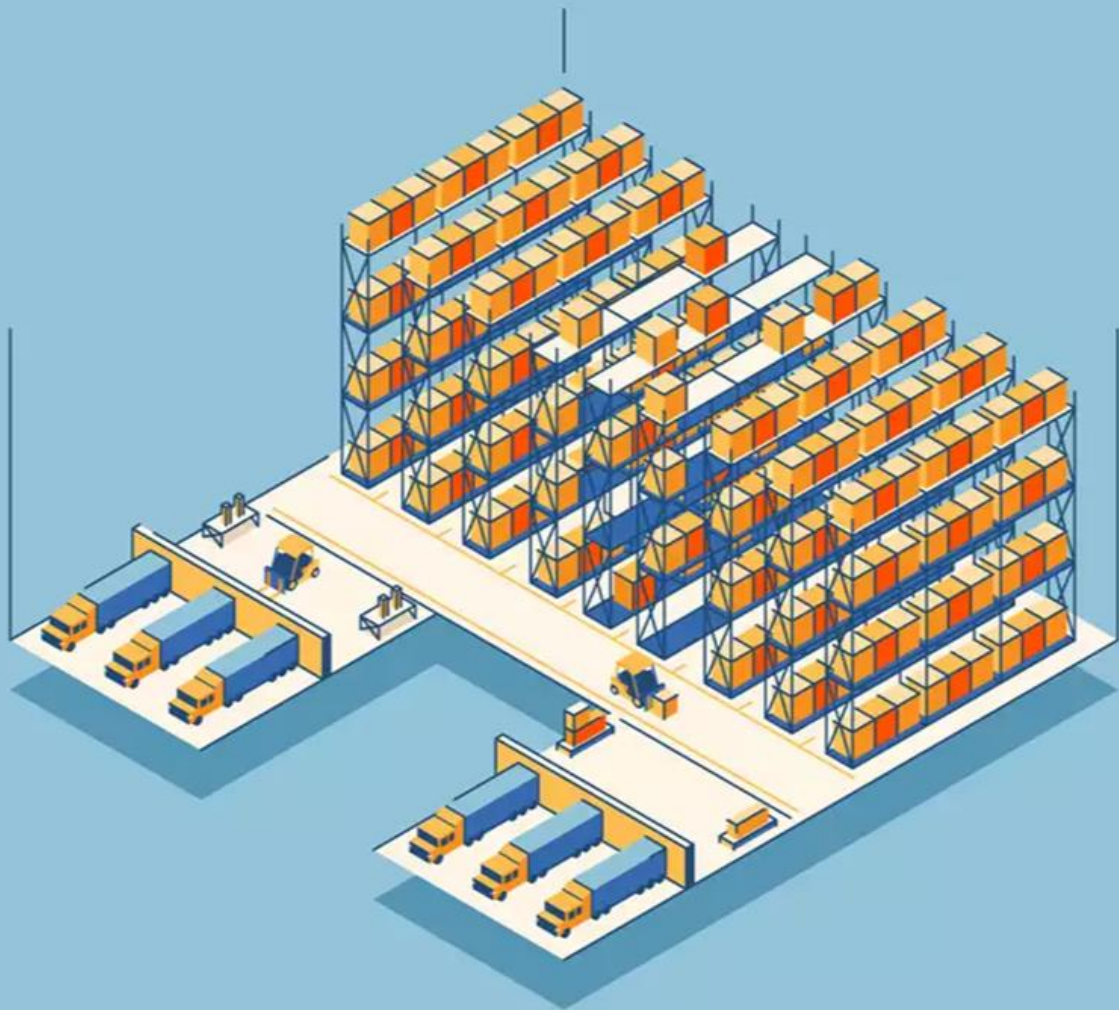


mathematics

Optimisation of distribution centres



by

Duursma, Tjeerd and Meershoek, Stijn

Supervisor - J. Jonges MSc

Optimisation of Distribution Centres

How can a distribution centre be efficiently organised?

S. (Stijn) Meershoek & T.M. (Tjeerd) Duursma

Mathematics

Alfrink College Zoetermeer

under supervision of: J. (Jochem) Jonges, MSc

13/10/2023

Index

<u>Preface</u>	6
<u>Glossary</u>	7
<u>Definitions</u>	9
<u>Chapter 1: Introduction</u>	
1.1 Introduction	10
1.2 Hypothesis	11
1.3 Plan of action	11
<u>Chapter 2: Materials and method</u>	12
<u>Chapter 3: The number of products that should be stored</u>	
3.1 Reorder point	13
3.2 Safety stock	14
3.3 Average demand	16
3.4 Standard deviation	17
3.5 Order quantity	18
3.6 Conclusion	20
<u>Chapter 4: The layout of a distribution centre</u>	
4.1 Capacity per pallet	21
4.2 Storage slots per product	22
4.3 Dimensions distribution centre	24
4.3.1 Pallet placement	24
4.3.2 Mean travel time	25
4.3.3 Length and width	27
4.3.4 Levels	30
4.4 Conclusion	34

<u>Chapter 5: The program</u>	
5.1 Variables	35
5.2 Pallet position values	36
5.3 Assigning pallets to a product	37
5.4 Conclusion	37
<u>Chapter 6: Conclusion</u>	38
<u>Chapter 7 Epilogue</u>	39
<u>Chapter 8: Summary</u>	40
<u>References</u>	41
<u>Appendix 1: Interview with prof.dr.ir René (M.) B.M. de Koster</u>	43
<u>Appendix 2: Source code of the program</u>	51

Preface

'Choosing mathematics for your PWS? Why would you do that?' is one of the things we heard when we first talked about the subject of our PWS. We were initially sceptical about this choice as well. However, as we went through the process, we became increasingly enthusiastic. We could not have done it alone and we want to thank those who helped us through the journey. First, we would like to extend our thanks to Mr. Jonges, our PWS supervisor. He helped us pick a suitable topic, offered useful advice on tackling mathematical challenges, and on how to structure our work. We would also like to express our gratitude to prof. De Koster, who generously agreed to do an interview with us. His explanations clarified aspects of our research that we initially got stuck on, and he shared additional information that added depth to our project. Lastly, we want to thank our families and friends for their support during our project. And for unintentionally helping us with their comments. We hope that our PWS not only educates fellow students about the intricacies of organizing distribution centres but also inspires them to explore mathematics in their own PWS. Mathematics is so much more than just numbers and equations.

Glossary

Abbreviation	Meaning
a	Ratio between length (L) and width (B) (LB-ratio)
Am	Main aisle width (2 times the turning circle (T))
B	Total width of distribution centre
b	Width of one bay
C	Total number of storage slots per product
c	Total area of storage slots
d	ratio between V_x and V_y (forklift speed-ratio)
D	Annual demand
Ep	Extra pallets (main aisle)
H	Total height of distribution centre
Hu	Holding costs per unit
HC	Holding costs
L	Total length of distribution centre
l	Length of one bay
LT	Lead time
n	Number of floors
Ou	Ordering costs per unit
OC	Ordering costs
P	Total number of storage slots
Q	Order quantity
ROP	Reorder point
ss	Safety stock

Glossary

T	Turning circle of the forklift
t	Average travel time to an arbitrary point in the distribution centre
TC	Total costs
V_x	Horizontal speed of the forklift
V_y	Vertical speed of the forklift
Z_a	Safety factor for the service level
β	Slack factor
μ_d	Average demand per day
σ_d	Standard deviation per day
σ_{dlt}	Standard deviation over lead time

Definitions

Bay	Two pallets stored opposite of each other with an aisle in between for the forklift to drive and turn
Distribution centre	A place between producers and costumers, where various products are stored and distributed to shops or costumers.
Order quantity	The number of pallets that are ordered in one order.
Rack	One side of a row from floor to ceiling.
Reorder point	The point at which a certain amount of stock is left, and the distribution centre should order new stock.
Row	One aisle perpendicular to the main aisle
Safety stock	The extra products that are necessary to account for the change in demand. These extra products never leave the distribution centre if the demand is equal to the average demand
Service level	The percentage of times the distribution centre has enough stock to fulfil the demand during the lead time.
Slack factor	Gives the number of pallets which can be shared between different products, because at a certain point some products use up their maximal storage space while other products use up very little storage space.
Storage slot	A space for one pallet within the warehouse with the dimensions 80 by 120 by 220cm.
Unit load	One fully stocked pallet.

Chapter 1 Introduction

1.1 introduction

‘How is a distribution centre organized?’ ‘How can you make sure it is as efficient as possible?’ These were the questions that popped into our minds when we encountered this subject for the first time in an article from the book *Succesformules* (Mols & Smeets, 2014). After reading this short article we got plenty of ideas and after a little while we came up with the idea to create a program that could design the most efficient layout for a distribution centre.

However, what is a distribution centre and why are they so important? Warehouses go back to the beginning of human civilization, where people already stored their items in some form of warehouse. Even today these warehouses exist within people’s homes and shops. In our PWS we will focus specifically on warehouses that stand between producer and consumer, the so called ‘distribution centres.’ The first distribution centre was built in 1971 by Walmart, which changed the public view on warehousing. By this time forklifts and the first AS/RS equipment were already invented which sped up the process of retrieving products from the distribution centre (Staff & GreyOrange, 2023). Then barcodes were invented which made it easier to identify the inventory, the place where it is stored and the number that can be sold, (Hopstack, 2023). Making it even more efficient.

In the 1980s distribution centres began using computers and IT programs which helped with the organization of inventories. Robotics can be found inside distribution centres starting from the year 2000 when the Autonomous Mobile Robot (AMR) and the Automated Guided Vehicle (AGV) were invented. Nowadays robots cannot be ignored within distribution centres. The usage of robotic arms, mobile racks, bot sorter arms, roaming shuttle arms and collaborative bots increases the productivity of employees within the distribution centre (Staff & GreyOrange, 2023). Yet only 20% of the distribution centres have some sort of automatization inside their distribution centre (canary7, 2023). This shows that most distribution centres can become more efficient, which also shows the importance of our subject.

The question we asked ourselves is: *How can mathematics be used to organise a distribution centre efficiently?* This is the question at the heart of our research. To answer it we will use the following two sub-questions:

- How many products should be stored?
- How can we make the layout as efficient as possible?

These questions will provide us with the information needed to make a program which designs the most efficient layout for a distribution centre. We will also interview prof.dr.ir René (M.) B.M. de Koster, who is a professor in logistics and operations management at the Department of Technology and Operations Management, Rotterdam School of Management, Erasmus University (RSM).

1.2 Hypothesis

We expect that the mathematical formulas behind warehouses are complicated with many intricate parts. However, we think it is possible to come up with a comprehensive strategy for efficiently organising a distribution centre by looking all the different aspects in organising a distribution centre. This includes looking at the number of items that need to be stored, the different types of products and where in the distribution centre each product should be stored.

1.3 Plan of action.

- First, we will investigate some literature by prof. De Koster and other sources. The most important part of this investigation is understanding the mathematical formulas and getting an understanding of the kind of mathematics we will encounter during our research, and we will conduct research into the context around our subject.
- Second, we will divide the sub-questions into their various aspects and investigate those aspects. However, the added information from the later interview, will help guide us in our answers to the sub-questions. We will also start programming.
- Third, we will prepare the interview with prof. De Koster, the interview will take place at the beginning of September. With this information, we can finish our sub-questions and finish the program.
- Finally, we will be finishing off our project with all the other important parts, such as preface, epilogue, etc. Then we will prepare for our presentation in November.

Chapter 2 Materials and method

During this chapter we will explain how we got to where we are now. It began when we found this subject in *Succesformules* (Mols & Smeets, 2014). We then wrote an email to prof. De Koster, on advice from Mr. Jonges, in which we asked if we could have an interview with him. He wrote back and told us we could have an interview after the summer break, and he generously attached the paper: Warehouse Math (Koster & Van Nunen, 2010). At this moment, our main investigation process started, trying to get an understanding of the equations in that paper. As well as using various sources to clarify what was in the paper.

Then the writing could start. We divided the sub-questions and chapters, Tjeerd wrote chapter 3 and Stijn wrote chapter 4. This was our guideline, however we helped each other with parts of the other's chapter. During the writing we explored other areas deeper based on the advice from Mr. Jonges and refined the text to make it as clear as possible. While we were writing the chapters Stijn also started programming the program that can design the optimal layout for a distribution centre.

Getting back to the email we sent before the vacation to prof. De Koster, we had some further email contact and planned the interview on the 8th of September. We asked questions about the slack factor, his thinking process behind the formula: $L = 2B$ and the average demand. He also sent us another paper, this one specifically about the slack factor. The interview took 30 minutes, and we recorded the interview, after asking for permission, so that we could rewatch it. This gave us an even a closer understanding of these aspects. We included the answers and insights in the chapters and the whole interview can be read in appendix 1.

At this point our draft version had to be handed in. And after the feedback of Mr. Jonges, we started clarifying and refining our chapters as well as writing the other necessary aspects. The program got some extra features, which brings us to the point where we are now.

Chapter 3: The number of products that should be stored

In a distribution centre, products go in and out every day which results in a variable amount of stock. This creates a problem. A distribution centre should not go out of stock, because when a consumer cannot buy a product, the consumer will look elsewhere for the same product, which results in less revenue for the business. However, a distribution centre should also not store too much stock, because that costs a lot of money, uses lots of space and can also create dead stock, which is stock that does not sell. Therefore, we need to ensure we do not store too little stock, but not too much either. How can we do this? This chapter will provide an answer to that question.

3.1 Reorder point

The reorder point (or level) is the point at which the distribution centre should order new stock, it is expressed in the amount of stock that is left in the distribution centre. When the demand is 100% predictable, we can calculate the reorder point using the following formula (Koster & Van Nunen, 2010).

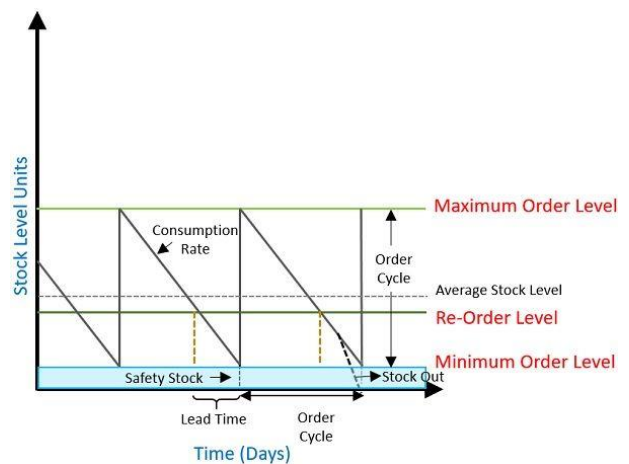


Figure 1 order cycle (S, 2021)

$$ROP = LT \cdot \mu_d$$

- ROP represents the reorder point.
- LT represents the lead time, the time it takes for the order to arrive.
- μ_d represents the average demand per day.

Example:

When we look at a package of cornflakes, with an average demand per day of 50 packages and a lead time of 10 days. This means that the new order of cornflakes arrives 10 days after the reorder point is reached.

$$ROP = 10 \cdot 50 = 500$$

3.2 Safety stock

However, demand is always changing. The demand could, and most likely will, vary from the previous day, being either higher or lower. The number of products sold (demand) during the lead time is normally distributed, with $L \cdot \mu$ being the average of the normal distribution.

Because of this changing demand, safety stock is needed to account for the possibility that the demand exceeds the average. The following formula takes that into account (Koster & Van-Nunen, 2010).

$$ROP = LT \cdot \mu_d + ss \quad (1)$$

- ss represents the safety stock.

To calculate the safety stock, we need to take two factors into consideration. The standard deviation of the demand over lead time and the service level. The service level is the percentage of times the distribution centre has enough stock to fulfil the demand during the lead time. First, we calculate the standard deviation over lead time using this formula:

$$\sigma_{dlt} = \sigma_d \cdot \sqrt{LT}$$

- σ_{dlt} represents the standard deviation of the demand over lead time.
- σ_d represents the standard deviation per day, which is a product-dependent number.
- LT represents the lead time, the time it costs for the order to arrive.

Theory square root N-law: To get to the formula above, we can rewrite

$\sqrt{\sigma_1^2 + \sigma_2^2 \dots + \sigma_n^2}$ like this: First we factor out n and we get $\sqrt{n \cdot \sigma^2}$. Then we simplify and we get $\sqrt{n} \cdot \sigma$. This follows from the rule: $\sigma_{x+y} = \sqrt{\sigma_x^2 + \sigma_y^2}$. This was also done with the average, there it was just $LT \cdot \mu_d$.

The other factor that we need to take into consideration is the service level. To determine the service level we need a safety factor, this is Z_a . When the safety factor is 0, the service level is 50% and therefore there is a 50% chance of not having enough stock. It is economically impossible to have a service level of 100%, which means that the distribution centre is never out of stock, see figure 1 to see the full list of values (Eclipse, 2023). Most businesses have a service level of 95% which means that 5% of the time there is not enough stock (King, 2011). This results in a safety factor of 1.64. This results in the following formula:

$$ss = Z_a \cdot \sigma_d \cdot \sqrt{LT} \tag{2}$$

- Z_a represents for the safety factor for a service level.
- σ_d represents the standard deviation per day.
- LT represents the lead time, the time it costs for the order to arrive.

Service Level	Service Factor	Service Level	Service Factor
50.00%	00.00	90.00%	01.28
55.00%	00.13	91.00%	01.34
60.00%	00.25	92.00%	01.41
65.00%	00.39	93.00%	01.48
70.00%	00.52	94.00%	01.55
75.00%	00.67	95.00%	01.64
80.00%	00.84	96.00%	01.75
81.00%	00.88	97.00%	01.88
82.00%	00.92	98.00%	02.05
83.00%	00.95	99.00%	02.33
84.00%	00.99	99.50%	02.58
85.00%	01.04	99.60%	02.65
86.00%	01.08	99.70%	02.75
87.00%	01.13	99.80%	02.88
88.00%	01.17	99.90%	03.09
89.00%	01.23	99.99%	03.72

Figure 1 (Eclipse, 2023)

3.3 Average demand

To enhance the accuracy of μ in formula (1), we need to have a more precise average demand and standard deviation per day. The average demand can be made more precise using forecasting. Forecasting is predicting the future, this means that it is always a guess and is therefore not 100% accurate. The forecasting method that should be used is the time-series model. However, this method is very complex. Prof. De Koster mentioned this in the interview, shown in appendix 1, and therefore we will look at a simplified version.

First, we need to clarify that we will look at weeks and not days, because it is not possible to compare a Saturday with a Thursday (see figure 3). People buy more products in the weekend than during the week. However, weeks can be compared, because weeks repeat and therefore the difference in shopping habits per day do not matter (This is excluding seasonal differences).

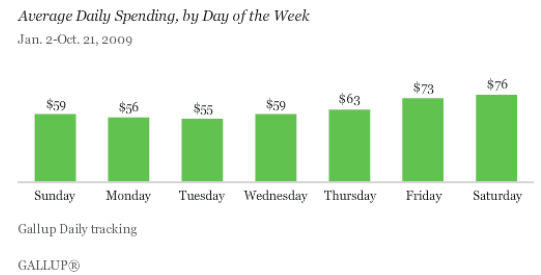


Figure 2 Daily spendings (Jones, 2021).

The average demand per week taken over a full year is the standard average. The reason that we use a year and not half a year is because products can be seasonal. Some products have higher sales in the summer than in the winter. To account for these changes during the year, we look at the week before. The standard average can be renewed every week. Shifting everything by one week, removing the oldest, and adding the newest. This will keep up with the changing demand. This is the moving average time series model (Ronald Moy, Ph.D., CFA, CFP, 2022).

$$\begin{aligned}
 \mu_{std} &= \frac{1}{52} \cdot (\mu_1 + \mu_2 + \mu_3 + \dots + \mu_{52}) \\
 d_{w-1} &= \mu_{w-1} - \mu_{std} \\
 \mu_w &= \mu_{std} + q \cdot d_{w-1}
 \end{aligned} \tag{3}$$

- d_w represents the deviation from the standard average per week.
- μ_w represents the average per week.
- μ_{std} represents the standard average per week over a year.
- q represents a coefficient specific to a product.

This coefficient q is different number for each product. It depends on the standard deviation, a high standard deviation means a high coefficient. A low standard deviation means a low coefficient. This coefficient is always between 1 and 0.

3.4 Standard deviation

The standard deviation per day can be calculated with formula (4), by using the average of formula (3) from (Getal & Ruimte, 2016).

$$\sigma_w = \sqrt{\frac{1}{52} \sum_{w=1}^{52} (d_w)^2} \quad (4)$$

- σ_w represents the standard deviation per week.
- d_w represents the deviation from the standard average per week.

Example:

We will look at packages of cornflakes, the standard average is 50 products sold per week and last week 53 products were sold, then d_w will be:

$$d_w = 53 - 50 = 3$$

If this is done every week during the year it will give results as shown in the table. d_w shows the outcome of a particular week and n the number of times that that difference occurs in a year, thus n counts to 52.

d_w	0	1	2	3	4	5	6	7	8	9	10	11
n	9	9	6	4	5	4	3	3	3	2	2	2

Now we can calculate the standard deviation per week.

$$\sigma_w = \sqrt{\frac{0^2 \cdot 9 + 1^2 \cdot 9 + 2^2 \cdot 6 + 3^2 \cdot 4 + 4^2 \cdot 5 + \dots + 11^2 \cdot 2}{52}} = 5$$

To get the average deviation per day and the average per day, we use the square root N-law.

$$\sigma_d = \frac{\sigma_w}{\sqrt{7}}$$

$$\mu = \frac{\mu_w}{7}$$

When put all these formulas and calculations together, we get formula (5).

$$ROP = LT \cdot \mu_d + Z_a \cdot \sigma_d \cdot \sqrt{LT} \quad (5)$$

- ROP represents the reorder point.
- LT represents the lead time, the time it costs for the order to arrive.
- μ_d represents the average demand per day.
- Z_a represents the safety factor for a service level.
- σ_d represents the standard deviation per day.

Example:

We look at a package of cornflakes, with an average demand per day of 50 packages and a lead time of 10 days. The safety factor is 1.65, which results in a service level of 95%. For the standard deviation per day, we will use 5.

$$ROP = 10 \cdot 50 + 1.65 \cdot 5 \cdot \sqrt{10} \approx 526$$

3.5 Order quantity

We now know when we need to order new stock. But, how big should the order be? Using the costs, we can calculate the optimal number of products we should order. From now on we will look at unit loads instead of individual products. One unit load is a fully loaded pallet. A unit load has ordering costs and holding costs, combining these costs results in the total costs. The total costs are the lowest when the holding and ordering costs are equal, proof that they should be equal follows on the next page. The following two formulas are for the ordering and holding costs (Koster, 2010).

$$OC = \frac{D}{Q} \cdot O_u$$

- OC represents ordering costs.
- D represents the annual demand in unit loads.
- Q represents the order quantity in unit load.
- O_u represents the order costs per order.

$$HC = \frac{Q}{2} \cdot H_u$$

- HC represents holding costs.
- H_u represents the unit holding costs per year.
- Q represents the order quantity in unit load.

Combining these two formulas results in formula (6) for the total costs.

$$TC(Q) = \frac{D}{Q} \cdot O_u + \frac{Q}{2} \cdot H_u \quad (6)$$

Setting the derivative equal to 0 to get the lowest point of the graph.

$$TC(Q) = DO_u Q^{-1} + \frac{1}{2} H_u Q$$

$$TC'(Q) = -DO_u Q^{-2} + \frac{1}{2} H_u$$

$$TC'(Q) = -\frac{DO_u}{Q^2} + \frac{1}{2} H_u$$

$$\frac{DO_u}{Q^2} - \frac{H_u}{2} = 0 \quad (\text{This proves that } OC = HC)$$

$$\frac{DO_u}{Q^2} = \frac{H_u}{2}$$

$$H_u Q^2 = 2DO_u$$

$$Q^2 = \frac{2DO_u}{H_u}$$

$$Q = \sqrt{\frac{2DO_u}{H_u}}$$

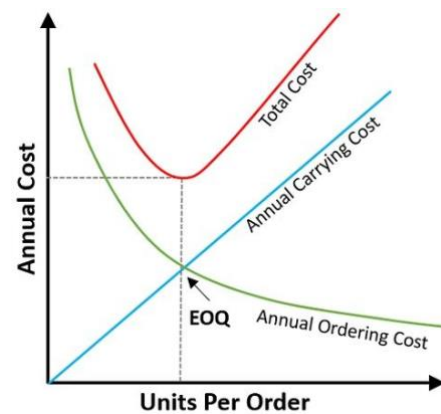


Figure 3 (S, 2021)

The order quantity is therefore calculated using formula (7)

$$Q = \sqrt{\frac{2DO_u}{H_u}} \quad (7)$$

- Q represents the order quantity in unit load.
- D represents the annual demand in unit loads.
- O_u represents the order costs per unit load.
- H_u represents the unit holding costs per year.

From formula (7) we can derive that the average maximal amount of stock is $ss_{in\ unit\ load} + Q$ because ss is the average stock that is left after the order arrives. This also means that the average stock is $ss_{in\ unit\ load} + \frac{Q}{2}$.

Example:

We look at a package of cornflakes, with an annual demand of $50 \cdot 365 = 18250$, the formula is in unit loads so $\frac{18250}{320} = 57,03$. This should be rounded to the nearest integer because we cannot have half a pallet. The number 320 is calculated in the next chapter, it is the products per pallet. So, the annual demand is 58. The transport costs per unit load is €40 and holding costs per year is €185.60.

$$Q = \sqrt{\frac{2 \cdot 58 \cdot 40}{185.60}} = 5$$

3.6 Conclusion

The question we asked in the beginning of this chapter is: how can we make sure we store as little stock as possible while never going out of stock? By using safety stock, we minimise the probability of going out of stock. The formula for order quantity gives the most efficient number of unit loads that should be bought, which ensures that there is not too much stock. Finally, the ROP gives the point where both the order quantity and the safety stock are optimal.

Chapter 4 Layout of the distribution centre

After figuring out how we can calculate the number of products that should be stored for the optimal service level and the lowest costs, we need to figure out the actual layout of the distribution centre. To determine the optimal layout, many factors need to be taken into consideration: the dimensions and weight of each product, how many pieces of that product can be stored per pallet, and the number of storage slots we need. Furthermore, it is also important to know in which orientation to store the pallets, wide or deep. Then we need to decide how many levels is optimal in the distribution centre, which is limited by the reach of the forklift being used. Besides that, we should consider using a single input/output point or split it into two points on opposite sides. We also need to think about optimal product placement to be able to take the fastest route through the distribution centre for restocking and retrieving a product for an order.

In the calculations in this chapter, we assume the pallets are a standard euro pallet (EPAL) with dimensions 80 x 120 x 14.4cm, a maximal combined height of 220cm and a weight limit of 1500kg. And we will solely use products with a cubic shape, or that are stored within a cubic shape.

4.1 Capacity per pallet

To calculate how many pallets will fit on one pallet we first calculate the optimal orientation of the product because when you rotate a product that is not a perfect square or cube it changes the number of products that fit on one pallet. We do this by calculating every one of the six, 3!, possible orientations and continue with our calculations using the orientation that results in the most products per pallet. However, if that number of products exceeds the weight limit, we lower the number until it fits within the weight limit. This information provides the fundament of the organisation of a distribution centre.

Example:

We will look at a package of cornflakes with dimensions 20 by 10 by 30cm weighing 720g.

*For one orientation: width: $\frac{80}{20} = 4$ length: $\frac{120}{30} = 4$ height: $\frac{220-14.4}{10} \approx 20.57 = 20$
 $4 \cdot 4 \cdot 20 = 320$ pieces can be stored per pallet*

When you switch the denominators 3! times, you find that 320 is the highest and therefore the optimal number. And $320 \cdot 0,720\text{kg} < 1500\text{kg}$ this means that we are within the weight limit, and we do not need to lower the number of products.

4.2 Storage slots per product

Next, we need to calculate how many storage slots we need per product. For this we need the results from the last chapter and the number of pieces per pallet. We can do this with the following formula (Koster & Van Nunen, 2010).

$$C = \left\lceil \left(\frac{Q}{2} + ss_{in\ unit\ load} \right) \cdot (1 + \beta) \right\rceil \quad (1)$$

- C represents the number of storage slots per product.
- Q represents the order quantity in unit load.
- $ss_{in\ unit\ load}$ can be calculated by ss divided by products per pallet.
- β is the slack factor.

The number of storage slots needs to be rounded up to the nearest integer, because even if there are a few pieces left they need a whole pallet to be stored.

The slack factor gives the number of pallets which can be shared between different products, because at a certain point some products use up their maximal storage space while other products use up very little storage space. When not using a slack factor, all products need their maximal number of storage slots, which is $Q + ss_{in\ unit\ load}$.

Example:

We look at a package of cornflakes, where the order quantity is in unit load is 5. The safety factor is $\frac{1.65 \cdot 5 \cdot \sqrt{10}}{320} = 0,082$. For the slack factor we will use 40%.

$$\left\lceil \left(\frac{5}{2} + 0,082 \right) \cdot (1 + 0,40) \right\rceil = \lceil 3,61 \rceil = 4 \text{ storage slots}$$

When we do not use the slack factor but instead use the maximal storage space needed per product we get. $\lceil 5 + 0,082 \rceil = \lceil 5,082 \rceil = 6$, this shows that 2 storage slots are saved.

The exact number for the slack factor is still a discussion point, this can be seen in appendix 1 where prof. De Koster states that he has not found the exact number even after he wrote a paper about it. However according to the paper, it falls between 30% and 40% (Yu et al., 2015). The paper also explains that the slack factor depends on the number of classes the storage system has. These classes are groups of products with similar turnover rates. The pareto principle states that 20% of the products sold produce 80% of the revenue. By using classes, we can account for this discrepancy.

But how many classes would be ideal? For instance, in a distribution centre where twenty products are stored. If the products are not divided into classes we have twenty classes, one for each product. This means that every individual product needs space for their maximal amount of stock because they cannot share space or storage slots with other products. The advantage of this is that the twenty products can all be assigned individual storage slots throughout the distribution centre following the average demand of the product. Products with a high demand will be closer to the input/output point and products with a low demand will be further away from the input/output point. If these products were divided into two classes instead of twenty, they would be able to share space between them and therefore less space is needed overall. However, when we assign storage slots for the classes, some products with a higher annual demand will be further away from input/output point. This leads to longer travel time. The ideal number of classes will fall between 3 and 8 classes (Yu et al., 2015).

4.3 Dimensions distribution centre

Then we need to determine the optimal width and height of the distribution centre based on the number of pallets that need to be stored? To calculate this, we first need to know how the pallets are placed, then we need to look at the mean travel time and lastly, we need to look at how this would work in practice.

4.3.1 Pallet placement

Now that we know how many pallets should be stored, we face a new question: which way should the pallets be stored, deep or wide? This question arises because one of the things that needs to be optimized is the amount of space that is used. For this we will need to calculate the area of one bay, where a bay is defined as two pallets stored opposite of each other with an aisle in between for the forklift to drive and turn (see figure 5).

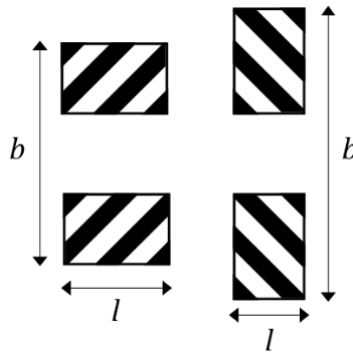


Figure 4. A bay is two pallets stored at opposite sides of the aisle, stored wide (left) or deep (right)

In the calculations we assume the specific forklift used is a pallet stacker capable of lifting about 4m high. We also need the turning circle, this is the diameter of the shortest possible turn the forklift can make. To calculate the dimensions of the bay, we use these formulas:

$$\begin{aligned} b &= T + 2(b_{pallet} + 0.2) \\ l &= l_{pallet} + 0.2 \end{aligned} \quad (2)$$

- b represents the width of one bay.
- l represents the length of one bay.
- T represents the turning circle of the forklift.
- 0.2 represents the safety space.

Example:

The turning circle of our forklift in this example is 2.1m when it is carrying a pallet wide.

If we store the pallets deep the turning circle increases by 0.4m because the pallet is turned. The area of the bay is calculated the following way:

Wide: $b = 2.1 + 2 \cdot (0.8 + 0.2) = 4.1m$ and $l = 1.2 + 0.2 = 1.4m \rightarrow 5.74m^2$

Deep: $b = 2.5 + 2 \cdot (1.2 + 0.2) = 5.3m$ and $l = 0.8 + 0.2 = 1.0m \rightarrow 5.3m^2$

Storing the pallets wide results in an area of $5.74m^2$. And storing the pallets deep results in an area of $5.3m^2$. This very clearly shows that pallets should be stored deep and not wide in a distribution centre.

4.3.2 Mean travel time

We also need to take in to account the mean travel time. To do this we will look at the mean travel time to a random place in the distribution centre. The mean travel time width wise is $\frac{1}{2}B$. This is because direction B only includes the main aisle, from where we can access all the individual rows. The mean travel time length wise is determined in the same way. When we enter a row from the main aisle, the mean travel time inside that row is half the length of that row. And because there are rows on both sides of the main aisle the mean travel time is $\frac{1}{4}L$. This gives us the one-way travel time: $\frac{1}{2}B + \frac{1}{4}L$. But we also need to go back so we need to double this. Giving us the formula below for the total mean travel time to any random place in the distribution centre. However, doing it in this way ignores the width of the main aisle and is therefore an approximation. This explanation has come from the interview with prof. De Koster shown in appendix 1.

$$t = B + \frac{L}{2} \quad (3)$$

- t represents the mean travel time.
- L represents the length of the distribution centre.
- B represents the width of the distribution centre.

There is also a choice of whether to split the input/output (I/O) point into an input and an output point, or not, because this affects the travel time. When the I/O point is split the optimal configuration would be that they are on opposite sides of the distribution centre. But we can conclude that a single I/O point is more optimal. Because if we take products out for an order and right after that we need to restock products. Separate points add a travel time of B since we need to drive from the output to the input point on the other side, whereas using a single I/O point there is no added travel time. Although there might be other factors influencing this decision, in most cases travel time is the most important.

Using formula (3) we can calculate the optimal length and width of a distribution centre. We can say that $L \cdot B \geq c$. Where c represents the total area of storage slots needed. The distribution will be the smallest when $L \cdot B = c$. When we take this formula and formula (3) together we get:

$$L \cdot B = c \rightarrow B = \frac{c}{L}$$

$$\begin{cases} B = \frac{c}{L} \\ t = B + \frac{L}{2} \end{cases} \rightarrow t = \frac{c}{L} + \frac{L}{2}$$

Calculating the derivative equal to zero

$$t' = -c \cdot L^{-2} + \frac{1}{2}$$

$$-c \cdot L^{-2} + \frac{1}{2} = 0 \rightarrow -c \cdot + \frac{1}{2}L^2 = 0 \rightarrow c = \frac{1}{2}L^2$$

$$\begin{cases} c = \frac{1}{2}L^2 \\ B = \frac{c}{L} \end{cases} \rightarrow B = \frac{\frac{1}{2}L^2}{L} \rightarrow B = \frac{1}{2}L \rightarrow L = 2B$$

This concludes into $L = 2B$, meaning that a distribution centre should optimally be twice as long as it is wide when the area of the distribution centre does not include the aisles. To calculate the ratio between L and B when we do include the aisles. We divide the ratio by the percentage of space that is used for storage. For the total length (L) this is the percentage of used space: $\frac{l_{pallet}}{l}$ and for the total width ($2B$) we get this as the percentage used: $\frac{2b_{pallet}}{b}$. By dividing the total L and B by the percentage of space that is used, we get the ratio between L and B that includes the aisles. This results in the following equation: $\frac{L}{\frac{l_{pallet}}{l}} = \frac{2B}{\frac{2b_{pallet}}{b}}$.

This equation is simplified on the next page.

$$\frac{L}{\frac{l_{pallet}}{l}} = \frac{2B}{\frac{2b_{pallet}}{b}} \rightarrow L = \frac{2l_{pallet}}{2lb_{pallet}}B \rightarrow L = \frac{2bl_{pallet}}{2lb_{pallet}}B \rightarrow L = \frac{bl_{pallet}}{lb_{pallet}}B$$

We will look at this formula as $L = aB$ with $a = \frac{bl_{pallet}}{lb_{pallet}}$. We say that a is the LB-ratio.

Example:

When we fill in the numbers from the last example when the turning circle of the forklift was 2.1, $b = 5.3$ and $l = 1.0$.

$$a = \frac{5.3 \cdot 0.8}{1.0 \cdot 1.2} = 3.533 \dots = 3.5$$

This results in the following ratio:

$$L = 3.5B$$

4.3.3 Real length and width

We first need to consider that a distribution centre has extra pallets on the end of the main aisle. The number of pallets on the main aisle can be calculated by:

$$E_p = \left\lfloor \frac{A_m}{l} \right\rfloor n$$

- E_p represents the number of pallets on the main aisle.
- A_m represents the width of the main aisle ($2 \cdot T$).
- l represents the length of one bay.
- n represents the number of levels.

To calculate the real width and length of the distribution centre, we need to consider the number of rows, the number of bays per row and the width of each bay. To calculate these dimensions, we need to know how the number of pallets we can store relates to the width and length of the distribution centre. We can neatly describe that relation using formula (4) where the number of pallets to store equals: $\frac{B}{b}$ (the number of rows) multiplied by $\frac{L}{l}$ (the number of bays per row) for the total number of bays. Then we multiply that by two because each bay contains two pallets. And finally multiply that by the number of levels in our distribution centre, which gives us the total number of pallets we can store.

$$M = \frac{L}{l}$$

$$N = \frac{B}{b}$$

$$P = 2nMN \quad (4)$$

- P represents the number of pallets to be stored (E_p excluded).
- n represents the number of levels in the distribution centre.
- M represents the total number of bays that fit inside the length (bays per row).
- N represents the number of rows.

Combining the formula above with $L = aB$ we can then reformulate it into formula (5) and (6) for the real width and for the real length of the distribution centre like this:

$$\begin{cases} P \approx 2n \frac{B L}{b l} \rightarrow P = 2n \frac{B a B}{b l} \\ L = aB \end{cases}$$

$$\frac{P}{2n} = \frac{aB^2}{bl}$$

$$2aB^2n = Plb$$

$$B^2 = \frac{Plb}{2an}$$

$$B = \sqrt{\frac{Plb}{2an}}$$

$$B = \sqrt{\frac{Plb}{2an}} \quad (5)$$

$$L = \sqrt{\frac{Plba}{2n}} \quad (6)$$

- B represents the optimal width of the distribution centre in *meters*.
- L represents the optimal length of the distribution centre in *meters*.

These two formulas will provide the width and length in *meters*. To use those values, we calculate the number of rows by dividing $B_{optimal}$ by the width of each bay (calculating N) and then use the ceil function. We do it this way because these formulas will give us approximate answers and we cannot have fractional amounts of rows or bays. However, doing it this way means we do not need formula (6) for $L_{optimal}$ since it is determined by the number of bays per row, which we can find by using formula (7):

$$M = \left\lceil \frac{P}{2nN} \right\rceil \quad (7)$$

- M represents the number of bays per row.
- N represents the number of rows.
- P represents the number of pallets to be stored (E_p excluded).

And if we apply all these formulas, we get this result as a top-down view of the first level of the distribution centre:

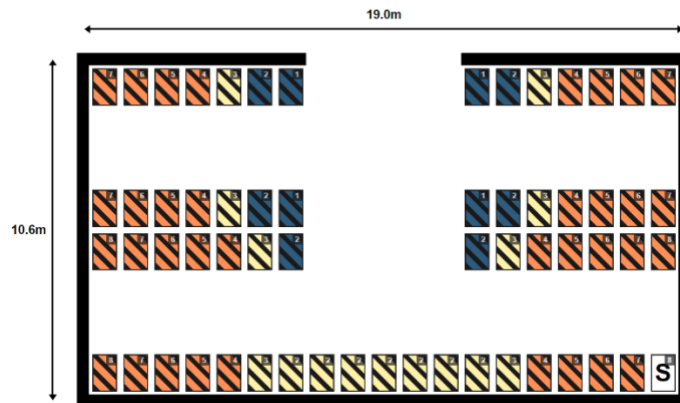


Figure 5. Top-down view of the ground floor.

4.3.4 Levels

To generate the most optimal distribution centre we also need to calculate the most optimal height. With the optimal height we can calculate the number of levels. As a reminder, we assume the pallets are a standard euro pallet (EPAL) with dimensions 80 x 120 x 14.4cm, a maximal combined height of 220cm and a safety space of 20cm.

To calculate the number of levels we need to divide the total height (H) by the height of a pallet, which is the maximum height plus safety space this is $2.4m$. We also need to consider that the pallet itself has a height, in this case 14.4cm. To lift a pallet we need to be able to reach at least the pallet height, this is why we subtract it from the total height. Finally we round it up because if you can reach the bottom of the level that is enough for a full level since we only need to reach the bottom of the pallet.

$$n = \left\lceil \frac{H - 0,144}{2,4} \right\rceil \quad (8)$$

- H represents the optimal height of the distribution centre in *meters*.
- n represents the number of levels in the distribution centre.
- 2,4 is the level height, the maximal height plus the safety space ($2,2 + 0,2$).
- 0,144 is the pallet height.

To calculate the number of levels we use formula (8), We divide the optimal height by the level height. We need to consider that there is a minimum of one level and therefore we use the ceil function. However, the forklift needs to lift an additional pallet height (0,144) to retrieve the pallet from its storage slot.

Example:

When we would take an optimal height of 5.1 meters, the formula would give:

$$n = \left\lceil \frac{5.1 - 0,144}{2,4} \right\rceil = \lceil 2,065 \rceil = 3$$

So, with an optimal height of 5.1 meters there would be 3 levels within the distribution centre.

The next thing we need to consider is how the height and length of the warehouse are connected. This can be done by looking at the shape of the pallet rack. A rack is one side of a row from floor to ceiling. It can be optimised when looking at the speed of the forklift. A forklift has a horizontal speed, a vertical speed, and can lift and drive simultaneously. Therefore, the most efficient shape of a rack is a square in time not necessarily in dimensions. This means that the farthest storage slot in the rack should have an equal horizontal and vertical time to reach it. However, the horizontal speed is often higher than the vertical speed which results in a rectangular shape. The ratio between the height and length will determine the rectangular shape of the rack, the speed ratio.

$$d = \frac{V_x}{V_y}$$

- d represents the speed-ratio.
- V_x represents the horizontal speed of the forklift.
- V_y represents the vertical speed of the forklift.

With the speed-ratio we can compare the length and the height. In the formula (9) we use half of the length because from the main aisle left or right is only half of the length and therefore the shape of the rack depends only on half of the length.

$$dH = \frac{1}{2}L \tag{9}$$

- H represents the height of the distribution centre in *meters*.
- d represents the speed ratio.
- L represents the length of the distribution centre.

When we substitute formula (9) in formula (8) we get formula (10):

$$n = \left\lceil \frac{\frac{L}{2d} - 0.144}{2.4} \right\rceil \tag{10}$$

Now the number of levels can be calculated using the length of the distribution centre instead of the optimal height which we can't calculate directly. When we substitute formula (10) in formula (6) we get formula (11), this finally enables us to calculate to optimal length.

$$\begin{cases} L = \sqrt{\frac{Pbla}{2n}} \rightarrow L^2 = \frac{Pbla}{2n} \rightarrow 2nL^2 = Pbla \\ n = \left\lceil \frac{\frac{L}{2d} - 0.144}{2.4} \right\rceil \end{cases} \rightarrow \frac{\frac{L}{2d} - 0.144}{2.4} * 2L^2 = Pbla \rightarrow$$

$$\rightarrow \frac{2L^3}{2d} - 0.288L^2 = 2.4Pblad \rightarrow$$

$$L^3 - 0.288dL^2 - 2.4Pblad = 0 \quad (11)$$

- L represents the length of the distribution centre.
- d represents the speed-ratio.
- P represents the number of pallets to be stored (E_p excluded).
- b represents the width of one bay.
- l represents the length of one bay.
- a represents the LB-ratio.

We are going to solve the cubic function by identifying discriminants using the cubic formula (Partheepan & Sivalingam, 2020). When we look at a standard form of a cubic function $x^3 + ax^2 + bx + c = 0$. We can say that in formula (11) $a = -0.288d$; $b = 0$; $c = -2.4Pblad$.

Then we will find p , where $p = b - \frac{a^2}{3}$; and we will find the q , where $q = \frac{2a^2}{27} - \frac{ab}{3} + c$.

With p and q , we can calculate the discriminant (Δ), where $(\Delta) = \frac{q^2}{4} + \frac{p^3}{27}$.

Our discriminant is always positive, so we will always have only one solution. Which can be

found by using this formula: $L = \left(\frac{-q}{2} + \sqrt{\Delta}\right)^{\frac{1}{3}} + \left(\frac{-q}{2} - \sqrt{\Delta}\right)^{\frac{1}{3}} - \frac{a}{3}$. This results in the following formula:

$$L = \left(\frac{-\left(\frac{2(-0.288d)^2}{27} - 2.4Pblad\right)}{2} + \sqrt{\frac{\left(\frac{2(-0.288d)^2}{27} - 2.4Pblad\right)^2}{4} + \frac{\left(\frac{-(0.228d)^2}{3}\right)^3}{27}} \right)^{\frac{1}{3}} +$$

$$\left(\frac{-\left(\frac{2(-0.288d)^2}{27} - 2.4Pblad\right)}{2} - \sqrt{\frac{\left(\frac{2(-0.288d)^2}{27} - 2.4Pblad\right)^2}{4} + \frac{\left(\frac{-(0.228d)^2}{3}\right)^3}{27}} \right)^{\frac{1}{3}} + \frac{0.288d}{3} \quad (12)$$

The length (L) that we get from formula (12) can be filled in formula (10). This will give the number levels that would be optimal for the distribution centre. The n can be used in the formulas (5) and (7) to calculate the real width and length. We calculate the number of levels first because that number has the biggest impact on the layout of the distribution centre and the length has the least impact and is therefore calculated last.

Example:

We will not write the whole calculation down due to the length of the calculation.

1) We will use the same numbers as the last few examples: $b = 5.3; l = 1.0; a = 3.533$.

2) We will also use a speed-ratio of 3, and 70 pallets that needs to be stored: $d = 3; P = 70 - 5n$ ($5n$ are the extra pallets, these are excluded)

3) When we fill in these numbers in formula (12), we get $L = 20,366 \dots$

4) This number can be filled in formula (10), we get $n = 1.354 \dots = 2$

5) This number can be used in formula (5), we get $B = 4.744 \dots$

6) divide this number with b , ceiling function, gives $N: N = \left\lceil \frac{4.744 \dots}{5.3} \right\rceil = 1$

7) use formula (7) to get the $M: M = \left\lceil \frac{60}{2 \cdot 2 \cdot 1} \right\rceil = 15$

*8) calculate H, B and L from n, N and $M: H = 1 * 2.4 = 2.4m; B = 1 * 5.3 = 5.3m;$*

*$L = 15 * 1.0 = 15m$*

9) the length must take the width of the main aisle, 5m, into account therefore $15 + 5$ which is 20m.

9) conclusion, the dimensions of the most efficient distribution centre are: $5.3 \times 20 \times 2.4$



Figure 6 The warehouse from the dimensions calculated in the example. 1st and 2nd level.

4.4 Conclusion

To conclude, using the formulas in this chapter combined with some of the information in Chapter 3 we can now determine the optimal values for the number of pallets to store, which uses the slack factor to get a minimum number. The slack factor is determined by the number of classes that a distribution centre has. As well we found out that pallets need to be stored deep, because that is more efficient. We also saw that a single input/output point is the most optimal. The optimal number of pallets to store leads to the optimal values for the width, length, and height of the distribution centre, which is optimal at $L = aB$ and $\frac{1}{2}L = dH$. However, these ratios are almost impossible to reach. By solving a cubic formula, we were able to calculate the optimal width, length, and height.

Chapter 5: The program

in the final step of our PWS we will, to help the reader understand everything that is said so far, we bring these formulas to life. To do that we created a program that designs the most efficient layout of a distribution centre given the necessary variables. This also connects to the purpose of this PWS, warehousing is becoming increasingly important, and it is crucial to optimise it to save space, money, and time. This program can be found on the website <https://meershhoek.com/warehouse/>. the source code, of the parts of the program discussed here, can be found in Appendix 2.

5.1 Variables

To generate this layout we start at the very beginning, we need to receive the necessary data. This includes the following variables for each individual product you want to store:

- Dimensions
- Weight
- Annual demand
- Standard deviation per day
- Lead time
- Transport costs
- Holding costs

And these variables for the forklift specifications:

- Speed (horizontal & vertical)
- Reach
- Turning circle

When the user submits these variables the program has enough information to calculate everything needed to display the optimal layout. The program then calculates everything according to the formulas mentioned in chapter 3 and 4, however the program does not know which pallet corresponds to which product. And to put products with higher demand closer to the I/O point the travel time to each individual pallet. This also gives us the average travel time overall in the warehouse, which is interesting to see how it is affected by all the variables.

5.2 Pallet position values

We need to take into consideration that a computer has no idea of visuals and space and the pallets in the warehouse are contained in just a simple list. The steps below are a description of what the program computes, the actual code can be viewed in Appendix 2, generateLayout.tsx, lines 45-79

1. The program splits the list of the total pallets into separate lists for each row in the warehouse.
2. Then it divides the number of total bays in each row by two, then round one half up and the other half down. This gives us the number of pallets to the left and right of the I/O point in a row.
 - a. The program names these variables: left and right.
3. To check if the pallets are left of the main aisle it checks if the index of the pallet in the list is strictly less than $2 \cdot left$.
 - a. If true, the position (travel time) of the pallet is:
$$left - \left\lfloor \frac{palletIndex}{2} \right\rfloor + [rowIndex \cdot aisleWidth]$$
 - b. If false, the position (travel time) of the pallet is:
$$\left\lfloor \frac{palletIndex}{2} \right\rfloor + 1 - |left - right| - right + [rowIndex \cdot aisleWidth]$$
4. If the pallet is one of the extra pallets that is neither left or right of the I/O point, then this is the position of those pallets:
 - a. $[totalRows \cdot aisleWidth]$
5. This only takes care of the first level, since the vertical speed is less than the horizontal speed, we need to correct for that in the position.
 - a. Check if pos is strictly less than $\left\lfloor \frac{speedRatio \cdot levelIndex \cdot (maxPalletHeight + safetySpace)}{aisleLength} \right\rfloor$
 - i. If true, do nothing the position is correct.
 - ii. If false, set the position to the calculation in 5.a.

When you execute these steps correctly you now have a separate list of pallets for each row, that all have a position assigned to them.

5.3 Assigning pallets to a product

This can then be used to assign pallets to a product so that the products with the highest annual demand are stored on pallets with the lowest position value to make sure the fastest moving products are as close as possible to the I/O point.

This is done by combining the separate lists for each row into a single list and sorting the pallets from lowest distance to highest distance, then start at the beginning of the list and start with the product with the highest demand and assign the number of pallets that that product needs. Then you remove those pallets from the list because they are now in use. Then proceed to the next product and repeat the process until every product is assigned its respective pallets. This code can be viewed in Appendix 2, Level.tsx,

5.4 Conclusion

In conclusion, this chapter has detailed the practical implementation of our warehouse optimization approach. We introduced a program that leverages the formulas discussed in earlier chapters to design an efficient layout for a distribution centre. The program, accessible at <https://meershoek.com/warehouse/>, considering things like product size, weight, and demand to design the best layout for a distribution centre.

The process involves getting user-inputted variables, after which the program calculates the optimal layout by considering factors such as travel time and pallet positions. The complicated steps for determining pallet positions, as shown in section 5.2, ensure that products with higher demand are placed closer to the I/O point increasing efficiency.

Furthermore, section 5.3 describes the step of assigning products to pallets based on their demand and position values. By combining separate lists for each row and sorting pallets by distance, the program efficiently allocates pallets to products, optimizing the overall layout.

This program may not be a perfect tool for a business (yet). But it is more of an example to show that with math, you can address the growing importance of warehousing optimization, aiming to save space, money, and time. This underscores the relevance and applicability of our program in the evolving landscape of warehouse management.

Chapter 6: Conclusion

The research question we asked ourselves was: *How can mathematics be used to organise a distribution centre efficiently?*

In our hypothesis we stated that there are many different aspects in organizing a distribution centre. We mentioned the number of products, details about the products and where they should be placed throughout the distribution centre. Now we can say that this hypothesis is correct, however these things turned out to be much more complex than we first thought, and we left out many other variables.

We can now say that the first thing that is needed for organising a distribution centre is the number of products which leads to the amount of storage slots needed. Then there is the safety stock, which minimises the chance of going out of stock. And the order quantity, which tells us what the most efficient number of units loads that should be bought is and prevent too much stock. Furthermore, the reorder point is needed, to make sure that the safety stock and order quantity work together. These numbers are found using the time-series forecasting method. And finally, the storage slots can be minimised by using the slack factor. By using this slack factor, the products are sharing storage space resulting in less space used overall.

To use space within the distribution centre efficiently we found out that pallets need to be stored deep. We then move on to the optimal number of storage slots, and at the travel time to each pallet which needs to be optimised to make it as efficient as possible. This can provide the width and length of the distribution centre. We also saw that a distribution centre needs to have a single input/output point. The number of levels depends on many things including the speed ratio of the forklift.

By using all this information and using the formulas in this paper we were able to create the most efficiently organised distribution centre. However, there are still questions that arise. Such as, how are forecasting methods used in a more precise way? What is the most efficient number for the slack factor? How does innovation change the most optimal distribution centre? Unfortunately we were not able to answer these questions, but we hope that we can do so in the future.

Chapter 7: epilogue

We are proud of what we have achieved with this PWS. The strongest parts of our PWS are the program and the interview with prof. De Koster. While the part about average demand is not our favourite, because it is not complete. It barely scratches the surface, there is so much more to find out about that subject.

This PWS was not without hick-ups. First, we had a tough time understanding the mathematical equations. We have spent hours, together and alone, trying to understand the different formulas. Which resulted in the next issue: writing it down in a way that other people can understand it as well as we can. It got to the point where we felt we needed to provide examples besides our text. We tackled it by spending a few hours together rereading what we had written and rewrite the parts that were unclear. The programming also had some hick-ups. The main hick-up was calculating the travel time and then defining which product went where and how to label them with colours. By trying different ways, we finally got something that worked.

The things we learned during this PWS is that we are stronger together, parts that looked impossible were tackled together, but also how many things mathematics has to offer. When we started, we did not think that it would end here.

Chapter 8: Summary

In chapter 3 we discussed the challenges faced by distribution centres in determining the optimal number of products to store without going out of stock. We introduced the concept of the reorder point (ROP), safety stock (ss) and order quantity (Q) to ensure stock while minimizing excess stock. We also explored a method for improving the precision of average demand estimation, following a time-series forecasting model. Below are the most important formulas.

For the reorder point:

$$ROP = L \cdot \mu_d + ss$$

For the safety stock:

$$ss = Z_a \cdot \sigma_d \cdot \sqrt{L}$$

For the order quantity in unit loads:

$$Q = \sqrt{\frac{2DO_u}{H_u}}$$

In chapter 4 we delved into the layout of distribution centres, emphasizing the importance of optimizing space utilization and efficient product placement. We discussed factors such as the capacity per pallet, number of storage slots per product (c), pallet placement, the dimensions of a distribution centre and we explored the choice of a single input/output point versus multiple points. Below are the most important formulas.

For storage slots per product:

$$c = \left[\left(\frac{Q}{2} + ss_{in\ unit\ load} \right) \cdot (1 + \beta) \right]$$

For the optimal width and length:

$$L = \frac{bl_{pallet}}{lb_{pallet}} B$$

For the number of levels in the warehouse:

$$n = \left\lceil \frac{\frac{L}{2d} - 0.144}{2.4} \right\rceil$$

References

- Canary7. (2023, March 15). Important warehouse automation Statistics - Meteor space. Meteor Space - Warehousing & Order Fulfillment Services in Ireland & Europe: Meteor Space. <https://www.meteorspace.com/important-warehouse-automation-statistics/#:~:text=Currently%2C%20more%20than%2080%25%20of,started%20to%20catch%20on%20recently.>
- Eclipse. (2023, 2 oktober). Safety Stock Formula: How to calculate and use. Linnworks. <https://www.linnworks.com/blog/safety-stock-formula/>
- Getal&Ruimte: vwo D deel 3 (11th ed.). (2016). Noordhoff Uitgevers Groningen.
- Hopstack, T. (2023). Evolution of Warehousing Systems: history and timelines. Hopstack Inc. <https://www.hopstack.io/blog/evolution-warehousing-systems-history-timelines>
- Jenkins, A. (2021). What is dead stock? why it's bad & how to avoid. *Oracle NetSuite*. <https://www.netsuite.com/portal/resource/articles/inventory-management/dead-stock.shtml>
- Jones, B. D. J. a. J. M. (2021, August 23). Consumers spend more on weekends, payday weeks. Gallup.com. <https://news.gallup.com/poll/123839/consumers-spend-more-weekends-payday-weeks.aspx>
- King, P. L. (2011). Crack the code: Understanding safety stock and mastering its equations. *APICS Magazine*.
- Koster, R. B. M., & Van Nunen, J. (2010). Warehouse Math. L. Kroon, T. Li, R. Zuidwijk (eds.), Liber amicorum.
- Koster, R. B. M. (2010, February 11). WISKUNDE WISKUNDE IN DE INTERNE LOGISTIEK. Retrieved July 5, 2023, from http://www.fi.uu.nl/nwd/nwd2010/handouts/Koster_NWD_2010.pdf

Laoyan, S. (2022, December 8). Het Pareto-principe begrijpen (de 80/20-regel) [2022] • Asana. Asana. <https://asana.com/nl/resources/pareto-principle-80-20-rule>

Mols, B., & Smeets, I. (2014). *Succesformules*. Platform Wiskunde Nederland.

Partheepan, S., & Sivalingam, D. (2020). SOLUTION TO CUBIC EQUATION USING JAVA PROGRAMMING. Progressive Academic Publishing, 7(1).
<https://www.idpublications.org/wp-content/uploads/2020/01/Full-Paper-SOLUTION-TO-CUBIC-EQUATION-USING-JAVA-PROGRAMMING.pdf>

Ronald Moy, Ph.D., CFA, CFP. (2022, July 8). Moving Average (MA) Time Series model [Video]. YouTube. <https://www.youtube.com/watch?v=qaAVqh1ZMok>

S, S. (2021, August 23). Difference Between Re-Order Level and Re-Order Quantity (with Formula, Example, Graph and Comparison Chart) - Key Differences. Key Differences. <https://keydifferences.com/difference-between-re-order-level-and-re-order-quantity.html>

Staff, G., & GreyOrange. (2023). The evolution of warehouse automation. GreyOrange. <https://www.greystone.com/blog-posts/warehouse-automation-101/#:~:text=Warehouse%20Automation%20in%20the%20Past&text=Commercial%20warehouses%20began%20using%20it,efficient%20process%20than%20ever%20before.>

Yu, Y., De Koster, R., & Guo, X. (2015). Class-Based Storage with a Finite Number of Items: Using More Classes is not Always Better. *Production and Operations Management*, 24(8), 1235–1247. <https://doi.org/10.1111/poms.12334>

Appendix 1: Interview with prof.dr.ir René (M.) B.M. de Koster

Interviewee: prof.dr.ir René (M.) B.M. de Koster

Interviewers: Tjeerd and Stijn

Tjeerd: Het wordt vanaf nu opgenomen.

Koster: Ja

Tjeerd: Wij zitten op het Alfrink College in 6 vwo. We doen allebei wiskunde B en D. We houden ons profielwerkstuk over het efficiënt indelen van een magazijn. Daarbij is onze hoofdvraag: hoe kan je een magazijn zo efficiënt mogelijk indelen? Vervolgens kwamen we bij u terecht via het boekje succesformules.

Koster: Ja, dat klopt ja, heb ik het zelf geschreven? Ik weet niet of ik het zelf heb geschreven. Volgens mij was het een interview.

Stijn: Ja, maar het ging om het idee dat u daar iets over dit onderwerp had geschreven waardoor wij het idee hebben kregen om ons profielwerkstuk over magazijnen te doen.

Koster: Ik heb ooit een lezing gegeven, op een congres voor docenten wiskunde van middelbare scholen en volgens mij was het daarop gebaseerd? Ik weet het niet helemaal zeker.

Tjeerd: Het document dat u naar ons heeft gestuurd, heeft ons heel erg vooruitgeholpen. U had geschreven: Het is middelbare school wiskunde, maar het was af en toe toch best pittig om het goed te kunnen begrijpen.

Stijn: Voor ons PWS hebben we een stukje over economie. Hoeveel producten je moet opslaan voor een optimaal servicelevel, dat je niet te weinig voorraad hebt, maar ook niet te veel is dat het teveel gaat kosten. Dat is het eerste gedeelte van het PWS. Daarna komt het gedeelte van de met de wiskunde met oppervlakte van de magazijnen. Hoeveel pellets je moet hebben voor alle producten en hoeveel ruimte dat dan in beslag neemt. De hoogte, breedte en lengte van het magazijn. Dat zijn de twee delen waar we ons profielwerkstuk in verdeeld hebben. Tjeerd heeft een vraag over een van de formules die over dat eerste stukje ging.

Tjeerd: Ja, in formule 1. Want u stelt dat het maximale aantal producten dat je kan hebben, gelijk is aan de safety stock plus de bestelling hoeveelheid, dus $ss + Q$. Het lijkt mij dat dat de reorder point plus de bestelling hoeveelheid zou moeten zijn. Waarom geldt dan het eerste?

Koster: Nog een keer, wat denk jij dat het maximaal is?

Tjeerd: De reorder point is natuurlijk het punt waarop je nieuwe producten bestelt. Als je dan niks meer verkoopt, dan heb je dus de producten die binnenkomen en je hebt de reorder point, dus mij leek dat de reorder point plus de bestelling hoeveelheid maximaal is.

Koster: Ja het punt is die safety stock. Dat is de dat zeg maar de hoeveelheden die je overhoudt. Die je verwacht over te houden. Tussen het moment van bestellen wat gebeurt op het moment dat er reorder point onderschreden wordt en op het moment dat het binnenkomt. Dus die safety stock is in feite de hoeveelheid die aanwezig is op het moment dat die hoeveelheid binnenkomt. Maar het is niet het moment dat je bestelt op het reorder point. Tot moment dat het binnenkomt, verstrijkt er wat tijd en de verwachte hoeveelheid die je overhoudt op het moment dat het binnenkomt. Dat is die safety stock, vandaar die formule.

Tjeerd: Ja, dat had ik ook gedacht, alleen ik dacht dat de safety stock het gemiddelde is wat je over houdt, want het is volgens mij zo'n 95% kans dat je niet out of stock raakt en dus onder je safety stock komt. Maar dan is er ook een kans dat je dus boven je safety stock blijft.

Koster: Ja, dat klopt. Die safety stock is de verwachte hoeveelheid die je overhoudt. Dus met andere woorden, is dit nummer precies? Nee, Dat is niet helemaal precies. Het gemiddelde maximum.

Tjeerd: Oh ja, Als je het zo zegt, gemiddeld maximum.

Stijn: Dat is inderdaad een stukje logischer, dat het gemiddelde maximum is en niet het echte totale maximum.

Koster: ja het kan afwijken daarvan. Op het moment dat je minder verkoopt gedurende die levertijd van het product zal de hoeveelheid die je overhoudt op het moment dat er binnenkomt groter kunnen zijn dan de safety stock. Dus dit is allemaal probabilistisch.

Tjeerd: Oké, ja dat dan is dat duidelijk.

Stijn: Ik had ook nog een vraag over het gedeelte wat daarna komt dus over het magazijn zelf. Dat is ik zal even kijken waar dat precies is. Dat is aan het begin van 2.3 die formules die daar staan en dan de formules voor de gemiddelde reis.

Koster: B plus een half L.

Stijn: ja

Koster: Het is gebaseerd op die figuur 3 en daar verwaarloos ik in feite de breedte van de midden gang. De L is netto wat kleiner, want een gedeelte van de L wordt gebruikt als midden gang, en wordt dus niet helemaal gebruikt voor opslag. Ik heb die breedte van die midden gang verwaarloosd ten opzichte van de totale lengte van het magazijn. Ik doe net alsof die nul is.

Stijn: Ik heb het stukje daarvoor ook gelezen en dan staat er dus dat eigenlijk het enige wat invloed heeft op de lengte is de tijd, omdat heftruck niet tegelijkertijd omlaag omhooggaat, maar alleen rijdt.

Koster: Je kunt zeggen, dat de lengte niet is uitgedrukt in meters, maar uitgedrukt is in seconde rijtijd. Maar dat je bepaalde gemiddelde horizontale snelheid hebt van de heftruck.

Stijn: Hoe komt u bij dat die formule de gemiddelde tijd is, want er staat in de tekst staat het als een feit, maar er staat niet bij uitgelegd waarom het zo is.

Koster: Als ik even mijn scherm deel, dan kan ik misschien met mijn muis aanwijzen. Even kijken hoe ik dat moet doen.

Oké, dus hier staat de gemiddelde reistijd. B plus een half L . B en L zijn dan uitgedrukt in travel tijd. Dus uitgedrukt in secondes en niet in meters. Dat kan je doen omdat die truck alleen maar horizontaal rijdt en die rijdt met een bepaalde constante gemiddelde snelheid. Dat is de aanname dus verwaarloos ook nog een keer de versnelling en vertraging. Als je natuurlijk wegrijdt, dan moet je versnellen. Het duurt één seconde voor je op topsnelheid zit en als je gaat stoppen duurt het ook één seconde voordat je stilstaat. Die heb ik verwaarloosd. Wat is nu de gemiddelde reistijd? De veronderstelling is dat je naar een random locatie rijdt, dus een locatie die willekeurig getrokken wordt op het niveau. Dus we gooien alle niveaus plat op een laag, daarbij verwaarloos ik al de verticale verplaatsing. Dat kan ik doen omdat die verticale verplaatsing sequentieel volgt op de horizontale verplaatsing, dus ik optimaliseer alleen het oppervlak en daarna kijk ik pas naar het aantal lagen dat ik zou moeten hebben. Dus stel dat ik kom van locatie X en ik moet naar locatie Y . De locatie is willekeurig getrokken. Dan is de totale verplaatsingstijd een half B . Gemiddeld moet ik half de diepte in rijden, dan moet ik gemiddeld een kwart L gang in rijden. Ik heb de breedte van het midden gang verwaarloosd. Dus dit is L , maar ik rij hem maar half zo gang in, want ik heb gangen aan twee kanten, dus het is een kwart L . Maar dan moet ik ook weer terug, want het is een volledige cyclus. Ik sla iets in of ik haal iets op, als ik iets insla kom ik leeg terug en als ik iets ophaal, ga ik een leeg naartoe. Uiteindelijk maak ik dezelfde cyclus, dus ik ga dan weer terug een kwart L plus een half B , bij elkaar B plus een half L .

Stijn: Oké, dat is dat klinkt inderdaad heel logisch.

Koster: Hier is de layout wat anders (2 verschillende inslag/uitslag punt), maar dan heb ik een vergelijkbaar resultaat. Het enige verschil is dat je soms van inslag naar uitslag moet. Daarom moet je daar soms extra verplaatsen.

Stijn: Dat was eigenlijk ook precies mijn volgende vraag, want u heeft hier natuurlijk twee verschillende layouts getekend. Is er een voordeel bij één van deze twee of maakt het niet uit?

Koster: Nou ja, je snapt wel Als je af en toe moet verplaatsen van uitslag naar inslag, brengt dat extra verplaatsingstijd met zich mee. De vraag is, hoe vaak moet je dat doen?

Stijn: Dus welke van deze twee zou dan beter zijn?

Koster: In het algemeen is één inslag uitslag beter, want daar kun je ook makkelijker een dubbelspel rijden. Een dubbelspel betekent dat ik iets wegbrengen, dan moet ik naar een andere gang rijden, een willekeurige andere gang, waar ik het volgende product ophaal. Je snap wel dat als ik een dubbelspel doe, dat ik dan voordeel heb van de eerste layout ten opzichte van de tweede layout. Ik mis alle verplaatsingstijd tussen de inslag en de uitslag die ik bij de tweede nodig heb.

Stijn: Ik zag ook dat in de eerste met een enkel input en output. Dat op de achterste rij ook meer opslag is, dus dat je ook meer producten in het magazijn kan opslaan bij de eerste layout.

Koster: Dat is niet noodzakelijk zo. Ik heb die arcering aangebracht om aan te geven waar je snelle lopers zou moeten neerleggen als er één inslag uitslag punt is, en Waar zou je de snelle lopers neer moeten leggen als je een verschillend inslag en uitslag punt hebt. Bij de tweede zou je ze meer rond de koppen van de gangen moeten verspreiden.

Ik ben nu uitgegaan naar deze formule van willekeurige opslag, dus elke locatie is even waarschijnlijk dat je daar naartoe moet. Maar in de praktijk zul je zien dat dat bedrijven altijd proberen om snellopers wat dichterbij de buurt van het depot te leggen. Op het moment dat je twee depots hebt, wordt het ook weer wat moeilijker waar je die snellopers neer moet leggen en worden deze ook meer verspreid, zoals hier bij het rechter plaatje. Dus dat is ook alweer een nadeel van het hebben van twee depots.

Stijn: Oké, doe je dat dus dat verheldert een hele hoop. Dat waren eigenlijk de vragen die ik daarover had, Dank u wel, dat helpt.

Tjeerd: Nu heb ik nog een vraag over de vorige formule, de formule met de opslag per product. Daar komt een slack factor bij, de β . Er staat erboven aangegeven dat die ongeveer 20% zou moeten zijn.

Koster: 20%? Dat weet ik niet meer hoor.

Stijn: Ja, het staat een stukje boven bij de definities die aangeven wat alle letters betekenen. Dus aan het begin van 2.1 staat een lijstje met wat al die symbolen betekenen.

Koster: Ja, dat is Tomkins. We hebben zelf daar een ander paper over geschreven, daar hebben we gekeken van wat is hier nou? Want die hangt van een aantal dingen af. Waarvan bijvoorbeeld van afhangt, is de omloopsnelheid van de producten. De ABC krommen heet het ook wel. Dus de Pareto vraag kromme.

Dan ga ik er wat tekenen even kijken dat lukt. Ik heb hier twee assen. Op de verticale as zet ik het percentage van de vraag neer en op de horizontale as het percentage van de producten die in opslag zijn. Allebei hebben ze een maximum van 100% en ik doe dat cumulatief. Als ik dat niet cumulatief zou doen dan staat hier gewoon het product per producten. Het is niet als percentage, maar gewoon de producten, is gesorteerd op aflopende omloopsnelheid, dan krijg je zo'n kromme. Er zijn een paar producten die

hardlopen en een groot aantal producten die minder hardlopen. Als ik dat cumulatief teken, dan krijg ik een Pareto kromme.

Dus uiteindelijk wordt die 100%. Pareto heeft ooit beweerd, dat 20% van de producten, 80% van de aanvraag vertegenwoordigen. Dat is in de praktijk natuurlijk lang niet altijd zo, maar het is wel zo dat er vaak een kleiner percentage producten toch een relatief grote proportie van de vraag vertegenwoordigd. Daar hangt die bèta onder andere van af. De bèta hangt af van de steilheid van die kromme.

Ik heb er al een paper over hoe je die bèta redelijk kunt bepalen, maar dat is redelijk technisch paper.

Stijn: Ja, daar wordt redelijk ingewikkeld om allemaal in een PWS te stoppen, denk ik.

Koster: Dat hoeft je er niet in te stoppen, maar ik zal hem er toch even sturen want dan heb je wat achtergrondinformatie. Het is eigenlijk een onopgelost probleem, want ik heb het alleen maar approximatief opgelost. Zoals allemaal onopgeloste dingen zijn, is dit een van de open problemen, maar ik leg het zo meteen wel uit.

Stijn: Maar dus niemand heeft het opgelost?

Koster: Ik heb het ook niet opgelost. Ik heb het alleen maar een beetje opgelost.

Hoeveel ruimte heb je nodig bij een bepaald aantal opslag klassen? Het vervelende is dat de ruimte van formule C geeft aan hoeveel ruimte je nodig hebt voor de verschillende producten, Maar die ruimte hangt ook af van hoeveel ladingen je hebt van elk product. Je kunt je voorstellen als ik maar één pallet heb van het product. Dan is het makkelijk uit te rekenen. Wat gebeurt er nou als die Q groter is dan één? Dus ik heb meer pallets van een product als de optimale bestelhoeveelheid. Stel nu dat de optimale bestelhoeveelheid 2,5 pallet is. Dan moet ik natuurlijk naar boven afronden. Je kan geen halve pallet opslaan. Die neemt gewoon een hele plek in beslag, dus voor het aantal plekken moet ik naar boven afronden. En, de vraag is dan. Hoe hangt het aantal totaal benodigde plekkjes in mijn magazijn pellet plaatsen af van die bestel hoeveelheden. Dat hebben we in dit paper proberen uit te rekenen en dat hangt af van hoeveel opslag klassen je ze verdeelt. Als je elk product zijn eigen klasse geeft, dus voor elk product reserveer je ruimte voor. Heb je veel meer ruimte nodig dan wanneer je zegt: Ik ga ruimte reserveren voor alle top 100 snellopers. Laten we zeggen dat ik van elk van die producten de bestelhoeveelheden heb. Als je bestelhoeveelheid 3 is kan je je voorstellen dat je dan 300 plekken nodig hebt. Als je elk product zijn eigen plekje geeft, maar als je je ruimte deelt heb je minder pellets nodig. Ik heb niet altijd tegelijkertijd die 3 pallets van elk product staan. Van sommige producten heb ik één pallet staan, van een ander product heb ik net besteld, dan komen de 3 binnen, dus je kunt je voorstellen dat je de ruimte deelt, je minder ruimte nodig hebt in totaal. Dat hebben we in dit paper geprobeerd uit te rekenen. Wat is nou de verwachte maximale ruimte die je nodig hebt op het moment dat je producten deelt met anderen? Ingewikkeld hé?

Tjeerd&Stijn: Inderdaad ingewikkeld.

Koster: Nou, dan komen we uit op ongeveer 30%. Dus die bèta blijkt ongeveer 30% te zijn. Dat is ondergrens plus 30% niet de bovengrens plus 30%. Dat is afhankelijk van het aantal klassen en afhankelijk van de steilheid van die kromme. Hier staat het aantal klassen. Wat je eigenlijk probeert te doen, is de reistijd minimaliseren? Het zijn twee soorten magazijnen, hier is het magazijnvierkant in de tijd. Lengte en breedte is een net zo groot en hier is het magazijn niet vierkant. Dan is dat bolletje hier, het open bolletje het optimale aantal klassen wat de kortste speltijd aangeeft.

Ik heb ook nog een plaatje met de ruimte die daarvoor nodig is. The total required storage space, hier zie je de totale extra ruimtebehoefte en dan zie je dat die ongeveer 1 is. De gemiddelde aantal locaties, dat is Q gedeeld door 2 plus safety stock en dan zie je dat je daar bij een bepaald aantal klassen, ongeveer 40% in dit geval.

Afhankelijk van de steilheid van de ABC komen die s parameter. Die geeft de steilheid van de ABC kromme aan. Hoe kleiner die s hoe steiler die krommen is; dus hoe minder producten verantwoordelijk zijn voor een groter deel van de omzet. Kleinere s , steilere kromme. Dus je ziet ook ongeveer, ik zei 30, maar het is iets meer. Het zit tussen de 30 en de 40%. Bij 3 klassen, wat het meest voorkomt, heb je A, B, C producten, snel lopers, midden lopers, langzaam lopers. Dan heb je ongeveer 40% meer ruimte nodig boven het gemiddelde. Maar nogmaals, het is allemaal approximatief, niet exact.

Stijn: Dus, die bèta zou ergens tussen de 30% en de 40% moeten zitten. Is die 20% dan op een andere manier berekend?

Koster: Uit onze berekeningen zijn we op iets meer uitgekomen, namelijk op 40%.

Tjeerd: Dan heb je dus je gemiddelde wat je in je magazijn hebt plus die 40%.

Koster: Precies.

Dat blijkt dat bij 3 opslagplaatsen. Dus als je 3 opslagplaatsen ABC. Dan heb je ongeveer die 40%. Je ziet ook dat het stijgt. Naar mate je meer opslag klassen hebt, heb je meer ruimte nodig. Op moment dat je 100 opslagplaatsen hebt. Dat betekent in feite dat je elk product zijn eigen klassen geeft en dan heb je twee keer zoveel nodig. Dat is precies de Q plus safety stock in plaats van Q gedeeld door 2 plus safety stock, zie je dat?

Tjeerd&Stijn: Ja, interessant

Koster: Daar heb je twee keer zoveel nodig per product.

Tjeerd: Ik had alleen nog een vraag hoe je de gemiddelde verkoop van je product benadert.

Koster: De gemiddelde verkoop?

Tjeerd: De verkoop die staat benoemd bij de formule $s = L \cdot \mu + Z_a \cdot \sigma d \cdot \sqrt{L}$. Mijn vraag is hoe ik het gemiddelde vraag per dag zou kunnen benaderen?

Koster: Ja, op het moment dat je nog geen verkoop hebt, wordt het natuurlijk lastig.

Tjeerd: Ja dat begrijp ik, maar als je zeg maar wel een aantal jaar verkoop zou hebben.

Koster: Ik neem aan dat je dan de gemiddelde vraag per dag weet als jij benzine verkoopt, weet je hoeveel liter benzine je per dag verkoopt. Maar je weet ook de spreiding gedurende de levertijd. Dus die sigma DL. Wat je doet is, je weet je levertijd dat je bijvoorbeeld elke week geleverd krijgt. Dan je turf je de vraag gedurende een week per dag. En dan kun je de standaarddeviatie daaruit afleiden. Dat is één gedeeld door het aantal dagen min 1, de wortel van de waarneming per dag en dat gesommeerd. Natuurlijk over alle dagen min gemiddelde en dat gekwadraterde. Dat is de formule voor steekproef spreiding en die heb je al gehad.

Tjeerd: Ja dat klopt

Koster: Dus dat is sigma DL. Dat is dan gemeten gedurende de levertijd en dan bepaal je de spreiding over die periode. De vraag verandert natuurlijk per gemiddelde.

Tjeerd: Kan je dan gewoon een gemiddelde van een jaar berekenen?

Koster: Dat hangt af. Je kijkt natuurlijk in de toekomst. In feite is het een voorspelling. De vraag is, hoe voorspel je de gemiddelde vraag per dag op basis van je historische gegevens, dan kun je een tijdreeks methode voor gebruiken. Dus een forecasting methode die je loslaat op metingen uit het verleden en die geven dan informatie over de vraag in de toekomst. Maar dat is een voorspelling, dus μ is een voorspelling van de vraag per dag.

Tjeerd: Ja precies, dus uiteindelijk moet je het inderdaad voorspellen.

Koster: Ja, in dit geval moet alles uitgedrukt worden in eenheid ladingen. En die Q wordt uitgedrukt in eenheidslading en niet in stuks van een product, maar die Q moet uitgedrukt worden in pallets.

Stijn: Ja, dat hadden we door.

Koster: Maar als je 200 stuks hebt en er 100 stuks op een pallet passen, zijn dat twee pallets.

Stijn: Ja, dus ik reken eerst alles per stuk en vervolgens zet ik alles om in pallets. Dat klinkt logisch allemaal. Ik denk dat het wel een beetje onze vragen waren.

Koster: Die vierkante haken betekenen naar boven afgerond. Het kan ook dat iemand stuks bij elkaar optelt, dus Ik heb bijvoorbeeld 300 stuks is Q. En Ik heb nog 50 stuks veiligheidsvoorraad. Dan heb ik in de eerste formule 150 plus 50 is 200. Dat past precies op één pallet, dan rond ik de Q niet af, ik rond het geheel af op hele aantal pallets,

Tjeerd: En dat doe je ook keer de slack factor

Koster: Ja, maar strikt genomen zou dat niet hoeven. In de paper wat ik je net gestuurd hebt, hebben we dat niet gedaan, maar daarna moet je dan wel afronden. Dus je zou kunnen zeggen, dat die haken eigenlijk erbuiten moeten. Eerst vermenigvuldig met β en dan pas naar boven afronden. Dat zou accurater zijn. Hoewel het helemaal niet accuraat is, want het is allemaal approximatief. De β hangt af van zoveel verschillende factoren, dus niemand weet wat die β precies is. Die β is gewoon onbekend.

Tjeerd: Ik vind het wel heel interessant allemaal.

Koster: Als je wiskunde gaat studeren dat kan je dit probleem nog eens een keer exact kan oplossen. Dat zou wel heel mooi zijn.

Tjeerd: Ik wil econometrie gaan studeren.

Koster: Nou dat het daar ook wel, de logistieke kant. Ik weet niet of je het in Rotterdam wil doen, dan kan je kiezen tussen financiële econometrie en econometrie die meer toegespitst is op logistiek. Dat komt pas bij je master.

Tjeerd: Ja, heel leuk.

Stijn: Dus dat waren eigenlijk onze vragen die wij aan u hadden, maar misschien heeft u nog vragen aan ons over ons PWS?

Koster: Ik wens jullie heel veel plezier en veel succes met het afronden van jullie werkstuk.

Stijn: Ja dank u wel ook voor uw tijd. Ik weet niet of u het heel druk heeft, maar in ieder geval dank u wel.

Koster: Succes jongens, fijne dag nog.

Appendix 2: Source code of the program

generateLayout.tsx

```
1 import { maxPalletHeight, safetySpace } from "./definitions";
2
3 const generateLayout = (
4   LBRatio: number,
5   aisleWidth: number,
6   aisleLength: number,
7   speedRatio: number,
8   extraPallets: number,
9   totalPallets: number,
10  warehouseLevel: number
11 ): Warehouse | undefined => {
12   if (warehouseLevel !== 1 && totalPallets < extraPallets *
warehouseLevel) {
13     return undefined;
14   }
15
16   // Calculate Optimal Width - Formula (5)
17   const optimalWidth = Math.sqrt(
18     ((totalPallets - extraPallets) * (aisleWidth * aisleLength)) / (2
* warehouseLevel * LBRatio)
19   );
20
21   // Calculate number of rows and number of bays per row
22   const totalRows = Math.ceil(optimalWidth / aisleWidth) || 1;
23   const totalBaysPerRow = Math.max(
24     0,
25     Math.ceil((totalPallets - extraPallets * warehouseLevel) / (2 *
totalRows * warehouseLevel))
26   );
27
28   // Calculate total storage capacity - Formula (4)
29   const storageCapacity = 2 * warehouseLevel * totalRows *
totalBaysPerRow + extraPallets * warehouseLevel;
30
31   // Calculating how many pallets are stored left and right of the
main aisle
32   const left = Math.ceil(totalBaysPerRow / 2);
33   const right = Math.floor(totalBaysPerRow / 2);
34
35   // Calculating the real warehouse dimensions
36   const warehouseWidth = totalRows * aisleWidth;
37   const warehouseLength = (totalBaysPerRow + extraPallets) *
aisleLength;
38
39   // Genereting representation of the layout in an array form
40   const layout = [...Array(warehouseLevel)].map((_, levelIndex) => {
41     return [...Array(totalRows)].map((_, rowIndex) => {
42       const isLastRow = rowIndex === totalRows - 1;
43
44       return [...Array(totalBaysPerRow * 2)].map((_, palletIndex) =>
{
```

```

45     let pos =
46         (palletIndex < left * 2
47         ? // left side
48         left - Math.floor(palletIndex / 2)
49         : // right side
50         Math.floor((isLastRow ? palletIndex : palletIndex) / 2)
+ 1 - Math.abs(left - right) - right) +
51         Math.round(rowIndex * aisleWidth);
52
53     if (levelIndex > 0) {
54         if (pos < Math.round(speedRatio * levelIndex *
(maxPalletHeight + safetySpace)) / aisleLength) {
55             pos = Math.round((speedRatio * levelIndex *
(maxPalletHeight + safetySpace)) / aisleLength);
56         }
57     }
58
59     return {
60         pos: pos,
61     };
62 });
63 });
64 });
65
66 // Generating representation of the extrapallets in an array form
67 const extra = [...Array(warehouseLevel)].map( (_, levelIndex) => {
68     return [...Array(extraPallets)].map( () => {
69         let pos = Math.round((layout[0].length - 1) * aisleWidth);
70
71         if (levelIndex > 0 && pos < Math.round(speedRatio * levelIndex
* (maxPalletHeight + safetySpace)) / aisleLength) {
72             pos = Math.round((speedRatio * levelIndex * (maxPalletHeight
+ safetySpace)) / aisleLength);
73         }
74
75         return {
76             pos: pos,
77         };
78     });
79 });
80
81 // Calculating the farthest away position number
82 const highestLevel = layout[layout.length - 1];
83 const lastRow = highestLevel[highestLevel.length - 1];
84 const highestPosArr = [...lastRow.map((el) => Number(el.pos))];
85
86 const highestPos = highestPosArr.length === 0 ? 0 :
Math.max(...highestPosArr);

```

```

87
88 // Summing up all the positions in the warehouse
89 const totalPos = layout
90   .map((l, levelIndex) => {
91     const sum = l
92       .map((row, rowIndex) => {
93         let sum = row.reduce((a, b) => {
94           return a + b.pos;
95         }, 0);
96
97         let extraSum =
98           extraPallets *
99           (levelIndex > 0 && Math.round(rowIndex * aisleWidth) <
Math.round(speedRatio * levelIndex)
100             ? Math.round(speedRatio * levelIndex)
101             : Math.round(rowIndex * aisleWidth));
102
103         return sum + extraSum;
104       })
105       .reduce((a, b) => {
106         return a + b;
107       }, 0);
108
109     return sum;
110   })
111   .reduce((a, b) => {
112     return a + b;
113   }, 0);
114
115   return {
116     average: (totalPos - (storageCapacity - totalPallets) *
highestPos) / totalPallets,
117     layout: layout,
118     extraLayout: extra,
119     optimalWidth: optimalWidth,
120     warehouseWidth: warehouseWidth,
121     warehouseLength: warehouseLength,
122     left: left,
123     right: right,
124   };
125 };
126
127 export { generateLayout };
128

```

Level.tsx

```
1 // deciding which product gets which place based on distance from the
I/O point to each pallet
2   createEffect(() => {
3     const allPallets = [
4       ...(Array.from(document.querySelectorAll(".pallet")) as unknown
as HTMLCollectionOf<HTMLElement>),
5     ]
6     .reduce((acc, cur) => {
7       (acc[cur.dataset.pos as any] as any) = [...(acc[cur.dataset.pos
as any] || []), cur];
8       return acc;
9     }, []);
10    .map((arr: any[]) => {
11      return arr.sort((a, b) => Number(b.attributes["data-
top"].value) - Number(a.attributes["data-top"].value));
12    })
13    .flat();
14
15    let palletsUsed = 0;
16
17    // Giving the pallets the colour of their assigned product
18    props.products.forEach((product, productIndex) => {
19      for (let i = 0; i < product.pallets; i++) {
20        let pallet = allPallets[palletsUsed + i];
21        pallet.style.setProperty("--colour",
` ${productColours[productIndex]} `);
22        pallet.setAttribute("data-product", product.name);
23      }
24      palletsUsed += product.pallets;
25    });
26
27    // Giving any extra left over pallets a different colour and
letter
28    allPallets.splice(0, palletsUsed);
29    allPallets.forEach((pallet) => {
30      pallet.innerText = "S";
31      pallet.style.background = "#fff";
32    });
33  });
```