiva System).*

Question 12.4 (Exploration)**.** *What types of skus should be stored in carousels: Large ones or small ones? Ones that move few cases or ones that move many cases? Ones that are infrequently requested or ones that are frequently requested? If possible, base your answer on models.*

Question 12.5. *In each case explain why the warehouse action described below is probably unwise.*

- *Installing a single carousel*
- *Storing all the most popular skus on the same carousel*
- *Assigning each item in the carousels to a single location*

Question 12.6. *Prove that the retrieval path generated by the Nearest Item heuristic ("travel to the closest unvisited location") is always strictly less than one complete revolution of the carousel when retrieving a single customer order.*

Question 12.7. *Prove that the retrieval path generated by the Nearest Item heuristic ("travel to the closest unvisited location") is never greater than twice the optimal distance when retrieving a single customer order.*

Question 12.8. *Make an example in which two customer orders are to be retrieved from a single carousel, but it is suboptimal to retrieve them by minimizing travel for the orders individually. How extreme can you make this example? That is, how large can you make the wasted travel compared to optimizing the total travel by the dynamic program of Figure 12.5?*

Question 12.9. *Prove that the shortest matching of endpoints of a set of orders does not exceed half a revolution of the carousel.*

sku	Picks	Flow
A	200	35.0
B	100	8.0
C	25	40.0
D	85	7.0

Table 12.1: Question 12.11 refers to these skus, which are candidates for storage in an A-frame. (Flows have been scaled as a fraction of the volume available in the A-frame.)

Question 12.10. *What are the most convenient storage positions on a carousel? Explain.*

Question 12.11. *Consider an A-frame that is restocked from carton flow rack, which is itself restocked from reserve. Restocks to the A-frame average about 1.0 minutes each and restocks to the flow rack average about 4.0 minutes each. The flow rack holds about twice the volume of A-frame.*

Suppose the SKUs of Table 12.1 are being considered for storage in the A-frame. Each can be picked either from reserve at an average of 2.0 minutes per pick, or else dispensed at no cost from the A-frame.

A. Which skus have strongest claim to space in the A-frame if allocated space to minimize restocks? Rank the skus from most to least suitable.

B. What would be the net benefit of storing SKUs A and B in the A-frame in volumes to minimize restocks?

C. How would your answer to part B change after purchasing additional flow rack to doubling the total volume of intermediate storage? Assume that no other costs change significantly.

Question 12.12. *Why would you* not *want to set up a mini-load so that all input arrives at one side and all output departs the opposite side?*

Question 12.13. *Our analysis of the throughput of an AS/RS system was based on an assumption about how product was assigned storage locations. Explain why that assumption is likely to represent the worst-case. What is the implication for our estimation of throughput?*

Question 12.14. *The storage rack of a Unit Load AS/RS system is 300 ft long and 75 ft high. The I/O point is located at the lower left-hand corner of the rack. The S/R machine travels at 300 FPM in the horizontal direction and 60 FPM in the vertical direction. The time required for a pick-up or deposit operation is 0.25 minutes.*

A. What is the expected time to complete a SC cycle, assuming shared (random) storage? What is the corresponding throughput per hour, measured in operations per hour?

B. What is the expected time to complete a DC cycle, assuming shared (random) storage? What is the corresponding throughput per hour, measured in operations per hour? What is the percent increase in throughput if one uses a DC cycle?

C. Assume that during peak activity 75% of the operations are retrievals, 20% of the retrievals are performed on a DC basis, and 8% of the operation time must be set aside for maintenance. How many operations per hour can an S/R machine handle during peak activity?

D. A preliminary stock assignment plan has been worked out that would concentrate approximately 80% of the activity with a rectangular sub-region A whose northeast vertex is $(100, 37.5)$*, measured in feet, and whose southwest vertex is the I/O point* $(0, 0)$*. Within region A and its complement, region B, the activity is uniformly distributed. What is the expected percent increase in throughput? Assume all commands are single-cycle.*

Question 12.15. *A Unit Load AS/RS system consists of 10 aisles with an S/R machine in each aisle. The storage rack is 400 feet long and 80 feet high. The I/O point is located at the lower left-hand corner of the rack. The S/R machine travels at 300 FPM in the horizontal direction and 60 FPM in the vertical direction. The time required for a pick-up or deposit operation is 0.25 minutes.*

A. What is the expected round-trip SC cycle time, assuming a policy of shared (random) storage?

B. What is the expected round-trip SC cycle time, assuming class-based storage with a square-in-time rectangular sub-region A with length 100 feet and a hit rate of 50%?

C. What is the expected increase in throughput by using class-based storage?

D. What percent of the time with the S/R machine spend actually traveling (on average)?

E. What is the expected round-trip DC cycle time, assuming a uniform distribution?

F. Assume during peak activity that 80% of the operations are retrievals and that 100% of the putaways are performed on a DC basis. How many operations per hour can each S/R machine handle during peak activity without being utilized more than 90%?

G. Assume during peak activity that 70% of the operations are retrievals and that 60% of the retrievals are performed on a DC basis. How many operations per hour can each S/R machine handle during peak activity without being utilized more than 85%?

Part V

Special topics

Some warehouses serve special needs and their design and operation raise special issues or present familiar issues in new light.

Chapter 13

Crossdocking

Crossdocks are high speed warehouses.

If an arriving item has already been requested by a customer there is no need to store it as anticipation inventory; instead, the item can move directly from receiving to shipping, without intermediate storage and retrieval. Thus the item can move much more quickly through the facility and the most costly part of warehouse labor can be avoided.

In a high-volume crossdock the turnover times may be measured in hours. To support this velocity of movement, a crossdock may be nothing more than a slab of concrete with a roof and walls punctuated with doors for trailers. Freight is pulled off arriving trailers, sorted and loaded onto departing trailers without intermediate storage.

There is little or no storage provided in a crossdock because items do not stay long enough; but there is generally a lot of material-handling equipment, such as forklifts and pallet jacks, to move freight. Labor is frequently the main cost and it is devoted to unloading incoming trailers, moving the freight to the appropriate outgoing trailers, and loading. Consequently, the issues within a crossdock are those of material-handling and product flow rather than location and retrieval.

13.1 Why crossdock?

The biggest reason to have a crossdock is to reduce transportation costs. This can be achieved by consolidating multiple shipments so that full truck loads can be sent.

The Home Depot is a major retailer and the largest user of *Less-than-Truck-Load (LTL)* shipping in North America. (LTL means sending shipments that do not fill a trailer and so are not economical to send by themselves. Instead, an LTL freight company consolidates many such shipments and so achieves efficiencies.) At the present writing, LTL costs about twice the cost of *Truck Load (TL)* shipping, so there is a strong incentive to fill trailers. The Home Depot has begun doing this by having vendors ship full trailers to its crossdock. (The trailers are full because they hold product for many stores.) At the crossdock the product is sorted out for individual stores and consolidated with product from other vendors bound for the same store. The result is that each

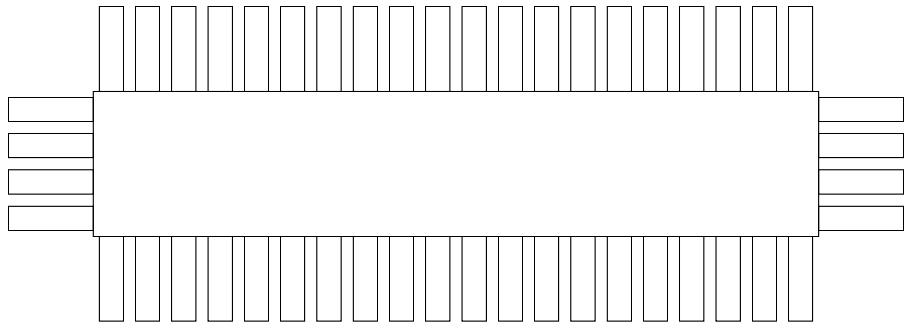


Figure 13.1: View from above of a typical high-volume crossdock, which receives freight, sorts, and disgorges it. Each door is devoted to either arriving trailers, which are unloaded, or to departing trailers, which are loaded. Ideally, freight should flow directly across the dock rather than along its length.

store has enough freight that it or it and a few close neighbors generate a full truck load from the crossdock. The result can be considerable savings.

Additional benefits include less inventory (because all product flows right through) and less labor (because product does not have to be put away and later retrieved).

13.2 Operations

Most crossdocking freight terminals are laid out as long, narrow warehouses with doors around the perimeter. Figure 13.1 illustrates a typical terminal with trailers parked at doors around the perimeter. Terminals range in size from fewer than 10 doors to more than 500 doors.

Inside a terminal, a variety of material handling methods is used to transport freight. Forklifts and palletjacks carry heavy or bulky items, and carts transport smaller items. In addition, large terminals may have draglines, which circulate carts around the inside perimeter of the dock.

There are two types of doors in a terminal: *receiving*, or *strip*, doors, where full trailers are parked to be unloaded, and *shipping*, or *stack*, doors, where empty trailers are put to collect freight for specific destinations. Once established, the designations of these doors do not change, although the trailers parked at them will. A shipping door always receives freight for the same destination. A receiving door may be occupied by any incoming trailer, regardless of its origin or contents.

Arriving trucks may deliver their trailers directly to an unoccupied receiving door; or, if none is available, they may place them in a queue. After the trailer is backed into a receiving door, a worker unloads the freight. After unloading items of a shipment onto a cart, the worker walks to the destination trailer and loads the items into that trailer; or he places the cart on the dragline, if the terminal is so equipped. To handle pallet loads, the worker uses a palletjack, or hails a forklift driver, or finds a forklift and delivers the load himself, if union rules permit.

After a trailer has been completely stripped, a driver replaces it with another incoming trailer from the queue of trailers waiting to be stripped. After an outgoing trailer has been filled, a driver replaces it with an empty trailer to be filled with freight for the same destination.

13.3 Freight flow

The patterns of freight flow within a terminal—and therefore the work—are determined by:

Layout by which we mean the specification of doors as either receiving or shipping doors and the assignment of destinations to the shipping doors.

Geometry The shape of a terminal determines the travel distances between doors and the susceptibility to congestion. (For example, narrow docks tend to be more congested because workers have less room to manœuver.)

Material handling systems For example, palletjacks are slower than forklifts, but they may be more available; draglines reduce walking time, but can impede forklift travel.

Freight mix For example, terminals having a higher mix of pallet freight require more forklift travel than those receiving a majority of carton freight.

Scheduling In real time, the dock supervisor determines freight flow patterns by assigning incoming trailers to receiving doors.

Changing the geometry or material handling systems of a terminal is expensive; changing the freight mix is a marketing decision with implications outside the terminal. The two remaining ways to take work out of the system—change the layout or change the scheduling—are inexpensive. In particular, the layout can be changed simply by changing the labels on the doors of the crossdock.

There are two kinds of doors on a typical crossdock: Those reserved for outgoing trailers (for example, the "Miami trailer") and those reserved for incoming trailers. The outbound doors are reserved for specific destinations but the incoming doors are not so specific and may be used by any incoming trailer (because, while departures are scheduled to specific destinations, the terminal does not have full control over arrivals).

13.3.1 Congestion

As more freight flows across a dock, congestion increases, which interferes with the flow.

There are several distinct types of congestion on a crossdock:

Competition for floor space: Freight may be docked outside a receiving door if, for example, it consists of many unpalletized cartons going to the same shipping door. Then there is an incentive to accumulate it all so that fewer carts must

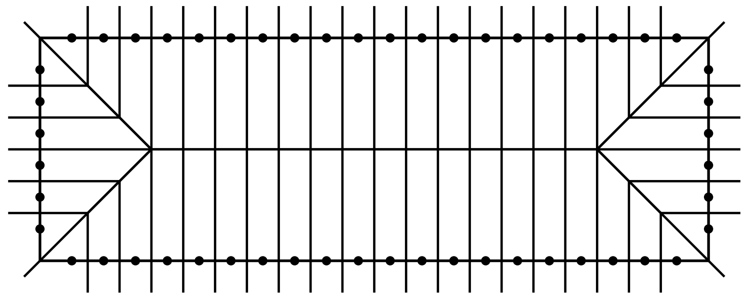


Figure 13.2: There is less floor space per door at outside corners and therefore more likely to be congestion that retards movement of freight.

travel to the destination door. On the other hand, freight is very likely to be docked outside a shipping door while the loader figures out how to pack the trailer tightly. When several nearby doors compete for space to dock freight, some invariably interferes with other traffic. At the very least, it takes longer for a worker to manœveur through the docked freight.

The effects of docked freight are most severe near the inside corners of the dock, where there is less space per door, as shown in Figure 13.2.

The need to dock freight suggests that busy outgoing trailers be parked away from the corners of the dock.

Interference among fork lifts: Despite the intention of moving freight simply "across the dock", most doors will be to the left or right of a door with an incoming trailer and so a significant amount of freight must travel along the length of the dock. Most crossdocks set up two forklift "highways", one along each long side of the dock. (It is a good idea to set up two so that, when one is blocked, some freight can still flow.) However, the flow of forklifts back and forth along the length of the dock may be interrupted by forklifts making left hand turns into doors with outgoing trailers. This effect can be reduced by parking busy outgoing trailers away from the very middle of the dock (which is also the most convenient location). Note that this works opposite to convenience, which tends to push busy outgoing doors towards the middle of the dock.

Competition for drag line capacity: Each door receiving arriving trailers will need empty carts from the dragline and, after loading a cart, will need empty cart positions on the dragline. This means that there will be diminished dragline capacity downstream of this door. If the door is far from a busy outgoing door then the region of diminished capacity can be large. This creates an incentive to intersperse incoming doors with outgoing doors. In particular, this suggests that current practice, which is to create large banks of incoming doors, reduces the capacity of the dragline.

13.4 Design

13.4.1 Size

The first decision in designing a crossdock is “how many doors?”.

Generally doors are devoted to one of two types of trailers:

- Incoming, from which freight must be removed; and

- Outgoing, in which freight must be loaded

It is easier to unload than to load. A loader must try to get a tight pack and so may have to dock freight and this double-handling slows him down. A good rule of thumb is that it takes twice as much work to load a trailer as to unload one.

To achieve frictionless flow, the capacity for flowing freight into the dock must be balanced with the capacity to flow freight out of the dock. Accordingly, one should plan to have twice as many outgoing doors as incoming doors. Alternatively, one can balance the rates of flow by assignment or workers. For example, if there are equal numbers of incoming and outgoing doors, balance can be achieved by assigning twice as many workers to load. Note, however, that crossdocks with many doors are generally less efficient than crossdocks with fewer doors. The reasons are as follows. A door can only have a few near neighbors on a dock and so a dock with more doors means that each door is likely to have few more near neighbors but many more distant neighbors. This means that in general freight must move farther across a large dock. Consequently, labor costs are generally higher at larger docks.

An additional factor is that on larger docks more freight flows past the central doors, which are the most important because they tend to be close to many doors. In fact, the total flow of freight past a centrally-located door tends to be proportional to the *square* of the total number of doors. Therefore a dock with twice the doors tends to have 4 times the congestion in front of its central doors, which diminishes their value.

This follows from the following simple model: Imagine a rectilinear dock as a line with $2n$ doors (numbered from left to right), and assume that equal amounts of freight move between every pair of doors. Then the flow into any door is of intensity $O(n)$. But the total flow passing the area between door i and $i+1$ is $i(2n-i)$, which means that the greatest total flow passes by the middle of the dock, door n, past which flows $O(n^2)$ units. But these central doors are exactly those that are nearest to most other doors and therefore are the best locations! Thus, as a dock design grows in length, the lengthwise traffic past the central doors increases rapidly while traffic directly across the dock remains unchanged. Increased traffic means congestion, which helps explain why docks can lose their efficiency as they grow. There are few docks larger than about 200 doors. Most are 80–120 doors long.

Do not forget to allow enough parking space in the yard for two trailers for every door. This means that for each origin or destination you can have a trailer at the door plus one full and one empty in the yard. This helps you handle surges in freight flow.

13.4.2 Geometry

What is a good shape for a crossdock? In general, one wants to enable efficient flow of freight from incoming trailers to outgoing trailers.

Typically, a crossdock is a long rectangle, with doors for trailers around it. The capacity of a dock is increased if it has many doors, but without being too close together so that trailers (outside) or freight (inside) interfere with one another.

Figure 13.3: A typical crossdock is built in the shape of the letter I (actually, an elongated rectangle), so that freight can flow across from incoming trailers to outgoing trailers.

A typical dock, such as illustrated in Figure 13.3, is generally around 120 feet wide (36.6 meters). This is to allow freight to be staged on the floor. A standard (large) trailer is 48 or 53 feet long (14.6 or 16.2 meters) and a "pup" is 28 feet long (8.5 meters); all are 9 feet wide (2.7 meters). The width of the dock should include enough space for the trailer on each side of the dock to stage its freight (about 100 feet total, or 30.8 meters) plus allow space for travel along the length of the dock (for example, two aisles, each 10 feet wide, or about 3.0 meters). We have seen docks as narrow as 80 feet (24.4 meters), but this is practical only when it is possible to avoid staging most freight, such as when the material is palletized and also easily stackable and may be loaded in any order. If a dock is much wider than this, it just adds to the travel time to move the product from incoming trailer to outgoing trailer.

A dock does not have to be shaped like the letter I. For example, Figure 13.4 shows docks in the shapes of an L, U, T, and H. But, as illustrated in Figure 13.5, every corner in a dock reduces effective capacity:

- On the outside of a corner you lose floor space per door on which to dock freight. This increases congestion on the dock, which interferes with the flow of freight.

Figure 13.4: Crossdocks have been built in a variety of shapes. Clockwise from upper left: An L-shaped terminal of Yellow Transport; a U-shaped terminal of Consolidated Freightways; a T-shaped terminal of American Freightways; an H-shaped terminal of Central Freight

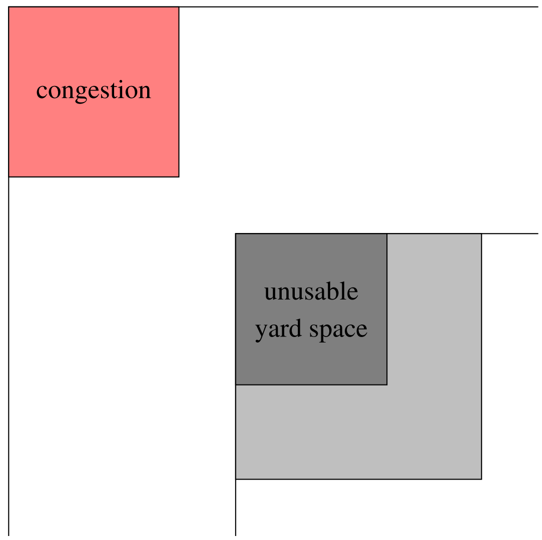


Figure 13.5: An external corner reduces floor space per door and so can create congestion (red area). And trailers cannot be parked close to an internal corner (gray area) and so the crossdock must be larger to accommodate a given number of doors—which means freight must travel further.

(This can be ameliorated by putting doors there that have less need to dock freight, such as those that move full trailers of all pallets to a single location.)

- On the inside of a corner, you lose door positions because trailers will interfere with each other in the yard. Because doors are lost, the dock must be longer to accommodate a given number of doors, which means that on average freight will have to travel farther to cross the dock. Thus, for example, freight has to travel farther to cross an H-shaped dock, with four inside corners, than to cross an I-shaped dock. (Because the door positions will be lost anyway, inside corners are a good place to locate administrative spaces or hazardous materials storage.)

It is hard to make generalizations independent of specific bills of lading; but in general an L-shaped crossdock is inferior: It incurs the costs of one inside and one outside corner but without getting anything in return. The result is that freight must travel farther because the dock must be longer from end to end to make up for lost doors at the inside corner. Furthermore, there is congestion at the outside corner.

The same observations hold even more strongly for a U-shaped dock.

An X-shaped or a T-shaped dock also incur corner costs but they have a compensating benefit: The longest distance from door-to-door is less than that for an I-shaped or L-shaped dock with the same number of doors.

13.5 Trailer management

One can reduce labor costs in a crossdocking freight terminal by parking incoming and outgoing trailers so that freight can be efficiently moved across the dock. For example, if much of the freight flowing through the terminal is bound for Miami, the Miami trailers should probably be parked in a convenient location. The challenge is to formalize the notion of "convenient"; then labor-reducing door assignments can be made with optimization models based on the geometry of the terminal, the material handling systems within, and the mix of freight passing through.

13.6 Resources

More detailed technical treatment of these issues can be found in [8, 9].

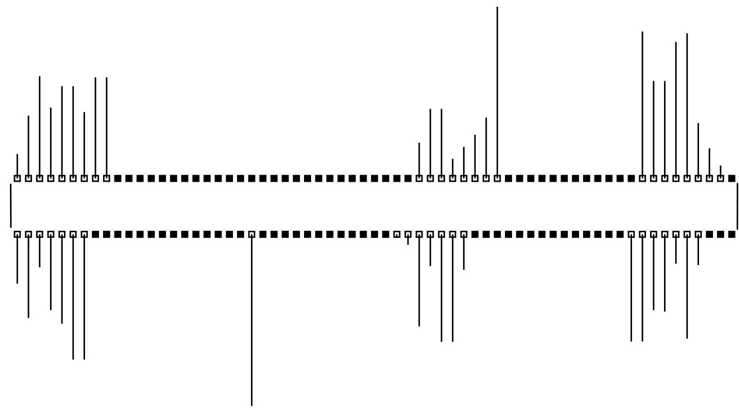


Figure 13.6: Critique this layout of an actual LTL crossdock (Question 13.8).

13.7 Questions

Question 13.1. *Crossdocks remove two of the fundamental warehouse activities: What are they and what enables crossdocks to omit these activities?*

Question 13.2. *For what reasons is freight likely to be docked at the door of an outgoing trailer? An incoming trailer?*

Question 13.3. *Explain the cost of a corner in a crossdock.*

Question 13.4. *Where are the most convenient doors in a rectangular crossdock?*

Question 13.5. *What is a typical width for a crossdock? Explain the logic behind this width?*

Question 13.6. *Give two reasons why an I-shaped dock is almost certainly more efficient than an L-shaped dock for the same number of doors.*

Question 13.7. *Crossdocks are generally set up so that outside (tractor) traffic circulates counterclockwise. Can you guess why?*

Question 13.8. *Figure 13.6 shows the layout and flows on a crossdock on which freight moves mostly by forklift truck and by a dragline running counterclockwise. Each solid mark indicates a door reserved for arriving trailers and each bar represents the amount of freight bound for the destination for which that door is reserved. Describe three problems the following crossdock will likely experience and explain why.*

Part VI

Measuring warehouse performance

It hard to judge the effectiveness of a warehouse based on daily observation because events are fast-paced and distributed over a large area. But it is possible to infer details of warehouse activities from a history of customer orders together with a map of the warehouse. This can then be used to compare with other warehouses to identify and fix inefficiencies.

There are two main ways of evaluating a warehouse and each has strengths and weaknesses. One way is to compare a warehouse with an ideal expressed as a mathematical model. Issues can be clarified by basing analysis on an abstract idealization of the actual warehouse; but usefulness depends heavily on appropriateness of the model.

Another way to evaluate a warehouse is to compare not with an ideal but with actual warehouses to suggest standards of performance proven to be achievable. But this assumes that other warehouses, possibly including competitors, are willing to share best practices.

Chapter 14

Activity profiling

A warehouse is a complicated and busy place and it can be hard to get an accurate sense of what is happening. *Warehouse activity profiling* is the careful measurement and statistical analysis of warehouse activity. This is a necessary first step to almost any significant warehouse project: Understand the customer orders, which drive the system.

14.1 Basics

There are several simple statistics that are the first things to learn about a warehouse. Each gives some hint as to the economics of that warehouse; but these are to be treated carefully because many are simple averages and so can be misleading. Their primary advantage is to summarize the warehouse environment succinctly (but at the cost of hiding much complexity).

The key facts to learn include the following.

- What is the business? Who are the customers? What are the service requirements? What special handling is required?
- Area of warehouse (a larger warehouse will require either more labor or more equipment to move product) and types of storage, material handling equipment
- Average number of skus in the warehouse (a rough indicator of complexity of work)
- Average number of pick-lines shipped per day
- Average number of units (pieces, cases, pallets) per pick-line
- Average number of customer orders shipped in a day (more shipments mean a larger shipping dock and/or more labor)
- Number of order-pickers and how many shifts devoted to pallet movement, to case-picking, and to broken-case picking (suggests where to look for opportunities to reduce operating expenses, which are primarily due to order-picking)

- Average number of shipments received in a day (more shipments mean a larger receiving dock and/or more labor)

- Average rate of introduction of new skus (it is difficult to maintain a rational storage policy when the population of skus changes quickly)

- Seasonalities

14.2 Warehouse activity profiling

14.2.1 ABC analysis

It is a truism of engineering that only a few things within any operation account for most of the activity. This is encoded in folklore by various rules-of-thumb, such as 80-20 rules (for example, "Twenty percent of the skus account for 80 percent of the activity"); or in ABC analysis, which simply classifies skus as A (the small fraction of skus that account for most of the activity), B (moderately important), or C (the bulk of the skus but only a small portion of the activity).

One of the first things to know about any warehouse is what skus matter. This is usually a simple matter of ranking the skus by various criteria. This helps reveal the contours of the economic terrain within the warehouse.

It is a popular misconception that an ABC analysis refers exclusively to the ranking of skus by *dollar-volume*, which is dollars/year in sales of each sku. This is merely one of many useful ways of looking at the activity of a warehouse. In fact, dollar-volume will be of little interest to us because it represents a financial perspective, while we are interested mainly in efficient warehouse operations. Consequently we will want to see the extent each sku consumes resources such as labor and space.

Frequently, an ABC analysis yields surprising results. For example, here are three different views of the activity at the national distribution center of a large retail drugstore chain. First, let us see which skus accounted for the most cases moving through the warehouse. This would be of interest to the the receiving, put-away, and restocking operations because each case must be handled separately to put it on a shelf. It also might reveal what is flowing in greatest quantity along a conveyor in the warehouse. Table 14.1 gives the ten most important skus by number of cases moved. Note that skus with relatively few pieces per case, such as the number 1 item, can appear on this list even though its total sales (pieces) are only moderate. Effects like this sometimes make the results of ABC analysis surprising.

Most of the labor in a warehouse operations is devoted to order-picking and so it is useful to rank skus by the number of times they were picked during some recent interval, such as in Table 14.2:

Finally, consider the number of pieces sold of each (Table 14.3). This is of interest because each piece must be handled by a sales clerk ringing up merchandise in a retail store. Surprisingly, the ten busiest skus with respect to pieces sold are almost all baseball cards and microwave popcorn. It seems that much retail labor is devoted to handling these.

	SKU	Cartons
1	UL SLIMFAST BONUS CHOC ROYALE	514.17
2	BANDAID FAMILY TWIN PACK	374.00
3	SATHERS PIXY STIX	360.00
4	GEMINI VIDEO TAPE T-120	302.50
5	HOUSE BRAND ASPIRIN 5 GR.	262.00
6	HOUSE BRAND COMPLETE ALLERGY CAPS	243.75
7	ACT II MICRO BUTTER	238.62
8	HOUSE BRAND PAIN REL CAPLETS 500MG	233.50
9	HOUSE BRAND GESIC	231.75
10	SATHERS S/F ASST SOUR MIX	210.00

Table 14.1: Top ten items of a chain of retail drug stores, as measured in number of cartons moved during 3 weeks

	SKU	Picks
1	ACT II MICRO BUTTER	806
2	BEACH BAG SET	781
3	ACT II MICRO LITE BUTTER	570
4	HOUSE BRAND PAIN REL CAPLETS 500MG	569
5	ACT II MICRO WHITE CHEDDAR	553
6	HOUSE BRAND COMPLETE ALLERGY CAPS	538
7	HOUSE BRAND OINTMENT TRIPLE ANTIBIO	534
8	WRIGLEY PLEN-T-PAK BIG RED	530
9	WRIGLEY PLEN-T-PAK DOUBLEMINT	526
10	UL SLIMFAST BONUS CHOC ROYALE	525

Table 14.2: Top ten items of a chain of retail drug stores, as measured by the number of customer requests (picks) during 3 weeks

	SKU	Pieces sold
1	UPPER DECK BASEBALL LOW#1992	70,524
2	ACT II MICRO BUTTER	34,362
3	SCORE 92 BASEBALL SERIES II	25,344
4	ACT II MICRO LITE BUTTER	21,276
5	TOPPS 92 WAX PACK BASEBALL	18,684
6	ACT II MICRO WHITE CHEDDAR	15,870
7	WRIGLEY PLEN-T-PAK DOUBLEMINT	14,736
8	ACT II MICRO NATURAL	13,284
9	WRIGLEY PLEN-T-PAK BIG RED	12,792
10	HERSHEY REESE PEANUT BUTTER CP	12,708

Table 14.3: Top ten items of a chain of retail drug stores, as measured by the number of pieces sold during 3 weeks

	SKU	Picks
1	TAPE,TRANS,MAGIC,3/4"W,1"CO	2,225
2	CLIP,BINDER,SMALL	2,171
3	FOLDER,FILE,LETTER,1/3,MAN	2,163
4	CRTDG,INK,DESKJT,BK	2,157
5	DISK,3.5,DS-HD,IBM FRMT	2,097
6	MARKER,SHARPIE,FN,PERM,BLCK	2,075
7	NOTE,HIGHLAND,3X3,YELLOW	2,062
8	CLIP,GEM,SIZE 1,REGULAR	2,049
9	PAD,LEGAL,LTR SIZE,WHITE	2,009
10	PEN,BALL PT,MED,STICK,BK	2,008

Table 14.4: Top ten office products measured by customer requests during a year

	SKU	Total weight shipped
1	CRTDG,TONER,3035,4045,BK	45,490.1
2	FLDR,LT,11PT,SGL,1/3MA10330	37,080.6
3	PPR,TW,25%RAG,8.5X11,20#,WE	28,194.5
4	CARD,INDEX,CNT,3X5,5C/yPK	21,411.0
5	POCKET,FLE,9.5X14.75,3.5,RR	20,426.5
6	FLDR,LT W/2B FST/150L-13	19,885.8
7	FLDR,LG,11,SGL,1/3MA 15330	16,231.2
8	PROTECTOR,SURGE,6OUT,6',PTY	13,578.2
9	FOLDER,LTR,2 PLI,STRT,24110	13,495.4
10	FASTENER,P/S,2/68220	12,910.7

Table 14.5: Top ten wholesale office products by weight shipped during a year

We find similar surprises in examining activity at a wholesale distributor of office products, for whom the ten most frequently requested skus were as shown in Table 14.4.

Notice that the ABC distribution for office products is not strongly skewed (that is, the number of picks falls off relatively slowly as you move down the list). This is a reflection of the maturity of the product and is typical of product movement in hardware and staples. In contrast, the ABC analysis of fashion products can be extraordinarily skewed; for example, the top-selling 100 music CDs from a population of 100,000+ may account for 25% of all sales.

If we examine the same population of office products by total weight sold, we get a clue as to which skus account for most of our shipping costs, which are based most strongly on weight (Table 14.5).

14.2.2 Statistical analysis

To design a new warehouse, retrofit an existing warehouse, or improve warehouse operations requires detailed understanding of the workload in the facility. One must analyze the patterns of customer orders and how this determines the workload within the facility.

Data sources

There are three main types of data required to support profiling: data pertaining to each sku, data pertaining to customer orders, and data pertaining to locations within the warehouse.

Sku data Useful information to gather about each sku include

- A unique ID that distinguishes it from all other skus, which allows us to connect this data with that from other sources
- A short text description, which is useful in validation and error checking
- Product family, which may have implications for storage and/or handling. These tend to be particular to an industry and so require knowledge of the context. For example, product families for a drug store chain might include hair care products, dental products, shaving products and so on, which are displayed together at the retail store. For a grocery distributor product families might include dry goods, dairy, produce, refrigerated, frozen and so on. For a candy distributor product families might include chocolate (sensitive to heat), mint-flavored candies (odoriferous), and marshmallow (light and tends to absorb the smells of its neighbors), and so on. For an apparel distributor product families might include garment type, mill, style, color, or size. Note that a sku might be in more than one product family.
- Addresses of storage locations within the warehouse. This might include zone, aisle, section, shelf and position on the shelf.
- For each location at which this sku is stored
  - Scale of the storage unit, such as pallets or cases. This is useful in validation and error checking
  - Physical dimensions of the storage unit (length, width, height, weight), which are useful in understanding space requirements.
  - Scale of the selling unit, such as cases or pieces, which is useful for validation and error-checking
  - Number of selling units per storage unit. This could be 1.
- Date introduced, which helps identify skus that may be underrepresented in activity because newly introduced
- Maximum inventory levels by month or week, which helps determine how much space must be provided for this sku

It is particularly important to understand the conventions the warehouse uses to distinguish among different types of storage units and selling units. For example, the word “case” is often called by other names (“carton”, “box”) and, depending on its use, can have substantially different meanings for a warehouse. For example, a vendor may ship a case that contains several inner packs each of which contains several boxes each of which contains pieces (Figure 2.2). A standard example is an office products distributor supplying a standard type of ball point pen. The manufacturer may supply the product 12 pieces to a box (as you find it in the store), 12 boxes to an inner pack (stored in a thin carton container), and 4 inner packs to a case for a grand total of 576 pens in the vendor’s case or shipping unit. While each of the terms each, box, inner pack, case, shipping unit are commonly used, there is no convention as to which level of packing they apply.

It is important to understand how this packing data is stored in the database. Often, the retail customer (of the distributor) is required to purchase an integer multiple of a selling unit. For the pen the selling unit may be an each, which means that a customer can actually order less than a full box of 12 pens. In the database this information may be stored as either the number “1” or by the symbol “EA” for each. If the customer is required to purchase boxes, then the selling unit may be listed as “12” or “box”. Now suppose the database records a customer purchase of 12, which appears on the order picker’s pick ticket? What exactly does this mean: 12 pens (1 box) or 12 boxes? If you think it means 12 pens when in fact it means 12 boxes, you would be underestimating demand by a factor of 12. If the manufacturer sells the pens as pieces, accounting will record transactions in units of a single pen, notwithstanding the restriction on the outbound side on how it is to be resold. To ensure consistency, one could always record demand in the smallest physical unit, pens, but this would require an order picker to know that 156 = 13 boxes, for example. To facilitate picking accuracy, the demand may be recorded as 13, and the order picker would know that this means 13 boxes. (Of course, if another pen is resold as pieces, this would lead to confusion.) To avoid this confusion, many facilities separate how demand is recorded for accounting purposes from how it is presented on a pick ticket, e.g., the pick ticket should say something like demand $= 156 = 13$ boxes of 12 each.

There is another reason why such packing data is important. It is always more efficient to store and handle product in some kind of easily-held container as opposed to loose pieces. In the case of the pen this handling unit may be the box of 12 or the inner pack of 12 boxes. For purposes of restocking a shelf of product it would be much easier to restock in units of the inner pack. For purposes of space efficiency on the shelf and order picking efficiency, it would be better to store the product as boxes neatly stacked.

The sku data may reside in different databases within a company and so it can present a challenge to collect it all. As a general rule, if you think there is the smallest chance that some data may be relevant, collect it!

Order history The order history is simply a concatenation of all the shopping lists submitted by all the customers during the preceding year. It contains the following information.

- Unique ID of this order, to distinguish it from the shopping lists of other customers and from the shopping list of the same customer on another day or later on the same day
- Unique ID of sku, which allows us to look up the sku to see where it is stored
- Customer
- Special handling
- Date/time order picked
- Quantity shipped

For analyzing warehouse operations you have to be careful where you obtain this data. Often this data is from a sales transaction database, which tracks financial rather than warehouse events. Consequently, the date recorded may represent the date the order was placed or when it was printed, not when it was processed in the facility. Similarly, the order that appears in the sales transaction database may actually have been processed at another facility. Generally, though, the information is available somewhere because each day the order pickers had to know what to pick.

Keep in mind that an order history is primarily financial information. This is good in that it is likely to be very accurate; but it can also be misleading because the transactions it represents are financial and not necessarily operational. For example, it can happen that a sku is shown to have been requested in a negative amount; but this generally means something like the item was returned and restocked to the shelf.

There is a simple check as to whether the order data received is approximately correct. Most companies keep track of the lines shipped each day. As a very first validation check, count the number of lines in the database by time period. These numbers should closely match what is recorded. (If you are obtaining only what personnel believe has been shipped, do not be surprised if the numbers obtained through careful processing of a database are substantially different.)

The order data will be the largest file you must manage. As a rough estimate expect about 50 bytes per line. The number of lines can range from 2,000–8,000 lines per day (0.5–2 million lines per year) for a moderately active facility (for example, office product, fine paper, telecommunications distribution) to 10,000–40,000 lines per day (2.5–10 million lines per year) for an extremely active facility (for example, service parts, retail drug) to more than 80,000 lines per day (20 million lines per year) for the most active facilities (for example, pharmaceutical or catalog distribution). Consequently, a year's order data could exceed 100 megabytes.

Warehouse layout and location addresses A map of the warehouse allows us to see where each sku is stored. We can infer that an order-picker had to travel to this location to retrieve the product; and from the map we can infer something about the required travel. This will enable us to evaluate alternative layouts and warehouse designs.

This type of information is generally least standardized and may be found in the form of blueprints, sketches, CAD files such as `dwg` format, and so on.

Where and when is the work?

How can we estimate the work in a warehouse? Work is generated by the customer orders; each customer order is a shopping list comprised of "pick lines"; and each pick line generates travel to the appropriate storage location and subsequent picking, checking, packing and shipping the product. Pick lines are then a strong indicator of work; and fortunately there is almost always a historical record of them because they correspond to entries in a sales invoice, which is one of the first pieces of information to be computerized.

We use this information to infer where the work is; that is, how it is distributed among

- Skus
- Product families
- Storage locations
- Zones of the warehouse
- Time (time of day, days of the week, weeks of the year, and so on)

Sometimes this is referred to as *activity analysis* because we examine the activity of each sku, in particular, how many times was it requested; and how much of the sku was sold? Notice that these are two different questions: The first asks "on how many customer orders did this sku appear?"; and the second asks "How many pieces, cases or pallets moved through the warehouse?".

If a customer requests a quantity that is less than a full case, this is termed a *broken-case pick*. A broken-case pick can be further classified as to as an inner-pack pick, and so on, depending on how the product is packaged. If a customer requests a quantity that is an integer multiple of a case quantity but less than a pallet (unit) load, this is termed a *full-case pick*. A *pallet pick* represents an order quantity that is a multiple of a pallet load quantity. It is not uncommon for a customer request quantity to involve a *mixed pick*; that is, a pick involving both a broken- and full-case quantity or both a full-case and pallet-load quantity. Broken-case picking requires more time to process than a full-case pick, which takes more time to process than a pallet pick, when normalized by the quantity handled. It is therefore desirable to know how much of each activity is taking place each period.

Here is an application. In fine paper distribution, many skus have a considerable amount of both broken and full case picking. Many of the cases are also quite heavy. In one facility, order pickers were making circuitous, inefficient routes so that they could first store the case quantities on the bottom of their cart, and then store the broken case quantities loose on top so as to not crush or damage the loose quantities of paper. It was decided that there should be separate broken and full case picking zones to reduce this inefficiency. A mixture of shelving and flow rack was decided upon for storing the case quantities from which to execute the broken case picking activity. Pallet rack, as before, was to be used to store the full or partial pallet loads from which to execute the full case picking, and from which to restock the broken case picking area. To decide

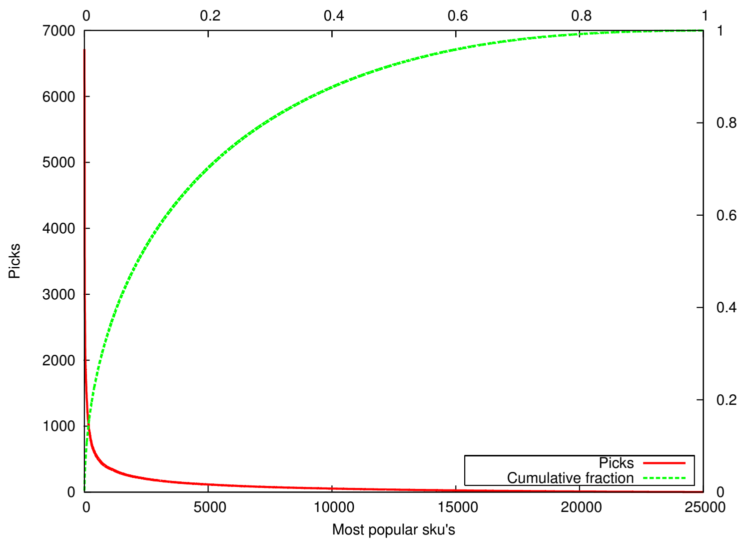


Figure 14.1: How picking is distributed over the skus. The horizontal axis lists the skus by frequency of requests (picks). By showing both actual values and cumulative fractions it is easy to see how concentrated the picking is amongst the most popular skus.

the appropriate amounts of space (slot types) to each sku in each zone would require a breakdown of each skus broken versus full-case picking activity, both by picks and by demand. Those skus that had only a very small portion of its activity of one type may not have been assigned to both zones.

Once such activity has been calculated, a variety of Pareto curves can be generated. For example, one can rank the skus by popularity (number of requests, picks), sort the list, and then produce a graph that shows the percentage of all picks among the most popular skus. For example, Figure 14.1 shows data for a warehouse in which the 5,000 busiest skus account for over 75% of the picks. This suggests that it might make sense to layout the warehouse to support a fast-pick area (discussed in detail in Chapter 8).

Another useful Pareto curve can be generated by examining the broken-case or full-case picks separately. Similarly, curves can be generated by examining key subsets of skus, such as those from one region of the warehouse, or with common seasonality, or from the same product family. Analyzing subsets of skus in this way may suggest which areas of the warehouse or which product families stand to have the most opportunity for improvement, or which customers are responsible for the most workload. Finally, one could replace the word "picks" with any type of activity, such as demand. If demand is to be analyzed, it must be normalized into cases or pallets so that there is a common basis for comparison.

There are other distributions that can reveal patterns in the way product moves through the warehouse. For example, Figure 14.2 shows a bird's eye view of a warehouse in which a darker shading indicates more frequent visits by order-pickers. It is

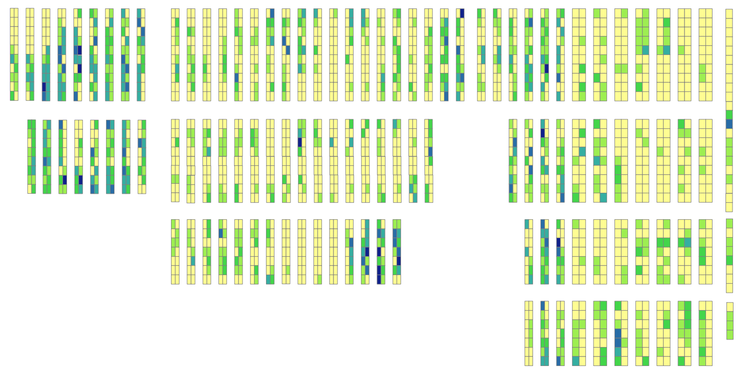


Figure 14.2: A bird's eye view of a warehouse, with each section of shelf colored in proportion to the frequency of requests for the skus stored therein.

clear from this that popular skus are stored all throughout the warehouse and so management can expect greater labor costs and reduced responsiveness because of all the walking necessary to retrieve a typical order.

Seasonalities

It is important to understand how the intensity of work varies over time. Most products have some natural "cycle" that repeats over the year or quarter or month or week. In North America, and particularly in the US, the annual holiday season of roughly November–December is by far the busiest time for retail sales and this determines the timing of product flow upstream. Predictably, supply chains to meet this demand are full in the months preceding.

The selling season in the US has been increasingly extended into January due to increased sales of gift certificates. Typically, these are given as holiday gifts and are then redeemed in the weeks following the holiday.

Not all seasonalities are obvious. For example, the manager of Allied Foods in Atlanta, Georgia tells us that even dog food has seasonalities: Demand increases slightly but dependably over the end-of-the-year holidays.

One sku with easily predictable seasonality is AA batteries. This is a mature technology, not subject to fashion or obsolescence, and so demand is fairly steady for most of the year . . . except of course for the month of December, when demand almost doubles. In fact, most of the demand in December comes on Christmas day and it is concentrated mostly at convenience stores, such as 7-11 or Quik-Trip.

Other seasonalities include:

- Office products sell most heavily on Mondays and Fridays, in January and in August. Among these, calendars sell most briskly in January, with sales dropping until June, when it disappears.

- The two fastest-moving items at Home Depot at Father's Day are (barbecue)

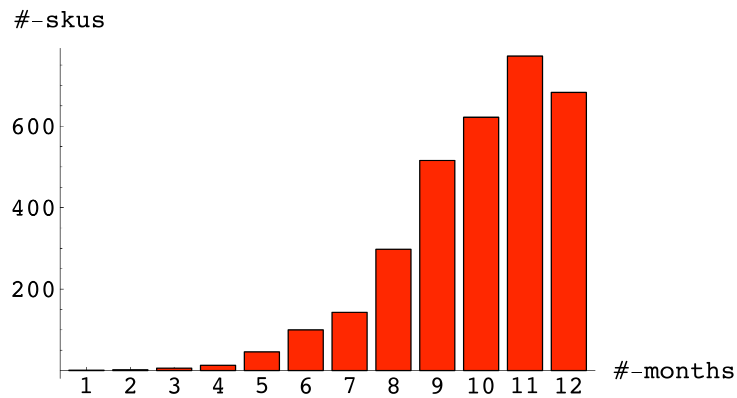


Figure 14.3: Number of the most popular skus that were requested during only n months of the year ($n = 1, \ldots, 12$).

grills and (electric) drills. Barbecue grills sell in the spring up to July 4, when sales plummet.

Finally, see whether you can guess the seasonalities of the following. (Answers at the end of the chapter.)

- Sales of large screen color televisions
- Sales of refrigerators; sales of other large kitchen appliances in the US
- Sales of CDs and other recorded music
- Consumption of avocados in the US
- Sales of disposable diapers
- Rental of tuxedos
- Sales of belts

The best way to learn seasonalities is to learn the business; but it can be useful to search for statistical evidence of seasonalities. One simple first exploration is to count the number of months in which each sku is requested. Figure 14.3, which shows the results for the most popular skus of a home-furnishings retailer, indicates that only nine skus sold exclusively during a 3-month period or less. (There is no point in examining the less popular skus as it is unclear how to recognize any seasonality.)

Patterns of work

Here we want to go beyond measuring the quantities of work to understand the *patterns* of work generated by the customer orders.

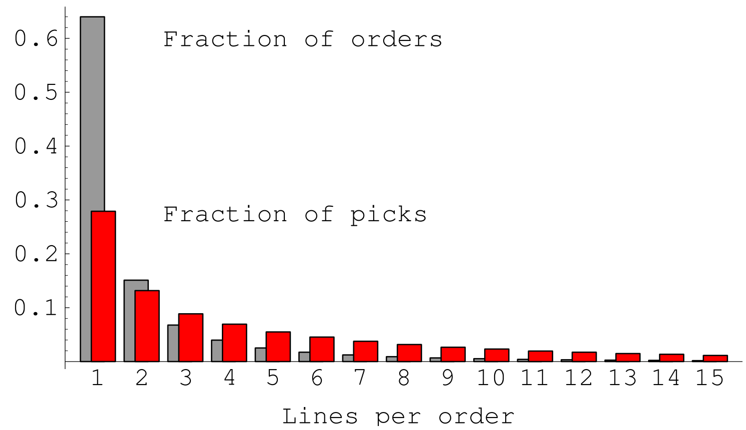


Figure 14.4: About two-thirds of the orders are for a single line but these account for only about one-third of the picks.

If no customer ordered more than one sku, then the preceding activity analysis gives a sufficient view of the warehouse; but this is rarely the case and customers order multiple skus. It is then important to understand the patterns in the customer orders. For example, one indication of inherent work is the average lines per order. When this number is small, say up to about 4, it may be preferable to batch orders and assign one picker to each batch. When the number of lines per order exceeds that, orders will typically be assembled by some form of zone picking, with orders progressively assembled. If the number of lines is much larger, the zones may pick in parallel and feed pick-lines to a downstream process to accumulate and sort the orders (to reduce the number of packages sent to each customer and thereby reduce downstream shipping and handling costs).

But, as always, one must beware of averages. It is always more informative to examine the complete distribution of lines per order. It shows the fraction of orders for a single sku, for exactly two skus, and so on, as in Figure 14.4. In this example, most orders are for a single line, which suggests some opportunities for efficient handling. For example, if these are mostly back-orders, then they could be cross-docked. If they are rush orders, they could be grouped together into a single batch and then picked in storage sequence.

A related graph is the distribution of picks by order-size. That is, it depicts the fraction of all picks that come from single-line orders, two-line orders, and so on. Because picks are a good indication of work, this shows which types of orders, small or large, contain the most aggregate work.

Here is an application. At a telecommunications facility, workers pushed order-picking carts through a series of zones to progressively assemble orders. Due to space limitations on each cart, the *transfer batch* was small. An analysis of orders showed that about 10% of the orders were relatively large, for more than 100 lines each, whereas the remaining 90% averaged less than 2 lines per order. It was decided to assign one worker to pick each extremely large order, one at a time, so that the remaining orders could be

Order ID	Families represented
100	A, B, C
200	A,B
300	C,D,E
400	B,D,E
500	D,E
600	A,D,E
700	B,D,E

picked and transferred in larger batches, thereby increasing order-picking efficiency.

It is frequently also useful to generate the distribution of families per order; that is, the fraction of all orders that involve exactly one product family, two product families, and so on. Here is an application. Generally it is helpful to locate skus near one another if they tend to be requested together (for example, running shoes and socks; flashlights and batteries). Such product assignment can reduce travel time during order-picking. But which products tend to be picked together? In one paper distribution facility, skus were classified into three categories. It was readily verified that customers rarely ordered across categories, so, the warehouse could be divided into three zones, each of which functioned as a smaller and more efficient warehouse. Further inspection of orders revealed that there were typically many lines per order; and that most orders requested the products of no more than two vendors. From this information, a vendor-based stock assignment plan made the most sense: store all skus within a vendor in the same area of the warehouse.

Of course, the next question might be: which vendors should be located near one another? This brings us to the concept of *family pairs-analysis*. For example, consider the following list of multi-family orders and the families involved in each order (Table 14.2.2).

With five product families A–E there are ten family pairs, with the following frequency among orders:

Family pairs:	AB	AC	AD	AE	BC	BD	BE	CD	CE	DE
Frequency:	2	1	1	1	1	2	2	1	1	5

and it is clear that any order requesting a sku from family D is likely to request a sku from family E as well and so one should consider storing these families near one another.

It is possible to consider triples of product families and beyond, but the possibilities increase exponentially and so the work required quickly outstrips any computing power brought to bear on the analysis.

In one distribution center two product families were moderately correlated but on closer inspection it was realized that one of the product families was very active, while the other product family much less so. However, there was a 98% chance that if an order requested a sku from the less active family it would also request a sku from the more active family. Since the less active family consumed little space, it made sense to store this family next to the larger, more active one.

A whole new picture takes place if one aggregates skus by location or zone within a facility. The symbols A–E above could represent a zone in the warehouse. The distributions so obtained represent “order-crossings” and reflect the degree of order-accumulation across zones.

One distribution center progressively assembled orders by picking pieces. The product was in cases stored in shelving across five zones of about three aisles each. A non-powered roller conveyor was used to move the totes from one zone to the next. The conveyor jammed frequently because too many totes were being staged. One way to reduce this congestion is to relocate the stock so that many more orders could be completed entirely within one zone or within neighboring zones. This may be achieved by seeing which families tend to be ordered together.

Finally, it should be noted that when speaking of orders, it is possible to partition the order file into groups of sub-orders. For example, one can consider only that portion of the order pertaining to a zone within the facility. This is especially helpful and required if picking policies are different in different zones.

14.2.3 Doing it

How do you actually do profiling? A personal computer is adequate to the task, but it can be a messy business. The first problem is getting the data.

Getting the data

You will probably be working off-line, with a snapshots from corporate databases (otherwise you may interfere with daily operations when your queries slow the database that supports the warehouse).

Budget several weeks to complete activity profiling. Most of this time will be spent checking, validating, and purifying the data.

As of this writing, there are hundreds, possibly thousands, of different warehouse management or record-keeping software systems, many of them written in-house. You must appeal to the IT department, which like all IT departments, is overworked and behind schedule. Running the queries to get your data may steal CPU cycles from the computers that are managing the business. Do not expect to be welcomed back, so you might as well ask for copies of every database and sort it out later, on your own.

Time permitting, make a sample request for a small piece of the data (say, one day’s worth) and check that it makes sense. Review the meaning of every data field with people from both operations and from IT. Show them summary statistics you extract from that day of data to see whether they seem reasonable. Check the text description of the biggest, busiest skus to see that there are no suspicious results.

Only after successfully sampling the data should you submit your request for the full data dump.

The preferable way to receive the data is on CD-ROM, which means you do not need unusual equipment to read it and you cannot accidentally erase it. Alternatively, most facilities are capable of making data available via `ftp`.

We find it best to receive the data in some simple, neutral format, such as tab-delimited ASCII (successive fields in a line of data are separated by a tab character). This keeps options open because the data is easily transferable to a range of software tools.

Data-mining

There is no packaged software that is universally applicable to extract important patterns from the data. The essential functionality you will need are the abilities to

- Sort the rows of a table;
- Select a subset of rows of a table, such as all the order-lines from a single day.
- Count distinct entries in a table, such as the number of times each sku appears in the order history;
- Connect the row of one table with a corresponding rows in another table (for which the database jargon is *join*). An example is connect order-lines (rows in the orders table), through the unique sku identifier, with additional information about each sku (rows in the skus table), such as where it is stored.
- Graph results.

It is a fact of life that corporate data resides today mostly in relational databases and so some facility in manipulating them is essential. Many commercial databases support some form of "data-mining" by providing an intuitive querying capability. This is a front end that produces the nearly universal language of relational databases, SQL (Structured Query Language). However, these front ends are proprietary and vary considerably from product to product. In any event, you are likely to have to write some SQL; fortunately it is fairly straightforward. For example, the following command returns the number of lines in the table named "Lines":

```
SELECT COUNT(*) FROM Lines
```

and to show all lines shipped on January 20, 2009 the command is

```
SELECT * FROM Lines
WHERE DateShipped = 2009-01-20.
```

To return a view of the total units shipped for each sku, the command is

```
SELECT SkuID, SUM(QtyShipped) FROM Lines
GROUP BY SkuID
```

and to return the total number of requests (picks) for each sku,

```
SELECT SkuID, COUNT(*) FROM Lines
GROUP BY SkuID
```

To return a list of orders and the zone from which each line was requested, the command is

```
SELECT Lines.OrderID, Skus.Zone FROM Lines, Skus
WHERE Lines.SkuID = Skus.SkuID
```

An alternative to using a database and writing SQL is to directly program, in some lower-level computer language, the ability to query databases. This is not hard and may be worthwhile for the savings in disk space and running time. Importing ASCII data into a commercial database may inflate its size by a factor of five to ten.

The main thing to look for is a set of tools that are flexible, because each warehouse is so different that it does not seem possible to standardize on a general set of queries. Plan on building or buying a library of queries that you can revise and adapt.

Discrepancies in the data

There will be discrepancies in your data. Expect them and have a strategy for dealing with them. First find them and then document their severity. For example, exactly what percentage of skus that appear in customer orders do not appear in the sku database? If it is a small problem, it can be ignored for purposes of warehouse profiling; but the warehouse IT staff should be informed so they can fix this.

Be sure to check with special care the busiest skus, locations, times, and so on. Errors tend to appear at the extremes of the ABC distributions. Here are two examples that happened to us. In one case we discovered that a particular style of paper accounted for a huge fraction of all the cubic feet shipped from the warehouse. This surprised our client. On checking further we learned that sales of this paper had been recorded by the sheet, not by the case, as we had assumed. Similarly we were surprised to find that a popular writing pad accounted for exactly zero cubic feet of product shipped. The pad had been measured on an automated device that captured its dimensions as 8.5 inches $\times$ 11 inches $\times$ 0 inches $=$ 0 cubic inches.

Many problems arise because the order history spans an interval of time; while the file of skus may represent a snapshot of one instant. Consequently, you are likely to find skus appearing in one database but not in the other.

The importance of cross-checking

You can reduce problems by energetically cross-checking all data and analysis. Get in the habit of asking the same question in different ways or addressing it to different sources (people, databases). Check that the answers are consistent. For example, how many lines does the warehouse ship per day? How many orders? What is the average number of lines per order? Does the math check?

A few handy tools make it easier to cross-check.

- On average the total flow into a warehouse is the same as the total flow out and therefore Little's Law applies: $L = \lambda W$.
- Approximation: One tightly-packed 53-foot (16.15-meter) long trailer holds about 2,000 cubic feet (56.6 cubic meters) of product, which is about 20 pallets.

- A person walks about 4 miles (6.44 kilometers) per hour.

Interpreting patterns

It is easy to make mistakes in interpreting the data. Here are some to avoid.

Beware of small numbers If skus are unusually slow-moving, such as is typical at a service parts distribution center, then many skus will sell only tiny quantities during a year. A statistical fluctuation in demand that is small in absolute terms might represent a huge percentage increase and so mislead analysis.

At one site we estimated the time supply represented by the on-hand inventory of each sku so we could evaluate the appropriateness of inventory levels. (In financial reports, time supply is generally called *Days Inventory Outstanding* or *DIO*.) This warehouse provided service parts to the US military and held hundreds of thousands of skus. Like most service parts warehouses, most skus were requested infrequently.

We received a snapshot of the inventory levels along with a sales history of the previous year. We estimated time supplies as follows: If a sku sold Q units last year and had q units currently on hand then we estimated there was enough to last for q/Q years. Then we counted the number of skus with 1 year supply, 2 years supply, and so on. Here is the result.

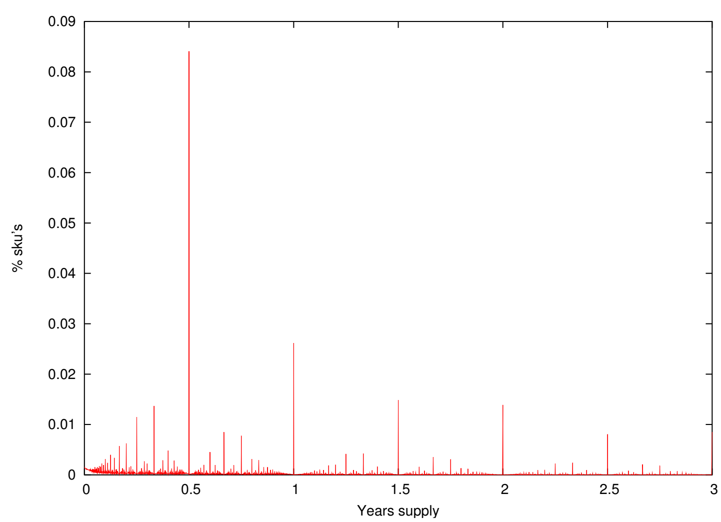


Figure 14.5: Why are there so many skus with exactly a half-year supply?

At first glance this pattern is very puzzling. There are three things to explain.

- Why are so many items grouped by time supply?
- Why are the most common values of time supply exactly six months, one year, and so on?

- Why should we have observed exactly those values on an arbitrary date?

In discussions with the inventory managers and the purchasing department some suggested that the pattern reflected purchasing patterns; others thought that there might be requirements to hold, for example, a year's supply of certain skus; and some thought that the inventory levels might have been set indirectly by the military training schedules that would consume the supplies. All these may be true to some extent and seem to explain why there might be peaks in the time-supplies, and while some can even explain why those peaks might occur at natural intervals such as one-half or one year, none explain why those natural intervals were observed on the arbitrary day for which we examined the data. Even if a particular sku is required to be held in quantities of at least one-year's supply, why should we have found the inventories at exactly one-year's supply? Why not at 1.2 years?

In fact, the explanation is quite different from any of these and it does not involve logistics at all. The pattern is simply a numerical artifact. Recall that the distribution center holds very many skus that are very slow moving. There are tens of thousands of skus of which only one or two were shipped last year. Because these are such slow movers, the DC may hold only a few in stock. For example, there are thousands of skus that sold only a single unit last year and for which there are exactly two currently in stock, or two-year's supply. For all these skus with few units sold and few units in stock, there are only a few possible values of estimated days-of-supply (units in stock divided by last year's sales). For example, all the skus that sold 1 or 2 units last year and that have 1–3 units in stock now may be considered to have either 1/2 or 1 or 1.5 or 2 or 3 years supply on-hand. In other words, this is an artifact of small numbers.

Notice that because these are slow movers this condition persists. A sku that sold one unit last year and has two in stock now has an estimated two year supply; and, furthermore, this inventory level would be observed almost any time this sku was examined.

Beware of sampling biases If you want to understand how long skus are resident in the warehouse you will have to measure or estimate the time between arrival and departure and this can depend heavily on where you sample the skus. For example, if you observe the shipping department you can examine each pallet. Its date of departure is the current date; and you may find its arrival date in the inventory management system; and the time in residence is the difference in the current and arrival dates. However, such measurements will over-sample those pallets that spend little time in the warehouse because those are the pallets most frequently on the shipping dock.

On the other hand, you might walk through the warehouse and estimate departure times. If you assume that demand for each sku follows a Poisson distribution then the time between departures is exponentially distributed. Thus if a pallet of the sku has been in the warehouse for d days we expect it to depart after another d days and so its estimated time in residence is $2d$ days. The problem with this estimate, however, is that it over-samples the *slow-moving* pallets, since those are the ones more likely to be found sitting in the warehouse.

14.2.4 Visualization

Readers are strongly encouraged to study E. Tufte's "The Visual Display of Quantitative Information" and the companion books [45, 46, 47].

Some general principles:

- Use graphs to show large scale patterns, especially when you want to compare with other patterns. And when you want graphs to be compared, draw them to the same scale.
- Show percentages. Use tables for closer looks when the actual numbers are important.
- Scale the data to make it easier to understand. For example, it is better to report average lines per day rather than total lines over the period of study. Most people will find it easier to understand the implications of the numbers.
- The defaults on standard office software, such as MS Excel, are generally poor. If you use these tools, you will need to intervene to make a quality graph.

14.3 Summary

An activity profile is essential to really understand what matters in a warehouse. You can build this from data about the physical layout of the warehouse, the skus stored therein, and the patterns of your customer orders. The activity profile will enable you to understand, manage, and improve use of labor, space, and equipment.

Warehouse activity profiling is a special case of *data-mining*, which is simply the rummaging through databases to look for patterns that might be exploited to improve operations. As in mining for minerals, success depends on having good tools to support the search and on knowing where to look. Therefore you must be comfortable with SQL and databases and you must understand warehouse operations.

14.4 On the lighter side

Getting data Everyone who has profiled a warehouse has stories about how hard it was to get the data. One consultant described the following experience. Without checking in advance, a client attempted to e-mail him an enormous data file for analysis. When his mail server rejected it as too large, the client broke the file into 25 (still large) data files and sent them along. Meanwhile, the consultant was working on-site elsewhere and was expecting an urgent e-mail; but whenever he connected to his mail server the large files began to download over his slow phone connection. Calls to the mail service provider did not help. He finally resigned himself to sacrificing his machine for a few days while he paid the connection charges to receive the data files. To add insult to injury, his mail server had accepted only 15 of the data files before overflowing his disk allotment, so it was all for naught.

	SKU	Cases Moved
1	Mayonnaise	62,788
2	Charcoal	48,446
3	Sani-cat litter	46,632
4	2-liter gingerale	41,596
5	Charcoal briquets	40,660
6	2-liter cola	37,719
7	Apple juice	34,764
8	Granulated sugar	28,251
9	2-liter orange drink	27,301
10	Charmin bathroom tissue	24,379

Table 14.6: Top ten items moving from a grocery warehouse, as measured by number of cases

Grocery distribution An ABC analysis of a grocery distribution center provides an interesting glimpse into US eating habits. Most product moves out of a grocery warehouse in cases and the ten most popular skus, as measured by number of cases purchased, reveals where order-picking labor is concentrated (Table 14.6).

Seasonalities Many seasonalities are based on local custom and can be hard to predict without knowing the culture. Here are some peak seasons for some products in the US.

- Sales of large screen color televisions: During the two weeks preceding Superbowl. (Source: Mitsubishi Consumer Electronics, Atlanta, Georgia)
- Sales of refrigerators peak in the hot weather, when old refrigerators are more likely to break down. In the US, sales of other large kitchen appliances peak during the "cooking season" surrounding end-of-year holidays (November, December).
- Both barbeque grills and electric drills are especially popular just before Father's Day in June. (Source: L. Kapiloff, The Home Depot)
- Sales of CDs and other recorded music: During the two weeks following Christmas, when presumably people received CD or DVD players as gifts. The most popular first purchases are "golden oldies". (Source: SuperClub Music and Video, Atlanta, Georgia)
- Consumption of avocados in the US: The day at which consumption is highest is *Cinco de Mayo*, followed by the day of the Superbowl, when presumably it is consumed as guacamole. (Source: *Wall Street Journal*, January 26, 1999). At neither of these times is consumption unusually high in Mexico.
- There is a slight peak in sales of dog food over the December holidays. (Source: Acme Pet Food, Atlanta, Georgia)

- Rental of tuxedos: Proms in May and weddings in June. (Source: Mitchell Tuxedo, Atlanta, Georgia)
- Sales of men's belts: Easter in March/April; Father's Day in June; back-to-school in August; Christmas. (Source: Italian Design Group, Atlanta, Georgia)
- Sales of disposable diapers: No reliable seasonalities, but interestingly the rates differ by region of the world. For example, Jim Apple of The Progress Group says that the rate of use in the US is about seven per day per baby but it is only five per day in Europe.

Some seasonalities can be hard to predict even if the culture is understood: For example, in Mexico it is popular to eat a sweet bread called marinela when it rains; but rain is very hard to predict beyond a week.

14.5 Questions

Question 14.1. *Why is number of pick-lines generally a better indicator of labor than quantity-ordered or quantity-shipped?*

Question 14.2. *In a warehouse activity profile, is it possible for a single product group (in apparel, for example: mill, style, color, size) to complete more of the orders than any other but yet not have very many picks? Explain.*

Question 14.3. *Consider a lines-per-order distribution in which 30% of the orders are for single lines; 25% for 2-lines; 20% for 3-lines; 12% for 4-lines; 10% for 5-lines and the remaining 3% for 6-lines. Historically there have never been orders for 7 or more lines. What size order accounts for most of the picking in the warehouse?*

Question 14.4. *Which of the following statistics do you expect to follow an ABC-type distribution and why?*

- *Skus ranked by broken-case (less-than-full-carton) picks*
- *Skus ranked by flow (cubic volume sold per year)*
- *Skus ranked by volume of the standard carton in which it is packed*
- *Skus ranked by total weight shipped per year*
- *Skus ranked by weight of one carton*

Question 14.5. *Explain the bias inherent in estimating product turnover times by sampling product at the shipping dock.*

Chapter 15

Benchmarking

A warehouse, like any other enterprise, should constantly measure its performance, compare with others, and plan to improve. Each of these presents challenges: What to measure? With whom to compare? How to improve?

Generally there is not much to learn by comparing with an ideal because that ideal might not be practical; and, furthermore, it gives no hints on how to achieve similar performance. It makes more sense to compare a warehouse with its peers. If a peer is outperforming our warehouse, we can examine its facilities and processes to understand why and then try to adopt similar practices.

15.1 Performance measurement

What should you measure to judge the performance of a warehouse or distribution center? It is not enough to measure only output because that says nothing about the expense required to generate that output. Instead, we typically measure performance by a ratio

$$\frac{\text{units of output achieved}}{\text{units of input required}}.$$

These ratios generally are intended to summarize one of the following:

- Operating costs, such as warehouse costs as a percentage of sales
- Operating productivity, such as pick-lines, orders, cartons, pallets handled per person-hour
- Response time, measured, for example, as order-cycle time (minutes per order)
- Order accuracy, measured, for example, as fraction of shipments with returns

Many warehouses are managed from a list of such measurements, which are referred to as *key performance indicators* (KPI).

Ideally, a measure of productivity will be unbiased, customer-focused, and consistent with corporate goals. None of the KPI's mentioned above fits these criteria perfectly. For example, "total units shipped" is probably inconsistent with corporate goals because it omits concern for whether the correct items were shipped; furthermore, it is biased because it depends on total units ordered by the customer and so is not under direct control of the warehouse.

Similarly, "pick-lines per labor hour", a popular performance indicator, is biased because it depends on the units being picked: One would expect a higher score if picking cartons instead of pieces. And it is not focused on the customer.

"Order-cycle time" seems more defensible; but "warehousing costs as a percentage of sales" is biased because it depends on sales and so could be skewed by marketing. In addition, it is not customer-focused.

15.2 Benchmarking

Benchmarking is the comparison of one warehouse with others. One can try to make the comparison more meaningful by suitably restricting the community of warehouses to which comparison is made. Benchmarking may be done internally, on the processes within a single company; or externally, on the same process in other industries; or on the same process in competitors. There are challenges of diplomacy and information-sharing when benchmarking against other companies, especially competitors.

15.2.1 Ratio-based benchmarking

In ratio-based benchmarking, it is typical to report a collection of KPI's for a warehouse. But how do you compare two such scores? This sounds like a simple question but there can be surprising subtleties.

Consider, for example, the following three warehouses, each of which has been evaluated by three KPI's. (For simplicity assume the score for each KPI has been scaled to lie within $[0, 1]$, with 1 being the best possible. This is similar to the scoring scheme used by *Consumer Reports*.)

	KPI_1	KPI_2	KPI_3
Warehouse A	0.75	0.25	0.50
Warehouse B	0.50	0.75	0.25
Warehouse C	0.25	0.50	0.75

Which of these warehouses is performing "best"? A simple way of comparing them is to count the number of KPI's in which each warehouse scores best. Warehouse A beats B in KPI_1 and KPI_3 but loses to B in KPI_2. Because A beats B in two out of three KPI's, it is tempting to say that A is performing better warehouse than B. Similarly, B beats C. But—and here is where the trouble starts—C beats A! Strangely, while it seems to make sense to compare one warehouse with another in this way, it does not seem to make sense to ask which is the best.

Worse yet, it is possible to imagine a series of warehouse "improvements" that lead nowhere. Suppose, for example, that we are managing warehouse A. If we observe that warehouse B is more productive, we might reëngineer warehouse A to resemble B. But later we discover that our new warehouse is bested by warehouse C and so we reëngineer again to copy the configuration of C. But this configuration is not as efficient as the original configuration of warehouse A.

It is possible to make more sophisticated methods of aggregating the KPI's, such as by weighting the relative importance of each; but all such attempts are capable of paradoxical outcomes. How do we know this? One can interpret ratio-based benchmarking as a type of election in which each KPI is a voter and each warehouse is a candidate for most efficient. Deep results in the theory of voting and social choice tell us that there is no voting scheme that is free of undesirable behavior [40].

In summary, the fundamental problem with simple, ratio-based performance indicators is that each represents a limited and therefore possibly misleading point of view; and there is no wholly satisfactory model or structure to combine the measures of productivity into some integrated view.

15.2.2 Aggregate benchmarking

In systems-based benchmarking, we take an aggregate point of view and consider, not single inputs or outputs, but entire portfolios of inputs and outputs. We want to measure how well we are achieving a *portfolio* of outputs for a given *portfolio* of inputs.

The simplest comparison

Consider these two warehouses:

	Labor (hrs $\times 10^3$)	**Capital ($M)**	**Annual pick-lines (M)**
Warehouse A	100	1.0	1.6
Warehouse B	50	0.5	0.8

Which is more efficient? Each warehouse uses two inputs, labor and capital, to produce one output, picks (retrievals of product for customers). (This is a purposefully simplistic view of the inputs and outputs of a warehouse. In a more realistic model we would itemize the various types of labor, capital, and ways of measuring output; but for now our simple model serves a pedagogical purpose.)

Most people would find it plausible to consider warehouses A and B to be equally efficient. Warehouse B produces only half the output of warehouse A but consumes only half of each input and so they are perfectly proportional. One seems to be an exact but scaled version of the other. Presumably one could build an exact replica of warehouse B beside the original and this combined warehouse would be just as efficient as warehouse A.

An important assumption

But what about these two warehouses?

	Labor (hrs $\times 10^3$)	**Capital ($M)**	**Annual pick-lines (M)**
Warehouse A	100	1.0	1.6
Warehouse C	90	0.9	2.0

It is not so obvious how to compare warehouses A and C directly. One step toward that is to scale the outputs to a common value to get a sense of how much input each one uses per unit of output. We have done this here:

	Labor (hrs $\times 10^3$)	**Capital ($M)**	**Annual pick-lines (M)**
Warehouse A	100	1.00	1.6
Warehouse C'	72	0.72	1.6

where we see that Warehouse C' produces the same level of output with only 72% of the inputs. Warehouse C' seems more efficient than Warehouse A.

But we must be careful here: Does it make sense to talk of a scaled warehouse as if it could be realized in practice? What if scaling down warehouse C forfeited economies of scale and so C' was not really representative of what could be achieved?

Not everything about a warehouse scales linearly. While it may seem reasonable to think that half as many order-pickers would accomplish half the number of picks, would one expect half the output from a warehouse management system that costs half as much? In assuming the ability to scale all the inputs and outputs of a warehouse we are glossing over issues that we will have to reconsider when trying to devise a plan for improvement based on a benchmark. For now, we will gloss over this question and assume that it is at least not unreasonable to talk of scaled warehouses as if they could be realized in practice. Economists refer to this assumption as *constant returns to scale*.

The comparison between warehouses A and C' can be visualized geometrically, as in Figure 15.1, wherein each warehouse, scaled to the same output, is represented as a point in "resource space". The point $(0, 0)$ represents the ideal warehouse that achieves the same output with no inputs whatsoever.

In this model, a warehouse that is closer to the origin is generally more efficient than one farther away. More precisely, any warehouse that has smaller consumption of all resources for the same output must be more efficient. Such warehouses will be plotted below and to the left of warehouse A.

Because warehouses A and C' are on the same ray from the origin, we can make a more precise statement about their relative efficiencies: Warehouse A is only 72% as efficient as warehouse C'. This is reflected geometrically in that the distance from the origin to warehouse C' is 0.72 of the distance to warehouse A.

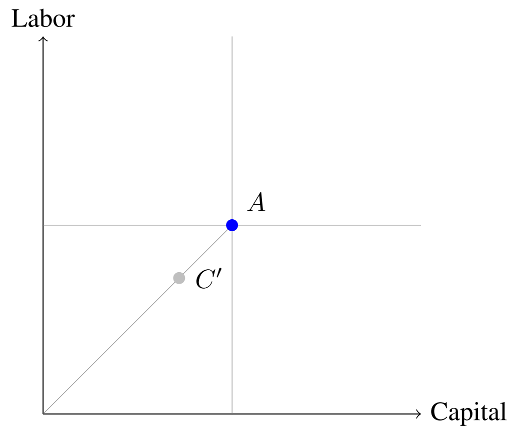


Figure 15.1: Warehouse C, scaled to the same output as A and plotted as C', reveals inefficiencies of A

A more challenging comparison

Now compare warehouses A and D:

	Labor (hrs $\times 10^3$)	**Capital ($M)**	**Annual pick-lines (M)**
Warehouse A	100	1.00	1.60
Warehouse D	30	0.24	0.64

As before, we invoke our assumption of scalability of warehouses so that we can compare their inputs at the same output level. Scaling warehouse D to match the output of warehouse A, gives warehouse D' below.

	Labor (hrs $\times 10^3$)	**Capital ($M)**	**Annual pick-lines (M)**
Warehouse A	100	1.00	1.60
Warehouse D'	75	0.60	1.60

And again we can interpret geometrically, as in Figure 15.2, where we see that warehouse D' lies below and to the left of A, reflecting the fact that it uses less of each input to produce the same output, and hence is more efficient. D' does not lie on the line connecting A to the origin, so we cannot use it directly to assign a relative efficiency score to A. But we could imagine another warehouse, just like D', but with additional, unproductive labor. This warehouse, D'', is a purely hypothetical facility that uses 75×10^3 person-hours of labor, but reveals A to be no more than 75% efficient.

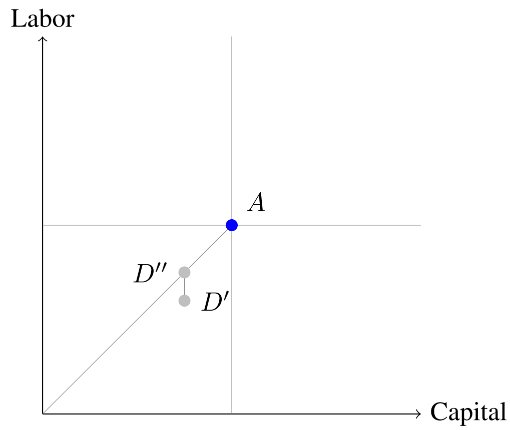


Figure 15.2: Any warehouse that, like D', lies within the southwest quadrant determined by A uses less of each input than A to achieve the same output. But only warehouses like D'' on the line connecting A to the origin allow direct comparison with A.

Comparing warehouses that are quite different

Now let us consider a more general problem: How does warehouse A compare to E and F:

	Labor (hrs $\times$ 10^3)	**Capital ($M)**	**Annual pick-lines (M)**
Warehouse A	100	1.00	1.60
Warehouse E	55	0.15	0.80
Warehouse F	40	2.60	3.20

Again, we scale all warehouses to the same level of output so that we may compare them on equal terms:

	Labor (hrs $\times$ 10^3)	**Capital ($M)**	**Annual pick-lines (M)**
Warehouse A	100	1.00	1.60
Warehouse E'	110	0.30	1.60
Warehouse F'	20	1.30	1.60

It is not possible to say whether warehouse E' or F' is more efficient than A, or not. Each is more efficient in one input but less efficient in another. It does not seem possible to usefully compare the warehouses by way of simple comparison of ratios. But when

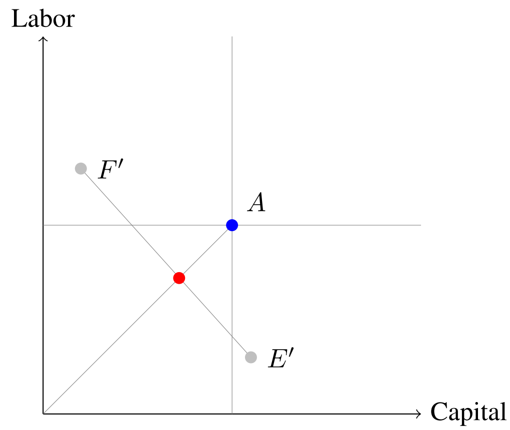


Figure 15.3: The synthetic warehouse (red) is a blend of scaled warehouses E' and F'; and it reveals warehouse A to be no more than 0.72 efficient.

we plot these warehouses, as in Figure 15.3, we see that a warehouse that blends E' and F' *can* be directly compared with warehouse A; and this benchmark warehouse achieves the same output as A but consumes only 0.72 of the inputs. By constructing this synthetic warehouse—blended from scaled versions of actual warehouses E and F—we have established that warehouse A deserves an efficiency score no better than 0.72. We have constructed a hypothetical warehouse that blends the best of warehouses E and F and that can be compared directly with A.

It is important to realize that this blending relies on another assumption, that any convex combination of two warehouses is realizable as another warehouse. Again, this is a subtle point. It seems reasonable when the warehouses use similar equipment to distribute similar product. But it is not clear how one could realize a convex combination of, for example, a service parts warehouse and finished-goods warehouse. The best way to avoid these sorts of questions is to base benchmarking on a restricted community of warehouses that are not "too different".

Benchmarking against a community of warehouses

Now we come to the heart of the question of benchmarking: How well does the warehouse of concern compare to a *community* of warehouses? For example, consider the collection of warehouses plotted in Figure 15.4.

We will rephrase the question as this: What is the most efficient synthesis of existing warehouses that can be compared directly with warehouse A? Any candidate must lie along the *efficient frontier*—(the dashed line in Figure 15.4—that represents all scaled and blended warehouses that are not dominated by any other. The benchmark warehouse is that one that consumes inputs in exactly the same proportion as warehouse A and so may be compared directly with it.

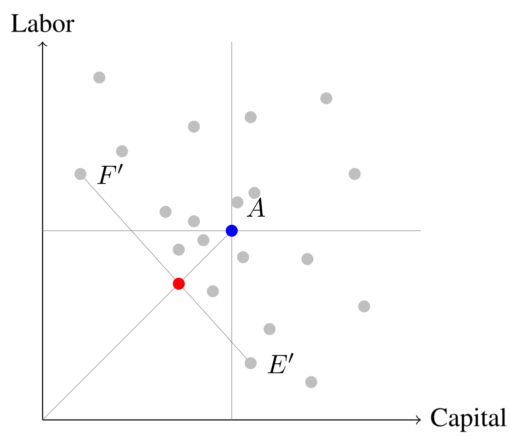


Figure 15.4: The benchmark warehouse (red) may be considered to have taken all of the best ideas from other warehouses and blended them perfectly to expose inefficiencies in A.

Constructing the benchmark warehouse

So how do we find the benchmark warehouse, that one that suggests what might be possible if we could blend the best ideas of all others? Fortunately, we can use algebra to explore and manipulate the geometrical model. In fact, for any target warehouse A, we can synthesize a benchmark warehouse, a scaled and blended warehouse that can be compared directly to A and that most dramatically reveals any weaknesses of A. Furthermore, this direct comparison will allow us to assign an efficiency score in the range $(0, 1]$ to A.

This evaluation is done by linear programming and here is the formulation that computes the relative efficiency of warehouse A in comparison to a collection of $i = 1, \ldots, n$ other warehouses. Suppose that warehouse i uses I_{ik} units of each input k to produce outputs O_{ij}. Then the efficiency score of warehouse A may be determined by solving the following linear program.

$$\min \theta \text{ subject to} \tag{15.1}$$

$$\sum_{i=1}^{n} O_{ij}\lambda_i \geq O_{Aj} \text{ for outputs } j \tag{15.2}$$

$$\sum_{i=1}^{n} I_{ik}\lambda_i \leq I_{Ak}\theta \text{ for inputs } k \tag{15.3}$$

$$\sum_{i=1}^{n} \lambda_i = 1 \tag{15.4}$$

$$\lambda_i \geq 0 \tag{15.5}$$

This linear program constructs a synthetic warehouse by scaling and blending existing warehouses, where λ_i represents the intensity of the contribution of warehouse i. The resulting synthetic warehouse has the following properties.

- It produces at least as much of each output j as warehouse A (constraints 15.2).
- It consumes no more than fraction θ of each input k used by warehouse A (constraints 15.3), with θ as small as possible (objective 15.1).

The standard methods of linear programming will construct a basic feasible solution to problem 15.1 and so the number of $\lambda_i > 0$ in the optimal solution will not exceed the number of inputs plus the number of outputs. Typically this will be many fewer than the total number of warehouses in the benchmarking community and so the set of warehouses i represented in the basic feasible solution form a small subset of peers to which the target is compared.

Finally, to evaluate each of a community of warehouses, such a linear program would have to be solved for each target warehouse.

Use

This technique is called *data envelopement analysis* (DEA) and has been used to study the efficiency of many complex economic systems [17, 24].

In a richer DEA model than we have presented, there would be tens of inputs and tens of outputs, as in the model of Hackman *et al.* [27], and it would be impossible to analyze the community of warehouses by plotting points.

DEA has some important advantages over simple ratio-based benchmarking. First, it assigns a single number as a score for efficiency, and most people find this easier to comprehend than a vector of KPI's. The score is not a simple vote by KPI's or anything like that, but rather arises organically from multi-dimensional comparison with a community of other warehouses. In practice, such benchmarking can be done by a neutral third-party, who holds the data, while users can remain anonymous if they so choose.

To work best, such benchmarking requires a large data set, preferably with hundreds of warehouses. It also requires that participants enter accurate data, especially because assigned scores tend to be determined by the best warehouses, and so outliers can skew the results.

Conclusions of the Georgia Tech study

The Georgia Tech study conducted by S. Hackman and colleagues reached three conclusions [27]:

- There were no discernible differences between union and non-union warehouses. This seems contrary to conventional wisdom.
- Warehouses with low capital investment tended to outperform those with high capital investment. Presumably this is due to the high cost and inflexibility of automation.
- Smaller warehouses tended to outperform larger warehouses.

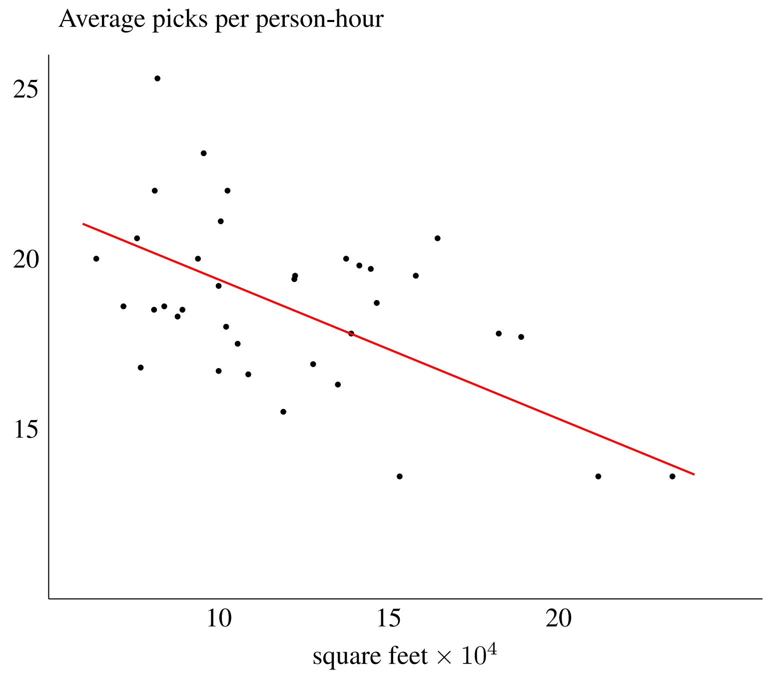


Figure 15.5: Pick rates at 36 similar warehouses as a function of the floor area.

15.3 Are smaller warehouses more efficient?

The GT benchmarking survey indicated that smaller warehouses are, in general, more efficient that larger ones. Why is that? Where are the benefits expected from economies of scale?

The short answer is that a larger facility can be more efficient *only if it changes its processes*.

Pure size hurts efficiency. If the warehouse is physically larger simply because it holds more of each sku, it will be *less* efficient. The additional product occupies more space, which creates more travel and makes the warehouse less efficient. This is illustrated in Figure 15.5, which shows the pick-lines per person-hour, a rough measure of productivity, in 36 warehouses of the same US company. All 36 distribute the same skus and have roughly the same levels of automation, information technology, and other resources. This special data set allows a very clean comparison based on size alone.

The straight line in Figure 15.5 is a best fit through the data points; and it is clear that productivity decreases with size.

Part of the confusion in this issue arises over what exactly is meant by “larger”. For the 36 warehouses above, larger means simply “more floor area”; everything else is the same. Sometimes “larger” is used to mean “holds more skus” or “receives more orders” or “receives orders for more pick-lines” or “receives orders for greater quantities". If a warehouse is larger for any of these reasons, it can lose efficiency unless it changes processes. Additional skus occupy additional space, which increases travel. Furthermore, more space is required for accessibility: If the warehouse is scaled by a factor of α, then the total length of aisles to reach all locations increases by factor α^2; and the aisle area increases by α^3 because greater material flow requires wider aisles. All of these factors increase travel and reduce efficiency.

Walmart is an example of an operation that has used size in all of the above senses to become more efficient. As they have grown, they have changed their processes to store and handle handle larger units (pallets instead of cartons; cartons instead of eaches); to crossdock instead of put-away; and to replace labor by automation and then run it 24x7 to realize return on investment.

15.4 Questions

Question 15.1. *What are the key assumptions underlying data envelopement analysis? For each one, describe warehouse inputs that may reasonably be considered to satisfy these assumptions and some that are questionable.*

Question 15.2. *Consider the following three warehouses.*

	Labor (hrs $\times 10^3$***)***	***Capital ($M)***	***Annual pick-lines (M)***
Warehouse A	100	1.00	1.0
Warehouse B	100	2.00	2.0
Warehouse C	50	0.25	0.5

Using DEA graphically, describe the ideal warehouse to which warehouse A *should be compared. What is the efficiency score of warehouse* A *when compared to* B *and* C*?*

Question 15.3. *Write the linear program whose solution would answer Question 15.2*

Question 15.4. *What factors do you think are likely to explain why some of the warehouses represented in Figure 15.5 lie above the trend line and some below it?*

Part VII

Miscellaneous

This is a collection of shorter pieces touching various issues of warehousing and distribution.

The presentation to date has focused on warehousing and distribution in the great centers of consumption, North America and Europe, and so reflects the concerns of an environment in which labor is relatively expensive. Of course, this is not the case for much of the world and, unsurprisingly, warehouses reflect this.

It is worth reminding the reader that most of the tools and insights in the book remain useful even when labor costs are relatively low. First, any savings are worth realizing. Second, as we have often remarked in the text, when one can realize savings in time (labor), it is frequently possible to "give back" some of that savings and take it in the form of space (capital). Whether this is worthwhile depends on the exchange rates of time and space and on extraneous requirements.

Chapter 16

Warehousing around the world

Almost every warehouse is optimized — for something. The result depends on the business issues and the relative costs of space, labor, and capital.

Here is a brief survey of how warehousing issues differ around the world.

16.1 North America

North America is driven by mass consumption. Think WalMart. This enables huge economies of scale and, indeed, the trend has been for ever larger distribution centers and ever accelerating rates of product flow. As telecommunications enables better coördination along the supply chain, the uniformity of market and of distribution infrastructure allows fewer, more centralized and therefore larger distribution centers. The Amazon.com distribution center shown in Figure 16.1 is typical: One level, with conveyors and sortation equipment but little other significant automation. Such warehouses are generally built in the countryside surrounding major metropolitan areas, so that land is cheap but there is still ready access to large markets.

The fairly high costs of labor are held down somewhat by constant immigration into the US and Canada.

Warehouses in North America are coördinated by increasingly sophisticated warehouse management systems and so very rich data sets are available with which to evaluate and refine performance.

16.2 East Asia

Business in Asia has traditionally been based on personal relationships and less on computational models. Because of this tradition, data is not robust and not widely available; consequently the opportunities to improve operations by science are not fully developed at present.

In general, the most active economic areas are separated by lots of water, which means lots of product conveyed by air (for high-value or time-sensitive products) or

Figure 16.1: This Amazon.com distribution center is typical of the large, high-volume DCs in North America.

ship (for bulky items or commodities). For both air and sea cargo, the large fixed costs increase incentives to consolidate freight. Consequently one expects to see the emergence of strong regional hubs, such as Singapore and Hong Kong, to support this consolidation.

16.2.1 India

India, like many developing countries, has both inexpensive land and low labor costs. Capital costs are relatively high in relation to the cost of labor and so there is less pressure to install specialized storage, even pallet rack. And because labor costs are low there is less incentive to increase efficiency. For example, it is not an attractive proposition to reduce labor costs by picking from flow rack: The labor savings cannot justify the cost of the rack or the forklift trucks.

In addition, warehouses in India distribute mainly to the local economy and so supply a market that is not wealthy. Consequently, the skus are not likely to be high cost items and so there is not much savings available from reducing inventories by precise timing. Consequently information technologies cannot generate much savings.

Finally, inefficiencies in transport make India in effect a collection of smaller markets. These inefficiencies include the physical, such as roads in less-than-ideal condition, as well as the administrative. For example, each state within India levies customs duties on freight transported across the border. This slows interstate commerce and increases the expense. Such factors increase the costs of transportation and so favor a strategy of having more, smaller distribution centers rather than fewer, larger ones, where the volume of activity could better justify capital investment. The national government is attempting to revise its tax structure fix these inefficiencies.

Figure 16.2: The relatively low cost of labor, high cost of capital, and artificially small market mean that this warehouse in India may be economically efficient. (Photo courtesy of Rohan Reddy)

India is increasingly becoming a global sourcing hub and so suitable distribution centers are being built in around large ports, such as Mumbai (Bombay). However, land can be expensive there. Caleb Tan of Menlo Worldwide Logistics has observed prices comparable to those of Singapore or Hong Kong. Apparently this is due to a lack of land because of encroachment of slums as more and more people migrate from the countryside to economically vibrant areas.

16.2.2 China

A distinctive feature of the logistics scene in China is the seemingly boundless supply of very low cost of labor together with relatively inexpensive land. Consequently warehouses tend to be large, low buildings as in North America; but with some striking differences. For example, it is not unusual as of this writing to find a warehouse of 250,000 square feet with a single fork lift truck. The reason is that equipment is expensive but labor is cheap (Figure 16.3).

Despite cheap labor, China does have some capital-intensive warehouses, with the latest information technology and storage equipment. Such warehouses are most likely devoted to the distribution of high-value goods for export. Because such goods, such as consumer electronics, have high-value and short life-cycle, the warehouses can justify their equipment by substantial reductions in inventory costs.

The very different costs in the US and China sometimes leads to behavior that makes sense locally but may make the supply chain inefficient. For example, The Home Depot receives some Chinese-built product at its Import Distribution Center in Savannah, Georgia, USA. The shipping department in the warehouse in China de-palletizes freight in order to pack each trailer as tightly as possible for the drive to the sea port. Thus an expenditure of relatively cheap labor will reduce the relatively significant costs of equipment and transportation. But this means the product arrives in the US as loose cartons in containers and so The Home Depot must re-palletize the cartons before storage in deep, drive-in pallet rack. And most of it will, shortly after, be de-palletized once more when it is picked as cartons for shipment to stores.

Figure 16.3: In the US warehouse on the left, cartons of beer have been palletized because labor is expensive compared to capital. The reduction in labor is worth the expense of a forklift plus the additional storage space. In the Chinese warehouse on the right, cartons have been stacked by hand and must be unstacked by hand; but labor is cheap and capital is expensive.

16.2.3 Singapore, Hong Kong, Japan

Some economic powers such as Singapore, Japan, and Hong Kong suffer from limited space so land is much more expensive than elsewhere. Consequently, many of the warehouses are high-rise, such as shown in Figure 16.4. In addition, as first-world economies, labor in these places is expensive and so warehouses here are more likely to be automated. Freight elevators are likely to be bottlenecks to material flow in these facilities.

Space constraints have led to an interesting type of warehouse in Hong Kong and in Singapore: A multi-floor facility with no automation or elevators, but, instead, a spiral truck ramp so that trailers may be docked at any floor (Figure 16.5). In effect, each floor becomes a ground floor—but the cost is that significant land area, determined in part by the turning radius of a truck, is lost to the ramp and unavailable for storage. That storage space must be reclaimed up above.

This design is a clever and efficient way of using space if each floor is occupied by an independent tenant. But one must be careful if some tenant occupies multiple floors for then it may become necessary to shuttle trucks among floors. For example, a multi-story warehouse with spiral ramp would be unsuitable as a local distribution center: Trailers departing with small shipments for many customers must be loaded in reverse sequence of delivery to avoid double-handling. But since the load may not match the layout of product amongst the floors, this could require much shuttling of the trailer among floors to load.

Similarly, this might be an inefficient warehouse in which to receive shipments of diverse items that may be stored on different floors: Because rear-entry trailers restrict the sequence in which freight can be accessed, a trailer may have to shuttle among the floors to deliver all the freight to the appropriate places. Alternatively, the trailer would have to be loaded to match the allocation of product among floors at the warehouse. In either case, extra work is required.

Figure 16.4: Multi-story warehouses are common in Singapore, Hong Kong, and Japan where land is expensive.

Figure 16.5: Multi-story warehouse with no automation but accessibility provided by a spiral truck ramp.

Sembawang Kimtrans claims to be among the first with a spiral truck ramp. The main function of their Singapore facility is storage: It receives truckloads of product from the sea port or from factories in Malaysia and stores it for subsequent distribution to factories within Singapore. Thus a typical trailer-load, either in-bound or out-bound, is full of pallets of a single sku and so routing between floors is not an issue.

Over 50% of the product movement through this facility is by container.

The warehouse is 5 stories tall, with the first four floors used for warehousing and the fifth floor occupied by administration. Some of the floors are leased out to other companies and trucks to those warehouses are unlikely to visit other floors.

Each floor has four docks, each of which can accommodate a 45-foot trailer. The facility typically handles about 100 trailers in an 8-hour day.

About 30% of the space of the facility is consumed by the spiral ramp. Some of the ramp space is reclaimed for warehouse use as the hollow core of the ramp is used for miscellaneous handling of product.

In effect, the trailers and spiral ramp function as an AS/RS (automated storage and retrieval system). The point of this design is to get space utilization by building high storage; but to avoid installing automation, which can be a risky investment because inflexible. It also enables the owner to rent floors to different tenants.

Even simple automation, such as a freight elevator (lift), can be problematical because it is subject to queues and congestion, at least when deliveries are unscheduled, as they would be if different tenants occupy the facility.

The spiral ramp can simultaneously support more than 4 trucks traveling in each direction. It would take a bank of at least 8 elevators to accomplish the same.

In addition, an elevator requires maintenance, which is not the case for the spiral ramp.

Finally, it must be observed that the most significant savings may be the reduction in land requirements. For example, if one imagines that the Sembawang Kimtrans facility in Singapore puts three warehouses in the space of 1.5, the land savings may be enough to pay for the cost of the building.

Incidentally, Hong Kong and Singapore make an interesting comparison. Both are premier logistics hubs due to splendid harbors, airports, and IT infrastructure; but they serve different purposes: Singapore is a point of transshipment for much manufacturing leaving Thailand, Malaysia, Indonesia, and other ASEAN nations and so it must be highly attuned to the challenges of handling international freight: Relatively little of it stays in Singapore; instead freight is likely to arrive from one country, receive value-added processing, and be forwarded to another country.

On the other hand, Hong Kong acts primarily as a logistics hub for manufactured goods leaving China. Consequently it receives goods from the same country, and frequently by truck or train; but it dispatches them overseas by ship or air.

16.3 Central and South America

This is a region of developing markets that are separated by geography such as the Amazonian rain forest and the Andes mountains. Consequently, markets tend to be relatively small and inappropriate for significant capital investment in warehouses.

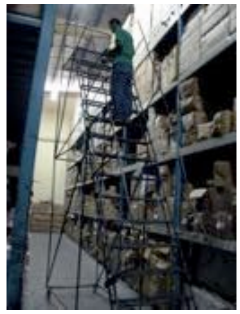


Figure 16.6: A ladder is much cheaper than a person-aboard truck, though much slower. (Note the product stored as loose cartons.)

Data may not be available or may not be transmitted to supply chain partners. In part this is because there has not been a strong, reliable stock market and so wealth has generally been invested in family businesses, which are less inclined to share data.

Labor costs are relatively low, but some segments of the workforce enjoy strong political protections, which results in labor inflexibility. Consequently, while labor can be cheap, it can be hard to shrink or re-deploy a workforce. Most of labor is devoted to handling small quantities, because retail stores are small and customers purchase tiny amounts daily.

Space is the main concern. Infrastructure is not well-developed and so, to reduce travel on bad roads, warehouses must locate close to the customer. But customers are concentrated in a few, very large cities such as Mexico City, Sao Paolo, and Lima, which are congested and where space is expensive.

Most warehouses in this region hold goods for domestic distribution. Because markets are not highly developed, inventory tends to be of less expensive goods and so the costs of moving inventory slowly are relatively low. (In contrast, China has many distribution centers supporting export of high-value goods, such as consumer electronics. The high cost of holding such inventory, justifies investment in facilities that enable rapid movement of product. This also holds for Mexico, which can get product quickly to market in the US and so able to justify advanced distribution centers.)

The photograph of Figures 16.6 reflects the low labor costs relative to the costs of capital in this region.

There is also a distinctive architecture to warehouses in Central America, an area with high rainfall. Warehouses are built with steeply pitched roofs to shed rain, as in Figure 16.7. This makes it hard to use all the vertical space because it is uneconomical to purchase a high-reach forklift truck when only a small part of the warehouse has high ceilings.

16.4 Europe

Warehouses in Europe, especially in Germany and France, are shaped by the relatively high labor costs and inflexibility of the work force. These facts push designers to find

Figure 16.7: To shed rain, this roof is steeply angled, which makes it more difficult to utilize vertical space within.

Figure 16.8: A highly automated distribution center in Germany. (Photo courtesy of Kai Wittek)

engineering solutions rather than social solutions to logistics challenges. For example, there is a greater inclination to use automation than in comparable facilities in North America.

In the past, the economies of Europe were separate. More recently the economies are integrating into a common market, which will create economies of scale, which will likely lead to larger warehouses. However, urban areas, many of which have grown out of ancient towns, will still present challenges to the efficient flow of product.

All this is reflected in Figure 16.8, which shows a distribution center of a major drugstore chain in Germany. The multi-story portion of the building, visible in the background, houses a high-rise automated storage-and-retrieval system. This is not so much to conserve space as to reduce labor costs. (Unlike the Singapore warehouses, this facility is tall only where the AS/RS is installed.)

According to the Rossman web page, the AS/RS is 30 meters high, the aisles are 127 meters long and the facility provides 14,000 pallet positions. The delivery trucks in the foreground are relatively small, at least compared to the 48-foot trailers common in North America. The small trucks are necessary to deliver to stores in the centers of cities, the ground plans of which may have been laid out centuries ago and cannot accommodate large trucks. But the warehouse is large, reflecting the extent of the market it serves.

Chapter 17

In preparation

These topics are generally presented in class and exist at this moment as slides and class notes. As soon as they become presentable to a more general audience they will appear in the text proper:

- Cycle-counting: What to count and when, to reconcile book and physical inventories
- Staff-scheduling: How many workers to hire and when/where should they work to meet shipping schedules
- Warehouse location: Where to locate that next warehouse or crossdock
- An introductory chapter on the role of optimization in warehouse design and operations.

Any other topics you would like to see? If so, please contact the authors (`john.bartholdi@gatech.edu`).

Bibliography

[1] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, Englewood Cliffs, NJ., 2nd edition, 1996.

[2] K. T. Alligood, T. D. Sauer, and J. A. Yorke. *Chaos: An Introduction to Dynamical Systems*. Springer, 1996. ISBN 0-387-94677-2.

[3] J. Y. Park B. C. Park and R. D. Foley. Carousel system performance. *Journal of Applied Probability*, 40(3):602–612, September 2003.

[4] J. J. Bartholdi, III. *Building Intuition: Insights from Basic Operations Management Models and Principles*, chapter The Knapsack Problem, pages 19–32. Springer, 2008.

[5] J. J. Bartholdi, III, L. A. Bunimovich, and D. D. Eisenstein. Dynamics of two- and three-worker "bucket brigade" production lines. *Oper. Res.*, 47(3):488–491, 1999.

[6] J. J. Bartholdi, III and D. D. Eisenstein. The bucket brigade web page. http://www.bucketbrigades.com.

[7] J. J. Bartholdi, III and D. D. Eisenstein. A production line that balances itself. *Oper. Res.*, 44(1):21–34, 1996.

[8] J. J. Bartholdi, III and K. R. Gue. Reducing labor costs in an ltl cross-docking terminal. *Oper. Res.*, 48(6):823–832, 2000.

[9] J. J. Bartholdi, III and K. R. Gue. The best shape for a crossdock. *Transportation Sci.*, 38(2):235–244, 2004.

[10] J. J. Bartholdi, III and S. T. Hackman. Allocating space in a forward pick area of a distribution center for small parts. *IIE Transactions*, 40(11):1046–1053, November 2008.

[11] J. J. Bartholdi, III and L. K. Platzman. Retrieval strategies for a carousel conveyor. *IIE Transactions*, 18:166–173, 1986.

[12] J. J. Bartholdi, III, T. D. Seeley, C. A. Tovey, and J. H. Vande Vate. The pattern and effectiveness of forager allocation among food sources in honey bee colonies. *Journal of Theoretical Biology*, 160:23–40, 1993.

[13] G. Bengü. An optimal storage assignment for automated rotating carousels. *IIE Transactions*, 27(1):105–107, 1995.

[14] G. Berbeglia, J.-F. Cordeau, and G. Laporte. Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202(1):8–15, 2010.

[15] Y. A. Bozer and J. A. White. Travel time models for automated storage/retrieval systems. *IIE Transactions*, 16(4):329–338, 1984.

[16] Y. A. Bozer and J. A. White. Design and performance models for end-of-aisle order picking systems. *Management Science*, 36(7):852–866, 1990.

[17] A. Charnes, W. W. Cooper, A. Y. Lewin, and L. M. Seiford. *Data Envelopment Analysis: Theory, Methodology, and Application*. Kluwer Academic Publishers, Boston, 1994.

[18] T. H. Corman, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[19] György Dósa. The tight bound of first fit decreasing bin-packing algorithm is FFD$(l) \leq (11/9)$ OPT$(l)+6/9$. In *Combinatorics, Algorithms, Probabilistic and Experimental Methodologies*, volume 4614/2007, pages 1–11. First International Symposium, ESCAPE 2007, Springer, April 2007.

[20] H. A. Eiselt, M. Gendreau, and G. Laporte. Arc routing problems, part ii: The rural postman problem. *Operations Research*, 43(3):399–414, May–June 1995.

[21] E. H. Frazelle. *World-Class Warehousing*. Logistics Resources International, Atlanta, GA, 1996.

[22] E. H. Frazelle. *World-Class Warehousing and Material Handling*. McGraw Hill, New York, New York, 2002.

[23] G. N. Frederickson, M. S. Hecht, and C. E. Kim. Approximation algorithms for some routing problems. *SIAM J. Comput.*, 7(2):178–193, 1978.

[24] H. O. Fried, S. S. Schmidt, and C. A. Knox Lovell. *The Measurement of Productive Efficiency: Techniques and Applications*. Oxford University Press, 1993.

[25] R. Graham. The combinatorial mathematics of scheduling, 1978.

[26] K. R. Gue and R. D. Meller. Aisle configurations for unit-load warehouses. *IIE Transactions*, 41(3):171–182, 2009.

[27] S. T. Hackman, E. H. Frazelle, P. M. Griffin, S. O. Griffin, and D. A. Vlatsa. Benchmarking warehousing and distribution operations: an input-output approach. *Journal of Productivity Analysis*, 16:241–261, 2001.

[28] S. T. Hackman and M. J. Rosenblatt. Allocating items to an automated storage and retrieval system. *IIE Transactions: Industrial Engineering Research and Development*, 22(1):7–14, 1990.

[29] R. W. Hall. Distance approximations for routing manual pickers in a warehouse. *IIE Trans.*, 25(4):76–87, 1985.

[30] J. L. Heskitt. Cube-per-order index—a key to warehouse stock location. *Transportation and Distribution Management*, 3:27–31, 1963.

[31] S. A. Jernigan. *Multi-Tier Inventory Systems With Space Constraints*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2004.

[32] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM J. Comput.*, 3(4):299–325, 1974.

[33] E. G. Coffman Jr., M. R. Garey, and D. S. Johnson. Approximation algorithms for bin-packing — an updated survey. In Ausiella, Lucertini, and Serafini, editors, *Algorithm Design for Computer System Design*, pages 49–106. Springer-Verlag, 1984.

[34] C. Kallina and J. Lynn. Application of the cube-per-order index rule for stock location in a distribution warehouse. *Interfaces*, 7(1):37–46, 1976.

[35] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Schmoys, editors. *The Traveling Salesman Problem*. John Wiley & Sons, 1985.

[36] J. D. C. Little. A proof of the queueing formula $L = \lambda W$. *Operations Research*, 9:383–387, 1961.

[37] N. Litvak. Optimal picking of large orders in carousel systems. *Operations Research Letters*, 34:219–227, 2006.

[38] H. Luss and S. Gupta. Allocation of effort resources among competing activities. *Oper. Res.*, 23:360–366, 1975.

[39] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementation*. John Wiley & Sons, 1990.

[40] Hannu Nurmi. *Voting Paradoxes and How to Deal with Them*. Springer, 1999.

[41] H. D. Ratliff and A. S. Rosenthal. Order-picking in a rectangular warehouse: A solvable case of the traveling salesman problem. *Oper. Res.*, 31(3):507–521, 1983.

[42] K. J. Roodbergen and R. de Koster. Routing methods for warehouses with multiple cross aisles. *Int. J. Prod. Res.*, 39(9):1865–1883, 2001.

[43] K. J. Roodbergen and R. de Koster. Routing order pickers in a warehouse with a middle aisle. *Eur. J. Oper. Res.*, 133:32–43, 2001.

[44] L. Schwarz. *Building Intuition: Insights from Basic Operations Management Models and Principles*, chapter The economic order-quantity (EOQ) model, pages 135–154. Springer, 2008.

[45] E. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, New Haven, CT, 1983.

[46] E. Tufte. *Envisioning Information*. Graphics Press, New Haven, CT, 1990.

[47] E. Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, New Haven, CT, 1997.

[48] J. P. van den Berg. Multiple order-pick sequencing in a carousel system: A solvable case of the rural postman problem. *Journal of the Operational Research Society*, 47(12):1504–1515, December 1996.

[49] J. P. van den Berg, G. P. Sharp, A. J. R. M. Gademann, and Y. Pochet. Forward-reserve allocation in a warehouse with unit-load replenishments. *European Journal of Operational Research*, 111:98–113, 1998.

[50] R. G. Vickson and A. Fujimoto. Optimal storage locations in a carousel storage and retrieval system. *Location Science*, 4(4):237–245, 1996.

Part VIII

Appendices

Appendix A

Inventory and the economic order quantity (EOQ)

A.1 The Economic Order Quantity

There is a rich literature on inventory theory and it will not be surveyed here. Instead, we shall summarize a few key results that will be useful to our view of inventory within the warehouse.

First, it should be observed that inventory theory typically presents a financial, not an operational, point of view. In particular, inventory models generally do not contain any details about units of measure or space required. Nevertheless, they may guide high level decisions, such as how much inventory to order and when, and consequently they have implications for warehouse operations.

Figure A.1 shows a year's worth of the daily sales of a cutting wheel from an east-coast hardware distributor. This particular product is the one most frequently requested from this distributor. The product maintains a fairly constant rate of sales throughout the year. It is a mature consumable product, with no complications of style or fashion, and so is apparently free of the strong seasonalities typical of many consumer goods. (There seems to be a slight increase in sales during the first quarter, perhaps when their customers, which are mostly maintenance shops, get their annual budget renewed.)

Figure A.2 graphs the daily inventory in the warehouse. It displays a distinctive saw-tooth pattern in which the following cycle repeats itself: Sales depletes inventory on hand until the product is reordered from the manufacturer; when the ordered inventory arrives, it becomes available for purchase and the inventory jumps up, to begin a new cycle.

The most basic inventory model is an idealization of Figure A.2. Imagine a single stock-keeping unit that, when stocks are depleted, is reordered in quantity Q and becomes instantly available. Thereafter, the inventory level is reduced by customer demand that is constant and predictable, of rate D units of inventory per unit of time. In this idealization the inventory level over time appears as a as in Figure A.3.

The greatest simplification in this model is the assumption that customer demand is

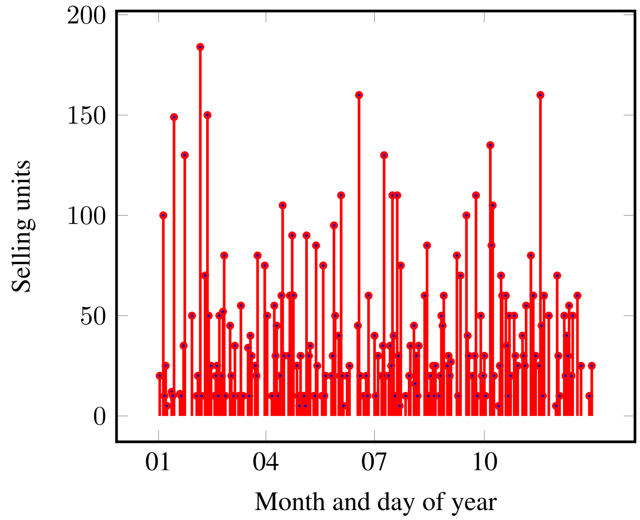


Figure A.1: Daily sales of a cutting wheel

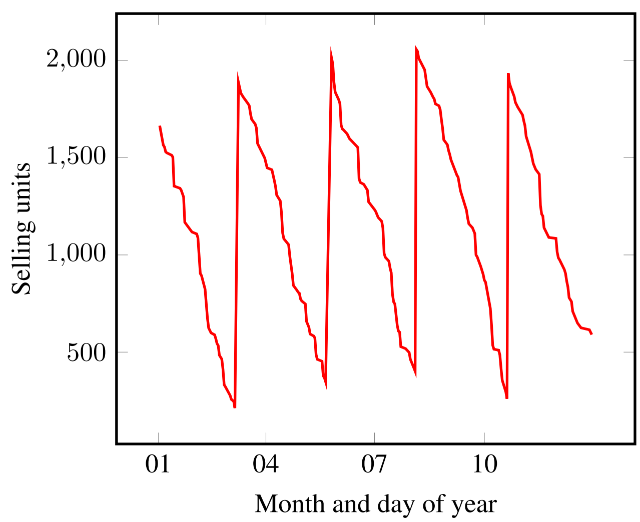


Figure A.2: Daily inventory of a cutting wheel

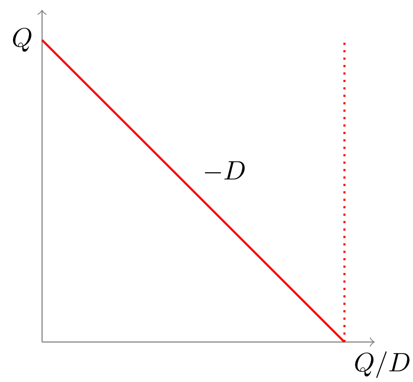


Figure A.3: Inventory Level $I(t)$ Over Time

constant. In actual fact, customer demand varies over time and may be intermittent and erratic. Figure A.1 describes the most popular product in the warehouse; consider what a chart of daily demand would look like for a product that is ordered very infrequently: perhaps a few intermittent spikes scattered throughout the year. Such a product would not be well-described by this model. Nevertheless, this model can be useful, and it provides the point of reference for most other inventory models.

It is worth pausing to point out some implications of this model that will be useful in the context of warehouses. First, the average inventory over time is $Q/2$. This means that the average time in inventory of this particular item is $Q/(2D)$. This expression is frequently called *turnover time*. The inverse value is the average number of inventory *turns*. Note that the number of turns is not the same — indeed, is twice the value of — the number of times the item is ordered and replenished.

The obvious question to ask is what is the best quantity Q to order? To answer this we must enlarge the model by including some costs. The main costs common to inventories are the costs of reordering and the costs of holding inventory. These are modeled as:

c_T Transaction (reordering) cost; the total variable costs of placing and receiving an order. This might include labor in the purchasing department and the cost of transporting the merchandise. It would *not* include the costs, for example, of receiving the cartons into the warehouse (counting, labeling, scanning, etc.) as such costs are fixed with respect to the order quantity and depend only on the total number of cartons demanded by customers over the year.

c_H The variable cost of holding inventory. This is approximated most simply as a fraction of the purchase price, to reflect the cost of money. For example, if the cost of money is 20% annually and one widget cost \$200, then the cost of holding a widget for one year is estimated to be \$40. For some products, such as consumer electronics or fashionable apparel, there may be an additional component to capture costs of obsolescence.

An order cycle is initiated by the placing of the order, which incurs cost c_T. Thereafter, holding costs are accrued for inventory-time (for example, widget-years), which is the area under the inventory curve (Figure A.3. Summing these two cost components gives the total cost of one inventory cycle:

$$c_T + c_H\left(\frac{Q^2}{2D}\right).$$

The cost per unit time is given by dividing the cost per cycle by the duration Q/D of a cycle, which gives:

$$\left(\frac{c_T D}{Q}\right) + c_H\left(\frac{Q}{2}\right).$$

Minimizing this rate of cost gives the *economic order quantity*:

$$Q^* = \sqrt{2D\left(\frac{c_T}{c_H}\right)}, \tag{A.1}$$

from which it follows that the optimal order cycle is of length

$$\frac{Q^*}{D} = \sqrt{\left(\frac{2}{D}\right)\left(\frac{c_T}{c_H}\right)}. \tag{A.2}$$

A.2 Safety stock and reorder points

The basic inventory model can be made more realistic by reflecting the fact that reordered inventory does not appear instantaneously. Instead, some *lead time* is required between the placing of an order and receipt of the merchandise. For example, in Figure A.4 the lead time is given by LT, from which it follows that the order for replenishment must be submitted as soon as inventory falls to level LTD.

This inventory system would work perfectly except for the fact that customer D is not constant, nor is lead time LT. Customer demand during lead time might be unexpectedly reduced because bad weather discourages in-store shopping; or lead time might be lengthened because of a strike among transportation workers. Because there are myriad hard-to-predict events that can affect lead-time demand, it is typically modeled as a random variable.

Most businesses want to avoid being caught unexpectedly out of stock. To protect against this they hold extra stock, *safety stock*, specifically to protect against the variance in lead-time demand. This cushion of safety stock is highlighted in Figure A.5.

The *reorder point* R is then set to be the sum of the expected lead-time demand plus a level of safety stock:

$$R = \text{LTD} + S. \tag{A.3}$$

The level of safety stock is set in hopes of ensuring a desired level of service (probability of not stocking out). If lead-time demand is described by an approximately normal distribution, then setting safety stock S to two standard deviations worth of inventory will provide almost 98% protection.

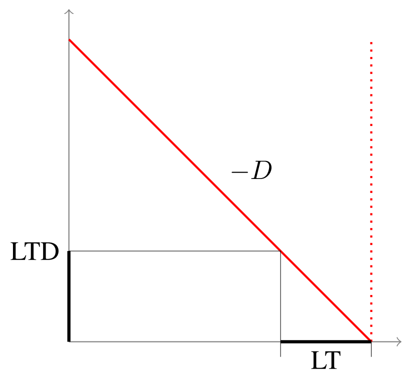


Figure A.4: Lead-time (LT) and customer demand during lead-time (LTD)

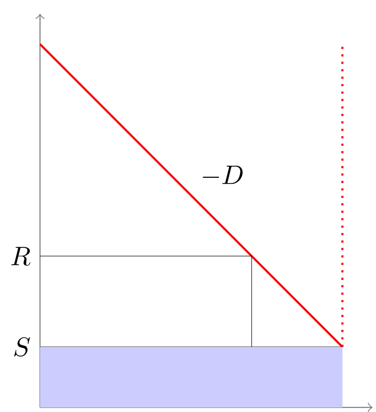


Figure A.5: Safety stock protects against variance in lead-time demand.

So: To set up a basic inventory control scheme for a stock-keeping unit requires selecting an appropriate reorder quantity Q; computing mean and standard deviation of lead-time demand LTD, with which to set an appropriate level of safety stock S; and setting the reorder point R to be the sum of expected lead-time demand and safety stock.

This model has been extended in countless ways. Readers will find a thorough review of the main ideas in [44].

A.3 Implications for the warehouse

For any particular sku the total inventory within a warehouse will tend to follow a pattern somewhat like the sawtooth curve: Replenishment, followed by more gradual depletion. This also tends to be true of each storage location with the warehouse: It will be filled, then depleted more or less gradually, and then refilled. Consequently one can think of every storage location in the warehouse "breathing" at some rate: a sharp intake of breath followed by more gradual exhalation.

One immediate implication is that, at any moment in time, some locations will be nearly full and some will be nearly empty, but—unless action is taken to avoid this—on average only half the storage space will be occupied.

Appendix B

The Knapsack Problem

The *Knapsack Problem* is the fanciful name given to a problem of linear optimization subject to a single constraint. The motivating example is traditionally given as follows. Suppose you want to pack your knapsack with items for a long day hike; what should you choose? The knapsack can hold only a limited volume, which we normalize to 1, and you may not exceed this limit. Each item i occupies a certain volume v_i (also normalized to lie within $[0, 1]$), and, if it is packed, confers value a_i, which we may assume to be positive (else the item would certainly not be packed).

The Knapsack Problem is an idealization of problem structure in which all decisions are limited primarily by the aggregate consumption of a single constrained resource, such as space in the knapsack.

If the items to be packed may be split arbitrarily, with part taken and part left behind, such as would be the case with drinking water, then the problem of choosing the most valuable subset of items to pack may be formalized as the following continuous-valued linear program (LP) in which the choice variables x_i may assume any value between 0 and 1, representing the fraction of item i packed:

$$\begin{aligned} \max \sum_{i=1}^{n} a_i x_i & \\ \sum_{i=1}^{n} v_i x_i & \leq 1 \\ x_i & \in [0, 1] \end{aligned}$$

This linear program, with but a single constraint, can be solved quite easily: Rank the candidate items by *bang-for-buck* a_i/v_i, which is a measure of the value delivered per unit of volume occupied; then choose those items with greatest bang-for-buck. When a point is reached at which the next item will not entirely fit in the knapsack, divide it and take only what will fit, so that the knapsack is entirely filled.

This is an optimal solution to the linear program by the following reasoning. Consider any solution for which less of some item has been packed even though it has greater bang-for-buck; that is, some $x_i < x_j$, but $a_j/v_j < a_i/v_i$. Such a solution cannot be optimal because a feasible solution of greater value can be constructed by decreasing x_j by ϵ, where $0 < \epsilon \leq x_j$, and increasing x_i by $v_j\epsilon/v_i$. The new solution is feasible because for sufficiently small ϵ the new values remain within $[0, 1]$. Furthermore, $v_i\left(x_i + v_j\epsilon/v_i\right) + v_j\left(x_j - \epsilon\right) = v_i x_i + v_j x_j$ and so the consumption of the constrained resource is unchanged. Finally, the new solution has a greater value because $a_i\left(x_i + v_j\epsilon/v_i\right) + a_j\left(x_j - \epsilon\right) = a_i x_i + a_j x_j + \left(\frac{a_i}{v_i}\right) v_j\epsilon > a_i x_i + a_j x_j + \left(\frac{a_j}{v_j}\right) v_j\epsilon = a_i x_i + a_j x_j + a_i\epsilon$, and the latter term is strictly greater than $a_i x_i + a_j x_j$ because $a_i > 0$.

The Knapsack Problem acquires a different character when each item, like a knife or a compass, must be either packed completely or else left behind. Now the appropriate formulation is as an integer linear program (ILP), in which the choice variables x_i are restricted to values either 0 or 1:

$$\max \sum_{i=1}^{n} a_i x_i$$

$$\sum_{i=1}^{n} v_i x_i \leq 1$$

$$x_i \in \{0, 1\}$$

This version of the Knapsack Problem is NP-Complete and so "hard" in a technical sense. However, in practice—and especially in the context in which we use this model—the Knapsack Problem is easy to solve approximately. The obvious approach is to try the same idea as for the LP: Pack the knapsack with items of greatest bang-for-buck until the next item does not fit. The result will be a "good" solution, in that it may be expected to be close to optimum.

This claim can be justified in a number of ways. Note that the greedy solution cannot differ from optimum by more than the value contributed by that last item tried, the first that does not fit. Call this the *stopping item*. If no item takes up too much space and if many items have been packed already, we might reasonably expect that, when the stopping item is encountered, there is not much space left in the knapsack and the stopping item does not have much value to contribute.

Even though the greedy solution may be expected to be close to optimum, it may be worthwhile to continue the search for easy, though small, improvements: After finding the stopping item, simply continue down the list of items, adding anything that fits.

The Knapsack Problem is a useful model when there is a single constraint that is known in advance to be most significant. For many of the problems we discuss this constraint is space. Space tends to be a significant constraint because it seems there is never enough, and adding space may require an expensive capital investment. On the other hand, time (in the form of labor hours) is rarely a hard constraint, at least in North America, because it can be increased incrementally, such as through the use of overtime or of temporary workers.

When there are additional constraints, the greedy procedure can no longer be relied upon because each choice might have a different value of bang-for-buck with respect to each constraint (limited resource). In this case, the greedy heuristic might not be able to reconcile the conflicting trade-offs in a single ranking of items, from most desirable to least.

For more on the Knapsack Problem, consult [4, 18] or the extensive reference [39], which is freely available on-line.

Appendix C

The Shortest Path Problem

A *graph* is a collection of vertices connected by edges, as in Figure C.1. The graph model may be enriched by associating a number with each edge, such as a distance. In such a model the *Shortest Path Problem* asks for the shortest path from a particular origin vertex to a destination vertex. This is a good model for situations in which travel is restricted to aisles, roads, or other relatively narrow paths where a multiplicity of choices makes it hard to know the shortest route to an intended destination.

Dijkstra's algorithm is the an elegant recursive method to identify the shortest path in a graph in which all edges are of non-negative length. It computes the length of the shortest route from a designated origin vertex to all other vertices by assigning each a tentative label that overestimate distances from the origin and then successively improving the estimate until it is exact.

Step 1: Initialize. Assign the origin vertex a permanent label of 0 and all other vertices tentative labels of ∞. Let the origin be the *current vertex* (Figure C.2).

Step 2: Starting at the current vertex v, relabel every neighboring vertex w with the smaller of its label or the label at the current vertex plus the length of the edge (v, w) (Figures C.3 and following).

Step 3: Examine all vertices with permanent labels and find a neighbor with smallest tentative label. (Ties may be broken arbitrarily.) This label correctly gives the shortest distance from s; make this the current vertex, make its label permanent, and return to Step 2.

Continuing in this way, at least one label is made permanent in each pass, and so the algorithm will soon halt. In a straightforward implementation the algorithm runs in time proportional to the square of the number of edges.

This is a much-studied problem and finds application in many contexts, most obviously in reducing travel time on road networks, where the huge size of the network can challenge even fast algorithms. Consequently there has been much research into how to speed up shortest path computations. However, most algorithms are still based on the approach described here (and which is easily fast enough for our purposes). Those

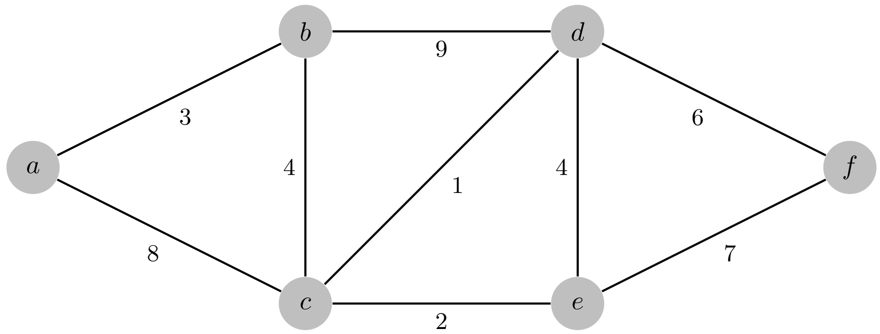


Figure C.1: A graph is a collection of vertices connected by edges. In this case each edge has an associated length that is non-negative. In this example, we find the shortest path from vertex a to vertex f.

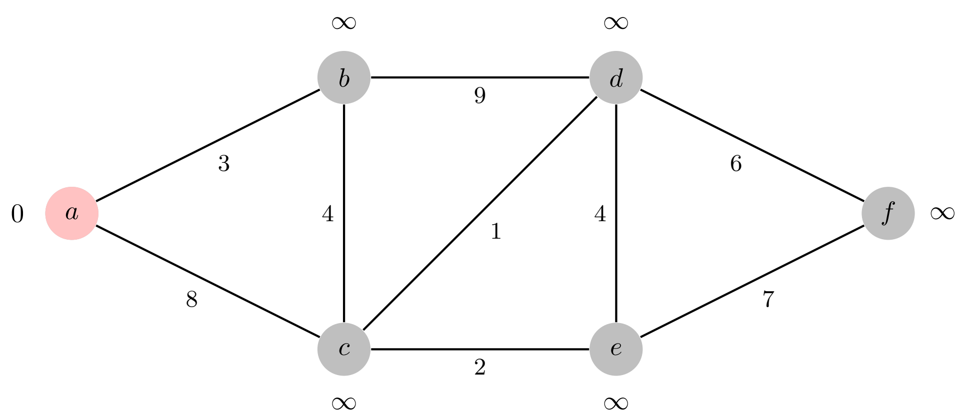


Figure C.2: Initialization: The origin vertex a is assigned a permanent label and the others are assigned tentative labels that overestimate the distances.

wishing to know more will find this material well-covered in [1, 18], which include proofs of correctness, analysis of running time, and extensions.

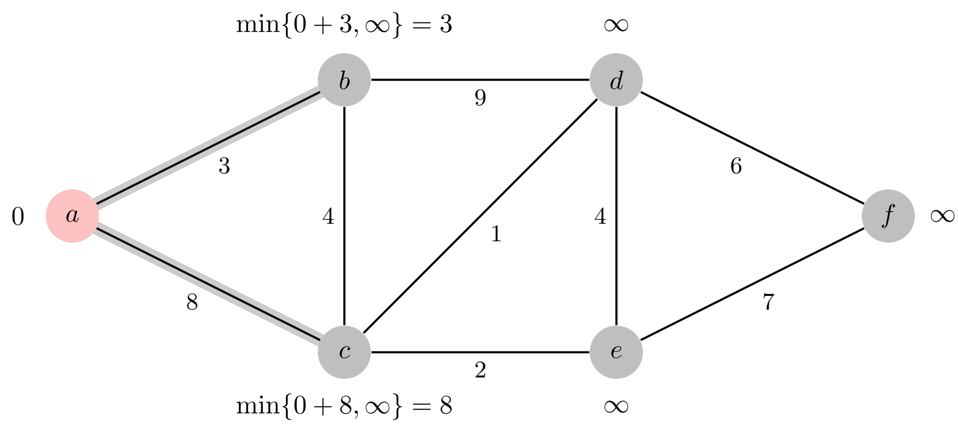


Figure C.3: Vertex a is the current vertex from which tentative labels of its neighbors are updated.

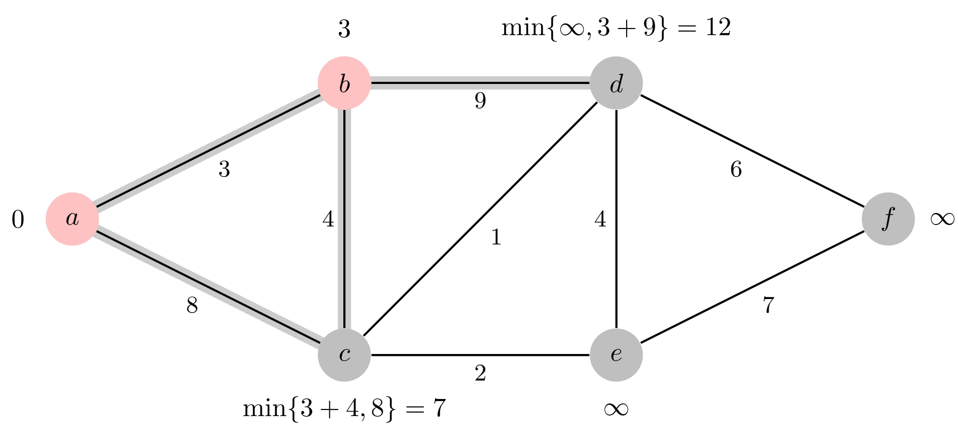


Figure C.4: Vertex b is the current vertex from which the tentative labels of its neighbors are updated.

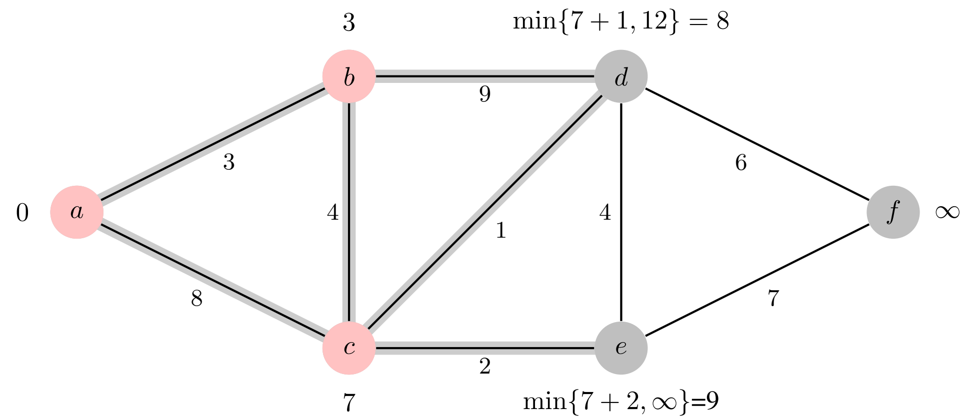


Figure C.5: Vertex c is the current vertex from which tentative labels of its neighbors are updated.

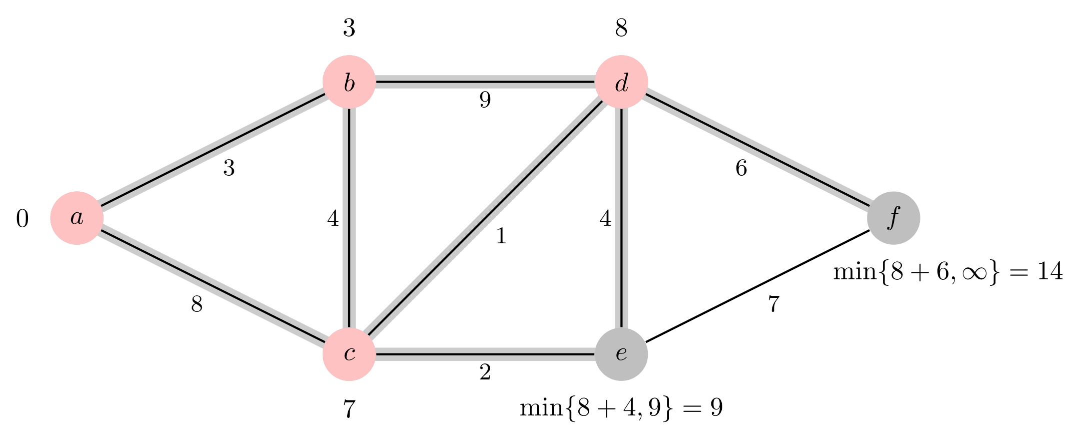


Figure C.6: Vertex d is the current vertex from which tentative labels of its neighbors are updated.

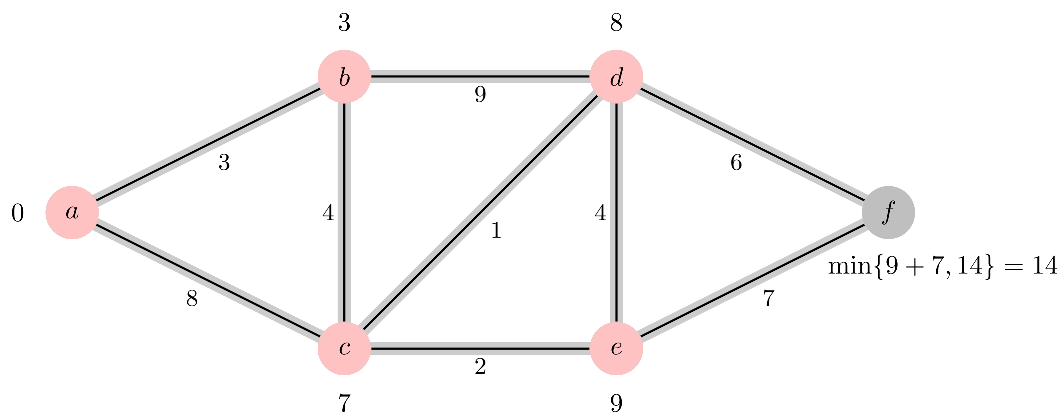


Figure C.7: Vertex e is the current vertex from which tentative labels of its neighbors are updated. The next current vertex is f, which has no neighbors with tentative labels. The label of vertex f is made permanent and the algorithm halts.

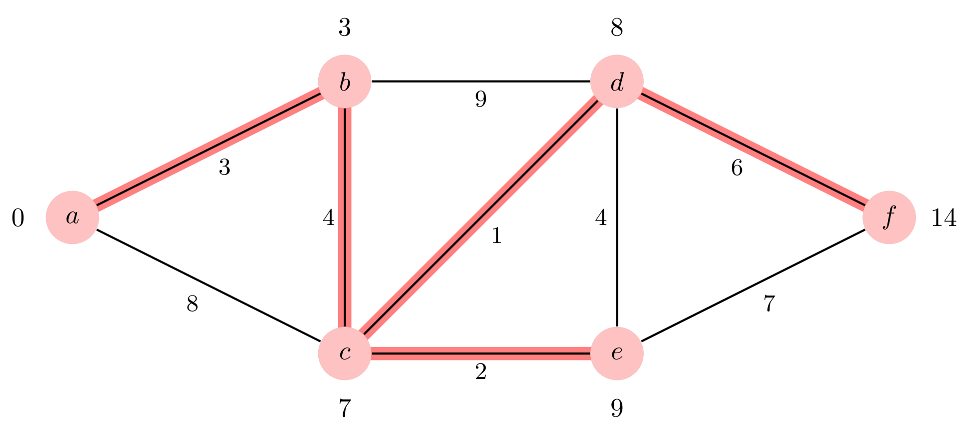


Figure C.8: The tree of shortest paths from a is identified by asking each vertex to mark that edge that was responsible for its permanent label.

Index