Aalto University
School of Science
Master's Programme in Mathematics
and Operations Research

Ella Warras

Optimizing Shelf Space Allocation in Grocery Retail

Master's Thesis
Espoo, May 22, 2019

Supervisor: Professor Harri Ehtamo
Advisor: Tuomas Viitanen D.Sc. (Tech.)

Aalto University
School of Science
Master's Programme in Mathematics
and Operations Research

ABSTRACT OF
MASTER'S THESIS

Author:	Ella Warras		
Title:	Optimizing Shelf Space Allocation in Grocery Retail		
Date:	May 22, 2019	**Pages:**	viii + 72
Major:	Systems and Operations Research	**Code:**	SCI3055
Supervisor:	Professor Harri Ehtamo		
Advisor:	Tuomas Viitanen D.Sc. (Tech.)		

Grocery retail is a competitive industry with high sales volumes and low profit margins, which makes managing costs and optimizing processes especially important. Store and warehouse labor costs constitute a large part of the retail cost structure, and it is also an area where large savings can be obtained by optimizing different processes. Optimizing the use of shelf space can reduce the amount of time the employees have to spend bringing stock from the backroom storage to the shelf. Other benefits of an optimized allocation of shelf space include reduced lost sales and overall increases in customer satisfaction.

The goal of this thesis is to find a way to divide the available shelf space between a given set of products so that the need for restocking the shelves is reduced and the opportunity cost in the form of lost sales is minimized. This approach is different from the existing methods in literature, many of which focus largely on the space elasticity of the demand. In this thesis, the shelf space allocation problem is formulated as an optimization problem, where the function to be minimized is the expected quantity of lost sales. The main constraint is the available shelf space.

The optimization problem is solved using the simulated annealing algorithm, and different variations of the algorithm are compared. The algorithm performs well with a linear cooling schedule and a static step size of 1. Good results are also obtained with a logarithmic cooling schedule, when the control parameter is chosen carefully. Using a method known as thermodynamic simulated annealing did not result in improvements for the test cases. In all of the variations, the selection of the initial temperature was found to have a significant impact.

The simulated annealing algorithm is a valid option for solving the shelf space allocation problem. There are variations of the algorithm that are suitable for different situations, and by optimizing the values of the different parameters one can improve the results. Further research is still needed before using these results in real-life applications.

Keywords:	space planning, shelf space allocation, simulated annealing, integer optimization, nonlinear knapsack problem
Language:	English

Aalto-universitetet
Högskolan för teknikvetenskaper
Master's Programme in Mathematics
and Operations Research

SAMMANDRAG AV
DIPLOMARBETET

Utfört av:	Ella Warras		
Arbetets namn:	Optimering av hyllutrymmesallokering inom dagligvaruhandeln		
Datum:	22 maj 2019	**Sidantal:**	viii + 72
Huvudämne:	Systems and Operations Research	**Kod:**	SCI3055
Övervakare:	Professor Harri Ehtamo		
Handledare:	Teknologie doktor Tuomas Viitanen		

Dagligvaruhandeln är en bransch med hård konkurrens, höga försäljningsvolymer och låga vinstmarginaler, vilket betyder att det är särskilt viktigt att hålla kostnaderna under kontroll och optimera processerna. Arbetskraftskostnaderna i butiker och lager utgör en stor del av detaljhandelns kostnadsstruktur, och det finns även stor potential för besparingar inom det området. Genom att optimera användningen av hyllutrymme är det möjligt att minska på tiden de anställda är tvungna att använda på att föra varor från lagret till hyllan. Andra fördelar är en minskning av den förlorade försäljningen och en allmän ökning i kundnöjdheten.

Målet med detta diplomarbete är att hitta ett optimalt sätt att fördela det tillgängliga hyllutrymmet mellan en given uppsättning produkter så att behovet att fylla på hyllorna minskar och möjlighetskostnaderna i form av förlorad försäljning minimeras. Denna prioritering skiljer sig från befintliga metoder i litteraturen, varav många fokuserar starkt på efterfrågans utrymmeselasticitet. I detta arbete formuleras hyllutrymmesallokeringsproblemet som ett optimeringsproblem, där funktionen som minimeras är den förlorade försäljningens väntevärde. Huvudsakliga bivillkoret är det tillgängliga hyllutrymmet.

Optimeringsproblemet löses med hjälp av metoden simulerad glödgning, och olika varianter av algoritmen jämförs. Algoritmen presterar väl med en linjär nedkylningsfunktion och en konstant stegstorlek på 1. Goda resultat nås även med en logaritmisk nedkylningsfunktion, då kontrollparametern väljs noggrant. En metod som kallas termodynamisk simulerad glödgning ledde inte till förbättringar i resultaten för testfallen i denna studie. I alla varianter av algoritmen hade valet av starttemperatur en betydande inverkan.

Simulerad glödgning är ett fungerande alternativ för att lösa hyllutrymmesallokeringsproblemet. Det finns varianter av algoritmen som lämpar sig för olika situationer, och genom att optimera värdena på de olika parametrarna kan resultaten förbättras. Fortsatt forskning behövs ännu innan dessa resultat kan användas för verkliga tillämpningar.

Nyckelord:	hyllplanering, allokering av hyllutrymme, simulerad glödgning, heltalsoptimering, ickelinjärt kappsäcksproblem
Språk:	engelska

Aalto-yliopisto
Perustieteiden korkeakoulu
Master's Programme in Mathematics and Operations Research

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Ella Warras		
Työn nimi:	Päivittäistavarakaupan hyllytilan allokoinnin optimointi		
Päiväys:	22. toukokuuta 2019	**Sivumäärä:**	viii + 72
Pääaine:	Systems and Operations Research	**Koodi:**	SCI3055
Valvoja:	Professori Harri Ehtamo		
Ohjaaja:	Tekniikan tohtori Tuomas Viitanen		

Päivittäistavarakaupan alalla kilpailu on kovaa ja myyntimäärät ovat suuria, mutta voittomarginaalit pieniä, minkä vuoksi kulujen hallinta ja prosessien optimointi on erityisen tärkeää. Myymälöiden ja varastojen työvoimakulut muodostavat suuren osan vähittäiskaupan kustannusrakenteesta, ja se on myös osa-alue, jolla voidaan saavuttaa suuria säästöjä optimoimalla eri prosesseja. Hyllytilan käytön optimointi voi vähentää työntekijöiltä aikaa, joka kuluu tavaran siirtämisessä takahuoneesta hyllyyn. Muita etuja optimoidussa hyllytilan allokoinnissa ovat menetetyn myynnin väheneminen sekä yleinen asiakastyytyväisyyden kasvu.

Tämän diplomityön tavoite on löytää optimaalinen käytössä olevan hyllytilan jako annettujen tuotteiden välillä siten, että hyllytystyön tarve vähenee ja vaihtoehtoiskustannukset menetetystä myynnistä laskevat. Tämä lähestymistapa on erilainen verrattuna kirjallisuudesta löytyviin menetelmiin, sillä monet niistä keskittyvät pääosin kysynnän tilajoustoon. Tässä diplomityössä hyllytilan allokointiongelma muotoillaan optimointiongelmana, jossa minimoitava funktio on menetetyn myynnin odotusarvo. Ongelman tärkein rajoite on käytössä oleva hyllytila.

Optimointiongelma ratkaistaan käyttämällä simuloitu jäähdytys -menetelmää, ja algoritmin eri variaatioita vertaillaan. Algoritmi tuottaa hyviä tuloksia lineaarisella jäähdytysfunktiolla ja staattisella askelkoolla 1. Tulokset ovat myös lupaavia kun käytetään logaritmista jäähdytysfunktiota, mutta se vaatii säätöparametrin huolellista valintaa. Termodynaaminen simuloitu jäähdytys -niminen menetelmä ei tuottanut parannuksia testien tuloksiin. Kaikissa variaatioissa alkulämpötilan valinnalla osoittautui olevan suuri merkitys.

Simuloitu jäähdytys on toimiva algoritmi hyllytilan allokointiongelman ratkaisuun. Algoritmista on variaatioita, jotka soveltuvat erilaisiin tilanteisiin, ja tuloksia voi parantaa optimoimalla eri parametrien arvoja. Aiheesta tarvitaan vielä jatkotutkimusta ennen kuin tuloksia voi käyttää tosielämän sovelluksissa.

Asiasanat:	hyllysuunnittelu, hyllytilan allokointi, simuloitu jäähdytys, kokonaislukuoptimointi, epälineaarinen selkärepun täyttöongelma
Kieli:	englanti

Acknowledgements

This thesis has been a challenging project, and there are many people that have helped me along the way. First, I would like to thank RELEX Solutions for giving me the opportunity to write this thesis. A special thanks goes to my advisor Tuomas Viitanen, whose support, dedication and encouraging attitude have been invaluable for the thesis process. I also wish to thank my supervisor Harri Ehtamo for providing his feedback and expertise.

I have had the opportunity to experience and learn so much during the past years, and I have made many wonderful friends along the way. The community in Otaniemi and especially Teknologföreningen, my second home during these years, have played a large part in my life and I cannot thank all the people involved enough.

A huge thanks also goes to my colleagues at RELEX for supporting me and making the workdays fun during the last couple of years. All the discussions and support have helped me in the process of completing this thesis.

Most importantly, I want to thank my friends and family for all their love and support. Bra faijor, without you the past years would have been much more boring, and I hope we have many more adventures together in the future! Finally, I want to thank my mom and dad for always supporting me in everything I do.

Espoo, May 22, 2019

Ella Warras

Glossary

days of supply	number of days the current stock will last when taking into account future demand
facing	one unit of a product that is visible on the front on the shelf or other fixture
fixture	any type of shelf or other structure that can be used for presenting products
heuristics	simple rules applied empirically to find a "good enough" solution quickly
lead time	time between order and delivery
macro space planning	floor space planning, decisions about where on the store map product categories are placed
metaheuristics	methods that are more general and problem-independent than heuristics, provide a more thorough approach
micro space planning	shelf space planning, decisions about where individual products are placed on the shelf
planogram	shelf plan in picture form, shows where each product is to be placed on the shelf
stock-out	when the product is sold out

Contents

Chapter 1

Introduction

In the world of retail, the combined sales for the global top 250 companies reached US$4.4 trillion in 2016 (Deloitte, 2018). In the USA, grocery retailers sold US$648 billion worth of products in 2016, and even in Finland the grocery market reached EUR18.2 billion in value in 2018 (United States Department of Agriculture, 2018; Nielsen, 2019). Grocery retail is an enormous industry, but profit margins are relatively small. It was ranked one of the least profitable industries in 2017, with a net profit of only 2.2% (Biery, 2017). Grocery is also an increasingly competitive industry, and all of this means managing costs and optimizing processes is especially important, as even small improvements can result in huge savings in expenses for the retailer. Research by e.g. Angerer (2006) shows that there is a lot of potential for improvement in the fast moving consumer goods industry by using different technological solutions for optimizing the store replenishment process.

Store and warehouse labor costs constitute a large part of the retail cost structure, and it is also an area where large savings can be obtained by optimizing different processes. Effective space planning can save time for the store employees in the shelf stacking process. The savings can come from many different parts of the process, but one aspect is the shelf space allocation, which, if done optimally, can reduce the amount of time the employees have to spend bringing stock from the backroom storage to the shelf. Optimizing the use of shelf space brings many other benefits too, such as reduced lost sales when customers are not met with an empty shelf where their preferred product should be, and overall increases in customer satisfaction when the full assortment of products is presented in a clear way, without out-of-stocks.

A review by Hübner and Kuhn (2012) shows that in the area of retail category management, there is a large amount of high-quality research on how to best manage the space and assortment aspects in the stores. However, that research knowledge has not reached the software solutions that exist today. Most systems still use simple rules and settings for making space and assortment planning decisions, while the methods found in the literature are more advanced. There is potential for closer cooperation between the two; practical software solutions can become more intelligent by incorporating research findings, and research studies can benefit from some real-life insights about the use cases.

1.1 Problem Statement

In this thesis, the goal is to formulate a practical method to be used in shelf space allocation planning. More specifically, the idea is to find a way to divide the available shelf space between a given set of products so that the need for restocking the shelves is reduced and the opportunity cost in the form of lost sales is minimized.

The objective of this study can be formulated as follows:

What factors should be taken into account when allocating shelf space between products in a retail store, and how can the optimal allocation be solved efficiently?

In order to answer this question, there are some steps that need to be taken. Firstly, it is important to study the current available methods that have been used for solving the shelf space allocation problem. After that, the shelf space allocation is formulated as an optimization problem, which means deciding what a good objective function is, as well as defining the optimization constraints. Then, based on the characteristics of the optimization problem, a suitable solution method needs to be found and implemented. The selected algorithm needs to be suited for the different requirements of the optimization problem. Different variations of the algorithm are tested and compared in order to find the most efficient method for the optimization.

1.2 Scope of the Thesis

This study is focused on shelf space allocation in retail stores, and specifically grocery retail. The findings can be applicable to some non-grocery retail stores, but the examples and test data are from the grocery industry. Shelf space optimization can only be done if the shelf space is, in fact, limited. This is the case in most grocery stores, but excludes some specialty items such as premium class watches or clothing. The space planning decisions of those retailers typically do not concern a limited amount of space that needs to be filled, so the approach is quite different.

The study is limited to products with previous sales history, so completely new products that are being introduced are not included in the scope. It is, however, possible to use the sales history of another product as a reference, if there is a comparable reference product to be assigned. The products in this study are also assumed to have regular and reasonably frequent deliveries to the store, since the allocated shelf space is meant to satisfy enough demand so that few refills of the shelf are needed before the next delivery. It could also work in some situations with infrequent deliveries, but the main focus of this thesis is the frequently delivered products, such as grocery products.

The changes in demand as a function of changes in the amount of space were not included in the study, although the topic is discussed in Section 2.3 of this thesis. Space elasticity was determined to be such a wide topic, that it was excluded from the scope of the thesis. The same applied to substitution effects and cross-space elasticity.

1.3 Structure of the Thesis

The thesis starts with a chapter on the general background of space and assortment planning, with the key concepts presented. Shelf space planning is specifically discussed in more detail, including some key concepts related to replenishment and demand elasticity. After this, a review of relevant literature on shelf space planning is presented. The chapter is concluded with a section on the formulation of the shelf space allocation problem as an optimization problem, with an objective function and constraints.

Next, in Chapter 3, a general introduction into optimization is given, after which the different possible solution methods for this type of an optimization problem are discussed. The selected method, the simulated annealing

algorithm, and its advantages are presented in detail, along with descriptions of the variations that are tested in this study. The results are presented in Chapter 4, along with a description of the test data and other details related to the test setup. The chapter is divided into sections for the different methods that were tested. In Chapter 5 the key findings of the study are presented, and some analysis is provided on the quality and practical implications of the results. In the final section, some suggestions are presented for future research possibilities.

Chapter 2

Shelf Space Allocation

In this chapter, a general overview of space and assortment planning is presented, along with definitions for some key concepts in this area. After that, some previous approaches to solving the shelf space allocation problem are reviewed.

In retail stores there are many space and assortment planning aspects to consider. Assortment planning refers to determining the set of products that the store should sell in each category, while space planning covers a wide range of major and minor decisions related to the placement of the products in the store.

2.1 Assortment Planning

Before deciding on where to place products in the store, an important step is determining which products should be included in the assortment of each store. The assortment planning process includes decisions about which product groups should be included and how the product hierarchies with different categories should be constructed. Larger stores such as grocery hypermarkets have the possibility to provide a larger assortment of different products, whereas smaller stores have to limit the size of the assortment in order to have enough room for the stock. Some stores might want a smaller number of different products so that the core assortment is always covered and the customer clearly sees what the options are in the store, and others prioritize a large assortment to satisfy all the different customers' needs. It has been shown that in many cases reducing the size of the assortment can lead to an

increase in sales (Boatwright and Nunes, 2001).

More detailed assortment decisions include the selection of brands and specific products for the assortment of each store. All these decisions can be made locally at the store, centrally at chain level or as a combination of these two. In the case of chains with regional variations in the demand, it can be useful to let the local staff have some degree of influence in deciding the assortment. The department store chain Macy's in the USA saw significant increases in sales when a local approach to the assortment was adopted (Clifford, 2010).

When deciding the assortment for a store, the retailer needs to analyze the decision making process of the customer. This can be done using a consumer decision tree, that shows the different levels of decisions that the customer is making in their mind before deciding on a specific product. The first level of the decision tree is the first decision that a customer typically makes in the process of purchasing a product. The different characteristics that the customer can consider are the flavor or color of the product, price level (budget, mainstream, premium), brand, package size and many different types of subcategories that can be related to the intended target group (demographic factors such as age or gender, other considerations). The decision tree can be modified for different store locations, if there is variation in the decision process of the customers.

In Figure 2.1 there is an example of a decision tree for the yoghurt category. In this case, the first level in the decision hierarchy is the choice between plain and flavored yoghurt. After that, the next decision is full fat or low fat yoghurt, and other aspects come after that. This is only one example of what the decision process may look like, but in any case, this type of analysis is an important part of the assortment planning process. In this example case, it is important to stock both plain and flavored yoghurts, both full fat and low fat, but having many different package sizes and flavors is seen as less important.

The optimal assortment for a store depends on the substitution decisions and preferences of the customers. Customers can make substitution decisions before going to the store, knowing the store assortment, and not change their decision based on availability. Thus, if they decide on product A and it is out of stock, they do not purchase anything else instead. This type of substitution behavior is called *static* or *assortment-based*. If the customer makes their decision in the store, based on the availability of the products, that is called *stock-out-based* or *dynamic* substitution behavior. It is important to have a large enough variety in the assortment, so that the preferences of the

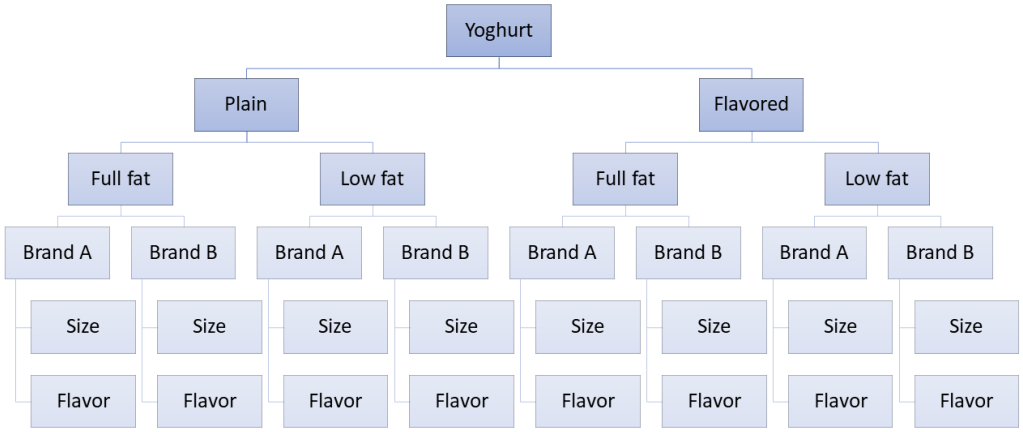

Figure 2.1: Example of a decision tree

customers are satisfied, but it should also be noted that there is a possible return to be had when one product is out of stock. If the retailer aims to have enough stock to satisfy the demand of every product in assortment, the customers will never have to resort to buying their second most preferred product, which might have a higher profit margin. (Mantrala et al., 2009; Honhon et al., 2010)

The stock-out-based substitution behavior depends on the category of products in question. Substitution mostly occurs within categories, so a customer who wants to buy e.g. a specific type of cookies but discovers they are out of stock will typically not substitute that choice for a box of cereal. There are, however, some pairs of categories where cross-category substitution can occur, but those are less common. An example of this would be chocolates and chips; they are in separate categories but both are common choices for a movie night snack. For some categories or products, stock-out-based substitution might not occur almost at all, if the products are considered unique in some way by the customers. On the other hand, there are many categories in which the stock-out-based substitution is quite strong. In basic categories, such as bread and milk, the customer most likely buys a substitute product of a different brand or variety if their first choice is not in stock. The above examples might not apply in all cases, since customer behavior is different in different parts of the world, but they do illustrate the concepts.

One aspect that grocery retailers need to consider when making decisions about the assortment is the different product attributes related to dietary requirements and preferences. The retailers need to be aware of all the available options of food items that are e.g. lactose free, gluten free or vegan, and decide which ones should be part of the assortment. These attributes are important to consider separately, since they affect the substitution behavior in different ways.

All of these above-mentioned substitution effects are far from straightforward to consider in the assortment planning process. Sales history data can be analyzed in order to find substitution relationships and other relevant information, which can then be used to support the decision making. Using this data can be complicated, since it is often difficult to distinguish fluctuations caused by substitution relationships from other types of fluctuations in the sales. When the sales of product A decrease because its substitute product B is on sale or because a new product C has been introduced into the assortment, this effect is known as *cannibalization*. These events can be detected from past sales data, if it is possible to isolate the cannibalization effect from other effects on the sales during the same period. Besides all the information received from the historical data, retailers typically also have knowledge about their particular assortment of products, region and customer base that they consider when making assortment decisions in their stores.

The assortment planning does not include decisions about how much to order of each product, how many facings[1] of each product to stack on the shelf or which product groups are placed next to each other in the store layout. Those questions are addressed in the space planning process.

2.2 Floor Space Planning

After the assortment decisions have been made, space planning is needed to determine where in the store to place each product. Space planning decisions can be roughly divided into two categories: floor planning (sometimes referred to as macro space planning) and shelf space planning (also known as micro space planning). Floor planning refers to store layout planning on

[1] A facing is a unit of a product that is visible at the front of a shelf or other type of display. Having three units of the same cereal box next to each other on a shelf means that the product has three facings, regardless of the number of units stocked behind the front box.

category level, whereas micro space is about shelf space allocation within a category. An overview of shelf space planning is presented in Section 2.3.

Floor planning decisions are important to consider for optimal results in the sales of a retail store and ensuring maximum utilization of each square meter of floor space. When this process is started, it is assumed that the assortment decisions have already been made. This usually means that the different categories or product groups are defined, and typically these groups somehow represent the groupings when it comes to allocating shelves to the categories; the products in e.g. the cereals group will most likely be placed together in one place, either on one shelf or on adjacent shelves.

At the floor planning stage, it is not necessary to know the list of specific products that will be placed on the shelf, since the planning is done on category level. Categories can be allocated one or more shelving units[2] or other fixtures, such as pegboards or racks for hanging items. For products that require storing in cold temperatures, there are different types of cold cabinets and freezer boxes. In Figure 2.2 there is an example of how the floor layout of a grocery store can look. The floor plan is typically made on department level first, and then on category level inside each department. The example in the figure shows e.g. dairy products grouped together in one section of the store, with the more specific categories yoghurts and milk shown separately inside the department.

There are many aspects to consider in the floor planning process. Placing certain categories next to each other can increase their sales, if there are complementary products in the two groups or if the groups are otherwise related in some aspect. It is important for the retailer to consider the foot traffic paths of the customers in order to be able to influence them. One common way of increasing sales in grocery stores in general is to place staple items, such as milk, in the back of the store. This way the customers have to walk through the whole store and pass by many categories and products on the way. The retailer also has to decide how much space to allocate to the shelves overall, and how much space should be left empty for customers to be able to walk between the shelves.

[2]A shelving unit is a system of multiple shelves that are stacked vertically.

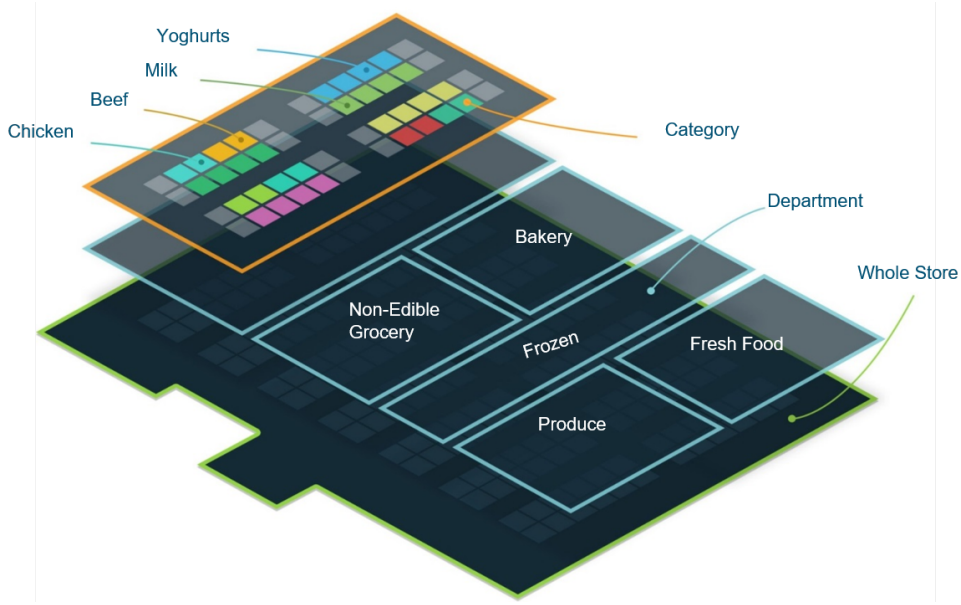

Figure 2.2: Example of a floor plan in a grocery store
(Image: RELEX Solutions)

2.3 Shelf Space Planning

In shelf space planning, the categories and their specific assortments are known beforehand. The decisions that need to be made usually include how much shelf space to allocate to each product, which products to group together on the shelf and in which exact spot to place the products on the shelf (vertical and horizontal location). The result of this planning is normally a *planogram*, i.e. a picture of the shelf or other fixture with all the products placed on it. Planograms are used for communicating the layout of the products to the store staff. In Figure 2.3 there is a generic example of a planogram in a grocery store, in this case from the cereals category.

The three main decisions that need to be made concerning shelf space planning are (1) which products to place next to or near each other on the shelf, (2) where on the shelf to place each product or group of products, taking into account horizontal and vertical location and (3) how much shelf space to allocate for each product (not necessarily in this particular order).

The retailer needs to decide (1) which criteria to use for grouping products together in the shelf. For this, the retailer can use a consumer decision tree. The decision tree and the decision making process are presented in

Figure 2.3: Example of a planogram in a grocery store
(Image: RELEX Solutions)

more detail in Section 2.1 and Figure 2.1. The first level of the decision tree represents the first decision that the customer makes, which means the products should perhaps be grouped according to that aspect on the shelf. This makes the shopping experience more convenient for the customer, as they are more likely to find many valid options in the same place on the shelf, and they might even find and purchase a new product that matches their preferred criteria.

When it comes to the shelf location of the products (2), there are different aspects that can be considered. Regarding the vertical location, a retailer might want to place high-margin products on eye-level for them to be noticed and thus hopefully increase sales, whereas budget alternatives are commonly placed on the lower shelves. Another option to consider is placing the most popular items around eye level, because that way customers do not have to spend much time looking for the desired product and the customer satisfaction increases. The horizontal location is in most cases less important to consider than the vertical location (Hansen et al., 2010), but placing a product right at the start of the aisle might still make a difference compared to placing it further inside the aisle.

The amount of space (number of facings) to be allocated to each product (3) depends on various factors. The decisions can be made on subcategory level first, so the amount of space per subcategory can be decided before making decisions for individual products. These subcategories or groupings are the

ones that the retailer has determined to be the most relevant when analyzing the customer's decision making process (see point (1) above). When making decisions about the shelf space allocation, the assortment of that category is already determined. In addition to the list of products that are to be placed on each shelf, it is necessary to know any possible minimum limits to the shelf allocation, which can be general limits ("at least two facings of each product") or product-specific.

The forecasted demand of the products can be used to determine how much space each product should be allocated. Other metrics can also be used, e.g. past sales or the sales margin, or a combination of two or more of these. Using the forecasted demand as a basis for the allocated space increases, for example, the probability of a customer finding what they need in the shelf (assuming that they find the correct shelf in the first place, but that has to do with floor planning). The substitution behavior of the customers can also be considered here, since there is often another optional product that the customer will buy in the event of a stock-out of their most preferred product.

2.3.1 Replenishment Costs

One related aspect to consider in shelf space allocation is minimizing different costs. The costs that are related to shelf space allocation include shelf stacking costs (personnel costs), delivery costs and other inventory costs. Orders from the supplier or warehouse to the store can be placed daily, weekly or with some other suitable interval. Once an order is placed, the time it takes until the order is delivered to the store is called the *lead time*. If the lead time is 3 days and the orders are made once a week on Mondays, the deliveries arrive on Thursdays.

If the goods are delivered less frequently, the delivery costs are lower, but as a consequence the need for inventory space grows, as do the inventory handling and storing costs. In the case of perishable goods (a majority of grocery products, for example), the products are less fresh when they arrive at the shelf and also at the customer's home, which increases spoilage and possibly makes the products less appealing to the customers in the store. Here it is important to consider the different spoiling times of the products, since they vary greatly between product categories. Taking into account the demand of the products in the shelf space planning process helps in lowering costs. If the shelf space is allocated in such a way that stock-outs are minimized, there is less need for restocking the shelves from the backroom storage. In addition, if the order quantity always fits in its allocated shelf space, there is no need

for a backroom storage at all, which further streamlines the replenishment process. A relevant metric to consider in the context of replenishment is the number of *days of supply*. The number is calculated using an estimate for the future demand, and the days of supply number represents how many days the current stock will last before there is a stock-out.

2.3.2 Space Elasticity

The space elasticity of demand is often considered in decisions related to shelf space allocation. Space elasticity is a factor that represents how much the sales of a product increase as a result of increasing the allocated shelf space for that product (or in reverse). Curhan (1972) presented the space elasticity mathematically as follows:

$$E = \frac{(U_1 - U_0)/U_0}{(S_1 - S_0)/S_0}, \tag{2.1}$$

where U is the unit sales of the product, S is the amount of space allocated to the product and 0 and 1 refer to the moments in time before and after making the change in the amount of space allocated. So for example a space elasticity of 0.25 would mean that if the allocated shelf space for that product was increased by 100 percent, the sales of the product would increase by 25 percent.

There are many complexities related to the space elasticity effect that need to be considered when planning shelf space allocation. The space elasticity of a product can vary depending on the product group, the location on the shelf (horizontal, vertical), the location of the shelf in the floor plan, the price, possible promotions and many other things. Space elasticity can be measured empirically, but it is difficult to isolate the space elasticity effect from all these other factors. Space elasticity is typically positive, but there can be cases where the it is close to zero or even negative, if an increased shelf space area makes the product seem less attractive to consumers. More research that touches on the topic of space elasticity is presented in Section 2.4.

2.3.3 Cross-Space Elasticity

The cross-space elasticity is another, less investigated effect on the demand of products in retail stores, where a change in the shelf space of one product

affects the demand of another product (Corstjens and Doyle, 1981). Using the same notation as Equation (2.1), the cross-space elasticity between products i and j can be defined in the following way:

$$E_{ij} = \frac{(U_{1,i} - U_{0,i})/U_{0,i}}{(S_{1,j} - S_{0,j})/S_{0,j}}. \tag{2.2}$$

If the cross-space elasticity is -0.25, then if the shelf space allocated to product j increases by 100 percent, the sales of product i decrease by 25 percent. In this case i and j can be considered substitutes. A real-life example of substitutes is two different brands of a similar soft drink. If the cross-space elasticity is positive, the products are complements (for example pasta and pasta sauce).

Based on Equation (2.2) it can be seen that the cross-space elasticity between i and j is not the same as the cross-space elasticity between j and i. In other words, if the sales of pasta A increase by 75 percent after the shelf space of pasta sauce B is increased by 100 percent ($E_{AB} = 0.75$), that does not imply that the sales of the pasta sauce would react in the same way to a change in the shelf space of the pasta ($E_{AB} \neq E_{BA}$).

2.4 Shelf Space Planning in Literature

There are a number of approaches in literature for solving different problems related to shelf space planning. Regarding shelf space allocation, there have been some simple heuristic approaches in the 60's and 70's, that Zufryden (1986) reviewed. These methods applied heuristic rules to allocate shelf space simply based on past sales or sales margin, in order to make it operationally practical to use.

Zufryden (1986) concluded that these methods were not leading to optimal solutions, so he then approached the shelf space optimization problem with dynamic programming. The objective function included space elasticity and different cost components, as well as demand-dependent marketing components. Zufryden enabled a general form of the objective function as well as integer requirements for the solution. The shelf was divided into a number of predefined slots, one shelf would be e.g. 40 slots and the number of slots allocated for a product needed to be a multiple of the size of that product (expressed in number of slots). He also brought up practical concerns related to computational resources.

Cox (1964) found that there was not much research available on the topic of space elasticity at that time, so he conducted an experiment with four different products in six supermarkets. He did not find any conclusive results regarding the importance of considering space elasticity, which is logical considering the small sample size. Curhan (1973) reviewed some of the research on shelf space elasticity that existed at the time, and he concluded that although there had been experiments on the topic, most of them were not good enough to use in any general case. He also noted that the results had not notably affected the way the shelf space is managed in retail businesses.

Corstjens and Doyle (1981) developed a model for shelf space allocation that utilized space and cross-space elasticities, as well as product margins and inventory costs. They investigated data from one retailer with 140 stores and concluded that including space and cross-space elasticities in the optimization problem yields higher profits than previous models that did not include these demand components. However, the space and cross-space elasticities are not simple factors to take into account. They vary greatly with factors such as the location on the shelf, package color and product class, and thus it is difficult to estimate the space and cross-space elasticities in a reliable way (Curhan, 1973; Drèze et al., 1994). Corstjens and Doyle (1981) did not include spoilage of products as a factor in their model, in addition to other aspects such as the location of the products on the shelf.

In their article Drèze et al. (1994) presented the results of experiments related to the link of the sales profit to the allocated shelf space and the shelf location. The research scope consisted of 60 stores from one retailer, all of them large stores (> 4000 m^2). Drèze et al. found that the location impacted the sales more than the amount of allocated space, assuming there was some minimum limit for the allocated space. The vertical location was shown to have a larger effect on the sales than the horizontal location.

Desmet and Renaudin (1998) investigated the shelf space elasticity of different product groups. Their results indicated that the space elasticity is a more significant factor for products that are classified as impulse-buy products. The type of store was not shown to have any significant impact.

Assortment decisions also have an impact on shelf space aspects. If there are too few facings of a product on the shelf and the product runs out, the outcome is normally affected by possible substitute products. The lost profit depends on the existence of direct substitutes, the degree to which customers choose substitutes instead of not buying any product at all, and the profit margins of the substitutes. Urban (1998) presented a generalized model of the shelf space allocation problem that includes inventory costs in

addition to space elasticity effects. The article also includes the possible effect of substitute products through a factor that represents "the degree of substitutability" between two products.

Yang (2001) presented a heuristic method for solving the shelf space allocation problem. In this method, a priority ranking is calculated based on the profit to size ratio of each product, where the size is the width of one facing. The allocation is then done step by step for each product in the determined order, taking into account possible minimum and maximum limits for the number of facings of the product. Here the profit is formulated as a linear function of the number of allocated facings. Yang compares the problem to a knapsack problem, which is something this thesis discusses more in depth in Section 3.1.

These previous approaches used linear profit functions, but as Yang (2001) mentioned in his conclusions and Hansen et al. (2010) explained more thoroughly, the increase in profit from adding a second facing of a product is normally greater than the profit that comes from adding a third facing (which is greater than the added profit of a fourth, etc.). If this aspect is to be taken into account, the profit function needs to be nonlinear.

One approach that includes a nonlinear profit function was presented by Lim et al. (2004). The article compares different metaheuristic methods as well as simple heuristics for solving optimization problems, with linear and nonlinear profit functions (see Section 3.2 for more details on the topic of metaheuristics). In addition to the nonlinear profit function, Lim et al. presented a modification of the profit function that considered the effect of grouping related products together on the shelf.

Hansen et al. (2010) built a model for shelf space decision making that included shelf space allocation and location factors. As opposed to Lim et al. (2004), Hansen et al. used an optimization model with a nonlinear profit function that was modified so that it could be solved using linear programming. The article also included the horizontal and vertical location aspects in the profit function and concluded that they each had an important impact, although the vertical location effect was approximately twice the size of the horizontal location effect. This is fairly consistent with the findings of Drèze et al. (1994).

Hansen et al. (2010) also discussed space elasticity, saying that the only reason for retail stores to place more facings of some products on the shelf is the increase in sales that is assumed to follow. This conclusion completely ignores the labor costs that incur each time a shelf has to be restocked, as

well as the opportunity cost of the lost sales that may occur with stock-outs. Shelf stacking represents a significant share of the total labor, and labor costs are in general quite a large part of the total expenses of a retail store. This is supported by e.g. Hübner and Kuhn (2012).

Pricing aspects are in many cases useful to take into account when it comes to space and assortment planning. Hübner and Kuhn (2012) presented a review of some commercial models as well as scientific research related to different category management areas. The article mentions that pricing, among other things, has an influence on the substitution aspects, and that this has been researched by McIntyre and Miller (1999) and Murray et al. (2010).

2.5 Shelf Space Allocation as an Optimization Problem

There are many areas of category management that could be considered and optimized in combination with shelf space. In this thesis, some assumptions are made in order to limit the scope. The assortment of the products is assumed to be predetermined, so the shelf space allocation starts with a given list of products, usually from the same product category. Another assumption is that the delivery schedule of each product is known.

In this thesis, the shelf space allocation problem is formulated as an optimization problem with an objective function and a number of constraints. The objective function expresses the quantity to be optimized as a function of the allocated shelf space to each product i ($i \in [1, N]$, where N is the number of different products to be placed on the shelf). The quantity can be something that should be minimized or maximized.

A logical option for the objective is to directly maximize the (monetary) profit, which is the model that Corstjens and Doyle (1981) followed. In this model the objective function consists of a profit component and a cost component. The profit component includes sales, profit margin, space elasticity and cross-space elasticity of sales. The cost component consists of the inventory costs of the product. The model by Corstjens and Doyle does not include shelf stocking costs in any way, but there is a cost elasticity component.

One issue with the Corstjens and Doyle model is that it relies heavily on the space elasticity components, while it has been shown that the importance of space elasticity varies significantly depending on the characteristics of the

store as well as the location on the shelf. The profit is also not a linear function of the allocated space. The benefits from adding a facing of a product to a shelf are extremely low after a certain threshold point. (Drèze et al., 1994)

When choosing the objective for the shelf space allocation problem, the specifics of the situation affect the priorities. Depending on various characteristics of the specific market, country or region, store, etc., the objectives can differ considerably. In some countries around the world the labor costs are not part of the main considerations, whereas in other countries they are.

This thesis assumes the premise that minimizing shelf stacking labor is the most important goal. It is not optimal to minimize these labor costs solely by minimizing the number of times a shelf has to be restocked, since this approach may lead to large amounts of lost sales. Instead, the allocation should aim to provide enough stock of each product to the shelf, so that the stock lasts for as long as possible and thus the required shelf stacking times are reduced. If there is a backroom storage, shelf restocking can occur also in between deliveries, but if there is no storage room, shelves are only restocked when deliveries arrive. In both of those cases, the number of times the shelf has to be restocked is minimized when the stock on the shelf satisfies the demand for as long a time period as possible: if there is a backroom, the number of restocking times between deliveries can be lowered, and both with or without a backroom, the delivery frequency can in some cases be reduced. In this study, the problem is approached by minimizing the lost sales in the period between deliveries, because this approach considers the uncertainty of the sales and leads to the highest level of satisfied demand. Regarding the product availability, the optimal allocation is also a beneficial solution for the shelf stacking labor, since a high level of availability between two deliveries leads to less need for restocking during that period, and possibly less frequent deliveries over all.

In conclusion, minimizing the risk of a stock-out was chosen as the priority, and specifically minimizing the estimated lost sales quantity. The objective function could have been set as the minimum of the days of supply of each product, the goal being to maximize this (and thus maximize the time it takes until the next stock-out), but as such it is not a suitable objective. This approach only improves the days of supply of the "worst" product on the shelf (the one with the lowest days of supply), not the overall result. If the days of supply of the "worst" product can not be improved further, due to e.g. the facing width being larger than the remaining empty space on the shelf, the optimization does not seek to fill the remaining space with other

products, even if there were some with a sufficiently small facing width. This leads to a suboptimal solution.

In order to calculate the estimated lost sales that occur due to stock-outs, one needs to have some kind of approximation for the future sales, as well as the corresponding degree of uncertainty. For this, the average daily sales and the corresponding sales standard deviation can be used, if one makes the assumption that the sales follow the normal distribution. Some other known distribution could also be used for this purpose, but the normal distribution was chosen for the sake of simplicity. In addition to the sales estimates, the number of days until the next delivery for each product is also needed for minimizing the risk of stock-outs. In practice this could be represented by the maximum number of days until the following delivery of each product, since that way it represents the worst-case time until the next delivery. That can be calculated as the sum of the number of days between orders and the longest lead time of the product.

The approach of minimizing the lost sales assumes that when there is a stock-out, the lost sales are equal to the unsatisfied demand for those days. If there were no sales on the stock-out day, the lost sales are equal to the forecast for that day, and otherwise they are equal to the difference between sales and forecast. This method of determining lost sales ignores possible substitution effects, i.e. that a customer might choose another similar product if the preferred product is not available. However, in this case the goal is to minimize stock-outs in order to minimize labor costs from shelf stacking, hence the substitution effects can be considered less important. Additionally, the fact that one facing represents more than one unit when looking at the depth of the shelf is ignored, since only the relative amounts between different products matters, not the absolute numbers, and it is assumed all products have room for roughly the same number of items per one facing. This is also a simple factor to include in the model later if there are in fact notable differences between the depth dimensions of different products.

2.5.1 Objective Function

In this thesis, the goal of the optimization is to minimize the lost sales. When modeling the uncertainty of sales using the normal distribution assumption with the average daily sales and standard deviation, there is some risk of a stock-out on any given day. Of course, if the demand is quite stable and there is plenty of stock to satisfy the demand, the risk is low, but it still exists

assuming there is some variance in the sales. Based on this, the expected quantity of the lost sales can be calculated.

Before introducing the lost sales calculations in detail, it needs to be mentioned that this approach to the objective function does not include many aspects such as placement and grouping on the shelf, as well as the space elasticity components. These aspects, which have been discussed in more detail in Section 2.3, and more can be included in the objective function at a later stage by adding new multiplicative or additive components in the calculations.

The demand of a product during the days before the next delivery is denoted as x. This variable x is measured in units sold. It is assumed to be normally distributed with the mean $\mu = RA$ and the standard deviation $\sigma = \sqrt{R}B$, where R is the maximum number of days until the next delivery, A is the average daily sales of the product and B is the standard deviation of the daily sales. The probability density function of this distribution is denoted as $f_x(x)$, and the number of facings (items) of the product on the shelf as Z. The latter is the quantity that will be varied in order to reach the objective.

After calculating the expected quantity of the lost sales of a product, the quantity needs to be divided by the number of days until the next delivery R. This is necessary in order to scale the lost sales to be comparable between the different products even though they may have different delivery schedules; one unit of lost sales in one day has a greater impact than one lost sale in a seven-day period.

The probability that the number of units x that are (or would have been) sold is greater than the number of units on the shelf Z can be expressed using the complementary cumulative distribution function of x:

$$\mathrm{P}(x > Z) = \bar{F}_x(Z) = \int_Z^\infty f_x(x)\,dx, \tag{2.3}$$

in which x follows the normal distribution where μ is the mean and σ^2 is the variance. The cumulative distribution function $F_x(Z)$ gives the probability that a random value x is less than or equal to Z. Thus the complementary cumulative distribution function $\bar{F}_x(Z) = 1 - F_x(Z)$ represents the probability of x being greater than Z. The cumulative distribution function can be calculated as the integral of the probability density function f_x from 0 to the limit Z, or in the complementary case, from Z to infinity.

The expected quantity of the lost sales is then the sales that go over the limit Z multiplied by the probability, as presented below:

$$\mathrm{E}[x_L] = \int_Z^\infty (x - Z) f_x(x)\, dx, \tag{2.4}$$

where x_L is the quantity of lost sales. This can be further modified in order to reach a more accurate solution in the calculations. The substitution $t = x - \mu$ is used, so that the probability density function f_t is centered around zero. That gives the following function:

$$\mathrm{E}[x_L] = \int_{Z-\mu}^\infty (t + \mu - Z) f_t(t)\, dt, \tag{2.5}$$

where t is normally distributed with the mean 0 and the variance σ^2. Next, the substitution $u = -t$ is used so that the integral starts from negative infinity, thus canceling out some terms and simplifying the equation. After this, as well as separating the terms, the result is:

$$\mathrm{E}[x_L] = -\int_{-\infty}^{\mu-Z} u f_u(u)\, du + (\mu - Z) \int_{-\infty}^{\mu-Z} f_u(u)\, du, \tag{2.6}$$

where u is normally distributed with the mean 0 and the variance σ^2, similarly to t. After integrating by parts and canceling out terms (using $F(-\infty) = 0$), the result is the final expression for the expected quantity of the lost sales:

$$\mathrm{E}[x_L] = \int_{-\infty}^{\mu-Z} F_u(u)\, du, \tag{2.7}$$

with u following the normal distribution with the mean 0 and the variance σ^2. This is what will be used as the basis for the objective function of the optimization problem. The objective for this optimization is to minimize the expected quantity of the lost sales, when taking the sum over all products i, $i \in [1, N]$. Here the lost sales is also divided by the number of days R_i before the sum is calculated. The objective function is presented in its complete

form below:

$$\min_{Z_i} \sum_{i=1}^{N} \left[\int_{-\infty}^{\mu_i - Z_i} F_u(u)\, du \Big/ R_i \right]. \tag{2.8}$$
$$\forall i \in [1, N]$$

The objective function is not linear, and behaves in ways that may not be intuitive at first. The number of facings allocated to each product depends on the past sales of the products, but also on the uncertainty of the sales history. If all of the products have the same average sales, but the sales have different standard deviations, the objective function does not necessarily allocate more facings to the products that have a higher degree of uncertainty. Doing that would make it more likely for those products to have enough stock in the case of high upward fluctuations in sales, but on the other hand it might cause stock-outs for products with a more stable demand. In an example case like this, optimizing the objective function in Equation (2.8) might lead to the stable-selling products having more facings than the fluctuating ones, or vice versa. The result depends on the specifics of the problem in question, but the aim is in any case always to minimize the estimated lost sales.

The objective function in Equation (2.8) can be modified to include other factors that may affect the shelf space planning decisions, such as the location on the shelf, grouping and space and cross-space elasticity. This can be done in different ways depending on what the effect in question is.

2.5.2 Constraints

In addition to the objective function, some constraints are needed for the optimization. The constraints that are considered in the optimization are presented in this section. The quantity that will be varied in the optimization is the number of units of a product on the shelf, Z. One clear constraint is that Z needs to be non-negative for all products:

$$Z_i \geq 0. \tag{2.9}$$

It might be beneficial to add a minimum limit for Z for each product, or the optimization can be allowed to find the optimal solution without any

minimum limits. In practice, retailers often set minimum shelf fill levels for each product.

Another essential constraint for this problem is that each Z should be an integer, as presented below:

$$Z_i \in \mathbb{Z}. \tag{2.10}$$

The most important constraint for the shelf space allocation problem is the size of the shelf or shelves. Depending on the problem setup, one can explicitly include all dimensions of the shelf, but at least the width is a relevant constraint to consider. The height of the shelf is also important, especially if the plan is to not just stack the items next to each other but also on top of each other in the same shelf. However, a choice can be made to ignore the height dimension in the problem formulation if it can be assumed that items are only placed next to each other on the shelf, and that the items all fit in the shelf heightwise. That is what is done in this thesis. When it comes to the depth of the shelf, in most cases it is possible to place multiple items behind one another in the shelf. In this case, the choice has been made to simplify the problem and only assume that the shelf contains as many items of a product as there are facings. This way only the width dimension of the shelf space is included.

The shelf space constraint is formulated as follows:

$$\sum_i w_i Z_i \leq S, \tag{2.11}$$

where the width of each product i is represented by w_i, the number of facings is Z_i, the total width of the shelf is S and the number of different products is N. In order to simplify the problem further, the problem is limited to only one shelf in this thesis. In reality, one would have multiple shelves and then one would need to make a series of additional decisions when it comes to shelf space allocation: how many different shelves can facings of one product be placed on, which products should be placed close to each other, etc. If all questions related to placement and grouping are ignored, the optimization problem simply becomes the one-shelf problem copied x times, where x is the number of shelves.

Below is a summary of the optimization constraints:

$$
\begin{aligned}
Z_i &\geq 0 \\
Z_i &\in \mathbb{Z} \\
\sum_i w_i Z_i &\leq S \\
\forall i &\in [1, N]
\end{aligned}
$$

Chapter 3

Solution Algorithm

The shelf space allocation problem can be solved using different optimization methods. In this chapter, some general optimization techniques are introduced first, and then some more specific options for solution algorithms are investigated. The chosen simulated annealing algorithm is presented in more detail, as well as some subfunctions and variations for the algorithm. The specific test setup and the test data used for this thesis are also presented.

3.1 Optimization

Before deciding on what kind of optimization algorithm to use for solving the shelf space allocation problem, some characteristics of the problem need to be identified. The objective function, which is seen in Equation (2.8), is nonlinear. The optimization problem is constrained by the limited shelf space as well as possible minimum limits for each individual product. All variables (numbers of facings) are required to be integers.

In nonlinear optimization, one common approach is to use the Karush-Kuhn-Tucker (KKT) conditions for optimality (Karush, 1939; Kuhn and Tucker, 1951). The KKT conditions are first-order derivative necessary conditions for any solution to be a local optimum. They are useful for solving many nonlinear optimization problems, but in the case of the shelf space allocation problem presented in Section 2.5, the variables are required to be integers. This makes it unsuitable for solving using the KKT conditions, since they assume differentiable functions. One option for the shelf space allocation problem would be to use *relaxation*, i.e. to make the variables continuous,

only include the maximum shelf space restriction, and solve it using the KKT necessary conditions. Then, after an optimum is found, the variables (numbers of facings of each product) are rounded to the nearest integer. However, because the problem depends heavily on the integer requirement and the varying facing sizes, the differentiable function approach is not likely to be suitable for all the different forms of the problem.

For continuous variables Z_i, the objective function of the shelf space allocation problem is likely to be a convex function, based on the fact that adding a facing to any product always improves the result of the objective function, but we cannot be sure solely based on this. Due to the shelf space constraint as well as the integer requirement for the numbers of facings, the problem has solutions where no more facings can be added while the solution is not optimal. Being in one of these solution states means that in order to reach the optimum, some facings have to be removed, thus temporarily increasing the value of the function to be minimized, before adding facings to other products and reaching a lower function value. For solving this kind of a problem, it is advantageous to use an algorithm that does not get easily stuck on these non-optimal solutions. In addition, if the optimization problem is to be generalized to include other aspects of shelf space planning such as product groupings and location on the shelf, the objective function will have a different form, which could possibly be non-convex. Based on this, it is useful to choose a solution method that is suitable for non-convex problems with integer variables, too. The topic of choosing an algorithm is discussed in more detail in Section 3.2.

This type of problem, where a limited amount of space is distributed between a number of items that each have a different value, can be considered a variation of the well-known knapsack problem. The knapsack problem is a combinatorial optimization problem with numerous applications, and the general form of the problem asks how the value of a knapsack can be maximized, given a knapsack with a maximum limit for the weight, along with a set of items that each have a given weight and value. There are different variations of the problem, the most simple being the 0-1 knapsack problem, where each item can only be included once. The formulation of the 0-1 knapsack problem takes the following form (Kellerer et al., 2004):

$$\begin{aligned} &\max \sum_{i=1}^{n} v_i x_i \\ &\text{s.t.} \sum_{i=1}^{n} w_i x_i \leq W, \\ &x_i \in \{0, 1\}, \end{aligned} \tag{3.1}$$

where n is the number of items available, v_i is the value of item i, x_i is the variable that determines if the item i is included or not (1 if included, 0 if not), w_i is the weight of item i and W is the maximum limit for the weight, i.e. the capacity of the knapsack.

The shelf space allocation problem can be seen as a nonlinear knapsack problem with linear constraints. It is a nonlinear problem because the objective function (the expected quantity of lost sales) is nonlinear, but the shelf space constraint and the possible product-specific minimum quantity constraints are linear. In the problem formulation used in this thesis, the goal is to minimize the expected lost sales instead of maximizing the direct profits, but the idea remains the same.

Thus, the shelf space allocation problem, as described in Section 2.5, can be considered a nonlinear knapsack problem with linear constraints. However, the most relevant characteristics of the problem are the *nonlinear* objective function, the *integer* requirement for the variables and the *linear constraints* defining the feasible set.

3.2 Possible Solution Methods

There are many different methods that have been used to solve knapsack problems and other similar problems in the past. As discussed in Section 2.4, previous approaches especially to the shelf space allocation problem include some simple heuristic methods. In order to reach an optimal or near-optimal solution while maintaining some level of computational efficiency, there are different metaheuristic methods one can use, that have been implemented to similar optimization problems. Metaheuristics is a collective name for methods that are more general and problem-independent than heuristics, and they provide a more thorough approach, thus they are more likely to reach an optimal solution. Lim et al. (2004) present a metaheuristic method for

solving a shelf space allocation problem that is based on the heuristic method by Yang (2001). The method by Lim et al. optimized the search by proposing more complex methods for the neighborhood moves of the algorithm. This method is also mentioned in Section 2.4.

An optimization method for solving an assortment planning problem was presented by Kök and Fisher (2007). The assortment planning problem in their research is nonlinear and discrete in a similar way as the shelf space allocation problem, and Kök and Fisher solved it as a series of nonlinear knapsack problems. The method is an iterative heuristic that models the different substitution effects in the assortment.

Bretthauer and Shetty (2002) mention several possible solution methods to the nonlinear knapsack problem. One of them, the branch-and-bound method, works by solving a series of subproblems separately (see e.g. Bretthauer and Shetty, 1995). This method is for a nonlinear integer problem and it assumes a convex objective function. While the chosen function in this thesis is likely to be convex, because of the constraints and possible future extensions of the function, a method should be chosen that is suitable for non-convex problems as well, so the branch-and-bound method is not suitable in this case. The same problem is found with the method presented by Hochbaum (1995), in which the nonlinear integer knapsack problem is converted into a linear 0-1 knapsack problem.

One method that was considered for solving this shelf space allocation problem is the particle swarm optimization (PSO). It starts by initializing a set of initial candidate solutions, i.e. a swarm of particles, and the position of each particle is then improved based on the best known position of the particle itself and of the whole swarm. As PSO is a metaheuristic, an optimal solution is not guaranteed to be found, but the algorithm is simple to implement and reaches near-optimal solutions. (Kennedy and Eberhart, 1995)

Instead of the particle swarm optimization, a decision was made to implement a version of an optimization algorithm called *simulated annealing* and focus on improving that for solving the shelf space allocation problem presented in this thesis. The advantages of the simulated annealing algorithm are discussed in Section 3.3.

3.3 Simulated Annealing

Simulated annealing (SA) is an optimization algorithm, a metaheuristic, that is considered to have been invented almost simultaneously by Kirkpatrick et al. (1983) and Černý (1985). The method is an analogy of the process of annealing in metallurgy. The technique of annealing involves heating a metal and then cooling it in a controlled fashion in order to improve its structure. The SA algorithm can be used to solve many different kinds of optimization problems, including nonlinear and non-convex problems, since the objective function does not need to be differentiable. The objective function value represents the energy of the system, and the objective is to minimize the energy level. The variables of the optimization problem represent the "state" of the system. There have been various applications of the simulated annealing algorithm in fields such as image processing, molecular biology and chemistry (Eglese, 1990). Simulated annealing has also been applied to the job shop scheduling problem by van Laarhoven et al. (1992).

3.3.1 Algorithm Description

In more detail, the simulated annealing algorithm works as follows: it begins with an initial state s_0, which is the first "current state" s. A new state s_{new} is then generated using some chosen neighbor function, and the current "temperature" of the system is calculated based on how far in the cooling process the algorithm has advanced. The temperature typically describes how close the algorithm is to reaching the maximum number of iterations; in the beginning the temperature is high, and it reaches zero at the end. Next, a probability of acceptance for the new state is generated. The probability is a function of the energy level (objective function value) of the current state s, the energy level of the new state s_{new} and the current temperature T. If the new energy level is lower, then the new state is always accepted (assuming the problem in question is a minimization problem). If the new energy level is higher, the probability of acceptance depends on the temperature; a higher temperature results in more accepted moves. This process is then repeated until the maximum number of iterations $k_{\max}$ is reached, or until some possible other condition for terminating the algorithm is met. A simplified version of the algorithm logic is presented below.

```
# pseudocode for the simulated annealing algorithm
s = s0
for k in 1:kmax
    snew = neighbor(s)
    T = temperature(k)
    if probability(E(s),E(snew),T) >= random(0,1)
        s = snew
output: s
```

A significant advantage of the simulated annealing method compared to iteratively improving methods is that the algorithm is less likely to get stuck, since it is always possible to jump away from a local optimum while the temperature is above zero (Kirkpatrick et al., 1983). In the beginning of the process, at higher temperatures, moves to higher energy levels are more likely to be accepted. Towards the end it is unlikely (but possible) that those moves will be accepted, since at that stage the algorithm is likely to be approaching the global optimum or some value near it. Another advantage of the simulated annealing algorithm is the simplicity of implementation. It is easy to implement once the subfunctions have been determined. It is also adaptable to many different kinds of combinatorial optimization problems. However, in some cases the algorithm can be computationally heavy if all the parts are not constructed carefully. (Eglese, 1990)

The neighbor, temperature and probability functions can be selected in different ways. In this thesis, the **probability function** is only used in its recommended standard form. The probability function exists to determine whether a move from the current state s to a new state s_{new} should be accepted or not. Kirkpatrick et al. (1983) presented the idea of a probability function that accepts all moves that decrease the objective function value (downhill moves), but in the case of moves that increase the result (uphill moves), the probability is determined based on the difference in energy levels as well as the current temperature. The function by Kirkpatrick et al. uses the Boltzmann factor $e^{-\Delta E/(k_B T)}$ as a basis for the probability of acceptance in cases where $\Delta E > 0$ (uphill moves). This has later been simplified to the form $e^{-\Delta E/T}$ (e.g. Eglese, 1990), since T is a control parameter that can be scaled as needed. In conclusion, the probability function is of the following form:

$$P(\Delta E) = \begin{cases} 1, & \text{if } \Delta E < 0 \\ e^{-\Delta E/T}, & \text{otherwise.} \end{cases} \tag{3.2}$$

The **temperature function** is used to generate the current temperature used in the probability function. Its purpose is to be a control parameter that follows a cooling schedule, typically starting from a high number and always ending in zero. It is a relevant part to modify since it controls the cooling schedule, so it is possible to make the algorithm converge towards the optimum faster by choosing the right kind of temperature function. A higher temperature corresponds to a higher probability of acceptance of a new candidate state. This leads to more uphill moves being accepted in the beginning of the process, when it should be more likely to jump out of a local optimum. Correspondingly, at the end of the process, the temperature is low and less uphill moves are accepted, which is favorable as the algorithm approaches the optimal solution. There are different options for choosing the temperature function, and these can be either strictly decreasing or not. However, due to the nature of the simulated annealing algorithm, they still reach zero at the end. Three different cooling schedules were tested in this thesis: the two simpler variations are discussed in more detail in Section 3.3.2 and the more complex variation is presented separately in Section 3.3.3.

An important part of implementing the simulated annealing algorithm is selecting the **neighbor function**. It is used to generate a candidate for the next state that the algorithm moves to, using some kind of stochastic process. In the case of the shelf space allocation problem, the decision can be made to either increase or decrease the number of facings of a product, as long as the resulting state stays within the constraints of the problem setup. Another choice to be made in the neighbor function is either modifying the number of facings of multiple products at once or just one at a time, as well as choosing which one(s) to modify. In addition, the step size of the increase or decrease in facings can be chosen, and it can be static or dynamic. These choices can all be determined randomly, or they can be defined experimentally based on some rules (see Section 3.3.2 for more information).

The constraints of the optimization problem can be considered in different ways in the simulated annealing algorithm, either by simply having the neighbor function only generate candidates that are within the constraints, or by introducing some penalty to the cost function that would make it extremely unlikely to end up choosing candidates outside the boundaries. For this thesis, it was decided that a penalty function would be unnecessarily complicated, and just limiting the possible candidates from the beginning would be the best choice in this case. This was also supported by Zhang and Wang (1993).

3.3.2 Test Setup

There are many possibilities to vary the simulated annealing algorithm in order to reach optimal results for different kinds of optimization problems. For this thesis, it was decided to test out different options for the neighbor function as well as the temperature function. The probability function is only used in its standard form, as presented in Equation (3.2).

Regarding the neighbor function, the choice of modifying multiple facings or just one was selected to be constant for all the tests; due to the shelf space allocation being a constrained problem, it was decided that modifying only one facing at a time was the best option. This was based on the research by Zhang and Wang (1993), and the method in question is called the orthogonal move approach. The choice of increasing or decreasing the number of facings and the selection of the product to modify were chosen to be random in all the experiments. For the step size, different methods are tested to find optimal performance of the algorithm (see Chapter 4, Results). First a simple neighbor function, where the step size is 1 throughout the algorithm, is tested and evaluated. In every iteration, one of the products is randomly selected to have the number of facings either increase or decrease by 1. This new state then becomes the candidate state for the next move if it complies with all the constraints of the optimization problem. If it does not comply, then the process is repeated until a valid candidate state is found.

In addition to the neighbor function with the constant step size, a more dynamic approach to the step size is implemented. A suitable method to test is found in the article by Zhang and Wang (1993). The basic principle is to observe the ratio between accepted and rejected moves in the probability function and try to keep it close to 1 by modifying the step size. If there are too many accepted moves, the step size is increased and vice versa; the step size is decreased if the number of accepted moves decreases significantly. In practice, this is done by starting with a large initial value for the step size, generated based on the shelf size, product width and number of different products. This is then adjusted after every iteration based on the ratio of accepted versus rejected moves of the last 100 iterations. The moving time window of 100 iterations was chosen as opposed to taking the ratio of the all time history, since this way the impact of a single iteration stays constant throughout the algorithm, even if the total number of iterations would be 10^5. The step size is increased by 1 if the share of accepted moves is over 0.7, and decreased by 1 if the share is under 0.3. However, in the case of an increase in step size, the adjustment is only done if the restrictions of the

shelf allow it, i.e. if it is possible to use that step size for at least one of the products while still being within the problem constraints. In the case of a decrease in step size, the minimum limit for the step size value is 1.

In addition to the different versions of the neighbor function, some variations of the temperature function are also implemented. The first and most simple form to test out for the temperature setup is a linear cooling schedule:

$$T_k = T_0 \cdot \left(1 - \frac{k}{k_{\max}}\right). \tag{3.3}$$

Here T_0 is the initial temperature, k is the current iteration count and $k_{\max}$ is the maximum number of iterations.

The second version of the temperature function to be implemented in this thesis is a logarithmic temperature function:

$$T_k = \frac{c}{\ln(k+1)}. \tag{3.4}$$

This approach was introduced by Geman and Geman (1984) and again presented by Nourani and Andresen (1998). Nourani and Andresen came to the conclusion that the logarithmic cooling schedule was not to be recommended for the simulated annealing algorithm, but it is nevertheless included in the analysis in this thesis for comparison. Different values of the parameter c are tested in order to find the most suitable one for this particular problem.

Lastly, a temperature function based on the thermodynamic simulated annealing (TSA) approach by de Vicente et al. (2003) was tested. A combined method using the linear cooling schedule together with the TSA temperature function was also included in the study. These methods are presented in detail in Section 3.3.3.

3.3.3 Thermodynamic Simulated Annealing

de Vicente et al. (2003) introduced a new variation to the simulated annealing algorithm called thermodynamic simulated annealing (TSA). It provides a new way to configure the cooling schedule of the algorithm using the logic of thermodynamic laws, the goal being to improve the performance of the SA algorithm. The article by de Vicente et al. is used as a reference for this entire section.

The cooling schedule, or how quickly the temperature is lowered, is an important part in the performance of the algorithm, especially for avoiding getting stuck in local minima. In the basic version of simulated annealing described in Section 3.3.1 the cooling schedule is linear; the temperature function starts from a value t_0 and decreases by an equal amount with every iteration, until reaching zero at the last iteration. This is a valid option, however, in many cases the algorithm reaches the optimum (or near-optimum) faster if the temperature function is modified experimentally during the execution of the SA algorithm.

Thermodynamic simulated annealing uses a specific temperature function where the temperature is adjusted according to the progress of the algorithm up until each iteration. The temperature is not restricted to be strictly decreasing in the TSA algorithm, instead it can move freely up and down based on thermodynamic laws. If there are two states A and B, the temperature after the transformation from A to B can be presented as:

$$T = \frac{E_B - E_A}{H_B - H_A}, \tag{3.5}$$

where E_A and E_B are the energy levels, i.e. the values of the cost function, at the corresponding states, and H_A and H_B are the levels of entropy at the states A and B.

In information theory, the difference in information when receiving a message with a probability P is defined as:

$$\Delta I = -\ln P. \tag{3.6}$$

The TSA algorithm by de Vicente et al. (2003) uses this relation to represent the variation in entropy ΔH when the transition has a probability P:

$$\Delta H = \ln P. \tag{3.7}$$

The transformation from A to B can then be analyzed as a series of k transformations, with P_i as the probability of step i being accepted. T_i is the temperature at the transformation and ΔC_i is the difference in the energy levels before and after the transformation. The difference in entropy between states A and B can then be presented as in Equation (3.8), where there are k steps in the transformation from A to B. The same equation can be expressed as a sum, as seen in Equation (3.9):

$$\begin{aligned}\Delta H_{AB} &= \ln\left(P_1 \cdot P_2 \cdot ... \cdot P_k\right) \\ &= \ln P_1 + \ln P_2 + ... + \ln P_k;\end{aligned} \tag{3.8}$$

$$\Delta H_{AB} = \sum_{i=1}^{k} \ln P_i. \tag{3.9}$$

In the simulated annealing algorithm, the probability function, in the case of uphill moves is $e^{-\Delta E/T}$; with the acceptance probability of downhill moves being 1, and thus the corresponding logarithms being zero. Equation (3.9) can then be formulated using the probability function in Equation (3.2) as:

$$\begin{aligned}\Delta H_{AB} &= \sum_{i \in M_+^k} \ln e^{-\Delta E_i/T_i} \\ &= -\sum_{i \in M_+^k} \frac{\Delta E_i}{T_i}.\end{aligned} \tag{3.10}$$

Here M_+^k is the set of candidate moves that increase the energy level, i.e. moves where $\Delta E > 0$, until iteration k. Both accepted and rejected moves are included.

Regarding the variation in the energy level $E_B - E_A$, or ΔE, that is found in Equation (3.5), it can also be expressed as a sum of individual moves. When all accepted moves until iteration k are denoted as M_{accepted}^k, the energy level difference between A and B can be formulated as:

$$\Delta E_{AB} = \sum_{i \in M_{\text{accepted}}^k} \Delta E_i. \tag{3.11}$$

When inserting Equation (3.10) and Equation (3.11) into Equation (3.5), the result is the temperature after all the moves up until the kth iteration, i.e. the temperature at iteration $k+1$. Following the process by de Vicente et al. (2003), a control parameter k_A is introduced, which can be used to adjust the temperature function for different problem setups. The resulting function is:

$$T_{k+1} = -k_A \frac{\sum_{i \in M^k_{\text{accepted}}} \Delta E_i}{\sum_{i \in M^k_+} \Delta E_i / T_i}. \tag{3.12}$$

In order to avoid negative values for the temperature, as well as dividing by zero, the temperature is set to the initial value T_0 if $\sum_{i \in M^k_{\text{accepted}}} \Delta E_i$ is positive or if $\sum_{i \in M^k_+} \frac{\Delta E_i}{T_i}$ is equal to zero. Thus the final form of the temperature function is:

$$T_{k+1} = \begin{cases} T_0, & \text{if } \sum_{i \in M^k_{\text{accepted}}} \Delta E_i \geq 0 \text{ or } \sum_{i \in M^k_+} \frac{\Delta E_i}{T_i} = 0, \\ & \\ -k_A \frac{\sum_{i \in M^k_{\text{accepted}}} \Delta E_i}{\sum_{i \in M^k_+} \Delta E_i / T_i}, & \text{otherwise.} \end{cases} \tag{3.13}$$

Another variation of the TSA cooling schedule is also implemented in this work. It was found during the testing that the temperature seemed to drop quite quickly when using the TSA temperature function, so a new approach was introduced in an attempt to slow down the cooling. It is a combination of the temperature function in Equation (3.13) and the linear temperature function in Equation (3.3). It works as the TSA temperature function, but with the linear function as a lower limit:

$$T_{k+1} = \begin{cases} T_0, & \text{if } \sum_{i \in M^k_{\text{accepted}}} \Delta E_i \geq 0 \text{ or } \sum_{i \in M^k_+} \frac{\Delta E_i}{T_i} = 0, \\ & \\ \max \left[-k_A \frac{\sum_{i \in M^k_{\text{accepted}}} \Delta E_i}{\sum_{i \in M^k_+} \Delta E_i / T_i}, \quad T_0 \Big(1 - \frac{k}{k_{\max}} \Big) \right], & \text{otherwise.} \end{cases} \tag{3.14}$$

Chapter 4

Results

A summary of all the variations of simulated annealing that are tested in this thesis is presented in Table 4.1. In this chapter, the results of all those tests are presented. The results are grouped into sections based on the variations that were tested.

For all test cases, the convergence of the algorithm is investigated by running the tests multiple times for different values of $k_{\max}$. For each chosen value of $k_{\max}$, the test is run from the start, since the temperature function depends on the $k_{\max}$ value, and thus it is not possible to simply run the algorithm once with a large $k_{\max}$ while saving intermediate results for different values of k. The chosen $k_{\max}$ values range from 10^2 to 10^5 with different intermediate steps in between. The objective function value (the lost sales quantity) is plotted as a function of $k_{\max}$, and these plots illustrate if and how the different variations of the SA algorithm converge towards some optimum when the number of iterations is increased. In some test cases, some additional information is also presented in the plots.

Because the performance of the algorithm depends on stochastic components to some extent, the experiments are conducted multiple times with different seed numbers for the probabilistic functions. The code uses a list of 10 predetermined seed numbers, runs each test 10 times with different seed numbers and finally computes the average value of the objective function of those 10 results. This is then the final result. Some of the figures in this chapter have error bars included in the plots. The error bars show the equivalent of one standard deviation in either direction based on the 10 tests with different seed numbers. Some additional information that is received in the experiments, such as the temperature as a function of the iteration k, is

not averaged since in those cases it is important to look at the development during a single run, not the averages. In those cases the first seed number in the list is used.

The data used for the SA algorithm tests comes from a European grocery retailer. The data contains a number of products and the following information about those products: product group, average daily sales and standard deviation of sales. All of the products are from the cereals category.

Three product groups were chosen for testing, containing 5, 10 and 20 products respectively. Each of these three groups was tested with three different values for the shelf width[1] S, making nine test sets in total. The tested combinations were: 5 products and $S = 25$, 50 and 100; 10 products and $S = 50$, 100 and 200; and 20 products and $S = 100$, 200 and 400. The product width was set as 3.7 for each product, and the maximum days to the next delivery was 5 for all products. All the product groups consist of products with average daily sales of the same order of magnitude, so there are no significant differences between the different product groups, except the number of products in each group (5, 10 or 20).

In each section below, some example cases of the results are presented in more detail, even though all of the tests were conducted for all 9 test cases mentioned above. The rest of test results can be found in Appendix A. There the summary plot is included for all test cases, and for the rest, selected plots are presented.

4.1 Linear Cooling Schedule

The first tests were run using the simple neighbor function with step size 1, randomized direction (add or remove facing) and product selection. The temperature function was linear in the first tests, and different initial values for the temperature were tested. The difference in energy ΔE that is used in the calculations has a value of around 0.1-0.2 units for the first iteration round in these tests. This set of tests was executed for multiple different values of T_0 in order to find the optimal setting. The results of the comparison are presented in Figure 4.1 for one of the shelf setups. A lower value of $T_0 = 0.01$ was also included in the tests, but the lost sales results were in most cases not converging at all, so it was excluded from the plot for clarity. Based on the

[1]The widths of the products and the shelf could be considered as centimeters, so $S = 200$ would be 2 m of shelf space.

Table 4.1: List of test setups

Neighbor	Temperature	T_0	k_A	c
step size 1	linear	0.01		
step size 1	linear	0.05		
step size 1	linear	0.1		
step size 1	linear	0.3		
step size 1	linear	0.5		
step size 1	linear	1		
dynamic	linear	1		
step size 1	logarithmic	0.1		0.1
step size 1	logarithmic	0.1		0.5
step size 1	logarithmic	0.1		1
step size 1	logarithmic	0.1		10
step size 1	TSA	1	0.1	
step size 1	TSA	5	1	
step size 1	TSA	10	1	
step size 1	TSA	20	1	
step size 1	TSA-linear	0.1	0.1	
step size 1	TSA-linear	0.1	0.5	
step size 1	TSA-linear	0.1	0.9	
step size 1	TSA-linear	0.1	1	
step size 1	TSA-linear	0.1	1.1	
step size 1	TSA-linear	0.1	2	

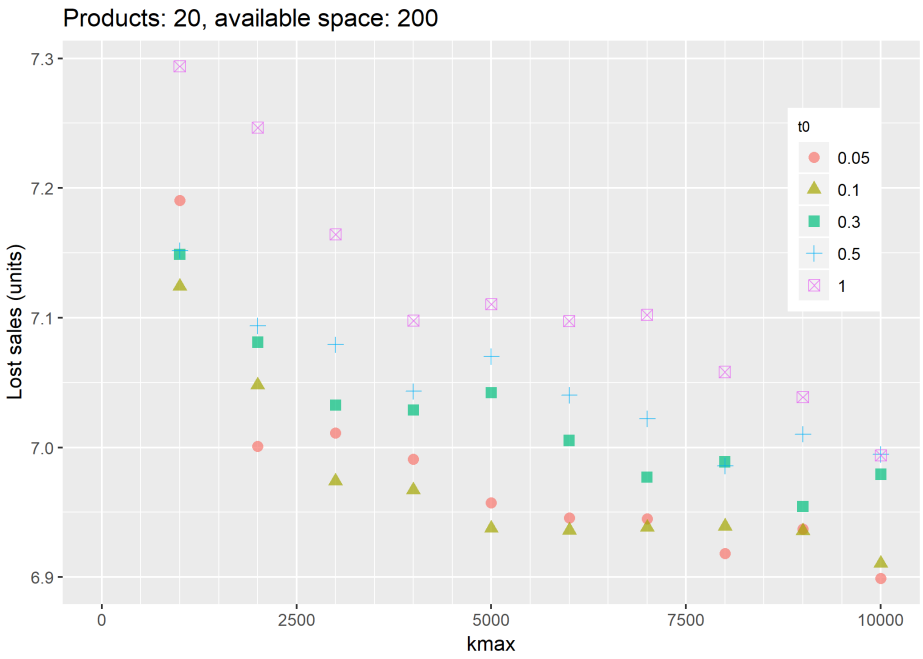

Figure 4.1: Lost sales results using the linear cooling schedule, the static step size 1 and different values for the initial temperature T_0.

result in Figure 4.1 as well as the other tests presented in Appendix A, the value of $T_0 = 0.1$ was found to be optimal for this variation of the simulated annealing algorithm.

As an example of the final allocation, the minimum lost sales in Figure 4.1 is obtained with $T_0 = 0.05$, and in the plot the lost sales is 6.90 units. Since this is an average of 10 tested seed numbers, there is no allocation of facings that corresponds directly to that. The best result of the 10 test in this case was 6.87 units of lost sales, and the corresponding allocation of facings was the following: 6, 1, 1, 4, 2, 4, 2, 1, 4, 3, 2, 4, 4, 4, 3, 3, 1, 1, 3 and 1 (20 products). The amount of space this set of products occupied was 199.8 (out of 200).

4.2 Dynamic Neighbor Function

The results of the dynamic step size tests are found in Figure 4.2, with the lost sales as a function of $k_{\max}$. The initial value for the temperature was set to $T_0 = 1$ for these tests. The variant with the static step size 1 is included

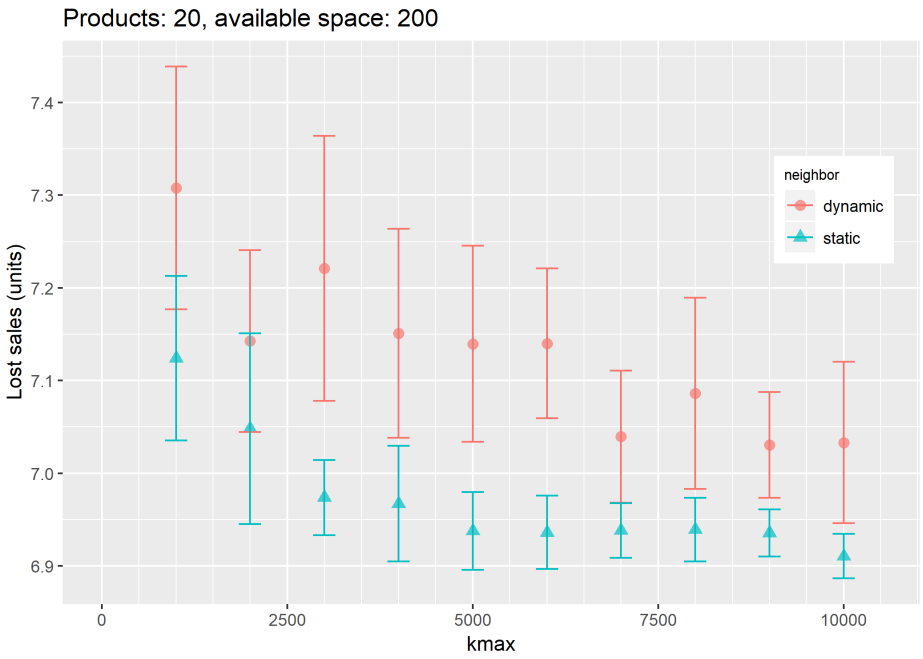

Figure 4.2: Comparison of the lost sales results using the dynamic and static neighbor functions.

in the plot for comparison, also with $T_0 = 1$. Here Figure 4.2 shows that the dynamic neighborhood function converges significantly slower than the method with the static step size. This was the case in most of the tests conducted, and the dynamic function did not produce improved results in any of the test cases. The uncertainty in the results in Figure 4.2 is quite high, but when all the results are taken into consideration, it is still clear that the dynamic method does not provide an improvement to the simulated annealing method.

In Figure 4.3 the development of the step size in one example case is shown, as well as the share of accepted moves out of all candidate moves in the probability function, as a function of the iteration k. The data points have been taken every 100 iterations.

Figure 4.4 shows the duration in seconds for the tests presented in Figure 4.2. As can be seen in Figure 4.4, the more complex dynamic method is more computationally heavy than the simple method with the step size of 1. Because of this and the comparison of convergence in Figure 4.2, the following tests after this were chosen to only use the static neighbor function with the step size 1.

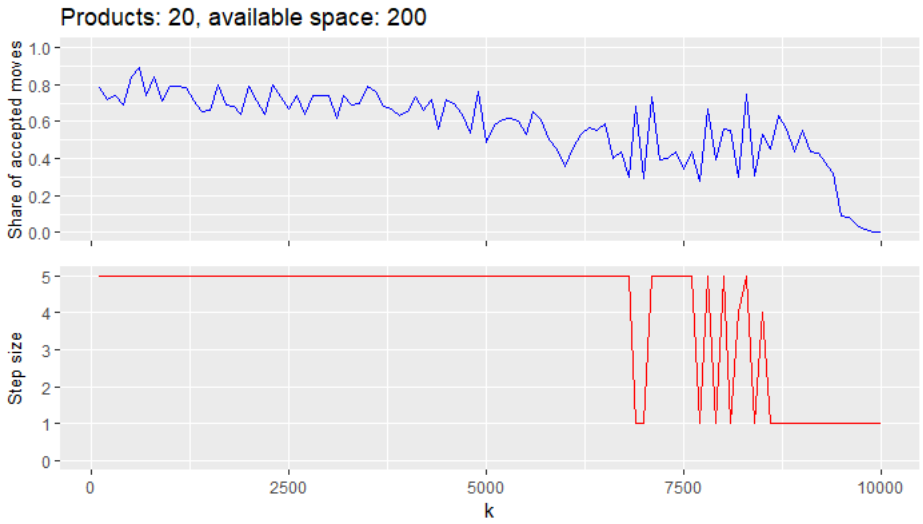

Figure 4.3: The share of accepted moves and the step size during the SA algorithm with the dynamic neighbor function.

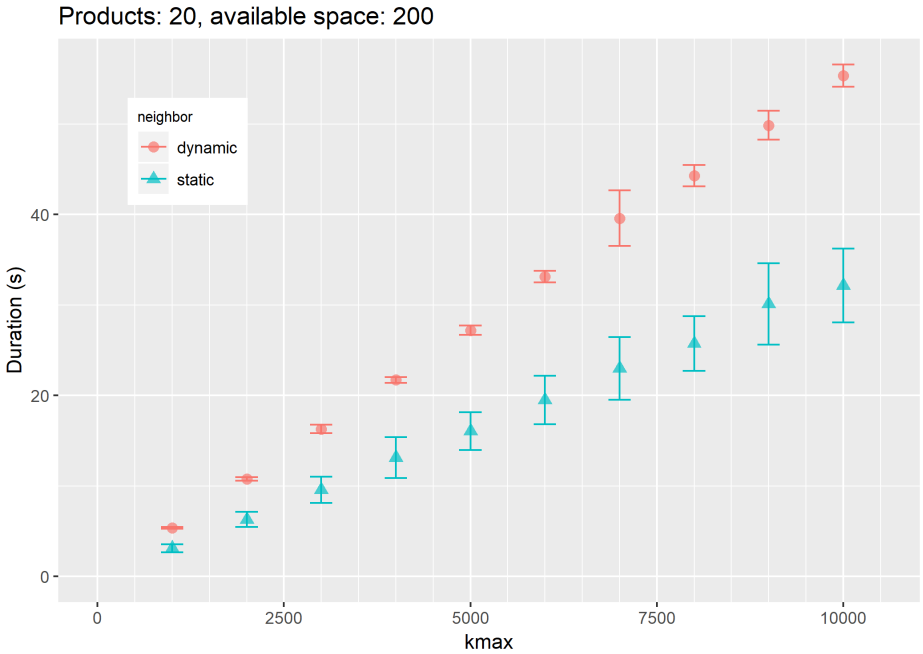

Figure 4.4: Comparison between the durations of the static and the dynamic neighbor functions (using mean and standard deviation of 10 different seeds).

4.3 Logarithmic Cooling Schedule

After the decision was made to keep the neighbor function as the simple version with the static step size, the temperature function was modified. The goal of this new cooling schedule would be to improve the performance of the SA algorithm by finding the optimum, or a value near the optimum, faster. First, the logarithmic temperature function was implemented. The approach is presented in more detail in Section 3.3.2 and Equation (3.4). The results of two of the test sets with the SA algorithm using the logarithmic cooling schedule are found in Figure 4.5 and Figure 4.6. The tests were run with different values for the parameter c, and the linear cooling schedule results are included in the plots for comparison.

In Figure 4.5 it seems that the optimal value is $c = 0.1$, while in Figure 4.6 that particular parameter value gives far from optimal results. It seems, based on all the test sets for the logarithmic method found in Appendix A, that for problem setups with more shelf space per unique product, the low values for the parameter c resulted in lower lost sales. On the other hand, the problems with little shelf space per unique product, the higher ones (0.5 and 1) gave lower results for the lost sales. When considering all of the different test cases, the optimal parameter value was determined to be $c = 0.5$, since it gave consistently either the lowest or second lowest lost sales results for all the setups (not including the linear cooling schedule).

The linear cooling schedule results in lower lost sales than the logarithmic method with $c = 0.5$ in Figure 4.5. With the parameter value $c = 0.1$, however, the lost sales converge faster than with the linear cooling schedule in that particular case.

4.4 TSA Cooling Schedule

Besides the logarithmic cooling schedule, another option for improving the simulated annealing algorithm was the thermodynamic simulated annealing (TSA) method introduced in Section 3.3.3. The TSA algorithm was run until $k_{\max}$ iterations each time, so no additional stopping criteria were introduced. The initial value for the temperature T_0 was set to different values (1, 5, 10 and 20) for the TSA tests. The results are presented in Figure 4.7, and the original linear cooling schedule is included for comparison.

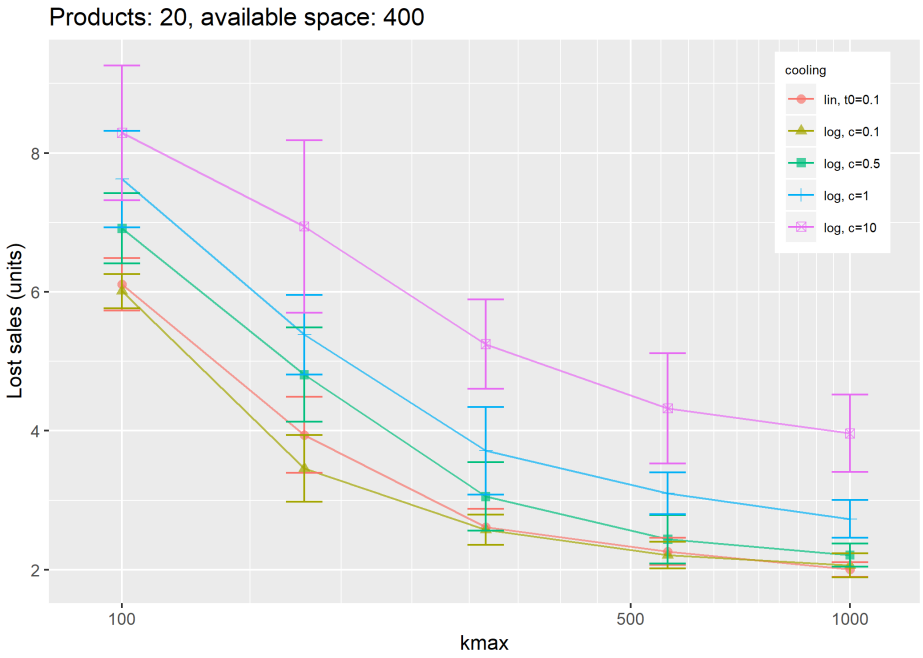

Figure 4.5: Comparison of the lost sales using the logarithmic and linear cooling schedules with different values for c ($\log_{10}$ scale).

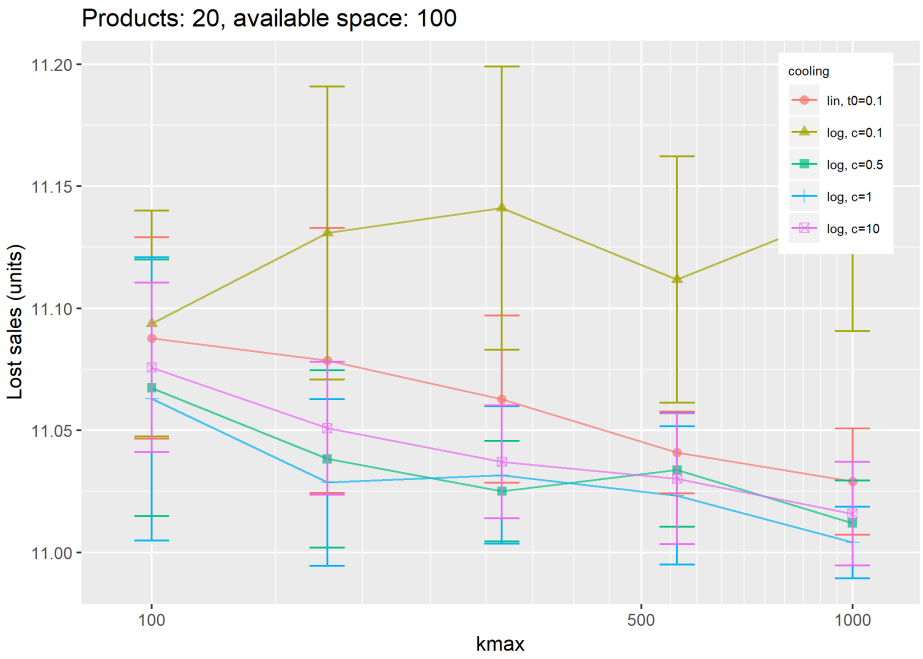

Figure 4.6: Comparison of the lost sales using the logarithmic and linear cooling schedules with different values for c ($\log_{10}$ scale).

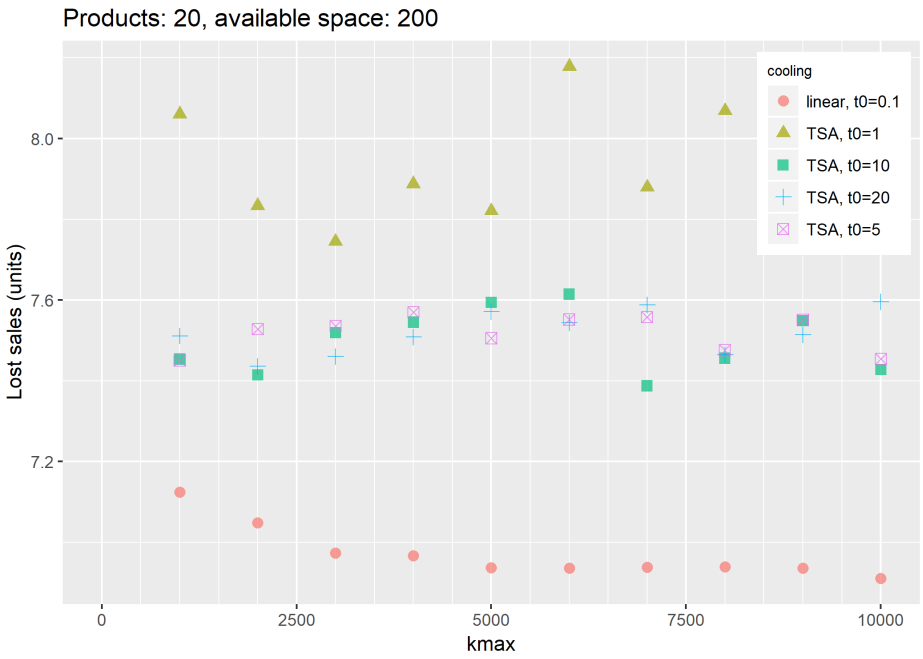

Figure 4.7: Comparison of the lost sales results using the TSA and linear cooling schedules, with different values for the initial temperature T_0.

In Figure 4.7 the results show that the objective function does not converge towards any value, at least within the given range of $k_{\max}$. In the test case with 20 products and $S = 400$ in Figure 4.8, the TSA results do converge, but it is the only case where that happens.

In Figure 4.9 the temperature is plotted as a function of the iteration k for one of the test rounds with $T_0 = 10$. Here one can see that when using the TSA method, the temperature drops to near zero quite early in the process. The k scale in Figure 4.9 is logarithmic, since most of the variation in the temperature happens for low values of k. Because of the drop in the temperature, the TSA cooling schedule was modified for the next tests (see Section 4.5).

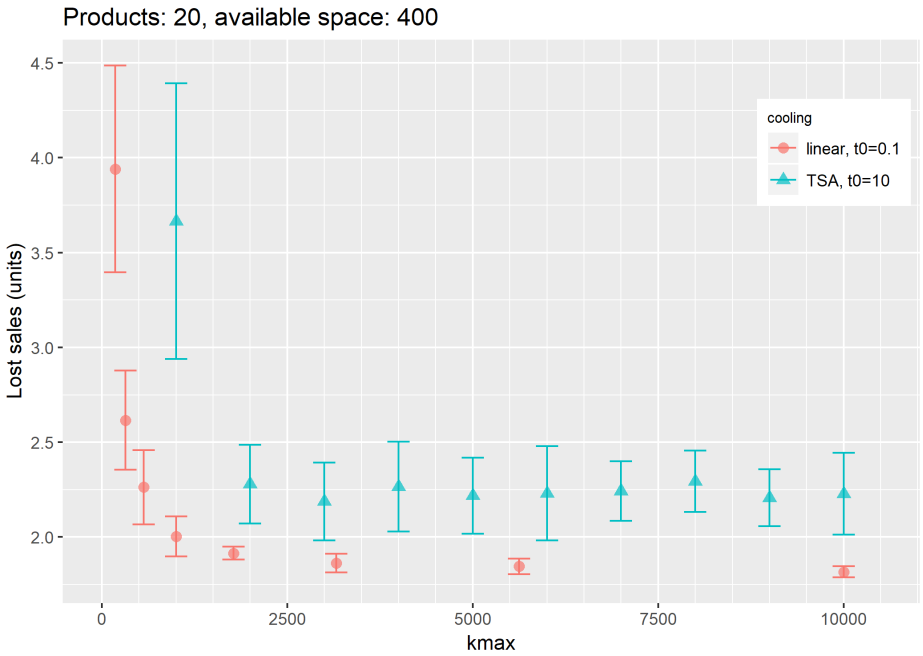

Figure 4.8: Comparison of the lost sales results using the TSA and linear cooling schedules.

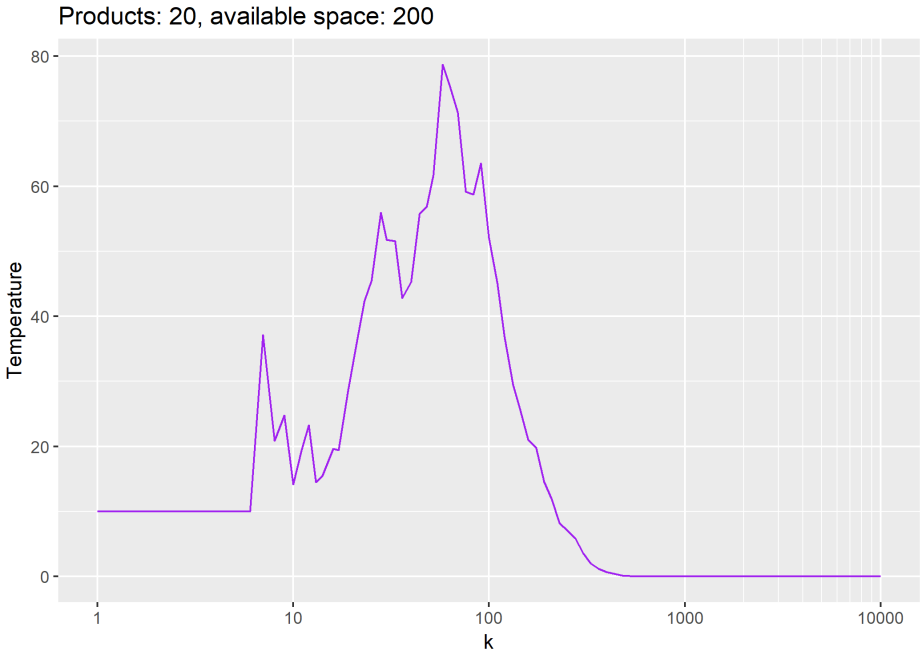

Figure 4.9: The temperature of the TSA algorithm as a function of the iteration k ($\log_{10}$ scale).

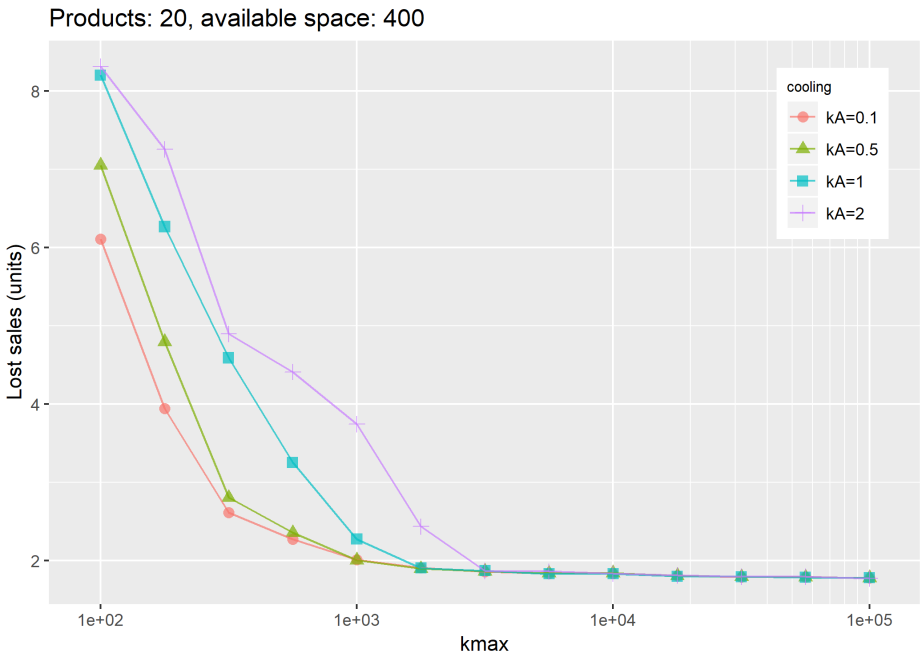

Figure 4.10: Lost sales results using the TSA-linear combination cooling schedule with different values for the control parameter k_A ($\log_{10}$ scale).

4.5 Adapted TSA Cooling Schedule

The adapted TSA cooling schedule is otherwise similar to the TSA cooling schedule, but the lower limit for the temperature is set as the linear temperature function value. This is also presented in Section 3.3.3 and Equation (3.14). The results of the combined TSA-linear temperature function are presented in Figure 4.10. Different values for the parameter k_A were tested, and the optimal value for this problem seemed to be $k_A = 0.1$ when taking into consideration all of the test cases. The tests were done using the initial temperature $T_0 = 0.1$, which was the selected value based on the results in Section 4.1.

However, when the initial temperature is $T_0 = 0.1$ and the parameter $k_A = 0.1$, the combined TSA-linear cooling schedule uses the linear cooling schedule almost exclusively. This happens when the temperature of the TSA function would be lower than the linear function value, and in the tests only some of the cases contained small individual spikes above the linear function line. As a consequence, the lost sales results of the adapted TSA cooling schedule are nearly identical with the linear results presented in Section 4.1.

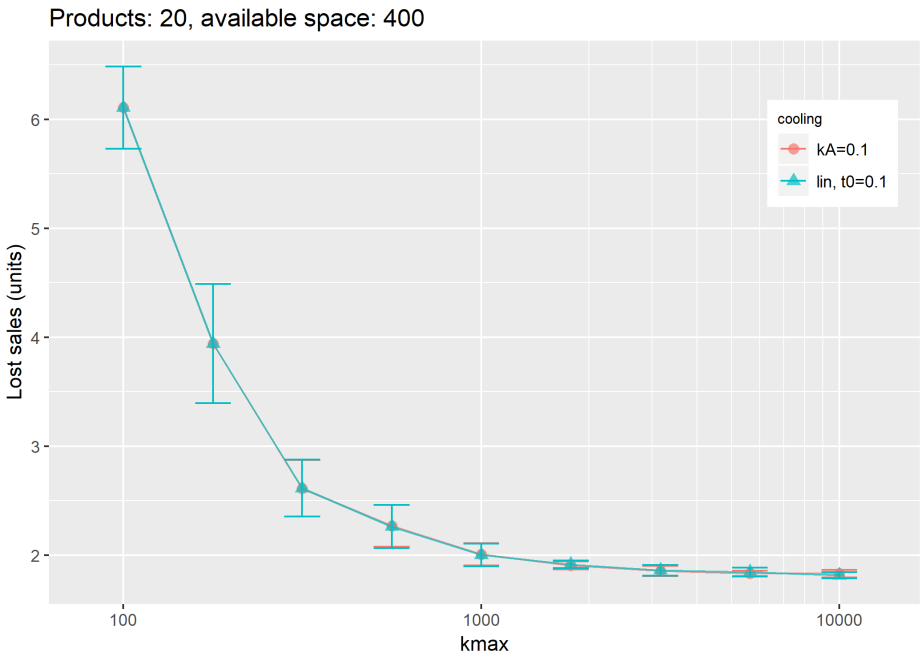

Figure 4.11: Comparison of the lost sales results using the TSA-linear combination cooling schedule and the linear cooling schedule ($\log_{10}$ scale).

Because of this result, it was decided to not extend the study further on this method. In Figure 4.11 the TSA-linear combination method is compared to the results of the linear cooling schedule from Section 4.1, and as expected, the results are almost identical.

The temperature T as a function of the iteration k is presented in Figure 4.12 for one of the tests using the TSA-linear combination cooling schedule with $k_{\max} = 10\ 000$. The horizontal axis scale is logarithmic, so the curved line in the plot is the linear temperature function. As mentioned previously, the temperature stays on the linear curve at nearly every iteration k.

4.6 Summary

In Figure 4.13 all the different versions of the simulated annealing algorithm are compared until $k_{\max} = 10\ 000$. The variations are, in the order of the plot: the dynamic neighbor function with $T_0 = 1$, the linear cooling schedule (and static step size) with $T_0 = 0.1$, the logarithmic cooling schedule with $T_0 = 0.1$ and $c = 0.5$, the adapted TSA-linear cooling schedule with $T_0 = 0.1$ and $k_A = 0.1$, and the TSA cooling schedule with $T_0 = 10$ and $k_A = 1$.

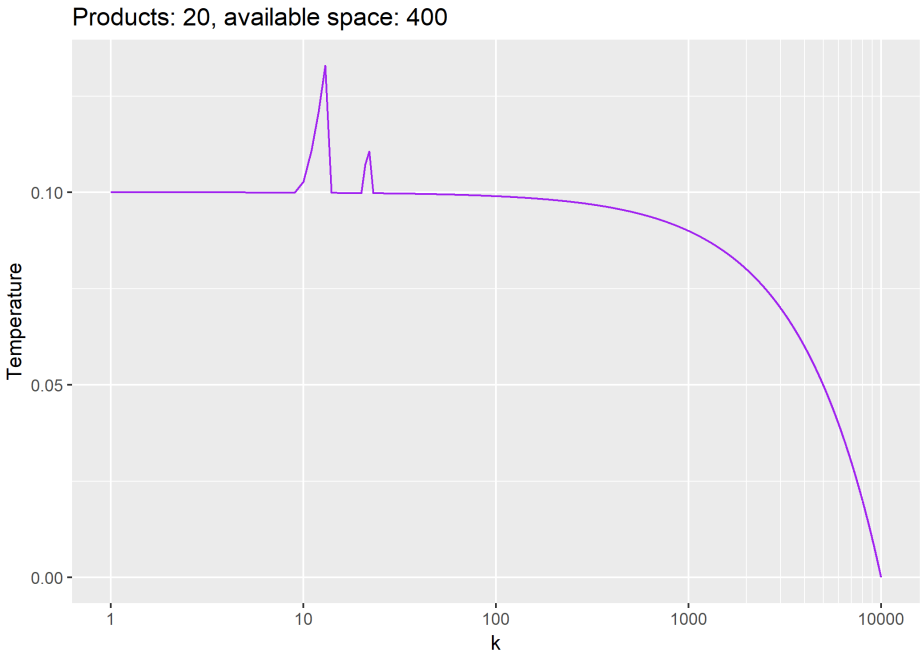

Figure 4.12: The temperature of the combined TSA-linear cooling schedule as a function of the iteration k ($\log_{10}$ scale).

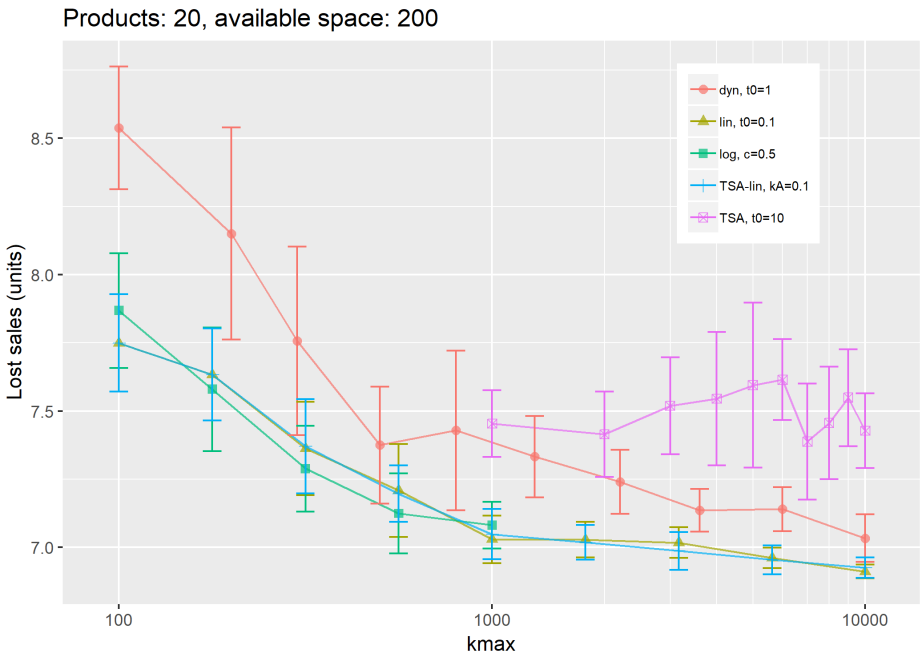

Figure 4.13: Comparison of all the different versions of the SA algorithm that were tested ($\log_{10}$ scale).

Chapter 5

Conclusions

In this section, the key findings of the research are summarized. Then the validity and reliability of the results are evaluated, and finally some suggestions for possible future research are presented.

5.1 Key Findings

The goal of this thesis was to formulate the shelf space allocation problem as an optimization problem and find a suitable method to solve it. Minimizing lost sales by optimizing the allocation of shelf space means there is less need for restocking the shelves as well as a higher availability level, which in turn can lead to savings for the retailer. The objective function to be minimized was formulated as the expected quantity of lost sales.

The selected method for solving the shelf space allocation problem was the simulated annealing (SA) algorithm. It was chosen for its simplicity and ability to jump out of local minima, which was suitable for the combination of a nonlinear objective function, the integer requirement and the restriction of shelf space. Different versions of the simulated annealing algorithm were tested, and the end value of the objective function was recorded for varying iteration counts in order to see if and how fast the algorithm converged. The subfunctions of the SA algorithm that were varied were the neighbor function and the temperature function.

The simple, static neighbor function with the step size 1 proved to give stable results for all of the tested data sets. The dynamic version of the neighbor

function did not improve the results for this type of a problem, as is shown for example in Figure 4.2.

The most relevant part of the investigation became the comparison between the results of the different temperature functions, or cooling schedules. The linear cooling schedule converged well for all of the tested cases, and it is the simplest one of the temperature functions to implement. It is important to note that the selection of the initial temperature T_0 had a great impact on the performance of the optimization algorithm. Varying the value of T_0 gave different results, as can be seen in Figure 4.1. The probability function in the SA algorithm uses the ratio of the jump in energy ΔE (the difference in the objective function value) to the temperature T in deciding whether or not to move to a new candidate state. The difference in energy between the initial state and the first candidate state was around 0.1-0.2 for these tests, and the best value for T_0 was 0.1. Based on this, it can be seen that for this type of a problem, values for T_0 that are nearly equal to the initial energy difference ΔE are to be preferred, since this seems to produce more optimal solutions on average.

A logarithmic function was implemented for the cooling schedule. It was fairly simple to implement, and the results were quite promising. The adjustment of the control parameter c in the logarithmic temperature function proved to be significant for the convergence of the lost sales. If the parameter c is chosen carefully, the lost sales can converge faster with the logarithmic cooling schedule than with the simple linear function. Figure 4.5 shows an example of this, with $c = 0.1$. When looking at all the test results of the SA algorithm with the logarithmic cooling schedule, one finding was that the optimal choice of the parameter c can depend on the amount of available shelf space in relation to the number of unique products on that shelf. More specifically, the number of unique products does not necessarily matter for the calculations, but it is the need to cover more demand that changes the situation. All of the products in the test data had average daily sales of roughly equal volume, and the item widths were also set as identical, so 10 unique products will need more space than 5 unique products in order to cover the demand for the same number of days. Based on the results, lower values of the parameter c (e.g. 0.1) result in lower lost sales and faster convergence in cases with a high amount of space per product, and higher values of c (e.g. 1) work better in situations with less space per product, at least in situations similar to these test cases.

The thermodynamic simulated annealing (TSA) cooling schedule was implemented and tested extensively, but it did not improve the results. In fact,

the lost sales did not even converge towards any value in most of the tests. The combined TSA-linear cooling schedule did converge, however, the best results were obtained when the temperature function only used the linear schedule. Hence it can be seen that for this kind of a problem formulation, the thermodynamic simulated annealing does not offer improvements compared to the simple linear cooling schedule, especially since the linear cooling schedule is notably simpler to implement.

Overall, the linear temperature function works well in all of the test cases. The TSA method did not show improvements to the lost sales results, neither on its own nor as a combined method with the linear function. The logarithmic temperature function showed good results when the parameter c was chosen carefully. The choice of the initial temperature T_0 had a significant impact in all of the tested methods.

5.2 Discussion

This study was conducted in order to find a method for solving the shelf space allocation problem in a way that considers replenishment in the process and lowers costs for the retailer. The results show that the simulated annealing method is a good option for solving the shelf space allocation problem where the goal is to minimize the expected lost sales. When discussing the validity of the results, it is important to note that some assumptions were made in this study. The sales were assumed to be normally distributed, but the problem formulation can easily be modified to use another distribution instead. Some type of distribution is in any case required to be assumed in this setup. There is no assumption regarding the existence of a backroom storage in the store, since the minimization of lost sales leads to benefits for the retailer in both cases. If there is a backroom, the optimized allocation reduces the need for moving stock from the storage to the shelf, as well as the size of the total inventory in the backroom. If the full delivery is placed directly on the shelf, then the optimization can lead to a need for less frequent deliveries, which also applies if there is a backroom. In both cases, the revenue increases for the retailer if the amount of lost sales is reduced. Related to this, another assumption of this study is that if the stock on the shelf is not enough to satisfy the demand for the day, then the unsatisfied demand becomes the lost sales for that day. If we assume that the sales follow the normal distribution, then that is the logical conclusion that follows.

The lost sales formula used sales quantity (units) instead of sales value (euros or other currency) as the metric in the calculations. When considering the possible shelf stacking work needed in the case of a stock-out on the shelf, the costs that incur are dependent on the quantity of products needed, or more specifically, the product size also affects the time it takes to move and stack the products. This also applies in the case of a reduced need for deliveries, although in that case the delivery costs are relevant, too. On the other hand, for minimizing the loss in sales that occur when there is a stock-out, the sales value is perhaps the more relevant metric to use. With that in mind, the objective function could be modified to include the lost sales value in addition to the quantity, but the general idea of the optimization problem remains the same. Regardless, the lost sales quantity provides a good enough approximation for the purposes of this thesis, since minimizing the restocking labor is a priority.

In the test data, all products were set to be the same width. The algorithm was built to handle different product widths, but those were not tested in this thesis. The delivery schedules, i.e. the number of days to the next delivery that is used in the calculations, was also set as equal for all of the products. In addition, the model did not include the shelf depth as a dimension, but assumed for simplicity that only the visible facings were available on the shelf, whereas typically there is room for multiple items in the space that is occupied by one facing. These aspects were not considered to heavily change the outcome of the results, however, this was not studied. The lack of these variations in the test setup is one of the limitations of this study.

This study uses a somewhat simplified model of the shelf space allocation problem. Especially having only one shelf in the model instead of dividing the space into several shelves is an unrealistic situation in most cases. The choice to do this was justified by the added complexity that dividing the space into multiple shelves would entail. If there are multiple shelves, it is by consequence necessary to decide if one product can have facings on only one of the shelves, or two, or all of them. Then questions regarding the placement and grouping of the different products will appear, since placing facings of a product on two different shelves means having to know where the products are placed (so the chosen shelves are one top of the other, and the facings are placed adjacently). If no restrictions are placed on the number of shelves a product can be on, the end result of the optimization may be a series of identical shelves with the same set of products each. All these aspects of placement and grouping are quite complex to take into account, therefore a simplified model was chosen. It is important to note that the solution algorithm has not been tested with a comprehensive shelf space allocation

model including these different components, however, the results of this study remain valid as a basis for future research.

The goal for this study is lowering operating costs for retailers by optimizing the shelf space allocation. Shelf space planning can also be utilized to actively increase sales by taking advantage of the concept of space elasticity. Another factor that can impact sales is the cross-space elasticity, which describes the different substitution patterns that occur (typically) within product groups. These effects all play a part in the complex retail shelf space picture, but for the sake of this study, it was considered most important to focus on lowering the lost sales.

The results of this study can be utilized when approaching other shelf space allocation problems with similar specifications, where the required assumptions are valid. This study is mostly focused on the field of grocery retail, since grocery volumes are large and shelf space planning typically plays a larger role there than in other fields of retail. Spoilage problems also occur much more in the grocery field. However, the results are applicable to any type of retail store, as long as the required assumptions are fulfilled.

The utilization of the simulated annealing algorithm for the shelf space allocation problem is a valid possibility for similar situations as the ones described in this study. There are variations of the algorithm that are suitable for different situations, and by optimizing the values of the different parameters one can improve the results further. However, this study is only done on a small set of test data, with some specific assumptions, so it is important to note that further research is still needed before using these results in real-life applications.

5.3 Future Research

The results of this study showed that the simulated annealing algorithm is a useful method for solving one version of the shelf space allocation problem. In order for the results to be useful in practical applications, some testing is needed on the areas that were not covered by this study. One additional aspect that can easily be tested is the different product sizes and delivery schedules, which would provide more data for confirming the validity of the model in different use cases.

One area that can clearly be recommended for future research studies is the different placement and grouping aspects, which would also imply a model

with multiple shelves included in the calculations. Extending the model in this way would make it more useful for practical use, since the placement of the products is a meaningful part of shelf space planning in real-life situations, as are the other related factors.

Another future step after this study could also be a more extensive testing study of the same methods that were presented in this thesis. Even though several tests were conducted, the scope of the test data was still quite limited. It could be beneficial to perform tests on product groups from different categories besides cereals, with different sales volumes with higher or lower variance in the sales, etc. In addition, the models that were tested in this study could undergo even more extensive testing, with more fine-tuning of the parameters, since it was found that those can impact the results in a significant way. This would give a wider base of results to use in the shelf space planning process.

Bibliography

A. Angerer. *The impact of automatic store replenishment on retail: Technologies and concepts for the out-of-stocks problem*. Deutscher Universitäts-Verlag, 2006.

M. E. Biery. These industries generate the lowest profit margins. *Forbes*, 2017. URL https://www.forbes.com/sites/sageworks/2017/09/24/these-industries-generate-the-lowest-profit-margins/{#}3fcac23bf49d. Accessed: 2019-05-08.

P. Boatwright and J. C. Nunes. Reducing assortment: An attribute-based approach. *Journal of Marketing*, 65(3):50–63, 2001.

K. M. Bretthauer and B. Shetty. The nonlinear resource allocation problem. *Operations Research*, 43(4):670–683, 1995.

K. M. Bretthauer and B. Shetty. The nonlinear knapsack problem - algorithms and applications. *European Journal of Operational Research*, 138 (3):459–472, 2002.

V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985.

S. Clifford. Atlanta hats? Seattle socks? Macy's goes local. *The New York Times*, 2010. URL http://www.nytimes.com/2010/10/02/business/02local.html. Accessed: 2019-04-05.

M. Corstjens and P. Doyle. A model for optimizing retail space allocations. *Management Science*, 27(7):822–833, 1981.

K. Cox. The responsiveness of food sales to shelf space changes in supermarkets. *Journal of Marketing Research*, 1(2):63, 1964.

R. C. Curhan. The relationship between shelf space and unit sales in supermarkets. *Journal of Marketing Research*, 9(4):406, 1972.

R. C. Curhan. Shelf space allocation and profit maximization in mass retailing. *Journal of Marketing*, 37(3):54, 1973.

J. de Vicente, J. Lanchares, and R. Hermida. Placement by thermodynamic simulated annealing. *Physics Letters A*, 317(5-6):415–423, 2003.

Deloitte. Global powers of retailing 2018: Transformative change, reinvigorated commerce. *Deloitte*, 2018.

P. Desmet and V. Renaudin. Estimation of product category sales responsiveness to allocated shelf space. *International Journal of Research in Marketing*, 15(5):443–457, 1998.

X. Drèze, S. J. Hoch, and M. E. Purk. Shelf management and space elasticity. *Journal of Retailing*, 70(4):301–326, 1994.

R. W. Eglese. Simulated annealing: A tool for operational research. *European Journal of Operational Research*, 46(3):271–281, 1990.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayes restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

J. M. Hansen, S. Raut, and S. Swami. Retail shelf allocation: A comparative analysis of heuristic and meta-heuristic approaches. *Journal of Retailing*, 86(1):94–105, 2010.

D. S. Hochbaum. A nonlinear knapsack problem. *Operations Research Letters*, 17(3):103–110, 1995.

D. Honhon, V. Gaur, and S. Seshadri. Assortment planning and inventory decisions under stockout-based substitution. *Operations Research*, 58(5): 1364–1379, 2010.

A. Hübner and H. Kuhn. Retail category management: State-of-the-art review of quantitative research and software applications in assortment and shelf space management. *Omega*, 40(2):199–209, 2012.

W. Karush. Minima of functions of several variables with inequalities as side constraints. *M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago*, 1939.

H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems*. Springer Berlin Heidelberg, 2004.

J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, 1995.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

A. G. Kök and M. L. Fisher. Demand estimation and assortment optimization under substitution: methodology and application. *Operations Research*, 55 (6):1001–1021, 2007.

H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492. University of California Press, 1951.

A. Lim, B. Rodrigues, and X. Zhang. Metaheuristics with local search techniques for retail shelf-space optimization. *Management Science*, 50(1): 117–131, 2004.

M. K. Mantrala, M. Levy, B. E. Kahn, E. J. Fox, P. Gaidarev, B. Dankworth, and D. Shah. Why is assortment planning so difficult for retailers? A framework and research agenda. *Journal of Retailing*, 85(1):71–83, 2009.

S. H. McIntyre and C. M. Miller. The selection and pricing of retail assortments: An empirical approach. *Journal of Retailing*, 75(3):295–318, 1999.

C. C. Murray, D. Talukdar, and A. Gosavi. Joint optimization of product price, display orientation and shelf-space allocation in retail category management. *Journal of Retailing*, 86(2):125–136, 2010.

Nielsen. Päivittäistavarakaupan myymälärekisteri 2018, 2019. URL https://www.nielsen.com/fi/fi/press-room/2019/grocery-store-register-2018.html. Accessed: 2019-05-08.

Y. Nourani and B. Andresen. A comparison of simulated annealing cooling strategies. *Journal of Physics A: Mathematical and General*, 31(41):8373–8385, 1998.

United States Department of Agriculture. Retail trends, 2018. URL https://www.ers.usda.gov/topics/food-markets-prices/retailing-wholesaling/retail-trends.aspx. Accessed: 2019-05-08.

T. L. Urban. An inventory-theoretic approach to product assortment and shelf-space allocation. *Journal of Retailing*, 74(1):15–35, 1998.

P. J. M. van Laarhoven, E. H. L. Aarts, and J. K. Lenstra. Job shop scheduling by simulated annealing. *Operations Research*, 40(1):113–125, 1992.

M. H. Yang. Efficient algorithm to allocate shelf space. *European Journal of Operational Research*, 131(1):107–118, 2001.

C. Zhang and H. P. Wang. Mixed-discrete nonlinear optimization with simulated annealing. *Engineering Optimization*, 21(4):277–291, 1993.

F. S. Zufryden. A dynamic programming approach for product selection and supermarket shelf-space allocation. *The Journal of the Operational Research Society*, 37(4):413–422, 1986.

Appendix A

Complete Test Results

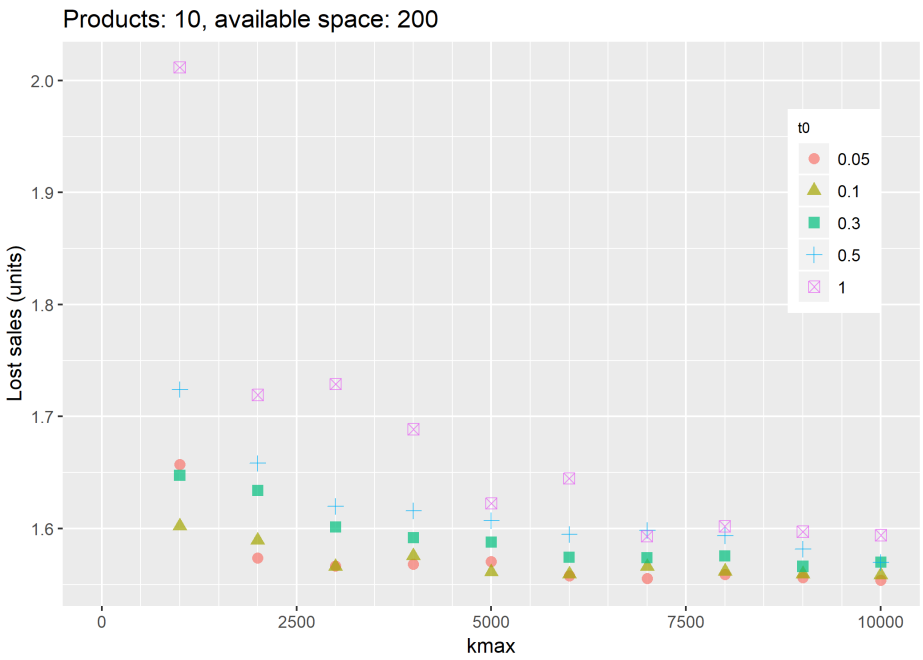

Figure A.1: Linear cooling schedule, static step size 1, different values for T_0

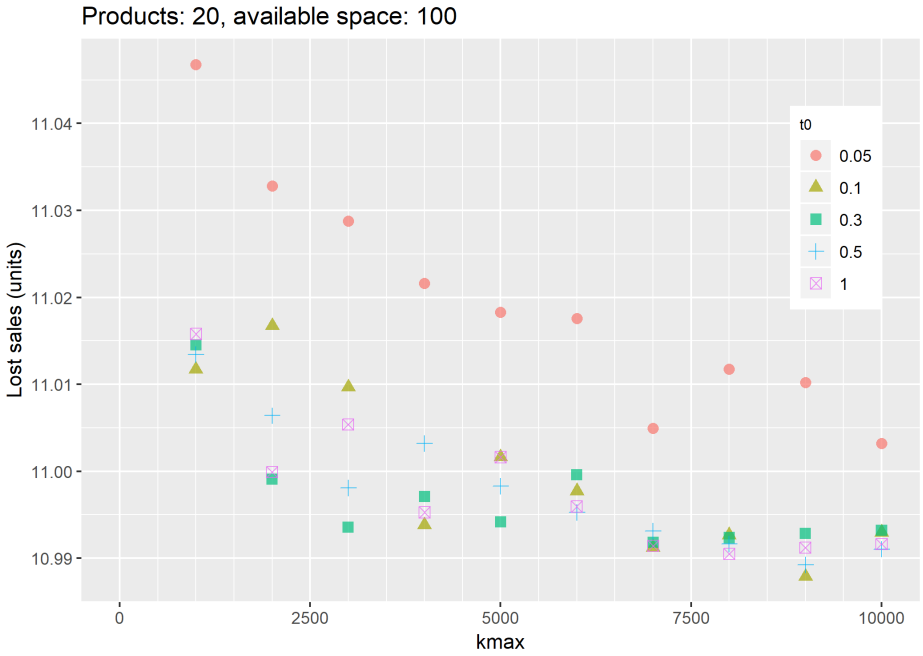

Figure A.2: Linear cooling schedule, static step size 1, different values for T_0

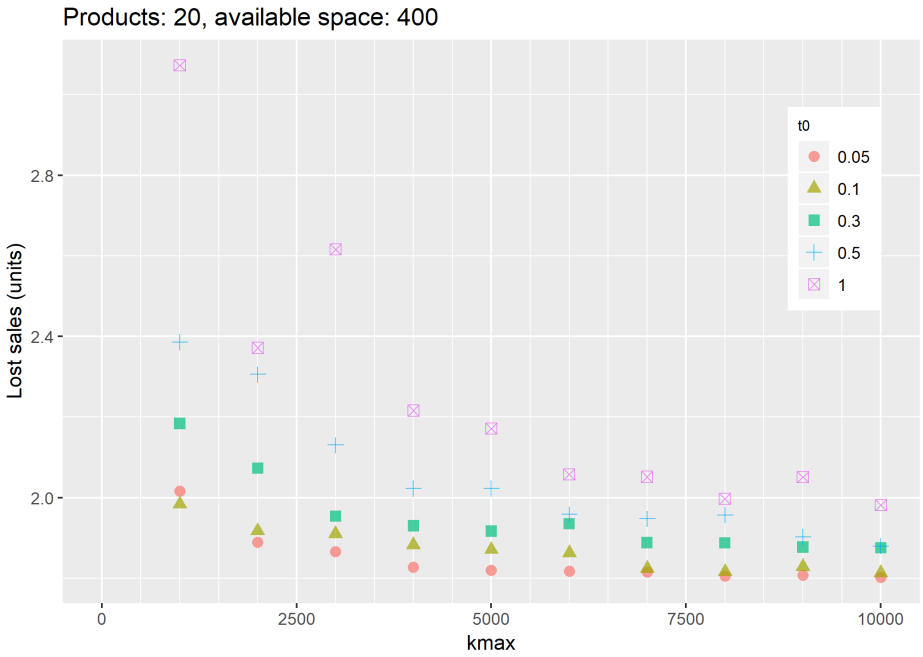

Figure A.3: Linear cooling schedule, static step size 1, different values for T_0

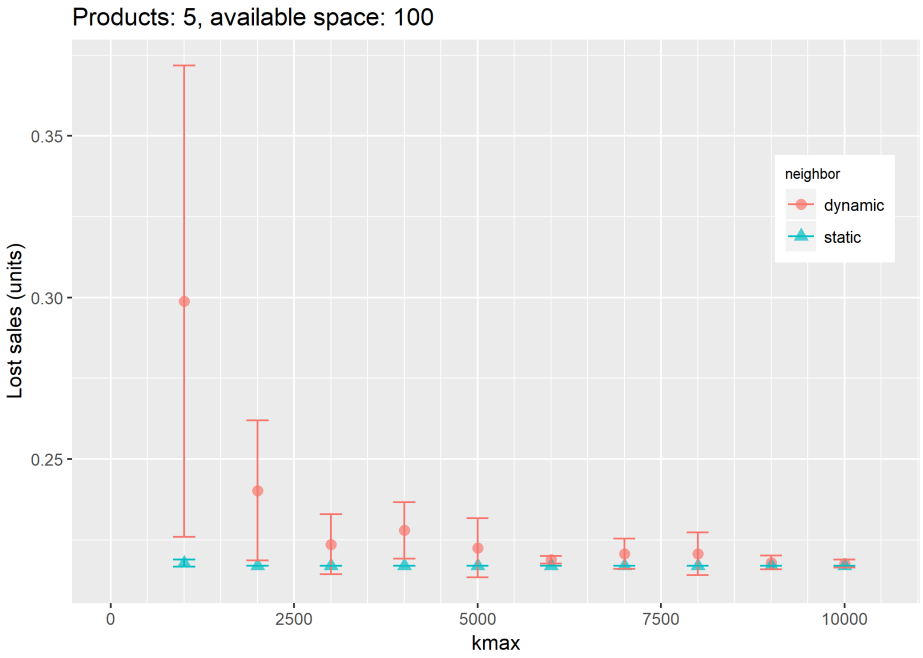

Figure A.4: Comparison between dynamic and static step size, linear cooling schedule

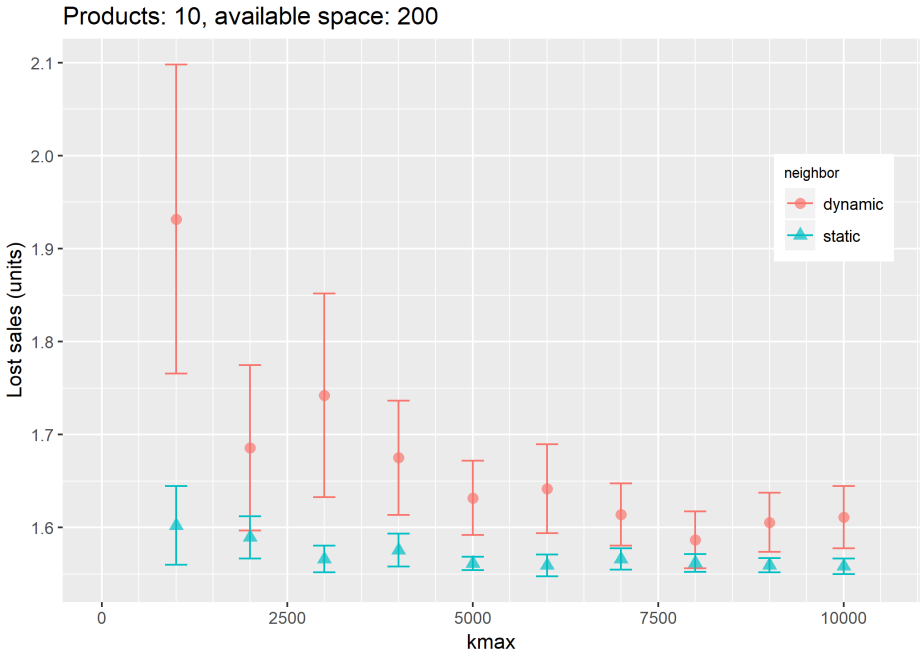

Figure A.5: Comparison between dynamic and static step size, linear cooling schedule

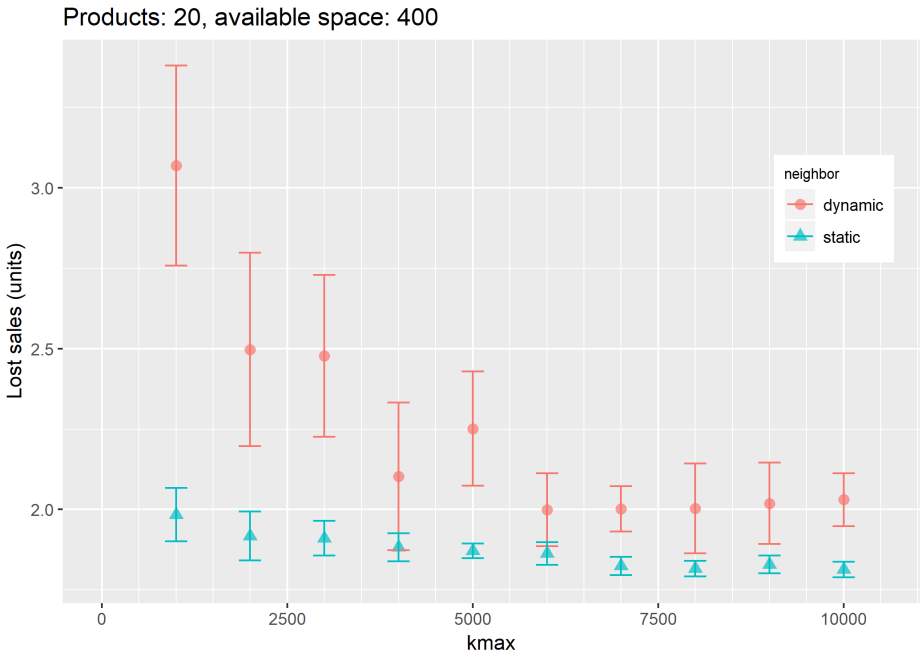

Figure A.6: Comparison between dynamic and static step size, linear cooling schedule

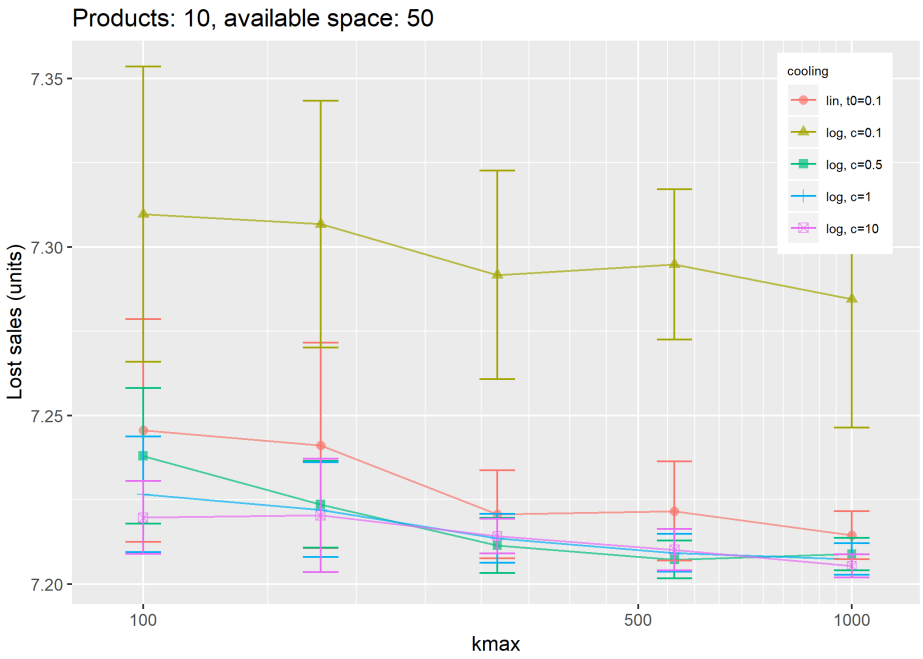

Figure A.7: Logarithmic cooling schedule with different values for the parameter c, linear cooling schedule for comparison ($\log_{10}$ scale).

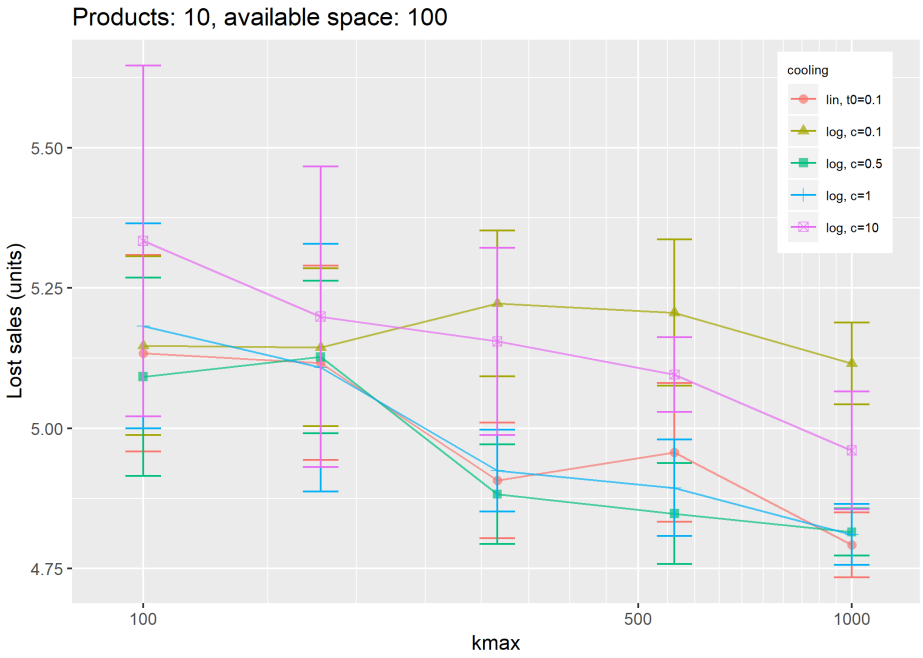

Figure A.8: Logarithmic cooling schedule with different values for the parameter c, linear cooling schedule for comparison ($\log_{10}$ scale).

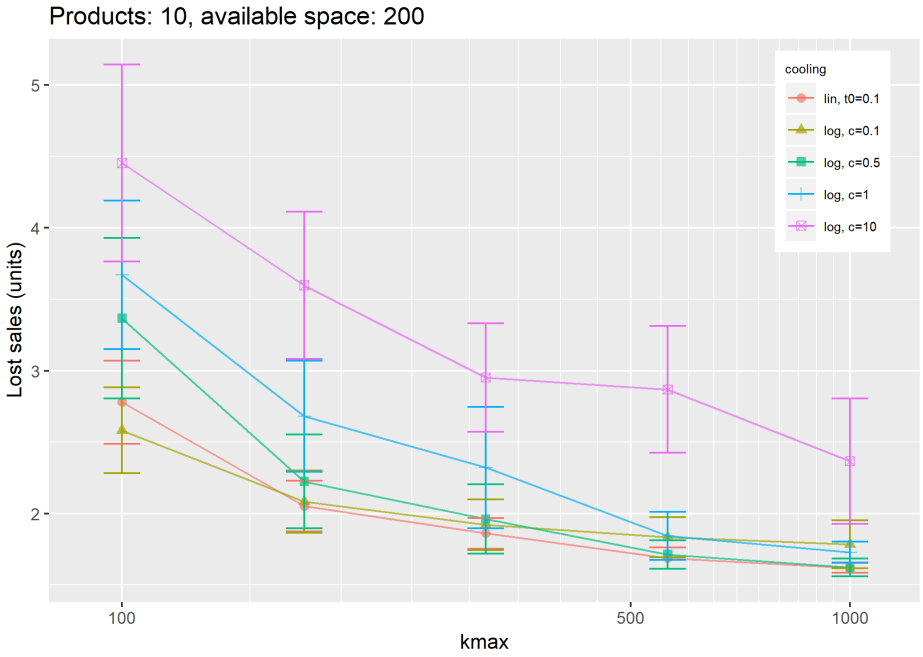

Figure A.9: Logarithmic cooling schedule with different values for the parameter c, linear cooling schedule for comparison ($\log_{10}$ scale).

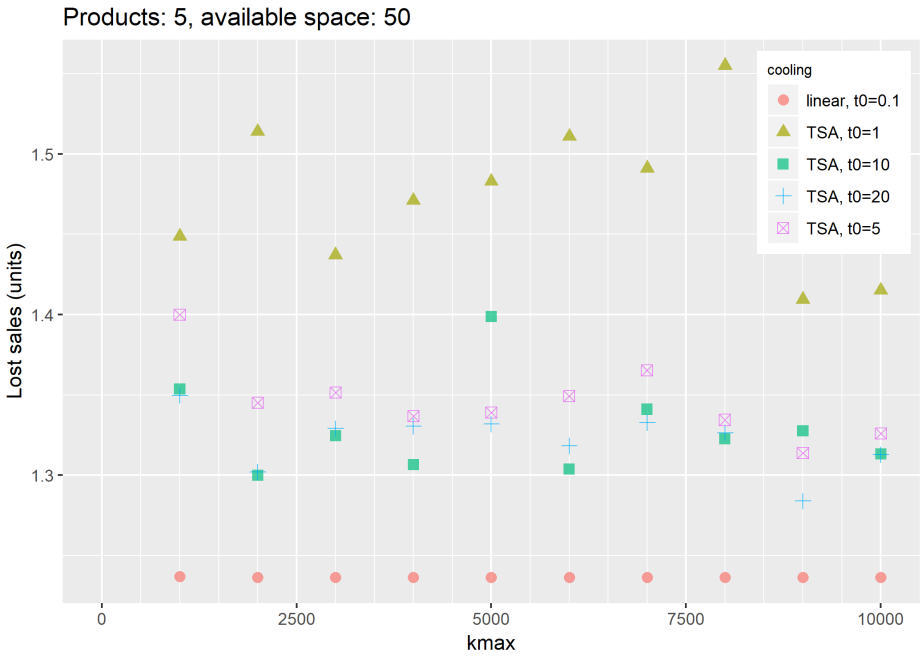

Figure A.10: Comparison of the lost sales results using the TSA and linear cooling schedules, with different values for the initial temperature T_0.

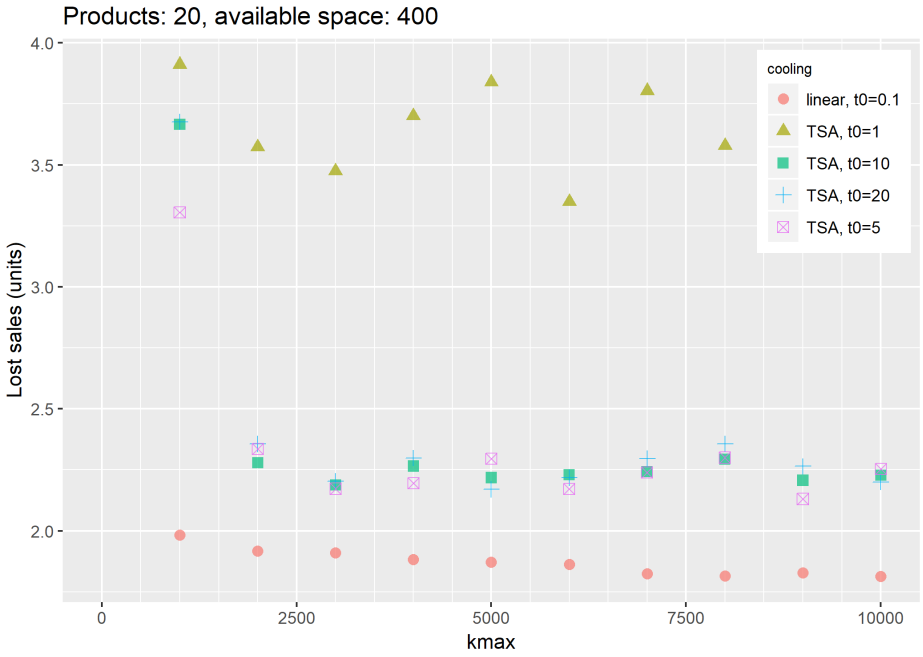

Figure A.11: Comparison of the lost sales results using the TSA and linear cooling schedules, with different values for the initial temperature T_0.

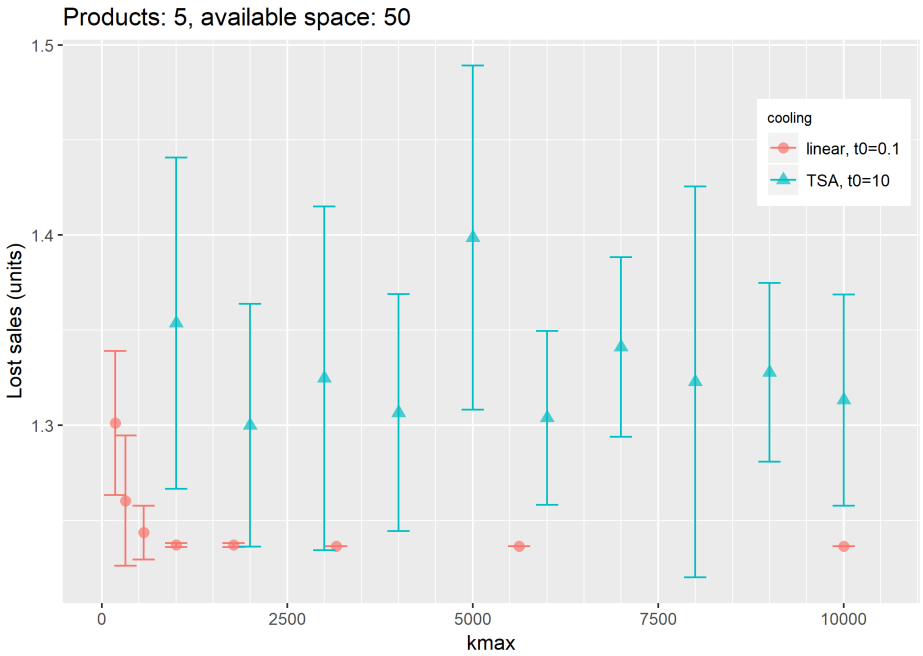

Figure A.12: Comparison of the lost sales results using the TSA and linear cooling schedules.

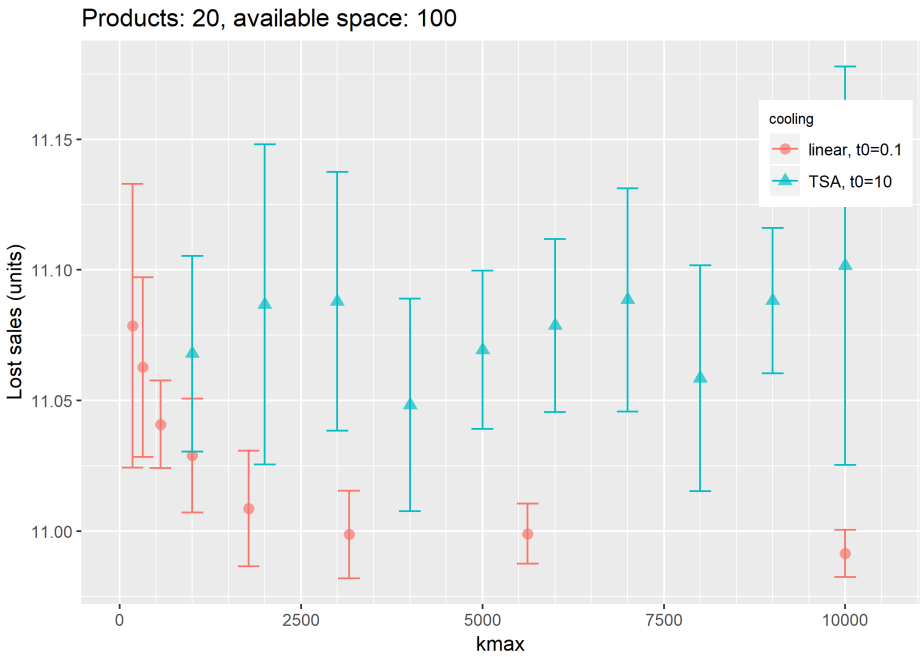

Figure A.13: Comparison of the lost sales results using the TSA and linear cooling schedules.

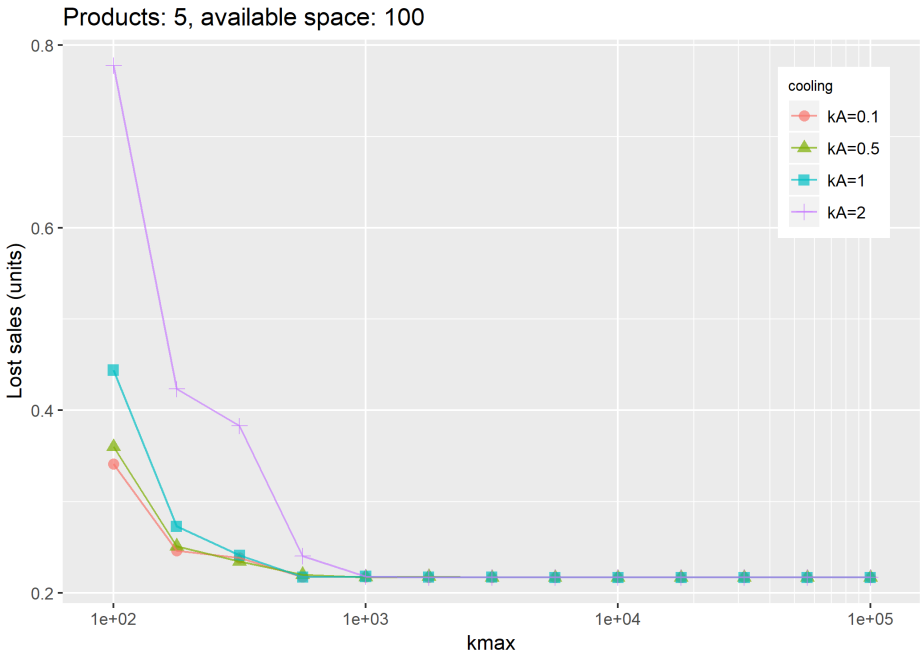

Figure A.14: Lost sales results using the TSA-linear combination cooling schedule with different values for the control parameter k_A ($\log_{10}$ scale).

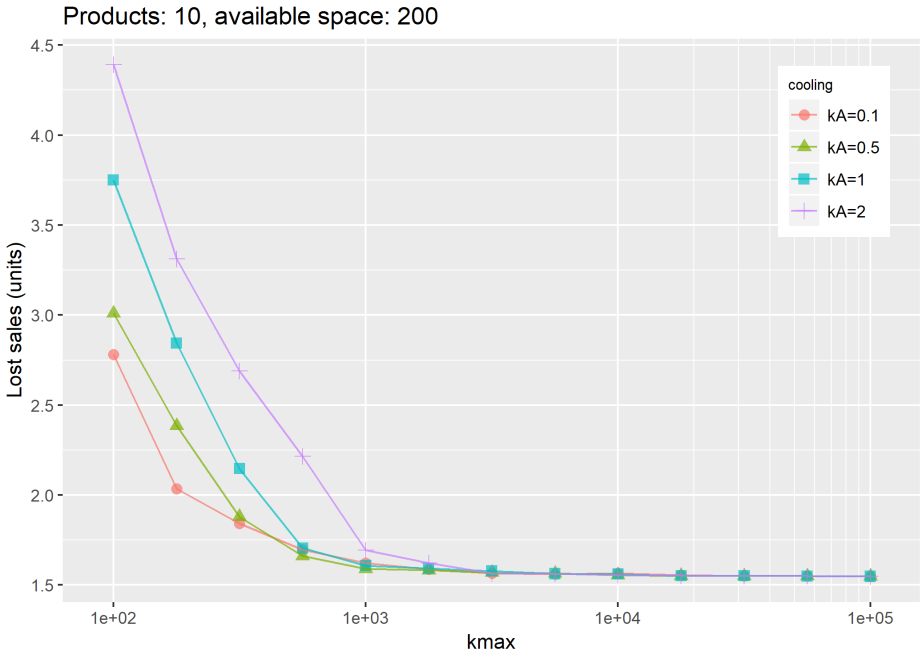

Figure A.15: Lost sales results using the TSA-linear combination cooling schedule with different values for the control parameter k_A ($\log_{10}$ scale).

Figure A.16: Lost sales results using the TSA-linear combination cooling schedule with different values for the control parameter k_A ($\log_{10}$ scale).

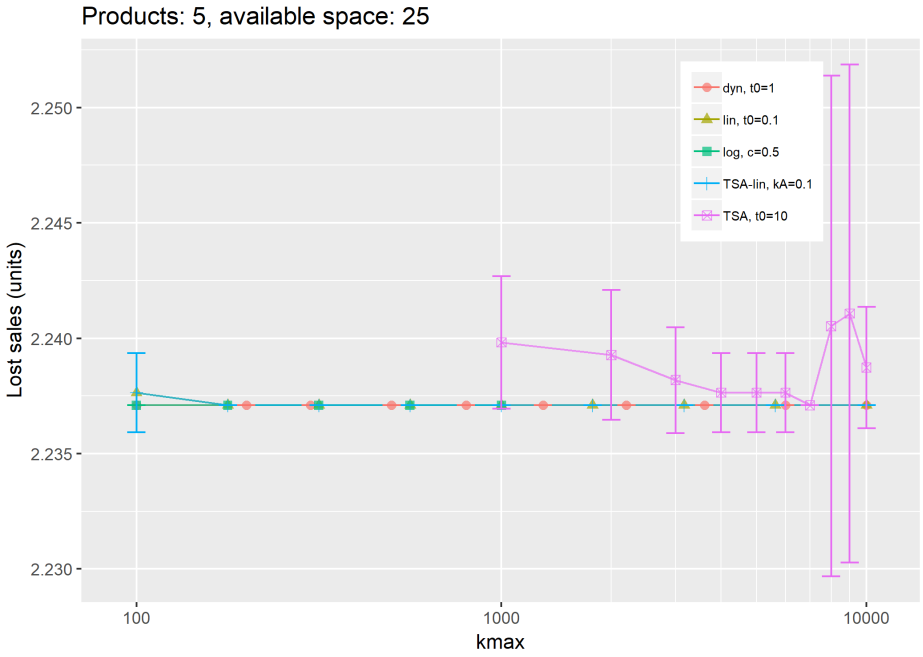

Figure A.17: Summary of all the different versions of the SA algorithm that were tested ($\log_{10}$ scale).

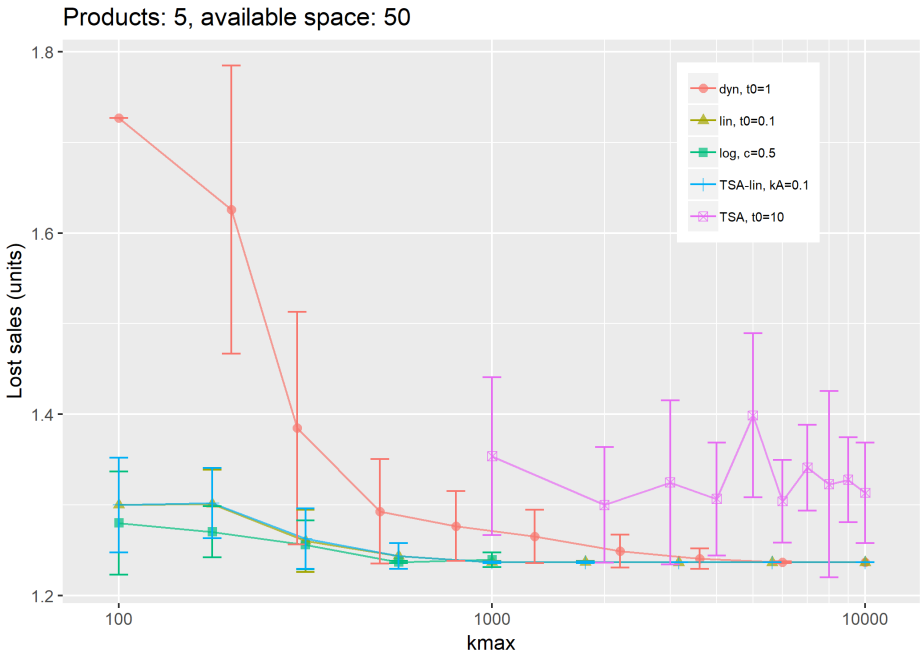

Figure A.18: Summary of all the different versions of the SA algorithm that were tested ($\log_{10}$ scale).

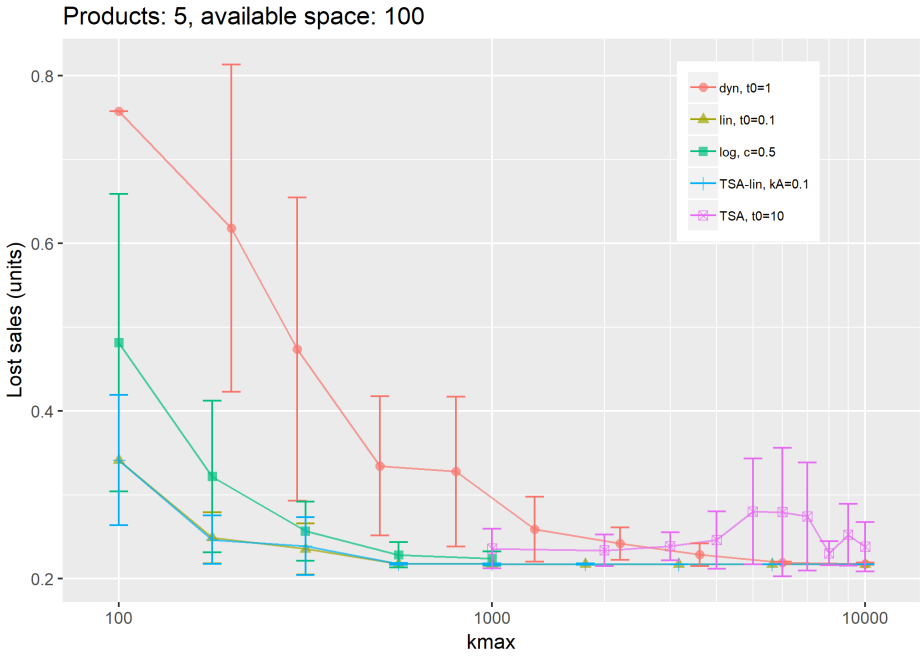

Figure A.19: Summary of all the different versions of the SA algorithm that were tested ($\log_{10}$ scale).

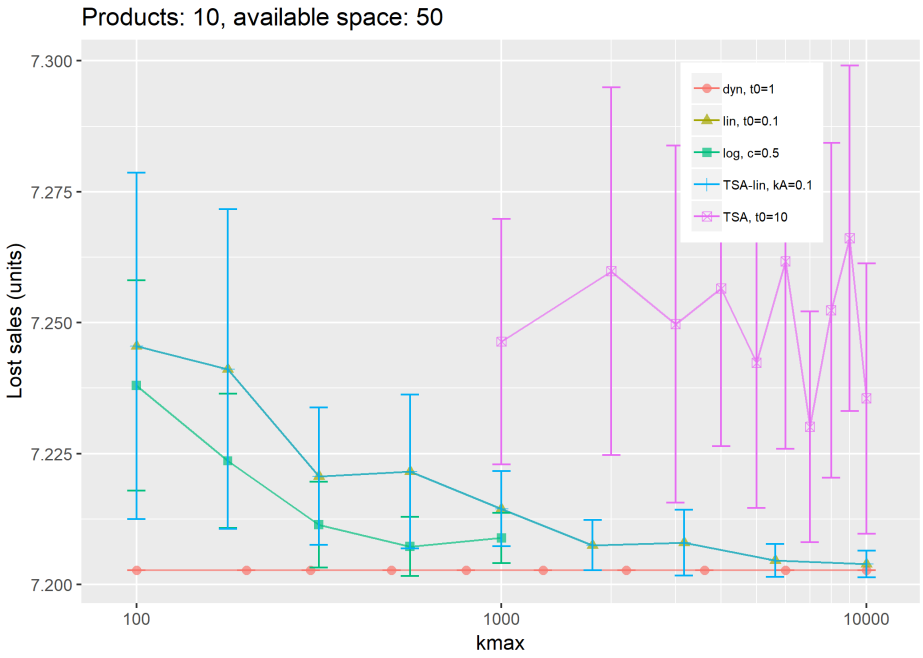

Figure A.20: Summary of all the different versions of the SA algorithm that were tested ($\log_{10}$ scale).

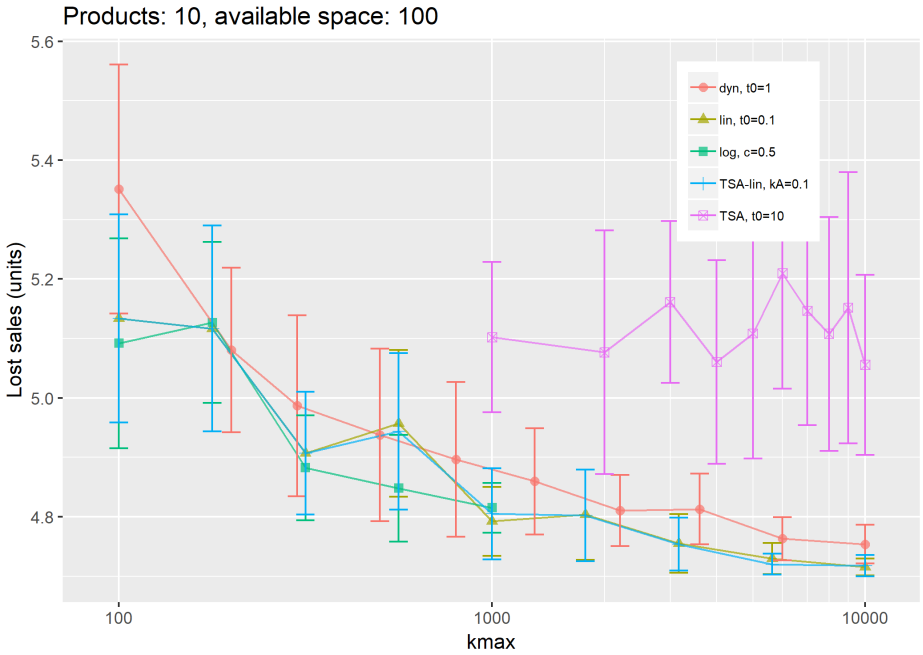

Figure A.21: Summary of all the different versions of the SA algorithm that were tested ($\log_{10}$ scale).

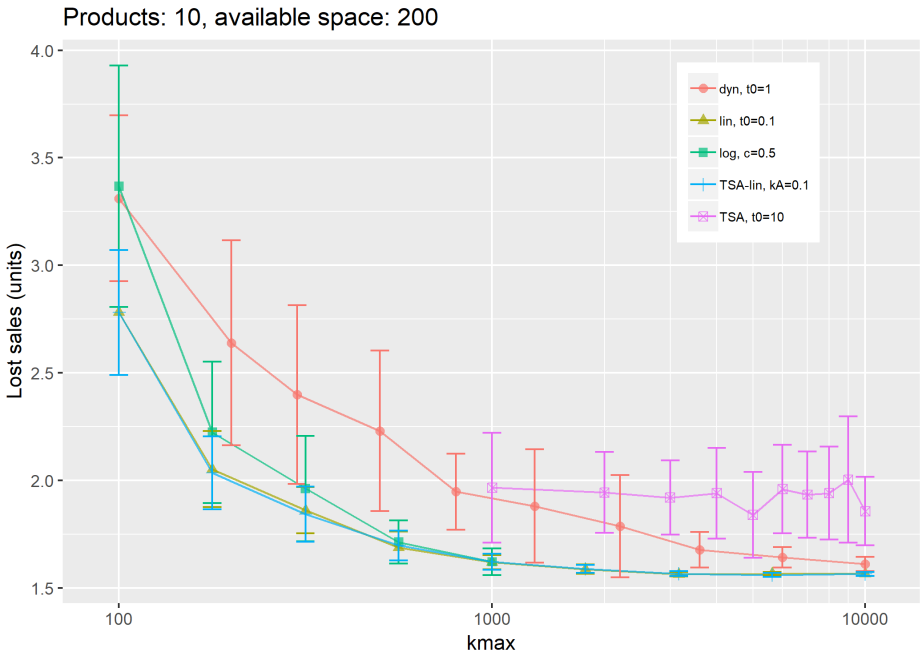

Figure A.22: Summary of all the different versions of the SA algorithm that were tested ($\log_{10}$ scale).

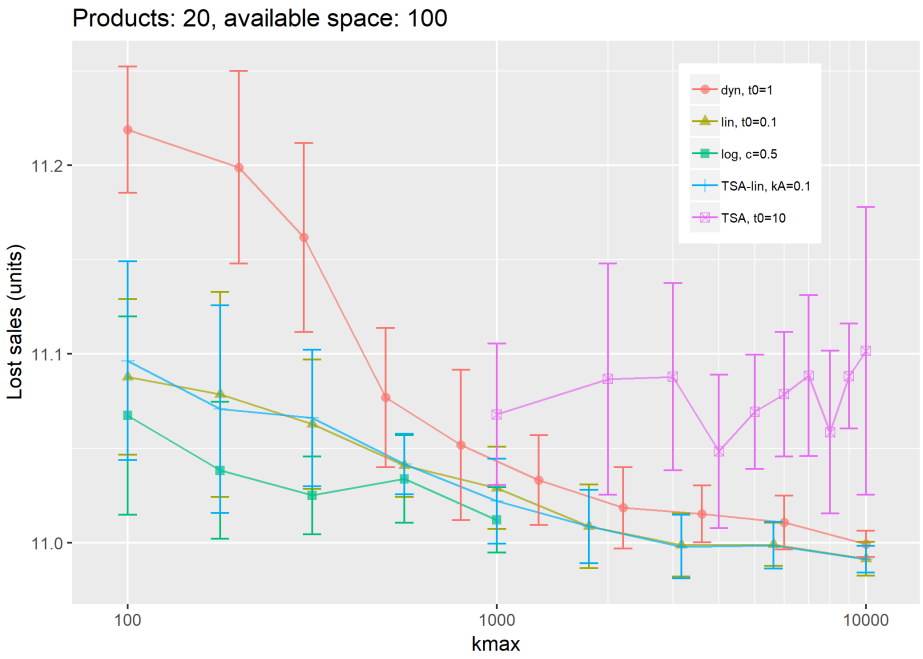

Figure A.23: Summary of all the different versions of the SA algorithm that were tested ($\log_{10}$ scale).

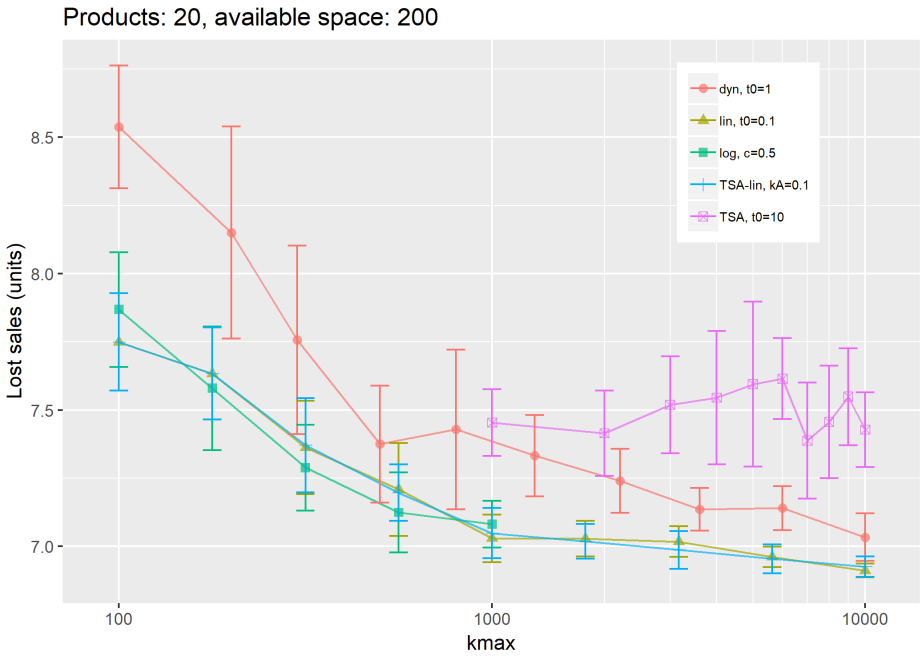

Figure A.24: Summary of all the different versions of the SA algorithm that were tested ($\log_{10}$ scale).

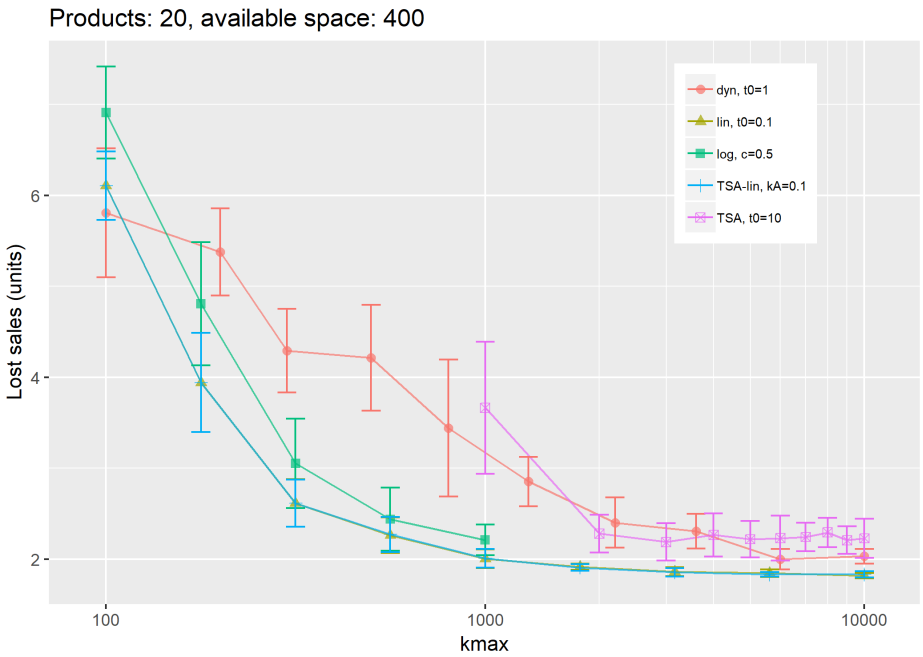

Figure A.25: Summary of all the different versions of the SA algorithm that were tested ($\log_{10}$ scale).